Finite Element Modeling and Multivariate Optimization Over Fibre Orientation and Volume Fraction of Fibre Composite Parts Aimed at Minimizing Targeted Displacements

Pascal Gadoury

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE
in Mechanical Engineering

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering
University of Ottawa
Ottawa, Canada

July 2013

Abstract

A software program was written that implements a finite element analysis (FEA) solution as the basis of an optimization function used for guiding the inverse design problem of aligning fibres, minimizing displacements in a fibre-reinforced polymer composite part in response to a given loading condition, for various part geometries.

The software program developed utilizes an implementation of the Delaunay tetrahedralization algorithm provided by the Tetgen tetrahedral mesh generator for segmenting a piecewise linear complex definition of the part geometry which is optimized. The set of tetrahedra generated forms the basis of the finite element solution, which makes use of superlinear RGNTet4 elements. The RGNTet4 element includes 3 displacement and 3 rotational degrees of freedom (DoFs) at each of its 4 nodes. This finite element software program was tested using simple test cases in order to ascertain the accuracy of the solver. The implementations and algorithms used for storing and operating on both sparse and dense vectors and matrices were sourced from the Eigen linear algebra and numerical solver library. The finite element analysis part of the software was proven to converge to expected results for simple tension and cantilevered beam bending test cases.

Multivariate optimization of fibre orientations and volume fractions was carried out for a simple test case using each of the optimization algorithms provided by the NLOpt nonlinear optimization library, using part displacement minimization as an optimization metric. Both derivative free and gradient-based algorithms using the adjoint method were tested. Low-Storage Broyden-Fletcher-Goldfarb-Shannon was found to be most effective algorithm for this application.

Then, more complex parts were optimized aiming demonstrating the use of the software program developed. Reductions of 48%, 66%, 58% and 32% were achieved in displacements at the loaded end of the parts examined, from a first-guess design, for the four proof of concept cases examined.

Acknowledgements

Foremost, I wish to express my sincere gratitude to my advisor Professor François Robitaille for his guidance and help. His patience, knowledge of composites and motivation have been invaluable throughout my thesis research and redaction.

Further, I appreciate the feedback from Professors Natalie Baddour and Michel Labrosse on the topics of design optimization, finite element analysis and mathematical notation.

I would like to thank my friends and classmates Jean-Michel Roy and Alexandre Bergeron with whom I shared my graduate courses. I will remember fondly the time we spent studying and musing together.

Finally, my sincere thanks go to my family: my parents Richard and Marianne and my sister Dominique, for their ongoing encouragement and support. Without their aid, this thesis would not have been possible.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol	**Description**
$\mathbf{A}$	(3 x 3) orthogonal transformation matrix
$\mathbf{A}$	(m x n) matrix being factorized
$\mathbf{A}_R$	Rotation matrix corresponding to axis $\vec{\boldsymbol{\omega}}$ and angle θ
$\mathbf{A}_{\boldsymbol{\varepsilon}}$	(6 x 6) strain transformation matrix
$\mathbf{A}_{\boldsymbol{\sigma}}$	(6 x 6) stress transformation matrix
A_f	Area corresponding to the product of the total area times the fibre volume fraction
A_{yz}	Cross-sectional area of the specimen normal to the applied load
(a_i, b_i, c_i)	Factors of the inverse of the Jacobian matrix, $\mathbf{Q}$
$\mathbf{B}$	Matrix relating the element displacement vector $\vec{\boldsymbol{u}}_e$ and the strain at a given point within the element
$\mathbf{b}_{i,j}$	(6 x 6) matrix block used in the calculation of $\mathbf{B}$ for RGNTet4 element
$\mathbf{D}$	Linear strain operator
d_y	Displacement in y
$\mathbf{E}$	Elasticity matrix
$\mathbf{E}'$	Elasticity matrix (on-axis)

