To the University of Wyoming:

The members of the Committee approve the dissertation of Evan M. Anderson presented on August 28, 2019.

Dr. Dimitri J. Mavriplis, Chairperson

Dr. Victor E. Ginting, External Department Member

Dr. Ray S. Fertig III

Dr. Yang Liu

Dr. Mark R. Garnich, University of Canterbury

APPROVED:

Dr. Carl P. Frick, Dept Chair, Mechanical Engineering

Dr. Paul A. Dellenback, Dean, College of Engineering and Applied Sciences

Anderson, Evan M., Development of an Open-Source Capability for High-Fidelity Thermoelastic Modeling and Adjoint-Based Sensitivity Analysis of Structures, Ph.D., Department of Mechanical Engineering, December, 2019.

The design and optimization of many advanced engineering systems requires high-fidelity modeling and simulation to adequately capture the physical behavior and provide meaningful results. Since high-fidelity simulations can be computationally costly, methods of design optimization that require a minimal number of simulations are desirable for such applications. Gradient-based optimization using objective sensitivities obtained using the adjoint method is an efficient approach for systems requiring high-fidelity simulations with a large number of design parameters. For this reason, it has become a common approach in many disciplines including aerodynamic and aeroelastic design and optimization.

An open-source, Fortran-based computational tool has been developed for finite element modeling of thermal and elastic response of structures, capable of obtaining adjoint-based sensitivities of solution-dependent objectives with respect to structural design parameters. The package is designed with the goal of tight coupling with analysis codes of other disciplines such as computational fluid dynamics solvers in mind, yet functions independently as a purely structural solver. Static and dynamic analysis can be performed, for either geometrically linear or nonlinear problems with a versatile library of three-dimensional elements for high-fidelity modeling. Exact sensitivities are obtained through direct differentiation of the analysis code.

Thermoelastic response is predicted for various test cases using the newly developed analysis tool and compared with analytical solutions and results from commercial finite element software for validation. Case studies are performed using the analysis tool to investigate topics of interest in the field of structural optimization, including fatigue load minimization in wind turbine blades, and the application of constraints to prevent structural instability and buckling. The tool is demonstrated to be effective and efficient in gradient-based optimization, and capable in multidisciplinary design and optimization.

DEVELOPMENT OF AN OPEN-SOURCE CAPABILITY FOR HIGH-FIDELITY THERMOELASTIC MODELING AND ADJOINT-BASED SENSITIVITY ANALYSIS OF STRUCTURES

by
Evan M. Anderson

A dissertation submitted to the University of Wyoming
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
MECHANICAL ENGINEERING

Laramie, Wyoming
December 2019

ACKNOWLEDGEMENTS

This work was made possible by several contributors of mentoring leadership, moral support and financial support. I most likely would not have returned to pursue my doctorate without the influence of my former advisor and enduring friend, Dr. Mark Garnich. During my undergraduate years, he found me a niche where I could use my skills effectively and gained interest in research and computational modeling. As a master's student he taught me a great deal of what I know about fundamental mechanics and structural analysis. During my years abroad, he kept me sharp by continuing to engage me in the ongoing work at the University of Wyoming, and when the opportunity arose, he ushered me back to pursue my doctorate. I am thankful for his mentorship and continued involvement in my career.

My doctoral advisor, Dr. Dimitri Mavriplis has been integral to my growth as well. His is a rare combination of achievement and humility, and the chance to work with him was an important factor in bringing me back to Wyoming for my Ph.D. My understanding of not only fundamental fluid mechanics, but also of advanced aspects of simulation and optimization methods has enriched greatly under his tutelage, and I am truly privileged to have had the opportunity to work with him.

I am thankful to the other members of my committee as well. Dr. Ray Fertig III has always had an enthusiasm for both excellent research and effective mentorship, and never shies away from a challenge. These are all qualities that I admire. Dr. Victor Ginting is one of the brightest professors I know. I have thought highly of him since my undergraduate years, and am grateful for his ongoing willingness to serve on my committee. Dr. Richard Schmidt and Dr. Yang Liu, though both serving for only a portion of the program, provided valuable input and participation, and their time is much appreciated.

My fellow graduate students have also provided valuable assistance. Andrew Kirby was greatly helpful in the setup of my workstation and getting situated in the lab. Faisal Bhuiyan's input was critical in the study of fatigue in wind turbine blades. The post-doctoral and research staff in the department have been a great resource of insight and experience.

My wife, Vicky has been supportive and patient every step of the way, willing to do what was necessary for me to finish my doctoral work as we began to raise our daughter, Miana. There were some challenging times, but having someone special in my corner and standing by me no matter what made more difference than she will ever know. I love you my 小猪猪.

Funding for this work was provided by AFOSR STTR contract FA9550-0035, the U.S. Department of Energy, Office of Science, Basic Energy Sciences, award DE-SC0012671, the NASA University Leadership Initiative (ULI) grant NNX17AJ95A entitled “Advanced Aerodynamic Design Center for Ultra-Efficient Commercial Vehicles”, and the University of Wyoming’s College of Engineering and Applied Sciences Energy Graduate Assistantship. Computational resources have been provided by the University of Wyoming Advanced Research Computing Center (ARCC) and by the NCAR-Wyoming Supercomputer Center (NWSC).

Table of Contents

Chapter 1

Background and Motivation

High-performance computing, simulations and sophisticated data processing are an increasingly integral part of modern science and technology. Computational science has come to be considered a third major branch of scientific discovery, joining the ranks of theory and experimentation [1]. Advanced computational tools for prediction and analysis are now prevalent in countless fields, including but not limited to meteorology, economics, business, medicine, physics, chemistry, mathematics and engineering. The algorithms and methods employed for such applications are as diverse as the fields themselves, and an exhaustive look is beyond any single work.

In the context of engineering and design, one ultimate goal for predictive computational tools is to improve the performance of systems, and to obtain new designs that are optimal from the standpoint of certain objectives. Some examples of common objectives are aerodynamic performance, energy efficiency, maximum ratio of structural strength and durability to weight/mass, and cost effectiveness. These objectives are often at odds with one another, and there is an ever-present challenge to find the best balance of goals within an infinite domain of possible designs.

The ability to identify an optimal design is predicated on the ability to predict the behavior and performance of a system accurately. In practice this is rarely possible by analytical means, and so numerical methods and computational tools must be relied upon. Since computer resources are finite, the task of analyzing complex systems to a sufficient degree of accuracy at an affordable cost has challenged the field for decades. Even now, with computer technology widely accessible, and orders of magnitude beyond what it was only thirty years ago in both speed and storage capacity, engineers and designers find themselves hard-pressed to obtain the fidelity desired in capturing physical phenomena. It therefore remains a continuing goal to not only maximize the capabilities of computing resources, but also find methods of running simulations and optimizations as efficiently as possible.

The present work documents the development of an open-source computational tool dedicated to the goal of high-fidelity structural finite element modeling and efficient gradient-based optimization. The primary intent is for tight coupling with other modeling packages for multidisciplinary analysis and optimization, but it runs independently as well. The remainder of this chapter provides background and historical context regarding four specific topics relevant to the motivation for this work. The first two topics are broad, pertaining to methods of modeling and optimization of physical systems in general. The second two are more focused, relating to specific topics of interest for this particular work.

1.1 High-Fidelity Multidisciplinary Optimization

In a sense, optimization defines the work of an engineer or designer. As long as humanity has employed technology in any capacity, there has been a constant process of modification and improvement of the tools and devices we use. Designs are conceived, built and tested, shortcomings and potential for improvements are identified, and the designs are revised and modified in an endless cycle in the pursuit of excellence. Modern computer technology was absent through most of the history of this process, and great achievements were made nevertheless. To this day there remains no substitute for the ingenuity of the human mind.

Still, the classical approach to design has difficulties that can be greatly alleviated with the assistance of computer technology. The behavior of complex systems involving multiple interacting phenomena is difficult to predict exactly on mere experience and intuition, and simplifications and estimates must be made, leading to a great deal of trial and error. It is very expensive and time-consuming to physically build and test a large number of designs, so finding truly optimal solutions through testing alone is nearly impossible. The designer must then settle for the best design tested within the range of time and resources allotted to development [2]. But with high-fidelity computer simulations and optimization, not only can designers capture the complex behavior of multidisciplinary systems, but numerous designs can be analyzed without the cost of fabrication, greatly increasing the potential for enhancing designs.

In the field of aviation, it has always been a fundamental design challenge to balance the interests of aerodynamic performance and structural integrity. Early airplane designs were driven largely by structural considerations [3]. The biplane configurations common in the early 20th century were motivated by an interest to reduce wing loads and promote structural stability with reinforcing cables between the two wings, much at the expense of aerodynamic performance. Later with advancements in materials and design analysis, the monoplane configuration emerged as an aerodynamically superior alternative.

But along with the single wing section and broader wing spans of the monoplane design came increased flexibility and higher wing deflections due to aerodynamic loads. The coupled interdependence between structural displacements and aerodynamic loads became a significant consideration in design. The standard approach to addressing aerodynamic wing deflection is to design the wing in a *jig* shape, so that when it deforms under the loads of normal operating conditions, the ideal aerodynamic shape is attained. The advantage of this approach is that the aero-structural interdependence becomes effectively one-sided, in that the structural response depends entirely on the aerodynamic loads, but the aerodynamic analysis is affected only by structural weight. This enables a two-level approach to optimization, with structural optimization "nested" within each cycle of aerodynamic optimization, avoiding the cost of accounting for the fully-coupled phenomena directly [4,5].

The jig-shape approach has the limitation that it must be targeted to a single operating condition, usually the conditions at cruise, where the greatest amount of time and fuel is expended. But designs optimized for one specific condition can often perform poorly at other conditions. For the most robust and well-rounded designs, fully coupled multipoint aeroelastic optimizations must be used.

Toward the latter half of the 20th century, the value of fully integrated multidisciplinary optimization began to be recognized. In 1988, Grossman *et al*. [6] performed an investigation using a combination of lifting-line aerodynamics with finite element beam models in aero-structural optimization. Although their methods were low-fidelity and somewhat simplistic, they demonstrated that superior designs could be attained through multidisciplinary optimization than when examining the disciplines individually. In a 1994 study, Dudley *et al*. [7] used a technique called variable complexity interlacing to streamline the coupled modeling process and perform a more sophisticated aero-structural optimization of a high-speed civil transport vehicle, with 26 aerodynamic design variables, and 40 structural design variables. Still, their approach neglected the effect of aerodynamic shape variables on the actual structural weight, leaving some inaccuracy in the results.

Over the past three decades, the capability of computing technologies has advanced vastly, making high-fidelity modeling and optimization increasingly accessible to engineers and researchers. Yet challenges continue to be faced in obtaining reliable results when complex physics are involved. Lyu *et al.* [8] performed aerodynamic shape optimization of an ONERA M6 wing, using both Reynolds-Averaged Navier Stokes (RANS) and Euler-based solvers. They found that the optimized shape was significantly different between the two flow solvers, and that the shape obtained using the Euler equations in fact performed worse than the baseline shape when tested using the RANS solver. They concluded that optimization results can be highly dependent on the assumptions used, and that any relevant physics excluded from an analysis tend to be exploited in the results. This reinforces the importance of not only high-fidelity modeling, but also the consideration of all participating factors affecting a system in optimization analysis.

Current trends in aerodynamic design call for high-fidelity multidisciplinary optimization more than ever. There is a constant push for improved fuel efficiency in aviation, and a number of advancements in design are emerging in an effort to address the issue. One example is the pursuit of fixed-wing commercial aircraft designs with slender, high-aspect ratio wings enabled by high-performance materials to reduce drag [9-12]. As was the case with the original introduction of the monoplane, an increase in wingspan and aspect ratio is inevitably accompanied by stronger interdependence between aerodynamic load and wing deflections. In a recent study, Brooks *et al*. [10] developed a modified version of the NASA Common Research Model (CRM), to represent the undeflected shape of the wing for the purposes of multidisciplinary aeroelastic optimization. They developed two models, one with an aspect ratio of 9, and a second with a high aspect ratio of 13.5 to represent a more futuristic geometry. They found that the high aspect ratio wing performed poorly when given an initial distribution of cross-sectional twist and structural sizing parameters similar to the model with aspect ratio 9, due to the increased flexibility of the wing, and that a fully coupled aeroelastic optimization was required to obtain a well-performing design.

Another example of current advancements toward high-efficiency aircraft is a NASA-funded project presently underway to implement a slotted, natural-laminar-flow (SNLF) wing design [13-20]. The airfoil design incorporates two elements, one fore and one aft, separated by a narrow slot so that the flow promotes a favorable pressure gradient, and ideally preserves laminar flow over the entire upper surface of the fore element. The onset of turbulence is delayed until near the trailing edge of the aft element, thereby simultaneously reducing drag and enhancing lift. A conceptual illustration is given in Fig. 1.1. The goal of the project is to reduce fuel burn in commercial aircraft by 70% compared to a 2005 baseline.

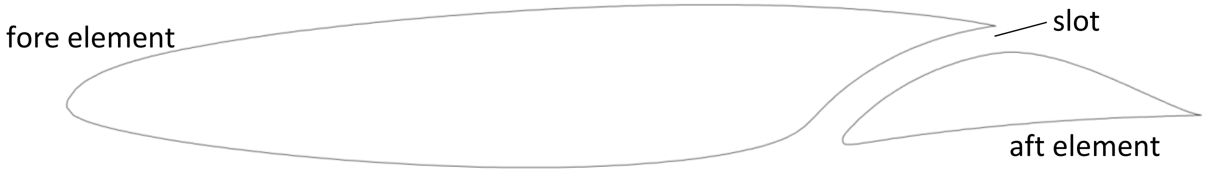


Figure 1.1 Conceptual depiction of the slotted, natural-laminar-flow airfoil design (ref 18).

The SNLF airfoil is a novel design with a great deal of potential to impact the aviation industry, but there are many challenges involved in the effective simulation and optimization of this configuration. The very nature of the design concept necessitates the accurate prediction of the onset of turbulence in the flow over the wing. But RANS-based CFD simulation tools that have become the standard in high Reynolds number applications typically assume fully turbulent flow, and must be augmented with special *transition* models [19,20] to capture the physics and predict the flow behavior accurately.

From a structural standpoint, the unique geometry of the SNLF design may require unconventional topology in the internal wing box structure, and additional connectors/actuators governing the interaction between the fore and aft elements. These features may affect both the flow between elements and the structural loads in ways that call for detailed resolution in the fluid and structural models.

These are just a few examples that highlight the need for high-fidelity and multidisciplinary analysis in the optimization of systems, especially in complex aerodynamics applications. Some further implications regarding appropriate methods for such cases will be discussed in the next section.

1.2 The Adjoint Method in Gradient-Based Optimization

Having established the importance of optimization in the field of engineering and design, the next question is how to go about the optimization process. There are numerous methods and algorithms with which to perform automated optimization, owing to the wide variety of contexts and applications. Genetic algorithms are popular for many structural applications, and any case in which it is desired to perform global-level optimization on a system with design variables of a discrete, rather than continuous nature, such as topology optimization [21,22]. Machine learning algorithms have had a great amount of interest lately, in which a set of sample data can be used to "train" an algorithm to recognize the difference between a favorable and unfavorable configuration based on some criteria. A search through the design space of the system can then be performed to identify optimal solutions [23]. Machine learning is particularly useful for finding patterns and correlations in stochastic, nondeterministic sets of data.

While these methods are powerful in their proper contexts, they have limitations in applications requiring high-fidelity simulations. Often, genetic and machine learning algorithms require hundreds or thousands of objective function evaluations to assess the favorability of numerous states within the design space. But this can be very costly if a function evaluation requires expensive, high-fidelity simulations. As discussed in the previous section, high-fidelity multi-disciplinary simulation is often required for the most meaningful results in the field of aerodynamics and aeroelastic analysis. Plus, the design space for such problems often consists of continuous design variables, like shape and sizing parameters, making the aforementioned algorithms less suitable.

Gradient-based optimization is a well-suited alternative for high-fidelity applications, in that it usually requires relatively few objective function evaluations (typically between 10 and 50) to find an optimal solution [2]. Gradient-based algorithms are inherently local optimizers, seeking local optima in the neighborhood of the initial design configuration, and are not guaranteed to find the absolute optimum in an entire design space. Nevertheless, if the objective and design space are well posed, with a reasonable initial configuration this is usually not of great concern, since absolutely optimal solutions are rarely guaranteed in practice with any type algorithm.

There is, however, an important consideration in implementing gradient-based optimization, in that it requires obtaining the gradient, or sensitivities of the objective function at each design cycle. Some of the most intuitive ways of doing this, such as using finite difference approximations, require the solution of governing equations for every design variable at each design cycle, which can be very costly for problems with a large number of design variables. This was a substantial hurdle in the field of design for many years. One of the earliest works implementing CFD analysis fully integrated with automatic optimization algorithms for aeronautical wing design was that of Hicks and Henne [24,25]. Although their work was pioneering, the cost of obtaining sensitivities in their analysis was a major obstacle. Borland *et al.* [26] performed fully coupled aero-structural optimization on a commercial aircraft wing, but were limited in their design space to three aerodynamic design variables and 20 structural design variables, due to computational cost.

Eventually, the problem of design optimization came to be posed from the point of view of control theory [27], in which the objective sensitivity is found with an indirect order of operations, by first solving a single *adjoint* equation, derived from the governing equations of the system. The objective sensitivities can subsequently be found for an arbitrary number of design variables with negligible additional cost. Using this approach, a gradient-based optimization algorithm can be implemented for the cost of only two solutions of the governing equations at each design state, regardless of the number of design variables.

Needless to say, the benefits of using the adjoint approach in obtaining objective sensitivities in high-fidelity applications with numerous design variables is substantial. The concept was first applied to problems governed by elliptic equations by Pironneau [28]. For Euler-based compressible transonic flow applications in aerodynamic design, the pioneering work was by Jameson [29-32]. The adjoint method was proven very effective for the design of airfoils and three-dimensional wing design [33-37], and was eventually extended to RANS-based CFD applications [38], upon observing that viscous effects should not be ignored in transonic-flow wing design.

In the years following, the adjoint method has become commonplace in the field of aerodynamic design [39-42], and it is permeating into other applications as well. Gradient-based optimization with adjoint-based sensitivities is a powerful tool for high-fidelity applications with smooth objective and continuous design variables. For multidisciplinary applications, such as aeroelastic optimization, the capability of obtaining sensitivities using the adjoint method must be present in all modeling disciplines involved. For this reason, the structural modeling capability developed in this work was mandated to have support for obtaining adjoint-based sensitivities, as is described further in Chapter 4.

1.3 Longevity Optimization of Wind Turbine Structures

There are few objectives that have received more focus and attention in the global research community in recent years than the development of renewable energy technologies and infrastructure. Though the importance of the problem is widely acknowledged, the challenges of shifting to a renewables-based system are equally vast in both difficulty and magnitude. A critical factor in the viability of any renewable source is cost-effectiveness. Whether a technology is competitive in the market mainly boils down to levelized cost of power and rate of return of investment on installation and maintenance of systems [43]. The factors determining this are complex and constantly changing, and highly dependent on the technology.

Wind energy is currently one of the most prominent forms of renewable energy, and has great potential to sustainably meet a large portion of global energy needs. For wind technology, two important factors that determine the levelized cost of power are annual energy output, and operating lifespan of turbine structures. The operating lifespan of a dynamic structure is largely determined by its resilience against breakdown and failure. Wind turbines experience repeated, cyclic loads in a variety of forms including aerodynamic, gravitational and centrifugal forces. Minimizing the accumulation of damage from these loads, while maximizing power output would then serve to optimize the cost effectiveness of a wind power system.

A significant part of managing the loads and power output of a wind turbine is enabling them to adapt to various wind conditions. Wind speed and direction varies in any location over time, and the optimal blade configuration in the interest of both power output and blade loading is dependent on these conditions. Most large-scale wind turbines are built with active control systems that adjust the pitch of the blade mechanically as wind conditions change. But designing control systems responsive enough to keep up with sudden gusts and the full range of wind velocities seen is difficult, especially with the increasingly large turbine sizes that the industry is trending toward [44]. There can also be additional cost and maintenance problems associated with active control systems.

Numerous authors have approached the problem of optimization of wind blade loads and power output from the perspective of *passive* adaptation. The idea is that if a turbine blade's internal structure and material layup is designed so that the blade naturally twists about its longitudinal axis as it bends under aerodynamic loads, then favorable aerodynamic and structural conditions can be preserved through a range of wind conditions. This phenomenon is known as bend-twist coupling.

Lobitz and Veers [45] employed finite element beams with analytical bend-twist formulations to minimize blade loads leading to fatigue. They concluded that fatigue damage could be significantly reduced through bend-twist coupling, although their structural definition and fatigue model were relatively low-fidelity. Vesel and McNamara [46] used a genetic algorithm to minimize levelized cost of energy through bend-twist coupling. They defined 83 design variables, and used an analytical bend-twist coupling model, with a 2D panel code to predict aerodynamic loads. The optimization results indicated a decrease in blade loads while preserving power performance. Zhou *et al*. [47] examined the performance of wind blades with varying composite ply angles. They found that an optimal choice of ply angle could reduce blade loads by 10% while also improving power capture. They also noted that analytical blade modeling based on laminate theory was more efficient but less accurate than finite element modeling.

The cost of high-fidelity multidisciplinary optimization loomed as a consistent theme over all these works, and some simplifying assumption is almost always made in the interest of computational savings. Maheri *et al.* [48] used a combined finite element and analytical approach, where the results from a high-fidelity finite element analysis of baseline reference configuration was extrapolated to other similar cased using certain analytical assumptions. They found that this technique reduced cost by 95%, with only 1% error in power output predictions within the scope of conditions they examined. While acknowledging the cost of high-fidelity analysis, they also noted that closed form predictions of induced twist could be inadequate. Another study by Murray *et al.* [49] performed experimental verification of finite element prediction of composite laminates. Results showed that the bend-twist behavior of composite structures was sensitive to the ply angle and thickness, and that high-fidelity modeling would likely be required for turbine structures outside of normal operating conditions.

In recent years, structural optimization using the adjoint method has been surfacing in more and more studies as well. Oest *et al.* [50] employ adjoint-based sensitivities in the optimization of the jacket structure at the base of an offshore wind turbine. Their optimization reduced the mass of the structure by 40%, while maintaining constraints on the fatigue limit. This study focused only on the jacket structure, however, and did not make any consideration of the structure of the turbine itself.

A review of the literature with regard to optimizing the lifespan of wind turbines seems to call for high-fidelity prediction of fatigue damage and associated metrics. A challenge lies therein, since wind turbine structures are largely composed of fiber-reinforced composite materials, for which prediction of fatigue has long been notoriously difficult [51,52]. In order to maximize fatigue life, a means of quantifying the driving force behind fatigue damage must be identified. Previous researchers have shown that fatigue in fiber-reinforced composites is predominantly matrix-driven, and that the physics of bond-breaking and damage propagation in the polymer matrix can be appropriately modeled with the kinetic theory of fracture [53-57]. The *off-fiber-axis* stresses, or shear stresses and normal stresses perpendicular to the fiber direction, are mainly responsible for such damage propagation. Fertig *et al.* [58-61] identified a scalar stress criterion representing the effective off-axis stress in the matrix of a unidirectional fiber composite, and demonstrated its utility as the driving stress in the kinetic theory of fracture.

Though many works as previously mentioned have examined the mitigation of loading on wind blades, there has never been a study posing the effective matrix stress derived by Fertig as an objective in gradient optimization. Many (though not all) previous studies do not capitalize on the efficiency of adjoint-based sensitivities for problems with many design variables. One study of this work employs the developed adjoint-based structural optimization capability to investigate the potential to alleviate fatigue damage in a wind turbine blade by tailoring composite fiber angles to minimize the Fertig off-axis matrix stress criterion. The study is documented in Chapter 6, Section 6.1.

1.4 Buckling Constraints in Structural Optimization

In the analysis and optimization of many structures, it can be important to consider the possibility of buckling. Under some circumstances structural instability can cause a structure, or part of a structure, to excite an undesirable mode of deformation in an accelerated manner, potentially leading to failure. This is particularly true with long, slender structures such as wings and wind turbine blades. There is renewed interest in the accurate and efficient analysis of buckling in structural design, in part because of current trends in aeronautics toward more slender, high aspect ratio wing designs which capitalize on high-performance materials and novel configurations in the interest of reducing drag and minimizing fuel burn [9-12]. An inevitable goal of such designs is to minimize structural mass, while preserving stiffness and strength. But optimizations focused solely on mass reduction and stress constraints have a tendency to produce designs with poor stability [22]. It therefore becomes more important than ever to consider buckling constraints in the optimization process.

A challenging aspect about incorporating buckling as an optimization constraint or objective is that it is an inherently structural-level phenomenon, which can occur either locally in individual members of an assembly, or globally in the structure as a whole. It can occur long before the onset of any yielding or material-level failure, and cannot be detected by simply scanning through a finite element structure at a given state and examining some aspect like stress at each individual point. The geometry and state of the entire structure must be considered, and some quantifiable criterion indicating the risk of buckling must be identified. For gradient-based optimization, this criterion should preferably be smooth and differentiable. The problem is particularly challenging in topology optimizations, in which prominent buckling modes are difficult to identify or estimate pre-analysis [22,62].

Two main approaches to incorporating buckling constraints into design and optimization are prevalent in the literature. The first is to treat a given structure as a collection of simplified members, usually beams or plates, and apply some closed-form analytical solution for the critical buckling load to each member, assuming some geometry and boundary conditions. The task then becomes to ensure that the critical load is not exceeded for each member. This is relatively straightforward for truss structures represented as assemblies of beams, and has been demonstrated by a number of works [63-65]. The application to shell structures as assemblies of plates is somewhat more involved in derivation, but has been implemented as well. Kennedy *et al.* [9] performed an aero-structural optimization and design of a NASA common research model (CRM) wing model to minimize fuel burn and structural weight. They imposed buckling constraints by defining critical loads under the assumption that each panel of the wing was flat (with no curvature) and simply-supported on all edges. A similar approach was used in the follow up work on establishing a benchmark for the undeflected CRM wing by Brooks *et al*. [10].

Breaking the structure up into a collection of simplified members as described only directly addresses the buckling of the individual members. If global buckling of the structure as a whole is to be considered, some method of accounting for this must be used on top of the local member constraints. Several authors have put forth methods for this as well [66-68].

Having buckling criteria defined as closed-form analytical functions of loading or strain has the advantage of low computational cost. But it has several notable drawbacks as well. First, it lacks accuracy for general structures with complex and arbitrary geometry. Analytical solutions for critical buckling loads only exist for a limited domain of geometry and boundary conditions, so usually significant approximations and simplifications must be made to represent a structure in this fashion. These simplifications can lead to considerable errors, and care must be taken to interpret the results conservatively. The implementation can also be cumbersome from the point of view of the user, since each applicable section of the structure needs physical parameters like dimensions and second moment of area for cross sections, which aren’t always conveniently available. It may also be difficult to accurately accommodate complex material behavior like that of anisotropic composite laminates.

The second major approach to applying buckling constraints in optimization is through eigenpair analysis of the structural system. The buckling deformation modes of a discretized structural system and their corresponding critical loads can be represented by the eigenpairs of the tangent stiffness matrix of the structure, constructed evaluated at a given state of loading and including the nonlinear terms of the governing equations. This concept is explained more thoroughly in Chapter 6. The problem becomes much like modal analysis for dynamic structures in vibration, except that it does not require consideration of the structure’s mass properties (unless that is a significant factor in the loading conditions).

Eigenpair analysis can be a very powerful tool, and unlike the simplified member approach it is generally applicable to any arbitrary finite element structure, yet it is also not without its difficulties. First and foremost, eigenpair analysis is rather costly for large systems, and if the eigenpairs factor into an optimization objective, they must be re-solved on every optimization cycle. Furthermore, in the context of gradient-based optimization, it is not just a question of finding the eigenpairs, but also posing them as an objective or constraint and finding their sensitivities with respect to design parameters. This in itself represents a considerable cost and investment, and the implementation with the adjoint method is discussed further in Chapter 4, Section 4.3.

These costs can be reduced by finding only a select few of the most critical eigenpairs, but if too few are considered, it can lead to abrupt discontinuities in the objective function, causing the optimizer to "bounce" between a few of the lowest buckling modes. This can slow convergence or even prevent convergence altogether, and the most appropriate number of eigenmodes to consider is not necessarily clear [69,62]. There can also be complications if eigenvalue multiplicity arises in the process [70].

Both of the main approaches described above leave something to be desired, and it would be of value to identify a way of approaching the application of buckling constraints in optimization that is generally applicable, affordable, robust and effective. The final study of this work investigates this issue and tests a novel solution that could potentially address the problem.

Chapter 2

AStrO: Adjoint-Based Structural Optimizer

Having established the importance of high-fidelity modeling and efficient optimization methods in advanced multidisciplinary physics applications, the fundamental motivation for the present work can be put into context. Extensive computational work has been done in recent years at the University of Wyoming College of Engineering and Applied Sciences, including but not limited to the area of computational fluid dynamics. Numerous studies have been performed on simulation and optimization for fixed wing, rotorcraft and wind energy applications [40,71-74]. These studies have been conducted largely by open-source tools developed in-house for a high degree of flexibility, specialization, and efficiency in tight coupling of multiple disciplines.

To obtain the most accurate results in CFD applications involving flexible structures, the coupled interaction between the dynamics of the fluid and the structure must be accounted for. It has been common practice in the field for analysis focused on aerodynamic considerations, such as lift and drag performance, to account for aeroelastic behavior using low-fidelity structural models, often constructed out of beam elements or the like. While even low-fidelity models can often give reasonably accurate solutions for overall displacement, this approach cannot account for the inevitable changes in structural properties due to aerodynamic shape changes in an optimization process. Furthermore, beam element models cannot represent high-fidelity stress distributions throughout an actual structure, so if an objective or constraint is to be defined in terms of stress or some other advanced structural consideration, a higher fidelity approach is required.

Many commercial codes, such as Abaqus [75], have advanced capability for high-fidelity structural modeling. These packages are well-documented and good for general use, but are not always optimal for specific advanced applications. They are also not open-source, so flexibility in modeling implementation is limited, and tight coupling with other packages is generally difficult or impossible. Many of the aforementioned tools for CFD analysis have built-in capability for obtaining sensitivities using the adjoint method, for efficient gradient-based optimization. To be a suitable companion for these tools, a structural modeling package would need to possess this capability as well.

Considering all these points, a need was identified to develop an open-source finite element structural modeling package, readily coupled with other codes and capable of high-fidelity simulation and obtaining objective sensitivities using the adjoint method. The package has come to be known as Adjoint-Based Structural Optimizer (AStrO), and its functionality and implementation is presented in the remainder of this chapter.

2.1 Functionality of AStrO

AStrO is first a package for finite element modeling of thermoelastic behavior of structures. It is ultimately intended for adjoint-based sensitivity analysis and coupling with solvers of other disciplines, but it also stands alone, running independently of other packages, and without any consideration of sensitivity analysis or optimization. It internally simulates both the temperature response due to heat conduction in thermally loaded structures, and the elastic displacement response due to thermal and mechanical applied loads, all in three dimensions.

AStrO runs either static or dynamic analysis, modeling either of the two mentioned disciplines alone or both coupled together. Dynamic time integration is implicit, using the Newmark Beta expansion with Hilber-Hughes-Taylor “alpha scheme” [76], discussed in further detail in Chapter 3. Modeling of geometrically nonlinear structural problems is supported, enabling analysis of structures undergoing large displacements and rotations, but there is presently no internal support for material nonlinearity, which may arise from phenomena like large strains, plastic deformation or material failure. Geometrically nonlinear problems may be solved with Newton-Raphson or modified Newton-Raphson iterations [77]. Two main types of linear solvers are built-in for the process of obtaining the finite element solutions and sensitivities: LDL^T factorization and conjugate gradient. If desired, solution of linear systems can be outsourced to other packages.

Model input files defined in the format of Abaqus commercial software can be processed directly by AStrO to define geometry, mesh, structural properties, loads, boundary conditions and constraints for an analysis. In this way the models can be conveniently constructed in the user interface of Abaqus, or readily converted from other common formats. The file processor does not, however, recognize the comprehensive set of all Abaqus commands, but only a fundamental subset for defining the essential characteristics of a structure. Table 2.1 details the specific keywords/commands of the Abaqus input interface that are recognized by AStrO. The specific element types supported by AStrO are listed in Table 2.2, by the identification codes of their approximate equivalents in Abaqus.

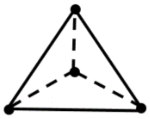
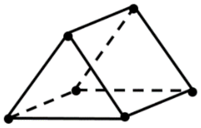
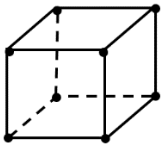
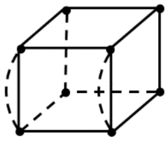
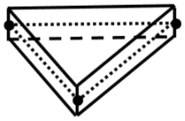
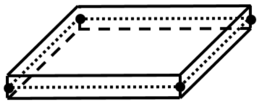
Table 2.1 Abaqus keywords supported by AStrO's input file processor.

Keyword	**Description**
`*Part`	Marks the beginning of a series of data inputs defining the characteristics of a single part in a structural assembly.
`*Node`	Marks the beginning of a list of nodes, where each entry defines the label (ID number), and x_1, x_2, x_3 coordinates of a single node in the finite element mesh.
`*Element`	Marks the beginning of a list of elements of a certain specified type, where each entry defines the label, and a list of all the nodes belonging to an element by node label.
`*Nset`	Defines a set of nodes within the structure that usually share a common characteristic or association. The set is given a name, followed by a list of elements that belong to it on subsequent lines.
`*Elset`	Defines a set of elements within the structure that usually share a common characteristic or association. The set is given a name, followed by a list of elements that belong to it on subsequent lines.
`*Material`	Marks the beginning of a series of data entries that define the properties of a certain material in the structure. Each material is identified with a user-defined name.
`*Elastic`	Defines the elastic material properties of the current material. AStrO recognizes only materials within the category of orthotropic, with elastic properties defined by three orthogonal elastic moduli, E_1, E_2, E_3, three orthogonal measures of Poisson's ratio, $\nu_{12}, \nu_{13}, \nu_{23}$, and three orthogonal shear moduli, G_{12}, G_{13}, G_{23}.
`*Density`	Defines the mass density of the current material.
`*Conductivity`	Defines the thermal conductivity of the current material, in three orthogonal directions.
`*Expansion`	Defines the coefficients of thermal expansion of the current material, corresponding to each component of strain.
`*Specific`	Defines the specific heat capacity of the current material.

Table 2.1 (continued)

*Orientation	Defines a local coordinate system for a section of a part. The coordinate system is defined by a vector that represents the 1-direction of the local coordinate system, and a point that lies in the 1-2 plane of the local coordinate system.
*Solid	Defines a section of a part that is composed of solid continuum elements, and is associated with a set of elements defined by *Elset, a material defined by *Material, and optionally a local coordinate system defined by *Orientation.
*Shell	Defines a section of a part that is composed of shell elements, and is associated with a set of elements defined by *Elset, a material defined by *Material, a value representing the section thickness, and optionally a local coordinate system defined by *Orientation and a value representing the section offset.
*Beam	Defines a section of a part that is composed of beam elements, and is associated with a set of elements defined by *Elset, and a list of section properties. The section properties are as follows: cross-sectional area A, second moments of area in the 2-3 plane I_2, I_{23}, I_3, polar moment of area J, three components of a unit vector defining the 2-direction n_1, n_2, n_3, elastic modulus E, Poisson's ratio ν, shear modulus G, and mass density ρ.
*Assembly	Marks the beginning of a set of commands defining the global assembly of a structure. These commands include *Instance *Nset and *Equation.
*Instance	Defines an instance of a specific part in a structural assembly, with an optional translation and rotation vector to define its global position. Each instance of a part is given a unique user-defined name.
*Cload	Defines a load in the form of a concentrated force on a set of nodes defined by *Nset.
*Boundary	Defines a boundary condition on the displacement or temperature of a set of nodes defined by *Nset.
*Equation	Defines a linear constraint or set of constraints placed on the relative displacement or temperature of sets of nodes defined by *Nset. The constraints are defined by set names and coefficients of an equation of the form $c_1 u_i^{set1} + c_2 u_j^{set2} + c_3 u_k^{set3} \ldots = 0$

Table 2.2 Abaqus element types with equivalent versions in AStrO.

Abaqus Identification Code	**Graphic Depiction**	**Description**
`C3D4`		Four-node tetrahedral solid continuum element.
`C3D6`		Six-node triangular prism solid continuum element.
`C3D8`		Eight-node hexahedral solid continuum element.
`C3D8I`		Eight-node hexahedral solid continuum element with incompatible modes.
`S3`		Three-node shell element.
`S4`		Four-node shell element.
`B31`		Two-node beam element.

In addition to performing thermoelastic analysis, AStrO is also capable of obtaining sensitivities of solution-dependent objectives with respect to design parameters of a structure using the adjoint method. To accomplish this, both the objective function and the specific design variables must be defined by the user. The objective function is typically hard-coded into a user-defined subroutine directly, in order to provide flexibility for the limitless possible objectives that could be defined. The design variable definition may be entered into a text input file according to a predetermined format and read in by a command from the main program, or alternatively hard-coded directly like the objective function into user-defined routines. More details about the design variable input format are given in Section 2.2.

AStrO offers a basic on-board optimizer, using a steepest-descent line search algorithm with backtracking [78]. This function is convenient for basic problems and quick troubleshooting, since it is self-contained and does not require linking with external packages. It does not necessarily perform efficiently for problems with complex constraints, however. Alternatively, more sophisticated optimizers can be implemented alongside AStrO from the main driver program.

AStrO is intended to be readily coupled with other packages for multi-disciplinary analysis. To that end, it contains a user-accessible module of data pertaining to the surface faces of the finite element structure. The primary use of this feature to date is for tight-coupling with NSU3D [71-73], a highly validated Reynolds-Averaged Navier Stokes (RANS) based CFD modeling package for aeroelastic analysis. NSU3D and AStrO can be linked together by a fluid-structure interface, which enables information to be passed between the two disciplines. To begin, the exterior surface faces and nodes of the structural finite element mesh and the interior surface nodes of the CFD grid are identified. An associative mapping is then constructed between the two surfaces, so that aerodynamic forces due to pressure and skin friction produced by the flow solver can be transferred as equivalent loads to the structural mesh. The resulting structural displacements due to those aerodynamic forces can be passed back to the CFD solver through the transpose of the same mapping, and the flow solution can be re-computed based on the updated displacement. This process can be repeated until the fully coupled solution converges to a stable state, as summarized below.

1. Compute the aerodynamic forces on the interface surface $\boldsymbol{F}_{CFD}$ from the CFD flow solution at the current surface deformation state $\boldsymbol{U}_{CFD}$.
2. Form the structural load vector $\boldsymbol{F}_{STR} = [T_{FSI}]\boldsymbol{F}_{CSD}$.
3. Compute the structural displacement response $\boldsymbol{U}_{STR}$ due to the aerodynamic loads.
4. Update the current displacement of the CFD surface $\boldsymbol{U}_{CFD} = [T_{FSI}]^T \boldsymbol{U}_{STR}$
5. Repeat steps 1-4 until solution reaches convergence.

This method of multidisciplinary modeling is essentially a disciplinary block Gauss-Seidel style solver, and is generally valid for applications with a high ratio of structural stiffness to dynamic pressure. For problems with, for example, very high-density fluid or unusually flexible structures, it can be slow to converge or fail to converge entirely. In those cases, a more robust approach would have to be employed. Nevertheless, the presented approach has been sufficient for all cases in the context of the present work.

In aeroelastic analysis, the structural finite element model and the CFD grid representing the fluid flow domain are usually defined and discretized separately, due to different resolution requirements and other modeling considerations. The fluid structure interface must therefore be able to tie together the boundaries of separate modeling domains that are mismatched in both resolution and surface topography, as illustrated in Fig. 2.1.

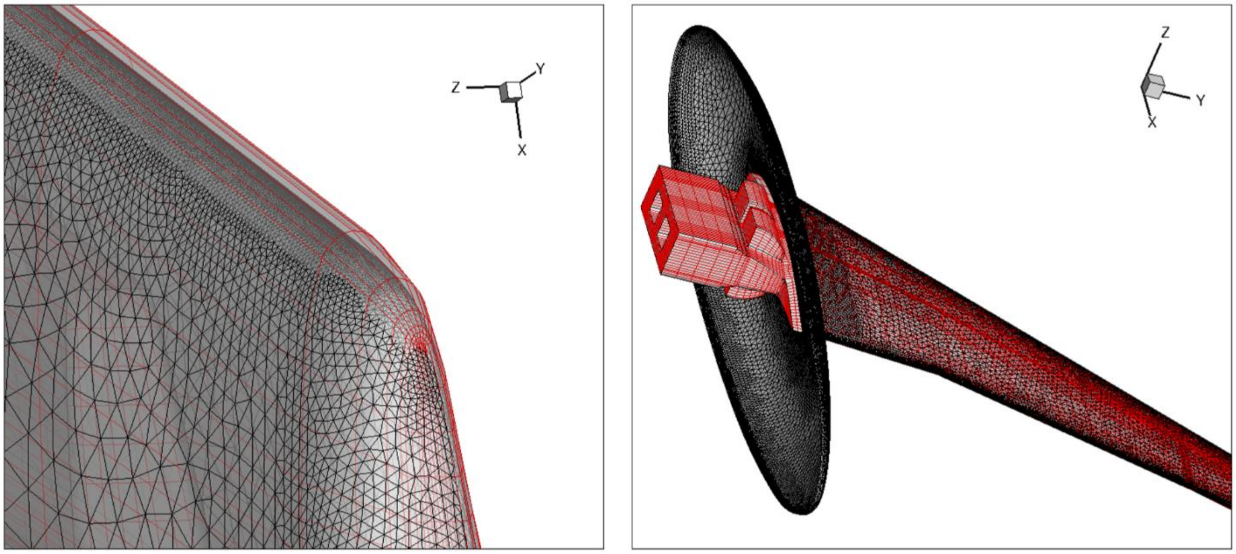


Figure 2.1 Mismatching surface discretization between structural (red) and CFD (black) meshes.

Typically, the resolution of the CFD grid is finer than the structural finite element mesh. The first step in the process of generating the fluid-structure mapping is to project each CFD grid point on the interface surface by its perpendicular distance to the nearest surface face of the structural mesh. An appropriate *weight value* is computed associating each projected CFD point to all of the structural nodes encompassing that face, based on the CFD point's location on the face. These weight values are then used to construct the mapping interface matrix $[T_{FSI}]$ as previously described.

2.2 Implementation

AStrO is coded entirely in Fortran, which was chosen primarily to be easily integrated with existing Fortran-based CFD applications. The main package consists of a collection of modules, each containing data and/or functions pertaining to a certain aspect of AStrO's function. The modules are intended to be pre-compiled, and referenced from a separate main program driver provided along with any additional packages to be used in tandem for optimization or multidisciplinary analysis by the user. Currently AStrO has no parallel implementation, though that is a long-term goal. The linear solvers and the element-by-element construction of the global matrix equations and sensitivity vectors are two main aspects with potential for parallel implementation. The best approach is a topic of future consideration, and outside the scope of the present work.

The basic procedure for setting up an analysis in AStrO can be summarized in the following steps:

1. Write the main program driver, based on the type of analysis desired.
2. Fill in user subroutines defining the objective function and constraints and construct the design variable input file within the working directory as necessary.
3. Compile the modules of the package in order based on their dependence (see Table 2.5).
4. Compile and run main program driver.

If only static or dynamic analysis is required for a simulation, with no need for sensitivity analysis or optimization, then step 2 is not required, and the user-defined subroutines can be left blank or in their default form.

The format of the design variable input file, if required, will now be described. The design variable input file contains a series of entries, each defining a design variable or a group of design variables. The heading for each entry takes the following format:

```
*DesignVariable, <D0>, <LB>, <UB>, <ACT>, <VTYPE>, <LEN>
```

with data entries defined in Table 2.3.

Table 2.3 Data entries for a heading line in the design variable input format of AStrO.

Data Entry	**Type**	**Description**
D0	Real	The initial value for the design variable(s).
LB	Real	The lower bound of the design variable's allowed range.
UB	Real	The upper bound of the design variable's allowed range.
ACT	Integer	Whether the variable is to be considered active in the present simulation (1 if yes, 0 if no).
VTYPE	Integer	A code specifying the exact type of design variable.
LEN	Integer	The length of the list of nodes or elements that the variable applies to.

As indicated by the final data entry of the heading, each variable definition is to be followed by a list of the labels of all elements or nodes to which the design variable(s) are meant to apply. The second to last data entry of the heading, VTYPE, is a six-digit integer defining the type of design variable being declared, which breaks down as described in the following paragraphs.

The first two digits of VTYPE represent the general category of the design variable. The numeric codes for the categories are given in Table 2.4.

Table 2.4 Category codes for design variable types supported by AStrO.

Category Code	Description
01	Elastic properties
02	Mass density
03	Thermal conductivity
04	Coefficients of thermal expansion
05	Specific heat capacity
06	Local material orientation/coordinate system
07	Section properties for shell or beam sections
08	Nodal coordinates in the undeformed configuration
09	Applied mechanical load
10	Applied thermal load

The third and fourth digit of VTYPE represent the component of the category property that the design variable is to control. Most of the design properties described in Table 2.4 have multiple components, and they can each be made independent design variables with this specification. For a more detailed description of each design variable category and its components, see Chapter 4, Section 4.2.2.

The fifth digit of VTYPE is the grouping option. It specifies whether the current entry is to define one single design variable that applies to all the elements or nodes listed, or a group of design variables, each of which applying to one of the elements or nodes. A value of 0 indicates a single design variable, a value of 1 indicates a group of design variables. The sixth and final digit of VTYPE indicates whether a list of coefficients is to be specified to indicate the "weight" of the design variable's effect on each individual node or element. To clarify, the input file processor declares any given property P of a node or element as a linear combination of the applicable design variables, as indicated by Eq. (2.1).

$$P(D_i) = P_0 + \sum_{i=1}^{n} c_i D_i \tag{2.1}$$

With the sixth digit set to 1, the coefficients c_i can be provided in a list immediately following the list of elements or nodes in the input file. If it is set to 0, all the coefficients are 1.0 by default. The coefficient option is often used when one design variable is being applied to a group of nodes, so that the coefficients represents some interpolation function that is being used as a mode of shape deformation.

Several tables are provided in the remaining pages of this chapter detailing the internal structure and features of AStrO's source code package. Table 2.5 lists the name of each module, along with a brief description and other modules of the package that it depends on. Table 2.6 highlights the main commands of the global function interface, which are meant to be called from the main program driver by the user. Tables 2.7 and 2.8 provide highlighted data members from the global data and surface data modules, which can be accessed for results, data, or to set modeling options and parameters.

Table 2.5 Descriptions of individual modules within AStrO.

Module Name	**Description**	**Uses Modules…**
AStrO_globalFunctions	Contains a collection of high-level functions intended to be called by the user from the main program.	(all other modules)
AStrO_userObjective	Contains the user-defined subroutines that define the objective function and its derivatives in terms of design variables and solution variables.	AStrO_constantVals, AStrO_globalData
AStrO_adjoint	Contains a collection of functions that primarily calculate the exact design-variable sensitivities of various components of the governing equations, which ultimately factor into the total objective sensitivities.	AStrO_constantVals, AStrO_globalData, AStrO_elementEqns, AStrO_userAdjointDesignVars
AStrO_userAdjointDesignVars	Contains functions that define the design space of a structure, by specifying each fundamental design property and its sensitivities with respect to all design variables. These functions can be set by the design variable input file or overwritten directly, and are accessed by the internal routines to construct the governing equations and sensitivity vectors.	AStrO_constantVals, AStrO_globalData
AStrO_processAbaqusInput	Contains upper-level routines that scan the Abaqus-format input file, and read the data into the structures in the AStrO_globalData module.	AStrO_constantVals, AStrO_globalData, AStrO_abaqusRead, AStrO_elementEqns
AStrO_abaqusRead	Contains a collection of functions that scan and process individual keyword commands in the Abaqus input file.	AStrO_constantVals, AStrO_elementEqns
AStrO_elementEqns	Contains a collection of functions that primarily construct and arrange the global governing equation matrices.	AStrO_constantVals, AStrO_globalData, AStrO_solvers AStrO_userAdjointDesignVars
AStrO_solvers	Contains a collection of functions dedicated to the solution of linear systems of equations. Various methods are implemented for both dense and sparse storage formats.	AStrO_constantVals
AStrO_surfaceData	Contains a collection of data pertaining to the surface faces of the finite element structure, for use in interfacing with other disciplinary solvers.	(none)
AStrO_globalData	Contains a collection of global data defining the geometric and material properties of the structure, as well as structures for generating the finite element solution.	(none)
AStrO_constantVals	Contains a set of commonly used real values stored as global variables.	(none)

Table 2.6 Descriptions of key interface commands in AStrO.

Global Command	Description
AStrO_processInput()	Reads model definition from the Abaqus-format input file with the name specified by the variable inFileName in the AStrO_globalData module.
AStrO_getSurfaceFaces()	Assembles the data defining the collection of element faces that make up the surface of a structure, stored in the AStrO_surfaceData module.
AStrO_constructStiffnessMatrix()	Builds the global finite element equation stiffness matrix for the equations of elasticity, based on specified modeling parameters.
AStrO_constructThermalMatrix()	Builds the global finite element equation stiffness matrix for the equations of thermal heat conduction, based on specified modeling parameters.
AStrO_constructLoadFromInputFile()	Constructs the global vectors representing the applied mechanical loads, and applied thermal loads on the nodes of the structure, based on the definitions given in the Abaqus model input file.
AStrO_constructGravityLoad()	Constructs the global applied mechanical load vector to simulate the force of gravity on a structure based on material mass density distribution, and lumped masses.
AStrO_setNodeTemp()	Sets a new boundary condition on the temperature of a node to a certain value, which may not be specified in the Abaqus input file
AStrO_setSurfaceFlux()	Applies a heat flux to a certain point of the surface of a structure, in the form of appropriate nodal thermal loads.
AStrO_getSurfaceTemp()	Calculates the temperature at some point on the surface of a structure, based on the nodal temperature solution.
AStrO_factorStiffnessMartrix()	Performs any necessary factorization on the global equation matrix for the equations of elasticity, based on the solver method.
AStrO_factorThermalMatrix()	Performs any necessary factorization on the global equation matrix for the equations of thermal heat conduction, based on the solver method.
AStrO_solveForDisplacement()	Solves for the displacement of the structure based on the equations of elasticity, either for the state of equilibrium for static analysis, or at the next time step for dynamic analysis. Procedure determined by modeling options and result stored in nodeDisp, of the AStrO_globalData module.
AStrO_solveForTemperature()	Solves for the temperature of the structure based on the equations of elasticity, for the state of equilibrium or at the next time step for dynamic analysis. Procedure determined by modeling options and result stored in nodeTemp, of the AStrO_globalData module.
AStrO_updateVelAcc()	Updates the time derivatives of the nodal degrees of freedom, nodeVel, nodeAcc, and nodeTdot, for the current time step based on the results just obtained for temperature and displacement in dynamic analysis.
AStrO_calculatedLdDAdjoint()	Calculates the sensitivities of the objective function at the current state using the adjoint method.
AStrO_calculatedLdDForward()	Calculates the sensitivities of the objective function at the current state using the tangent method.
AStrO_optimizeObj()	Runs a steepest-descent based optimization algorithm to minimize the objective function in the design space, using adjoint-based sensitivities.
AStrO_writeDisplacement()	Writes the nodal degrees of freedom and their time-derivatives at the current time step to disk in binary form, for use in sensitivity analysis for dynamic problems.
AStrO_readDisplacement()	Reads the nodal degrees of freedom and their time-derivatives at the current time step from disk in binary form, for use in sensitivity analysis for dynamic problems.
AStrO_writeResults()	Writes the results for nodal degrees of freedom, and stresses and strains at integration points to file in text format after analysis.
AStrO_deallocateAll()	Deallocates all applicable data members of the AStrO_globalData and AStrO_surfaceData modules, after the full analysis is complete.

Table 2.7 Descriptions of highlighted members of the global data module in AStrO.

Variable Name	**Description**
numNodes	Integer specifying the number of nodes in the finite element model's mesh.
nodeList	3 X (numNodes) array of real numbers, specifying the spatial coordinates of each node of the finite element model's mesh. nodeList(i,j) is the i-direction coordinate of node number j.
nodeOrder	Integer array of length (numNodes) specifying the new order of the nodes in the model's finite element mesh, after they have been re-ordered to minimize matrix bandwidth. nodeOrder(i) is the original rank of the node that is currently at rank i.
currentRank	Integer array of length (numNodes), which is the inverse list of nodeOrder. Therefore currentRank(i) is the current rank of the node that was originally at rank i.
nLastDof	Integer array of length (numNodes) specifying the rank in nodeDisp, nodeVel, nodeAcc and Rapp corresponding to the last degree of freedom of each node. nLastDof(i) is the rank of the last degree of freedom of node i, in the current nodal arrangement.
numEls	Integer specifying the total number of elements in the finite element mesh.
elementList	8 X (numEls) array of integers specifying the nodal connectivity of each element in the mesh. elementList(i,j) is the original rank (label) of the i^{th} node of the j^{th} element. Together, elementList and nodeList define the finite element mesh.
eTypes	Integer array of length (numEls) specifying the type code of each individual element. eTypes(i) is the type code of element number i.
Kdim	Integer specifying the dimension of the global equation matrix for the structure, which is also the total number of degrees of freedom in the structural system.
KNonZeroCt	Integer specifying the total number of non-zero terms in the global equation matrix.
Kglobal	Real array of length KNonZeroCt storing the values of the non-zero terms in the global equation matrix. Terms are stored row by row, with companion tags indicating the column of each term stored in Kcols.
Kcols	Array of integers of length KNonZeroCt labeling each term in Kglobal with its column in the global equation matrix. Kcols(i) is the column of the term stored in Kglobal(i).
KRange	Integer array of length Kdim + 1 indicating the range of each row of the global matrix in Kglobal. KRange(i) is the last term of Kglobal belonging to row i.
nodeDisp	Real array of length Kdim storing the current displacement degrees of freedom for every node in the finite element mesh. The displacements are arranged node by node, $[u_{11},u_{21},u_{31},u_{12},u_{22},u_{32},u_{13}\ldots]$ where u_{ij} is the i-direction displacement at node j according to the current node order.
nodeVel	Real array of length Kdim storing the current velocity degrees of freedom for every node in the finite element mesh. The velocities are arranged in the same manner as nodeDisp.
nodeAcc	Real array of length Kdim storing the current acceleration degrees of freedom for every node in the finite element mesh. The accelerations are arranged in the same manner as nodeDisp.
nodeTemp	Real array of length numNodes storing the current temperature degrees of freedom, according to the current node order.
nodeTdot	Real array of length numNodes storing the current 1st time derivative of the temperature degrees of freedom, according to the current node order.
Rapp	Real array of length Kdim storing the applied load vector for every node in the finite element mesh. The loads are arranged in the same manner as nodeDisp.
RtHG	Real array of length numNodes storing the nodal thermal loads, representing heat flux and internal heat generation, according to the current node order.
solverMethod	String specifying which solver method to use to obtain solutions for displacements and sensitivities. Possible values are 'ldlfact', 'conjgrad', 'pcconjgrad1', 'pcconjgrad2', 'pcconjgrad3'.
dynOn	Integer specifying whether dynamic (time-dependent) analysis is to be performed. dynOn=0 indicates that the problem is static, and no dynamic analysis is needed. dynOn=1 indicates that the problem is dynamic, and time integration is performed using the Hilber-Hughes-Taylor implicit method [76].

Table 2.7 (continued)

thermOn	Integer specifying whether thermal effects are to be considered in the analysis. A value of 1 if thermal effects should be considered, a value of 0 if not.
nLGeom	Integer specifying whether nonlinear geometry is to be considered in the analysis. A value of 1 if nonlinear geometry should be considered, a value of 0 if not.
delT	Real number specifying the time step to be used in the dynamic time integration.
numTSteps	Integer specifying the total number of time steps to be taken in dynamic analysis
NMal	Real number specifying the 'alpha' parameter in the Hilber-Hughes-Taylor time integration scheme.
NMbet	Real number specifying the 'beta' parameter in the Hilber-Hughes-Taylor time integration scheme.
NMgam	Real number specifying the 'gamma' parameter in the Hilber-Hughes-Taylor time integration scheme.
dampCK	Real number specifying the coefficient of the stiffness matrix in defining Rayleigh damping on the structure. ([C] = dampCK[K] + dampCM[M])
dampCM	Real number specifying the coefficient of the mass matrix in defining Rayleigh damping on the structure. ([C] = dampCK[K] + dampCM[M])
Dsize	Integer defined by the user, specifying the number of design variables in the optimization design space.
Dvec	Real array of length (Dsize) specifying the current values of the design variables in the optimization design space. These values are initialized by the user.
objFunc	Real number specifying the current value of the user-defined objective function.
dLdD	Real array of length (Dsize) specifying the gradient of the user-defined objective function in the design space. dLdD(i) is the derivative of the objective function with respect to Dvec(i).
sensOpt	Integer specifying the method to be used for obtaining objective function sensitivities. sensOpt=1 indicates the adjoint method is to be used. sensOpt=2 indicates the forward/tangent method is to be used.
meshDef	Integer specifying whether the adaptive mesh deformation option should be used in optimization. A value of 1 if mesh deformation should be used, a value of 0 if not.

Table 2.8 Descriptions of highlighted members of the surface module in AStrO.

Variable Name	**Description**
numComp	Integer specifying the number of components in the structural assembly (Abaqus documentation would refer to these as part instances).
numBface3	Integer specifying the number of triangular element faces on the surface of the structure.
numBface4	Integer specifying the number of quadrilateral element faces on the surface of the structure.
nodeComp	Integer array of length (numNodes) specifying which component each node belongs to
Bface3	Integer array of length 3 X (numBface3) specifying the nodal connectivity of each triangular face. Connected nodes are given face by face, as [n_{11}, n_{21}, n_{31}, n_{12}, n_{22}, n_{32}, n_{13},…] where n_{ij} is the i^{th} node of the j^{th} face.
Bface4	Integer array of length 4 X (numBface3) specifying the nodal connectivity of each quadrilateral face. Connected nodes are ordered in the same manner as in Bface3, following the right hand rule with a unit normal pointing in the outward direction from the structure surface.
Bface3Comp	Integer array of length numBface3, specifying which component each triangular face belongs to. Bface3Comp(i) is the component of face number i
Bface4Comp	Integer array of length numBface4, specifying which component each quadrilateral face belongs to. Bface4Comp(i) is the component of face number i.

Chapter 3

Finite Element Formulation

AStrO simulates the elastic and thermal behavior of deformable structures using the finite element method. As is typical in structural finite element analysis, the discretized equations of motion for the elastic response are derived from the principle of virtual work. Temperature distribution due to heat conduction through a structure is governed by the Poisson equation, which can be expressed in variational form and discretized in a manner similar to the equations of elasticity. Both formulations are equivalent to applying the Galerkin method to their corresponding partial differential equations and integrating the gradient term by parts.

Analysis can be performed for static and dynamic cases with linear and nonlinear geometry for three-dimensional solid continuum elements, shell elements and beam elements. In this chapter the details of the formulation of discretized governing equations are provided for all cases. In the developments that follow, scalar quantities are denoted with normal script text, vectors are denoted in boldface font, and matrices are denoted in square brackets [ ].

3.1 Overall Formulation

The fundamental governing equations of motion for the elastic and thermal response, and the common overall approach to discretizing the equations for all elements and cases is given in this section. The formulation for the equations of elastic response are given first in Section 3.1.1, followed by that of the thermal response in Section 3.1.2, and for coupled thermoelastic response in 3.1.3. Section 3.1.4 gives a discussion on numerical integration methods necessary to construct the equations.

3.1.1 Governing Equations of Elasticity

Static and dynamic response of systems of rigid or flexible bodies is commonly analyzed using Newton's second law or conservation of linear momentum as the governing equations. But in some contexts, including finite element analysis, it can be convenient to express governing equations in terms of work or energy formulations. In AStrO, the equations of elastic response for all types of elements begin with the principle of virtual work.

To understand the principle of virtual work, one must first understand the concept of a virtual displacement. Imagine a dynamic system of bodies in motion and subject to applied forces. Now imagine that system was frozen at some instant in time, and an arbitrary infinitesimal change, or *variation* was made to the displacement at all points throughout the structure. That imaginary change in displacement is a virtual displacement. According to Newton's second law, the combined forces and acceleration must balance, or sum to zero, at every point in a structural system at every moment in time. Consequently, the total work, or *virtual work* performed by all forces and acceleration throughout the system due to the virtual displacement must sum to zero. This is the principle of virtual work.

For a deformable elastic body subject to applied body forces and surface tractions, as well as damping force proportional to velocity, the principle of virtual work can be expressed mathematically as

$$\int_{\Omega}(\boldsymbol{\sigma}\cdot\delta\boldsymbol{\epsilon})d\Omega+\int_{\Omega}\xi(\dot{\boldsymbol{u}}\cdot\delta\boldsymbol{u})d\Omega+\int_{\Omega}\rho(\ddot{\boldsymbol{u}}\cdot\delta\boldsymbol{u})d\Omega-\int_{\Omega}(\boldsymbol{f}\cdot\delta\boldsymbol{u})d\Omega-\int_{S}(\boldsymbol{t}\cdot\delta\boldsymbol{u})dS=0 \qquad (3.1)$$

where $\delta\boldsymbol{u}$ is a virtual displacement field, which is a function of space throughout the body and $\delta\boldsymbol{\epsilon}$ is the variation of strain corresponding to that virtual displacement. Ω represents the entire volume domain occupied by the elastic body and S is the area domain of the surface of the body. Each of the terms in Eq. (3.1) represents the virtual work done by one contributor of force in the system due to the virtual displacement. The contributors are (from left to right): stress, damping, mass, body force, and surface traction. This equation must hold for any arbitrary infinitesimal virtual displacement, to be consistent with Newton's second law.

The principle of virtual work is in fact an application of the *fundamental lemma of variational calculus* [79], which can be used to show equivalence with Newton's second law. According to the fundamental lemma, if a functional given by

$$\Phi = \int_{\Omega} F\big(\boldsymbol{u}(\boldsymbol{x})\big)v(\boldsymbol{x})d\Omega \tag{3.2}$$

evaluates to zero for any arbitrary function v then the function F must be zero at every point in the domain Ω. In other words, the two conditions given in Eq. (3.3) are equivalent.

$$\Phi = \int_{\Omega} F\big(\boldsymbol{u}(\boldsymbol{x})\big)v(\boldsymbol{x})d\Omega = 0 \, for\ any\ v(\boldsymbol{x}) \ \Leftrightarrow\ F\big(\boldsymbol{u}(\boldsymbol{x})\big) = 0 \, for\ \boldsymbol{x}\ in\ \Omega \tag{3.3}$$

Often the fundamental lemma is used in the process of deriving governing equations for the extremization of a functional, and in fact the equations of elasticity for a conservative system can be derived from the variational minimization of the total potential energy function. For such cases the function v generally represents a variation of the primary variable(s), or the virtual displacement $\delta\boldsymbol{u}$ in the present context. However, the process can be taken in the other direction as well, that is to begin with a governing equation and put it back in variational form using the fundamental lemma.

Take the equations for Newton's second law applied to a point in a continuous elastic body:

$$\begin{aligned}
\frac{\partial\sigma_{11}}{\partial x_1} + \frac{\partial\sigma_{12}}{\partial x_2} + \frac{\partial\sigma_{13}}{\partial x_3} - \xi\dot{u}_1 - \rho\ddot{u}_1 + f_1 = 0 \\
\frac{\partial\sigma_{21}}{\partial x_1} + \frac{\partial\sigma_{22}}{\partial x_2} + \frac{\partial\sigma_{23}}{\partial x_3} - \xi\dot{u}_2 - \rho\ddot{u}_2 + f_2 = 0 \\
\frac{\partial\sigma_{31}}{\partial x_1} + \frac{\partial\sigma_{32}}{\partial x_2} + \frac{\partial\sigma_{33}}{\partial x_3} - \xi\dot{u}_3 - \rho\ddot{u}_3 + f_3 = 0
\end{aligned} \tag{3.4}$$

Note that Eq. (3.4) includes velocity-dependent terms representing a damping effect, and is not limited in application to conservative systems. The fundamental lemma applies to all three equations individually, for any three arbitrary weight functions v_1, v_2 and v_3. Without loss of generality, the state of Newton's second law can be expressed in variational form in a single equation:

$$\begin{gathered}\int_{\Omega}\left(\frac{\partial\sigma_{11}}{\partial x_1}+\frac{\partial\sigma_{12}}{\partial x_2}+\frac{\partial\sigma_{13}}{\partial x_3}-\xi\dot{u}_1-\rho\ddot{u}_1+f_1\right)v_1 d\Omega+\\ \int_{\Omega}\left(\frac{\partial\sigma_{21}}{\partial x_1}+\frac{\partial\sigma_{22}}{\partial x_2}+\frac{\partial\sigma_{23}}{\partial x_3}-\xi\dot{u}_2-\rho\ddot{u}_2+f_2\right)v_2 d\Omega+\\ \int_{\Omega}\left(\frac{\partial\sigma_{31}}{\partial x_1}+\frac{\partial\sigma_{32}}{\partial x_2}+\frac{\partial\sigma_{33}}{\partial x_3}-\xi\dot{u}_3-\rho\ddot{u}_3+f_3\right)v_3 d\Omega=0\end{gathered}\tag{3.5}$$

If, as previously discussed, v_1, v_2, and v_3 are taken to represent a field of virtual displacements δu_1, δu_2 and δu_3 then

$$\begin{gathered}\int_{\Omega}\left(\frac{\partial\sigma_{11}}{\partial x_1}+\frac{\partial\sigma_{12}}{\partial x_2}+\frac{\partial\sigma_{13}}{\partial x_3}-\xi\dot{u}_1-\rho\ddot{u}_1+f_1\right)\delta u_1 d\Omega+\\ \int_{\Omega}\left(\frac{\partial\sigma_{21}}{\partial x_1}+\frac{\partial\sigma_{22}}{\partial x_2}+\frac{\partial\sigma_{23}}{\partial x_3}-\xi\dot{u}_2-\rho\ddot{u}_2+f_2\right)\delta u_2 d\Omega+\\ \int_{\Omega}\left(\frac{\partial\sigma_{31}}{\partial x_1}+\frac{\partial\sigma_{32}}{\partial x_2}+\frac{\partial\sigma_{33}}{\partial x_3}-\xi\dot{u}_3-\rho\ddot{u}_3+f_3\right)\delta u_3 d\Omega=0\end{gathered}\tag{3.6}$$

The units of all terms in Eq. (3.6) are force times displacement, or work, indicating the physical significance that the total work performed by all contributors of force due to a virtual displacement is zero.

Now consider the stress-gradient terms in Eq. (3.6). These can each be integrated by parts, with respect to the variable of differentiation. For a given term, corresponding to σ_{ij},

$$\int_{\Omega}\frac{\partial\sigma_{ij}}{\partial x_j}\delta u_i d\Omega=-\int_{\Omega}\sigma_{ij}\delta\left(\frac{\partial u_i}{\partial x_j}\right)d\Omega+\int_{S}\sigma_{ij}n_j\delta u_i dS\tag{3.7}$$

The integrand of the first term on the right-hand-side of Eq. (3.7) summed over all components of stress can be written

$$\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}\delta\left(\frac{\partial u_i}{\partial x_j}\right) = \sigma_{11}\delta\left(\frac{\partial u_1}{\partial x_1}\right) + \sigma_{12}\delta\left(\frac{\partial u_1}{\partial x_2}\right) + \sigma_{13}\delta\left(\frac{\partial u_1}{\partial x_3}\right)$$
$$+ \sigma_{21}\delta\left(\frac{\partial u_2}{\partial x_1}\right) + \sigma_{22}\delta\left(\frac{\partial u_2}{\partial x_2}\right) + \sigma_{23}\delta\left(\frac{\partial u_2}{\partial x_3}\right) \tag{3.8}$$
$$+ \sigma_{31}\delta\left(\frac{\partial u_3}{\partial x_1}\right) + \sigma_{32}\delta\left(\frac{\partial u_3}{\partial x_2}\right) + \sigma_{33}\delta\left(\frac{\partial u_3}{\partial x_3}\right)$$

Because stress is symmetric when expressed as a second order tensor so that $\sigma_{ij} = \sigma_{ji}$, Eq. (3.8) can be rearranged as

$$\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}\delta\left(\frac{\partial u_i}{\partial x_j}\right) = \sigma_{11}\delta\left(\frac{\partial u_1}{\partial x_1}\right) + \sigma_{22}\delta\left(\frac{\partial u_2}{\partial x_2}\right) + \sigma_{33}\delta\left(\frac{\partial u_3}{\partial x_3}\right)$$
$$+\sigma_{12}\left(\delta\left(\frac{\partial u_1}{\partial x_2}\right) + \delta\left(\frac{\partial u_2}{\partial x_1}\right)\right) + \sigma_{13}\left(\delta\left(\frac{\partial u_1}{\partial x_3}\right) + \delta\left(\frac{\partial u_3}{\partial x_1}\right)\right) + \sigma_{23}\left(\delta\left(\frac{\partial u_2}{\partial x_3}\right) + \delta\left(\frac{\partial u_3}{\partial x_2}\right)\right) \tag{3.9}$$

Keep in mind that the virtual displacement field in Eqs. (3.6) and (3.7) is presumed to be infinitesimal, and therefore the corresponding variation in strain is also infinitesimal. Under this condition the following definition for the variation of normal and shear strains can be used:

$$\delta\epsilon_{ij} = \delta\left(\frac{\partial u_i}{\partial x_j}\right), \quad i = j$$
$$\delta\gamma_{ij} = \delta\left(\frac{\partial u_i}{\partial x_j}\right) + \delta\left(\frac{\partial u_j}{\partial x_i}\right), \quad i \neq j \tag{3.10}$$

This allows Eq. (3.9) to be written in terms of variations of strain:

$$\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}\delta\left(\frac{\partial u_i}{\partial x_j}\right) = \sigma_{11}\delta\epsilon_{11} + \sigma_{22}\delta\epsilon_{22} + \sigma_{33}\delta\epsilon_{33} + \sigma_{12}\delta\gamma_{12} + \sigma_{13}\delta\gamma_{13} + \sigma_{23}\delta\gamma_{23} \tag{3.11}$$

Given the result in Eq. (3.11) it becomes convenient to express stress and strain as one-dimensional vectors:

$$\boldsymbol{\sigma} = [\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}]^T$$
$$\boldsymbol{\epsilon} = [\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \gamma_{12}, \gamma_{13}, \gamma_{23}]^T \tag{3.12}$$

With this notation the following shorthand can be implemented for the virtual work due to a variation in strain:

$$\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}\delta\left(\frac{\partial u_i}{\partial x_j}\right) = \boldsymbol{\sigma}\cdot\delta\boldsymbol{\epsilon} \tag{3.13}$$

Returning now to Eq. (3.7), integrand of the second term on the right-hand side summed over all components of stress is equal to the dot product of the surface traction $\boldsymbol{t}$ and the virtual displacement:

$$\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}n_j\delta u_i = ([\sigma]\boldsymbol{n})\cdot\delta\boldsymbol{u} = \boldsymbol{t}\cdot\delta\boldsymbol{u} \tag{3.14}$$

Using Eqs. (3.13), (3.14) and (3.7), the integral sum of the stress gradient terms in the variational governing equations can be expressed

$$\int_{\Omega}\left(\sum_{i=1}^{3}\sum_{j=1}^{3}\frac{\partial\sigma_{ij}}{\partial x_j}\delta u_i\right)d\Omega = -\int_{\Omega}(\boldsymbol{\sigma}\cdot\delta\boldsymbol{u})d\Omega + \int_{S}(\boldsymbol{t}\cdot\delta\boldsymbol{u})dS \tag{3.15}$$

which can then be placed in Eq. (3.6), with all remaining terms expressed in vector form:

$$-\int_{\Omega}(\boldsymbol{\sigma}\cdot\delta\boldsymbol{\epsilon})d\Omega - \int_{\Omega}\xi(\dot{\boldsymbol{u}}\cdot\delta\boldsymbol{u})d\Omega - \int_{\Omega}\rho(\ddot{\boldsymbol{u}}\cdot\delta\boldsymbol{u})d\Omega + \int_{\Omega}(\boldsymbol{f}\cdot\delta\boldsymbol{u})d\Omega + \int_{S}(\boldsymbol{t}\cdot\delta\boldsymbol{u})dS = 0 \tag{3.16}$$

For convention, the sign can be reversed in Eq. (3.16) to yield the original statement of the principle of virtual work, given in Eq. (3.1). Using the principle of virtual work as a general governing equation of motion for dynamic elastic bodies is not only a versatile approach for applications requiring advanced considerations like nonlinear geometry and material behavior, but also provides a convenient means of discretizing equations for numerical solution methods, as will now be demonstrated.

In the finite element method, the fundamental variable(s) are assumed to take a known form with a discrete set of solution parameters. Typically, the solution is assumed to be a linear combination of pre-determined spatial basis functions, or interpolation functions, with the solution parameters representing the value of the variables at certain points or nodes. This approach is known as Lagrange interpolation. If the fundamental variable is displacement, as with the analysis of dynamic elastic bodies, then the assumed form of the displacement at any instant in time can be expressed

$$\boldsymbol{u} = [N](\boldsymbol{x})\boldsymbol{U}(t) \tag{3.17}$$

where $[N]$ is a matrix of basis functions and $\boldsymbol{U}$ is a vector of nodal solution parameters, or degrees of freedom. Since the basis functions are known, finding the displacement solution is now a matter of determining the nodal degrees of freedom. When it comes to implementing the principle of virtual work, the variations of displacement and strain must be expressed in terms of the discrete solution. A variation of the displacement $\boldsymbol{u}$ now corresponds to a variation in the nodal degrees of freedom $\boldsymbol{U}$, and can be represented using chain-rule differentiation:

$$\delta\boldsymbol{u} = \left[\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{U}}\right]\delta\boldsymbol{U} = [N]\delta\boldsymbol{U} \tag{3.18}$$

Similarly for a variation of strain,

$$\delta\boldsymbol{\epsilon} = \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]\delta\boldsymbol{U} \tag{3.19}$$

For now, the variation of strain will be left in the form in Eq. (3.19), as the form of strain as a function of the solution degrees of freedom is a topic of later sections.

Eqs. (3.18) and (3.19) can be substituted into the principle of virtual work in Eq. (3.1) as follows:

$$\int_{\Omega}\left(\boldsymbol{\sigma}\cdot\left(\left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]\delta\boldsymbol{U}\right)\right)d\Omega+\int_{\Omega}\xi\big(\dot{\boldsymbol{u}}\cdot([N]\delta\boldsymbol{U})\big)d\Omega+\int_{\Omega}\rho\big(\ddot{\boldsymbol{u}}\cdot([N]\delta\boldsymbol{U})\big)d\Omega \\ -\int_{\Omega}\big(\boldsymbol{f}\cdot([N]\delta\boldsymbol{U})\big)d\Omega-\int_{S}\big(\boldsymbol{t}\cdot([N]\delta\boldsymbol{U})\big)dS=0 \tag{3.20}$$

The variation of the nodal solution vector is not a function of space, and it appears in every term of the governing equation. This means that it can be pulled out of both the integrals and the summation, allowing Eq. (3.20) to be re-written

$$\left(\int_{\Omega}\left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^{T}\boldsymbol{\sigma}d\Omega+\int_{\Omega}\xi[N]^{T}\dot{\boldsymbol{u}}d\Omega+\int_{\Omega}\rho[N]^{T}\ddot{\boldsymbol{u}}d\Omega-\int_{\Omega}[N]^{T}\boldsymbol{f}d\Omega-\int_{S}[N]^{T}\boldsymbol{t}dS\right)\cdot\delta\boldsymbol{U}=0 \tag{3.21}$$

The argument behind the fundamental lemma of variational calculus applies in discrete form as well. Eq. (3.21) must hold for any arbitrary variation of the nodal solution parameters $\delta\boldsymbol{U}$, and it therefore follows that the vector in parentheses must be zero,

$$\int_{\Omega}\left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^{T}\boldsymbol{\sigma}d\Omega+\int_{\Omega}\xi[N]^{T}\dot{\boldsymbol{u}}d\Omega+\int_{\Omega}\rho[N]^{T}\ddot{\boldsymbol{u}}d\Omega-\int_{\Omega}[N]^{T}\boldsymbol{f}d\Omega-\int_{S}[N]^{T}\boldsymbol{t}dS=0 \tag{3.22}$$

Eq. (3.22) represents the discretized equations of motion for dynamic elastic bodies derived from the principle of virtual work, from which all element formulations in AStrO begin. The residual vector on the left-hand side of these equations is henceforth denoted $\boldsymbol{R}_u$, so that to satisfy the governing equations of motion is to ensure that $\boldsymbol{R}_u = 0$. Further details of implementation for different cases and element types are given in the following sections of this chapter.

3.1.2 Governing Equations of Heat Conduction

The finite element equations for temperature distribution due to heat conduction in a structure can be developed in a similar fashion as the equations of elasticity. Heat conduction through a solid medium is generally governed by the Poisson equation:

$$\boldsymbol{\nabla}\cdot\boldsymbol{q}\;+\rho C_p\dot{T}-Q=0 \tag{3.23}$$

where Q is the rate of internal heat generation per unit volume, and $\boldsymbol{q}$ is the heat flux vector, governed by the temperature gradient and thermal conductivity $\boldsymbol{k}$ as follows:

$$\boldsymbol{q} = -\left[k_1 \frac{\partial T}{\partial x_1}, k_2 \frac{\partial T}{\partial x_2}, k_3 \frac{\partial T}{\partial x_3}\right]^T \tag{3.24}$$

The variational form of the Poisson equation, following the discussion of the elasticity equations in Section 3.1.1 is

$$\int_\Omega \left(\boldsymbol{\nabla} \cdot \boldsymbol{q} \ + \rho C_p \dot{T} - Q\right) \delta T d\Omega = 0 \tag{3.25}$$

Applying integration by parts on the first term as before,

$$\int_\Omega (\boldsymbol{\nabla} \cdot \boldsymbol{q}) \delta T d\Omega = -\int_\Omega (\boldsymbol{q} \cdot \delta(\boldsymbol{\nabla} T)) d\Omega + \int_S (\boldsymbol{q} \cdot \boldsymbol{n}) \delta T dS \tag{3.26}$$

Then the governing equation becomes

$$-\int_\Omega \left(\boldsymbol{q} \cdot \delta(\boldsymbol{\nabla} T)\right) d\Omega + \int_\Omega \rho C_p \dot{T} \delta T d\Omega - \int_\Omega Q \delta T d\Omega + \int_S (\boldsymbol{q} \cdot \boldsymbol{n}) \delta T dS = 0 \tag{3.27}$$

Adopting the discretized solution for temperature, using $\boldsymbol{\phi}$ to denote the nodal temperature solution parameters gives

$$\begin{gathered} T = \boldsymbol{N} \cdot \boldsymbol{\phi} \\ \delta T = \frac{\partial T}{\boldsymbol{\phi}} \cdot \delta \boldsymbol{\phi} = \boldsymbol{N} \cdot \delta \boldsymbol{\phi} \\ \delta(\boldsymbol{\nabla} T) = \left[\frac{\partial(\boldsymbol{\nabla} T)}{\boldsymbol{\phi}}\right] \delta \boldsymbol{\phi} = [\nabla N]^T \delta \boldsymbol{\phi} \end{gathered} \tag{3.28}$$

Where the basis function gradient matrix is defined such that $\nabla N_{ij} = \frac{\partial N_i}{\partial x_j}$. Substituting Eqs. (3.28) into Eq. (3.27),

$$
\begin{aligned}
&-\int_{\Omega} \boldsymbol{q} \cdot ([\nabla N]^{T} \delta\boldsymbol{\phi}) d\Omega + \int_{\Omega} \rho C_p \dot{T} (\boldsymbol{N} \cdot \delta\boldsymbol{\phi}) d\Omega \\
&-\int_{\Omega} Q (\boldsymbol{N} \cdot \delta\boldsymbol{\phi}) d\Omega + \int_{S} (\boldsymbol{q} \cdot \boldsymbol{n})(\boldsymbol{N} \cdot \delta\boldsymbol{\phi}) dS = 0
\end{aligned}
\tag{3.29}
$$

With Eq. (3.29) holding for any variation of the nodal temperature solution parameters, it follows that

$$
-\int_{\Omega} [\nabla N] \boldsymbol{q} d\Omega + \int_{\Omega} \boldsymbol{N} \rho C_p \dot{T} d\Omega - \int_{\Omega} \boldsymbol{N} Q d\Omega + \int_{S} \boldsymbol{N} (\boldsymbol{q} \cdot \boldsymbol{n}) dS = 0 \tag{3.30}
$$

Eq. (3.30) represents the discretized governing equations for heat conduction in structures derived from the Poisson equation in variational form. The residual on the left-hand side will be denoted $\boldsymbol{R}_{\phi}$ from now on, so that to solve for the temperature solution at any moment in time is to ensure that $\boldsymbol{R}_{\phi} = 0$.

3.1.3 Modeling of Coupled Thermoelastic Response

In general, the thermal and elastic response of a structure are interdependent, as changes in temperature affect strain through thermal expansion, and deformation energy can partially dissipate as heat. AStrO is capable of modeling the coupled thermoelastic response of a structure but makes certain simplifying assumptions. Specifically, thermal material properties such as conductivity and specific heat capacity are assumed to have no appreciable dependence on strain. Furthermore, the heat generated by deformation is assumed to be negligible. In other words, the deformation of the structure is assumed to have a one-way dependence on the temperature distribution.

Although this assumption is not universally accurate, it is considered valid for the primary scope of aero-structural problems for which AStrO is intended. In most cases, strain is expected to be small and within the elastic regime, selected materials will have low internal damping characteristics and deformation rates will not produce significant heat through phenomena like viscoelasticity. Under these assumptions, in any given analysis the temperature distribution of a structure can be obtained first, followed by the deformation solution based on the temperature results. If there was known to be a slight dependence of the thermal response on the deformation of a structure, it could be accounted for by using corrective iterations as implemented by the fluid-structure interface, described in Chapter 2.

To account for the dependence of deformation on the temperature distribution, an adjustment to the definition of total strain is instated. Any point in a structure that is subject to a combination of applied stress and change in temperature will exhibit a measure of strain for each of those contributors. The total resulting strain can be expressed as in Eq. (3.31).

$$\boldsymbol{\epsilon}_{total} = \boldsymbol{\epsilon}_{stress} + \boldsymbol{\epsilon}_{therm} \tag{3.31}$$

In the governing equations of elasticity based on the principle of virtual work, the stress at a point under the assumption of linear elasticity can be expressed

$$\boldsymbol{\sigma} = [C]\boldsymbol{\epsilon} \tag{3.32}$$

where $[C]$ is the stiffness matrix of the local material. But in Eq. (3.32) $\boldsymbol{\epsilon}$ must be the strain due to the applied stress, so in the presence of thermal expansions, the definition from Eq. (3.31) must be implemented to give

$$\boldsymbol{\sigma} = [C]\boldsymbol{\epsilon}_{stress} = [C](\boldsymbol{\epsilon}_{total} - \boldsymbol{\epsilon}_{therm}) \tag{3.33}$$

The strain due to thermal expansion is assumed to be linearly related to temperature, such that the change in temperature from some reference T_{ref} multiplied by a vector of thermal expansion coefficients $\boldsymbol{\alpha}_{TE}$ gives the resulting thermal strain:

$$\boldsymbol{\epsilon}_{therm} = \left(T - T_{ref}\right)\boldsymbol{\alpha}_{TE} = \Delta T \boldsymbol{\alpha}_{TE} \tag{3.34}$$

If the stress in Eq. (3.22) is re-expressed using Eq. (3.33) and (3.34), then the governing equations for the elastic response become

$$\int_\Omega \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C](\boldsymbol{\epsilon}_{total} - \Delta T \boldsymbol{\alpha}_{TE}) d\Omega + \int_\Omega \xi [N]^T \dot{\boldsymbol{u}} d\Omega + \int_\Omega \rho [N]^T \ddot{\boldsymbol{u}} d\Omega \\ - \int_\Omega [N]^T \boldsymbol{f} d\Omega - \int_S [N]^T \boldsymbol{t} dS = 0 \tag{3.35}$$

Separating out the contribution of thermal expansion from the stress term gives

$$\int_\Omega \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C]\boldsymbol{\epsilon}_{total} d\Omega + \int_\Omega \xi [N]^T \dot{\boldsymbol{u}} d\Omega + \int_\Omega \rho [N]^T \ddot{\boldsymbol{u}} d\Omega \\ - \int_\Omega [N]^T \boldsymbol{f} d\Omega - \int_S [N]^T \boldsymbol{t} dS - \int_\Omega \Delta T \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C] \boldsymbol{\alpha}_{TE} d\Omega = 0 \tag{3.36}$$

Since the temperature solution is pre-computed, the effect of thermal expansion manifests as part of the load in the elasticity equations (or the final three terms in Eq. 3.36). Eq. (3.36) is the augmented form of the governing equations for thermoelastic modeling.

3.1.4 Numerical Integration with Gauss Quadrature

As a final note before continuing to more specific element formulations, the governing equations using the finite element method are clearly expressed as integrals over the structural domain. Generally, however, these integrals cannot be evaluated analytically due to complex geometry and lack of closed form solution. Therefore, numerical integration must be employed. Since a structural domain is represented by a collection of elements, the integral over that domain is the summation of integrals over each individual element. That is, for any field quantity $F(\boldsymbol{x})$ within the domain Ω,

$$\int_\Omega F(\boldsymbol{x}) d\Omega = \sum_{j=1}^{n_{el}} \int_{\Omega_j} F(\boldsymbol{x}) d\Omega \tag{3.37}$$

where n_{el} is the number of elements in the domain, and Ω_j is the portion of the domain occupied by element j. For each individual element, integrals are evaluated using Gauss quadrature [80]. To accomplish this, the space occupied by each element is defined in terms of a *natural element coordinate system*, which lines up and scales conveniently with that element's shape and orientation. The domain of the element in natural coordinates has the same upper and lower bounds for every element of a given type, regardless of shape and size. This makes integration in natural element space convenient and straight-forward. The evaluated integral can then be transformed from natural element space to real space using a mapping derived from the element's nodal coordinates. To illustrate, Fig. 3.1 graphically demonstrates the transformation between natural element space and real space for a quadrilateral element. The figure depicts a two-dimensional element for simplicity, even though all element types supported by AStrO are three-dimensional.

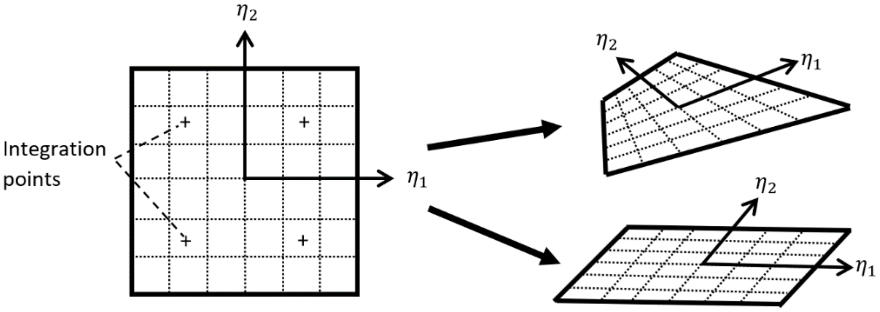


Figure 3.1 Mapping from natural element coordinates to arbitrary shapes in space.

The integral over an element's domain in real space Ω_j is transformed to the integral in natural element coordinate space $\widehat{\Omega}$ as follows

$$\int_{\Omega_j} F(\boldsymbol{x})d\Omega = \int_{\widehat{\Omega}} F(\boldsymbol{\eta})\left(\frac{d\Omega}{d\widehat{\Omega}}\right)d\widehat{\Omega} \tag{3.38}$$

The term $\left(\frac{d\Omega}{d\widehat{\Omega}}\right)$ represents the scaling between a differential element's volume in real space and its volume in natural element space, and its value is the determinant of the Jacobian matrix formed by the partial derivatives of real to natural coordinates, or

$$\left(\frac{d\Omega}{d\widehat{\Omega}}\right) = \left|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\eta}}\right| \tag{3.39}$$

The spatial coordinates $\boldsymbol{x}$ within an element can be interpolated with the same basis as the solution variables, and can be expressed

$$\boldsymbol{x} = [X]\boldsymbol{N} \tag{3.40}$$

where X_{ij} is the x_i coordinate at node j of the element, and N_j is the basis function associated with node j (for further explanation of basis function formulation for different element types, see Section 3.2). Differentiating Eq. (3.40) gives

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\eta}} = [X]\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right] \tag{3.41}$$

In numerical integration using Gauss quadrature, the integral over a region is approximated as a weighted sum over a set of *integration points* throughout the domain, with each term evaluated as the integrand times a predetermined weight value w_i for that point, or

$$\int_{\widehat{\Omega}} G(\boldsymbol{\eta})d\widehat{\Omega} = \sum_{i=1}^{n_{pts}} G(\boldsymbol{\eta}_i)w_i \tag{3.42}$$

where n_{pts} is the number integration points. The integration points and their weights are chosen so that the summation will give the exact value of the integral for any function within a certain space. In AStrO, the Gauss quadrature integration gives exact integrals for elements in which the Jacobian matrix $\left[\frac{\partial x}{\partial \eta}\right]$ is constant throughout the element. For linear-order tetrahedral shaped elements this is always the case, but for hexahedral elements it is contingent on the element having a parallelepiped shape with parallel opposing faces. Numerical errors occur for highly distorted elements for which this is not the case. When mapping from natural space to real space, the integral of Eq. (3.38) using Gauss quadrature, through combination of Eqs. (3.39) through (3.42) becomes

$$\int_{\Omega_j} F(\boldsymbol{x})d\Omega = \int_{\widehat{\Omega}} F(\boldsymbol{\eta})\left(\frac{d\Omega}{d\widehat{\Omega}}\right)d\widehat{\Omega} = \sum_{i=1}^{n_{pts}} F(\boldsymbol{\eta}_i)\left|[X]\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right](\boldsymbol{\eta}_i)\right| w_i \tag{3.43}$$

Integration of all governing equations in AStrO are performed in the manner shown by Eq. (3.43). The following sections give further details for specific element types.

3.2 Element Formulations for Static Analysis

This section is dedicated to the detailed finite element formulation AStrO employs for static, or steady-state structural and thermal analysis. For static cases, the governing equations for elastic and thermal behavior reduce to

$$\begin{gathered}\int_{\Omega}\left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T[C]\boldsymbol{\epsilon}_{total}d\Omega-\int_{\Omega}[N]^T\boldsymbol{f}d\Omega-\int_{S}[N]^T\boldsymbol{t}dS-\int_{\Omega}\Delta T\left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T[C]\boldsymbol{\alpha}_{TE}d\Omega=0\\ -\int_{\Omega}[\nabla N]\boldsymbol{q}d\Omega-\int_{\Omega}\boldsymbol{N}Qd\Omega+\int_{S}\boldsymbol{N}(\boldsymbol{q}\cdot\boldsymbol{n})dS=0\end{gathered} \tag{3.44}$$

The static terms in the governing equations represent the most substantial part of the general formulation, and adding the relatively simple dynamic terms will be discussed in Section 3.3. Three main classes of elements are incorporated into the package: solid continuum elements, shell elements, and beam elements. All three types are three-dimensional elements, but with differences in kinematic assumptions and the way that displacement variables are expressed, as detailed in the next three sections.

3.2.1 Solid Continuum Elements

Solid continuum elements, or simply solid elements, are the most general and versatile of element types. They are capable of representing any arbitrary geometry and make no specific kinematic assumptions other than the conforming of the solution to the space of the basis functions. Solid elements describe the state of a structure purely in terms of displacement or temperature from an undeformed reference state. That is, the fundamental solution variables consist only of displacements and temperatures at the nodes of the discretized mesh. Quantities like rotation, strain, stress, heat flux, etc. at a point in a structure are then derived from these fundamental variables and their gradients as defined by element basis functions.

The basic assumption is that displacement and temperature fields throughout a given element take the form of a linear combination of a set of basis functions, as shown previously in

Eqs. (3.17) and (3.28). Every element type has a primary set of nodal basis functions, each of which evaluates to one at a certain node, and zero at all other nodes. This ensures that the degrees of freedom of the finite element solution correspond to the values of the fundamental field variables at the nodes, which is a highly desirable property for post-processing and interpretation of results.

Some element types have additional basis functions beyond the primary nodal set. These exist to allow the solution to take more natural distributions for certain modes of deformation than what is possible with just the nodal basis, often making the solution more accurate. The degrees of freedom corresponding to these additional basis functions are not associated with nodes, and so are not shared between elements as are the nodal degrees of freedom. As a result, deformation in these modes occurs only in response to the loading and deformation of the nodes of the particular element they belong to, without any direct constraint or interaction between degrees of freedom of neighboring elements. One consequence of this is that the deformation response predicted by an element at its boundary can differ from that predicted by its neighboring element at the same boundary, essentially causing gaps and overlaps in the predicted displacement solution. The solution does, however, remain continuous and consistent at nodes, in contrast with a discontinuous Galerkin formulation. This concept is depicted in Fig. 3.2. Such a solution violates the condition of kinematic compatibility. For this reason, the modes of deformation represented by the additional basis are often called *incompatible modes*, and their degrees of freedom referred to as internal degrees of freedom.

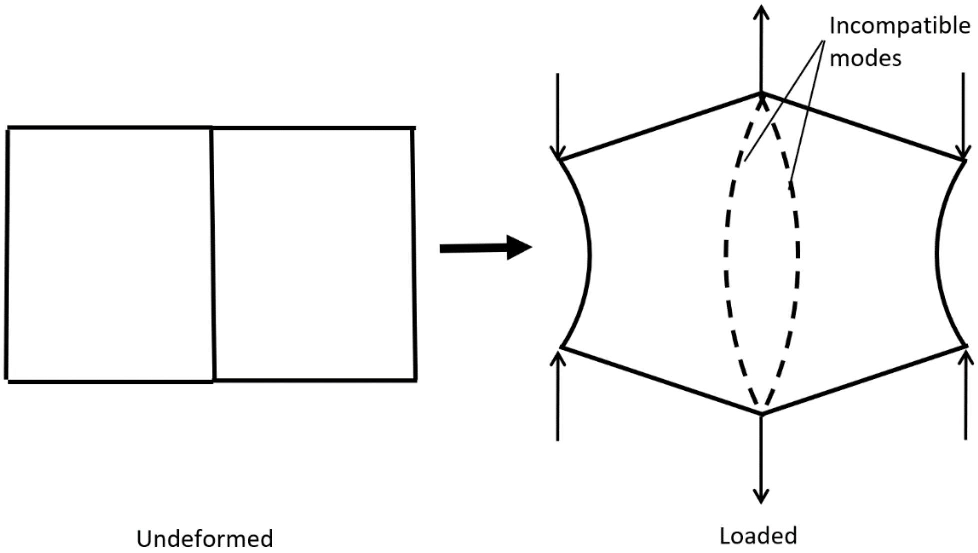


Figure 3.2 Deformation of incompatible modes under loading.

The concept of incompatible modes was first introduced by Bazely *et al.* [81], and was initially met with skepticism. Understandably, the idea of elements that violate kinematic compatibility would strike most as questionable at first glance. Yet since their introduction, elements with incompatible modes have proven to be high-performing and efficient, and are now a tried and true implementation in many commercial codes. The enhanced basis in the assumed solution within an element, coupled with the freedom of deformation in the absence of constraints between internal modes of neighboring elements makes the structure more compliant, particularly under bending loads. This counteracts the natural tendency for standard fully-integrated elements to behave stiffly due to effects like parasitic shear. On top of that, since the incompatible degrees of freedom are internal, they can be condensed out of the final global equation matrix, so they add negligible cost to computing the solution. More on this topic will be discussed in Section 3.4.

Table 3.1 shows diagrams of each type of solid continuum element implemented by AStrO, with their basis functions and integration points and weights employed by Gauss quadrature.

Table 3.1 Geometric and basis definition for solid continuum elements

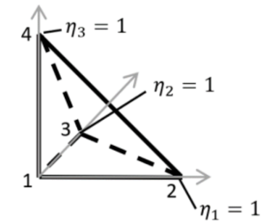
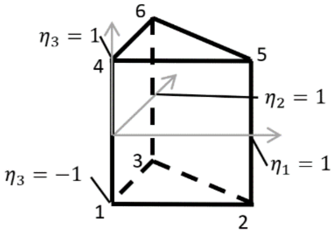
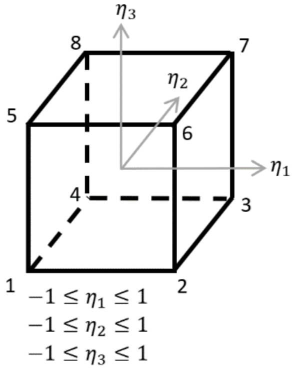
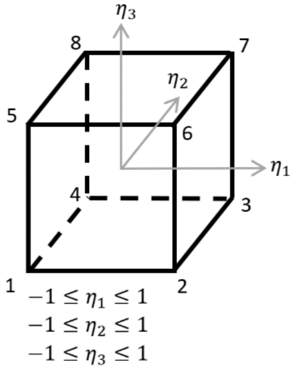
Element Type	**Basis Functions**	**Integration Points** $[\boldsymbol{\eta_1}, \boldsymbol{\eta_2}, \boldsymbol{\eta_3}, \boldsymbol{w}]$
four-node tetrahedral  4 $\eta_3 = 1$ $\eta_2 = 1$ 3 1 2 $\eta_1 = 1$	$N_1 = 1 - \eta_1 - \eta_2 - \eta_3$ $N_2 = \eta_1$ $N_3 = \eta_2$ $N_4 = \eta_3$	$p_1\colon \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{6}\right]$
six-node wedge  6 $\eta_3 = 1$ 4 5 $\eta_2 = 1$ $\eta_3 = -1$ 3 $\eta_1 = 1$ 1 2	$N_1 = \frac{1}{2}(1 - \eta_1 - \eta_2)(1 - \eta_3)$ $N_2 = \frac{1}{2}\eta_1(1 - \eta_3)$ $N_3 = \frac{1}{2}\eta_2(1 - \eta_3)$ $N_4 = \frac{1}{2}(1 - \eta_1 - \eta_2)(1 + \eta_3)$ $N_5 = \frac{1}{2}\eta_1(1 + \eta_3)$ $N_6 = \frac{1}{2}\eta_2(1 + \eta_3)$	$p_1\colon \left[\frac{1}{3}, \frac{1}{3}, \frac{-1}{\sqrt{3}}, \frac{1}{2}\right]$ $p_2\colon \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{\sqrt{3}}, \frac{1}{2}\right]$

Table 3.1 (continued)

eight-node hexahedral  η_3 8 7 η_2 5 6 η_1 4 3 1 2 $-1 \leq \eta_1 \leq 1$ $-1 \leq \eta_2 \leq 1$ $-1 \leq \eta_3 \leq 1$	$N_1 = \frac{1}{8}(1-\eta_1)(1-\eta_2)(1-\eta_3)$ $N_2 = \frac{1}{8}(1+\eta_1)(1-\eta_2)(1-\eta_3)$ $N_3 = \frac{1}{8}(1+\eta_1)(1+\eta_2)(1-\eta_3)$ $N_4 = \frac{1}{8}(1-\eta_1)(1+\eta_2)(1-\eta_3)$ $N_5 = \frac{1}{8}(1-\eta_1)(1-\eta_2)(1+\eta_3)$ $N_6 = \frac{1}{8}(1+\eta_1)(1-\eta_2)(1+\eta_3)$ $N_7 = \frac{1}{8}(1+\eta_1)(1+\eta_2)(1+\eta_3)$ $N_8 = \frac{1}{8}(1-\eta_1)(1+\eta_2)(1+\eta_3)$	$p_1: \left[\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_2: \left[\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_3: \left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_4: \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_5: \left[\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ $p_6: \left[\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ $p_7: \left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ $p_8: \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$
eight-node hexahedral with incompatible modes  η_3 8 7 η_2 5 6 η_1 4 3 1 2 $-1 \leq \eta_1 \leq 1$ $-1 \leq \eta_2 \leq 1$ $-1 \leq \eta_3 \leq 1$	$N_1 = \frac{1}{8}(1-\eta_1)(1-\eta_2)(1-\eta_3)$ $N_2 = \frac{1}{8}(1+\eta_1)(1-\eta_2)(1-\eta_3)$ $N_3 = \frac{1}{8}(1+\eta_1)(1+\eta_2)(1-\eta_3)$ $N_4 = \frac{1}{8}(1-\eta_1)(1+\eta_2)(1-\eta_3)$ $N_5 = \frac{1}{8}(1-\eta_1)(1-\eta_2)(1+\eta_3)$ $N_6 = \frac{1}{8}(1+\eta_1)(1-\eta_2)(1+\eta_3)$ $N_7 = \frac{1}{8}(1+\eta_1)(1+\eta_2)(1+\eta_3)$ $N_8 = \frac{1}{8}(1-\eta_1)(1+\eta_2)(1+\eta_3)$ *Internal:* $N_9 = (1+\eta_1)(1-\eta_1)$ $N_{10} = (1+\eta_2)(1-\eta_2)$ $N_{11} = (1+\eta_3)(1-\eta_3)$	$p_1: \left[\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_2: \left[\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_3: \left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_4: \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ $p_5: \left[\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ $p_6: \left[\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ $p_7: \left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ $p_8: \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$

Given the basis spaces in Table 3.1, there are two alternate ways to express the assumed solution for displacement within an element. One way is to construct the displacement degrees of freedom as a one-dimensional vector, and distribute the basis functions as appropriate in a 3 X n_{dof} matrix, where n_{dof} is the number of degrees of freedom, to multiply from the left to give the displacement vector as done in Eq. (3.17):

$$\boldsymbol{u} = [N]\boldsymbol{U} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & \cdots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & \end{bmatrix} \begin{Bmatrix} U_{x_1,node\ 1} \\ U_{x_2,node\ 1} \\ U_{x_3,node\ 1} \\ U_{x_1,node\ 2} \\ U_{x_2,node\ 2} \\ U_{x_3,node\ 3} \\ \vdots \end{Bmatrix} \tag{3.45}$$

A second way is to assemble the nodal degrees of freedom in a two-dimensional matrix, and multiply the basis functions as a one-dimensional vector from the right, as shown below.

$$\boldsymbol{u} = [U]\boldsymbol{N} = \begin{bmatrix} U_{x_1,node\ 1} & U_{x_1,node\ 2} & U_{x_1,node\ 3} & \\ U_{x_2,node\ 1} & U_{x_2,node\ 2} & U_{x_2,node\ 3} & \cdots \\ U_{x_3,node\ 1} & U_{x_3,node\ 2} & U_{x_3,node\ 3} & \end{bmatrix} \begin{Bmatrix} N_1 \\ N_2 \\ \vdots \end{Bmatrix} \tag{3.46}$$

Both forms are equally correct, but are each well-suited to different contexts. The form in Eq. (3.45) is more convenient for assembling the global matrix equations, while the form in Eq. (3.46) is more compact and efficient for evaluating certain quantities on the element level. The two forms are equivalent for expressing temperature, since it is a scalar field. The form in Eq. (3.46) will be used for the remainder of this section.

As mentioned previously, the nodal degrees of freedom defining the state of any structure represent the values of the field variables at the nodes. Because these degrees of freedom are shared between elements, they are all expressed in a single global coordinate system for a given structure or assembly. This makes it possible to add the individual element matrices to form the global equations as indicated in Eq. (3.37). However, individual sections or elements of a structure may often have local coordinate systems in which material properties or geometric characteristics are most conveniently expressed. For these cases the displacement field of a solid element can be transformed to local coordinates with a direction cosine matrix $[\alpha_L]$ as follows

$$\boldsymbol{u}_L = [\alpha_L][U]\boldsymbol{N} = [\alpha_L] \begin{bmatrix} \boldsymbol{U}_1^T \\ \boldsymbol{U}_2^T \\ \boldsymbol{U}_3^T \end{bmatrix} \boldsymbol{N} \tag{3.47}$$

The matrix $[\alpha_L]$ is constructed either from a local orientation definition supplied by the user in the model input file or from the nodal coordinates of an element if such an orientation is not specified. It can be useful to refer to the individual rows of the matrix of displacement degrees of freedom $[U]$ as vectors denoted $\boldsymbol{U}_i$ as shown in Eq. (3.47). The strain field within an element can then be derived as a function of displacement variables, which is a critical ingredient in the governing equations of elasticity. The form of strain employed by AStrO for solid elements is Green-Lagrange strain [80], defined in terms of displacement gradients $\nabla u_{ij} = \frac{\partial u_i}{\partial x_j}$ as a second-order tensor as

$$[\epsilon] = \frac{1}{2}([\nabla u] + [\nabla u]^T + [\nabla u]^T[\nabla u]), \tag{3.48}$$

For geometrically linear analysis, where displacements and rotations in the structure are assumed to be small, the final term in Eq. (3.48) is omitted. Table 3.2 shows the form for the strain field under the assumed solution, as well as derivatives with respect to displacement variables for geometrically linear and nonlinear problems. $[I]$ denotes the identity matrix.

Table 3.2 Definition of Green-Lagrange strain for solid elements in terms of discrete solution.

	Linear Geometry	**Nonlinear Geometry**
$[\epsilon]$	$\frac{1}{2}([\alpha_L][U][\nabla N] + [\nabla N]^T[U]^T[\alpha_L]^T)$	$\frac{1}{2}([\alpha_L][U][\nabla N] + [\nabla N]^T[U]^T[\alpha_L]^T + [\nabla N]^T[U]^T[U][\nabla N])$
$\frac{\partial \epsilon_{ij}}{\partial U_{kl}}$	$\frac{1}{2}\left(\alpha_{L,ik}\frac{\partial N_l}{\partial x_j} + \alpha_{L,jk}\frac{\partial N_l}{\partial x_i}\right)$	$\frac{1}{2}\left(\alpha_{L,ik}\frac{\partial N_l}{\partial x_j} + \alpha_{L,jk}\frac{\partial N_l}{\partial x_i} + \left(\boldsymbol{U}_k \cdot \frac{\partial \boldsymbol{N}}{\partial x_i}\right)\frac{\partial N_l}{\partial x_j} + \left(\boldsymbol{U}_k \cdot \frac{\partial \boldsymbol{N}}{\partial x_j}\right)\frac{\partial N_l}{\partial x_i}\right)$
$\frac{\partial \epsilon_{ij}}{\partial U_{kl}\partial U_{pq}}$	0	$\frac{1}{2}I_{kp}\left(\frac{\partial N_l}{\partial x_i}\frac{\partial N_q}{\partial x_j} + \frac{\partial N_l}{\partial x_j}\frac{\partial N_q}{\partial x_i}\right)$

In Table 3.2 the gradients of the basis functions are evaluated using the mapping from natural element coordinate space. The Jacobian matrix from Eq. (3.41) can be transformed to an element's local coordinate system using $[\alpha_L]$, then inverted to give the basis function gradients in the local coordinate system as follows:

$$[\nabla N] = \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right]\left[\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{x}}\right] = \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right]\left([\alpha_L][X]\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right]\right)^{-1} \tag{3.49}$$

When solving the Poisson equation for temperature distribution, the same form can be used to evaluate heat flux:

$$\begin{gathered}\boldsymbol{q} = -\left[k_1\left(\frac{\partial \boldsymbol{N}}{\partial x_1}\cdot\boldsymbol{\phi}\right), k_2\left(\frac{\partial \boldsymbol{N}}{\partial x_2}\cdot\boldsymbol{\phi}\right), k_3\left(\frac{\partial \boldsymbol{N}}{\partial x_3}\cdot\boldsymbol{\phi}\right)\right]^T \\ \left[\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{\phi}}\right] = -\left[k_1\frac{\partial \boldsymbol{N}}{\partial x_1}, k_2\frac{\partial \boldsymbol{N}}{\partial x_2}, k_3\frac{\partial \boldsymbol{N}}{\partial x_3}\right]^T\end{gathered} \tag{3.50}$$

Finally, AStrO assumes linear elastic material behavior for all analysis, and that the material of any given section falls under the category of orthotropic materials, which have three mutually orthogonal planes of symmetry. Note that fully isotropic and transversely isotropic materials fall under this category as well. Such materials can be described with no more than nine independent material constants, which are usually expressed in the form of three orthogonal elastic moduli (E_1, E_2, E_3), three orthogonal measures of Poisson's ratio $(\nu_{12}, \nu_{13}, \nu_{23})$, and three orthogonal shear moduli (G_{12}, G_{13}, G_{23}). With these constants, the material stiffness matrix relating stress to strain can be constructed as follows:

$$\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{13}\\ \sigma_{23}\end{Bmatrix} = \begin{bmatrix}\frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{13}}{E_1} & 0 & 0 & 0\\ & \frac{1}{E_2} & -\frac{\nu_{23}}{E_2} & 0 & 0 & 0\\ & & \frac{1}{E_3} & 0 & 0 & 0\\ & & & \frac{1}{G_{12}} & 0 & 0\\ & (sym) & & & \frac{1}{G_{13}} & 0\\ & & & & & \frac{1}{G_{23}}\end{bmatrix}^{-1} \begin{Bmatrix}\epsilon_{11}\\ \epsilon_{22}\\ \epsilon_{33}\\ \gamma_{12}\\ \gamma_{13}\\ \gamma_{23}\end{Bmatrix} \tag{3.51}$$

It is worth mentioning that strictly speaking, the appropriate work conjugate to Green-Lagrange strain is second Piola-Kirchhoff stress [80], which would need to be derived experimentally for a given material undergoing large strains. When strain in a structure is small, despite possibly large displacements, second Piola-Kirchhoff stress is nearly equivalent to Cauchy stress in a reference frame effectively aligning with a point in a structure, and the linear-elastic definition of Eq. (3.51) only holds for such cases. This is exactly the assumption currently made by AStrO, and the proper modifications would need to be made to consider large strains, or material nonlinearity.