$\mathbf{E}_{n\,i}$	Elasticity matrix at element n and node i
E	Young’s modulus
E_f, E_m	Young’s modulus of fibre and matrix constituents, respectively
E_x	On-axis longitudinal elastic modulus of the material
E_y, E_z	On-axis transverse elastic modulus of the material
F	Linear function in tetrahedral space
$\boldsymbol{F}_j^{\theta}(\vec{\boldsymbol{x}}_{\boldsymbol{i}})$	(3 x 3) anti-symmetric matrix which describes the deformation of the elastic continuum of the element as a result of nodal rotations
$F_{f,} F_m$	on-axis forces exerted by the fibres and matrix, respectively
$\vec{\mathbf{f}}_a$	Assembled force vector
G_{xy}, G_{yz}, G_{zx}	Shear moduli
$\mathbf{I}$	(3 x 3) identity matrix
I_y	Second moment of inertia about the y axis of the specimen’s cross-section normal to the applied load
$\vec{\mathbf{i}}$	Unit vector aligned along the positive x-axis
$\mathbf{J}$	Jacobian matrix of a system of equations
J	Jacobian determinant
$\mathbf{K}_a$	Assembled stiffness matrix
$\mathbf{K}_e$	Element stiffness matrix
$\mathbf{L}$	(m x m) lower triangular matrix
$\mathbf{N}$	Shape matrix
$\vec{\mathbf{p}}_e$	Element parameter vector
p_i, p_m	Material parameter
$\mathbf{Q}$	Inverse of the Jacobian matrix

R^2	Coefficient of determination
r	In subscript position, representation of rotation
$\mathbf{S}$	Orthotropic material compliance matrix
$\vec{\mathbf{s}}$	Surface tractions acting over element surface S_e
t	In subscript position, representation of translation
$\mathbf{U}$	(m x n) upper triangular matrix
$\vec{\mathbf{u}}$	Column vector describing the displacements in x, y and z at a point
$\vec{\mathbf{u}}_a$	Assembled displacement vector
$\vec{\mathbf{u}}_{\mathbf{e}}$	Element displacement vector
$\overrightarrow{\mathbf{u}_{\mathbf{e}\,i}}$	Displacement vector at node i for a reduced generalized node tetrahedron
$\vec{\mathbf{u}}_r$	Displacement resultant from the rotation of the point at $\vec{x}$ about node i
$\vec{\mathbf{u}}_t$	Displacement resulting from the translation of point at $\vec{x}$
u, v, w	Translational displacements in x, y and z
v_f	Fibre volume fraction
W_e	Work exerted on the system by the external forces
$\vec{\mathbf{x}}$	3D position vector in the Cartesian coordinate system
$\vec{\mathbf{x}}_i$	Cartesian coordinates of node i
$\tilde{x}_i$, $\tilde{y}_i$, $\tilde{z}_i$	Distance $(x - x_i)$, $(y - y_i)$, $(z - z_i)$ in Cartesian coordinates from arbitrary point $\vec{\mathbf{x}}$ to node i
ε	(3 X 3) Strain tensor before a change of coordinates
ε'	(3 X 3) Strain tensor after a change of coordinates
$\vec{\boldsymbol{\varepsilon}}$	6 parameter strain vectors before a change of coordinates
$\vec{\boldsymbol{\varepsilon}}\,'$	6 parameter strain vectors after a change of coordinates
ε_{ij}	Strain tensor component

ε_x	On-axis longitudinal strain
ε_{xx}, ε_{yy} and ε_{zz}	Normal engineering strains in the x, y and z directions
ζ_i	Tetrahedral coordinate corresponding to node i
$\vec{\zeta}$	Tetrahedral coordinates in vector form
$\vec{\boldsymbol{\theta}}_i$	3 row column vector $[\theta_{ix}\ \theta_{iy}\ \theta_{iz}]^{T}$ containing the local rotations around axes x, y and z at node i
γ_{ij}	Engineering shear strains which correspond to 2 ε_{ij}
$\overrightarrow{\lambda_p}$	Unit vector aligned along the desired fibre orientation
ν	Poisson ratio
ν_{xu}	Major Poisson's ratio, with $u = y$ or z
ν_{ux}	Minor Poisson's ratio, with $u = y$ or z
$\prod_{i=k}^{n}$	Product operator
Π	Total potential energy of a system or ratio of a circle's circumference to its diameter
$\sum_{i=k}^{n}$	Sum operator
$\boldsymbol{\sigma}$	(3 x 3) Stress tensor before a change of coordinates
$\boldsymbol{\sigma}'$	(3 x 3) Stress tensor after a change of coordinates
$\vec{\sigma}$	6 parameter stress column vector
$\vec{\sigma}\,'$	6 parameter strain vectors after a change of coordinates
σ_c, σ_f, σ_m	Average normal stress over entire composite cross-section, fibre cross-section and matrix cross-section, respectively
τ_{ij}	Shear stress component