Using the definitions for strain in Table 3.2, elastic properties in Eq. (3.51), and heat flux in Eqs. (3.49) and (3.50), the discretized finite element equations for static thermoelastic analysis in Eq. (3.44) can be assembled. The solution process is described further in Section 3.4.

3.2.2 Shell Elements

Another type of element implemented in AStrO is the shell element, a special purpose type of element that is meant to model structures that form a surface in three-dimensional space, and are relatively thin in the direction normal to that surface, such as hollow shells and flat plates. Shell elements are often used for high aspect ratio structures such as aircraft wings and turbine blades. For such cases, certain kinematic assumptions can be made which can make structural analysis more efficient than it would be using solid continuum elements. At the same time, shell elements are somewhat less versatile since these kinematic assumptions must be valid for them to be applicable.

The kinematic assumptions arise from the high aspect ratio geometry of shell structures. Under the majority of physically conceivable loading conditions, the out-of-plane stresses, or components of stress in the direction normal to the shell structure's mid-plane surface, are negligible compared to the in-plane stresses. The implications of this condition in terms of deformation are summed up in one of the most common forms of classical plate theory, Kirchhoff Plate Theory [82]. The Kirchhoff hypotheses state that for a plate or shell structure under loading, the following assumptions regarding deformation hold:

1. Straight lines within the shell that are perpendicular to the midplane of the shell before deformation, or transverse normals, remain straight and perpendicular to the midplane after deformation.
2. Transverse normals do not change in length due to deformation.

Under the above assumptions for structures under small deformations, the displacement field within a shell can be expressed in terms of displacements at the midplane:

$$
\begin{aligned}
u_1 &= u_{m,1} - x_3 \frac{\partial u_{m,3}}{\partial x_1} \\
u_2 &= u_{m,2} - x_3 \frac{\partial u_{m,3}}{\partial x_2} \\
u_3 &= u_{m,3}
\end{aligned}
\tag{3.52}
$$

In Eq. (3.52) $\boldsymbol{u}_m$ represents the displacement at the midplane of the shell, the 3-direction is normal to the midplane and x_3 is the perpendicular distance from the midplane projected to a point, as depicted in Fig. 3.3.

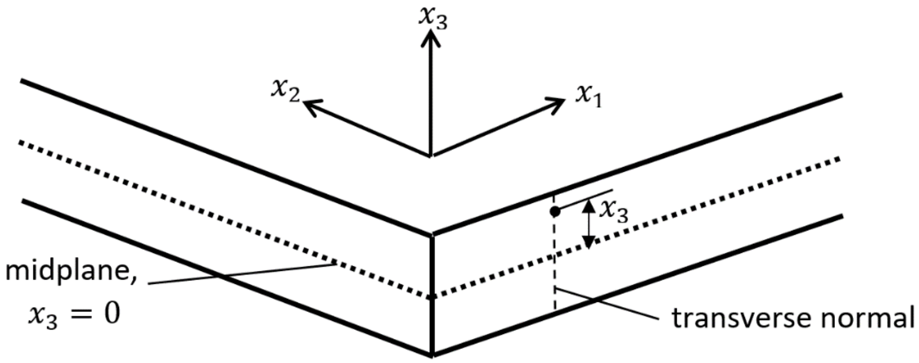


Figure 3.3 Diagram of shell element for kinematic definitions.

Numerous variations of plate and shell theory exist, as do numerous approaches to finite element formulation of shell structures. When considering an approach to take for implementation in AStrO, it was decided that certain guidelines ought to be followed based on long-term goals for the work.

First, it was decided that for shell elements the governing equations based on the principle of virtual work should remain in terms of fundamental stresses and strains as defined in Eqs. (3.10) and (3.33). Technically, the principle of virtual work in a deformable body can be expressed not only in terms of stress and strain, but also in terms of any alternate set of deformation modes and their corresponding work conjugates, and it can certainly be advantages to do so. For example, in classical composite laminate theory, the behavior of a layered composite is generally described in terms of midplane strains and curvatures, and the corresponding forces and moments per unit length. This way the virtual work can be evaluated by integrating the forces/moments times strains and curvatures over the midplane area, making it a two-dimensional domain of integration. However, constructing the matrix which maps in-plane strains and curvatures to forces and moments requires pre-integrating the shell material properties through the thickness. This creates an additional layer of complexity in sensitivity analysis when obtaining sensitivities of objectives with respect to material properties, orientations and shell thickness. Keeping the formulation in terms of fundamental stress and strain is more direct and integrates naturally with the existing infrastructure for solid elements.

Second, it was decided that the shell formulation should include nodal rotations in all three global dimensions as fundamental degrees of freedom. Equation (3.52) indicates that under Kirchhoff's assumptions, the behavior of a shell can, in theory, be described purely in terms of midplane displacement and its derivatives. However, a number of potential difficulties arise from implementing this form exactly as is. In practice, it is often useful to apply loading to shell structures in the form of concentrated or distributed moments, as opposed to forces. Also, the resulting rotations at points throughout a structure can be of interest when interpreting results and applying constraints. In a purely displacement-based formulation, those moments and rotations would have to be transformed to and from equivalents in terms of displacement variables. For geometrically linear analysis, this usually would not be a major issue, but with large displacements and rotations the process could become cumbersome and numerically unstable. Rotations would also be discontinuous at nodes, leading to further complications. Letting nodal rotations be fundamental degrees of freedom and incorporating them into the principle of virtual work is a more direct and robust approach.

With the above goals in mind, the shell element formulation implemented by AStrO begins with the definition of field displacement in Eq. (3.52), and replaces the derivatives of out-of-plane displacement with appropriate in-plane rotations:

$$\begin{aligned} u_1 &= u_{m,1} + x_3\theta_2 \\ u_2 &= u_{m,2} - x_3\theta_1 \\ u_3 &= u_{m,3} \end{aligned} \tag{3.53}$$

The out-of-plane component of rotation θ_3 does not appear in the above definition of field displacement alone. To complete the full kinematic definition of the relationship between displacements and rotations, an additional constraint must be imposed. Figure 3.4 illustrates the kinematic relationship between in-plane displacements $u_{m,1}$ and $u_{m,2}$ and the out-of-plane rotation θ_3, using the two specific deformation cases of rigid-body rotation about the 3-axis and pure shear deformation in the 1-2 plane. The net out-of-plane rotation at a point is found by averaging the apparent rotations of the in-plane axial directions due to deformation. The assumption of small angle changes, and the appropriate approximation is used in this definition.

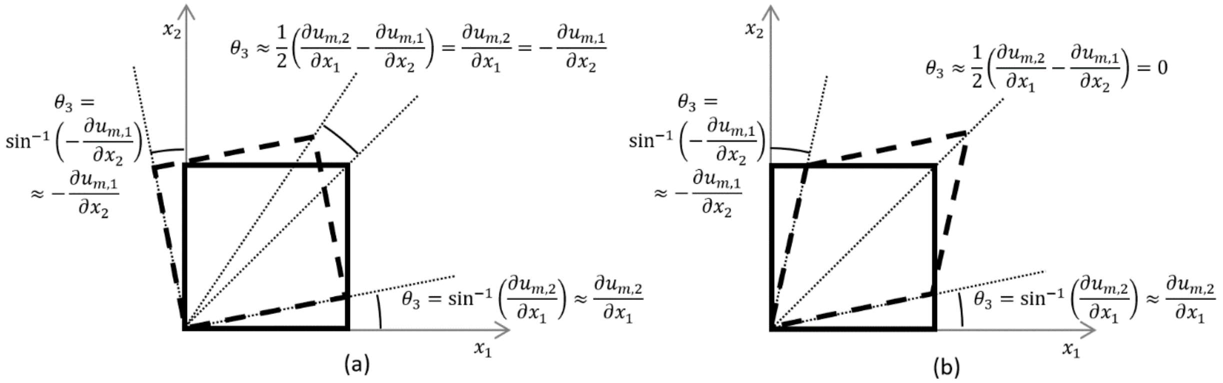

x_2
$\theta_3 \approx \frac{1}{2}\left(\frac{\partial u_{m,2}}{\partial x_1} - \frac{\partial u_{m,1}}{\partial x_2}\right) = \frac{\partial u_{m,2}}{\partial x_1} = -\frac{\partial u_{m,1}}{\partial x_2}$
$\theta_3 = \sin^{-1}\left(-\frac{\partial u_{m,1}}{\partial x_2}\right) \approx -\frac{\partial u_{m,1}}{\partial x_2}$
$\theta_3 = \sin^{-1}\left(\frac{\partial u_{m,2}}{\partial x_1}\right) \approx \frac{\partial u_{m,2}}{\partial x_1}$
x_1
(a)
x_2
$\theta_3 \approx \frac{1}{2}\left(\frac{\partial u_{m,2}}{\partial x_1} - \frac{\partial u_{m,1}}{\partial x_2}\right) = 0$
$\theta_3 = \sin^{-1}\left(-\frac{\partial u_{m,1}}{\partial x_2}\right) \approx -\frac{\partial u_{m,1}}{\partial x_2}$
$\theta_3 = \sin^{-1}\left(\frac{\partial u_{m,2}}{\partial x_1}\right) \approx \frac{\partial u_{m,2}}{\partial x_1}$
x_1
(b)

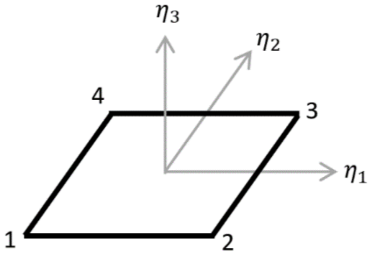
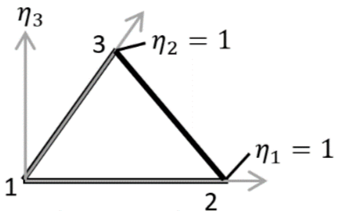
Figure 3.4 Kinematic relationship between in-plane displacements and out-of-plane rotation at a point, illustrated for a) rigid-body rotation and b) pure shear deformation.

The constraint illustrated in Fig. 3.4 and stated below in Eq. (3.54) is enforced in the formulation of shell elements in the form of an *artificial* strain component, as will be discussed further shortly. This completes the kinematic coupling relationship between all six nodal degrees of freedom describing the state of motion for this type of element.

$$\theta_3 = \frac{1}{2}\left(\frac{\partial u_{m,2}}{\partial x_1} - \frac{\partial u_{m,1}}{\partial x_2}\right) \qquad (3.54)$$

As with solid continuum elements, the field variables take the assumed form of the basis functions of a given element, except that with shell elements these variables consist of both displacements and rotations in the midplane of the element. The element nodes define the two-dimensional domain of the midplane, and the thickness of the element is defined as a separate physical parameter. There are two types of shell elements implemented in AStrO, with schematics given in Table 3.3.

Table 3.3 Geometric and basis definition for shell elements

Element Type	**Basis Functions**	**Integration pts** $[\boldsymbol{\eta_1}, \boldsymbol{\eta_2}, \boldsymbol{\eta_3}, \boldsymbol{w}]$
four-node quadrilateral  $-1 \leq \eta_1 \leq 1$ $-1 \leq \eta_2 \leq 1$ $-1 \leq \eta_3 \leq 1$	*Nodal:* $N_1 = \frac{1}{4}(1-\eta_1)(1-\eta_2)$ $N_2 = \frac{1}{4}(1+\eta_1)(1-\eta_2)$ $N_3 = \frac{1}{4}(1+\eta_1)(1+\eta_2)$ $N_4 = \frac{1}{4}(1-\eta_1)(1+\eta_2)$ *Internal, $u_{m,3}$:* $N_5 = \frac{1}{2}(1+\eta_1)(1-\eta_1)(1-\eta_2)$ $N_6 = \frac{1}{2}(1+\eta_1)(1+\eta_2)(1-\eta_2)$ $N_7 = \frac{1}{2}(1+\eta_1)(1-\eta_1)(1+\eta_2)$ $N_8 = \frac{1}{2}(1-\eta_1)(1+\eta_2)(1-\eta_2)$	p_1: $\left[\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ p_2: $\left[\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ p_3: $\left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ p_4: $\left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, 1\right]$ p_5: $\left[\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ p_6: $\left[\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ p_7: $\left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$ p_8: $\left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 1\right]$
three-node triangle  $-1 \leq \eta_3 \leq 1$	*Nodal:* $N_1 = 1 - \eta_1 - \eta_2$ $N_2 = \eta_1$ $N_3 = \eta_2$ *Internal, $u_{m,3}$:* $N_4 = \eta_1(1-\eta_1-\eta_2)$ $N_5 = \eta_1\eta_2$ $N_6 = \eta_2(1-\eta_1-\eta_2)$	p_1: $\left[\frac{1}{2}, 0, \frac{-1}{\sqrt{3}}, \frac{1}{6}\right]$ p_2: $\left[\frac{1}{2}, \frac{1}{2}, \frac{-1}{\sqrt{3}}, \frac{1}{6}\right]$ p_3: $\left[0, \frac{1}{2}, \frac{-1}{\sqrt{3}}, \frac{1}{6}\right]$ p_4: $\left[\frac{1}{2}, 0, \frac{1}{\sqrt{3}}, \frac{1}{6}\right]$ p_5: $\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{3}}, \frac{1}{6}\right]$ p_6: $\left[0, \frac{1}{2}, \frac{1}{\sqrt{3}}, \frac{1}{6}\right]$

As indicated in Table 3.3, shell elements have internal modes/degrees of freedom that only apply to out-of-plane displacement. This way the displacement can take on curvature as needed to model the behavior accurately without adding nodes to the global system.

Given the basis functions for each element, the midplane displacement field and midplane rotation field can be defined under the assumed solution as

$$\begin{gathered}\boldsymbol{u}_m = [U]\boldsymbol{N} \\ \boldsymbol{\theta} = [\theta]\boldsymbol{N}\end{gathered} \tag{3.55}$$

Due to the selective interpolation of $u_{m,3}$ as discussed, the matrices of degrees of freedom and the basis function vector above take the following structure:

$$[U] = \begin{bmatrix} \boldsymbol{U}_{nd,1}^T & 0 \\ \boldsymbol{U}_{nd,2}^T & 0 \\ \boldsymbol{U}_{nd,3}^T & \boldsymbol{U}_{int,3}^T \end{bmatrix}$$

$$[\theta] = \begin{bmatrix} \boldsymbol{\theta}_1^T & 0 \\ \boldsymbol{\theta}_2^T & 0 \\ \boldsymbol{\theta}_3^T & 0 \end{bmatrix} \tag{3.56}$$

$$\boldsymbol{N} = \begin{Bmatrix} \boldsymbol{N}_{nd} \\ \boldsymbol{N}_{int} \end{Bmatrix}$$

Substituting Eq. (3.55) into Eq. (3.53), the displacement field in a shell element with the assumed form can be expressed

$$\boldsymbol{u} = [U]\boldsymbol{N} + \begin{Bmatrix} x_3(\boldsymbol{\theta}_2 \cdot \boldsymbol{N}_{nd}) \\ -x_3(\boldsymbol{\theta}_1 \cdot \boldsymbol{N}_{nd}) \\ 0 \end{Bmatrix} \tag{3.57}$$

At this point it is important to note that, because the above definition of displacement is only valid for small displacements and rotations in a coordinate system where the 3-direction is normal to the shell midplane, a somewhat different approach is warranted for formulating linear geometry vs. nonlinear geometry for shell elements than for solid elements. Recall that for solid elements, the Green-Lagrange definition of strain in Eq. (3.48) contains nonlinear terms, which are simply omitted under the assumption of linear geometry. But the displacement gradients in that form are always computed in the original coordinate system. This works in part because the field displacement definition for solid elements is valid to any degree of deformation. But for shell elements, if Eq. (3.57) is to be used as a definition for the displacement field, some adjustment must be made to make it generally applicable.

The key in making such an adjustment lies in realizing that the primary source of error in Eq. (3.57) for large displacements is rigid body rotation. As a structure undergoes large rotations, the direction normal to the shell changes, and the projection from the midplane to a given point in the shell becomes a complex trigonometric function of $\boldsymbol{\theta}$. But, for any state of deformation of a given element, there exists a corresponding state that produces the same strain field, but with no rigid body rotation. The approach implemented by AStrO is to transform the nodal degrees of freedom of each element from global coordinates to a coordinate system that follows the element dynamically as the structure deforms, effectively removing rigid body rotation. This is referred to as the *instantaneous* coordinate system.

The distinction between linear geometry and nonlinear geometry is how to make the transformation from global to instantaneous degrees of freedom. Under the assumption of linear geometry, the orientation of any given element can be assumed constant, and the instantaneous coordinate system is one that lines up appropriately with the element in the undeformed state, or the *local* coordinate system. In this case a simple vector transformation using the direction cosine matrix for the local coordinate system is all that is needed. Under the assumption of nonlinear geometry, the instantaneous coordinate system for a given element is derived from the overall rotation of the element, or specifically the average nodal rotation. It begins as the local coordinate system in the undeformed state, then as the structure undergoes displacement the instantaneous coordinate system is updated by rotating the principal directions of the local coordinate system by the overall element rotation. The instantaneous displacement degrees of freedom are then found from the overall position vector of each node before and after deformation. Figure 3.5 illustrates this transformation graphically.

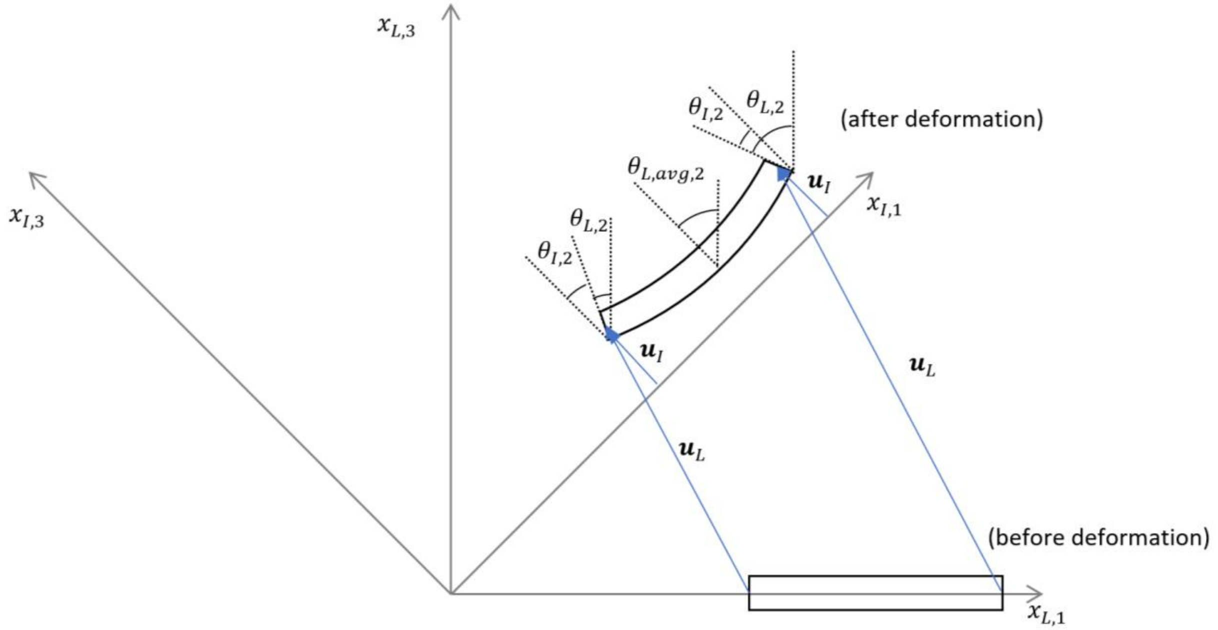


Figure 3.5 Transformation of displacements and rotations of a shell structure from a static, local coordinate system to the updated, instantaneous coordinate system after deformation.

Table 3.4 describes the relationships between global and instantaneous displacements and rotations mathematically. In the table, $[X_G]$, $[U_G]$ and $[\theta_G]$ are the nodal initial positions, displacements and rotations in global coordinates. $[\theta_{G,avg}]$ is a matrix which contains the average nodal rotation in global coordinates in each column. $[U_I]$ and $[\theta_I]$ are nodal displacements and rotations in instantaneous coordinates. $[\alpha_L]$ and $[\alpha_I]$ are the direction cosine matrices transforming from global coordinates to local, and instantaneous coordinates respectively. n_{nd} is the number of nodes in the element, $[I]$ is the identity matrix and $[\tilde{I}]$ is used to denote a matrix whose values are all one.

Table 3.4 Definition of instantaneous nodal degrees of freedom in a shell element.

Quantity	Linear Geometry	Nonlinear Geometry
$[U_I]$	$[\alpha_L][U_G]$	$[\alpha_I]([X_G]+[U_G])-[\alpha_L][X_G]$
$\frac{\partial U_{I,ij}}{\partial U_{G,kl}}$	$\alpha_{L,ik}I_{jl}$	$\alpha_{I,ik}I_{jl}$
$\frac{\partial [U_I]}{\partial \theta_{G,ij}}$	0	$\frac{\partial[\alpha_I]}{\partial\theta_{G,ij}}([X_G]+[U_G])$
$\frac{\partial U_{I,ij}}{\partial U_{G,kl}\partial\theta_{G,pq}}$	0	$\frac{\partial\alpha_{I,ik}}{\partial\theta_{G,pq}}I_{jl}$
$\frac{\partial [U_I]}{\partial\theta_{G,ij}\partial\theta_{G,kl}}$	0	$\frac{\partial^2[\alpha_I]}{\partial\theta_{G,ij}\partial\theta_{G,kl}}([X_G]+[U_G])$
$[\theta_I]$	$[\alpha_L][\theta_G]$	$[\alpha_I]\big([\theta_G]-[\theta_{G,avg}]\big)$
$\frac{\partial\theta_{I,ij}}{\partial\theta_{G,kl}}$	$\alpha_{L,ik}I_{jl}$	$\left(\frac{\partial[\alpha_I]}{\partial\theta_{G,kl}}\big([\theta_G]-\big[\theta_{G,avg}\big]\big)\right)_{ij}+\alpha_{I,ik}\left(I_{jl}-\frac{1}{n_{nd}}\tilde{I}_{jl}\right)$
$\frac{\partial^2\theta_{I,ij}}{\partial\theta_{G,kl}\partial\theta_{G,pq}}$	0	$\left(\frac{\partial^2[\alpha_I]}{\partial\theta_{G,kl}\partial\theta_{G,pq}}\big([\theta_G]-\big[\theta_{G,avg}\big]\big)\right)_{ij}+\frac{\partial\alpha_{I,ip}}{\theta_{G,kl}}\left(I_{jq}-\frac{1}{n_{nd}}\tilde{I}_{jq}\right)$ $+\frac{\partial\alpha_{I,ik}}{\theta_{G,pq}}\left(I_{jl}-\frac{1}{n_{nd}}\tilde{I}_{jl}\right)$

Using the instantaneous nodal degrees of freedom from Table 3.4, a generally applicable definition of the displacement field in instantaneous coordinates can be formed:

$$\boldsymbol{u} = [U_I]\boldsymbol{N} + \begin{Bmatrix} x_3(\boldsymbol{\theta}_{I,2} \cdot \boldsymbol{N}_{nd}) \\ -x_3(\boldsymbol{\theta}_{I,1} \cdot \boldsymbol{N}_{nd}) \\ 0 \end{Bmatrix} \tag{3.58}$$

From Eq. (3.58) the strain field can be defined. The assumptions of the Kirchhoff hypotheses imply that out-of-plane strains are zero. But here the out-of-plane shear strains are still included in the formulation in order to enforce the constraint between rotations and out-of-plane displacements.

$$\begin{aligned}
\epsilon_{11} &= \frac{\partial u_1}{\partial x_1} = (\boldsymbol{U}_{I,1} + x_3\boldsymbol{\theta}_{I,2}) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} \\
\epsilon_{22} &= \frac{\partial u_2}{\partial x_2} = (\boldsymbol{U}_{I,2} - x_3\boldsymbol{\theta}_{I,1}) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2} \\
\gamma_{12} &= \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} = (\boldsymbol{U}_{I,1} + x_3\boldsymbol{\theta}_{I,2}) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2} + (\boldsymbol{U}_{I,2} - x_3\boldsymbol{\theta}_{I,1}) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} \\
\gamma_{13} &= \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} = (\boldsymbol{\theta}_{I,2} \cdot \boldsymbol{N}_{nd}) + \left(\boldsymbol{U}_{I,nd,3} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) + \left(\boldsymbol{U}_{I,int,3} \cdot \frac{\partial \boldsymbol{N}_{int}}{\partial x_1}\right) \\
\gamma_{23} &= \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} = -(\boldsymbol{\theta}_{I,1} \cdot \boldsymbol{N}_{nd}) + \left(\boldsymbol{U}_{I,nd,3} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right) + \left(\boldsymbol{U}_{I,int,3} \cdot \frac{\partial \boldsymbol{N}_{int}}{\partial x_2}\right)
\end{aligned} \tag{3.59}$$

To enforce the previously mentioned constraint on the out-of-plane rotation in Eq. (3.54), the following “artificial strain” is defined:

$$\gamma_a = 2\theta_3 + \frac{\partial u_{m,1}}{\partial x_2} - \frac{\partial u_{m,2}}{\partial x_1} = 2(\boldsymbol{\theta}_{I,3} \cdot \boldsymbol{N}_{nd}) + \left(\boldsymbol{U}_{I,1} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right) - \left(\boldsymbol{U}_{I,2} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) \tag{3.60}$$

The logic behind this artificial strain is that, since the governing equations of elastic bodies seek to minimize the total strain energy in a structure, the system’s behavior will drive down all strains, with priority gauged by their associated stiffnesses. The artificial strain represents the residual of the kinematic constraint on θ_3 in Eq. (3.54), so driving it down by extension satisfies the constraint. A stiffness is assigned to the artificial strain with a value equal to G_{12}, to make it comparable to the other participating strains, and it is put in place of out-of-plane normal strain, since that component is inapplicable to shells. Since rigid body rotation is eliminated in the definition of instantaneous degrees of freedom, Eq. (3.60) is valid for both linear and nonlinear geometry. All the derivatives of strain with respect to global degrees of freedom are straight-forward with the usage of Table 3.4.

With the modifications for shell kinematic assumption and the introduction of the artificial strain, the orthotropic material stiffness matrix for shell elements can be expressed as shown in Eq. (3.61).

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_a \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & 0 & 0 & 0 & 0 \\ & \frac{1}{E_2} & 0 & 0 & 0 & 0 \\ & & \frac{1}{G_{12}} & 0 & 0 & 0 \\ & & & \frac{1}{G_{12}} & 0 & 0 \\ & (sym) & & & \frac{1}{G_{13}} & 0 \\ & & & & & \frac{1}{G_{23}} \end{bmatrix}^{-1} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \gamma_a \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} \tag{3.61}$$

These modified stresses and strains can be implemented in the governing equation based on the principle of virtual work. A final note is that with the introduction of rotations as separate nodal degrees of freedom in addition to displacements, the governing equation takes an extra load term, representing the virtual work due to applied distributed moments on the structure, $\boldsymbol{m}$. The augmented form then becomes

$$\begin{aligned} \int_{\Omega} \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C]\boldsymbol{\epsilon}_{total} d\Omega - \int_{\Omega} [N]^T \boldsymbol{f} d\Omega - \int_{\Omega} [N]^T \boldsymbol{m} d\Omega \\ - \int_{S} [N]^T \boldsymbol{t} dS - \int_{\Omega} \Delta T \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C]\boldsymbol{\alpha}_{TE} d\Omega = 0 \end{aligned} \tag{3.62}$$

The Poisson equation for the temperature distribution in shells is the same as it is for solid elements, using the nodal basis functions of Table 3.3 and the definition of heat flux in Eq. (3.50). These, along with Eqs. (3.59), (3.60), (3.61) and Table 3.4 for elasticity can be implemented in the governing equations Eq. (3.44) for static thermoelastic analysis of shells.

3.2.3 Beam Elements

Beam elements, like shell elements are formulated for structures with certain characteristics and kinematic assumptions. They are intended to model long, slender members that are relatively thin in *both* transverse normal directions, compared to the length in the longitudinal direction. The most common classical beam theory is Bernoulli beam theory [79]. The basic kinematic assumptions for beam structures are similar to those for shells, in fact Kirchhoff plate theory is often thought of as an extension of Bernoulli beam theory. These basic assumptions are:

1. Any cross-sectional plane normal to the longitudinal direction of a beam will remain planar and normal to the longitudinal tangent direction after deformation.
2. Any normal cross-sectional plane will not warp or change shape within its plane after deformation.

Under these assumptions, the displacement field for small deformations in a coordinate system with the 1-direction aligned in the longitudinal direction can be expressed:

$$\begin{gathered} u_1 = u_{m,1} - x_3 \frac{\partial u_{m,3}}{\partial x_1} - x_2 \frac{\partial u_{m,2}}{\partial x_1} \\ u_2 = u_{m,2} \\ u_3 = u_{m,3} \end{gathered} \tag{3.63}$$

Replacing the displacement derivatives with transverse rotations,

$$\begin{gathered} u_1 = u_{m,1} + x_3\theta_2 - x_2\theta_3 \\ u_2 = u_{m,2} \\ u_3 = u_{m,3} \end{gathered} \tag{3.64}$$

As is the case with shell elements, beam elements define both displacements and rotations in all three dimensions as nodal degrees of freedom, with internal degrees of freedom to represent curvature in transverse directions. Unlike shell elements, however, the internal modes of curvature exist in both principal directions normal to the longitudinal axis of the beam instead of just one direction normal to a midplane. The assumed form for displacement and rotation at the longitudinal axis of the beam can be expressed as

$$\begin{gathered} \boldsymbol{u}_m = [U]\boldsymbol{N} \\ \boldsymbol{\theta} = [\theta]\boldsymbol{N} \end{gathered} \tag{3.65}$$

where

$$\begin{gathered} [U] = \begin{bmatrix} \boldsymbol{U}_{nd,1}^T & 0 \\ \boldsymbol{U}_{nd,2}^T & \boldsymbol{U}_{int,2}^T \\ \boldsymbol{U}_{nd,3}^T & \boldsymbol{U}_{int,3}^T \end{bmatrix} \\ [\theta] = \begin{bmatrix} \boldsymbol{\theta}_1^T & 0 \\ \boldsymbol{\theta}_2^T & 0 \\ \boldsymbol{\theta}_3^T & 0 \end{bmatrix} \\ \boldsymbol{N} = \begin{Bmatrix} \boldsymbol{N}_{nd} \\ \boldsymbol{N}_{int} \end{Bmatrix} \end{gathered} \tag{3.66}$$

Only one type of beam element is implemented in AStrO, a two-node version with schematic details given in Table 3.5. The same nodal degrees of freedom and method for modeling linear vs. nonlinear geometry developed for shells in Section 3.2.2 apply to beam elements as well. The critical difference is in the set of deformation modes used in the principle of virtual work. Recall that the virtual work due to elastic strain can be defined in terms of any set of deformation modes and their corresponding work conjugates. Beam elements use an alternate set, other than fundamental strains and stresses due to geometric considerations.

Most importantly, beam structures in the context of finite element modeling are typically defined not in terms of an exact geometric shape, but in terms of length and a set of cross-sectional properties, namely area A, second moment of area in each transverse direction I_2 and I_3, and polar moment of area about the longitudinal axis J. These parameters alone cannot define the exact geometry of a beam, and therefore high-fidelity resolution of strain and stress distribution is not possible. In this case there is little to be gained from trying to keep the formulation in terms of fundamental stress and strain.

Instead, the set of deformation modes used in the principle of virtual work are longitudinal strain ϵ_1, curvature about the transverse axes κ_2 and κ_3, transverse shear strains γ_{12} and γ_{13}, and longitudinal twist θ_1' as defined in Eq. (3.67). The transverse strains are included for the same reason as described for shells in the previous section.

$$\begin{Bmatrix} \epsilon_1 \\ \kappa_2 \\ \kappa_3 \\ \gamma_{12} \\ \gamma_{13} \\ \theta_1' \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u_1}{\partial x_1} \\ \dfrac{\partial \theta_2}{\partial x_1} \\ \dfrac{\partial \theta_3}{\partial x_1} \\ \dfrac{\partial u_2}{\partial x_1} - \theta_3 \\ \dfrac{\partial u_3}{\partial x_1} + \theta_2 \\ \dfrac{\partial \theta_1}{\partial x_1} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{U}_{I,nd,1} \cdot \dfrac{\partial \boldsymbol{N}_{nd}}{\partial x_1} \\ \boldsymbol{\theta}_{I,2} \cdot \dfrac{\partial \boldsymbol{N}_{nd}}{\partial x_1} \\ \boldsymbol{\theta}_{I,3} \cdot \dfrac{\partial \boldsymbol{N}_{nd}}{\partial x_1} \\ \boldsymbol{U}_{I,nd,2} \cdot \dfrac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{U}_{I,int,2} \cdot \dfrac{\partial \boldsymbol{N}_{int}}{\partial x_1} - \boldsymbol{\theta}_{I,3} \cdot \boldsymbol{N}_{nd} \\ \boldsymbol{U}_{I,nd,3} \cdot \dfrac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{U}_{I,int,3} \cdot \dfrac{\partial \boldsymbol{N}_{int}}{\partial x_1} + \boldsymbol{\theta}_{I,2} \cdot \boldsymbol{N}_{nd} \\ \boldsymbol{\theta}_{I,1} \cdot \dfrac{\partial \boldsymbol{N}_{nd}}{\partial x_1} \end{Bmatrix} \tag{3.67}$$

The definitions for instantaneous nodal degrees of freedom are the same as those given in Table 3.4. The integration of total virtual work for beams is performed in one dimension along the longitudinal direction, so the effective stiffness matrix is composed of the coefficients that form the corresponding conjugates of work per unit length, shown in Eq. (3.68). The work conjugates by name are longitudinal force F_1, bending moments about transverse axes M_2 and M_3, transverse shear forces F_{12} and F_{13}, and torsion about the longitudinal axis T_1.

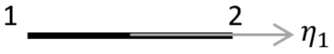
$$\begin{Bmatrix} F_1 \\ M_2 \\ M_3 \\ F_{12} \\ F_{13} \\ T_1 \end{Bmatrix} = \begin{bmatrix} E_1 A & & & & & \\ & E_1 I_2 & & & (0) & \\ & & E_1 I_3 & & & \\ & & & G_{12} A & & \\ & (0) & & & G_{13} A & \\ & & & & & G_{12} J \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \kappa_2 \\ \kappa_3 \\ \gamma_{12} \\ \gamma_{13} \\ \theta'_1 \end{Bmatrix} \tag{3.68}$$

The forces and moments in Eq. (3.68) can be put in place of stresses, and the strains and curvatures in place of strains in the principle of virtual work to form the governing equations of static elasticity. Again, the temperature distribution can be solved for the same way as before via the Poisson equation, except with fluxes and gradients only applicable in the 1-direction. These considerations lead to the modified variational governing equations for thermoelastic response:

$$
\begin{gathered}
\int_L (\boldsymbol{\sigma}\cdot\delta\boldsymbol{\epsilon})dx_1 + \int_L \xi(\dot{\boldsymbol{u}}_m\cdot\delta\boldsymbol{u}_m)dx_1 + \int_L \rho(\ddot{\boldsymbol{u}}_m\cdot\delta\boldsymbol{u}_m)dx_1 \\
-\int_L (\boldsymbol{f}\cdot\delta\boldsymbol{u}_m)dx_1 - \int_L (\boldsymbol{m}\cdot\delta\boldsymbol{\theta})dx_1 - (\boldsymbol{F}\cdot\delta\boldsymbol{u}_m)\Big|_0^L - (\boldsymbol{M}\cdot\delta\boldsymbol{\theta})\Big|_0^L = 0 \\
-\int_L q_1\delta\left(\frac{dT}{dx_1}\right)dx_1 + \int_L \rho C_p\dot{T}\delta T dx_1 - \int_L Q\delta T dx_1 + (q_1\delta T)\Big|_0^L = 0
\end{gathered}
\qquad (3.69)
$$

In Eq. (3.69), $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$ represent the force and strain/deformation vectors shown in Eq. (3.68), and ξ, ρ, $\boldsymbol{f}$, $\boldsymbol{m}$, and Q, are damping force, mass, applied force, applied moment, and heat generation per unit *length*, instead of per unit volume as with the solid/shell element formulations. In this one-dimensional case, integration by parts produces terms for concentrated force, moment and flux on the ends of a beam structure, analogous to the surface flux and tractions in previous cases. Using the assumed forms given in Eqs. (3.65) and (3.66), along with the basis function definitions given in Table 3.5, the discretized equations for beam elements can be derived as was done previously, with Eq. (3.69) as the starting point.

Table 3.5 Geometry and basis definition for beam element.

Element Type	**Basis Functions**	**Integration Points** $[\boldsymbol{\eta_1}, \boldsymbol{w}]$
two-node beam  $-1 \le \eta_1 \le 1$	*Nodal:* $N_1 = \frac{1}{2}(1-\eta_1)$ $N_2 = \frac{1}{2}(1+\eta_1)$ *Internal,* $u_{m,2}, u_{m,3}$: $N_3 = (1+\eta_1)(1-\eta_1)$	p_1: $\left[\frac{-1}{\sqrt{3}}, 1\right]$ p_2: $\left[\frac{1}{\sqrt{3}}, 1\right]$

3.3 Dynamic Analysis

In dynamic thermoelastic analysis, all terms of the governing equations, Eq. (3.30) and Eq. (3.36) participate, including the time-dependent terms. The solution is spatially discretized using the same element basis functions as for static analysis, but the nodal degrees of freedom become time-dependent. The time-derivatives of the fundamental field variables of displacement and temperature can then be expressed

$$\begin{aligned} \dot{\boldsymbol{u}} &= [N](\boldsymbol{x})\dot{\boldsymbol{U}}(t) \\ \ddot{\boldsymbol{u}} &= [N](\boldsymbol{x})\ddot{\boldsymbol{U}}(t) \\ \dot{T} &= \boldsymbol{N}(\boldsymbol{x}) \cdot \dot{\boldsymbol{\phi}}(t) \end{aligned} \tag{3.70}$$

AStrO is intended for general-purpose implicit time integration, so the time-evolution of variables is found by marching through a series of time steps, separated by a finite time interval Δt, and solving for the state at each step based on the instantaneous loading and the state at the previous step. The state of variables at a given time step relates to the state at the previous time step through Taylor-series-based expansions. Specifically, AStrO uses the Newmark Beta expansion [83],

$$\begin{aligned} \boldsymbol{U}^{n+1} &= \boldsymbol{U}^n + \Delta t \dot{\boldsymbol{U}}^n + \frac{1}{2}\Delta t^2 \left((1-2\beta)\ddot{\boldsymbol{U}}^n + 2\beta\ddot{\boldsymbol{U}}^{n+1}\right) \quad 0 < \beta \leq \frac{1}{2} \\ \dot{\boldsymbol{U}}^{n+1} &= \dot{\boldsymbol{U}}^n + \Delta t\left((1-\gamma)\ddot{\boldsymbol{U}}^n + \gamma\ddot{\boldsymbol{U}}^{n+1}\right) \quad 0 < \gamma \leq 1 \\ \boldsymbol{\phi}^{n+1} &= \boldsymbol{\phi}^n + \Delta t\left((1-\gamma)\dot{\boldsymbol{\phi}}^n + \gamma\dot{\boldsymbol{\phi}}^{n+1}\right) \quad 0 < \gamma \leq 1 \end{aligned} \tag{3.71}$$

Equation (3.71) is a scheme for implicit time integration, and is well-established for structural problems. The superscript n denotes the time step of each vector. The parameters β and γ can be user-defined within the ranges shown, and are typically between $\frac{1}{4} \leq \beta \leq \frac{1}{2}$, and $\frac{1}{2} \leq \gamma \leq 1$. With the parameters in the high range the scheme is more stable, but less accurate, while in the mid-range the scheme is most accurate but less stable (first-order accurate at $\beta = \frac{1}{2}, \gamma = 1$, second-order accurate at $\beta = \frac{1}{4}, \gamma = \frac{1}{2}$), [76, 83].

Now adopt the following shorthand notation for the terms in the full dynamic governing equations:

$$
\begin{gathered}
\boldsymbol{R}_{u,k}^{n} = \int_{\Omega} \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^{T} [C]\boldsymbol{\epsilon} d\Omega \ (\boldsymbol{U} = \boldsymbol{U}^{n}) \\
\boldsymbol{R}_{u,c}^{n} = \int_{\Omega} \xi [N]^{T} \dot{\boldsymbol{u}}^{n} d\Omega = \int_{\Omega} \xi [N]^{T} [N] \dot{\boldsymbol{U}}^{n} d\Omega \ = [C_{u}] \dot{\boldsymbol{U}}^{n} \\
\boldsymbol{R}_{u,m}^{n} = \int_{\Omega} \rho [N]^{T} \ddot{\boldsymbol{u}}^{n} d\Omega = \int_{\Omega} \rho [N]^{T} [N] \ddot{\boldsymbol{U}}^{n} d\Omega \ = [M_{u}] \ddot{\boldsymbol{U}}^{n} \\
\boldsymbol{R}_{u,app}^{n} = -\int_{\Omega} [N]^{T} \boldsymbol{f}^{n} d\Omega - \int_{S} [N]^{T} \boldsymbol{t}^{n} dS \\
\boldsymbol{R}_{u,th}^{n} = -\int_{\Omega} \Delta T \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^{T} [C] \boldsymbol{\alpha}_{TE} d\Omega \ (\boldsymbol{U} = \boldsymbol{U}^{n}) \\
\boldsymbol{R}_{\phi,k}^{n} = -\int_{\Omega} [\nabla N] \boldsymbol{q}^{n} d\Omega = [K_{\phi}] \boldsymbol{\phi}^{n} \\
\boldsymbol{R}_{\phi,m}^{n} = \int_{\Omega} \boldsymbol{N} \rho C_{p} \dot{T}^{n} d\Omega = \int_{\Omega} \rho C_{p} \boldsymbol{N} \boldsymbol{N}^{T} \dot{\boldsymbol{\phi}}^{n} d\Omega = [M_{\phi}] \dot{\boldsymbol{\phi}}^{n} \\
\boldsymbol{R}_{\phi,hg}^{n} = -\int_{\Omega} \boldsymbol{N} Q^{n} d\Omega + \int_{S} \boldsymbol{N} (\boldsymbol{q}^{n} \cdot \boldsymbol{n}) dS
\end{gathered}
\tag{3.72}
$$

Using this notation, the governing equations for the solution at time step $n+1$ can be expressed

$$
\begin{gathered}
\boldsymbol{R}_{u}^{n+1} = \boldsymbol{R}_{u,k}^{n+1} + \boldsymbol{R}_{u,c}^{n+1} + \boldsymbol{R}_{u,m}^{n+1} + \boldsymbol{R}_{u,app}^{n+1} + \boldsymbol{R}_{u,th}^{n+1} = 0 \\
\boldsymbol{R}_{\phi}^{n+1} = \boldsymbol{R}_{\phi,k}^{n+1} + \boldsymbol{R}_{\phi,m}^{n+1} + \boldsymbol{R}_{\phi,hg}^{n+1} = 0
\end{gathered}
\tag{3.73}
$$

One further modification is made to the equations of elasticity, known as the Hilber-Hughes-Taylor alpha method [76]. In this method, an additional scalar parameter α is introduced, set in the range $-1 < \alpha \leq 0$, and the governing equation for the $n+1$ time step is modified to

$$
\boldsymbol{R}_{u}^{n+1} = (1+\alpha)\left(\boldsymbol{R}_{u,k}^{n+1} + \boldsymbol{R}_{u,c}^{n+1} + \boldsymbol{R}_{u,app}^{n+1}\right) - \alpha\left(\boldsymbol{R}_{u,k}^{n} + \boldsymbol{R}_{u,c}^{n} + \boldsymbol{R}_{u,app}^{n}\right) + \boldsymbol{R}_{u,m}^{n+1} + \boldsymbol{R}_{u,th}^{n+1} = 0 \tag{3.74}
$$

If α is set to a value of zero, Eq, (3.74) becomes equivalent to the original Newmark Beta scheme. By adjusting the parameters β, γ, and α, within the correct range, the scheme can be made both second-order accurate and unconditionally stable. This makes it a robust and versatile method.

Using the Newmark Beta expansions of Eq. (3.71), the governing equations can be expressed entirely in terms of displacement and temperature at the new time step, $\boldsymbol{U}^{n+1}$ and $\boldsymbol{\phi}^{n+1}$, and all variables at the previous time step. The initial state of the system at time $t = 0$ must be known, and the state at every subsequent time step can be found with the following process:

```
$for\ n = 0\ to\ (number\ of\ time\ steps)$:
    (1) $solve\ \boldsymbol{R}_{\phi}^{n+1}(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^{n}, \dot{\boldsymbol{\phi}}^{n}) = 0\ \ for\ \boldsymbol{\phi}^{n+1}$
    (2) $solve\ \boldsymbol{R}_{u}^{n+1}(\boldsymbol{U}^{n+1}, \boldsymbol{U}^{n}, \dot{\boldsymbol{U}}^{n}, \ddot{\boldsymbol{U}}^{n}, \boldsymbol{\phi}^{n+1}) = 0\ \ for\ \boldsymbol{U}^{n+1}$        (3.75)
    (3) $update\ \dot{\boldsymbol{\phi}}^{n+1}, \dot{\boldsymbol{U}}^{n+1}\ and\ \ddot{\boldsymbol{U}}^{n+1}\ using\ Eq.$ (3.71)
$end\ for$
```

This overall approach is valid regardless of element type or the presence of nonlinear geometry. The difference between those cases lie mainly in the construction of the static terms as described in Section 3.2, and the details of the solution process on each time step described in Section 3.4.

3.4 Solution of Finite Element Equations

With the theoretical formulation of the finite element equations for thermoelastic modeling established, the process of obtaining the solutions can now be examined. A number of considerations exist regarding how to approach the solution process, each with several possible ways of addressing them. Certain methods chosen for implementation in AStrO, and the reasons for choosing them will now be described.

3.4.1 Overall Approach

In any finite element analysis, the task of obtaining the solution for fundamental field variables consists of determining the discrete set of degrees of freedom that define the solution. Since the degrees of freedom are invariably interdependent, this involves simultaneously solving the coupled discretized governing equations defining the interaction degrees of freedom. These equations may be linear or nonlinear, and they may apply to a single static state or to each of a series of time steps in a dynamic analysis. However, in all cases the core process to obtain solutions can be summed up succinctly.

For analyses assuming linear geometry (and all other forms of linearity), the solution for static equilibrium or for the state at a given time step can be found by solving the following linear system:

$$\begin{bmatrix} \left[\dfrac{\partial \boldsymbol{R}_\phi}{\partial \boldsymbol{\phi}}\right] & \left[\dfrac{\partial \boldsymbol{R}_\phi}{\partial \boldsymbol{U}}\right] \\ \left[\dfrac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{\phi}}\right] & \left[\dfrac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{U}}\right] \end{bmatrix} \begin{Bmatrix} \boldsymbol{\phi} \\ \boldsymbol{U} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{R}_\phi(\boldsymbol{\phi} = 0, \boldsymbol{U} = 0) \\ -\boldsymbol{R}_u(\boldsymbol{\phi} = 0, \boldsymbol{U} = 0) \end{Bmatrix} \tag{3.76}$$

For cases with nonlinearity, the system can be solved with the Newton/Newton-Raphson method, in which the following iterative process is performed:

```
(1) initialize $\boldsymbol{\phi} = \boldsymbol{\phi}_0, \boldsymbol{U} = \boldsymbol{U}_0$
(2) while solution convergence criteria is not satisfied:
    (a) solve $\begin{bmatrix} \left[\dfrac{\partial \boldsymbol{R}_\phi}{\partial \boldsymbol{\phi}}\right] & \left[\dfrac{\partial \boldsymbol{R}_\phi}{\partial \boldsymbol{U}}\right] \\ \left[\dfrac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{\phi}}\right] & \left[\dfrac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{U}}\right] \end{bmatrix} \begin{Bmatrix} \Delta\boldsymbol{\phi} \\ \Delta\boldsymbol{U} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{R}_\phi(\boldsymbol{\phi}, \boldsymbol{U}) \\ -\boldsymbol{R}_u(\boldsymbol{\phi}, \boldsymbol{U}) \end{Bmatrix}$    (3.77)
    (b) update $\boldsymbol{\phi} = \boldsymbol{\phi} + \Delta\boldsymbol{\phi}$
               $\boldsymbol{U} = \boldsymbol{U} + \Delta\boldsymbol{U}$
    (c) check solution convergence criteria
```

Linear analysis is in fact a special case of the iterative Newton-Raphson process, in which the solution is initialized at zero, and only a single iteration of the process is required to bring the residual of the equations to zero and obtain the final solution. As described in Section 3.1.3, AStrO assumes a one-way dependence of displacement on temperature. Thermal material properties are also assumed constant with respect to temperature. These assumptions simplify the solution process in two ways. First, the term $\left[\frac{\partial \boldsymbol{R}_\phi}{\partial \boldsymbol{U}}\right]$ is zero, meaning the temperature solution alone can be found first, followed by the displacement solution from the temperature result. Second, the equations for thermal distributions are always linear, whether or not linear geometry is assumed in the elastic response. Therefore, the temperature can be solved from the linear system

$$\left[\frac{\partial \boldsymbol{R}_{\phi}}{\partial \boldsymbol{\phi}}\right] \boldsymbol{\phi} = -\boldsymbol{R}_{\phi}(\boldsymbol{\phi} = 0) \tag{3.78}$$

Then, using the resulting distribution of ΔT in $\boldsymbol{R}_{u,th}$, the displacement can be found either from linear solution or Newton-Raphson iterations on $\boldsymbol{R}_u = 0$:

$$\begin{aligned} &Linear\text{:} \quad \left[\frac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{U}}\right] \boldsymbol{U} = -\boldsymbol{R}_u(\boldsymbol{U} = 0) \\ &Nonlinear\text{:} \begin{array}{ll} (1) & \left[\frac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{U}}\right] \Delta\boldsymbol{U} = -\boldsymbol{R}_u(\boldsymbol{U}) \\ (2) & \boldsymbol{U} = \boldsymbol{U} + \Delta\boldsymbol{U} \end{array} \end{aligned} \tag{3.79}$$

When performing dynamic analysis, the time-derivatives of the variables need to be updated after each solution or incrementation of $\boldsymbol{U}$ and $\boldsymbol{\phi}$, according to Eq. (3.71). Also, when forming the Jacobian matrix $\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]$, all terms must account for the dependence of the time-derivatives at the new time step on the fundamental variable at the new time step. With that in mind, the following differentiation of terms holds, given the Newmark Beta expansions:

$$\begin{gathered} \left[\frac{\partial \boldsymbol{R}_{u,k}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] = \int_{\Omega} \left(\left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}^{n+1}}\right]^T [C] \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}^{n+1}}\right] + \sum_{i=1}^{6} \sigma_i \left[\frac{\partial}{\partial \boldsymbol{U}^{n+1}} \left(\frac{\partial \epsilon_i}{\partial \boldsymbol{U}^{n+1}}\right)\right] \right) d\Omega = [K_u]\,(\boldsymbol{U}^{n+1}) \\ \left[\frac{\partial \boldsymbol{R}_{u,c}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] = \left[\frac{\partial \boldsymbol{R}_{u,c}^{n+1}}{\partial \dot{\boldsymbol{U}}^{n+1}}\right] \left[\frac{\partial \dot{\boldsymbol{U}}^{n+1}}{\partial \ddot{\boldsymbol{U}}^{n+1}}\right] \left[\frac{\partial \ddot{\boldsymbol{U}}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] = [C_u](\gamma \Delta t[I]) \left(\frac{1}{\beta \Delta t^2}[I]\right) = \frac{\gamma}{\beta \Delta t}[C_u] \\ \left[\frac{\partial \boldsymbol{R}_{u,m}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] = \left[\frac{\partial \boldsymbol{R}_{u,m}^{n+1}}{\partial \ddot{\boldsymbol{U}}^{n+1}}\right] \left[\frac{\partial \ddot{\boldsymbol{U}}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] + [M_u] \left(\frac{1}{\beta \Delta t^2}\ [I]\right) = \frac{1}{\beta \Delta t^2}\ [M_u] \\ \left[\frac{\partial \boldsymbol{R}_{\phi,k}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] = -\int_{\Omega} [\nabla N] \left[\frac{\partial \boldsymbol{q}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] d\Omega = [K_{\phi}] \\ \left[\frac{\partial \boldsymbol{R}_{\phi,m}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] = \left[\frac{\partial \boldsymbol{R}_{\phi,m}^{n+1}}{\partial \dot{\boldsymbol{\phi}}^{n+1}}\right] \left[\frac{\partial \dot{\boldsymbol{\phi}}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] = [M_{\phi}] \left(\frac{1}{\gamma \Delta t}[I]\right) = \frac{1}{\gamma \Delta t}[M_{\phi}] \end{gathered} \tag{3.80}$$

By differentiating Eqs. (3.73) and (3.74) using the appropriate substitutions from Eq. (3.80), the global equation matrices for a dynamic time step can be written:

$$\begin{gathered}\left[\frac{\partial \boldsymbol{R}_u^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] = (1+\alpha)\left([K_u] + \frac{\gamma}{\beta \Delta t}[C_u]\right) + \frac{1}{\beta \Delta t^2}[M_u] \\ \left[\frac{\partial \boldsymbol{R}_\phi^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] = [K_\phi] + \frac{1}{\gamma \Delta t}[M_\phi]\end{gathered} \tag{3.81}$$

As a final note, the damping properties of a structure are often not obtainable by any analytical means, and it is typical to represent the damping matrix as a linear combination of the mass matrix and the stiffness matrix:

$$[C_u] = a_{RD}[K_u] + b_{RD}[M_u] \tag{3.82}$$

The coefficients a_{RD} and b_{RD} are estimated or determined empirically from experiment. This technique is known as *Rayleigh damping* [84], and AStrO contains optional parameters that can be used to define the damping coefficients.

3.4.2 Boundary Conditions and Multi-Point Constraints

Most of the time, finite element structures have certain boundary conditions and constraints placed on the degrees of freedom of the solution. In fact, in static analysis some constraints must be present in order for a unique solution to exist, and the governing equations will be unsolvable without them. There are several ways to modify the governing equations to enforce boundary conditions and constraints, and the method for AStrO was chosen based on certain desired criteria.

Using the formulations implemented and described in the previous sections, the governing equation matrices for both the thermal and elastic equations are symmetric. This is a convenient and desirable property from the point of view of solving linear systems, and applying constraints in a way that preserves matrix symmetry is preferred. It is also advantageous to use a method that does not increase the dimension of the equation matrix or produce problematic characteristics like blocks of zeros on the diagonal. Furthermore, since the ultimate goal is to perform sensitivity analysis, a method of applying constraints that allows straightforward differentiation of the governing equations is desirable.

Considering these factors, the method chosen for applying constraints is known as the penalty method. If a vector $\boldsymbol{U}$ is governed by the linear system:

$$[K]\boldsymbol{U} = \boldsymbol{F} \tag{3.83}$$

where $[K]$ is a symmetric matrix, then $\boldsymbol{U}$ corresponds to the minimum of a scalar function given by

$$\psi = \frac{1}{2}\boldsymbol{U}^T[K]\boldsymbol{U} - \boldsymbol{U}^T\boldsymbol{F} \tag{3.84}$$

If it is then desired to apply a set of linear constraints to $\boldsymbol{U}$ of the form

$$\boldsymbol{r}_{bc} = [C_{bc}]\boldsymbol{U} - \boldsymbol{q}_{bc} = 0 \tag{3.85}$$

for a constant matrix $[C_{bc}]$ and a constant vector $\boldsymbol{q}_{bc}$, then one approach to satisfying both the linear system and the constraints is to simultaneously minimize the scalar function ψ and the 2-norm of the constraint residual $\boldsymbol{r}_{bc}$ as a single scalar function given by

$$\psi_{bc} = \frac{1}{2}\boldsymbol{U}^T[K]\boldsymbol{U} - \boldsymbol{U}^T\boldsymbol{F} + \frac{1}{2}\mu(\boldsymbol{r}_{bc}^T\boldsymbol{r}_{bc}) \tag{3.86}$$

where μ is a scalar parameter set by the user. Setting the gradient of the scalar function ψ_{bc} to zero gives a new composite linear system:

$$\begin{gathered}\frac{\partial\psi_{bc}}{\partial\boldsymbol{U}} = [K]\boldsymbol{U} - \boldsymbol{F} + \mu\left[\frac{\partial\boldsymbol{r}_{bc}}{\partial\boldsymbol{U}}\right]^T\boldsymbol{r}_{bc} = [K]\boldsymbol{U} - \boldsymbol{F} + \mu[C_{bc}]^T([C_{bc}]\boldsymbol{U} - \boldsymbol{q}_{bc}) = 0 \\ \Longrightarrow \big[[K] + \mu[C_{bc}]^T[C_{bc}]\big]\boldsymbol{U} = \boldsymbol{F} + \mu[C_{bc}]^T\boldsymbol{q}_{bc}\end{gathered} \tag{3.87}$$

The composite system after applying constraints remains symmetric, and the modification involves simply adding the constraint penalty term to the original residual of the equations. The governing equations for the temperature and displacement solutions are each augmented with a penalty term of this form to satisfy constraints and boundary conditions, so that the new residual vectors become

$$\begin{aligned}\boldsymbol{R}_u \rightarrow \boldsymbol{R}_u + \mu\left[C_{u,bc}\right]^T\left(\left[C_{u,bc}\right]\boldsymbol{U} - \boldsymbol{q}_{u,bc}\right) = \boldsymbol{R}_u + \boldsymbol{R}_{u,bc} \\ \boldsymbol{R}_\phi \rightarrow \boldsymbol{R}_\phi + \mu\left[C_{\phi,bc}\right]^T\left(\left[C_{\phi,bc}\right]\boldsymbol{U} - \boldsymbol{q}_{\phi,bc}\right) = \boldsymbol{R}_\phi + \boldsymbol{R}_{\phi,bc}\end{aligned} \tag{3.88}$$

The penalty method has two significant drawbacks despite its advantages. It does not satisfy the constraint equations exactly and it results in an ill-conditioned system. The minimization of ψ_{bc} could be thought of as a compromise between satisfaction of the original system and satisfaction of the constraints. The parameter μ represents the priority placed on minimization of the constraint residual, so it must be set sufficiently high to enforce the constraints strongly. But setting it too high gives the system a high condition number, possibly leading to errors or poor solution convergence.

Nevertheless, both of these drawbacks can generally be compensated for easily. Although the constraints are technically not exactly satisfied, with a value of μ around 10^4 to 10^5 times the average value of the matrix $[K]$, they are generally satisfied to a degree more than adequate for the majority of applications. If, however, it is required to satisfy constraints to a further degree of precision, it is possible to refine accuracy with a corrective iteration in which the applied load is updated to simulate the reaction forces on degrees of freedom due to constraints. This can be done inexpensively with most common linear solution methods. The degradation of the condition number can also generally be compensated for with proper pre-conditioning, if it creates a problem in the solution process. For these reasons, the benefits the penalty constraint method were decided to be worth the drawbacks for implementation in AStrO.

3.4.3 Condensation of Internal Degrees of Freedom

As explained in Section 3.2, several element types implemented in AStrO have internal degrees of freedom, which are not node-associated and do not interact directly with other elements. These include hexahedral solid elements with incompatible bending modes, as well as shell and beam elements with internal displacement modes representing out-of-plane and off-axis curvatures. Part of the power of these internal modes is that they can be condensed out of the global equation matrices, so they add compliance and richness of solution space without adding to the effective global matrix dimension. The details of this process will now be explained.

The stiffness matrix for a given element in a structure can be constructed as

$$\left[K_{u,el}\right] = \left[\frac{\partial \boldsymbol{R}_{u,el}}{\partial \boldsymbol{U}_{el}}\right] \tag{3.89}$$

with $\boldsymbol{R}_{u,el}$ as defined in Eq. (3.72), integrated over the domain of that element. The displacement degrees of freedom of that element are governed by the linear system:

$$\left[K_{u,el}\right]\boldsymbol{U}_{el} = -\boldsymbol{R}_{u,el} \tag{3.90}$$

If the displacement vector contains internal degrees of freedom, they can be segregated from the nodal degrees of freedom and the matrix system can be partitioned as follows:

$$\begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{21}] & [K_{22}] \end{bmatrix} \begin{Bmatrix} \boldsymbol{U}_{nodal} \\ \boldsymbol{U}_{int} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{R}_{nodal} \\ 0 \end{Bmatrix} \tag{3.91}$$

The load vector $-\boldsymbol{R}_{nodal}$ represents all the loads applied to the nodal degrees of freedom of the element, which includes forces due to elastic stress in neighboring elements, and cannot be known without resolving the global response for the entire structure. The internal degrees of freedom, however, do not have any directly applied loads, and their only interactions are with the degrees of freedom of the present element. This means the solution for the internal degrees of freedom can be completely represented from the second row block of Eq. (3.91) as

$$\boldsymbol{U}_{int} = -[K_{22}]^{-1}[K_{21}]\boldsymbol{U}_{nodal} \tag{3.92}$$

Substituting the above into the first row block of the matrix gives

$$\left[[K_{11}] - [K_{12}][K_{22}]^{-1}[K_{21}]\right]\boldsymbol{U}_{nodal} = -\boldsymbol{R}_{nodal} \tag{3.93}$$

The matrix multiplying $\boldsymbol{U}_{nodal}$ in Eq. (3.93) is a condensed matrix with the effect of the internal degrees of freedom absorbed within. The condensed matrices can be assembled for each element, to give a global system containing only the nodal degrees of freedom. After the nodal degrees of freedom are obtained from solution of the global system, the internal degrees of freedom can then be evaluated from Eq. (3.92) if need be. For geometrically linear analysis this is usually not necessary, but with nonlinear geometry and sensitivity analysis the internal degrees of freedom factor into the iterative re-evaluation of the stiffness matrix. AStrO stores the internal degrees of freedom and the internal range of each element matrix in separate data structures for this purpose.

3.4.4 Methods of Solution to Linear Systems

A finite element result is obtained through the solution of the discretized governing equations. Whether the equations are linear or nonlinear, the critical operation is the solution of linear systems. There are many ways to solve linear systems, each with strengths and weaknesses, and it is prudent to have some selection for different situations. There are two main methods for solution of linear systems built into AStrO: LDL^T factorization and conjugate gradient.

The default solver method is LDL^T factorization, a special case of LU factorization for symmetric matrices. The process involves constructing the matrices $[L]$ and $[D]$, such that

$$[L][D][L]^T = [K] \tag{3.94}$$

where $[L]$ is a lower-triangular matrix with all diagonal terms having a value of one, $[D]$ is a matrix of all zeros except on the diagonal terms, and $[K]$ is the original equation matrix in the linear system $[K]\boldsymbol{U} = \boldsymbol{F}$. The general algorithm for obtaining the matrices $[L]$ and $[D]$ can be described as follows:

```
Given: a symmetric matrix $[K]$ of dimension $n$
        for $i = 1$ to $n$:
            $D_{ii} = K_{ii} - \sum_{k=(minCol)}^{i-1} L_{ik}^{2} D_{kk}$
            for $j = (i+1)$ to $(maxRow)$:                                  (3.95)
                $L_{ji} = \frac{K_{ji} - \sum_{k=(minCol)}^{i-1} L_{ik} D_{kk} L_{jk}}{D_{ii}}$
            end for
        end for
```

The factorization is performed in place, so that the diagonal terms of $[D]$ and non-diagonal terms of $[L]$ overwrite the corresponding original values in $[K]$. The matrix $[K]$ is stored row by row, in the range from the first nonzero column in the row ($minCol$) to the diagonal column. This format is also known as *skyline* storage, and takes advantage of the sparseness of a matrix when the rows/columns are ordered so as to minimize the bandwidth of nonzero terms about the diagonal band. The function is built into AStrO, without accessing an external library. Following factorization, the solution to a linear system can be found using forward and backward substitution. This method is used as opposed to Cholesky factorization because it is valid for indefinite matrices as well as positive definite, while still taking advantage of matrix symmetry. This property also proved very useful for a later study (see Chapter 6).

LDL^T is the default solver method because it is reliable and robust, and its effectiveness is relatively insensitive to ill-conditioning. This is desirable not only because of the use of the penalty constraint method, as explained in Section 3.4.2, but also because many structures produce inherently ill-conditioned equation matrices due to geometry and boundary conditions. Another advantage is that matrix factorization allows for the quick solution for multiple different linear systems with the same matrix but different right-hand-side vectors $\boldsymbol{F}$, since the factorization step need only be performed once. This can be very useful for dynamic analysis assuming linear geometry. For these reasons, it is common in the structural analysis community to tend toward direct solver methods such as this.

The second solver method, conjugate gradient, is an iterative Krylov method designed for the minimization of the scalar function ψ, previously shown in Eq. (3.84), which is equivalent to the solution for symmetric linear systems. The conjugate gradient algorithm possesses a unique property among iterative methods in that it is, in mathematical theory, guaranteed to converge in a number of iterations no greater than the dimension of the matrix. It does so by storing only one search direction at a time, and using a recursive formula for each new direction to effectively increase the search-space dimension on every iteration. This is in contrast to the similar GMRES method, which is applicable to any matrix, symmetric or not, but must store a series of search directions and usually has no guaranteed maximum number of iterations to convergence. The conjugate gradient method is a powerful tool that lends itself well to solutions of symmetric systems such as those typically seen in structural finite element analysis.

One consideration regarding the conjugate gradient method is that its convergence rate is sensitive to the condition number of the matrix, and performance is degraded for ill-conditioned systems. In extreme cases, it can fail to converge altogether due to numerical issues. For structural problems, this means that adequate preconditioning is essential for effective application of the method.

Preconditioning a system involves applying some matrix or operator to the original linear system to produce an improved condition number. AStrO uses what is known as left-preconditioning, where the alternate system is constructed of the form

$$[P]^{-1}[K]\boldsymbol{U} = [P]^{-1}\boldsymbol{F} \tag{3.96}$$

where $[P]$ is a symmetric matrix intended to approximate the matrix $[K]$. Three basic options for defining the preconditioning matrix are available in AStrO. The first is partial LDL^T factorization, where $[P]$ is defined as $[K]$ in the region within a certain bandwidth around the diagonal, and zero outside that bandwidth. The inverse operator is applied by performing full factorization on $[P]$, and using forward and back substitution on a given vector. The second option is incomplete LDL^T factorization, where $[P]$ is defined as $[K]$ but the factorization is only performed on the nonzero entries, and zeros are assumed to remain zero. The third option is block-Jacobi factorization, where $[P]$ is taken to be $[K]$ within square block regions of a certain dimension around the diagonal, and zero outside of the block regions. Full factorization is performed on $[P]$ to apply the inverse operator.

Most analyses can be performed with either of the two main solver methods, and the choice comes down to the user's judgement. Although LDL^T factorization is the default, there are certain conditions in which conjugate gradient performs particularly well, and does not have all the memory requirements of a full matrix factorization. Currently, AStrO does not have a parallel implementation, but if that advancement were to be made it would be an additional consideration into the most appropriate solver method.

Regardless of the solver method used, the global equation matrices of finite element analysis are generally very sparse, and it is beneficial to arrange the degrees of freedom in a way to make the solution process as efficient as possible. As part of the input processing phase, AStrO re-orders the nodes of the finite element mesh according to their interactive connectivity in the equations and constraints. Exactly how this is done depends on the solver options chosen. If the default full LDL^T or partial factorization with the conjugate gradient solver is used, it attempts to minimize the overall bandwidth of the global matrix around the diagonal by doing breadth-first level sort. If the conjugate gradient solver with block-Jacobi preconditioning is chosen, it attempts to form blocks around the diagonals of the appropriate dimension with the highest density of nonzero entries as possible, to maximize the effectiveness of the preconditioner. Much more discussion could be given about different alternatives and options for linear solvers, but it is a topic outside the focus of this thesis. All analysis performed and documented in the present work employ the methods and options discussed in this section.

Chapter 4

Sensitivity Analysis and the Adjoint Method for Thermoelastic Modeling

For sensitivity analysis and optimization involving high-fidelity modeling, it is essential to use a method that minimizes the number of simulation runs required. Methods like genetic algorithms and machine learning are powerful tools for global optimization, and applications with discrete design parameters or stochastic, non-deterministic behavior. But typically, these methods require numerous evaluations of the objective function, sometimes multiple thousands or even millions. In high-fidelity computing applications that may take hundreds of core hours per simulation, this is generally not feasible, and alternative approaches must be used.

Gradient-based optimization is an efficient means of finding local extrema of objectives that are smooth, differentiable functions of a set of continuous variables. For such problems, the gradient, or the sensitivity of the objective with respect to each variable, can serve as a guide for seeking the local extrema. The concept is illustrated graphically in Fig. 4.1.

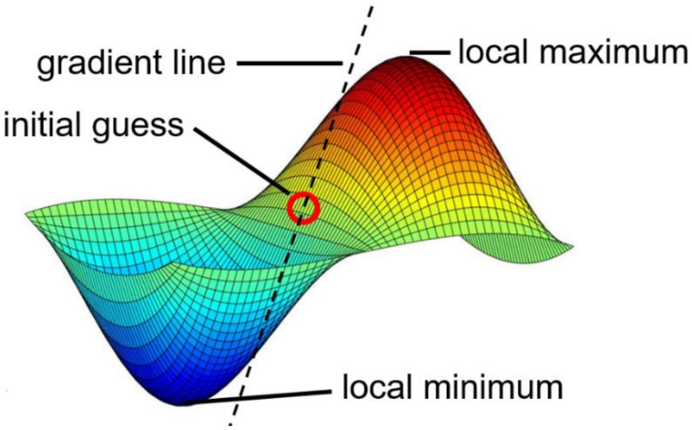


Figure 4.1 Conceptual depiction of the gradient as a guiding direction to local extrema.

A gradient-based optimization algorithm begins with an initial guess for the function variables, and goes through an iterative process in which each cycle the variables are updated in an attempt to bring the objective closer to the desired extremum. The direction and step length by which to increment the variables at each cycle are determined by some combined consideration of the objective gradient (and sometimes higher-order derivatives), and any applicable constraints. Many variations of gradient methods exist, and efficient algorithms can often seek out local extrema in only a few iterations, though this is very case-dependent.

Even with a good gradient-based optimizer, however, if computational savings is the goal, an equally important consideration is the method of obtaining the objective sensitivities. Many methods, such as finite difference, require function evaluations for every variable at every optimization cycle. For a large number of variables this can become prohibitively costly. The adjoint method provides a way to obtain objective sensitivities at a cost that is independent of the number of design variables, and is very efficient for large, high-fidelity gradient-based optimization problems. This chapter outlines the overall approach of the adjoint method, as well as its application to thermoelastic structural modeling.

4.1 General Formulation for the Discrete Adjoint Method

In this section the general approach to applying the discrete adjoint method in an optimization problem is presented. The adjoint method can be applied to any gradient-based optimization problem involving an objective function that is determined by the behavior of a physical or mathematical system, and ultimately by the solution to a deterministic set of governing equations. The adjoint can be applied to an equation in analytical form, termed the *continuous adjoint*, or to a set of discretized equations derived from the analytical form, the *discrete adjoint*. The present work focuses exclusively on the discrete adjoint, and the remainder of this chapter is dedicated to the application of the adjoint to the discretized governing equations developed for thermoelastic modeling in Chapter 3.

In gradient-based design optimization of dynamic systems, there exists some objective quantity L to be optimized (either minimized or maximized) with respect to a set of design variables $\boldsymbol{D}$, which define some set of physical characteristics or specifications of the system. The objective function can be posed as dependent on the design variables themselves, as well as on the system response, defined by the degrees of freedom of the discretized solution variables $\boldsymbol{U}$, which in turn depends on the design variables. This definition is expressed in Eq. (4.1).

$$L = L\big(\boldsymbol{D}, \boldsymbol{U}(\boldsymbol{D})\big) \tag{4.1}$$

From differentiation of Eq. (4.1), the total sensitivity of L with respect to a given design variable, D_i is

$$\frac{dL}{dD_i} = \frac{\partial L}{\partial D_i} + \left(\frac{\partial L}{\partial \boldsymbol{U}}\right) \cdot \left(\frac{\partial \boldsymbol{U}}{\partial D_i}\right) \tag{4.2}$$

Generally, the terms $\frac{\partial L}{\partial \boldsymbol{D}}$ and $\frac{\partial L}{\partial \boldsymbol{U}}$ are relatively inexpensive to calculate directly from the definition of the objective function. The remaining term $\frac{\partial \boldsymbol{U}}{\partial D_i}$, however, tends to be more involved and challenging to evaluate. The first step is understanding that the solution variables $\boldsymbol{U}$ must always satisfy the appropriate set of governing equations, with the equation residual vector denoted $\boldsymbol{R}$, which is also dependent on the design variables.

$$\boldsymbol{R}\big(\boldsymbol{D}, \boldsymbol{U}(\boldsymbol{D})\big) = 0 \tag{4.3}$$

Differentiating Eq. (4.3) gives an expression for the sensitivity of $\boldsymbol{U}$ with respect to a given design variable D_i,

$$\frac{d\boldsymbol{R}}{dD_i} = \frac{\partial \boldsymbol{R}}{\partial D_i} + \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right] \frac{\partial \boldsymbol{U}}{\partial D_i} = 0 \Rightarrow \frac{\partial \boldsymbol{U}}{\partial D_i} = -\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{-1} \frac{\partial \boldsymbol{R}}{\partial D_i} \tag{4.4}$$

Substituting the expression for $\frac{\partial \boldsymbol{U}}{\partial D_i}$ into Eq. (4.2), the objective sensitivity can be expanded out as:

$$\frac{dL}{dD_i} = \frac{\partial L}{\partial D_i} - \left(\frac{\partial L}{\partial \boldsymbol{U}}\right) \cdot \left(\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{-1} \frac{\partial \boldsymbol{R}}{\partial D_i}\right) = \frac{\partial L}{\partial D_i} - \left(\left(\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{T}\right)^{-1} \frac{\partial L}{\partial \boldsymbol{U}}\right) \cdot \left(\frac{\partial \boldsymbol{R}}{\partial D_i}\right) \tag{4.5}$$

From Eq. (4.5), there are two main ways to go about evaluating the objective sensitivity. One way is to solve what is called the forward, or tangent problem, in which the following steps are performed for each design variable individually:

$$\begin{gathered}(1)\ solve\ \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]\frac{\partial \boldsymbol{U}}{\partial D_i} = -\frac{\partial \boldsymbol{R}}{\partial D_i} \\ (2)\ evaluate\,\frac{dL}{dD_i} = \frac{\partial L}{\partial D_i} + \left(\frac{\partial L}{\partial \boldsymbol{U}}\right)\cdot\left(\frac{\partial \boldsymbol{U}}{\partial D_i}\right)\end{gathered} \tag{4.6}$$

In a sense the tangent method is straightforward in implementation, as it closely parallels the process of running the original simulation for the system. But notice that it requires the solution of a linear system for every design variable at a given design state. This can be very costly if the number of design variables is large, and there can be hundreds or thousands in some cases.

An alternate way of evaluating the objective sensitivity is to implement the adjoint. The critical observation is that the terms $\frac{\partial L}{\partial \boldsymbol{U}}$ and $\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]$ are invariant at a given design state, and do not take different values for each design variable. Therefore, the product of these terms can be evaluated a single time at each design state, and used repeatedly for each design variable. The process then becomes to first evaluate the adjoint, $\boldsymbol{\Lambda}$, by solving

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^T \boldsymbol{\Lambda} = \frac{\partial L}{\partial \boldsymbol{U}} \tag{4.7}$$

Then, for each design variable, obtain the sensitivity by evaluating

$$\frac{dL}{dD_i} = \frac{\partial L}{\partial D_i} - \boldsymbol{\Lambda}\cdot\left(\frac{\partial \boldsymbol{R}}{\partial D_i}\right) \tag{4.8}$$

This way only one linear system must be solved at a given design state to get the objective sensitivity, regardless of the number of design variables. The sensitivity of the governing equations, $\frac{\partial \boldsymbol{R}}{\partial D_i}$, is also generally inexpensive to evaluate, so the process for each design variable is little more than a dot product of two vectors with the dimension of the system. The use of the adjoint makes sensitivity analysis possible for problems that may be infeasible otherwise.

The only real drawback is from an implementation point of view, in that the application of the adjoint can be somewhat less intuitive than other methods. This stems from the fact that the adjoint is obtained from the *transpose* system of the governing equations, as indicated in Eq. (4.7). For problems involving static analysis, where all solution variables are solved for simultaneously, this is usually not a difficult adjustment to make, especially in disciplines that produce symmetric governing equation matrices, like structural elastic and thermal equations. For such cases, Eq. (4.7) can be implemented directly as-is, using the Jacobian matrix of the static governing equations. But in dynamic problems, or problems solved in a sequential manner, it requires a significantly different arrangement and order of operations, as will shortly be seen.

Consider the process of implicit time integration, such as is described in Chapter 3, Section 3.3. The state of a system is known at some initial point in time, and the state at a series of time steps are obtained in order, using the governing equations and the state of variables at the previous time step. Although the state at each time step is solved for individually and sequentially, the solution history obtained throughout the entire time interval can be thought of as a single long vector of solution variables, governed by a single global system of equations, as represented in Eq. (4.9).

$$\boldsymbol{U} = \begin{Bmatrix} \boldsymbol{U}^0 \\ \boldsymbol{U}^1 \\ \boldsymbol{U}^2 \\ \boldsymbol{U}^3 \\ \vdots \end{Bmatrix} \quad \boldsymbol{R} = \begin{Bmatrix} \boldsymbol{R}^0(\boldsymbol{U}^0) \\ \boldsymbol{R}^1(\boldsymbol{U}^0, \boldsymbol{U}^1) \\ \boldsymbol{R}^2(\boldsymbol{U}^1, \boldsymbol{U}^2) \\ \boldsymbol{R}^3(\boldsymbol{U}^2, \boldsymbol{U}^3) \\ \vdots \end{Bmatrix} \tag{4.9}$$

If the complete Jacobian matrix was formed for this global space-time system, it would have the following structure:

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right] = \begin{bmatrix} \left[\frac{\partial \boldsymbol{R}^0}{\partial \boldsymbol{U}^0}\right] & 0 & 0 & 0 & \cdots \\ \left[\frac{\partial \boldsymbol{R}^1}{\partial \boldsymbol{U}^0}\right] & \left[\frac{\partial \boldsymbol{R}^1}{\partial \boldsymbol{U}^1}\right] & 0 & 0 & \cdots \\ 0 & \left[\frac{\partial \boldsymbol{R}^2}{\partial \boldsymbol{U}^1}\right] & \left[\frac{\partial \boldsymbol{R}^2}{\partial \boldsymbol{U}^2}\right] & 0 & \cdots \\ 0 & 0 & \left[\frac{\partial \boldsymbol{R}^3}{\partial \boldsymbol{U}^2}\right] & \left[\frac{\partial \boldsymbol{R}^3}{\partial \boldsymbol{U}^{\mathbf{3}}}\right] & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{bmatrix} \tag{4.10}$$

Examining the structure of this global matrix, it is clear that obtaining the solution sensitivity in the tangent problem entails beginning at the initial time step and marching sequentially through all subsequent time steps, in a process that is essentially forward substitution. The two-step task on each time step when performing step 1 of Eq. (4.6) is

$$\begin{gathered} (1)\; set\; \widetilde{\boldsymbol{R}}_i = -\frac{\partial \boldsymbol{R}^{n+1}}{\partial D_i} - \left[\frac{\partial \boldsymbol{R}^{n+1}}{\partial \boldsymbol{U}^n}\right]\frac{\partial \boldsymbol{U}^n}{\partial D_i} \\ (2)\; solve\; \left[\frac{\partial \boldsymbol{R}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right]\frac{\partial \boldsymbol{U}^{n+1}}{\partial D_i} = \widetilde{\boldsymbol{R}}_i \end{gathered} \tag{4.11}$$

In contrast, following Eq. (4.7) the global adjoint would be found with the transpose of the global system:

$$\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \left[\frac{\partial \boldsymbol{R}^{m-3}}{\partial \boldsymbol{U}^{m-3}}\right]^T & \left[\frac{\partial \boldsymbol{R}^{m-2}}{\partial \boldsymbol{U}^{m-3}}\right]^T & 0 & 0 \\ \cdots & 0 & \left[\frac{\partial \boldsymbol{R}^{m-2}}{\partial \boldsymbol{U}^{m-2}}\right]^T & \left[\frac{\partial \boldsymbol{R}^{m-1}}{\partial \boldsymbol{U}^{m-2}}\right]^T & 0 \\ \cdots & 0 & 0 & \left[\frac{\partial \boldsymbol{R}^{m-1}}{\partial \boldsymbol{U}^{m-1}}\right]^T & \left[\frac{\partial \boldsymbol{R}^{m}}{\partial \boldsymbol{U}^{m-1}}\right] \\ \cdots & 0 & 0 & 0 & \left[\frac{\partial \boldsymbol{R}^{m}}{\partial \boldsymbol{U}^{m}}\right]^{\prime} \end{bmatrix} \begin{Bmatrix} \vdots \\ \boldsymbol{\Lambda}^{m-3} \\ \boldsymbol{\Lambda}^{m-2} \\ \boldsymbol{\Lambda}^{m-1} \\ \boldsymbol{\Lambda}^{m} \end{Bmatrix} = \begin{Bmatrix} \vdots \\ \frac{\partial L}{\partial \boldsymbol{U}^{m-3}} \\ \frac{\partial L}{\partial \boldsymbol{U}^{m-2}} \\ \frac{\partial L}{\partial \boldsymbol{U}^{m-1}} \\ \frac{\partial L}{\partial \boldsymbol{U}^{m}} \end{Bmatrix} \tag{4.12}$$

The transpose of the global system is upper-triangular in terms of time step blocks, instead of lower triangular. This means that the adjoint must be computed starting with the last time step and marching back, using the following two-step back-substitution for each time step:

$$\begin{gathered}(1)\ if\ n < m\ set\ \tilde{\boldsymbol{L}} = \frac{\partial L}{\partial \boldsymbol{U}^n} - \left[\frac{\partial \boldsymbol{R}^{n+1}}{\partial \boldsymbol{U}^n}\right]^T \boldsymbol{\Lambda}^{n+1} \\ (2)\ solve\ \left[\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n}\right]^T \boldsymbol{\Lambda}^n = \tilde{\boldsymbol{L}}\end{gathered} \tag{4.13}$$

An important implication that follows is that the entire solution history must be computed before beginning the adjoint computation process. The solution must be saved (written to disk) for every time step so that it is available for evaluation of $\frac{\partial L}{\partial \boldsymbol{U}}$ and the Jacobian matrix upon returning in the back-substitution process. This is in contrast to the tangent problem, in which the sensitivity of the solution $\frac{\partial \boldsymbol{U}}{\partial D_i}$ can be computed alongside the solution itself on each time step, factored into the objective sensitivity and then discarded all in one sweep, since it has the same order of operations as the solution process. Because of this, more disk storage is required when applying the adjoint to dynamic problems.

Based on this development, the key to applying the adjoint in any given discipline of analysis is to identify and define the global Jacobian matrix $\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]$ of the governing equations, and the sensitivity of the governing equations $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{D}}$. The following sections give the derivation of this for the finite element thermoelastic analysis implementation in AStrO.

4.2 Application to Thermoelastic Structural Analysis

This section is dedicated to the application of sensitivity analysis using the adjoint covered in Section 4.1 to the finite element formulation for thermoelastic analysis developed in Chapter 3. Two main ingredients are necessary to apply the adjoint: the Jacobian matrix of the governing equations and the sensitivity of the governing equations with respect to design variables. Section 4.2.1 is dedicated to the derivation of the former, and Section 4.2.2 to the latter.

4.2.1 Global Jacobian Matrix of Governing Equations

To obtain the adjoint for a system governed by a set of global equations requires solving Eq. (4.7), with the transpose of the Jacobian matrix of the governing equations $\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]$. The Jacobian matrix represents the sensitivities of the governing equations with respect to the degrees of freedom of the discretized solution. Forming the matrix therefore entails partial differentiation of each component of the governing equations by each applicable variable.

Let us first review the governing equations developed in Chapter 3, and then show the necessary differentiation to form the global Jacobian matrix. The comprehensive equations for dynamic analysis will be examined first, followed by the appropriate simplifications for static analysis. The discretized governing equations for dynamic thermoelastic modeling implemented by AStrO are given in Table 4.1:

Table 4.1 Summary of governing finite element equations for thermoelastic analysis.

$Heat\ Conduction, Poisson\ Equation$: $\boldsymbol{R}_{\phi}^{n+1} = \boldsymbol{R}_{\phi,k}^{n+1} + \boldsymbol{R}_{\phi,m}^{n+1} + \boldsymbol{R}_{\phi,hg}^{n+1} = 0$ $\boldsymbol{R}_{\dot{\phi}}^{n+1} = \boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^{n} - \Delta t\left((1-\gamma)\dot{\boldsymbol{\phi}}^{n} + \gamma\dot{\boldsymbol{\phi}}^{n+1}\right) = 0$ $0 < \gamma \leq 1$	$\boldsymbol{R}_{\phi,k} = -\int_{\Omega}[\nabla N]\boldsymbol{q}d\Omega = [K_{\phi}]\boldsymbol{\phi}$ $\boldsymbol{R}_{\phi,m} = \int_{\Omega}\rho C_p \boldsymbol{N}\boldsymbol{N}^T\dot{\boldsymbol{\phi}}d\Omega = [M_{\phi}]\dot{\boldsymbol{\phi}}$ $\boldsymbol{R}_{\phi,hg} = -\int_{\Omega}\boldsymbol{N}Qd\Omega + \int_{S}\boldsymbol{N}(\boldsymbol{q}\cdot\boldsymbol{n})dS$
$Elasticity, Principle\ of\ Virtual\ Work$: $\boldsymbol{R}_{u}^{n+1} = (1+\alpha)\left(\boldsymbol{R}_{u,k}^{n+1} + \boldsymbol{R}_{u,c}^{n+1} + \boldsymbol{R}_{u,app}^{n+1}\right) - \alpha\left(\boldsymbol{R}_{u,k}^{n} + \boldsymbol{R}_{u,c}^{n} + \boldsymbol{R}_{u,app}^{n}\right)$ $+\boldsymbol{R}_{u,m}^{n+1} + \boldsymbol{R}_{u,th}^{n+1} = 0$ $\boldsymbol{R}_{\ddot{u}}^{n+1} = \boldsymbol{U}^{n+1} - \boldsymbol{U}^{n} - \Delta t\dot{\boldsymbol{U}}^{n} - \frac{1}{2}\Delta t^2\left((1-2\beta)\ddot{\boldsymbol{U}}^{n} + 2\beta\ddot{\boldsymbol{U}}^{n+1}\right) = 0$ $\boldsymbol{R}_{\ddot{u}}^{n+1} = \dot{}\boldsymbol{U}^{n+1} - \dot{\boldsymbol{U}}^{n} - \Delta t\left((1-\gamma)\ddot{\boldsymbol{U}}^{n} + \gamma\ddot{\boldsymbol{U}}^{n+1}\right) = 0$ $0 < \beta \leq \frac{1}{2}$ $0 < \gamma \leq 1$ $-1 < \alpha \leq 0$	$\boldsymbol{R}_{u,k} = \int_{\Omega}\left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C]\boldsymbol{\epsilon}d\Omega$ $\boldsymbol{R}_{u,c} = \int_{\Omega}\xi[N]^T[N]\dot{\boldsymbol{U}}d\Omega = [C_u]\dot{\boldsymbol{U}}$ $\boldsymbol{R}_{u,m} = \int_{\Omega}\rho[N]^T[N]\ddot{\boldsymbol{U}}d\Omega = [M_u]\ddot{\boldsymbol{U}}$ $\boldsymbol{R}_{u,app} = -\int_{\Omega}[N]^T\boldsymbol{f}d\Omega - \int_{S}[N]^T\boldsymbol{t}dS$ $\boldsymbol{R}_{u,th} = -\int_{\Omega}\Delta T\left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C]\boldsymbol{\alpha}_{TE}d\Omega$

For full thermoelastic dynamic analysis, there are two sets of fundamental governing equations for the thermal and elastic responses, each accompanied by the equations defining the Newmark Beta expansions for time derivatives, for a total of five sets of governing equations applying at each time step. Each of the governing equations for dynamic analysis can be differentiated with respect to the solution variables at the current time step n and the next time step $n+1$, as shown in Tables 4.2 and 4.3.

Table 4.2 Differentiation of governing equations for thermal heat conduction.

	$\boldsymbol{R}_{\phi}^{n+1}$	$\boldsymbol{R}_{\dot{\phi}}^{n+1}$
$\frac{\partial}{\partial \boldsymbol{\phi}^{n}}$	$-\frac{1}{\gamma \Delta t}[M_{\phi}]$	$-[I]$
$\frac{\partial}{\partial \dot{\boldsymbol{\phi}}^{n}}$	$-\frac{(1-\gamma)}{\gamma}[M_{\phi}]$	$-\Delta t(1-\gamma)[I]$
$\frac{\partial}{\partial \boldsymbol{U}^{n}}$	0	0
$\frac{\partial}{\partial \ddot{\boldsymbol{U}}^{n}}$	0	0
$\frac{\partial}{\partial \dot{\boldsymbol{U}}^{n}}$	0	0
$\frac{\partial}{\partial \boldsymbol{\phi}^{n+1}}$	$[K_{\phi}]+\frac{1}{\gamma \Delta t}[M_{\phi}]$	$[I]$
$\frac{\partial}{\partial \dot{\boldsymbol{\phi}}^{n+1}}$	0	$-\gamma \Delta t[I]$
$\frac{\partial}{\partial \boldsymbol{U}^{n+1}}$	0	0
$\frac{\partial}{\partial \ddot{\boldsymbol{U}}^{n+1}}$	0	0
$\frac{\partial}{\partial \dot{\boldsymbol{U}}^{n+1}}$	0	0

Table 4.3 Differentiation of governing equations of elasticity.

	$\boldsymbol{R}_u^{n+1}$	$\boldsymbol{R}_{\ddot{u}}^{n+1}$	$\boldsymbol{R}_{\dot{u}}^{n+1}$
$\frac{\partial}{\partial \boldsymbol{\phi}^n}$	$-\alpha\left[\frac{\partial \boldsymbol{R}_{u,th}^{n+1}}{\partial \boldsymbol{\phi}^n}\right]$	0	0
$\frac{\partial}{\partial \dot{\boldsymbol{\phi}}^n}$	0	0	0
$\frac{\partial}{\partial \boldsymbol{U}^n}$	$-\alpha[K_u^n]-\frac{(1+\alpha)\gamma}{\beta\Delta t}[C_u]+[M_u]$	$-[I]$	0
$\frac{\partial}{\partial \ddot{\boldsymbol{U}}^n}$	$(1+\alpha)\left((1-\gamma)-\gamma\left(\frac{1}{2\beta}-1\right)\right)\Delta t[C_u]-\left(\frac{1}{2\beta}-1\right)[M_u]$	$-\frac{1}{2}\Delta t^2(1-2\beta)[I]$	$-(1-\gamma)\Delta t[I]$
$\frac{\partial}{\partial \dot{\boldsymbol{U}}^n}$	$\left((1+\alpha)\left(1-\frac{\gamma}{\beta}\right)-\alpha\right)[C_u]-\left(\frac{1}{\beta\Delta t}\right)[M_u]$	$-\Delta t[I]$	$-[I]$
$\frac{\partial}{\partial \boldsymbol{\phi}^{n+1}}$	$(1+\alpha)\left[\frac{\partial \boldsymbol{R}_{u,th}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right]$	0	0
$\frac{\partial}{\partial \dot{\boldsymbol{\phi}}^{n+1}}$	0	0	0
$\frac{\partial}{\partial \boldsymbol{U}^{n+1}}$	$(1+\alpha)\left([K_u^{n+1}]+\frac{\gamma}{\beta\Delta t}[C_u]\right)+\frac{1}{\beta\Delta t^2}[M_u]$	$[I]$	0
$\frac{\partial}{\partial \ddot{\boldsymbol{U}}^{n+1}}$	0	$-\beta\Delta t^2[I]$	$-\gamma\Delta t[I]$
$\frac{\partial}{\partial \dot{\boldsymbol{U}}^{n+1}}$	0	0	$[I]$

With the differentiated governing equations defined, the two essential blocks of the global Jacobian matrix at each time step can be assembled, shown in Eqs. (4.14) and (4.15). The transpose of the matrix in Eq. (4.15) is upper-triangular in terms of the five main sets of variables, and so the adjoint for each block can be found with back-substitution, after augmenting $\frac{\partial L}{\partial \boldsymbol{U}^n}$ with the matrix in Eq. (4.14), as described in Eq. (4.13). This procedure is repeated for every time step in reverse order after obtaining the solution at a given design state.

$$\left[\frac{\partial \boldsymbol{R}^{n+1}}{\partial \boldsymbol{U}^{n}}\right] = \begin{bmatrix} \left[\frac{\partial \boldsymbol{R}_{\phi}^{n+1}}{\partial \boldsymbol{\phi}^{n}}\right] & \left[\frac{\partial \boldsymbol{R}_{\phi}^{n+1}}{\partial \dot{\boldsymbol{\phi}}^{n}}\right] & 0 & 0 & 0 \\ \left[\frac{\partial \boldsymbol{R}_{\dot{\phi}}^{n+1}}{\partial \boldsymbol{\phi}^{n}}\right] & \left[\frac{\partial \boldsymbol{R}_{\dot{\phi}}^{n+1}}{\partial \dot{\boldsymbol{\phi}}^{n}}\right] & 0 & 0 & 0 \\ \left[\frac{\partial \boldsymbol{R}_{u}^{n+1}}{\partial \boldsymbol{\phi}^{n}}\right] & 0 & \left[\frac{\partial \boldsymbol{R}_{u}^{n+1}}{\partial \boldsymbol{U}^{n}}\right] & \left[\frac{\partial \boldsymbol{R}_{u}^{n+1}}{\partial \ddot{\boldsymbol{U}}^{n}}\right] & \left[\frac{\partial \boldsymbol{R}_{u}^{n+1}}{\partial \dot{\boldsymbol{U}}^{n}}\right] \\ 0 & 0 & \left[\frac{\partial \boldsymbol{R}_{\ddot{u}}^{n+1}}{\partial \boldsymbol{U}^{n}}\right] & \left[\frac{\partial \boldsymbol{R}_{\ddot{u}}^{n+1}}{\partial \ddot{\boldsymbol{U}}^{n}}\right] & \left[\frac{\partial \boldsymbol{R}_{\ddot{u}}^{n+1}}{\partial \dot{\boldsymbol{U}}^{n}}\right] \\ 0 & 0 & 0 & \left[\frac{\partial \boldsymbol{R}_{\dot{u}}^{n+1}}{\partial \ddot{\boldsymbol{U}}^{n}}\right] & \left[\frac{\partial \boldsymbol{R}_{\dot{u}}^{n+1}}{\partial \dot{\boldsymbol{U}}^{n}}\right] \end{bmatrix} \tag{4.14}$$

$$\left[\frac{\partial \boldsymbol{R}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] = \begin{bmatrix} \left[\frac{\partial \boldsymbol{R}_{\phi}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] & 0 & 0 & 0 & 0 \\ \left[\frac{\partial \boldsymbol{R}_{\dot{\phi}}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] & \left[\frac{\partial \boldsymbol{R}_{\dot{\phi}}^{n+1}}{\partial \dot{\boldsymbol{\phi}}^{n+1}}\right] & 0 & 0 & 0 \\ \left[\frac{\partial \boldsymbol{R}_{u}^{n+1}}{\partial \boldsymbol{\phi}^{n+1}}\right] & 0 & \left[\frac{\partial \boldsymbol{R}_{u}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] & 0 & 0 \\ 0 & 0 & \left[\frac{\partial \boldsymbol{R}_{\ddot{u}}^{n+1}}{\partial \boldsymbol{U}^{n+1}}\right] & \left[\frac{\partial \boldsymbol{R}_{\ddot{u}}^{n+1}}{\partial \ddot{\boldsymbol{U}}^{n+1}}\right] & 0 \\ 0 & 0 & 0 & \left[\frac{\partial \boldsymbol{R}_{\dot{u}}^{n+1}}{\partial \ddot{\boldsymbol{U}}^{n+1}}\right] & \left[\frac{\partial \boldsymbol{R}_{\dot{u}}^{n+1}}{\partial \dot{\boldsymbol{U}}^{n+1}}\right] \end{bmatrix} \tag{4.15}$$

For static analysis, the governing equations reduce to

$$\begin{gathered} \boldsymbol{R}_{\phi} = \boldsymbol{R}_{\phi,k} + \boldsymbol{R}_{\phi,hg} = 0 \\ \boldsymbol{R}_{u} = \boldsymbol{R}_{u,k} + \boldsymbol{R}_{u,app} + \boldsymbol{R}_{u,th} = 0 \end{gathered} \tag{4.16}$$

with the same definitions for terms as given in Table 4.1. As mentioned in Chapter 3, AStrO assumes one-way dependence of elastic displacement on temperature distribution for thermoelastic analysis, so that the temperature and displacement are solved for sequentially in a two-step process. This means that even in static problems, when both disciplines are considered, the basic procedure just described for dynamic problems holds but within the confines of a single steady-state system. The global Jacobian for this static problem takes the form

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right] = \begin{bmatrix} \left[\frac{\partial \boldsymbol{R}_\phi}{\partial \boldsymbol{\phi}}\right] & 0 \\ \left[\frac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{\phi}}\right] & \left[\frac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{U}}\right] \end{bmatrix} = \begin{bmatrix} [K_\phi] & 0 \\ \left[\frac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{\phi}}\right] & [K_u] \end{bmatrix} \tag{4.17}$$

So that finding the adjoint of the system can be broken down into three steps:

$$\begin{gathered} (1)\ solve\ [K_u]^T \boldsymbol{\Lambda}_u = \frac{\partial L}{\partial \boldsymbol{U}} \\ (2)\ set\ \tilde{\boldsymbol{L}} = \frac{\partial L}{\partial \boldsymbol{\phi}} - \left[\frac{\partial \boldsymbol{R}_u}{\partial \boldsymbol{\phi}}\right]^T \boldsymbol{\Lambda}_u \\ (3)\ solve\ \left[K_\phi\right]^T \boldsymbol{\Lambda}_\phi = \tilde{\boldsymbol{L}} \end{gathered} \tag{4.18}$$

Both $[K_u]$ and $[K_\phi]$ are symmetric, so in this case the transposes on those two matrices in Eq. (4.18) are meaningless, but still written for completeness. In static analysis where only one of the thermal and elastic disciplines applies, the system reduces to only the corresponding stiffness matrix for that discipline, and Eq. (4.7) applies directly.

4.2.2 Sensitivity of Governing Equations to Design Variables

The last major component in applying the adjoint to thermoelastic analysis is the sensitivity of the governing equations with respect to design variables, to complete the objective sensitivity defined in Eq. (4.8). A natural first step is to identify what the design variables might be for systems like this. AStrO has built-in support for ten main categories of design variables, as listed below. The first seven categories are material or section-based properties, that are associated with elements or sets of elements in the mesh. The last three types are properties associated with nodes and sets of nodes.

1. ***Elastic Properties*** – The elastic material properties used to form the material stiffness matrix for orthotropic materials as defined in Chapter 3, Section 3.2. This consists of three elastic moduli, three values of Poisson’s ratio and three shear moduli corresponding to the

material's local principle directions. The ordering convention is $E_1, E_2, E_3, \nu_{12}, \nu_{13}, \nu_{23}$, G_{12}, G_{13}, G_{23}.

2. ***Mass Density*** – The mass density ρ of a material used to form the mass matrices $[M_u]$ and $[M_\phi]$ for the thermal and elastic equations.
3. ***Thermal Conductivity*** – The thermal conductivity $\boldsymbol{k}$ in the principle directions of the local material orientation, used to form the stiffness matrix for the thermal equations $[K_\phi]$.
4. ***Coefficient of Thermal Expansion*** – The vector of coefficients of thermal expansion $\boldsymbol{\alpha}_{TE}$, which factors into the thermal load term in the elasticity equations, $\boldsymbol{R}_{u,th}$.
5. ***Specific Heat Capacity*** – The specific heat capacity C_p of a material, which factors into the thermal mass matrix $[M_\phi]$ in the thermal equations.
6. ***Local Material Orientation*** – The local coordinate system of an element or section, defining the orientation in which material properties are to be defined and denoted with the direction cosine matrix $[\alpha_L]$.
7. ***Section Properties*** – Parameters defining cross-sectional properties of an element or section, only applicable to sections composed of shell or beam elements. For shell sections, there are only two section properties, defining the thickness of the shell, and the offset from the shell midplane that is to serve as the reference plane for that shell. For beams, there are 12 properties defined in order as follows: cross-sectional area A, second moments of area in the 2-3 plane I_2, I_{23}, I_3, polar moment of area J, three components of a unit vector defining the 2-direction n_1, n_2, n_3, elastic modulus E, Poisson's ratio ν, shear modulus G, and mass density ρ. The final four properties for beams are material properties that are redundant of those previously listed. For beams, these material properties are given in the section definition in the model input file instead of a separate material definition, and their values are placed appropriately into the fields for elastic and mass properties.
8. ***Nodal Coordinates*** – The initial x_1, x_2, and x_3 coordinates of the nodes defining the mesh of the structure before deformation.
9. ***Applied Structural Load*** – The nodal loads applied on the structure in the equations of elasticity, denoted $\boldsymbol{R}_{u,app}$.
10. ***Applied Thermal Load*** – The nodal thermal loads representing surface flux and internal heat generation in the thermal equations, denoted $\boldsymbol{R}_{\phi,hg}$.

Categories nine and ten in the list are not what most would consider design parameters, but for some problems it can be useful to treat them as such, so they can be tailored to achieve an objective. These ten quantities can be defined by the user as functions of $\boldsymbol{D}$, so that their values and sensitivities can be accessed directly at any given design state from user subroutines.

What is ultimately needed is the sensitivity of the residual of governing equations, $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{D}}$. The next step, then, is to go through each component of the governing equations and perform differentiation considering any factor that can possibly depend on the above design variables. Table 4.4 shows the form of $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{D}}$ for every main component of the thermoelastic governing equations. All terms are expressed in natural coordinate integrals, to include all the relevant factors.

Table 4.4 Design variable sensitivities of thermoelastic governing equations.

Term	$\frac{\partial}{\partial D_i}$
$\boldsymbol{R}_{\phi,k}$	$-\int_{\hat{\Omega}}\left(\frac{\partial[\nabla N]}{\partial D_i}\boldsymbol{q}\lvert[J]\rvert + [\nabla N]\frac{\partial \boldsymbol{q}}{\partial D_i}\lvert[J]\rvert + [\nabla N]\boldsymbol{q}\frac{\partial\lvert[J]\rvert}{\partial D_i}\right)d\hat{\Omega}$
$\boldsymbol{R}_{\phi,m}$	$\int_{\hat{\Omega}}\left(\frac{\partial\rho}{\partial D_i}C_p\lvert[J]\rvert + \rho\frac{\partial C_p}{\partial D_i}\lvert[J]\rvert + \rho C_p\frac{\partial\lvert[J]\rvert}{\partial D_i}\right)\boldsymbol{N}\boldsymbol{N}^T\dot{\boldsymbol{\phi}}d\hat{\Omega}$
$\boldsymbol{R}_{u,k}$	$\int_{\hat{\Omega}}\left(\frac{\partial}{\partial D_i}\left(\left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T\right)[C]\boldsymbol{\epsilon}\lvert[J]\rvert + \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T\frac{\partial[C]}{\partial D_i}\boldsymbol{\epsilon}\lvert[J]\rvert + \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T[C]\frac{\partial\boldsymbol{\epsilon}}{\partial D_i}\lvert[J]\rvert + \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T[C]\boldsymbol{\epsilon}\frac{\partial\lvert[J]\rvert}{\partial D_i}\right)d\hat{\Omega}$
$\boldsymbol{R}_{u,c}$	$\int_{\hat{\Omega}}\left(\frac{\partial\xi}{\partial D_i}\lvert[J]\rvert + \xi\frac{\partial\lvert[J]\rvert}{\partial D_i}\right)[N]^T[N]\dot{\boldsymbol{U}}d\hat{\Omega}$
$\boldsymbol{R}_{u,m}$	$\int_{\hat{\Omega}}\left(\frac{\partial\rho}{\partial D_i}\lvert[J]\rvert + \rho\frac{\partial\lvert[J]\rvert}{\partial D_i}\right)[N]^T[N]\ddot{\boldsymbol{U}}d\hat{\Omega}$
$\boldsymbol{R}_{u,th}$	$\begin{aligned}\int_{\hat{\Omega}}\left(\frac{\partial}{\partial D_i}\left(\left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T\right)[C]\boldsymbol{\alpha}_{TE}\lvert[J]\rvert + \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T\frac{\partial[C]}{\partial D_i}\boldsymbol{\alpha}_{TE}\lvert[J]\rvert + \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T[C]\frac{\partial\boldsymbol{\alpha}_{TE}}{\partial D_i}\lvert[J]\rvert\right. \\ \left. + \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{U}}\right]^T[C]\boldsymbol{\alpha}_{TE}\frac{\partial\lvert[J]\rvert}{\partial D_i}\right)(\boldsymbol{N}\cdot\boldsymbol{\phi})d\hat{\Omega}\end{aligned}$

Several terms show up frequently in the sensitivity expressions of Table 4.4. To get the full definitions these must be broken down individually. Beginning with the most fundamental, the term $|[J]|$, the determinant of the natural element space Jacobian matrix and equal to $\left(\frac{\partial\Omega}{\partial\hat{\Omega}}\right)$, appears in every integral as explained in Chapter 3, Section 3.1. The Jacobian matrix is given by

$$[J] = \left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\eta}}\right] = [\alpha_L][X]\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right] \tag{4.19}$$

Since it depends on the nodal coordinates of the element $[X]$ (where X_{ij} = the x_i coordinate of node j) and the local material orientation $[\alpha_L]$ which can depend on design variables, the sensitivity of the Jacobian is

$$\frac{\partial[J]}{\partial D_i} = \left(\frac{\partial[\alpha_L]}{\partial D_i}[X] + [\alpha_L]\frac{\partial[X]}{\partial D_i}\right)\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right] \tag{4.20}$$

With Eqs. (4.19) and (4.20), the determinant and its sensitivity can be defined as

$$\begin{aligned}
|[J]| = \begin{vmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{vmatrix} &= J_{11}(J_{22}J_{33} - J_{23}J_{32}) + J_{12}(J_{23}J_{31} - J_{21}J_{33}) + J_{13}(J_{21}J_{32} - J_{22}J_{31}) \\
\Rightarrow \frac{\partial|[J]|}{\partial D_i} &= \frac{\partial J_{11}}{\partial D_i}(J_{22}J_{33} - J_{23}J_{32}) + J_{11}\left(\frac{\partial J_{22}}{\partial D_i}J_{33} + J_{22}\frac{\partial J_{33}}{\partial D_i} - \frac{\partial J_{23}}{\partial D_i}J_{32} - J_{23}\frac{\partial J_{32}}{\partial D_i}\right) \\
&+ \frac{\partial J_{12}}{\partial D_i}(J_{23}J_{31} - J_{21}J_{23}) + J_{12}\left(\frac{\partial J_{23}}{\partial D_i}J_{31} + J_{23}\frac{\partial J_{31}}{\partial D_i} - \frac{\partial J_{21}}{\partial D_i}J_{33} - J_{21}\frac{\partial J_{33}}{\partial D_i}\right) \\
&+ \frac{\partial J_{13}}{\partial D_i}(J_{21}J_{32} - J_{22}J_{31}) + J_{13}\left(\frac{\partial J_{21}}{\partial D_i}J_{32} + J_{21}\frac{\partial J_{32}}{\partial D_i} - \frac{\partial J_{22}}{\partial D_i}J_{31} - J_{22}\frac{\partial J_{31}}{\partial D_i}\right)
\end{aligned} \tag{4.21}$$

The inverse of the natural element space Jacobian is also of importance in the construction of the equations and sensitivities. Once $[J]^{-1}$ and $\frac{\partial[J]}{\partial D}$ are computed, the simplest way to obtain the sensitivity of $[J]^{-1}$ is to realize that, by definition, the inverse relationship

$$[J][J]^{-1} = [I] \tag{4.22}$$

must always hold, regardless of the state of $\boldsymbol{D}$. Therefore, it must follow by differentiation of Eq. (4.22) that

$$\frac{\partial[J]}{\partial D_i}[J]^{-1} + [J]\frac{\partial[J]^{-1}}{\partial D_i} = 0$$
$$\Rightarrow \frac{\partial[J]^{-1}}{\partial D_i} = -[J]^{-1}\left(\frac{\partial[J]}{\partial D_i}\right)[J]^{-1} \tag{4.23}$$

This technique is used in several places throughout the formulation within AStrO for the differentiation of inverse matrices.