Ω_e	Element volume
$\vec{\omega}$	Axis about which an orthotropic material is rotated
’	Prime operator - position after a transformation
Tilde notation (ex. $\tilde{c}_i$)	Represents $(c - c_i)$ with c being x, y or z
|	Pipe operator - concatenation of two vectors
!	Factorial operator

Glossary

Abbreviation	**Definition**
AMD	Advanced Micro Devices
APPML	AMD’s Accelerated Parallel Processing Math Libraries
BLAS	Basic Linear Algebra Subprograms
BOBYQA	Bound Optimization BY Quadratic Approximation
COBYLA	Constrained Optimization BY Linear Approximations
CPU	Central Processing Unit
DCFP	Directed Carbon Fibre Preforms
DoF	Degree of Freedom
FEA	Finite Element Analysis
GFRP	Glass Fibre-Reinforced Plastics
GPU	Graphical Processing Unit
LBFGS	Low-Storage Broyden-Fletcher-Goldfarb-Shannon
LU Factorization	Decomposition of a matrix as a product of a Lower triangular matrix and an Upper triangular matrix
MMA	Method of Moving Asymptotes
NLOpt	Nonlinear Optimization Library
OpenCL	Open Computing Language

PCG	Preconditioned Conjugate Gradient
PDE	Partial Differential Equation
PMC	Polymer-Matrix Composites
PRAXIS	Principal Axis
RGN	Reduced Generalized Node
RGNTet4	Reduced Generalized Node 4 Node Tetrahedron
SBPLX	Subplex
SLMVM	Shifted Limited-Memory Variable-Metric
SLSQP	Sequential Least Squares Quadratic Programming
TET4	4-Node Linear Tetrahedron
TNEWTON	Truncated Newton Algorithm
VAR1	Rank-1 Shifted Limited-Memory Variable-Metric
VAR2	Rank-2 Shifted Limited-Memory Variable-Metric
ViennaCL	Free open-source linear algebra library for computations on many-core architectures (GPUs, MIC)

Chapter 1

Background and Motivation

1.1. Contents

This chapter describes composite materials and discusses their use; it also states the rationale behind the approach taken for designing a software program that improves the design process for fibre-reinforced composite parts. The tools and libraries used as a result of the specific challenges brought by composite materials and finite element analysis (FEA) are also introduced.

Specifically, Section 1.2 describes the initial motivation and prior work done in this field of study. Section 1.3 introduces fibre-reinforced composite materials highlighting their definition, properties and prevalence in natural and man-made structures. Section 1.4 contains an introduction to the finite element method and its applications in the design process, as well as an overview of its integration into automated optimization schemes which is a central topic of this thesis. Section 1.5 elaborates on the contents and aims of this thesis, and highlights its original contributions.

1.2. Prior Work

This thesis focuses on the use of finite element analysis for predicting the material properties and structural behaviour of composite parts, and on the use of such models as a basis for the automated optimization of such composite parts. Early work was undertaken as a study on the characterization of 3D weaves, then moved on to the creation of models predicting their properties at the mesoscopic scale, and finally settled on the use of optimization algorithms for optimizing the structural stiffness of parts of arbitrary geometry subjected to known loading conditions, through the variation of the reinforcing fibre orientations throughout the part.

While a considerable body of literature is available pertaining to the individual aspects covered in this thesis, published work describing the use of volume elements for describing the behaviour of general 3D composite parts could not be found. Likewise, the use of optimization algorithms for generating a field of orientations and volume fractions best suited to a given part geometry, design specification and load case appears to be novel.