Another critical component in the governing equations is the gradient of the basis functions, $[\nabla N]$. As shown in Chapter 3, this matrix is can be constructed as

$$[\nabla N] = \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right]\left[\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{x}}\right] = \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right][J]^{-1} \tag{4.24}$$

Using the sensitivity of the inverse Jacobian defined in Eq. (4.23) the sensitivity of the basis function gradient is

$$\frac{\partial[\nabla N]}{\partial D_i} = \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\eta}}\right]\frac{\partial[J]^{-1}}{\partial D_i} \tag{4.25}$$

The sensitivity of the heat flux $\boldsymbol{q}$ in the governing equations for heat conduction easily follows, as

$$\boldsymbol{q} = -\left[k_1\left(\frac{\partial \boldsymbol{N}}{\partial x_1}\cdot\boldsymbol{\phi}\right), k_2\left(\frac{\partial \boldsymbol{N}}{\partial x_2}\cdot\boldsymbol{\phi}\right), k_3\left(\frac{\partial \boldsymbol{N}}{\partial x_3}\cdot\boldsymbol{\phi}\right)\right]^T$$
$$\frac{\partial \boldsymbol{q}}{\partial D_i} = -\left[\left(\frac{\partial k_1}{\partial D_i}\frac{\partial \boldsymbol{N}}{\partial x_1} + k_1\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}}{\partial x_1}\right)\right)\cdot\boldsymbol{\phi}, \left(\frac{\partial k_2}{\partial D_i}\frac{\partial \boldsymbol{N}}{\partial x_2} + k_2\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}}{\partial x_2}\right)\right)\cdot\boldsymbol{\phi}, \left(\frac{\partial k_3}{\partial D_i}\frac{\partial \boldsymbol{N}}{\partial x_3} + k_3\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}}{\partial x_3}\right)\right)\cdot\boldsymbol{\phi}\right]^T \tag{4.26}$$

The remaining non-trivial terms in Table 4.4 for the governing equation sensitivities are strain $\boldsymbol{\epsilon}$ and material elastic stiffness matrix $[C]$. Since the forms of these differ between element types, they will be presented separately for solid, shell and beam elements.

Sensitivities of Strain and Stiffness for Solid Continuum Elements

For solid continuum elements, the definition of strain in terms of displacement degrees of freedom is given in Table 3.2 of Chapter 3, Section 3.2.1. Differentiation of those expressions by design variables gives the sensitivities of strain, as needed for the evaluation of terms in Table 4.4. The sensitivities are shown in Table 4.5.

Table 4.5 Sensitivities of strain for solid continuum elements. The second row of terms only apply to analysis with nonlinear geometry.

	$\frac{\partial}{\partial D_i}$
$[\epsilon]$	$\frac{1}{2}\left(\begin{array}{c}\frac{\partial[\alpha_L]}{\partial D_i}[U][\nabla N]+[\alpha_L][U]\frac{\partial[\nabla N]}{\partial D_i}+\frac{\partial[\nabla N]^T}{\partial D_i}[U]^T[\alpha_L]^T+[\nabla N]^T[U]^T\frac{\partial[\alpha_L]^T}{\partial D_i}\\ +\frac{\partial[\nabla N]^T}{\partial D_i}[U]^T[U][\nabla N]+[\nabla N]^T[U]^T[U]\frac{\partial[\nabla N]}{\partial D_i}\end{array}\right)$
$\frac{\partial\epsilon_{jk}}{\partial U_{lm}}$	$\frac{1}{2}\left(\begin{array}{c}\frac{\partial\alpha_{L,jl}}{\partial D_i}\frac{\partial N_m}{\partial x_k}+\alpha_{L,jl}\frac{\partial}{\partial D_i}\left(\frac{\partial N_m}{\partial x_k}\right)+\frac{\partial\alpha_{L,kl}}{\partial D_i}\frac{\partial N_m}{\partial x_j}+\alpha_{L,kl}\frac{\partial}{\partial D_i}\left(\frac{\partial N_m}{\partial x_j}\right)\\ +\left(\boldsymbol{U}_l\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial\boldsymbol{N}}{\partial x_j}\right)\right)\frac{\partial N_m}{\partial x_k}+\left(\boldsymbol{U}_l\cdot\frac{\partial\boldsymbol{N}}{\partial x_j}\right)\frac{\partial}{\partial D_i}\left(\frac{\partial N_m}{\partial x_k}\right)+\left(\boldsymbol{U}_l\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial\boldsymbol{N}}{\partial x_k}\right)\right)\frac{\partial N_m}{\partial x_j}+\left(\boldsymbol{U}_l\cdot\frac{\partial\boldsymbol{N}}{\partial x_k}\right)\frac{\partial}{\partial D_i}\left(\frac{\partial N_m}{\partial x_j}\right)\end{array}\right)$

The orthotropic elastic stiffness matrix $[C]$ is formed for solid elements as shown in Eq. (3.52) of Chapter 3. This is another situation where the sensitivities of the inverse matrix $[S]$, known as the compliance matrix, are simpler to construct and evaluate directly than for $[C]$ itself. Therefore, the sensitivity of the material stiffness matrix is evaluated as

$$\frac{\partial[C]}{\partial D_i}=-[C]\frac{\partial[S]}{\partial D_i}[C] \tag{4.27}$$

where the sensitivity of $[S]$ takes the following form:

$$\frac{\partial[S]}{\partial D_i} = \begin{bmatrix} -\frac{1}{E_1^2}\frac{\partial E_1}{\partial D_i} & \left(-\frac{\partial \nu_{12}}{\partial D_i}\frac{1}{E_1}+\frac{\nu_{12}}{E_1^2}\frac{\partial E_1}{\partial D_i}\right) & \left(-\frac{\partial \nu_{13}}{\partial D_i}\frac{1}{E_1}+\frac{\nu_{13}}{E_1^2}\frac{\partial E_1}{\partial D_i}\right) & 0 & 0 & 0 \\ & -\frac{1}{E_2^2}\frac{\partial E_2}{\partial D_i} & \left(-\frac{\partial \nu_{23}}{\partial D_i}\frac{1}{E_2}+\frac{\nu_{23}}{E_2^2}\frac{\partial E_2}{\partial D_i}\right) & 0 & 0 & 0 \\ & & -\frac{1}{E_3^2}\frac{\partial E_3}{\partial D_i} & 0 & 0 & 0 \\ & & & -\frac{1}{G_{12}^2}\frac{\partial G_{12}}{\partial D_i} & 0 & 0 \\ & (sym) & & & -\frac{1}{G_{13}^2}\frac{\partial G_{13}}{\partial D_i} & 0 \\ & & & & & -\frac{1}{G_{23}^2}\frac{\partial G_{23}}{\partial D_i} \end{bmatrix} \tag{4.28}$$

Using Eqs. (4.27) and (4.28) to form the sensitivity of the elastic stiffness matrix, along with Table 4.5 for sensitivities of strain completes the definition of all terms required for the governing equation sensitivities for solid continuum elements.

Sensitivities of Strain and Stiffness for Shell Elements

As explained in Chapter 3, Section 3.2.2, the strain in shell elements is defined by computing a set of transformed nodal degrees of freedom at a given deformation state to remove rigid body rotation, termed instantaneous degrees of freedom. To find the sensitivities of strain then requires first finding the sensitivities of the instantaneous degrees of freedom. Table 4.6 shows the relevant expressions for instantaneous degrees of freedom from Chapter 3 differentiated by D_i.

Table 4.6 Sensitivities of instantaneous nodal degrees of freedom for shell elements.

Quantity	$\frac{\partial}{\partial D_i}$, Linear Geometry	$\frac{\partial}{\partial D_i}$, Nonlinear Geometry
$[U_I]$	$\frac{\partial[\alpha_L]}{\partial D_i}[U_G]$	$\frac{\partial[\alpha_I]}{\partial D_i}([X_G]+[U_G])+[\alpha_I]\frac{\partial[X_G]}{\partial D_i}-\frac{\partial[\alpha_L]}{\partial D_i}[X_G]-[\alpha_L]\frac{\partial[X_G]}{\partial D_i}$
$\frac{\partial U_{I,jk}}{\partial U_{G,lm}}$	$\frac{\partial \alpha_{L,jl}}{\partial D_i}I_{km}$	$\frac{\partial \alpha_{I,jl}}{\partial D_i}I_{km}$
$\frac{\partial[U_I]}{\partial \theta_{G,jk}}$	0	$\frac{\partial}{\partial D_i}\left(\frac{\partial[\alpha_I]}{\partial\theta_{G,jk}}\right)([X_G]+[U_G])+\frac{\partial[\alpha_I]}{\partial\theta_{G,jk}}\frac{\partial[X_G]}{\partial D_i}$
$[\theta_I]$	$\frac{\partial[\alpha_L]}{D_i}[\theta_G]$	$\frac{\partial[\alpha_I]}{\partial D_i}\left([\theta_G]-[\theta_{G,avg}]\right)$
$\frac{\partial\theta_{I,jk}}{\partial\theta_{G,lm}}$	$\frac{\partial \alpha_{L,jl}}{D_i}I_{km}$	$\left(\frac{\partial}{\partial D_i}\left(\frac{\partial[\alpha_I]}{\partial\theta_{G,lm}}\right)\left([\theta_G]-\left[\theta_{G,avg}\right]\right)\right)_{jk}+\frac{\partial\alpha_{I,jl}}{\partial D_i}\left(I_{km}-\frac{1}{n_{nd}}\tilde{I}_{km}\right)$

The instantaneous direction cosine matrix $[\alpha_I]$ is formed by rotating the principle directions of the local coordinate system of an element by its overall nodal rotation vector. This means it can be expressed as a product of the local direction cosine matrix $[\alpha_L]$, and a direction cosine matrix derived from the overall rotation, $[\alpha_\theta]$ as shown:

$$[\alpha_I]=[\alpha_L][\alpha_\theta] \tag{4.29}$$

which means its differentiation with respect to design variables and nodal rotations can be expressed as:

$$\begin{aligned}\frac{\partial[\alpha_I]}{\partial\theta_{G,jk}}&=[\alpha_L]\frac{\partial[\alpha_\theta]}{\partial\theta_{G,jk}}\\ \frac{\partial[\alpha_I]}{\partial D_i}&=\frac{\partial[\alpha_L]}{\partial D_i}[\alpha_\theta]\\ \frac{\partial}{D_i}\left(\frac{\partial[\alpha_I]}{\partial\theta_{G,jk}}\right)&=\frac{\partial[\alpha_L]}{\partial D_i}\frac{\partial[\alpha_\theta]}{\partial\theta_{G,jk}}\end{aligned} \tag{4.30}$$

Now, with the definitions in Table 4.6 and Eqs. (4.29) and (4.30), the sensitivities of strains for shells can be constructed by differentiating Eqs. (3.60) and (3.61) of Chapter 3:

$$
\begin{aligned}
\frac{\partial \epsilon_{11}}{\partial D_i} &= \left(\frac{\partial \boldsymbol{U}_{I,1}}{\partial D_i} + x_3 \frac{\partial \boldsymbol{\theta}_{I,2}}{\partial D_i}\right) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + (\boldsymbol{U}_{I,1} + x_3 \boldsymbol{\theta}_{I,2}) \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) \\
\frac{\partial \epsilon_{22}}{\partial D_i} &= \left(\frac{\partial \boldsymbol{U}_{I,2}}{\partial D_i} - x_3 \frac{\partial \boldsymbol{\theta}_{I,1}}{\partial D_i}\right) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2} + (\boldsymbol{U}_{I,2} - x_3 \boldsymbol{\theta}_{I,1}) \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right) \\
\frac{\partial \gamma_a}{\partial D_i} &= 2\left(\frac{\partial \boldsymbol{\theta}_{I,3}}{\partial D_i} \cdot \boldsymbol{N}_{nd}\right) + \left(\frac{\partial \boldsymbol{U}_{I,1}}{\partial D_i} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right) + \left(\boldsymbol{U}_{I,1} \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right)\right) \\
&\quad - \left(\frac{\partial \boldsymbol{U}_{I,2}}{\partial D_i} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) - \left(\boldsymbol{U}_{I,2} \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right)\right) \\
\frac{\partial \gamma_{12}}{\partial D_i} &= \left(\frac{\partial \boldsymbol{U}_{I,1}}{\partial D_i} + x_3 \frac{\partial \boldsymbol{\theta}_{I,2}}{\partial D_i}\right) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2} + (\boldsymbol{U}_{I,1} + x_3 \boldsymbol{\theta}_{I,2}) \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right) \\
&\quad + \left(\frac{\partial \boldsymbol{U}_{I,2}}{\partial D_i} - x_3 \frac{\partial \boldsymbol{\theta}_{I,1}}{\partial D_i}\right) \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + (\boldsymbol{U}_{I,2} - x_3 \boldsymbol{\theta}_{I,1}) \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) \\
\frac{\partial \gamma_{13}}{\partial D_i} &= \left(\frac{\partial \boldsymbol{\theta}_{I,2}}{\partial D_i} \cdot \boldsymbol{N}_{nd}\right) + \left(\frac{\partial \boldsymbol{U}_{I,nd,3}}{\partial D_i} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) + \left(\boldsymbol{U}_{I,nd,3} \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right)\right) \\
&\quad + \left(\frac{\partial \boldsymbol{U}_{I,int,3}}{\partial D_i} \cdot \frac{\partial \boldsymbol{N}_{int}}{\partial x_1}\right) + \left(\boldsymbol{U}_{I,int,3} \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{int}}{\partial x_1}\right)\right) \\
\frac{\partial \gamma_{23}}{\partial D_i} &= -\left(\frac{\partial \boldsymbol{\theta}_{I,1}}{\partial D_i} \cdot \boldsymbol{N}_{nd}\right) + \left(\frac{\partial \boldsymbol{U}_{I,nd,3}}{\partial D_i} \cdot \frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right) + \left(\boldsymbol{U}_{I,nd,3} \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_2}\right)\right) \\
&\quad + \left(\frac{\partial \boldsymbol{U}_{I,int,3}}{\partial D_i} \cdot \frac{\partial \boldsymbol{N}_{int}}{\partial x_2}\right) + \left(\boldsymbol{U}_{I,int,3} \cdot \frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{int}}{\partial x_2}\right)\right)
\end{aligned} \tag{4.31}
$$

The x_3 coordinate at a given integration point can be derived from the thickness of the shell h and the offset parameter n_z, defining the offset of the reference plane from the shell midplane as follows:

$$
x_3 = h\left(\frac{1}{2}\eta_3 - n_z\right) \tag{4.32}
$$

The sensitivity of x_3 can then be defined

$$
\frac{\partial x_3}{\partial D_i} = \frac{\partial h}{\partial D_i}\left(\frac{1}{2}\eta_3 - n_z\right) - h\frac{\partial n_z}{\partial D_i} \tag{4.33}
$$

The material compliance matrix for shell elements is defined in Eq. (3.62) of Chapter 3. Differentiation by design variables yields the sensitivities:

$$\frac{\partial[S]}{\partial D_i} = \begin{bmatrix} -\frac{1}{E_1^2}\frac{\partial E_1}{\partial D_i} & \left(-\frac{\partial \nu_{12}}{\partial D_i}\frac{1}{E_1} + \frac{\nu_{12}}{E_1^2}\frac{\partial E_1}{\partial D_i}\right) & 0 & 0 & 0 & 0 \\ & -\frac{1}{E_2^2}\frac{\partial E_2}{\partial D_i} & 0 & 0 & 0 & 0 \\ & & -\frac{1}{G_{12}^2}\frac{\partial G_{12}}{\partial D_i} & 0 & 0 & 0 \\ & & & -\frac{1}{G_{12}^2}\frac{\partial G_{12}}{\partial D_i} & 0 & 0 \\ & (sym) & & & -\frac{1}{G_{13}^2}\frac{\partial G_{13}}{\partial D_i} & 0 \\ & & & & & -\frac{1}{G_{23}^2}\frac{\partial G_{23}}{\partial D_i} \end{bmatrix} \tag{4.34}$$

With Eqs. (4.27) and (4.34), the sensitivity of the material stiffness matrix can be obtained. This completes the definition of sensitivities for all terms in the governing equations for shell elements.

Sensitivities of Strain and Stiffness for Beam Elements

For beam elements, the sensitivities of instantaneous degrees of freedom given for shell elements in Table 4.6 also apply. Using these, the sensitivities of strain, or rather the modes of deformation employed in the principle of virtual work for beams, can be found by differentiation of Eq. (3.66) of Chapter 3:

$$\begin{gathered} \frac{\partial \epsilon_1}{\partial D_i} = \frac{\partial \boldsymbol{U}_{I,1}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{U}_{I,1}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) \\ \frac{\partial \kappa_2}{\partial D_i} = \frac{\partial \boldsymbol{\theta}_{I,2}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{\theta}_{I,2}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) \\ \frac{\partial \kappa_3}{\partial D_i} = \frac{\partial \boldsymbol{\theta}_{I,3}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{\theta}_{I,3}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) \\ \frac{\partial \gamma_{12}}{\partial D_i} = \frac{\partial \boldsymbol{U}_{I,nd,2}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{U}_{I,nd,2}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) + \frac{\partial \boldsymbol{U}_{I,int,2}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{int}}{\partial x_1} + \boldsymbol{U}_{I,int,2}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{int}}{\partial x_1}\right) - \frac{\partial \boldsymbol{\theta}_{I,3}}{\partial D_i}\cdot\boldsymbol{N}_{nd} \\ \frac{\partial \gamma_{13}}{\partial D_i} = \frac{\partial \boldsymbol{U}_{I,nd,3}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{U}_{I,nd,3}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) + \frac{\partial \boldsymbol{U}_{I,int,3}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{int}}{\partial x_1} + \boldsymbol{U}_{I,int,3}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{int}}{\partial x_1}\right) + \frac{\partial \boldsymbol{\theta}_{I,2}}{\partial D_i}\cdot\boldsymbol{N}_{nd} \\ \frac{\partial \theta'}{\partial D_i} = \frac{\partial \boldsymbol{\theta}_{I,1}}{\partial D_i}\cdot\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1} + \boldsymbol{\theta}_{I,1}\cdot\frac{\partial}{\partial D_i}\left(\frac{\partial \boldsymbol{N}_{nd}}{\partial x_1}\right) \end{gathered} \tag{4.35}$$

The stiffness matrix mapping to the work conjugates for beams is straightforward to express and differentiate directly, using Eq. (3.67) of Chapter 3:

$$\frac{\partial[C]}{\partial D_i}=\begin{bmatrix}\frac{\partial E_1}{\partial D_i}A+E_1\frac{\partial A}{\partial D_i} & & & & & \\ & \frac{\partial E_1}{\partial D_i}I_2+E_1\frac{\partial I_2}{\partial D_i} & & & (0) & \\ & & \frac{\partial E_1}{\partial D_i}I_3+E_1\frac{\partial I_3}{\partial D_i} & & & \\ & & & \frac{\partial A}{\partial D_i}G_{12}+A\frac{\partial G_{12}}{\partial D_i} & & \\ & (0) & & & \frac{\partial A}{\partial D_i}G_{13}+A\frac{\partial G_{13}}{\partial D_i} & \\ & & & & & \frac{\partial G_{12}}{\partial D_i}J+G_{12}\frac{\partial J}{\partial D_i}\end{bmatrix} \quad (4.36)$$

Using the tables and equations presented in this section, the complete sensitivities of all governing equations for thermoelastic modeling in AStrO can be constructed, and used to obtain objective sensitivities.

4.3 Special Cases

There are a few special situations where, because of a unique definition of design variables or objective function, some extension or modification to the standard adjoint approach is warranted. Part of the assumption of the general development presented in Section 4.1 is that the terms $\frac{\partial L}{\partial \boldsymbol{D}}$ and $\frac{\partial L}{\partial \boldsymbol{U}}$, as well as the design properties listed in Section 4.2.2, are inexpensively obtainable from some direct closed-form definition. If this is not the case, appropriate adjustments to the process must be made.

Two specific examples of such cases are discussed in this section. The first case occurs when it is desired to let the initial coordinates a certain set of nodes within the finite element mesh (usually those located on the boundary of the structure) be defined directly as design variables, and to let the remaining nodes deform smoothly in response to the resulting shape changes, without being directly associated with design variables themselves. This allows for the preservation of mesh quality throughout a process of changing design without defining complex interpolation functions or repeatedly re-generating the mesh.

The second case is when an objective function or constraint is a function of the eigenpairs of a matrix derived from the governing equations. Common objectives, derived from quantities like displacement, strain and stress can be evaluated directly throughout a structure from the finite element solution at a state, using the formulas developed in Chapter 3 and earlier sections of this chapter. But if the objective is derived from the eigenpairs of a solution-dependent matrix, there is no direct closed-form way of evaluating $\frac{\partial L}{\partial \boldsymbol{D}}$ and $\frac{\partial L}{\partial \boldsymbol{U}}$. An extension to the process is needed to generate the necessary terms in this case. The following sections explain the revised approach for these two cases.

4.3.1 Adaptive Mesh Deformation

When it is desired for the initial mesh of a finite element structure to morph in adaptation to the shifting of a set of *control nodes,* which are directly controlled by design variables, it becomes appropriate to pose the objective function in a slightly different manner. The idea is, as the coordinates of the control nodes change with design variables, the remaining nodes will reposition according to linear elastic analogy, and move as if the structure was being deformed under prescribed displacements on the control nodes. Generally, however, the deformation is not taken as governed by the structure's actual stiffness properties, but by a stiffness matrix constructed to produce smoothness and minimal distortion. Here, this matrix will be referred to as $[K_{MD}]$, so the dependence of the undeformed nodal coordinates of the structure $\boldsymbol{X}$ on the design variables can be expressed as

$$\begin{gathered}[K_{MD}](\boldsymbol{X}-\boldsymbol{X}_0)=\boldsymbol{X}_c(\boldsymbol{D})\\ \Rightarrow \boldsymbol{X}=[K_{MD}]^{-1}\boldsymbol{X}_c(\boldsymbol{D})+\boldsymbol{X}_0\\ \Rightarrow \frac{\partial \boldsymbol{X}}{\partial D_i}=[K^{MD}]^{-1}\frac{\partial \boldsymbol{X}_c}{\partial D_i}\end{gathered} \tag{4.37}$$

where $\boldsymbol{X}_0$ are the initial undeformed nodal coordinates of the structure, and $\boldsymbol{X}_c$ is a vector defining the coordinates of the control nodes as a function of $\boldsymbol{D}$. Now, if the objective function is defined as

$$L=L\left(\boldsymbol{X}(\boldsymbol{D}),\boldsymbol{U}\big(\boldsymbol{X}(\boldsymbol{D})\big)\right) \tag{4.38}$$

then the objective sensitivities can be expressed

$$\frac{dL}{dD_i} = \left(\frac{\partial L}{\partial \boldsymbol{X}}\right)\cdot\left(\frac{\partial \boldsymbol{X}}{\partial D_i}\right) + \left(\frac{\partial L}{\partial \boldsymbol{U}}\right)\cdot\left(\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{-1}\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right]\frac{\partial \boldsymbol{X}}{\partial D_i}\right) = \left(\frac{\partial L}{\partial \boldsymbol{X}} + \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right]^T\left(\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^T\right)^{-1}\frac{\partial L}{\partial \boldsymbol{U}}\right)\cdot\left([K^{MD}]^{-1}\frac{\partial \boldsymbol{X}_c}{\partial D_i}\right) \tag{4.39}$$

The procedure for evaluating the objective sensitivities in this case can then be outlined in four steps:

$$\begin{gathered}
(1)\ \ solve\ \ \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^T \boldsymbol{\Lambda} = \frac{\partial L}{\partial \boldsymbol{U}} \\
(2)\ \ evaluate\ \tilde{\boldsymbol{L}} = \frac{\partial L}{\partial \boldsymbol{X}} + \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right]^T \boldsymbol{\Lambda} \\
(3)\ solve\ [K_{MD}]^T \boldsymbol{\Lambda}_{MD} = \tilde{\boldsymbol{L}} \\
(4)\ for\ every\ D_i, evaluate\ \frac{dL}{dD_i} = \boldsymbol{\Lambda}_{MD}\cdot\left(\frac{\partial \boldsymbol{X}_c}{\partial D_i}\right)
\end{gathered} \tag{4.40}$$

With these modifications, the process requires two linear system solutions and two adjoint vectors, but is still independent of the number of design variables. Generally, the mesh deformation matrix $[K_{MD}]$ can be considered independent of design variables, and only needs to be constructed once for any number of design states. How exactly to define $[K_{MD}]$ is somewhat open-ended, but a typical way is to assume isotropic elements with stiffness inversely proportional to their volume. This is to prevent small elements from becoming excessively distorted and taking negative volume during deformation. The sensitivity of governing equations with respect to nodal coordinates $\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right]$ can be evaluated as described in section 4.2.2, taking the design variable to be the appropriate nodal coordinate. With this method, adaptive mesh deformation can be implemented while still preserving the efficiency of the adjoint method.

4.3.2 Eigenpair-Based Objectives and Constraints

It can be useful to define objective functions and/or constraints involving the eigenpairs of a matrix derived from the governing equations. The most prominent example in the present context is in structural buckling analysis, as will be discussed further in Chapter 6. Another potential application would be in vibrations and modal analysis, in the tailoring of natural frequencies of a structure to resist phenomena like aerodynamic flutter.

For now, let us just assume there is a matrix $[K]$, which is a function of the design variables $\boldsymbol{D}$, and the solution variables $\boldsymbol{U}$ of a structural system. Let the objective be some function of the eigenvalues $\boldsymbol{\lambda}$ and eigenvectors $[V]$ of $[K]$:

$$L = L([V], \boldsymbol{\lambda}) \tag{4.41}$$

Often, the objective may only be dependent on a certain set of the eigenpairs, not necessarily the entire set. The objective sensitivities can be expressed

$$\frac{dL}{dD_i} = \sum_{j=1}^{n_{pairs}} \left(\frac{\partial L}{\partial \boldsymbol{v}_j}\right) \cdot \left(\frac{d\boldsymbol{v}_j}{dD_i}\right) + \frac{\partial L}{\partial \lambda_j}\frac{d\lambda_j}{dD_i} \tag{4.42}$$

To construct the sensitivities with respect to a given eigenpair, begin with the fundamental eigenproblem statement,

$$[K]\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j \tag{4.43}$$

Remembering that $[K]$ is dependent on $\boldsymbol{D}$, and by extension so are the eigenpairs, differentiation of Eq. (4.43) gives

$$\frac{d[K]}{dD_i}\boldsymbol{v}_j + [K]\frac{d\boldsymbol{v}_j}{dD_i} = \frac{d\lambda_j}{dD_i}\boldsymbol{v}_j + \lambda_j \frac{d\boldsymbol{v}_j}{dD_i} \tag{4.44}$$

If a magnitude of one is enforced on all eigenvectors, the following constraint also applies:

$$\begin{gathered} \boldsymbol{v}_j \cdot \boldsymbol{v}_j = 1 \\ \Rightarrow \boldsymbol{v}_j \cdot \left(\frac{d\boldsymbol{v}_j}{dD_i}\right) = 0 \end{gathered} \tag{4.45}$$

Rearranging Eq. (4.44) and combining with Eq. (4.45) gives

$$\begin{gathered}[K]\frac{d\boldsymbol{v}_j}{dD_i}-\lambda_j\frac{d\boldsymbol{v}_j}{dD_i}-\frac{d\lambda_j}{dD_i}\boldsymbol{v}_j=-\frac{d[K]}{dD_i}\boldsymbol{v}_j\\ \boldsymbol{v}_j\cdot\left(\frac{d\boldsymbol{v}_j}{dD_i}\right)=0\end{gathered}\tag{4.46}$$

Or in matrix form, after a sign reversal of the eigenvector constraint, Eq. (4.46) can be expressed

$$\begin{bmatrix}[K]-\lambda_j[I] & -\boldsymbol{v}_j\\ -\boldsymbol{v}_j^T & 0\end{bmatrix}\begin{Bmatrix}\dfrac{d\boldsymbol{v}_j}{dD_i}\\ \dfrac{d\lambda_j}{dD_i}\end{Bmatrix}=\begin{Bmatrix}-\dfrac{d[K]}{dD_i}\boldsymbol{v}_j\\ 0\end{Bmatrix}\tag{4.47}$$

With this relationship, the objective sensitivities can now be written as

$$\begin{aligned}\frac{dL}{dD_i}&=\sum_{j=1}^{n_{pairs}}\left[\left(\frac{\partial L}{\partial\boldsymbol{v}_j}\right)^T\quad\frac{\partial L}{\partial\lambda_j}\right]\begin{Bmatrix}\dfrac{d\boldsymbol{v}_j}{dD_i}\\ \dfrac{d\lambda_j}{dD_i}\end{Bmatrix}\\ &=\sum_{j=1}^{n_{pairs}}\left[\left(\frac{\partial L}{\partial\boldsymbol{v}_j}\right)^T\quad\frac{\partial L}{\partial\lambda_j}\right]\begin{bmatrix}[K]-\lambda_j[I] & -\boldsymbol{v}_j\\ -\boldsymbol{v}_j^T & 0\end{bmatrix}^{-1}\begin{Bmatrix}-\dfrac{d[K]}{dD_i}\boldsymbol{v}_j\\ 0\end{Bmatrix}\end{aligned}\tag{4.48}$$

The total sensitivity of the matrix $[K]$, assuming dependence on both $\boldsymbol{D}$ and $\boldsymbol{U}$, is

$$\frac{d[K]}{dD_i}=\frac{\partial[K]}{\partial D_i}+\sum_{k=1}^{n_{dof}}\frac{\partial[K]}{\partial U_k}\left(\left[\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{U}}\right]^{-1}\frac{\partial\boldsymbol{R}}{\partial D_i}\right)_k\tag{4.49}$$

Putting it all together, the process of obtaining the objective sensitivities using and adjoint approach in this case can be broken down into the following steps:

$$
\begin{aligned}
&(1)\ find\ the\ necessary\ eigenpairs\ of\ [K]\colon \boldsymbol{\lambda}, [V]\\
&(2)\ for\ every\ eigenpair\colon\\
&\qquad (a)\ solve\ \begin{bmatrix} [K]-\lambda_j[I] & -\boldsymbol{v}_j \\ -\boldsymbol{v}_j^T & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_v \\ \boldsymbol{\Lambda}_\lambda \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial L}{\partial \boldsymbol{v}_j} \\ \dfrac{\partial L}{\partial \boldsymbol{\lambda}_j} \end{Bmatrix}\\
&\qquad (b)\ evaluate\ \boldsymbol{y}, y_k = \ (\boldsymbol{\Lambda}_v)^T \frac{\partial [K]}{\partial U_k} \boldsymbol{v}_i \ \ , k = 1 \ldots n_{dof}\\
&\qquad (c)\ solve\ \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^T \boldsymbol{\Lambda} = \ \boldsymbol{y}\\
&\qquad (d)\ for\ every\ D_i\ update \frac{dL}{dD_i} = \frac{dL}{dD_i} - \ (\boldsymbol{\Lambda}_v)^T \frac{\partial [K]}{\partial D_i} \boldsymbol{v}_j - \boldsymbol{\Lambda}^T \frac{\partial \boldsymbol{R}}{\partial D_i}
\end{aligned}
\tag{4.50}
$$

The modified adjoint process in this case is still independent of the number of design variables, but requires an eigen-solve, and two solutions of linear systems for each of the resulting eigenpairs. Clearly, this implementation could still become costly if a large number of eigenpairs are needed, despite the use of the adjoint. For large systems, it is usually only a small subset of eigenpairs that are needed for the analysis, but finding potential alternative approaches to the analysis may be worthwhile nonetheless. This is a topic of study in Chapter 6.

Chapter 5

Demonstrations and Validations

With the methods and formulations of AStrO established, a series of tests and demonstrations will be presented to validate and confirm the reliability of its results. Each section of this chapter focuses on a particular aspect of AStrO's capability, comparing results with analytical solutions, experimental data and other numerical tools. The accurate and reliable computation of objective sensitivities using the adjoint method will also be verified. The specific topics of focus for each section are 1) static coupled thermoelastic modeling, 2) problems involving nonlinear geometry 3) structural dynamic modeling and 4) coupled aeroelastic modeling.

5.1 Static Thermoelastic Response

This section provides a general demonstration of the coupled thermoelastic modeling capability of AStrO. To begin, recall the governing partial differential equations for the thermal and elastic response of solid bodies in static equilibrium, as originally given in Chapter 3:

$$\begin{gathered}(\boldsymbol{\nabla}\cdot\boldsymbol{q}) - Q = 0\\ \frac{\partial\sigma_{11}}{\partial x_1}+\frac{\partial\sigma_{12}}{\partial x_2}+\frac{\partial\sigma_{13}}{\partial x_3}+f_1=0\\ \frac{\partial\sigma_{21}}{\partial x_1}+\frac{\partial\sigma_{22}}{\partial x_2}+\frac{\partial\sigma_{23}}{\partial x_3}+f_2=0\\ \frac{\partial\sigma_{31}}{\partial x_1}+\frac{\partial\sigma_{32}}{\partial x_2}+\frac{\partial\sigma_{33}}{\partial x_3}+f_3=0\end{gathered} \tag{5.1}$$

Now suppose an elastic body is subject to thermal loads in the form of internal heat generation, represented by Q, surface traction $\boldsymbol{t}$ and heat flux $\boldsymbol{q}$ in the absence of any volumetric body force. In that case, the governing equations for elastic response simplifies to

$$\begin{gathered}\frac{\partial \sigma_{11}}{\partial x_1}+\frac{\partial \sigma_{12}}{\partial x_2}+\frac{\partial \sigma_{13}}{\partial x_3}=0 \\ \frac{\partial \sigma_{21}}{\partial x_1}+\frac{\partial \sigma_{22}}{\partial x_2}+\frac{\partial \sigma_{23}}{\partial x_3}=0 \\ \frac{\partial \sigma_{31}}{\partial x_1}+\frac{\partial \sigma_{32}}{\partial x_2}+\frac{\partial \sigma_{33}}{\partial x_3}=0\end{gathered} \tag{5.2}$$

With stress computed from the combination of total strain and strain due to thermal expansion:

$$\sigma_{ij}=\sum_{k=1}^{3}\sum_{l=1}^{3} C_{ijkl}(\epsilon_{total,kl}-\alpha_{TE,kl}T) \tag{5.3}$$

If the elastic stiffness tensor and the thermal expansion coefficients are constant throughout the elastic body, then Eq. (5.2) becomes

$$\begin{gathered}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} C_{1jkl}\left(\frac{\partial \epsilon_{total,kl}}{\partial x_j}-\alpha_{TE,kl}\frac{\partial T}{\partial x_j}\right)=0 \\ \sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} C_{2jkl}\left(\frac{\partial \epsilon_{total,kl}}{\partial x_j}-\alpha_{TE,kl}\frac{\partial T}{\partial x_j}\right)=0 \\ \sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} C_{3jkl}\left(\frac{\partial \epsilon_{total,kl}}{\partial x_j}-\alpha_{TE,kl}\frac{\partial T}{\partial x_j}\right)=0\end{gathered} \tag{5.4}$$

Equation (5.4) provides a relationship between strain and temperature in the absence of applied volumetric body forces. The strategy of this test is to choose an analytical displacement/strain field within an arbitrary elastic body, and derive the corresponding temperature distribution based on Eq. (5.4). Then the internal heat generation field can be found from Eq. (5.1), and the derived loading and boundary conditions can be applied to a finite element model to compare the results for temperature and displacement generated by AStrO to the analytical solutions.

Consider a solid cube of dimension 1 with its lower corner at the origin of a global coordinate system, as shown in Fig. 5.1.

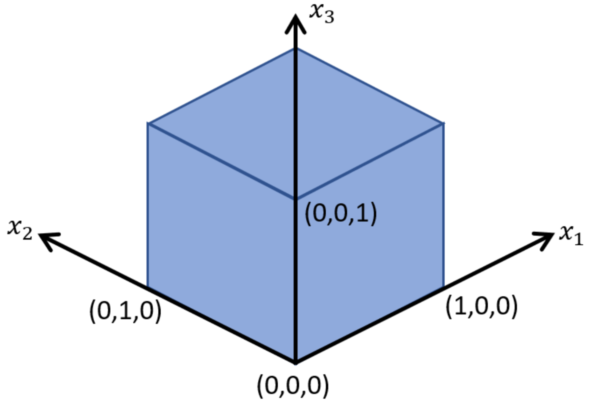


Figure 5.1 Geometric definition of thermally loaded solid block.

Let the analytical displacement field within the cube be given as follows:

$$u_1 = \left(\frac{1}{3}x_1^3 + x_2^2 + x_3^2\right)10^{-3}$$
$$u_2 = \left(x_1^2 + \frac{1}{3}x_2^3 + x_3^2\right)10^{-3} \tag{5.5}$$
$$u_3 = \left(x_1^2 + x_2^2 + \frac{1}{3}x_3^3\right)10^{-3}$$

In this exercise, linear geometry is assumed so that the total strain field can be expressed

$$\epsilon_1 = \frac{\partial u_1}{\partial x_1} = x_1^2 10^{-3}$$
$$\epsilon_2 = \frac{\partial u_2}{\partial x_2} = x_2^2 10^{-3}$$
$$\epsilon_3 = \frac{\partial u_3}{\partial x_3} = x_3^2 10^{-3}$$
$$\gamma_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} = 2(x_1 + x_2)10^{-3} \tag{5.6}$$
$$\gamma_{13} = \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} = 2(x_1 + x_3)10^{-3}$$
$$\gamma_{23} = \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} = 2(x_2 + x_3)10^{-3}$$

A further assumption is that the cube's material has orthotropic elastic properties, with its orthogonal planes of symmetry aligned with the global coordinate directions. Its thermal conductivity k is isotropic, as is the coefficient of thermal expansion α_{TE}, with no expansions directly associated with shear strains. Under these assumptions, Eq. (5.4) can be expanded out for each of the global coordinate directions, considering only non-zero terms of the summation. For the equilibrium in the 1-direction,

$$\begin{gathered} C_{1111}\left(\frac{\partial \epsilon_{11}}{\partial x_1} - \alpha^{TE}\frac{\partial T}{\partial x_1}\right) + C_{1122}\left(\frac{\partial \epsilon_{22}}{\partial x_1} - \alpha^{TE}\frac{\partial T}{\partial x_1}\right) + C_{1133}\left(\frac{\partial \epsilon_{33}}{\partial x_1} - \alpha^{TE}\frac{\partial T}{\partial x_1}\right) \\ + C_{1212}\left(\frac{\partial \epsilon_{12}}{\partial x_2}\right) + C_{1221}\left(\frac{\partial \epsilon_{21}}{\partial x_2}\right) + C_{1313}\left(\frac{\partial \epsilon_{13}}{\partial x_3}\right) + C_{1331}\left(\frac{\partial \epsilon_{31}}{\partial x_3}\right) = 0 \end{gathered} \tag{5.7}$$

Rewriting Eq. (5.7) in terms of strain in one-dimensional vector form,

$$\begin{gathered} C_{11}\left(\frac{\partial \epsilon_{1}}{\partial x_1} - \alpha_{TE}\frac{\partial T}{\partial x_1}\right) + C_{12}\left(\frac{\partial \epsilon_{2}}{\partial x_1} - \alpha_{TE}\frac{\partial T}{\partial x_1}\right) + C_{13}\left(\frac{\partial \epsilon_{3}}{\partial x_1} - \alpha_{TE}\frac{\partial T}{\partial x_1}\right) \\ + C_{44}\frac{\partial \gamma_{12}}{\partial x_2} + C_{55}\frac{\partial \gamma_{13}}{\partial x_3} = 0 \end{gathered} \tag{5.8}$$

Substituting the analytical solution for total strain into Eq. (5.8), the equation simplifies to

$$\begin{gathered} C_{11}\left(2x_1 10^{-3} - \alpha_{TE}\frac{\partial T}{\partial x_1}\right) + C_{12}\left(-\alpha_{TE}\frac{\partial T}{\partial x_1}\right) + C_{13}\left(-\alpha_{TE}\frac{\partial T}{\partial x_1}\right) \\ +2C_{44}10^{-3} + 2C_{55}10^{-3} = 0 \end{gathered} \tag{5.9}$$

The partial derivative of temperature with respect to x_1 can now be solved for as follows:

$$\frac{\partial T}{\partial x_1} = \frac{2(C_{11}x_1 + C_{44} + C_{55})10^{-3}}{\alpha_{TE}(C_{11} + C_{12} + C_{13})} \tag{5.10}$$

Equation (5.10) can be integrated in x_1 to give a partial representation of the temperature field:

$$T = \frac{2\left(\frac{1}{2}C_{11}x_1 + C_{44} + C_{55}\right)x_1 10^{-3}}{\alpha_{TE}(C_{11} + C_{12} + C_{13})} + F_1(x_2, x_3) \tag{5.11}$$

The same process can be followed for the equations of elastic equilibrium in the 2 and 3 directions, leading to the following supplemental representations of the temperature field:

$$\begin{aligned} T &= \frac{2\left(\frac{1}{2}C_{22}x_2 + C_{44} + C_{66}\right)x_2 10^{-3}}{\alpha_{TE}(C_{21} + C_{22} + C_{23})} + F_2(x_1, x_3) \\ T &= \frac{2\left(\frac{1}{2}C_{33}x_3 + C_{55} + C_{66}\right)x_3 10^{-3}}{\alpha_{TE}(C_{31} + C_{32} + C_{33})} + F_3(x_1, x_2) \end{aligned} \tag{5.12}$$

Combining Eqs. (5.11) and (5.12), the complete definition of the temperature field can be expressed

$$T = \frac{2}{\alpha_{TE}}\left(\frac{\left(\frac{1}{2}C_{11}x_1 + C_{44} + C_{55}\right)x_1}{(C_{11} + C_{12} + C_{13})} + \frac{\left(\frac{1}{2}C_{22}x_2 + C_{44} + C_{66}\right)x_2}{(C_{21} + C_{22} + C_{23})} + \frac{\left(\frac{1}{2}C_{33}x_3 + C_{55} + C_{66}\right)x_3}{(C_{31} + C_{32} + C_{33})}\right)10^{-3} \tag{5.13}$$

With isotropic thermal conductivity, the heat flux field can be written

$$\boldsymbol{q} = -k\boldsymbol{\nabla}T = -\frac{2k}{\alpha_{TE}}\left[\frac{(C_{11}x_1 + C_{44} + C_{55})}{(C_{11} + C_{12} + C_{13})}, \frac{(C_{22}x_2 + C_{44} + C_{66})}{(C_{21} + C_{22} + C_{23})}, \frac{(C_{33}x_3 + C_{55} + C_{66})}{(C_{31} + C_{32} + C_{33})}\right]^T 10^{-3} \tag{5.14}$$

and the internal heat generation field is

$$Q = (\boldsymbol{\nabla} \cdot \boldsymbol{q}) = -\frac{2k}{\alpha_{TE}}\left(\frac{C_{11}}{(C_{11} + C_{12} + C_{13})} + \frac{C_{22}}{(C_{21} + C_{22} + C_{23})} + \frac{C_{33}}{(C_{31} + C_{32} + C_{33})}\right)10^{-3} \tag{5.15}$$

By applying Eqs. (5.14) and (5.15) as thermal surface flux and heat generation loads to a finite element model of the solid cube, the analytical temperature distribution of Eq. (5.13), and subsequently the original displacement and strain fields of Eqs. (5.5) and (5.6) should be recovered.

Models of the solid cube were constructed using four-node tetrahedral elements and eight-node hexahedral elements with and without incompatible modes at four levels of mesh refinement. The nominal element size dX at each progressive refinement level was set to 1/4, 1/8, 1/16, and 1/32. The elastic material properties of the cube were set to be fully isotropic, with Young's modulus 10^6 and Poisson's ratio 0.3. Thermal conductivity and coefficient of thermal expansion were set to 1.0 and 10^{-4} respectively. Figures 5.2 and 5.3 show plots of the resulting temperature and displacement distribution under the specified loading computed by AStrO, for refinement level three.

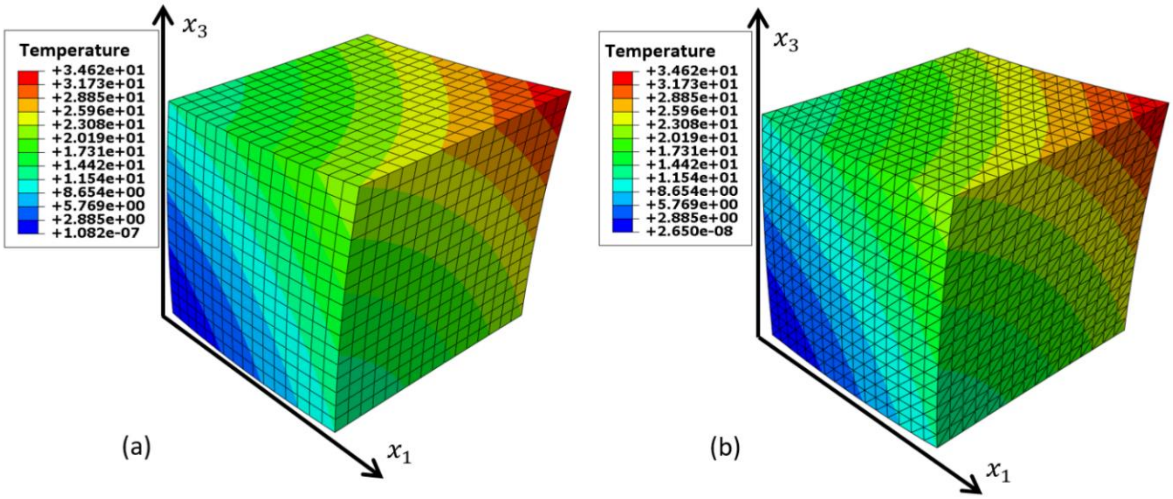


Figure 5.2 Temperature distribution in solid block for (a) hexahedral elements and (b) tetrahedral elements.

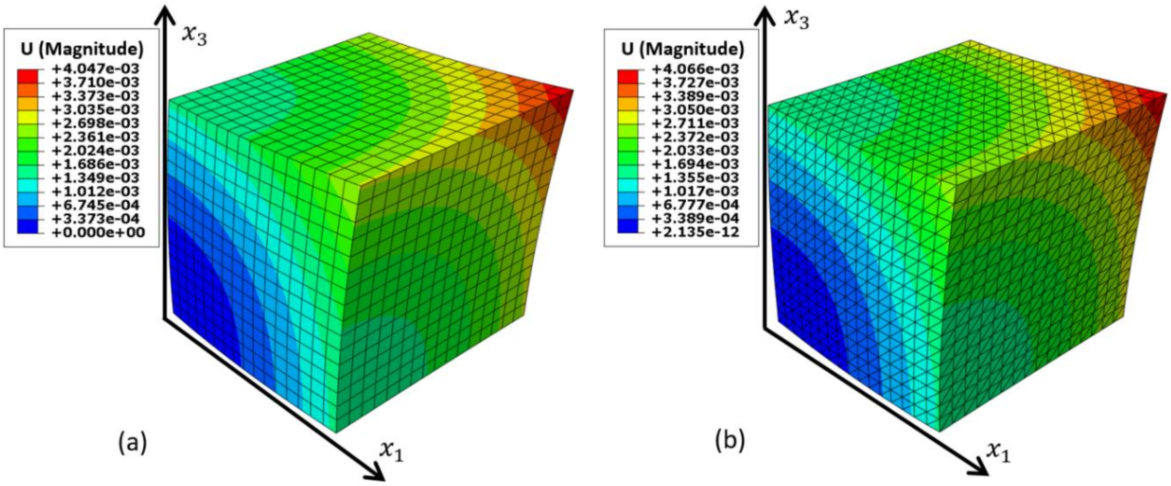


Figure 5.3 Distribution of displacement magnitude in deformed solid block for (a) hexahedral elements and (b) tetrahedral elements.

The convergence of the solution with mesh refinement is demonstrated in Figs. 5.4 through 5.6. For each element type, the error norm and the root mean square of error in temperature, displacement and strain are plotted for increasing mesh refinement, computed as follows

$$error = \frac{\sum_{i=1}^{n_{els}} \left| u_{FE,i} - u_{exact,i} \right| (Vol_i)}{Total\,Volume}$$
$$RMS(error) = \sqrt{\frac{\sum_{i=1}^{n_{els}} \left(u_{FE,i} - u_{exact,i} \right)^2 (Vol_i)}{Total\,Volume}} \tag{5.16}$$

with all quantities computed at the centroid of each element. All solutions asymptotically converge to the exact solution with increasing mesh refinement at a logarithmic slope of roughly two, which is to be expected for linear order elements. Eight-node hexahedral elements generally perform better than four-node tetrahedral elements at a given level of refinement, particularly for the elastic response. This is known to be true in general, since four-node linear interpolation results in unconditionally constant solution gradients within an element, meaning constant strain or constant heat flux. Yet the difference tends to be less pronounced for the thermal solution. Hex elements with incompatible modes tend to show improved performance compared to those without for displacement and strain results, due to their improved compliance and enriched basis.

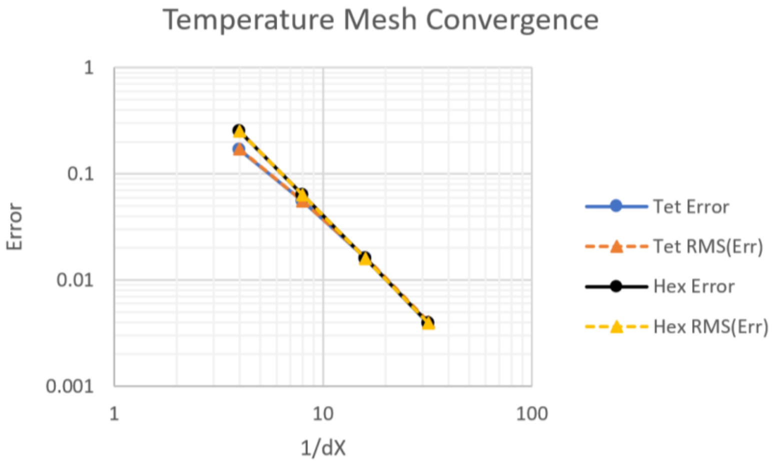


Figure 5.4 Error convergence of temperature solution in solid block.

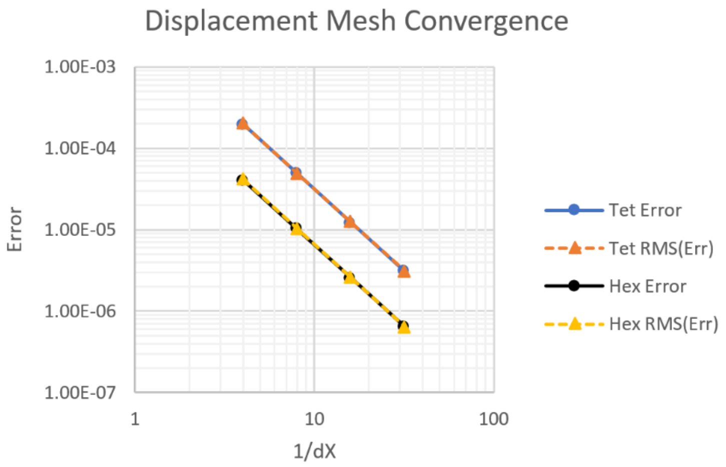


Figure 5.5 Error convergence of displacement solution in solid block.

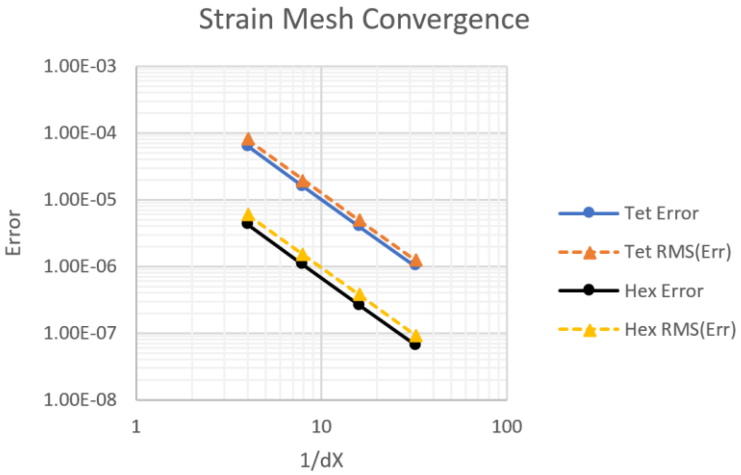


Figure 5.6 Error convergence of strain solution in solid block.

To test the adjoint-based sensitivities in this case, the elastic modulus, thermal conductivity and coefficient of thermal expansion of the block material were defined as design variables. The sensitivity of each property was scaled to the original value, resulting in the following design-dependent definitions:

$$\begin{aligned} E &= E_0 + 10^6 D_1 \\ k &= k_0 + D_2 \\ \alpha_{TE} &= \alpha_{TE0} + 10^{-4} D_3 \end{aligned} \tag{5.17}$$

The normal stress in the 1-direction at the center of the block was chosen as a sample objective function. The sensitivities of the objective were computed using both the adjoint and tangent methods, as well as with complex differentiation for comparison. Tables 5.1 through 5.3 show the results for the models of each element type, at the second level of refinement. The adjoint and tangent formulations are mathematically equivalent, using exact differentiation of the governing equations as described in Chapter 4, and they agree nearly to machine precision. Complex differentiation does not use the linearization of the governing equations, but works much like a high-precision finite difference, and should be numerically equal to the other two results, as seen in the tables. The results indicate that the adjoint implementation correctly differentiates the finite element solution in this case.

Table 5.1 Comparison of objective sensitivities for hex elements.

	Adjoint	**Tangent**	**Complex**
E	1.016231638217**54**E+02	1.016231638217**40**E+02	1.016231638217**49**E+02
k	-3.19733746691863E+01	-3.19733746691863E+01	-3.19733746691863E+01
α^{TE}	4.591523458546**06**E+01	4.591523458546**10**E+01	4.591523458546**07**E+01

Table 5.2 Comparison of objective sensitivities for hex elements with incompatible modes.

	Adjoint	**Tangent**	**Complex**
E	1.020040590717**04**E+02	1.020040590717**15**E+02	1.020040590717**40**E+02
k	-3.23030013695860E+01	-3.23030013695860E+01	-3.23030013695860E+01
α^{TE}	4.396485644427**60**E+01	4.396485644427**94**E+01	4.396485644427**69**E+01

Table 5.3 Comparison of objective sensitivities for tet elements.

	Adjoint	**Tangent**	**Complex**
E	4.421185706005**18**E+02	4.421185706005**08**E+02	4.421185706005**73**E+02
k	-4.0243937862772**8**E+01	-4.0243937862772**7**E+01	-4.024393786277**40**E+01
α^{TE}	2.6201358577098**4**E+02	2.6201358577098**7**E+02	2.6201358577098**8**E+02

5.2 Nonlinear Geometry

This section provides a demonstration of AStrO's capability for modeling geometrically nonlinear structural problems. The test model is a straight wing of length L = 10 and chord c = 1, with a NACA 0012 cross section, constructed from both solid continuum elements and shell elements, as shown in Figure 5.7.

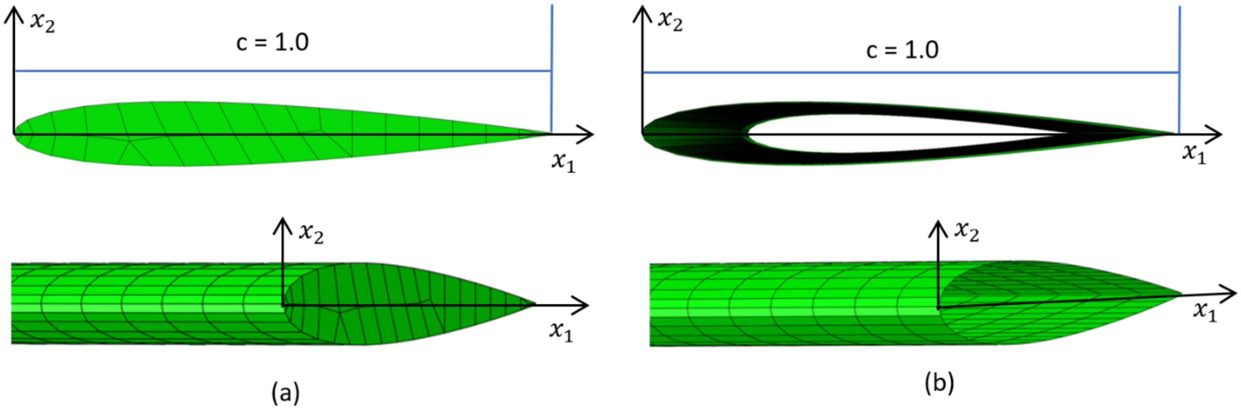


Figure 5.7 NACA 0012 wing section models constructed from (a) solid elements and (b) shell elements.