Notably, papers are outlined which describe the characterization of the elastic behaviour of composite materials, as well as the use of finite element analysis for predicting the elastic behaviour of composite parts constituted of arbitrarily shaped shells and plates. For instance, Lomov et al. [**1**] carefully lay out a road map detailing viable steps toward the modeling of textile composites from the micro scale (scale of individual fibres) to the macro scale (scale of parts) through successive scale simulation and homogenization techniques, through which bulk properties are calculated for a representative volume of structures at a given scale given a set of descriptive parameters, for use in larger scale domains. This avenue of research was further explored by De Carvalho, Pinho and Robinson [**2**] who describe a reduced unit cell model which provides an effective means of reducing the size of the domain required for the accurate finite element analysis simulation of meso scale (scale of textile unit cell) 2D textile composites. Of course, these methods require the precise definition of the textile architecture prior to the meshing and meso-scale modeling of the unit cell geometry. Work was done by Lomov et al. [**3**,**4**,**5**] and Robitaille et al. [**6**,**7**] to characterize textile geometries for 2D weaves, 2D woven and braided textiles and 2d warp-knitted textiles given an interlacing pattern and geometric parameters.

As discussed later, the optimization problem examined in this thesis is aimed at parts of arbitrary thickness. Further, the optimized parameters consisting of fibre orientations and volume fractions are not constrained by continuity of adherence to structures which can be manufactured with traditional textile methods. The effects of potentially discontinuous fibre distributions which could result from such optimizations are studied by Qian et al. [**8**] for 2D composite structures.

Alternatively, 3D textiles offer appealing benefits towards the structural performance of fibre-reinforced composite materials, imparting near net-thickness reinforcement and improved delamination resistance to components. The characterization of 3D textile composites is the subject of ongoing research, notably by Tan, Tong and Steven [**9**], Thomson et al. [**10**] and Tolosana et al. [**11**], all of whom utilize unit cell meshing, as in the works on modelling 2D textiles described above. Work is ongoing towards describing effective means of defining unit cell geometries for 3D textiles, notably by Tolosana et al. [**11**], Lomov et al. [**12**] and Fang and Liang [**13**]. Li, Wen and Aliabadi provide an alternative to the traditional finite element meshing paradigm, by applying the Galerkin method to the constitutive equations describing a specified geometry.

The methods used for describing 3D textile geometries enumerated above are limited in scope to the class of textiles that they describe. A more general approach is taken in this thesis, using a custom software program developed purposely which describes the fibre orientations by using a vector field suspended in a 3D voxel mesh. As a departure from the traditional approach used for modeling composite material parts, fibre structure was defined and simulated over complete part geometries meshed at the macro scale rather than describing fibre orientations in a repeating unit cell at the meso scale and using the properties calculated to model macro scale parts. The finite element model used for describing the resultant part behaviour was then integrated with a series of optimization routines aimed at minimizing displacements as a function of specified loadings, and so generating favourable fibre orientations. Works reviewed for supporting this process are cited throughout the thesis in the relevant sections.

1.3. Background on Composite Materials

1.3.1 Definition

A composite material is a macroscopic combination of two or more distinct materials, having a recognizable interface between them [**14**]. Because of the breadth of materials that may be regarded as composite materials and scope of applications thereof, in this thesis the moniker composite materials will be restricted to materials that feature a continuous matrix constituent which binds together and gives shape to an array of stronger and stiffer reinforcement fibres. Composite materials benefit from simultaneous advantages offered by their constituent materials while mitigating their undesirable qualities [**14**]. Even within the bounds of this definition, the array of possible composite materials is very large. Consequently, it should be noted that the focus of this thesis is primarily on fibre-reinforced composites.

Composite materials are widely found in natural and man-made structures. Wood, bone and teeth are notable examples of natural composites [**15**] whereas concrete, ceramic metal matrix composites as well as carbon, glass and aramid reinforced polymers are common examples of man-made composites.

1.3.2 Material Orthotropy

Several instances of composite materials feature a fibrous or directional reinforcement, the modulus and strength of which exceeding those of the surrounding continuous matrix. As shown in Figure 1-1, the distribution of fibre orientations is commonly biased along certain directions. As a result, there is a marked variation in properties measured in different directions. This phenomenon, known as anisotropy, is commonly exploited in both natural and man-made composites [**15**].