The solid element model was constructed to be solid through the thickness, dominated by eight-node hex elements with sparing use of six-node wedge elements. The shell model's cross section was made hollow with a shell thickness of 0.025. The wing material was given a thermal expansion coefficient of $\alpha_{TE} = 10^{-4}$, and a temperature gradient was applied through the cross section of the wing in the 2-direction, such that

$$T = -\frac{\pi}{\alpha_{TE} L} x_2 \tag{5.18}$$

This condition results in a normal strain proportional to the distance from the neutral axis of the wing, making the wing curl into an arc about the chord direction. The exact analytical response would form a 180° arc in the x_2-x_3 plane.

The displacement solution for both models was found with the $x_3 = 0$ end of the wing clamped with displacements and rotations set to zero. Analysis was performed using full Newton-

Raphson iterations, with the thermal load applied in successive increasing increments to ensure convergence, at four increasing levels of mesh refinement for both models. Nominal element sizes for each refinement level were 1/4, 1/8, 1/16, and 1/32. Figure 5.8 graphically shows the response for both the solid and shell models, for an element size of 1/16. Figure 5.9 plots the mesh convergence of both models, with error and root mean square as defined previously in Eq. (5.16).

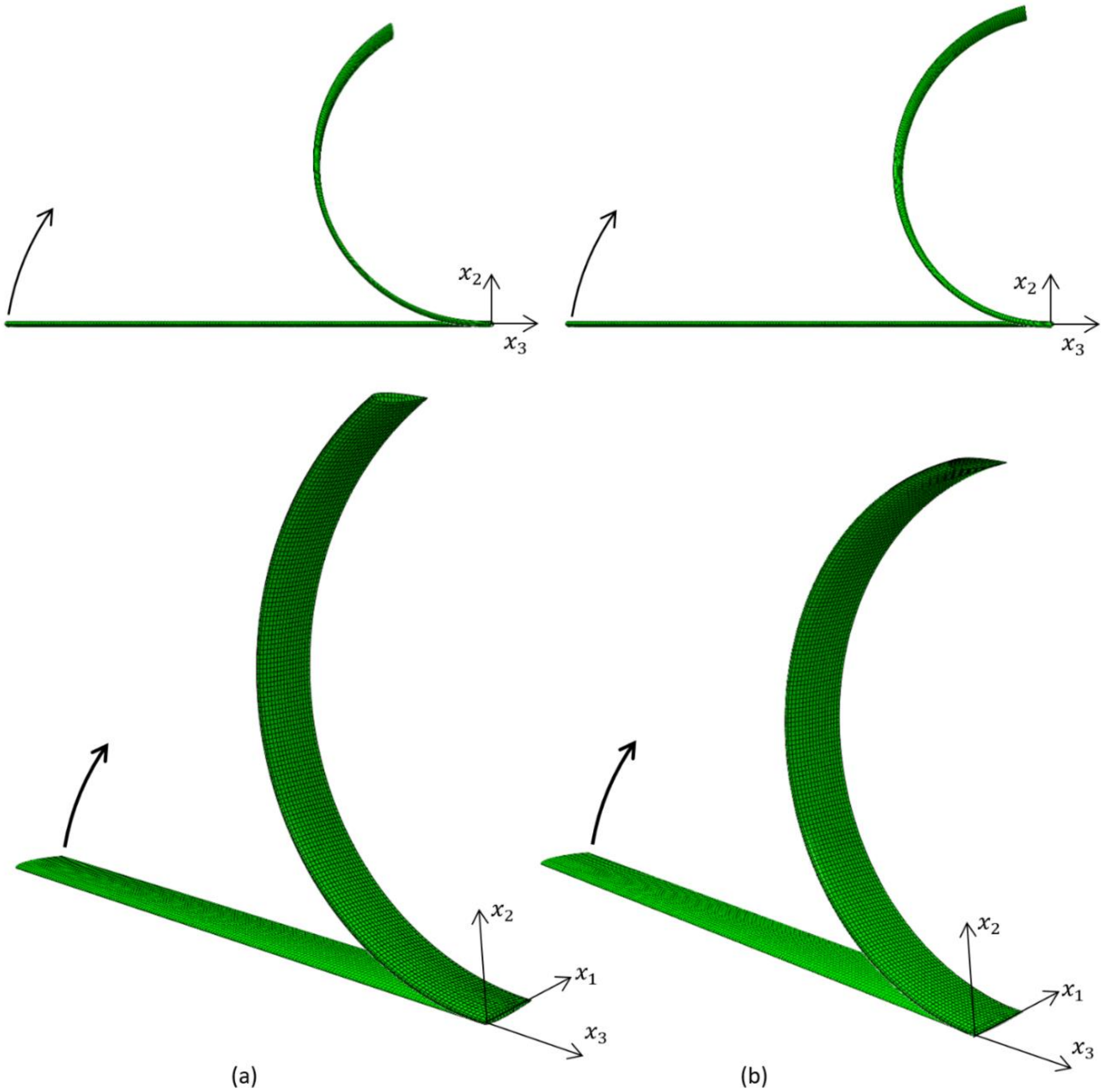


Figure 5.8 Deformed configuration of NACA 0012 wing under thermal loading for (a) solid model and (b) shell model.

These two models, although identical in outer mold line and length, are fundamentally different not only in element type, but also in cross sectional properties, one being solid and one being a hollow shell. As a result, some key differences are seen in the mesh convergence behavior. For the solid model, the mesh is three dimensional throughout the volume, with elements conforming to the curved shape of the airfoil. Consequently, there is a measure of error due to element distortion in the 1-2 plane. This is not a consideration for the shell model, since only a single layer of elements forming the outer surface is required, which can be structured in a very regular arrangement. On a related note, refinement of the solid mesh entails refining element size in all three dimensions, while shell structures are inherently one element thick, and so refinement is effectively in only two dimensions. Yet shell elements are somewhat more ideally formulated for slender structures such as this in general.

One consequence of these differences is that the solution for the solid model shows greater error from the exact solution than does the shell model, particularly for coarse levels of refinement. Figure 5.9 shows the mesh convergence for both models, with error computed as shown in Eq. (5.16). From the coarsest mesh to the next level up, both models seem to exhibit first order error convergence. But toward high levels of mesh refinement, the solid model approaches second order error convergence while the shell model remains steadily at first order convergence. This can be explained by the fact that second order convergence is predicated on the assumption of mesh refinement in all three dimensions. This cannot apply to the shell model at all as previously mentioned. For the solid model, because the thickness in the 2-direction is so small and the first mesh is so coarse, almost no refinement takes place in the 2-direction between the first two levels of mesh refinement, leading to the apparent first order convergence on that step.

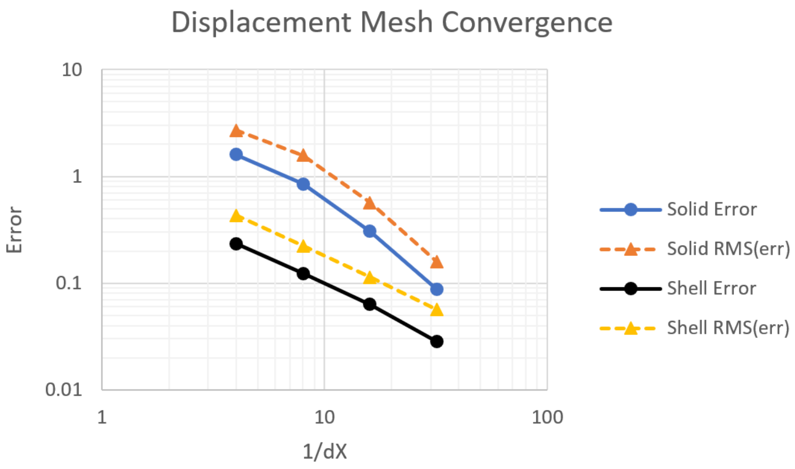


Figure 5.9 Error convergence of displacement for NACA 0012 wing models under thermal loading.

To examine sensitivities, the thermal expansion coefficient and the x_3 nodal coordinates of the models were set as design variables, given by

$$\begin{aligned} \alpha_{TE} &= \alpha_{TE,0}(1 + D_1) \\ X_3 &= X_{3,0}(1 + D_2) \end{aligned} \tag{5.19}$$

The magnitude of the tip deflection of the wing was chosen to be a test objective function, and the sensitivities were obtained using the adjoint, the tangent method and complex differentiation. Table 5.4 and Table 5.5 give the sensitivities for the solid model and the shell model, respectively. Again, the three measures of sensitivity should be exactly equivalent, but this time a bit more discrepancy is seen in the last few digits compared to the previous study. It is typical for structures that are inherently unstable, and sensitive to small perturbations in loading to produce more uncertainty in the solution, and by extension in the adjoint as well. The long, slender geometry and clamped boundary condition of these models make them an example of such cases. All in all, the results show accuracy in the geometrically nonlinear formulation and the adjoint-based sensitivities.

Table 5.4 Comparison of objective sensitivities for solid element model.

	Adjoint	**Tangent**	**Complex**
α_{TE}	4.90118414963**794**E+00	4.90118414963**837**E+00	4.90118414963**117**E+00
X_3	9.78884552153**048**E+00	9.78884552153**156**E+00	9.78884552153**689**E+00

Table 5.5 Comparison of objective sensitivities for shell model.

	Adjoint	**Tangent**	**Complex**
α_{TE}	4.77792761702**088**E+00	4.7779276170**0899**E+00	4.7779276170**6767**E+00
X_3	9.6772237349**8791**E+00	9.6772237349**7570**E+00	9.6772237349**1387**E+00

5.5 Dynamic Modeling

To demonstrate AStrO's capability for implicit dynamic time integration, consider again the NACA 0012 wing section models introduced in the previous section. With their constant symmetric cross sections and material properties, these models are essentially uniform beams, for which there are analytical solutions for the modes of free vibration subject to various boundary conditions. In this demonstration, the NACA models are clamped (constrained to zero displacements and rotations) at the end where $x_3 = 0$, and free at the opposite end where $x_3 = -L$. For such conditions, the first mode response of transverse free vibration can be represented as

$$\begin{gathered} u_2 = \psi(x_3)\varphi(t) \\ \psi(x_3) = \cosh(bx_3) - \cos(bx_3) - a(\sinh(bx_3) - \sin(bx_3)) \end{gathered} \tag{5.20}$$

Where $\varphi(t)$ is a periodic harmonic function of time, determined from the initial conditions and loading, as well as the material and geometric properties of the beam. The constants a and b are determined from the particular mode of vibration. In this case, for the first mode of vibration of a clamped-free beam extruded in the negative 3-direction, $a = 0.7340955$, and $b = -0.18751041$. The NACA 0012 wing models were subjected to a transverse load distribution proportional to the mode shape ψ from rest, as illustrated in Fig. 5.10. Under these conditions, the periodic function φ becomes

$$\begin{gathered} \varphi(t) = 1 - \cos(\omega t) \\ \omega = b^2\sqrt{\frac{EI}{\rho A}} \end{gathered} \tag{5.21}$$

The natural frequency ω can be determined as shown in Eq. (5.21) from the Young's modulus of the beam E, the second moment of area of the cross section I, the mass density ρ and the cross-sectional area A, along with the constant b for the given mode shape. The response for one full period of vibration was predicted by AStrO through implicit time integration for both the solid model and the shell model at the highest level of mesh refinement, and compared with the analytical solution for progressively increasing time resolution. The dynamic integration parameters were set to $\alpha = 0$, $\beta = \frac{1}{4}$, $\gamma = \frac{1}{2}$, for which second-order convergence in time is expected. Figure 5.11 shows the time resolution convergence for both models, again with the metrics of error defined as shown in Eq. (5.16). Second-order convergence is confirmed with both models, evident by the logarithmic slope of two.

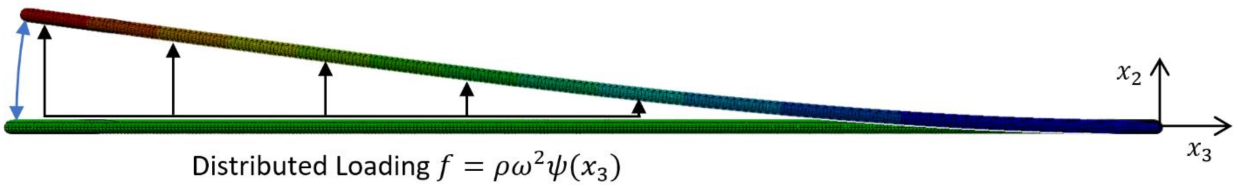


Figure 5.10 NACA 0012 wing model in first mode free vibration.

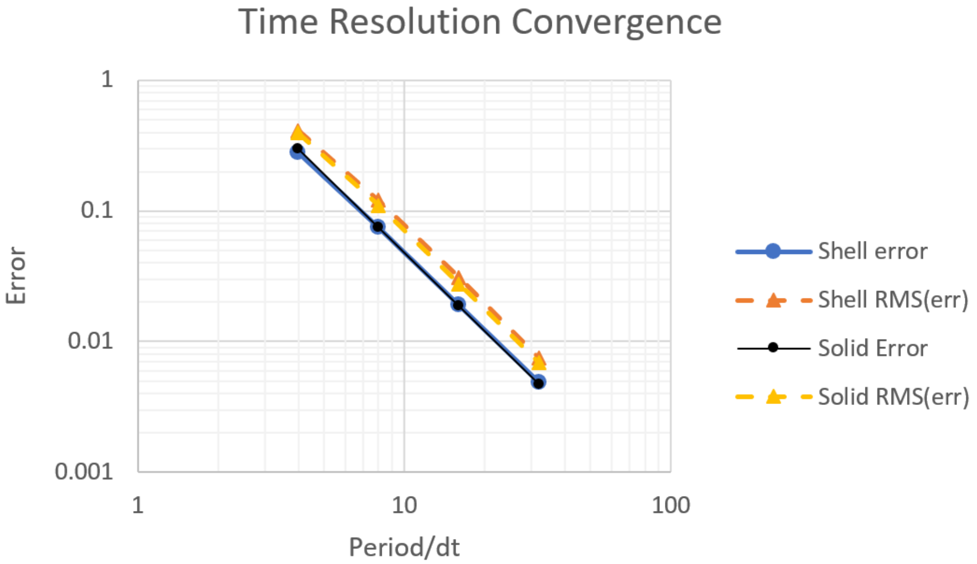


Figure 5.11 Time resolution convergence of error for NACA 0012 wing model in first mode free vibration.

As was done in previous demonstrations, the solution convergence test was followed up with a sensitivity test for selected design variables. In this case, the objective was set to be the tip deflection of the wing integrated over one period of vibration. Design variables were defined to be Young's modulus E, mass density ρ, nodal coordinates in the 3-direction X_3, and shell thickness

h, if applicable. Tables 5.6 and 5.7 show the objective sensitivities as obtained by the adjoint, tangent and complex step methods.

Table 5.6 Sensitivity results for solid NACA 0012 model in free vibration.

	Adjoint	**Tangent**	**Complex**
E	-2.343167976856**26**E+01	-2.343167976856**09**E+01	-2.343167976856**65**E+01
ρ	6.93622330325**826**E+00	6.93622330325**728**E+00	6.93622330325**849**E+00
X_3	7.69163259198**577**E+01	7.69163259198**956**E+01	7.69163259198**118**E+01

Table 5.7 Sensitivity results for shell NACA 0012 model in free vibration.

	Adjoint	**Tangent**	**Complex**
E	-2.28662547511 17**4**E+01	-2.2866254751117**6**E+01	-2.2866254751117**2**E+01
ρ	6.8090933737965**2**E+00	6.8090933737965**6**E+00	6.8090933737965**9**E+00
h	-6.879384994999**07**E-02	-6.879384994999**19**E-02	-6.879384994999**10**E-02
X_3	7.112018958 4**2075**E+01	7.1120189584**3516**E+01	7.1120189584**7536**E+01

As a further demonstration, consider a model of a wing section that was the subject of the High Reynolds Number Aero-Structural Dynamics (HIRENASD) aeroelastic test case [85]. For this wing geometry, numerous meshes for both CFD and structural finite element modeling, as well as experimental wind tunnel test data is readily available. In this section, the focus is on the pure structural dynamic analysis of a certain finite element model of the HIRENASD wing, and the prediction of its motion in first-mode free vibration.

The particular finite element model chosen is composed of approximately 42,000 eight-node hexahedral elements with incompatible modes, obtained from the website of the Aeroelastic Prediction Workshop (AePW) [86]. As a first step, a modal analysis was performed on the finite element model using Abaqus to obtain the first free-vibration mode shape of the structure. The results reported the natural frequency of the first mode to be 26.55 Hz, which was very consistent with the experimental value of 26.53 Hz reported on the AePW website. Taking this natural frequency with the first mode shape function $\boldsymbol{\psi}(\boldsymbol{x})$, the dynamic solution for the displacement of the wing in free vibration, with zero initial displacement should be

$$\boldsymbol{u}(\boldsymbol{x},t) = a\boldsymbol{\psi}(\boldsymbol{x})sin(26.55\pi t) \tag{5.22}$$

for some constant a. If a structural dynamic solver is working properly, given the appropriate initial velocity distribution based on Eq. (5.22), it should preserve the simple harmonic motion of the structure at the determined frequency and mode shape.

This test was performed on AStrO, scaling the velocity distribution to have an initial magnitude of 1.0 at the tip. The time step was set to the period of vibration times 10^{-2}. The results for the tip displacement of the dynamic solution predicted by AStrO are plotted along with the Abaqus solution in Fig. 5.12. The two solutions are nearly indistinguishable, indicating AStrO's predicted dynamic solution to be comparable to that of Abaqus and consistent with known experimental data.

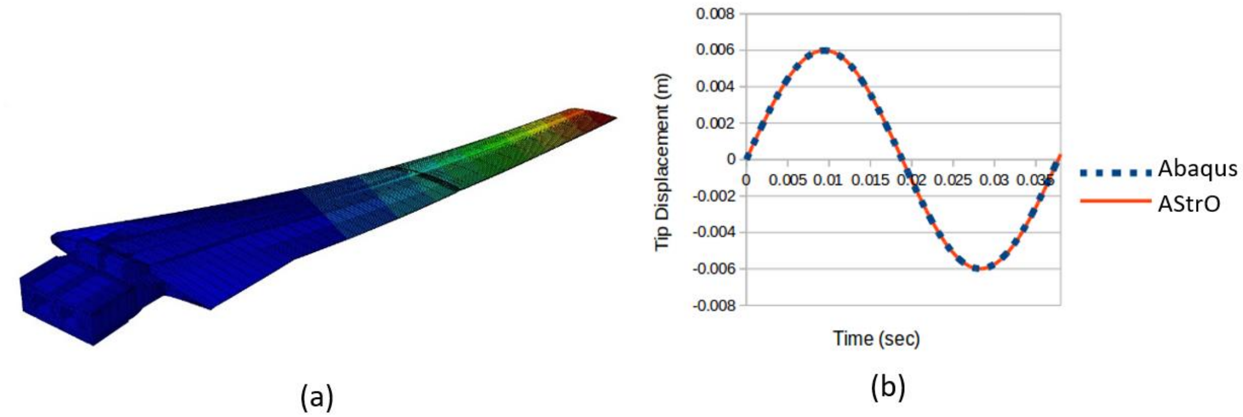


Figure 5.12 (a) Displacement distribution of HIRENASD wing in first mode free vibration. (b) Dynamic response of tip displacement HIRENASD wing in free vibration with Abaqus solution.

5.4 Coupled Aero-Structural Analysis

To validate the combined aero-structural modeling capability of AStrO and NSU3D, the static aeroelastic response of the HIRENASD wing described in the previous section was computed. The flow conditions were Mach=0.8, Incidence = 1.5 degrees, Re=7 million and a dynamic pressure of 40,055.4 Pa. The CFD mesh used was a coarse, node-centered, unstructured mesh available at the AePW web site, which contains a mixture of prismatic and tetrahedral elements with approximately 6.5 million points. The flow solver was run a total of 3 fluid-structure coupling cycles. Within each coupling cycle, the flow solver was run 200 multigrid cycles, the CFD forces were transferred to AStrO, which solved for the static displacements and returned them to the surface CFD mesh. NSU3D's mesh deformation solver was then run for 50 multigrid cycles, and the process was repeated. Figure 5.13 illustrates the convergence of the CFD solver for this case, showing diminishing jumps in convergence at each fluid-structure update. The flexible wing lift coefficient is computed as 0.3304, which compares well with the values reported in the AePW workshop for the static aeroelastic case using modal analysis [86]. The computed surface pressure profile and the distribution of static deflection vertical displacements are shown in Fig. 5.14.

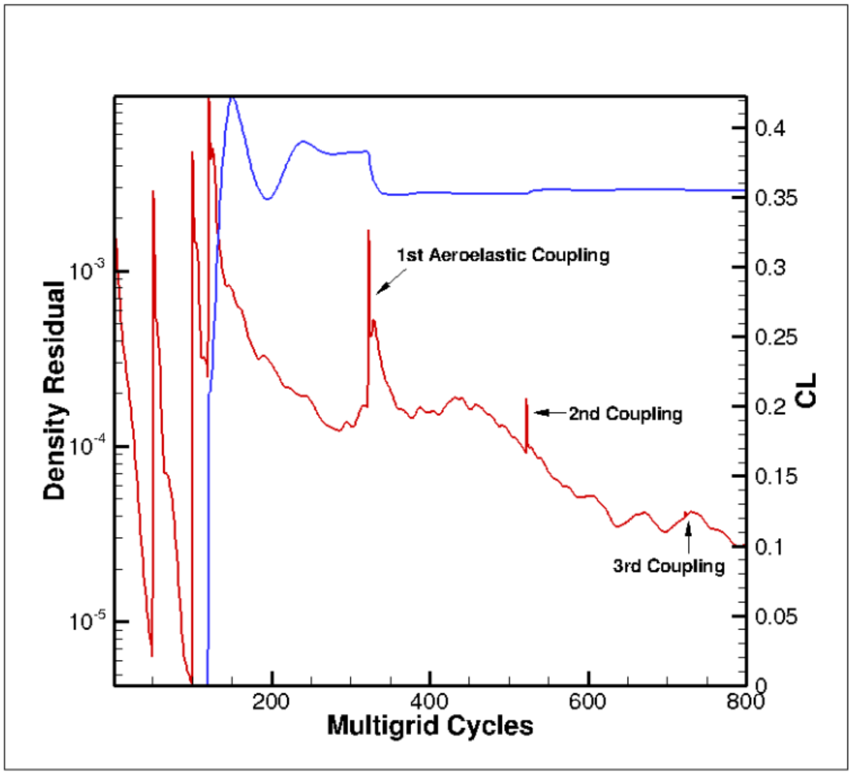


Figure 5.13 Convergence history of static aeroelastic HIRENASD case.

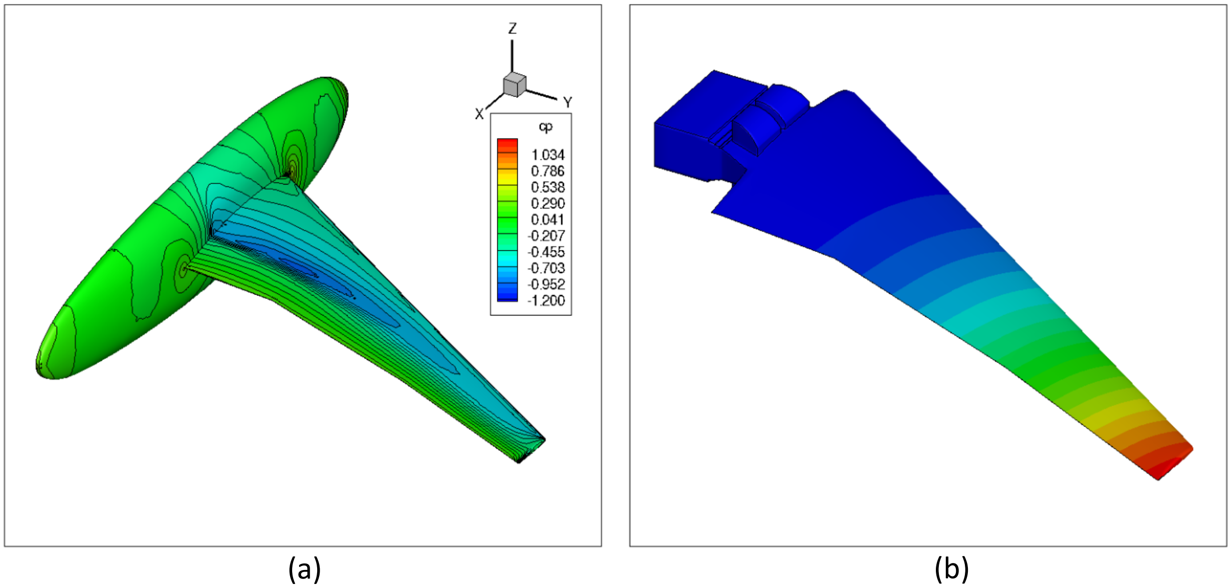
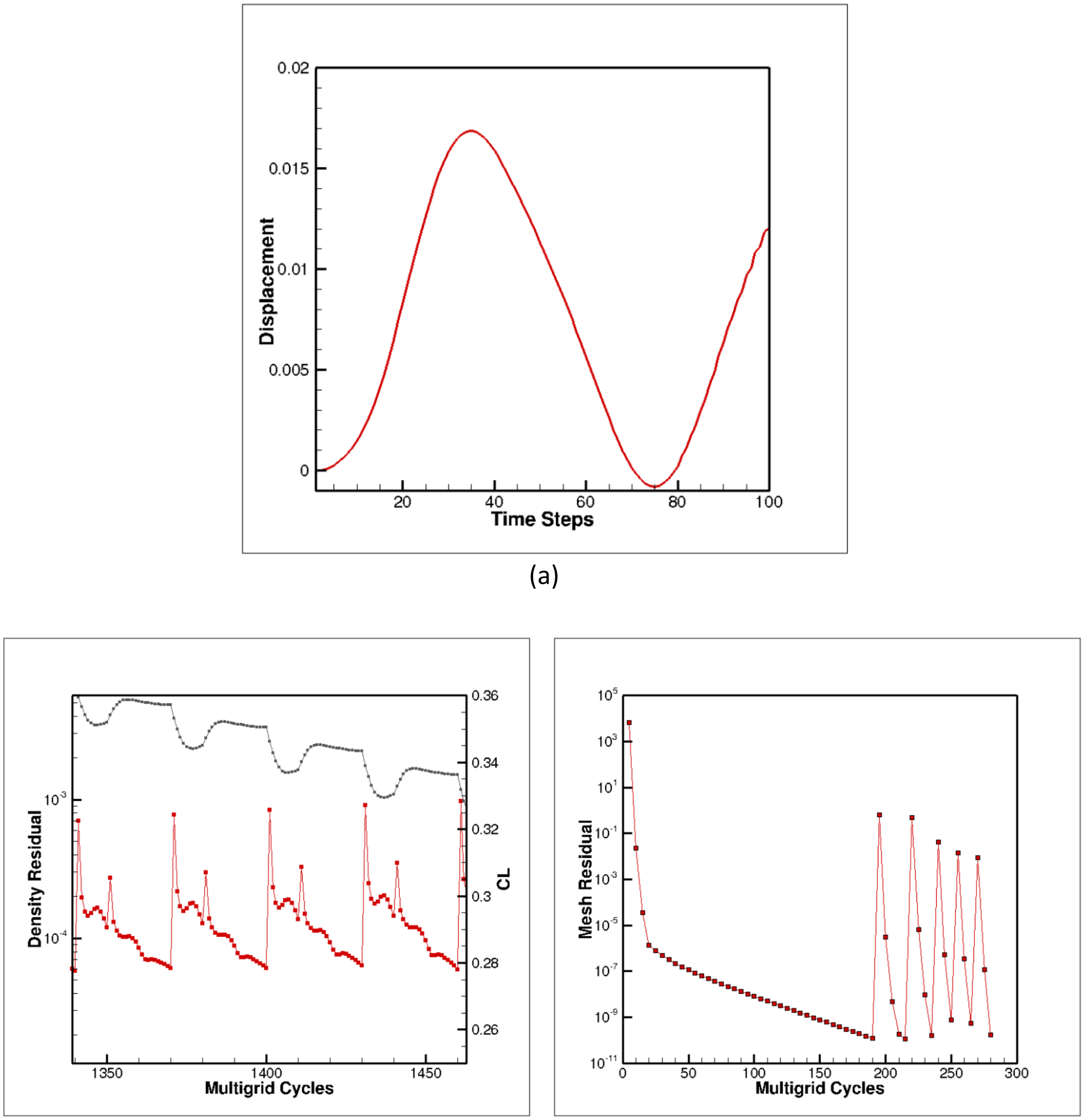


Figure 5.14 (a) Computed surface pressure coefficient contours on final flexible wing solution, (b) Computed displacement field on wing structure.

A time-dependent aeroelastic case was run using the same HIRENASD structural model and CFD mesh. Using the same flow conditions, the flow was first computed with the wing structure held fixed (i.e. rigid wing with no allowed displacements). At time t=0, the wing was suddenly released and responded to the aerodynamic loads with a periodic motion with decreasing amplitude which eventually converged to the previously computed steady-state aeroelastic solution. A time step size corresponding to 0.01 of the period of the natural frequency of the first bending mode of the wing is used in this calculation. Figure 5.15 (a) depicts the time response of the wing, as a plot of the displacement of a point near the wing tip. Figure 5.15 (b) illustrates the convergence of the coupled dynamic aeroelastic system, while Figure 5.15 (c) depicts the convergence history of the mesh deformation equations. A total of 3 coupling cycles were used at each time step, with 10 multigrid cycles employed for the CFD solver within each coupling cycle, resulting in a total of 30 flow cycles per time step. The jumps in residual and lift at the start of each coupling cycle were seen to decrease monotonically and the lift values at the end of each time step are well converged. The mesh deformation equations of NSU3D were converged 10 orders of magnitude at each coupling iteration. Since these equations were initialized with the displacements obtained from the previous coupling cycle, the initial residuals at each coupling iteration decrease monotonically, indicating that the deflections of the structural model were converging as the fully coupled aeroelastic problem was solved to tighter tolerances. This problem was run on 128 processors and required approximately 1 minute for each physical time step.



(a)

(b)

(c)

Figure 5.15 (a) Time history of tip displacement for dynamic aeroelastic HIRENASD test case, (b) Convergence history over selected time steps for flow solver, (c) Convergence history of mesh deformation problem at selected time steps.

Chapter 6

Case Studies and Results

In this chapter, two case studies are presented in which AStrO was used to perform investigations of interest in the field of aero-structural dynamics. The first case study is an attempt to minimize a measure of stress that has been correlated to the propagation of fatigue stress in fiber-reinforced composites, in a wind turbine blade subject to various loading conditions. The second case study investigates a novel way of applying buckling constraints in gradient-based optimization, which stands to cut computational costs compared to conventional methods. The results provide insight into what can potentially be achieved with adjoint-based structural optimization, and with the new methods and approaches considered.

6.1 Fatigue Stress Minimization of a Wind Turbine Blade

As discussed in Chapter 1, the economic viability of wind energy is strongly influenced by the ratio of operating lifespan of turbines to levelized power output. It therefore has always been of great interest to maximize turbine lifespan while maintaining performance and power output. Damage due to fatigue is a major contributor to the breakdown of wind turbine structures, and the reduction of such damage is a critical part of maximizing lifespan. The scalar off-axis matrix stress criterion developed by Fertig [61] has been shown to effectively correlate with the propagation of fatigue damage in fiber-reinforced polymer composites, but it has not previously been posed as an objective in an applied gradient-based optimization problem. In the present study, AStrO was employed to minimize the Fertig stress criterion throughout the structure of a wind turbine blade under several loading conditions, intended to simulate the effective loading on a blade in operation.

6.1.1 Structural Model of Sandia SWiFT 13 Meter Wind Turbine Blade

The wind turbine blade model used as the subject for this work is a 13 meter blade with a 0.6 meter cylindrical section diameter from Sandia National Laboratories used for the SWiFT project [87]. For this model detailed geometric and material composition data is readily available, and previous work has been done by Bhuiyan *et al.* [88] in generating the model in Abaqus for conducting fatigue studies. The blade's outer skin geometry was constructed by lofting together a series of airfoil cross sections defined at each of 34 stations along the spanwise length. The cross section at outboard stations consists of leading edge and trailing edge panels, and a central box beam with thick reinforced spar caps on the upper and lower sides and shear webs connecting the spar caps running normal to the chord. Figure 6.1 illustrates the general blade geometry.

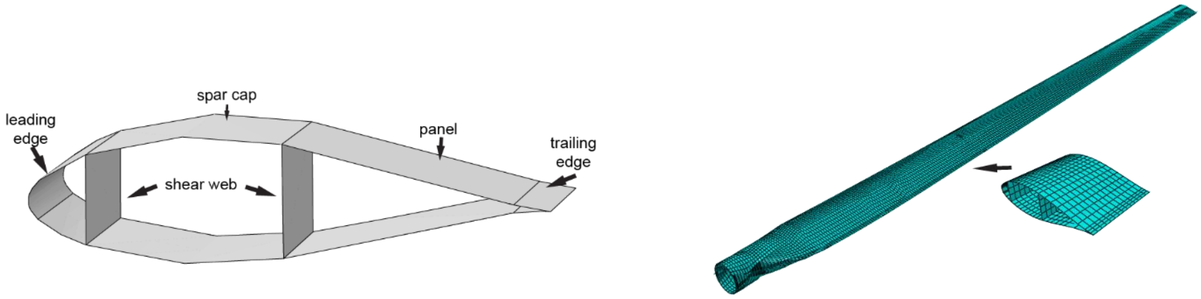


Figure 6.1 Cross-sectional design and spanwise geometry of SWiFT wind turbine blade (Ref. 88).

The finite element mesh for the model was generated in Abaqus out of 16,310 four-node shell elements, following the definition of each spanwise station cross section from the SWiFT report [87]. The blade structure is divided into 388 sections, each originally with a composite layup definition made from an assortment of materials, as illustrated in Fig. 6.2. For the present studies, all sections of the blade structure were considered to be a single ply of UD1200 glass fiber reinforced polyester resin laminate.

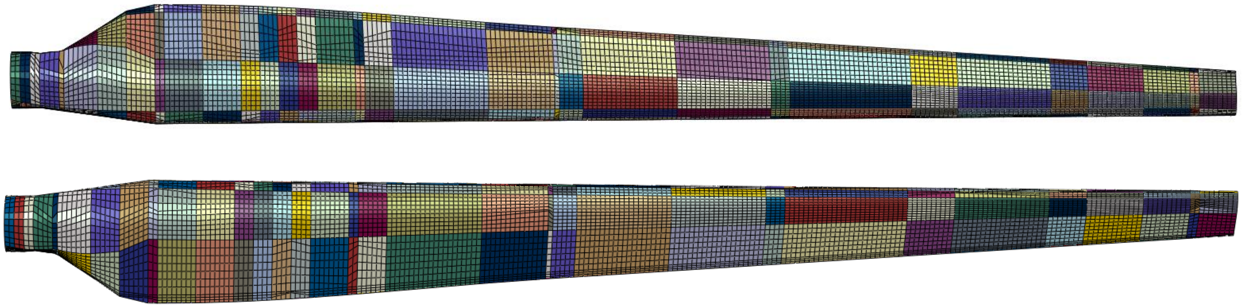


Figure 6.2 Section divisions of the SWiFT wind turbine blade originally defined by the structural finite element model.

In their previous work, Bhuiyan *et al.* [88] used the kinetic theory of fracture (KTF) applied to volume-averaged matrix constituent stresses to predict fatigue damage evolution in the blade. KTF treats polymer fatigue as a thermally activated process, quantifying damage with a scalar parameter n which ranges from zero in an undamaged state to unity at a state of failure. The damage accumulation rate is given by

$$\frac{dn}{dt} = (n_0 - n)^{\lambda} \frac{kT}{h} \exp\left(\frac{\gamma\sigma_{eff} - U}{kT}\right) \tag{6.1}$$

where λ is a damage accumulation exponent, U is an activation energy associated with microcracking, and γ is an activation volume associated with stress dependence on microcracking, all material-dependent. T is absolute temperature, h is Planck's constant and k is Boltzmann's constant. The parameter n_0 is a constant that forces the solution of Eq. (6.1) to reproduce Zhurkov's durability equation [89]. KTF has been shown to accurately predict time-dependent failure in polymers [54,90-91] and has been shown to accurately predict composite fatigue damage and failure when applied to the matrix constituent [55, 58-61].

A scalar effective off-axis volume-averaged matrix stress is required in Eq. (6.1). Using the strength-life equal rank assumption the following effective stress is defined [61]:

$$\sigma_{eff} = \sqrt{A_t\{I_{m,t}\}^2 + I_{m,s1} + A_s I_{m,s2}} \tag{6.2}$$

where

$$I_{m,t} = \frac{1}{2}\left(\sigma_{m,22} + \sigma_{m,33} + \sqrt{\left(\sigma_{m,22} + \sigma_{m,33}\right)^2 - 4\left(\sigma_{m,22}\sigma_{m,33} - \sigma_{m,23}\right)}\right)$$
$$I_{m,s1} = \sigma_{m,12}^2 + \sigma_{m,13}^2 \tag{6.3}$$
$$I_{m,s2} = \left(\frac{1}{4}\left(\sigma_{m,22} - \sigma_{m,33}\right)^2 + \sigma_{m,23}^2\right)$$

A_t and A_s are material parameters obtained from three static failure tests: transverse tension (S22$^+$), transverse compression (S22$^-$), and transverse shear (S12) of a unidirectional laminate. For this study, the static failure strengths were taken from Bhuiyan *et al.* [88] to be S22$^+$ = 63 MPa, S22$^-$ = -180 MPa, and S12 = 72 MPa. This gave the calculated values A_t = 0.35 and A_s = 1.3. $I_{m,t}$ is a transversely isotropic invariant giving the maximum matrix stress normal to the fiber direction. Macaulay brackets {} indicate that the term becomes zero if the quantity inside is negative. $I_{m,s1}$ and $I_{m,s2}$ are transversely isotropic invariants corresponding to maximum longitudinal and transverse shear, respectively.

Localization of composite-level stresses to volume-averaged matrix stresses was accomplished via multi-continuum theory [92, 93]. This approach yields a unique mapping for the volume-averaged matrix stress $\boldsymbol{\sigma}_m$ given by

$$\boldsymbol{\sigma}_m = \frac{1}{\left(1 - v_f\right)}\left([I] - [C_f][S_m]\right)^{-1}\left([I] - [C_f][S_c]\right)\boldsymbol{\sigma}_c \tag{6.4}$$

where v_f is the fiber volume fraction, $\boldsymbol{\sigma}_c$ is the composite stress, $[S_m]$ and $[S_c]$ are the matrix and composite compliance tensors respectively, $[I]$ is the 6 X 6 identity matrix, and $[C_f]$ is the material stiffness tensor for the fiber.

The composite stress in a structure was calculated using the finite element displacement and strain solution. From this the necessary matrix stresses in Eq. (6.3) were computed, followed by the effective off-axis stress in Eq. (6.1). Examining the form of Eq. (6.1) reveals that the driving force for fatigue damage and failure is the effective matrix stress. Thus, the minimization of this quantity will be discussed in subsequent sections.

6.1.2 Optimization Objective and Methodology

Based on the established model from the kinetic theory of fracture in Eq. (6.1), minimization of the effective off-axis stress in Eq. (6.2) will slow the accumulation of fatigue damage, and maximize the lifespan of the structure. It would therefore be natural to define the maximum effective off-axis stress in the structure as the objective function in optimization. However, in gradient-based optimization it is ideal to have objectives that are smooth and differentiable in the design space. The maximum stress in a structure itself does not possess this characteristic, and smooth aggregation functions are typically employed instead. One such aggregation function is the *p-norm*, which approximates the maximum value of a field variable in a domain, and is known to be well-suited for objectives that are confined to non-negative real values [94-96]. In this work a type of p-norm was employed as the objective function, specifically

$$L = \int_0^t \int_\Omega \sigma_{eff}^4 d\Omega\, dt \tag{6.5}$$

The exponent in a p-norm function should be chosen to be high enough to correlate strongly with the maximum value of the field variable, yet low enough not to produce excessively sharp curvature in the objective that could impede convergence in optimization. Duysinx and Sigmund [94] found an exponent of 4 to be a well-balanced choice.

The integral in Eq. (6.5) was evaluated numerically element-by-element after solving for the displacement. Effective stress was computed from displacement gradients at the centroid of each element, the optimal location to minimize effects such as parasitic shear in linear elements.

The objective was minimized by tailoring the composite fiber angles throughout the structure. Two different design spaces were considered, one in which each design variable defined the fiber angle for a section of the blade, as laid out in Fig. 6.2, and another in which each design variable defined the fiber angle for an individual element of the blade model. The former is far more realistic from a fabrication point of view, while the latter allows more freedom in the design, and potentially a more optimal result.

To be specific, the angle represented by each design variable is that of the fiber orientation in the local 1-2 plane of the corresponding section or element, with respect to the blade's longitudinal direction. The local 3-axis of each shell element is always defined normal to the element's midplane, so the fiber direction always stays within the midplane of the shell. A given design variable D_i defined the material local coordinate system for the i^{th} section or element as follows

$$[\alpha_i] = \begin{bmatrix} \cos(D_i) & \sin(D_i) & 0 \\ -\sin(D_i) & \cos(D_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} [\alpha_{i,0}] \tag{6.6}$$

where $[\alpha_{i,0}]$ is the initial coordinate system transformation matrix for section or element i. The initial coordinate system was defined such that the fiber direction aligned with the blade's longitudinal direction projected onto the element midplane (see Fig. 6.3). Eq. (6.6) was used to obtain the displacement solution, as well as the objective gradient by incorporating $[\alpha_i]$ and $\frac{\partial[\alpha_i]}{\partial D_i}$ into the sensitivity definitions given in Chapter 4.

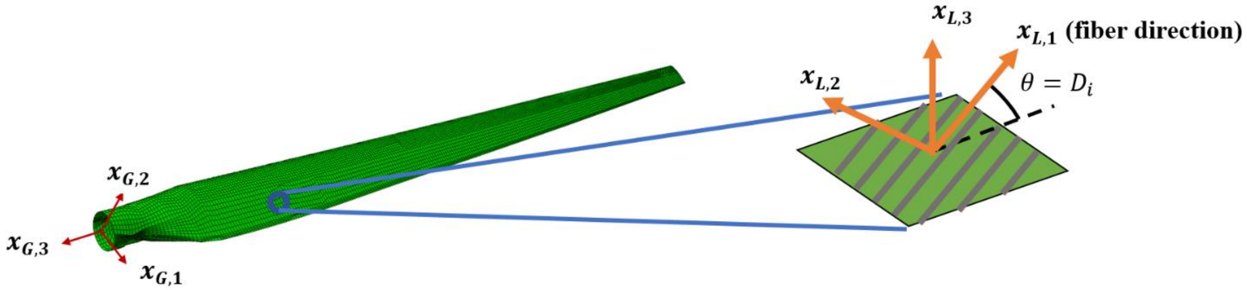


Figure 6.3 Global and local element coordinate systems for SWiFT blade model.

Aside from testing the capabilities of AStrO in sensitivity analysis and optimization, one goal of this investigation was to estimate the potential theoretical gain of composite design optimization with regard to fatigue life. Specific manufacturing and fabrication constraints were not of great concern. Consequently, the only design constraints imposed on how the fiber angles could be modified by the optimization are that they remained in plane with the section/element, and that each section maintained a uniform angle in cases where applicable. But these conditions were both satisfied implicitly in the way the design variables were defined, so that no external constraints needed to be imposed.

The optimization algorithm used was a steepest descent backtracking line search [78], using the gradient of the objective function obtained using the adjoint method as the search direction at each design cycle. Although not always the optimal choice, this method is straightforward to implement without interfacing with external codes, and in the absence of nonlinear design constraints has proven effective. Each design cycle involves first solving for the structural displacements at the current values of the design variables, then solving for the adjoint and calculating the gradient of the objective function with respect to the design variables. Solutions of linear systems were performed with the direct $[L][D][L^T]$ solver built into AStrO. For all the present studies, the optimizer was run through ten design cycles for each design problem.

An important consideration is that the optimum fiber angle distribution for the structure is inevitably dependent on loading. Wind turbines are routinely subject to three main types of applied loads: 1) centrifugal loads 2) gravitational loads and 3) aerodynamic loads. In this work the optimization analysis was performed with each type of loading individually before examining the total combined loading, and the optimal designs for all cases were compared. A total of five load cases were investigated, detailed in the following sections.

Static Analysis Under Centrifugal Loading Only

The first load case subjected the turbine blade to centrifugal loads only. The structural response of the blade was analyzed by AStrO in a coordinate system rotating with the blade with its 2-axis as the axis of rotation. The 1-axis runs in the blade's general chord direction and the 3-axis runs parallel with the longitudinal axis of the blade, as shown in Fig. 6.3. Under such conditions the centrifugal loads can be modeled as a static body force distributed throughout the structure. The body force per unit volume due to centrifugal loading was considered constant in time at any point in the structure, since the angular velocity is assumed constant, and is calculated by

$$\boldsymbol{f}_c = \rho\omega^2\boldsymbol{r}(\boldsymbol{x}) \tag{6.7}$$

where ρ is the mass density of the material at a point in the structure, ω is the angular velocity of the rotating blade, and $\boldsymbol{r}$ is the normal position vector projected from the axis of rotation (2-axis) to the point, or

$$\boldsymbol{r}(\boldsymbol{x}) = [x_1, 0, x_3]^T \tag{6.8}$$

The angular velocity was set to correspond with the optimal rate for power production for this particular turbine, or 43 rpm.

Dynamic Analysis Under Gravitational Loads Only

The second load case subjected the blade to gravitational loads only. In the local rotating frame of motion, the gravitational component of the body force is constant in magnitude at a given point on the structure, but its direction varies cyclically with a frequency matching the angular velocity of the blade. For this component,

$$\boldsymbol{f}_g = \rho g \boldsymbol{n}(t) \tag{6.9}$$

where g is the acceleration due to gravity, and the unit vector $\boldsymbol{n}$ varies in time as follows:

$$\boldsymbol{n}(t) = [\sin(\omega t)\,, 0, -\cos(\omega t)]^T \tag{6.10}$$

Because of the load’s dependence on time, this case was analyzed dynamically. The structure was started from rest, with both displacement and velocity at zero. Under such conditions there can be an initial transient period before the structure settles into its periodic response. Therefore, the blade was rotated through three full revolutions under the load field defined in Eq. (6.9) to more thoroughly capture the range of motion. The optimization was performed based on the stress response integrated over all three revolutions.

Static Analysis Under Aerodynamic Loads Only

The third load case subjected the blade to aerodynamic loads only. Aerodynamic loads on the blade structure were pre-generated by the NSU3D flow solver, by solving for the pressure distribution and skin friction over the blade surface and mapping the resulting forces onto the blade structural mesh as described in Chapter 2. Flow conditions were set for an inflow velocity of 12 m/s, with the blade rotation at 43 rpm, to match those in the reference work of Bhuiyan *et al.* [88]. In reality, these aerodynamic loads are time-varying, and to perform the most rigorous fatigue analysis would require fully coupled dynamic aeroelastic optimization accounting for wind gusts and other variations to capture an accurate time history of the aerodynamic loads. But for the present study, the aerodynamic loads were taken to be constant throughout the optimization process, corresponding to the steady-state solution under these conditions. The optimization was then performed based on the static response under these loads.

Static Analysis Under Combined Loading

The fourth load case subjected the blade to centrifugal, gravitational and aerodynamic loads combined, with the gravitational load in Eq. (6.9) evaluated at $\omega t = \pi/2$. That is, with the gravitational load purely in the direction normal to the blade axis, corresponding to the blade in the horizontal position. In this position the bending loads due to gravity are maximized, as are the maximum stresses in the structure. The optimization was then performed based on the static response under these loads.

Dynamic Analysis Under Combined Loading

In the fifth and final load case, all three types of loads were again applied simultaneously, but the full dynamic response of the structure was simulated. The time-dependence of the gravitational loads was accounted for, but the aerodynamic loads were still considered to be constant, corresponding to the steady-state solution. Again, the analysis was run through 3 full revolutions.

The centrifugal and aerodynamic loads were considered constant in time for this analysis as described above. As such these load cases on their own would not provide a basis to estimate fatigue life directly, as fatigue is an inherently time-dependent phenomenon. However, the present optimization objective was not fatigue life itself but the effective off-axis matrix stress that has been correlated with the propagation of fatigue damage. Static components of a combined dynamic loading would still contribute to the mean values of effective off-axis stress under all combined loads, and thus indirectly affect the rate of damage propagation. For this reason, it may still be of value to know the fiber angle distributions optimized for these loads individually and compare them with the results for all combined dynamic loads in an attempt to see what factors may be most dominantly driving the results. If, for example, it was consistently found that the optimal design under a static loading representing a high wind gust closely matched the optimal design under the realistic dynamic loading, then it may save computational expense to use that assumption and avoid the cost of time-dependent simulations. Addressing this comprehensively is far beyond the scope of the present work, but load cases were examined and compared individually with that mindset.

6.1.2 Optimization Results

For all of the load cases and design spaces described in the previous section, the optimization analysis was performed, and three key quantities were recorded at each design cycle: 1) the value of the objective function, 2) the root-mean-square of the objective gradient, and 3) the maximum value of the effective off-axis matrix stress encountered by any point in the structure at any moment in the time history. The progression of all these quantities, non-dimensionalized by their initial values can be seen in Fig. 6.4 through 6.8.

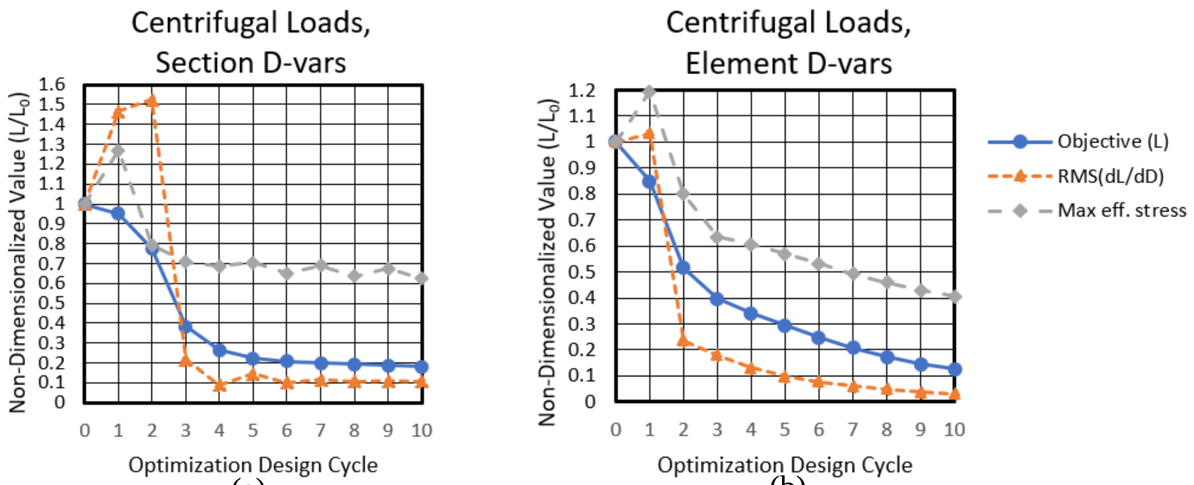
Table 6.1 shows the change in the maximum effective off-axis matrix stress and the change in maximum deflection of the blade after optimization for each load case and design space. The effectiveness of the optimization varied from case to case, and the maximum effective off-axis matrix stress was reduced by 18-60%. Cases with fiber angles defined for all elements individually consistently reduced the objective stress further than cases with fiber angles defined for all sections. However, the change in maximum deflection tended to be significantly greater for cases with design variables for all elements, which could have negative implications on the overall turbine design. There may well be a trade-off between lowering objective stress and preserving desired stiffness properties of the blade. The observation also further confirms the need to ultimately perform fully coupled aeroelastic analysis to capture interdependence between structural deformation and aerodynamic loading.

Table 6.1 Change in maximum off-axis matrix stress and maximum deflection in SWiFT blade structure due to optimization.

	Section Design Variables		**Element Design Variables**	
Load Case	**Change in Max Stress**	**Change in Max Deflection**	**Change in Max Stress**	**Change in Max Deflection**
Centrifugal	-37.67%	-0.64%	-59.41%	-7.43%
Gravitational	-45.69%	-0.50%	-55.49%	0.12%
Aerodynamic	-18.63%	-0.40%	-42.30%	-2.82%
Combined, Static	-19.24%	0.24%	-54.08%	-6.71%
Combined, Dynamic	-21.72%	0.13%	-51.11%	-9.05%

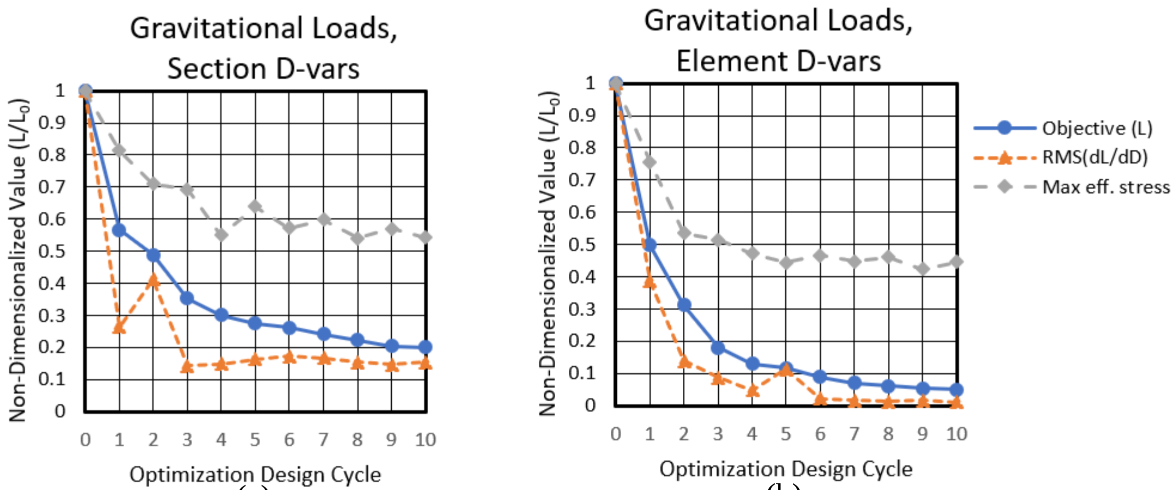
To understand the potential implications on fatigue life, data for stress amplitude vs. cycles to failure in fatigue was consulted for E-glass/epoxy published by Hashin and Rotem [51]. Figure 6.9 shows a logarithmic plot of stress versus number of cycles to failure for the published data points, along with a trendline fit to the data. The experimental composite stresses were converted to effective off-axis matrix stresses based on the 60^o angle of uniaxial loading.

For the present purposes, the trend in Fig. 6.9 is qualitative since the state of stress represented by the data does not match that in every point throughout the SWiFT blade structure. Nevertheless, it is clearly evident that the amount of off-axis stress reduction shown in Fig. 6.4 through 6.8 could have great potential to impact the number of cycles to failure. Based on this data, even a reduction in stress amplitude of 10% could increase the number of cycles to failure by a factor of 10, and even the least effective optimization performed here shows reduction of over 18%.



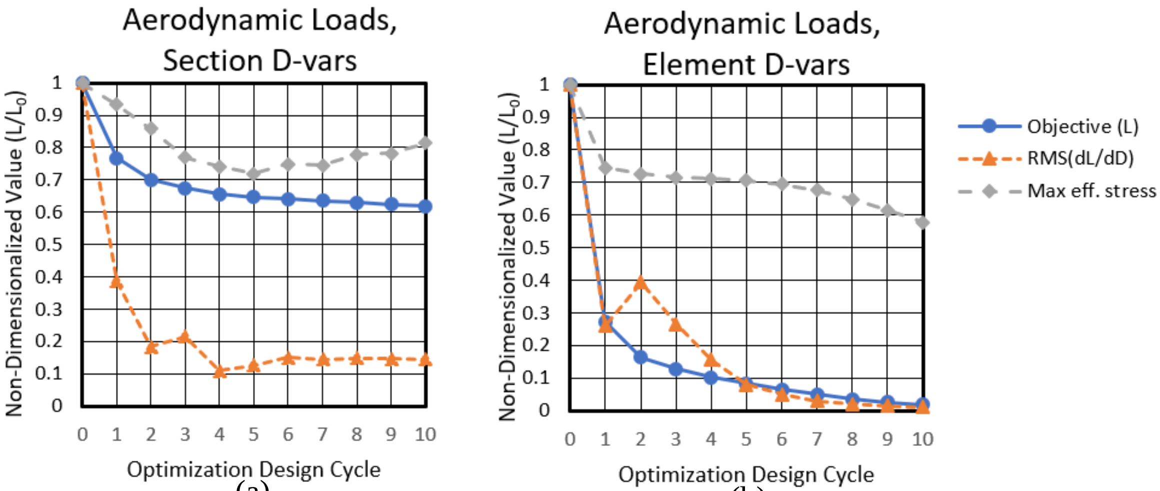
(a) (b)

Figure 6.4 Optimization history for centrifugal loading on the SWiFT turbine blade for (a) design variables defining the fiber angle for each section, and (b) design variables defining the fiber angle for each element.



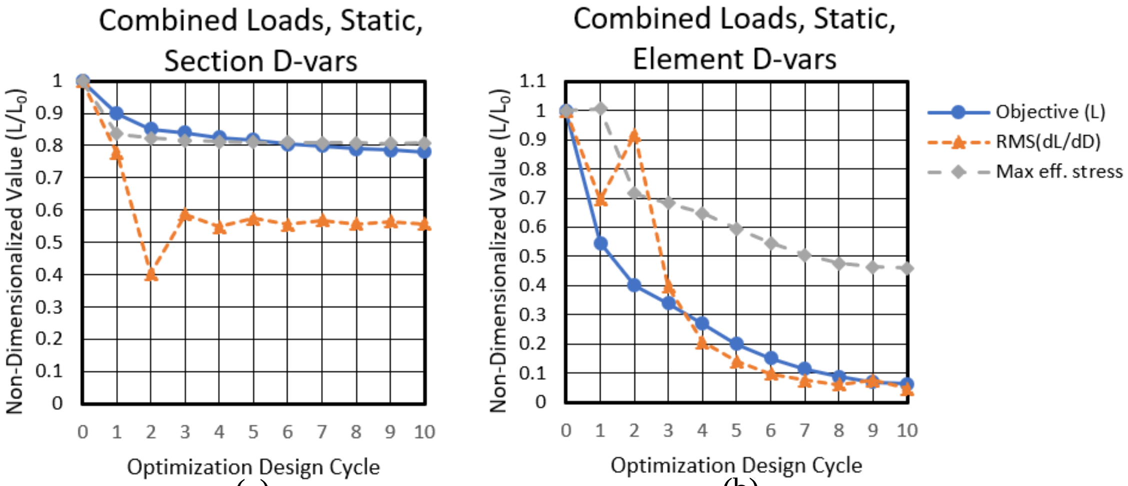
(a) (b)

Figure 6.5 Optimization history for gravitational loading on the SWiFT turbine blade for (a) design variables defining the fiber angle for each section, and (b) design variables defining the fiber angle for each element.



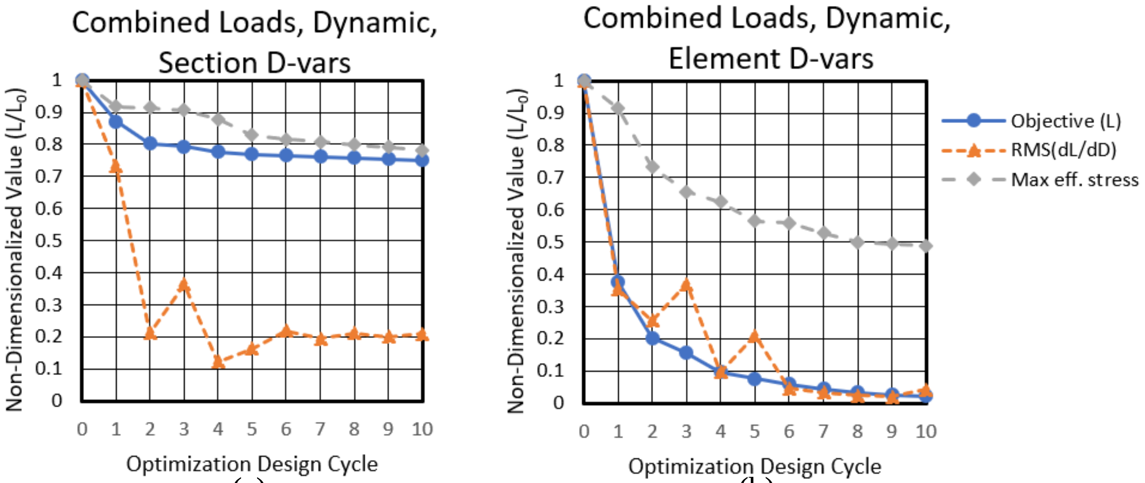
(a) (b)

Figure 6.6 Optimization history for aerodynamic loading on the SWiFT turbine blade for (a) design variables defining the fiber angle for each section, and (b) design variables defining the fiber angle for each element.



(a) (b)

Figure 6.7 Optimization history for combined static loading on the SWiFT turbine blade for (a) design variables defining the fiber angle for each section, and (b) design variables defining the fiber angle for each element.



(a) (b)

Figure 6.8 Optimization history for combined dynamic loading on the SWiFT turbine blade for (a) design variables defining the fiber angle for each section, and (b) design variables defining the fiber angle for each element.

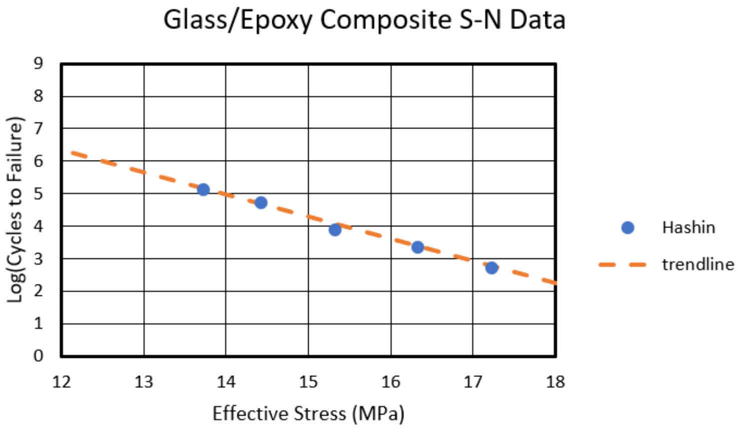


Figure 6.9 Published S-N data for E-glass epoxy composite (Ref. 51), loaded at 60^o from fiber direction with corresponding trendline. Composite stresses converted to effective off-axis matrix stresses.

To investigate the dependence of the optimization on loading conditions, the final optimized fiber angles for each element between the different cases were compared. The final load case, incorporating all loads in the dynamic simulation, was used as a benchmark of comparison for the others, as it is presumably the most realistic simulation of what a turbine blade experiences. The difference in the optimized fiber angles between the final case and each of the first four cases was first computed, denoted ΔD_i. The average absolute value and RMS value of ΔD_i were then computed for comparison, the results shown in Table 6.2.

Table 6.2 Comparison of final optimized ply angles between final load case (case 5), and all previous load cases. ΔD_i represents the difference in the optimized ply angles from final load case, $\Delta D_i = (D_i - D_{i,final\ case})$.

	Section Design Variables		**Element Design Variables**	
Load Case	**Average $\lvert\Delta D_i\rvert$**	**RMS (ΔD_i)**	**Average $\lvert\Delta D_i\rvert$**	**RMS (ΔD_i)**
Centrifugal Only	1.861°	4.863°	1.606°	6.054°
Gravitational Only	2.578°	6.945°	2.032°	7.620°
Aerodynamic Only	0.886°	2.527°	0.905°	4.189°
All Loads (static)	0.568°	1.511°	0.623°	3.120°

These metrics suggest that under the conditions assumed in this case, aerodynamic loading has the strongest influence on the optimum fiber angle distribution of the three main types of loading. The distribution for the static analysis under combined loading shows close agreement with the dynamically loaded case, but the difference is still significant. Figures 6.10 through 6.14 show the distribution of angle changes after optimization for each case, and Fig. 6.15 through 6.19 show the distribution of effective off-axis matrix stress before and after optimization. Even though the results in each case were generally favorable, it is apparent that the optimal design is indeed highly dependent on loading, and in practice many more simulations would need to be run to identify the best overall design for the range of real loading conditions.

As previously mentioned, it is not practical from a fabrication point of view to allow each structural finite element on the blade to have its own unique fiber angle. However, there may still be useful information to be learned from the results of these cases. Two key observations can be taken from Fig. 6.10 through 6.19. First, the vast majority of elements in the structure had very little change in their fiber angle due to optimization, less than 1° change. The most affected areas are concentrated near the root of the blade, which is intuitive since these are the areas of highest stress for a structure under primarily bending loads. Second, the most affected areas tend to be clustered together in certain key regions.

The implication from these observations is that if a blade were to be fabricated to accommodate optimizations such as these, it is likely only a few regions would need special customization. Even though it would not be feasible to give each point on the blade its own fiber direction, it may be possible to re-define the section divisions of the blade so that the most affected areas would be isolated as separate sections, and each section given a homogenized composite design appropriate for its local loading conditions. The main challenge would lie in identifying the best overall design for the range of loading.

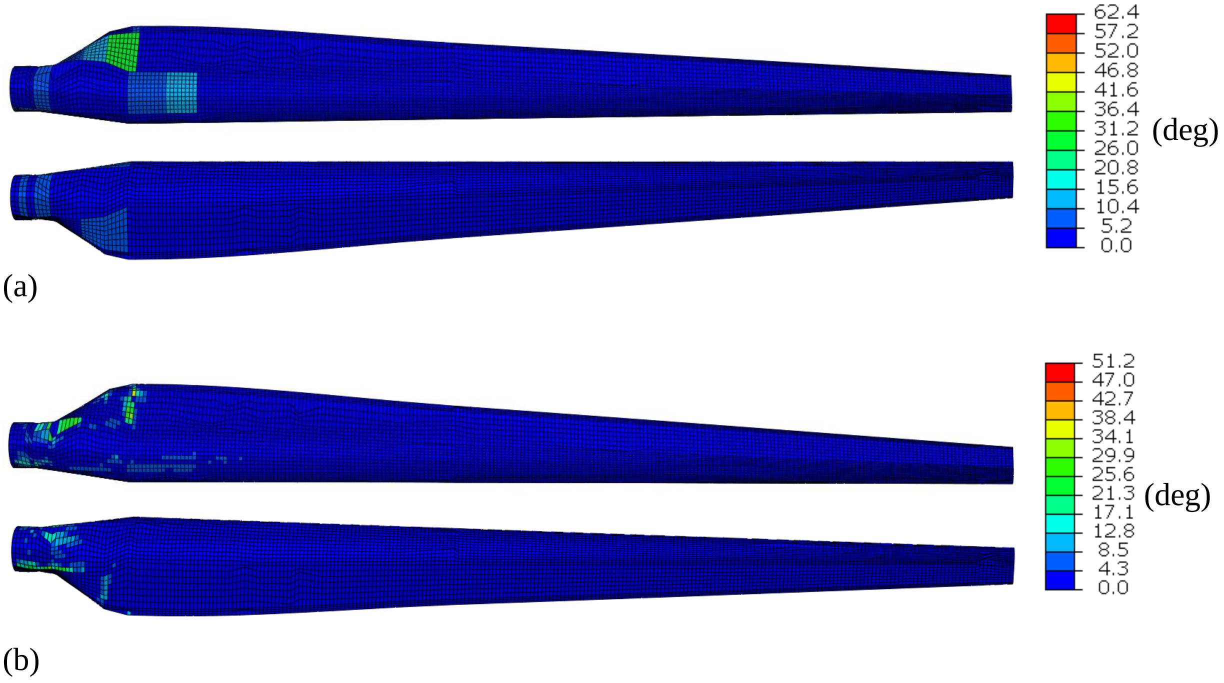
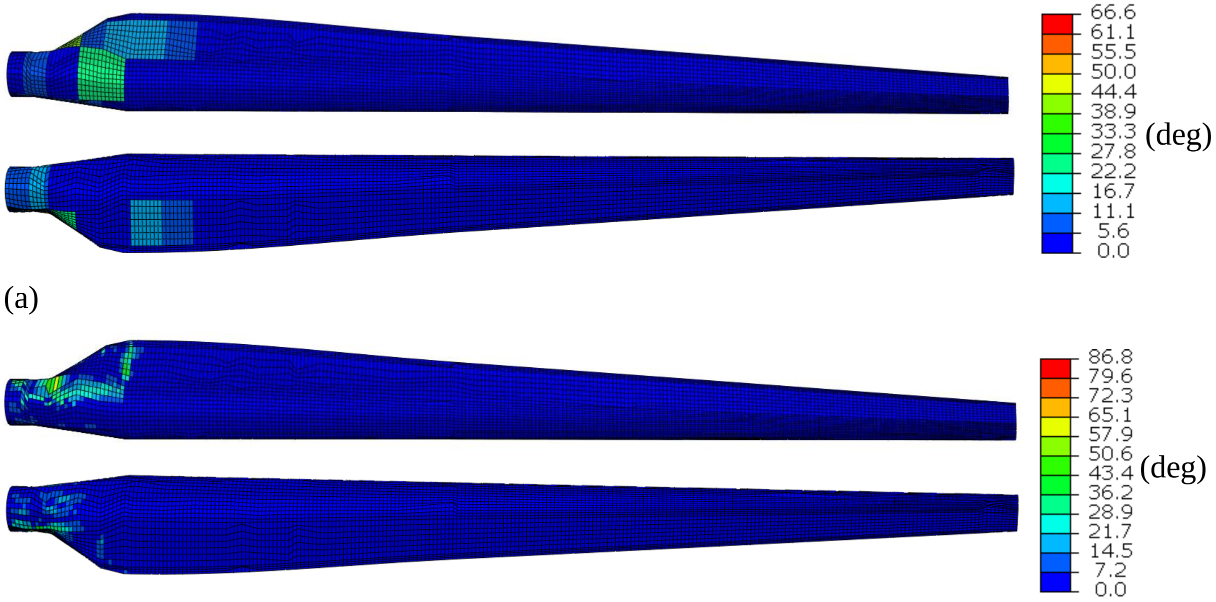


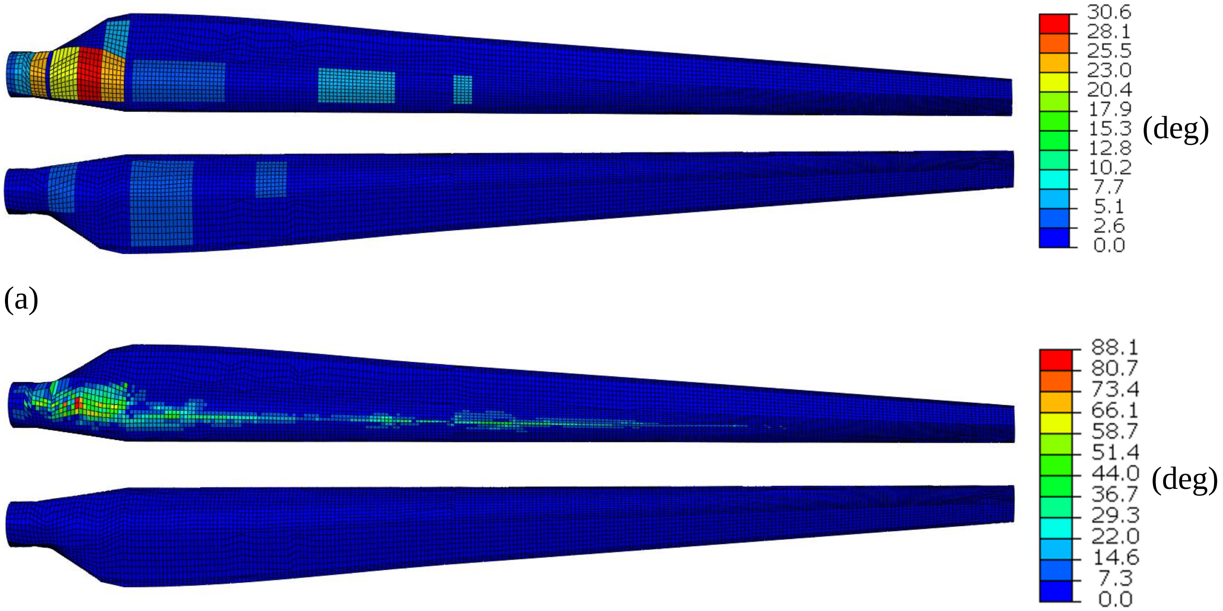
Figure 6.10 Spatial distribution of fiber angle change due to optimization subject to centrifugal loads for (a) design variables defined for each section, and (b) design variables defined for each element.



(a)

(b)

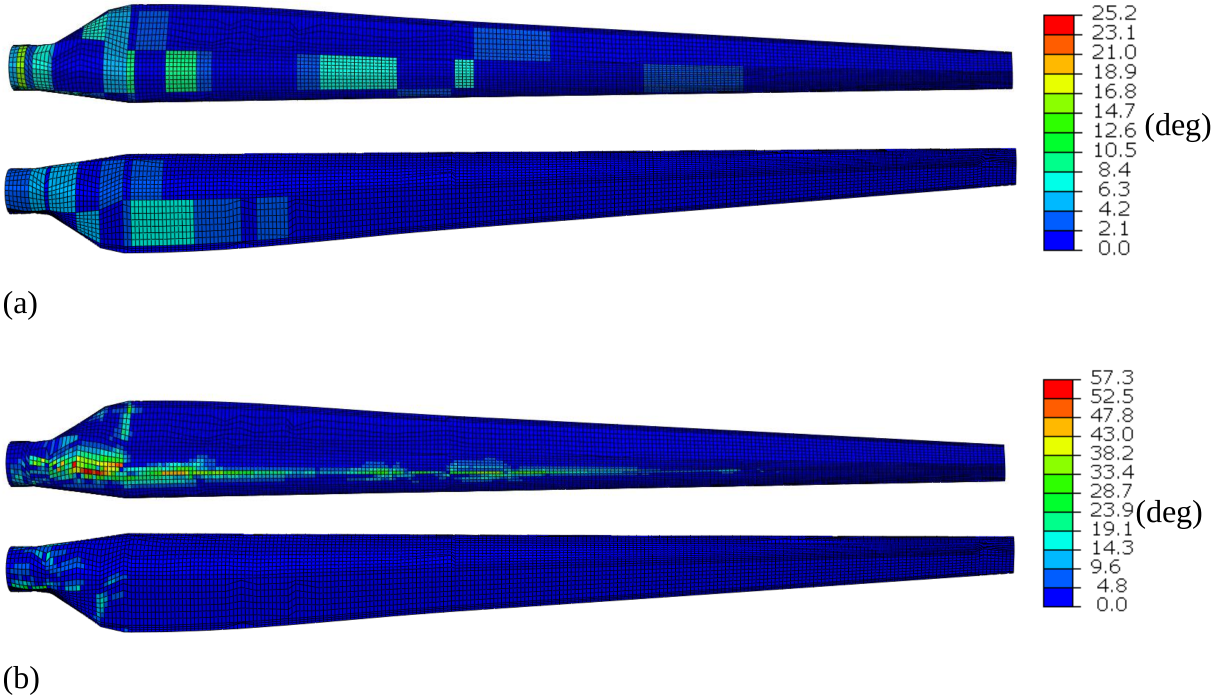
Figure 6.11 Spatial distribution of fiber angle change due to optimization subject to gravitational loads for (a) design variables defined for each section, and (b) design variables defined for each element.



(a)

(b)

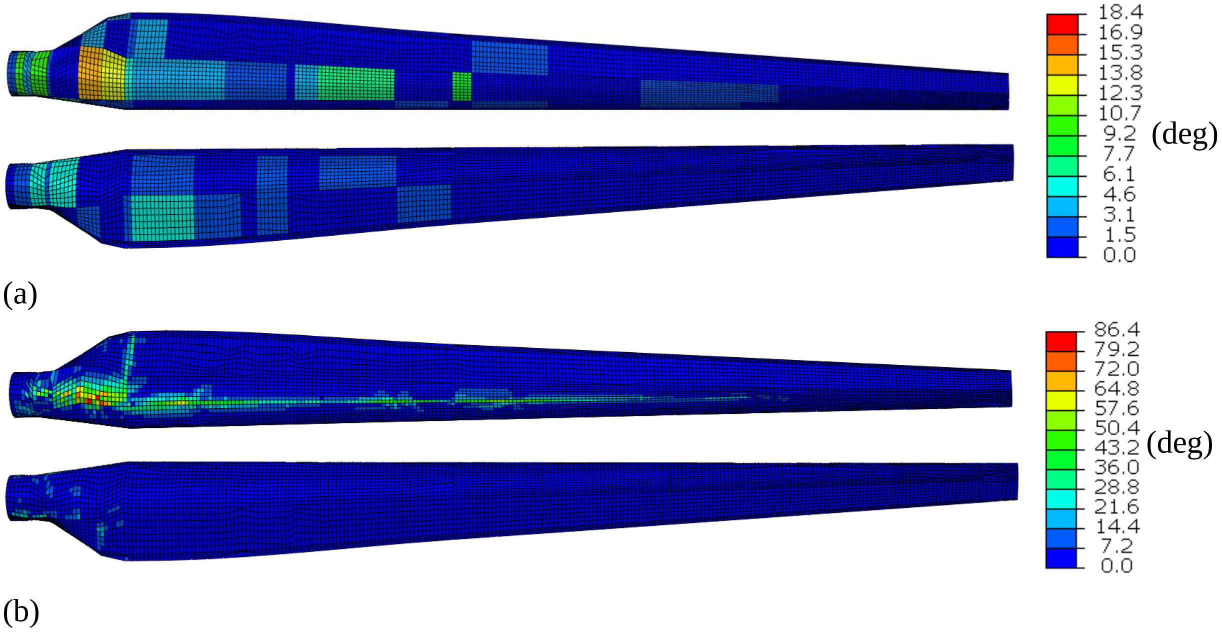
Figure 6.12 Spatial distribution of fiber angle change due to optimization subject to aerodynamic loads for (a) design variables defined for each section, and (b) design variables defined for each element.



(a)

(b)

Figure 6.13 Spatial distribution of fiber angle change due to optimization subject to combined static loads for (a) design variables defined for each section, and (b) design variables defined for each element.



(a)

(b)

Figure 6.14 Spatial distribution of fiber angle change due to optimization subject to combined dynamic loads for (a) design variables defined for each section, and (b) design variables defined for each element.

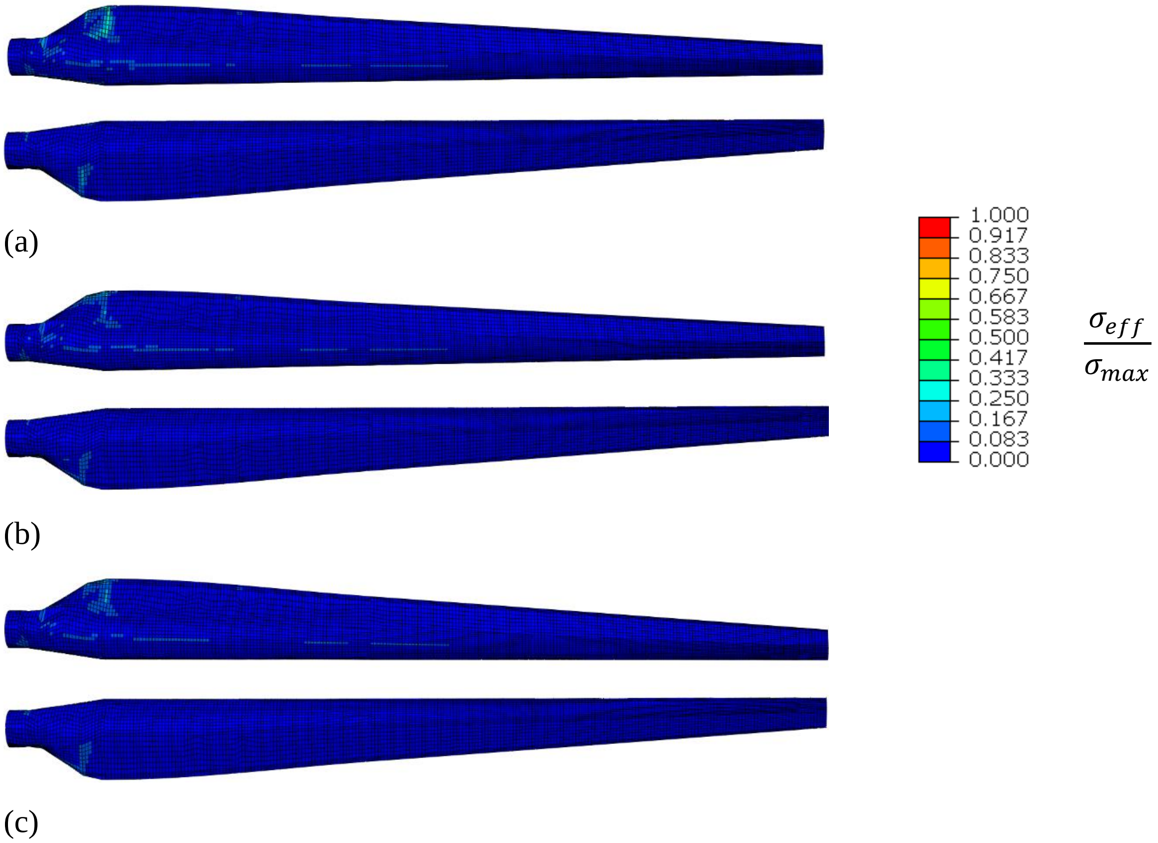


Figure 6.15 Spatial distribution of effective off-axis matrix stress, non-dimensionalized by the original maximum value for centrifugal loading (a) before optimization, (b) after optimization with section design variables and (c) after optimization with element design variables.

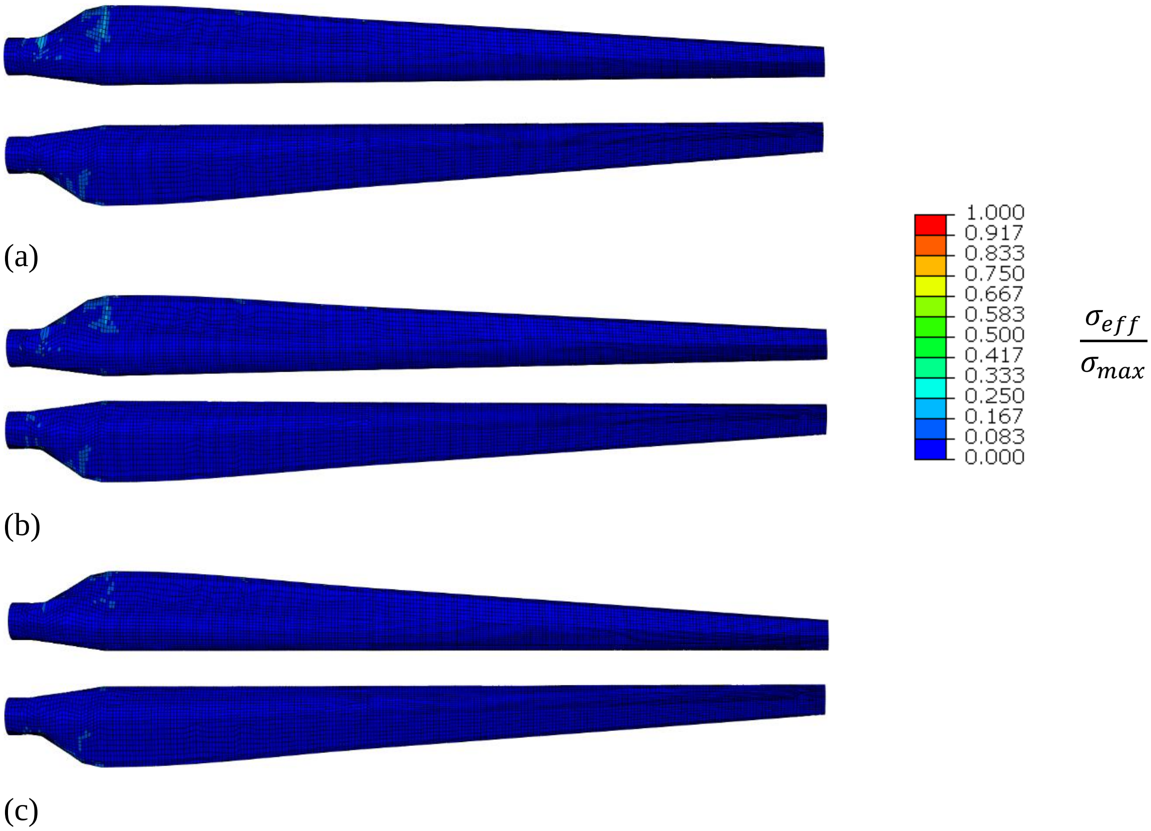


Figure 6.16 Spatial distribution of effective off-axis matrix stress, non-dimensionalized by the original maximum value for gravitational loading (a) before optimization, (b) after optimization with section design variables and (c) after optimization with element design variables.

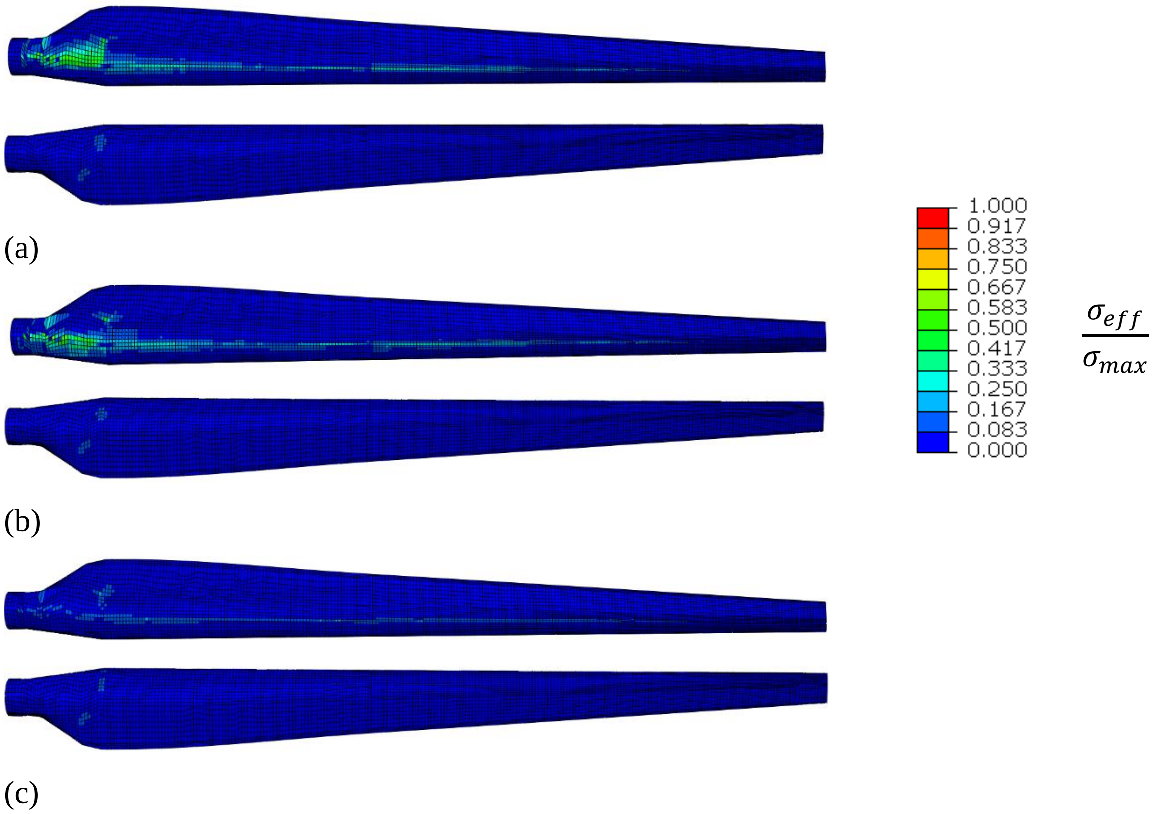


Figure 6.17 Spatial distribution of effective off-axis matrix stress, non-dimensionalized by the original maximum value for aerodynamic loading (a) before optimization, (b) after optimization with section design variables and (c) after optimization with element design variables.

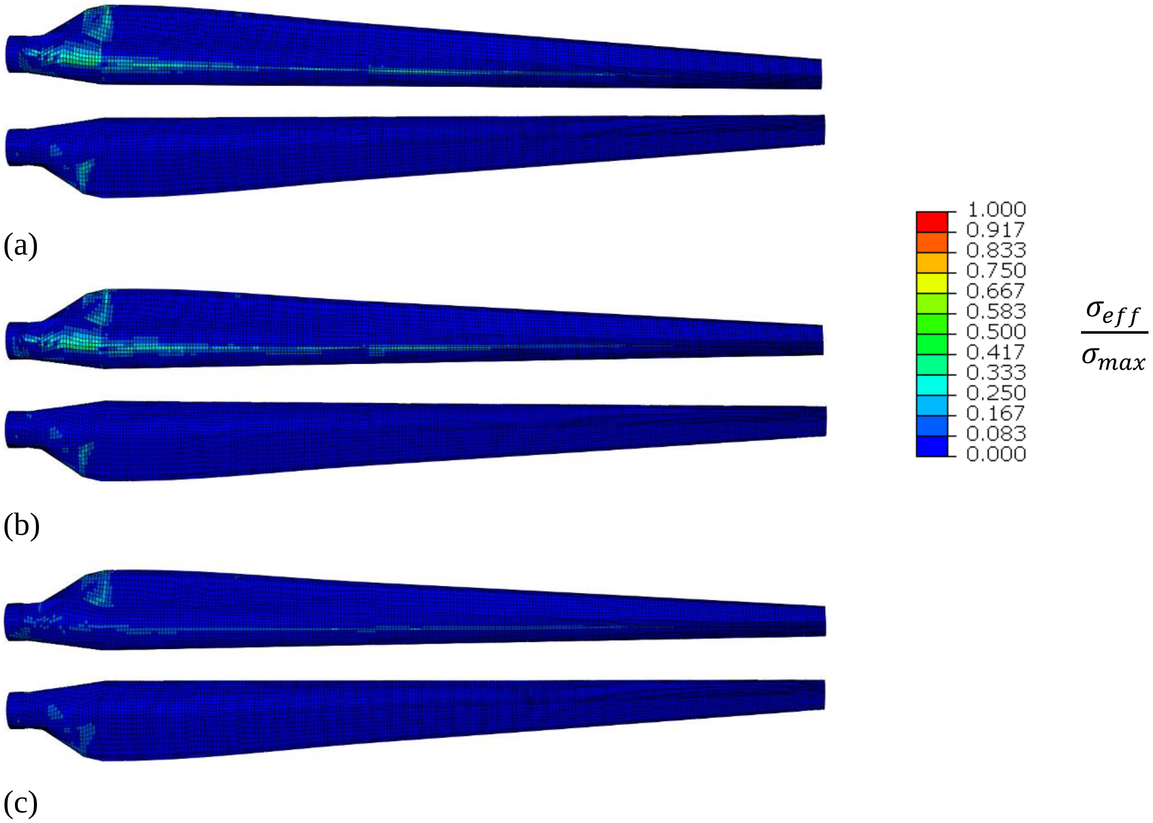


Figure 6.18 Spatial distribution of effective off-axis matrix stress, non-dimensionalized by the original maximum value for combined static loading (a) before optimization, (b) after optimization with section design variables and (c) after optimization with element design variables.

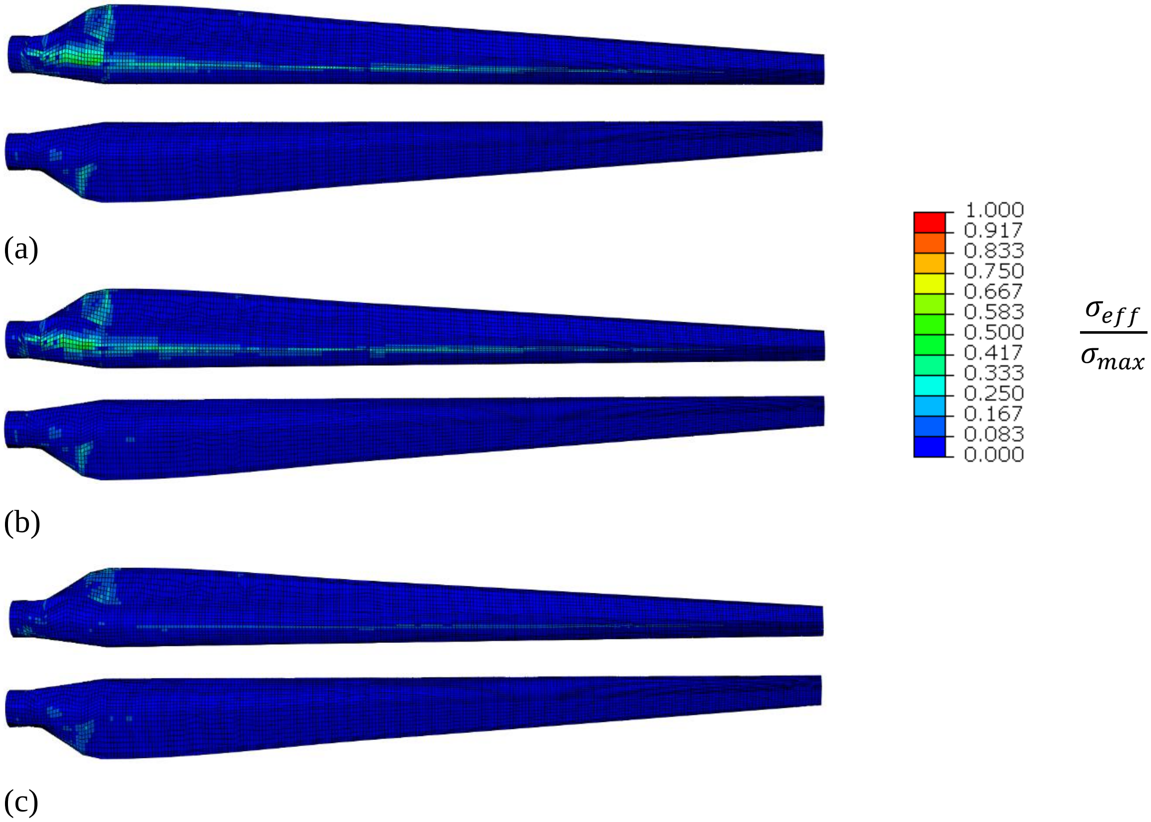


Figure 6.19 Spatial distribution of effective off-axis matrix stress, non-dimensionalized by the original maximum value for combined dynamic loading (a) before optimization, (b) after optimization with section design variables and (c) after optimization with element design variables.

6.2 Truss-Braced Wing: Methods of Buckling Constraints

Structural buckling is an important consideration when designing structures with high aspect ratio. This is especially true since buckling can occur in a structure without any maximum stress limits exceeded, and designs intended to minimize mass and stress tend to be prone to instability. The discussion in Chapter 1 reviewed two main approaches to buckling analysis common in the literature: analytical approximation of members and eigenmode analysis. The former is computationally cheap but lacks accuracy and generality, while the latter is expensive and can be prone to convergence problems in gradient-based optimization. In the present study, an alternative method is proposed which is generally applicable and less costly than eigenmode analysis, yet may also have drawbacks. The proposed method was tested and compared with conventional eigenmode analysis using a structural wing box model with a truss-braced design, which is currently of interest for next-generation aircraft, as a test case.

6.2.1 An Alternative Method for Buckling Constraints

In this section an alternative method for applying buckling constraints or objectives in gradient-based optimization is proposed which is both cost effective and generally applicable. It is appropriate to begin by returning to the theory of buckling and structural instability. The general condition for the critical state of buckling in a structure derives from the total potential energy of the system. When a system is in equilibrium, yet there exists a mode of deformation such that continued displacement in that mode would result in a decrease of total potential energy, it is said to be in unstable equilibrium. Buckling of an elastic structure is one example of this phenomenon.

For a continuous elastic body subject to conservative loads in the form of body forces and surface tractions, the total potential energy of the system can be expressed:

$$\Pi = \int_{\Omega} V d\Omega - \int_{\Omega} (\boldsymbol{f} \cdot \boldsymbol{u}) d\Omega - \int_{S} (\boldsymbol{t} \cdot \boldsymbol{u}) dS \tag{6.11}$$

where V is the elastic strain energy density, $\boldsymbol{f}$ is body force per unit volume and $\boldsymbol{t}$ is the traction on the surface. Ω and S are the domains representing the volume and surface of the elastic body respectively. A fundamental principle of dynamics for conservative systems is that equilibrium occurs when the gradient of total potential energy with respect to displacement degrees of freedom is zero. If the displacement field is represented as a function of a discrete set of variables $\boldsymbol{U}$ multiplied by a matrix of basis functions $[N]$ such that

$$\boldsymbol{u} = [N]\boldsymbol{U} \tag{6.12}$$

as in a finite element analysis, then the state of equilibrium is characterized by

$$\frac{\partial \Pi}{\partial \boldsymbol{U}} = \int_{\Omega} \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T \frac{\partial V}{\partial \boldsymbol{\epsilon}} d\Omega - \int_{\Omega} [N]^T \boldsymbol{f} d\Omega - \int_{S} [N]^T \boldsymbol{t} dS = 0 \tag{6.13}$$

The condition represented by Eq. (6.13) will be satisfied in any state of equilibrium, stable or unstable. But if the vector of displacement variables is perturbed by some finite vector represented by $\delta\boldsymbol{U}$, the resulting change in total potential energy $\delta\Pi$ can be approximated by the second-order Taylor series expansion

$$\delta\Pi = \sum_{i=1}^{n} \frac{\partial \Pi}{\partial U_i} \delta U_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial}{\partial U_i}\left(\frac{\partial \Pi}{\partial U_j}\right) \delta U_i \delta U_j = \left(\frac{\partial \Pi}{\partial \boldsymbol{U}}\right) \cdot \delta\boldsymbol{U} + \frac{1}{2} \delta\boldsymbol{U}^T \left[\frac{\partial}{\partial \boldsymbol{U}}\left(\frac{\partial \Pi}{\partial \boldsymbol{U}}\right)\right] \delta\boldsymbol{U} \tag{6.14}$$

Again, in any state of equilibrium the first derivative of total potential energy with respect to any displacement variable is zero, so the first term in Eq. (6.14) vanishes, leaving

$$\delta\Pi = \frac{1}{2} \delta\boldsymbol{U}^T \left[\frac{\partial}{\partial \boldsymbol{U}}\left(\frac{\partial \Pi}{\partial \boldsymbol{U}}\right)\right] \delta\boldsymbol{U} \tag{6.15}$$

The nature of any dynamic system is to reduce total potential energy. It then follows that if there exists any perturbation vector $\delta\boldsymbol{U}$ such that the expression in Eq. (6.15) would evaluate to be negative, then that perturbation mode would result in an accelerated reduction of potential energy, and therefore structural instability. If no such perturbation vector exists, the system is in stable equilibrium.

The expression in Eq. (6.15) is a symmetric Hessian matrix multiplied from either side by the perturbation vector. Thus, evaluating the stability of a structure in a given state is a matter of determining whether the matrix $\left[\frac{\partial}{\partial \boldsymbol{U}}\left(\frac{\partial \Pi}{\partial \boldsymbol{U}}\right)\right]$ is positive definite. Differentiation of Eq. (6.13) gives a representation of the matrix as

$$\left[\frac{\partial}{\partial \boldsymbol{U}}\left(\frac{\partial \Pi}{\partial \boldsymbol{U}}\right)\right] = \int_{\Omega} \left(\left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T \left[\frac{\partial}{\partial \boldsymbol{\epsilon}}\left(\frac{\partial V}{\partial \boldsymbol{\epsilon}}\right)\right] \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right] + \sum_{k=1}^{6} \frac{\partial V}{\partial \epsilon_k} \left[\frac{\partial}{\partial \boldsymbol{U}}\left(\frac{\partial \epsilon_k}{\partial \boldsymbol{U}}\right)\right] \right) d\Omega \tag{6.16}$$

Observing that the derivative of strain energy density with respect to strain is stress $\boldsymbol{\sigma}$, and a second order of differentiation yields the material stiffness matrix $[C]$ for linear elastic materials, Eq. (6.16) can be re-written as

$$\left[\frac{\partial}{\partial \boldsymbol{U}}\left(\frac{\partial \Pi}{\partial \boldsymbol{U}}\right)\right] = \int_{\Omega} \left(\left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right]^T [C] \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{U}}\right] + \sum_{k=1}^{6} \sigma_k \left[\frac{\partial}{\partial \boldsymbol{U}}\left(\frac{\partial \epsilon_k}{\partial \boldsymbol{U}}\right)\right] \right) d\Omega \tag{6.17}$$

The above is none other than the tangent stiffness matrix of the structure, formed from the principle of virtual work including geometrically nonlinear terms, as given in the developments of Chapter 3, Eq. (3.80). It is concluded that the presence of structural instability and the threat of buckling at a given deformation state can be verified by checking whether the tangent stiffness matrix of the structure is positive definite.

Perhaps the most intuitive way of determining whether a matrix is positive definite is by computing its eigenvalues, since having all positive eigenvalues is one fundamental way of defining positive definiteness. This is the standard approach to generalized buckling analysis, since eigenvalues can be used not only to determine the presence of structural instability but also to quantify it. However, as discussed in Chapter 4, effective gradient-based optimization can require the consideration of many eigenpairs, as well as their sensitivities, which can be costly and problematic for large systems.

An alternative approach may exist that is more cost-effective but still generally applicable. The key to determining positive definiteness of a matrix is to form some symmetric decomposition, consisting of a diagonal matrix, multiplied from either side by another matrix and its transpose. The eigenpair, or spectral decomposition of a matrix is one example of this, where a symmetric matrix $[K]$ can be written in terms of its normalized eigenvectors $[V]$ and a matrix $[\lambda]$ with the corresponding eigenvalues placed on the diagonals, as shown:

$$[K] = [V][\lambda][V]^T$$

$$[V] = [\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \boldsymbol{v}_3 \quad ...], \quad [\lambda] = \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \cdots \\ 0 & 0 & \lambda_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{6.18}$$

If the eigenvalues of the matrix are all positive, then the eigenvalue matrix can be split into a product of two identical matrices with the square roots of the eigenvalues on the diagonals, which would be all real values. Consequently, the inner product of $[K]$ with any vector $\boldsymbol{U}$ operating from both sides can be written as the dot product of a vector containing all real values with itself, as follows:

$$\boldsymbol{U}^T[K]\boldsymbol{U} = \boldsymbol{U}^T[V][\lambda][V]^T\boldsymbol{U} = \boldsymbol{U}^T[V][\lambda]^{\frac{1}{2}}[\lambda]^{\frac{1}{2}}[V]^T\boldsymbol{U} = \left([\lambda]^{\frac{1}{2}}[V]^T\boldsymbol{U}\right) \cdot \left([\lambda]^{\frac{1}{2}}[V]^T\boldsymbol{U}\right) \tag{6.19}$$

Since the dot product of any real vector with itself must always be greater than or equal to zero, it is concluded that any symmetric matrix with all positive eigenvalues must be positive definite.

The same argument can be applied to any symmetric decomposition of a matrix, with the understanding that the values of the diagonal matrix can be used to determine and quantify the positive definiteness of the matrix. An alternative decomposition is the $[L][D][L]^T$ factorization for symmetric matrices, which is often used to find solutions of linear systems. With this decomposition applied to the tangent stiffness matrix of a structure, it is simple to determine a perturbation vector $\delta\boldsymbol{U}$ that exploits any negative values in the matrix $[D]$, and therefore any instability in the structure at its current state. The inner product of Eq. (6.15) can then be used as a scalar criterion for detecting and quantifying instability, and applying the constraint of buckling resistance to a structure.

The proposed approach is as follows. Let the global nonlinear stiffness matrix of a structure be denoted $[K]$, and find the symmetric $[L][D][L]^T$ factorization of the matrix at a given state of loading/deformation. Then the inner product of Eq. (6.15) becomes

$$\delta\boldsymbol{U}^T[K]\delta\boldsymbol{U} = \delta\boldsymbol{U}^T[L][D][L]^T\delta\boldsymbol{U} \tag{6.20}$$

If a perturbation vector $\delta\boldsymbol{U}$ is then defined such that

$$[L]^T\delta\boldsymbol{U} = \begin{cases} D_{ii} \; in \; rows \; where \; D_{ii} < \; 0 \\ \\ 0 \; in \; rows \; where \; D_{ii} \geq 0 \end{cases} \tag{6.21}$$

then the inner product $\delta\boldsymbol{U}^T[K]\delta\boldsymbol{U}$ will be the sum of all the negative elements in the diagonal matrix $[D]$ cubed, effectively quantifying the instability at the present state and providing a scalar buckling criterion. Since $[L]^T$ is upper triangular, $\delta\boldsymbol{U}$ can be found with simple back-substitution.

There is a caveat to this approach when it comes to evaluating sensitivities. The perturbation vector is a function of the factorization of the tangent stiffness matrix, and by extension a function of the design variables $\boldsymbol{D}$. However, there is no known cost-effective way to obtain the sensitivities of the full matrix factorization, even with an adjoint-based approach. The sensitivity of the buckling criterion must then be approximated by treating $\delta\boldsymbol{U}$ as locally constant at any given state, while examining the sensitivity of the stiffness matrix itself. Under this assumption the sensitivity of the buckling criterion with respect to a design variable D_i becomes

$$\frac{d}{dD_i}(\delta\boldsymbol{U}^T[K]\delta\boldsymbol{U}) \approx \delta\boldsymbol{U}^T\left[\frac{d[K]}{dD_i}\right]\delta\boldsymbol{U} \tag{6.22}$$

Once $\delta\boldsymbol{U}$ is obtained, the sensitivity of the buckling criterion can be approximated with just the sensitivity of the stiffness matrix, which is inexpensive and easily implemented using the tools developed in AStrO. Using this approach, only a single matrix factorization and the differentiation of the stiffness matrix by design variables is required at each design state. Not only is it cost-effective, but it avoids the potential difficulties with repeated eigenvalues or the uncertainty in the appropriate number of eigenpairs to consider.

The critical question is whether the approximation in the sensitivity calculation is acceptable. Ferrari et al. [22] investigated the issue of approximate or inconsistent sensitivities in gradient-based topology optimization with eigenmodal buckling constraints. They compared optimization results using exact sensitivities with those obtained when a certain term in the sensitivity calculation was omitted. It was found that in some cases, the use of inconsistent sensitivities led to suboptimal results or caused convergence problems in optimization. Yet in other cases there was little influence on the results.

A similar observation could be made in other well-known applications. Take, for example, modified Newton-Raphson iterations as opposed to full Newton-Raphson iterations for the solution of nonlinear systems. In many cases, neglecting to re-evaluate the Jacobian matrix on every iteration can cut down computational cost, and still successfully obtain the desired solution. There can, however, be a greater risk of divergence due to the inaccuracy of the local derivatives.

It would seem to be a reasonable hypothesis that the potential difficulties with the use of inconsistent sensitivities in gradient-based optimization is an important thing to keep in mind when exploring alternative methods, but that concern should not discourage the investigation of an approach that could have considerable benefits over more common methods. This study seeks to investigate the proposed alternative approach to applying buckling constraints in gradient-based optimization, and identify strengths and weaknesses compared with standard eigenpair-based methods.

Because this investigation requires data structures and procedures that have not been used in any previous study, the two approaches were first tested on a basic flat plate geometry before moving on to a more meaningful application in aircraft design. The details and results are presented in the next two sections.

6.2.2 Test of Buckling Constraint Methods on Flat Plate in Compression

As a preliminary test of the LDL^T method of applying structural buckling constraints proposed in the previous section, as well as the eigenvalue-based approach, consider a square flat plate divided into four equal sections, clamped on one end with a uniform compressive loading applied to the other, as illustrated in Fig. 6.20.

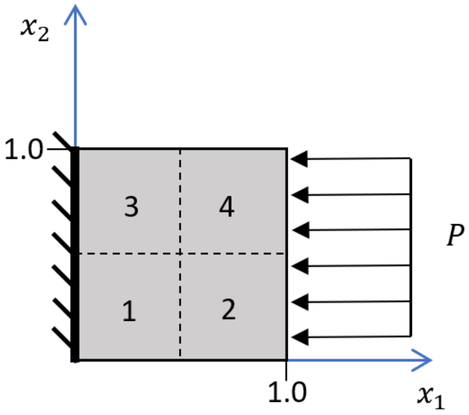


Figure 6.20 Four-sectioned clamped flat plate in compression.

For such a configuration, if all four sections have the same uniform material properties and thickness, the critical load to cause buckling is known from analytical theory to be

$$P_{cr} = \frac{EI\pi^2}{4L^2} \tag{6.23}$$

where E is Young's modulus of the material, I is the second moment of area about the buckling axis, and L is the length in the direction of loading. In the present study, the plate material was taken to be aluminum, with Young's modulus $E = 73.7$ GPa, and the distributed load was set to $P = 1.90\ (10^6)$ N. Under these conditions, buckling of the first mode should occur at a uniform plate thickness of 5 cm or less. The two buckling constraint methods of interest were tested by letting the thickness of each of the four sections of the plate be design variables, and attempting to minimize the total mass/volume of the plate while ensuring structural stability.