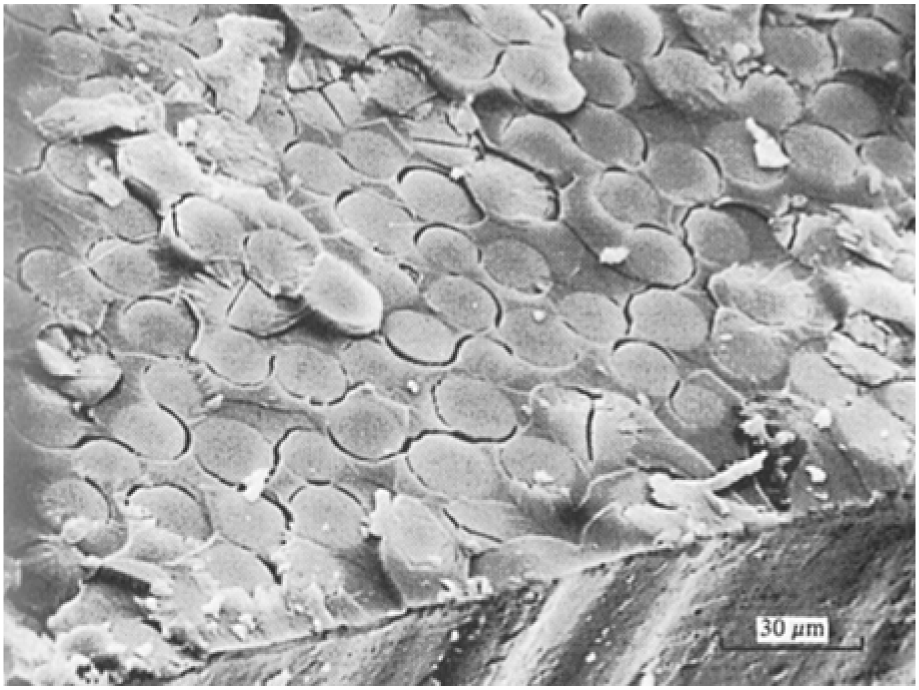


Figure 1-1: SEM micrograph of the fracture surface of a polyester/60% glass lamina [16**]**

Material orthotropy is a special case of anisotropy, which describes materials having three mutually perpendicular principal directions [**17**]. Further information regarding the elastic behaviour of orthotropic materials, including a mathematical model, can be found in Sections 2.7.3 and 2.7.4, and a more specific model pertaining to fibre-reinforced materials is described in Section 2.7.6.

1.3.3 Effect of Fibre Length and Continuity

The length of reinforcement fibres can have an effect on the composite material's strength and elastic modulus. Fibre-reinforced composites featuring reinforcements that are beyond a length threshold where any further increase of fibre length has a negligible impact on the material properties of the composite are known as continuous fibre-reinforced composites. In contrast, composites where fibre length is below the threshold are known as short fibre composites [**14**].

In general, short fibre composites have inferior material properties because the matrix material surrounding the reinforcing fibres must carry a larger proportion of mechanical loads [**14**]. Furthermore, it is more difficult to achieve the high directionality and volume fractions necessary for high performance applications using short fibre composites [**18**]. However, short fibre composite materials offer advantages in terms of manufacturing time and labour. Also, recent advances in processes such as directed carbon fibre preforms (DCFP) enable the manufacturing of higher volume fraction parts with better control over fibre orientation, achieving up to 83% and 31% of the strength and stiffness of a comparable unidirectional continuous fibre composite part [**18**].

1.3.4 Natural Composites

Examples of composites materials occurring naturally in the environment include wood and bone. The former features fibrous chains of cellulose suspended in a lignin matrix, while the latter is made of hydroxyapatite crystals suspended in a collagen matrix [**15**]. Remarkably, the structure of these materials at the macroscopic and microscopic scales is not only very well suited to resisting mechanical loadings common to their structural function, but it also serves in accomplishing a range of biological functions.

Wood fibres for instance are usually aligned longitudinally in the direction of the trunk and branches of a tree [**15**]. Because of wood's fibrous nature and since its fibres are predominantly aligned in one direction, wood as a material behaves in an orthotropic manner. Hence in the case of wood, structural properties vary between the longitudinal, radial and tangential directions [**19**]. However, the predominant fibre alignment not only serves in resisting the compression and bending loads associated with the tree's weight and wind forces acting on its limbs, both imparting high longitudinal stresses, but this alignment also helps in distributing water to the tree's limbs [**20**]. Figure 1-2 shows the fracture surface of a 2 x 2 cm piece of pine wood where the fibres are readily apparent. Figure 1-3 shows the microscopic structure of woody tissue, specifically the longitudinally aligned xylem tracheids.

Figure 1-2: Fracture surface of a 2 x 2 cm pine bar showing wood fibres

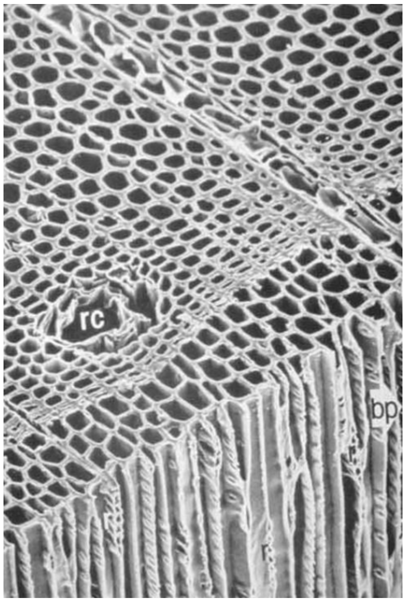


Figure 1-3: Micrograph of section through woody tissue, showing xylem tracheids [21**]**

Bones exhibit a structure that suits the mechanical loads that they typically encounter. Specifically, at the macroscopic level the cortex of a bone, or its outer layer, is dense and thick [**22**]. On the other hand, its core consists of a latticework of bars and plates called trabeculae. This type of bone referred to as cancellous or trabecular bone, supports the bone marrow and gives additional strength to the cortices [**22**]. Indeed, the macroscopic structure of most bones can be thought of as a sandwich structure [**23**]. Such an arrangement offers several advantages; namely it improves specific strength, impact and damage resistance as well as rigidity [**24**]. Figure 1-4 shows a photograph of the section of a human fourth lumbar vertebra, wherein the mesoscopic structure of the cortical and trabecular bone are evident.

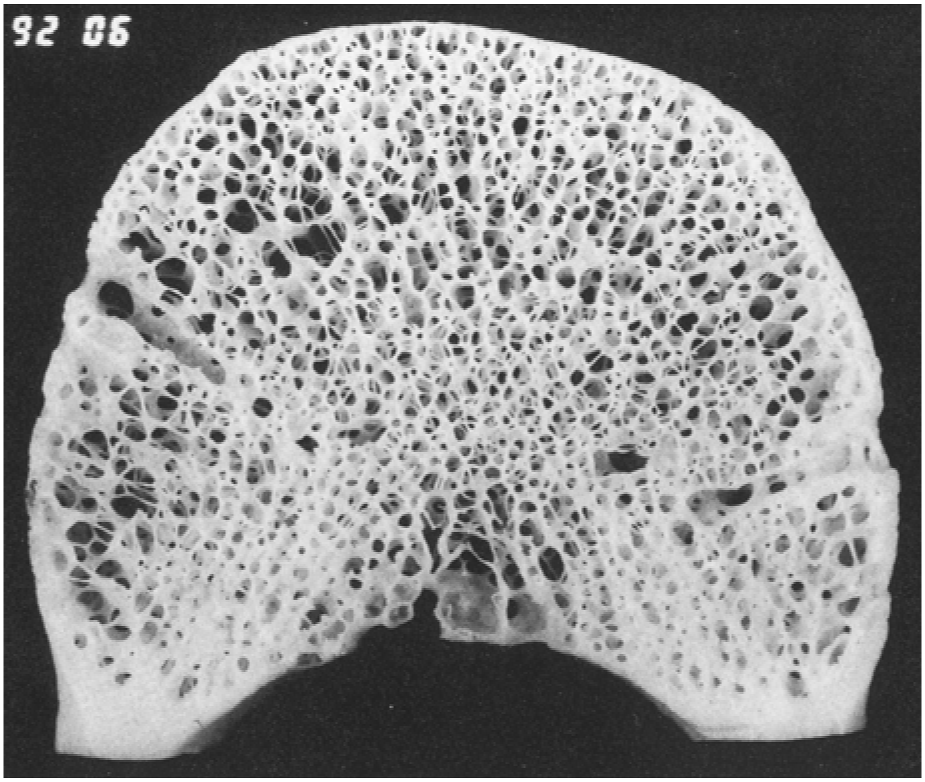


Figure 1-4: Photograph of horizontal section through human fourth lumbar vertebra [25**]**

The microscopic structure of bones further underlines their structural function. At a microscopic scale mature bones feature lamellae which are, in turn, made of a sheet of mineralized matrix containing grossly aligned collagen fibres. The sheets usually span entire lamellae and form branching bundles approximately 2 – 3 µm thick [**22**]. Figure 1-5 shows these structures in a human mandibular bone.