Since the accuracy of the objective sensitivities using the LDL^T method are a potential concern, the sensitivities were obtained using the adjoint method and compared with complex differentiation with the thickness of the four sections set at 49 mm, 48 mm, 47 mm, and 46 mm respectively. The results are shown in Table 6.3.

Table 6.3 Objective sensitivities of flat plate using LDL^T buckling criterion.

	Sensitivities		**Unit Direction**	
Section	**Adjoint**	**Complex**	**Adjoint**	**Complex**
1	-4.97325E+06	-1.49197E+07	-7.173427412188**40**E-01	-7.173427412188**84**E-01
2	-1.11251E+06	-3.33753E+06	-1.604687353993**71**E-01	-1.604687353993**92**E-01
3	-4.54578E+06	-1.36374E+07	-6.55685295716**613**E-01	-6.55685295716**418**E-01
4	-1.19571E+06	-3.58714E+06	-1.724701990518**47**E-01	-1.724701990518**52**E-01

In this case, the sensitivities are inconsistent in magnitude by a factor of three, but the gradient direction of the adjoint sensitivity is correct to nearly machine precision. This result is not guaranteed in all cases, as will be seen in the next section. The reason it is observed in this case, and the conditions under which it could be expected is an intended topic of future investigation. In the context of gradient optimization, having the correct sensitivity direction is generally the most critical result, and depending on the optimization algorithm the inaccuracy of magnitude may not even be of any significance.

A line search optimization was performed to minimize the mass of the flat plate while ensuring structural stability using the LDL^T buckling criterion, and the eigenvalue-based criterion. The objective for the LDL^T criterion was defined as

$$L = \sum_{i=1}^{n_{els}} Vol_i - \delta \boldsymbol{U}^T [K] \delta \boldsymbol{U} \tag{6.24}$$

For the eigenvalue criterion, the buckling constraint was expressed in terms of an exponential aggregation function of the lowest ten eigenvalues, based on classical Kreisselmeier–Steinhauser (KS) aggregation function [98]. This type of approach is typical with generalized buckling analysis, although there are many variations of the aggregation function and the number of eigenvectors used. The objective for the eigenvalue criterion was defined as

$$L = \sum_{i=1}^{n_{els}} Vol_i + c \sum_{j=1}^{n_{vals}} e^{-2\kappa\lambda_j} \tag{6.25}$$

with c and κ set to 0.05 and 3 respectively, based on the specifications of this problem. Adjoint-based sensitivities were obtained as outlined in Chapter 4, Section 4.3.2.

For both criteria, the flat plate was initialized to a uniform thickness of 6 cm, and the line search optimization was run until termination for failure to find a suitable design step. The progression of the objectives for both cases are plotted in Fig. 6.21.

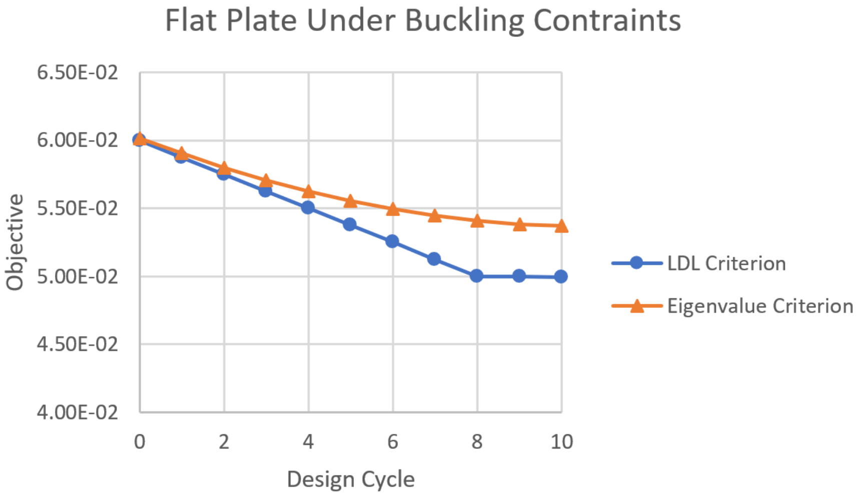


Figure 6.21 Optimization history of flat plate for volume minimization subject to buckling constraints.

A fundamental difference between the two buckling constraint methods is that the LDL^T criterion is defined so that it is exactly zero for a structure in a state void of any buckling risk. Although it is a smooth function when the tangent stiffness matrix becomes indefinite, it tends to produce rather abrupt changes in the objective when a critical buckling state is reached, as is evident from the curve in Fig. 6.21. An appropriate scaling parameter may alleviate this effect, but it can be somewhat challenging to identify a suitable value before running an optimization. In contrast, the eigenvalue criterion formed from KS aggregation has a constant presence, albeit a small one, even in a state far from any risk of buckling. Its influence gradually gains prominence as the structure approaches a critical buckling state, leading to the curved level-off of the objective seen in Fig. 6.21.

The LDL^T criterion produced a slightly lower final volume than the eigenvalue criterion, but both methods drove sections 1 and 3, the sections on the clamped end of the plate, to a higher thickness than sections 2 and 4 on the loaded end. The difference is more pronounced in the results for the eigenvalue method, as can be seen in Table 6.4. Both solutions ended up converging around the known critical state at a uniform thickness of 5cm, and both solutions are buckling-safe according to positive definiteness of the nonlinear stiffness matrix. Based on this initial test, it would seem that the LDL^T criterion is at least potentially effective, and is worth investigating further in more meaningful applications.

Table 6.4 Final design configuration of flat plate subject to buckling constraints.

Quantity	**LDL^T Criterion**	**Eigenvalue Criterion**
Sec. 1 thickness	5.0434E-02	5.6537E-02
Sec. 2 thickness	4.8023E-02	4.3509E-02
Sec. 3 thickness	5.0434E-02	5.6537E-02
Sec. 4 thickness	4.8023E-02	4.3509E-02
Total volume	4.9229E-02	5.0023E-02

6.2.3 Optimization of Truss Braced Wing

A design trend that has been receiving much interest in the aviation community lately is the truss-braced wing design for fixed-wing aircraft, as depicted in Fig. 6.22. This configuration is intended to reduce bending loads at the root of the wing, making it possible to implement more slender, high aspect ratio wing designs in the interest of improved aerodynamic performance and fuel efficiency. While there is potentially much to be gained from the truss-braced wing design, it also invites a two-fold increase in the risk of structural buckling. Not only are higher aspect ratio wings inherently more prone to instability, but the tension in the truss below the wing with the aerodynamic lift distribution creates a combined compressive and bending load on the interior section of the wing, which is a condition of high risk for buckling. It then becomes more important than ever to use care in the design to ensure that structural stability is enforced.

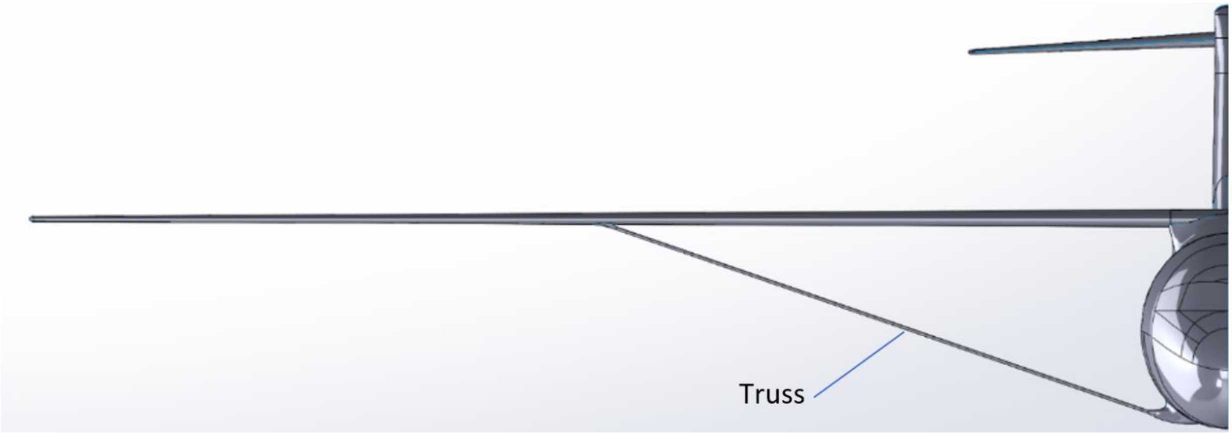


Figure 6.22 Conceptual illustration of next-generation truss-braced wing design.

A finite element model of a wing box for a truss-braced configuration was constructed and used as the test case to compare methods for applying buckling constraints in gradient-based optimization. The wing box cross section was fit to the NASA Common Research Model (CRM) airfoil [97] from 10% chord to 60% chord. The wingspan was set at 25.9 meters, with a quarter-chord sweep angle of 12.5^{o}, and a chord dimension ranging linearly from 3.3 meters at root to 1.2 meters at tip. Figure 6.23 visually depicts the overall wing box design.

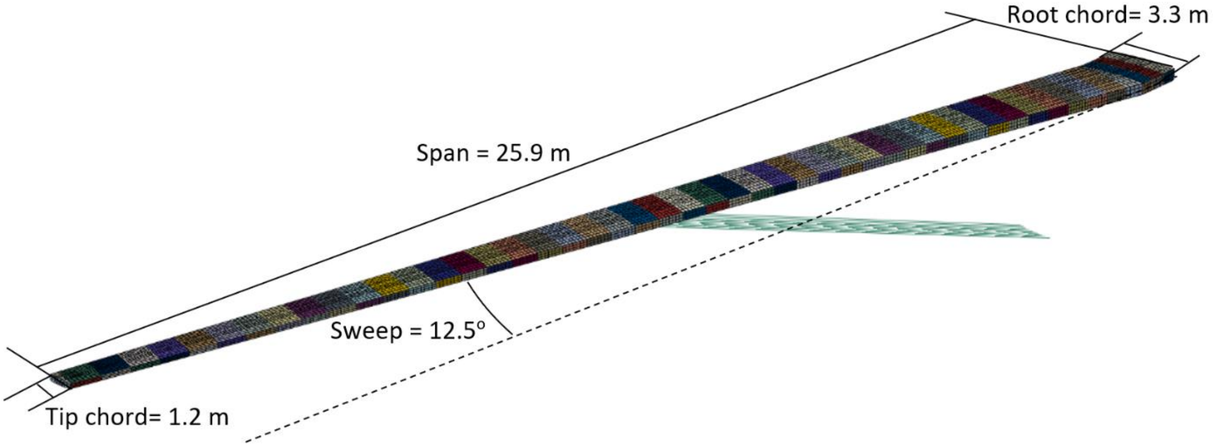


Figure 6.23 Overall design of the truss-braced wing finite element model.

The wing box model was divided into 48 spanwise sections, each with an upper panel, a lower panel, a leading edge spar section and a trailing edge spar section. Ribs run through the interior of the wing in the chord direction at the joining of the spanwise sections. All sections of the wing box are composed of four-node shell elements, with a row of two-node beam elements connecting the lower panel midway through the span to the underside of the fuselage, to simulate the structural effect of the truss. Figure 6.24 shows the interior design of the wing box model.

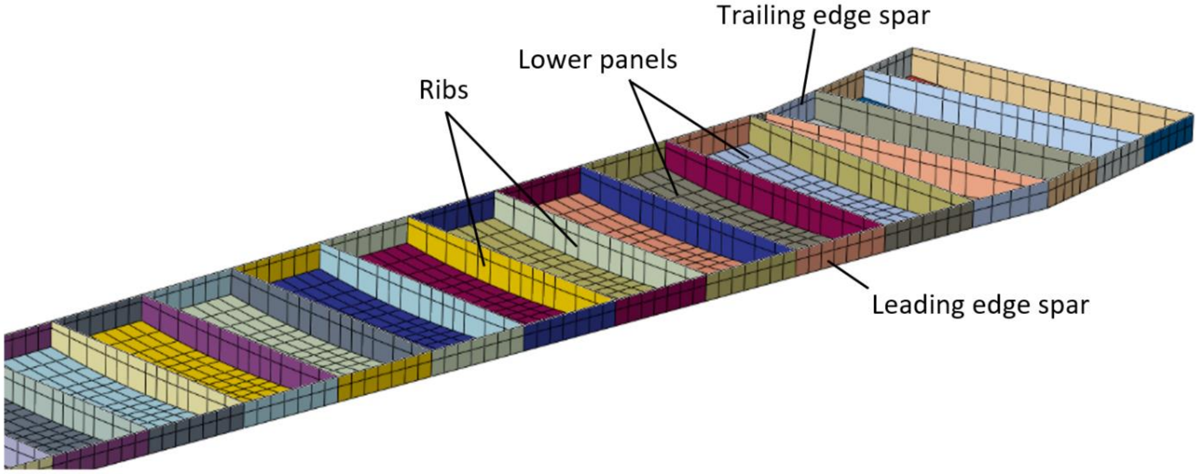


Figure 6.24 Interior structure of truss-braced wing box model.

It is a common objective in structural wing design to minimize the total mass/weight of the wing subject to constraints on maximum stress and buckling criteria under some target loading or operating condition. This was the objective chosen for the present study as well. Specifically, the goal was to minimize mass/weight by modifying the shell panel thicknesses, under an elliptic wing loading distribution as illustrated in Fig. 6.25. The loading was scaled to produce a total lift of 300 kN, or half the approximate weight of a commercial aircraft of this scale.

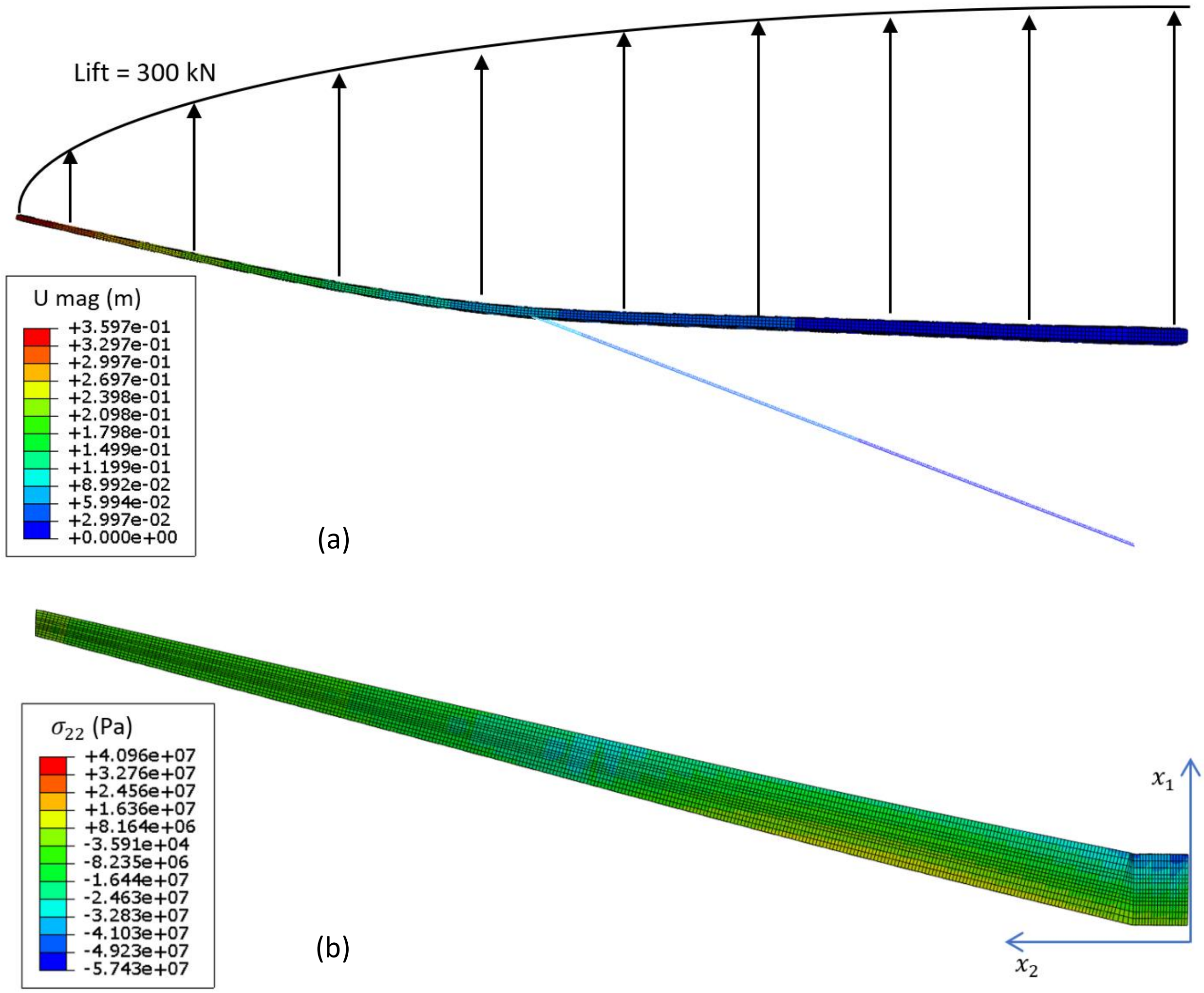


Figure 6.25 Response of (a) displacement magnitude and (b) normal stress in the span direction of truss-braced wing under elliptic loading in initial configuration of 40 mm uniform panel thickness.

Two optimization cases were run, one using the LDL^T buckling criterion and one using the eigenvalue criterion, as was done with the flat plate test case in the previous section. The objective function was defined in a similar fashion, except with an added constraint on the maximum von Mises stress not to exceed 345 MPa, the approximate yield strength of aluminum. This constraint was also applied with a KS aggregation-based approach. For the LDL^T criterion, the objective was defined as

$$L = \sum_{i=1}^{n_{els}} Vol_i - \delta \boldsymbol{U}^T [K] \delta \boldsymbol{U} + h \frac{1}{(Total\, Vol)} \sum_{k=1}^{n_{els}} e^{2\left(\frac{\sigma_{v,k}}{\sigma_m}\right)} Vol_k \tag{6.26}$$

where $\sigma_{v,k}$ is von Mises stress at the centroid of element k, σ_m is the maximum allowable stress, in this case 345 MPa, and h is a scaling parameter, set to be 0.5 in the present context. For the eigenvalue criterion, the objective was defined as

$$L = \sum_{i=1}^{n_{els}} Vol_i + c \sum_{j=1}^{n_{vals}} e^{-2\kappa\lambda_j} + h \frac{1}{(Total\, Vol)} \sum_{k=1}^{n_{els}} e^{2\left(\frac{\sigma_{v,k}}{\sigma_m}\right)} Vol_k \tag{6.27}$$

where λ_j are the applicable eigenvalues, in this case the lowest ten, of the tangent stiffness matrix and κ is a scaling parameter set at 0.3. The thicknesses for the upper and lower panels, spars and ribs of every section of the wing were defined as the design variables, such that

$$thickness\ of\ section\ i = t_{i,0}(1 + D_i) \tag{6.28}$$

where $t_{i,0}$ is the initial thickness of section i. All panels were set to an initial thickness of 40 mm, which is far out of range of any violation of buckling or stress constraints. With aluminum construction, this gives the wing a total initial mass of 8,500 kg.

Before proceeding with the optimization, the objective sensitivities were obtained for the LDL^T criterion using the adjoint and complex differentiation, with the thickness of all panels set at 1 mm to ensure that the wing would be at risk for buckling. The results for the first five design variables are shown in Table 6.5. Clearly, the degree of agreement in the gradient direction seen for the flat plate test is not present in this case, confirming that consistency in direction is not a general result. However, the angle of discrepancy at this state is 8.68°, which could at least be argued to be within a reasonable range. How much discrepancy is permissible is, of course, ambiguous and highly case dependent, but that discussion is beyond the present context.

Table 6.5 Objective sensitivities for truss-braced wing at 1mm panel thickness. Angle of difference between adjoint and complex $\theta = 8.68^{o}$.

	Sensitivities		**Unit Direction**	
Section	**Adjoint**	**Complex**	**Adjoint**	**Complex**
1	5.8177E+24	3.6156E+26	8.3054E-01	7.6316E-01
2	3.3838E+24	2.3923E+26	4.8308E-01	5.0496E-01
3	1.7267E+24	1.5660E+26	2.4650E-01	3.3055E-01
4	5.5874E+23	6.6266E+25	7.9766E-02	1.3987E-01
5	6.9036E+23	8.7057E+25	9.8557E-02	1.8376E-01

The optimization was performed on the truss-braced wing using both buckling constraint methods as described, and the objective progression history is plotted in Fig. 6.26. Both cases brought the objective down steadily for several design cycles before halting at the onset of a constraint violation. The gradual decrease in slope for the eigenvalue criterion is less visible in this case, but some subtle evidence of it can be seen on the last two cycles. The final wing mass was 4300 kg for the LDL^T criterion, and 3400 kg for the eigenvalue criterion. A plot of the final panel thickness distributions is given in Fig. 6.27. There are some qualitative similarities in the final configurations for the two methods. They both tend to generally focus on the larger panels of the wing, as they have the strongest influence on total mass. There are notable differences in the panels of the center box section of the wing, perhaps because these are the regions of highest compressive stress, as indicated in Fig. 6.25 (b).

Both optimization cases effectively satisfied both the stress and buckling constraints, but the LDL^T method is substantially less costly to implement. In both methods the cost of obtaining sensitivities via the adjoint method is independent of the number of design variables, but using the eigenvalue criterion requires roughly the cost of the LDL^T method for each eigenpair considered. In practice this could translate to a 10X to 100X speedup, depending on the number of eigenpairs necessary. The eigenvalue criterion did produce a lower structural weight in this case, and may yet have some advantages, but few generalizations can be made from these results alone. The present study is an initial test run of the feasibility of the LDL^T criterion, and calls for much further investigation, as discussed in the conclusions.

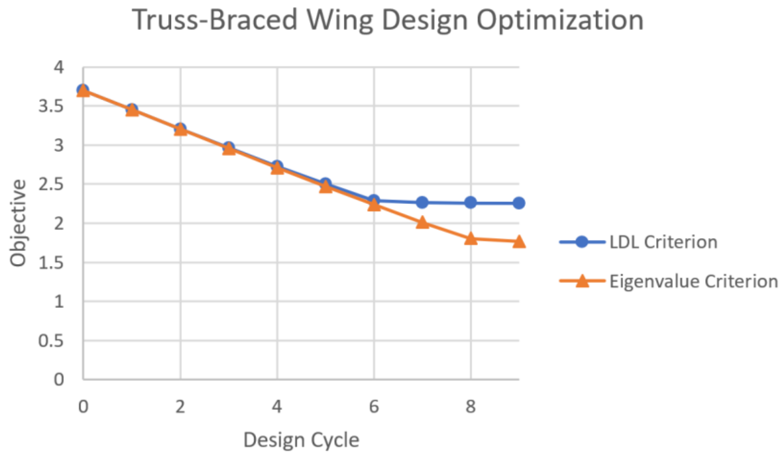


Figure 6.26 Objective optimization history for truss-braced wing.

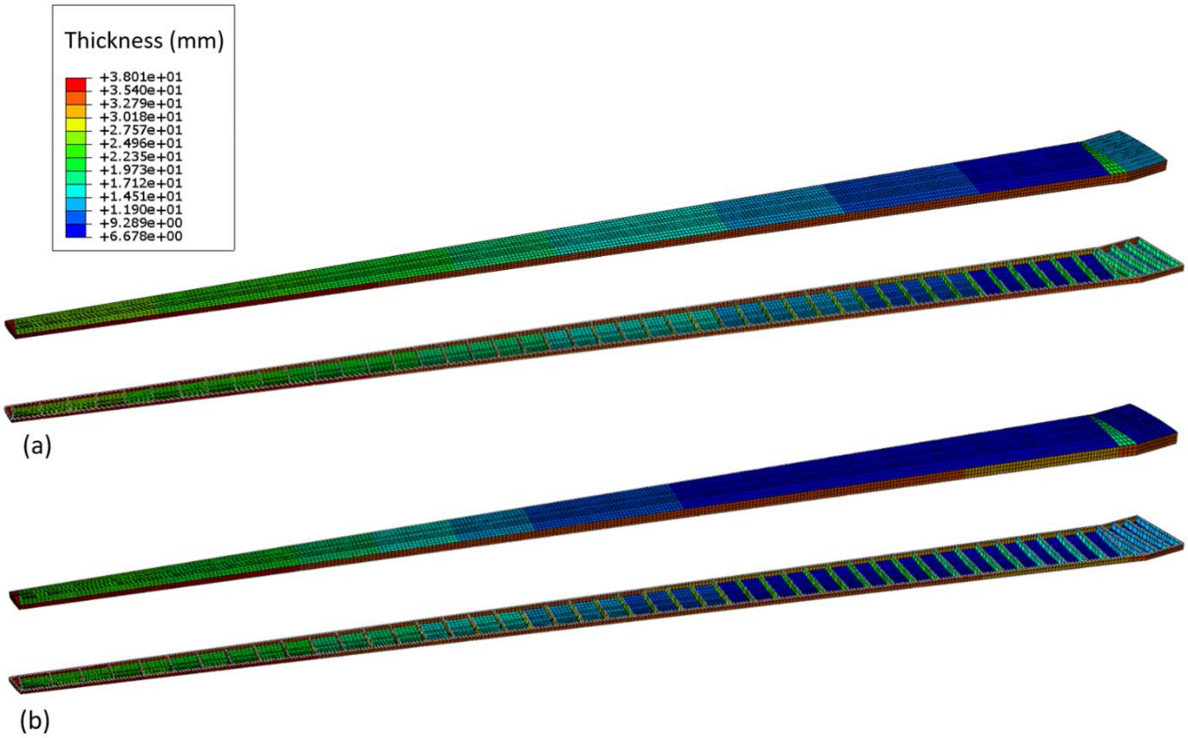


Figure 6.27 Panel thickness distributions for (a) LDL^T buckling criterion and (b) eigenvalue criterion.

Chapter 7

Conclusions

A high-fidelity, open-source package for the finite element thermoelastic modeling and adjoint-based sensitivity analysis of structures (AStrO) has been developed and demonstrated. Validation tests show the analysis results to be reliable within the assumptions of the fundamental formulations, and the adjoint-based sensitivities derived from the linearization of the analysis code to be consistent with direct differentiation of the results. The package is versatile for general applications, and has been successfully coupled with CFD codes for multidisciplinary analysis.

In this day and age, numerous codes and software packages exist for applications across physical disciplines, as well as for optimization and visualization. Many have undergone years of commercial testing and debugging, and there are those who would question the value of putting forth the effort to build an in-house code for disciplines in which commercial tools exist. The development of AStrO and the investigations conducted with it thus far have provided much insight on this question. Perhaps the most notable benefits of open-source tools are their capability to be tightly coupled for efficient multidisciplinary analysis, and to be linearized to produce exact adjoint-based sensitivities. But even beyond those general characteristics, having an open-source capability developed from the ground up such as AStrO has proven to have its advantages.

In the study of fatigue stress minimization on the SWiFT wind turbine blade, for example, the optimization objective was set to be the Fertig effective off-axis matrix stress criterion, which is a sophisticated formula that is not a standard metric in any commercial code. The flexibility to not only evaluate this criterion but also differentiate it exactly using the finite element solution for adjoint-based sensitivity analysis would not be a trivial task, if even possible with most commercial codes. The capabilities of AStrO made it possible to perform gradient-based optimization with this criterion which had not been done previously.

Similarly, the testing of the LDL^T buckling criterion required low-level manipulation of the data structures involved in the finite element equations, and the freedom to apply the adjoint in the highly specialized manner required for objectives derived from eigenvalues of the governing equation matrix. Not only having the open-source tools, but the intimate knowledge of the internal structure and organization of the code was vital to performing the investigation. The robustness and reliability of commercial codes and the flexibility of open-source tools complement each other well, and they both serve a valuable purpose in computational analysis and design.

With regard to the SWiFT wind turbine fatigue stress minimization study in general, all optimization runs successfully reduced the maximum objective stress by 18-60%. Using the adjoint method enabled analysis for 16,310 design variables at a cost of 9.8 seconds per static design cycle on a single core, compared to an estimated 3 to 45 hours per design cycle using finite difference or solving the tangent sensitivity problem, depending on the implementation. Although simplified assumptions were used in this study, the results provide some insight into the potential to improve fatigue life of wind turbine blades through composite optimization. If implemented, substantial gains could possibly be achieved without adding any mass, material cost or changing the external shape of the structure. However, results also suggest that optimal design is highly dependent on loading, and it is not clear how much benefit would actually be seen from optimizing over the full range of possible loading conditions, or how results would change for turbines of different size and design. It is evident that the design space chosen has a strong influence on the potential for reduction of objective stress, but that there may be an inevitable trade-off between effective stress reduction and preserving the ideal stiffness properties of a blade.

A number of recommendations for future work can be made from the observations of this study. For more meaningful and accurate results, fully coupled aeroelastic optimizations should be performed to capture accurate loading histories and account for fluid-structural interdependence. More realistic composite layup definitions should be defined, and reasonable constraints applied to see a more accurate picture of what might be gained from optimizations like these in practice. In the present study the objective was derived from the effective off-axis matrix stress associated with the propagation of fatigue damage, but it may be worth setting fatigue life itself as the objective using a model approach like that of Bhuiyan *et al.* [88], to confirm the correlation with the effective stress. The present work demonstrates AStrO as a capable tool for further studies following the above recommendations.

A few general conclusions can be drawn about the use of the proposed LDL^T buckling method in gradient-based optimization. The method is effective at detecting instability and the threat of buckling in any general structure, requiring no more information or input than is necessary for basic finite element analysis. It is cost-effective, requiring only a single matrix factorization and differentiation of the nonlinear stiffness matrix at any given design state, making it significantly cheaper than eigenvalue-based analysis with multiple eigenpairs. The use of the LDL^T criterion does not produce exact objective sensitivities, and although all cases observed showed reasonable if not exact agreement in gradient direction, the lack of consistency could prove problematic for some applications, or for algorithms where the magnitude of the sensitivity vector is crucial. The buckling criterion is a smooth, differentiable function but it has a tendency to cause abrupt changes in the objective function around critical buckling states.

The optimization results of the truss-braced wing are a reasonable first attempt, but it is likely that a more sophisticated algorithm could produce better results than the steepest descent line search used in this study. Further investigation using such algorithms would be a suitable next step. There is also potential to investigate numerous variations on aspects like choice of aggregation functions and parameters in the constraint definitions, number of eigenvalues used, etc. A great deal more work would need to be done to understand the method’s advantages and limitations fully, but there is much to be gained from the pursuit, not only for next-generation aircraft design but for other advanced applications as well.

In reflection on what has been learned so far and consideration of upcoming goals, some recommendations can be made for next steps and future work. From the beginning, one of the main purposes for which AStrO was intended was for high-fidelity multidisciplinary optimization. Completing the linearization and adjoint-based sensitivities for the fluid-structure interface will allow for the true fully coupled optimization that is required for the most meaningful and accurate results. If necessary, for high aspect ratio wing applications which may have difficulty converging with the current FSI approach, the flow and structural solver may be wrapped into a single system of nonlinear equations, and solved with a method like GMRES using the disciplinary solvers as block preconditioning.

AStrO currently has no parallel implementation, and although that has been a long-term goal from the beginning, the priority to date has been on development of the fundamental tools. Now, particularly after the investigations in generalized buckling analysis, the value of a parallel implementation is becoming more apparent. The operations in eigenvalue-based optimization are costly, but also highly parallelizable, and further studies in methods of buckling analysis could be expedited by such an advancement.

There is also growing interest among colleagues for structural applications that go beyond some of the general assumptions of AStrO's formulations, such as hypersonic flow and high-speed ballistic applications. These would require not only nonlinear geometry, but nonlinear material behavior as well, and the appropriate modifications would need to me made in the matrices resulting from the discretization of the governing equations, and subsequently the formulations for the adjoint-based sensitivities.

These are only a few possibilities for future advancement. The hope is that whatever directions research may take in the years ahead, AStrO will provide a versatile toolset for modeling and optimization, and enable future contributions in computational simulation and design.

References

[1] Gustafsson, B., *Scientific Computing: A Historical Perspective*, Springer, 2018.
doi: 10.1007/978-3-319-69847-2

[2] Jameson, A., “Aerodynamic Shape Optimization Using the Adjoint Method,” *VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics,* Rhode St. Genese, Belgium, 2003.

[3] Sobieszczanski-Sobieski, J., and Haftka, R.T., “Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments,” *Structural Optimization,* Vol. 4, No. 1, 1997, pp. 1-23.
doi: 10.1007/BF01197554

[4] Chattopadhyay, A., Pagaldipti, N., “A Multidisciplinary Optimization Using Semi-Analytical Sensitivity Analysis Procedure and Multilevel Decomposition,” *Computers & Mathematics with Applications,* Vol. 1, No. 7, 1995, pp. 55-66.
doi: 10.1016/0898-1221(95)00018-T

[5] Baker, M., and Giesing, J., “A Practical Approach to MDO and its Application to an HSCT Aircraft,” *1st AIAA Aircraft Engineering, Technology and Operations Cong.* CP95-3885, 1995.
doi: 10.2514/6.1995-3885

[6] Grossman, B., Strauch, G., Eppard, W.H., Gurdal, Z., and Haftka, R.T., “Integrated Aerodynamic/Structural Design of a Sailplane Wing,” *Journal of Aircraft,* Vol. 25, No. 9, 1988 pp. 855-860.
doi: 10.2514/3.45670

[7] Dudley, J.M., Huang, X., Haftka, R.T., Grossman, B., and Mason, W.H., “Variable Complexity Interlacing of Weight Equations and Structural Optimization for the Design of the High Speed Civil Transport,” *5th AIAA Multidisciplinary Analysis and Optimization Symposium,* CP94-4377, 1994, pp. 1129-1134.

[8] Lyu, Z., Kenway, G. K., Paige, C., and Martins, J. R. R. A., “Automatic Differentiation Adjoint of the Reynolds-Averaged Navier–Stokes Equations with a Turbulence Model,” *21st AIAA Computational Fluid Dynamics Conference,* San Diego, CA, Jul. 2013.
doi:10.2514/6.20132581

[9] Kennedy, G. J., Kenway, Gaetan W., and Martins, J. R. R. A., “High Aspect Ratio Wing Design: Optimal Aerostructural Tradeoffs for the Next Generation of Materials,” *52nd Aerospace Sciences Meeting,* CP0596, 2014.
doi: 10.2514/6.2014-0596

[10] Brooks, T., Kenway, G., and Martins, J. R. R. A., “Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings,” *AIAA Journal,* Vol. 56, No. 7, 2018.
doi: 10.2514/1.J056603

[11] Kitson, R. C., Lupp, C., and Cesnik, C. E., “Modeling and Simulation of Flexible Jet Transport Aircraft with High-Aspect-Ratio Wings,” *AIAA SciTech Forum, American Institute of Aeronautics and Astronautics*, Jan. 2016, pp. –. doi:10.2514/6.2016-2046.

[12] Brooks, T. R., Kennedy, G. J., and Martins, J. R. R. A., “High-fidelity Multipoint Aerostructural Optimization of a High Aspect Ratio Tow-steered Composite Wing,” *Proceedings of the 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, Grapevine, TX, January 2017.
doi:10.2514/6.2017-1350.

[13] Somers, D. M., “An Exploratory Investigation of a Slotted, Natural-Laminar-Flow Airfoil,” *NASA/CR-2012/217560,* 2012.

[14] Somers, D. M. and Maughmer, M. D., "Design and Experimental Results for the S414 Airfoil," *U.S. Army Research, Development and Engineering Command, RDECOM TR 10-D-112,* August 2010.

[15] Whitcomb, R. T. and Clark, L. R., "An Airfoil Shape for Efficient Flight at Supercritical Mach Numbers," *NASA-TM-X-1109,* 1965.

[16] Somers, D. M., "Laminar-Flow Airfoil," U.S. Patent 6,905,092 B2, June 2005.

[17] Coder, J. G., Maughmer, M. D., and Somers, D. M., "Theoretical and Experimental Results for the S414, Slotte d, Natural-Laminar-Flow Airfoil," *Journal of Aircraft*, Vol. 51, No. 6, 2014, pp. 1883-1890.
doi: 10.2514/6.2013-2407

[18] Maughmer, M. D., Premi, A., and Somers, D. M., "Exploration of a Slotted Laminar-FlowControl Airfoil Concept," *Final Technical Report, NASA Grant No. NNX13AB86A*, Nov. 2013.

[19] Langtry, R. B. and Menter, F. R., "Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes," *AIAA Journal*, Vol. 47, No. 12, 2009, pp. 2894-2906.
doi: 10.2514/1.42362

[20] Coder, J. G. and Maughmer, M. D., "Computational Fluid Dynamics Compatible Transition Modeling Using an Amplification Factor Transport Equation," *AIAA Journal*, Vol. 52, No. 11, 2014, pp. 2506-2512.
doi: 10.2514/1.J052905

[21] Rajeev, S., and Krishnamoorthy, C. S., "Discrete Optimization of Structures Using Genetic Algorithms," *Journal of Structural Engineering,* Vol. 118, No. 5, 1992,
doi: 10.1061/(ASCE)0733-9445(1992)118:5(1233)

[22] Ferrari, F., and Sigmund, O., "Revisiting Topology Optimization with Buckling Constraints," *Structural and Multidisciplinary Optimization*, Vol. 59, 2019, pp. 1401-1415.
doi: 10.1007/s00158-019-02253-3

[23] Gu, G. X., Chen, C. T., and Buehler, M. J., "De Novo Composite Design Based on Machine Learning Algorithm," *Extreme Mechanics Letters,* Vol.18, 2018, pp. 19-28.
doi: 10.1016/j.eml.2017.10.001

[24] Hicks, R. M., Murman, E. M., and Vanderplaats, G. N., "An Assesment of Airfoil Design by Numerical Optimization," *NASA TM X-3092*, Ames Research Center, Moffett Field, California, July 1974.

[25] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft,* Vol. 15, 1978, pp. 407-412.
doi: 10.2514/3.58379

[26] Borland, C. J., Benton, J. R., Frank, P. D., Kao, T. J., Mastro, R. A., and Barthelemey, J. F. M., "Multidisciplinary Design Optimization of a Commercial Aircraft Wing – An Exploratory Study," *5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* CP94-4305, Vol. 1, 1994, pp. 505-519.

[27] Lions, J. L., *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971.
doi: 10.1112/blms/4.2.236

[28] Pironneau, O., *Optimal Shape Design for Elliptic Systems,* Springer-Verlag, New York, 1984.
doi: 10.1007/BFb0006123

[29] Jameson, A., "Optimum Aerodynamic Design Using CFD and Control Theory," *12th Computational Fluid Dynamics Conference,* CP95-1729, 1995.
doi: 10.2514/6.1995-1729

[30] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing,* Vol. 3, 1988, pp. 233-260
doi: 10.1007/BF01061285

[31] Jameson, A., and Alonso, J.J., "Automatic Aerodynamic Optimization on Distributed Architectures," *34th Aerospace Sciences Meeting and Exhibit*, CP96-0409, 1996.
doi: 10.2514/6.1996-409

[32] Jameson, A., "Re-Engineering the Design Process Through Computation," *Journal of Aircraft,* Vol. 36, No. 1, 1999.
doi: 10.2514/2.2412

[33] Jameson, A., "Computational Aerodynamics for Aircraft Design," *Science,* Vol. 245, No. 4916, 1989, pp. 361-371.
doi: 10.1126/science.245.4916.361

[34] Jameson, A., "Automatic Design of Transonic Airfoils to Reduce the Shock Induced Pressure Drag," *Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics,* 1990, pp. 5-17.

[35] Jameson, A., "Optimum Aerodynamic Design via Boundary Control," *AGARD-VKI Lecture Series, Optimum Design Methods in Aerodynamics*. von Karman Institute for Fluid Dynamics, 1994.

[36] Reuther, J., Jameson, A. Alonso, J. J., Rimlinger, M.J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers," *Journal of Aircraft,* Vol. 36, No. 1, 1999.
doi: 10.2514/2.2413

[37] Reuther, J., Alonso, J. J., Vassberg, J. C., Jameson, A., and Martinelli, L., "An Efficient Multiblock Method for Aerodynamic Analysis and Design on Distributed Memory Systems," *13th Computational Fluid Dynamics Conference,* CP97-1893, 1997
doi: 10.2514/6.1997-1893

[38] Jameson, A., Pierce, N. and Martenelli, L., "Optimum Aerodynamic Design Using the Navier-Stokes Equations", *Theoretical Computational Fluid Dynamics,* Vol. 10, 1998, pp. 213-237.
doi: 10.1007/s0016200500

[39] Mavriplis, Dimitri, "Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes," *AIAA Journal,* Vol. 45, No. 4, April 2007, pp. 741-750.
doi: 10.2514/1.22743

[40] Mishra, A., Mani, K., Mavriplis, D. and Sitaraman, J., "Time-Dependent Adjoint-Based Aerodynamic Optimization For Coupled Fluid-Structure Problems," *Journal of Computational Physics,* Vol. 292, July 2015, pp. 253-271.
doi: 10.1016/j.jcp.2015.03.010

[41] Jameson, A., "Aerodynamic Shape Optimization Using the Adjoint Method," *VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics,* Rhode St. Genese, Belgium, 2003.

[42] Kenway, G. K. W. and Martins, Joaquim R. R. A., "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft,* Vol. 51, No. 1, 2014, pp. 144-160.
doi: 10.2514/1.C032150

[43] Vanek, F. M., and Albright, L. D., *Energy Systems Engineering: Evaluation and Implementation,* McGraw-Hill, 2008.
doi: 10.1036/0071495932

[44] Balas, G. J. and Doyle, J. C., "Robustness and Performance Trade-Offs in Control Design for Flexible Structures", *IEEE Transactions on Control Systems Technology,* Vol. 2, No. 4, 1994, pp. 352-361.
doi: 10.1109/87.338656

[45] Lobitz, D. W., and Veers, P. S., "Load Mitigation with Bending/Twist-Coupled Blades on Rotors Using Modern Control Strategies," *Wind Energy,* Vol. 6, 2003, pp. 105-117.
doi: 10.1002/we.74

[46] Vesel, R. W., and McNamara, Jack J., "Performance Enhancement and Load Reduction of a 5 MW Wind Turbine Blade," *Renewable Energy,* Vol. 66, 2014, pp. 391-401.
doi: 10.1016/j.renene.2013.12.019

[47] Zhou, X., An, L., Wang, Z., and Zhao, H., “Bend-Twist Coupling Characteristics Calculations of 5 MW Wind Turbine Blades Based on 3D Shell Model,” *2015 International Conference on Applied Science and Engineering Innovation.* 2015.
doi: 10.2991/asei-15.2015.396

[48] Maheri, A., Noroozi, S., and Vinney, J., “Combined Analytical/FEA-Based Coupled Aero Structure Simulation of a Wind Turbine with Bend-Twist Adaptive Blades,” *Renewable Energy,* Vol. 32, No. 6, 2007, pp. 916-930.
doi: 10.1016/j.renene.2006.04.007

[49] Murray, R. E., Doman, D. A., and Pegg, M. J., “Finite Element Modeling and Effects of Material Uncertainties in a Composite Laminate with Bend-Twist Coupling,” *Composite Structures,* Vol. 121, 2015, pp. 362-376.
doi: 10.1016/j.compstruct.2014.11.035

[50] Oest, J., Sorensen, R., Overgaard, L. C. T., and Lund, E., “Structural Optimization with Fatigue and Ultimate Limit Constraints of Jacket Structures for Large Offshore Wind Turbines,” *Structural and Multidisciplinary Optimization,* Vol. 55, No. 3, 2017, pp. 779-793.
doi: 10.1007/s00158-016-1527-x

[51] Hashin, Z. and Rotem, A., “A Fatigue Failure Criterion for Fiber Reinforced Materials,” *Journal of Composite Materials,* Vol. 7, No. 4, 1973, pp. 448-464.
doi: 10.1177/002199837300700404

[52] Talreja, R., “Fatigue of Composite Materials: Damage Mechanisms and Fatigue-Life Diagrams,” *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences,* Vol. 378, The Royal Society, 1981.
doi: 10.1098/rspa.1981.0163

[53] Coleman, B. D., “Time Dependence of Mechanical Breakdown Phenomena,” *Journal of Applied Physics,* Vol. 27, No. 8, 1956, pp. 862-866.
doi: 10.1063/1.1722504

[54] Regel, V., Leksovskii, A. M., and Slutsker, A. I., “Polymer Breakdown and Fatigue,” *Mechanics of Composite Materials,* Vol. 8, No. 4, 1972, pp. 516-527.
doi: 10.1007/BF00855576

[55] Regel, V. R., and Tamuzh, V. P., “Fracture and Fatigue of Polymers and Composites (survey),” *Polymer Mechanics,* Vol. 13, No. 3, 1977, pp. 392-408.
doi: 10.1007/BF00859423

[56] Sauer, J. and Richardson, G., “Fatigue of Polymers,” *International Journal of Fracture,* Vol. 16, No. 6, 1980, pp. 499-532.
doi: 10.1007/BF02265215

[57] Bhuiyan, Faisal H. and Fertig, Ray S. III., “A Physics-Based Combined Creep and Fatigue Methodology for Fiber-Reinforced Polymer Composites,” *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum,* CP201, 2017.
doi: 10.2514/6.2017-0201

[58] Fertig III, R. S., “Bridging the Gap Between Physics and Large-Scale Structural Analysis: a Novel Method for Fatigue Life Prediction of Composites.” *SAMPE 2009 Fall Technical Conference, Wichita, KS,* 2009.

[59] Fertig III, R. S. and D. J. Kenik, “Predicting Composite Fatigue Life Using Constituent-Level Physics,” *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference,* CP 1991, AIAA, Denver, CO, AIAA, 2011.
doi: 10.2514/6.2011-1991

[60] Fertig III, R. S. and D. J. Kenik, “Physics-Based Fatigue Life Prediction of Composite Structures,” *NAFEMS World Congress 2011. Boston, MA,* 2011.

[61] Jensen, E. M. and Fertig, R. S., "Physics-Based Multiscale Creep Strain and Creep Rupture Modeling for Composite Materials," *AIAA Journal,* Vol. 54, No. 2, 2015, pp. 703-711.
doi: 10.2514/1.J054193

[62] Dunning, P. D., Ovtchinnikov, E., Scott, J., and Kim, A., "Level-Set Topology Optimization with Many Linear Buckling Constraints Using Efficient and Robust Eigensolver," *International Journal for Numerical Methods in Engineering,* Vol. 107, No. 12, 2016, pp. 1029-1053.
doi: 10.1002/nme.5203

[63] Olhoff, N., and Rasmussen, S. H., "On Single and Bimodal Optimum Buckling Loads of Clamped Columns," *International Journal of Solids and Structures,* Vol. 13, No. 7, 1977, pp. 605-614.
doi: 10.1016/0020-7683(77)90043-9

[64] Cox, S., and Overton, M., "On the Optimal Design of Columns Against Buckling," *SIAM Journal on Mathematical Analysis,* Vol. 23, No. 2, 1992, pp. 287-325.
doi: 10.1137/0523015

[65] Kirmser, P. G., and Hu, K. K., "The Shape of the Ideal Column Reconsidered," *The Mathematical Intelligencer,* Vol. 15, No. 3, 1993, pp. 62-67.
doi: 10.1007/BF03024261

[66] Khot, N. S., Venkayya, V. B., and Berke, L., "Optimum Structural Design with Stability Constraints," *International Journal of Numerical Methods in Engineering,* Vol. 10, No. 5, 1976, pp. 1097-1114.

[67] Szyszkowski, W., and Watson, L., "Optimization of the Buckling Load of Columns and Frames," *Engineering Structures,* Vol. 10, No. 4, 1988, pp. 249-256.
doi: 10.1016/0141-0296(88)90046-6

[68] Rozvany, G., "Difficulties in Topology Optimization with Stress, Local Buckling and System Stability Constraints," *Structural Optimization,* Vol. 11, 1996, pp. 213-217.
doi: 10.1007/BF01197036

[69] Bruyneel, M., Colson, B. and Remouchamps, A., "Discussion on some Convergence Problems in Buckling Optimization," *Structural and Multidisciplinary Optimization,* Vol. 35, No. 2, 2008, pp. 181-186.
doi: 10.1007/s00158-007-0129-z

[70] Seyranian, A. P., Lund, E. and Olhoff, N., "Multiple Eigenvalues in Structural Optimization Problems," *Structural Optimization,* Vol. 8, No. 4, 1994, pp. 207-227.
doi: 10.1007/BF01742705

[71] Mavriplis, D. J., Long, M., Lake, T. and Langlois, M., "NSU3D Results for the Second AIAA High-Lift Prediction Workshop," *Journal of Aircraft,* Vol. 52, No. 4, 2015, pp. 1063-1081.
doi: 10.2514/1.C033042

[72] Mavriplis, D., Anderson, E., Fertig, R. S. and Garnich, M., "Development of a High-Fidelity Time-Dependent Aero-Structural Capability for Analysis and Design," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference,* CP1175, 2016.
doi: 10.2514/6.2016-1175

[73] Marviplis, D., Fabiano, E. and Anderson, E., "Recent Advances in High-Fidelity Multidisciplinary Adjoint-Based Optimization with the NSU3D Flow Solver Framework," *55th AIAA Aerospace Sciences Meeting,* CP17-1669, 2017.
doi: 10.2514/6.2017-1669

[74] Kirby, A., Yang, Z. and Mavriplis, D., "Visualization and Data Analytics Challenges of Large-Scale High-Fidelity Numerical Simulations of Wind Energy Applications," *2018 AIAA Aerospace Sciences Meeting,* CP18-1171, 2018.

doi: 10.2514/6.2018-1171

[75] Abaqus, Software Package, Version 6.14-4, Dassault Systemes Simulia Corp., Providence, RI, USA, 2017.

[76] Hilber, H. M., Hughes, T. J. R. and Taylor, R.L., "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," *Earthquake Engng Struct. Dynamics,* Vol. 5, No. 3, 1977, pp. 283-292.
doi: 10.1002/eqe.4290050306

[77] Ypma, T. J., "Historical Development of the Newton-Raphson Method," *SIAM Review,* Vol. 37, No. 4, 1995, pp. 531-551.
doi: 10.1137/1037125

[78] Nocedal, J. and Wright, S., *Numerical Optimization,* Springer Science & Business Media, 2000.

[79] Reddy, J. N., *Energy Principles and Variational Methods in Applied Mechanics*, John Wiley, 2002.

[80] Bathe, K. J., *Finite Element Procedures,* Prentice Hall, Pearson Education, Inc., 2006.

[81] Bazely, G. P., Cheung, Y. K., Irons, B. M. and Zeinkiewicz, O. C., "Triangular Elements in Plate Bending – Conforming and Nonconforming Solutions," *First Conference on Matrix Methods in Structural Mechanics,* Wright-Patterson ATBFB, Ohio, 1965.

[82] Kirchhoff, G., *Vorlesungen uber mathematiche Physik,* Vol. 1, B. G. Teubner, Leipzig, 1876.

[83] Newmark, N. M., "A Method of Computation for Structural Dynamics," *Journal of the Engineering Mechanics Division,* Vol. 85, No. 3, 1959, pp. 67-94.

[84] Rayleigh, L., *Theory of Sound,* (two volumes), re-issue, second edition, Dover Publications, New York, 1945.

[85] Reimer, L., Boucke, A., Ballmann, J., and Behr, M. "Computational Analysis of High-Reynolds Number Aero-Structural Dynamics HIRENASD," *International Forum of Aeroelasticity and Structural Dynamics* CP2009-130, 2009.

[86] Heeg, J., Chwalowski, P., Florance, J., Wieseman, C., Schuster, D. and Perry III, B., "Overview of the Aeroelastic Prediction Workshop," *51st Aerospace Sciences Meeting,* Grapevine, TX, CP2013-783, 2013.

[87] Resor, B. R. and LeBlanc, B., "An Aeroelastic Reference Model for the SWiFT Turbines," Sandia National Laboratories, Rept. SAND2014-19136, Albuquerque, NM, Oct. 2014.
doi: 10.2172/1503792

[88] Bhuiyan, F. H., Mavriplis, D. and Fertig, R. S., "Predicting Composite Fatigue Life of Wind Turbine Blades Using Constituent-Level Physics and Realistic Aerodynamic Load," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference,* CP988, 2016.
doi: 10.2514/6.2016-0988

[89] Zhurkov, S. N., "Kinetic Concept of the Strength of Solids," *International Journal of Fracture* Vol. 26, No. 4, 1984, pp. 295-307.
doi: 10.1007/BF00962961

[90] Regel, V. R., "Kinetic Theory of Strength as a Scientific Basis for Predicting the Lifetime of Polymers Under Load," *Mechanics of Composite Materials,* Vol. 7, No. 1, 1971, pp. 82-93.
doi: 10.1007/BF00856619

[91] Tomashevskii, E. E., Zakrevskii V. A., Novak, I. I., Korsukov, V. E., Regel, V. R., Pozdnyakov, O. F., Slutsker, A. I. and Kuksenko, V. S., "Kinetic Micromechanics of Polymer Fracture," *International Journal of Fracture* Vol. 11, No. 5, 1975, pp. 803-815.
doi: 10.1007/BF00012898

[92] Garnich, M. R. and Hansen, A. C., "A Multicontinuum Approach to Structural Analysis of Linear Viscoelastic Composite Materials," *Jounal of Applied Mechanics-Transactions of the ASME,* Vol. 64, No. 4, 1997, pp. 795-803.
doi: 10.1115/1.2788984

[93] Garnich, M. R. and Hansen, A. C., "A Multicontinuum Theory for Thermal-Elastic Finite Element Analysis of Composite Materials," *Journal of Composite Materials,* Vol. 31, No. 1, 1997, pp. 71-86.
doi: 10.1177/002199839703100105

[94] Duysinx, P. and Sigmund, O., "New Developments in Handling Stress Constraints in Optimal Material Distribution." *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* Vol. 1, 1998.

[95] Qiu, G. Y. and Li, X. S., "A Note on the Derivation of Global Stress Constraints," *Structural and Multidisciplinary Optimization,* Vol. 40, 2010, pp. 625-628.
doi: 10.1007/s00158-009-0397-x

[96] Lambe, A. B., Martins, J. R. R. A. and Kennedy, G. J., "An Evaluation of Constraint Aggregation Strategies for Wing Box Mass Minimization," *Structural and Multidisciplinary Optimization,* Vol. 55, No. 1, 2017, pp. 257-277.
doi: 10.1007/s00158-016-1495-1

[97] Vassberg, J. C., DeHaan, M. A., Rivers, S. M. and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," *26th AIAA Applied Aerodynamics Conference,* CP2008-6919, 2008.
doi: 10.2514/6.2008-6919

[98] Kreisselmeier G. and Steinhauser R., "Systematic Control Design by Optimizing a Vector Performance Indicator," *Symposium on Computer-Aided Design of Control Systems,* IFAC, Zurich, Switzerland, 1979 pp. 113–117.
doi: 10.1016/B978-0-08-024488-4.50022-X