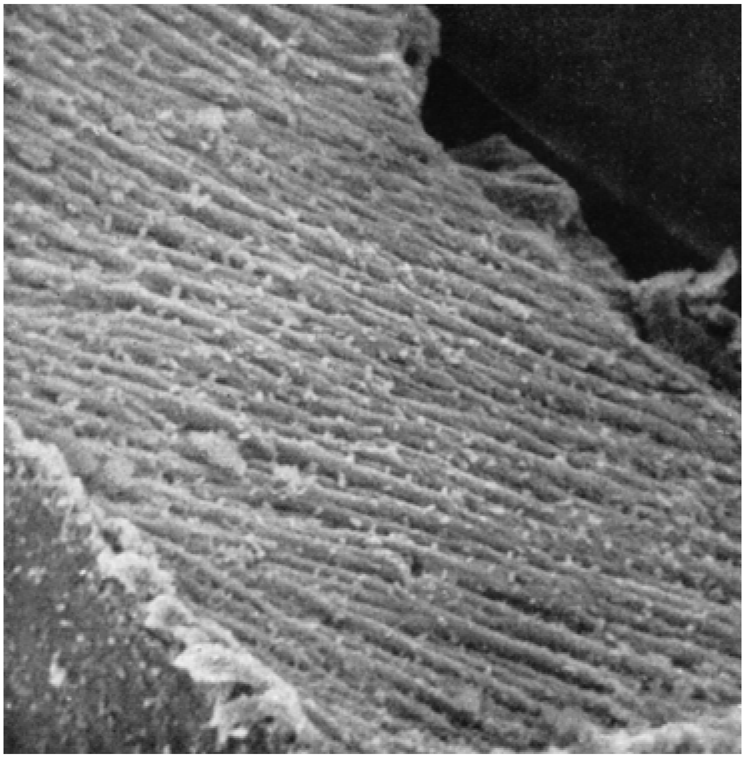


Figure 1-5: Scanning electron micrograph showing collagen fibre bundles smoothly impregnated in hydroxyapatite crystals [26]

In trabecular bone, these lamellar plies form the basis of branching plates and bars approximately 50 to 400 µm thick called trabeculae. The main direction of the collagen fibrils and their associated mineral constituent varies from lamella to lamella [**22**].

The outer layer of a bone, called cortical bone, is also comprised of lamellae. However, in cortical bone they form concentric layers centered about neurovascular channels approximately 50 µm in diameter, called Haversian canals. This arrangement, approximately 30 lamellae thick, is called an osteon. Osteons are usually aligned with each other. The alignment of each osteon depends on the predominant loading mode at their location; osteons are aligned longitudinally in areas predominantly subjected to tension, whereas they are more oblique in areas predominantly subjected to compression [**22**]. Like wood, because bone features fibres primarily aligned along certain directions, it behaves in an orthotropic manner [**15**].

Although the overall morphology of the skeleton is an evolutionary trait which suits the characteristics of its organism, the bones themselves are also in a continuous state of remodelling as a function of growth, repair and response to mechanical loading. The pattern and extent of remodelling is a function of the mechanical loads applied to the bone [**22**]. Summarily, this biological process tailors both the morphology and the microstructure of the skeleton to the loads it encounters, taking into account both the genetics of an organism but also as a direct consequence of past mechanical loading. For example, orthodontic treatment involves the application of steady lateral or medial forces which move teeth gradually though alveolar bone. Likewise, though repeated tension the bones in the racket arm of tennis players grow more robust than those of their opposed appendage [**22**].

1.3.5 Artificial Composites

Composite materials have been the subject of much interest from the aeronautics community from as early as World War II, during which polymer matrix composites, or PMCs, were developed to overcome shortcomings of existing aerospace structural alloys such as those based on aluminum. PMCs offered superior specific strength and stiffness as well as better corrosion resistance. Furthermore, they later proved to have longer fatigue lives, tailorable properties and the ability to manufacture complex shapes [**14**]. From then on, glass fibre-reinforced polymers (GFRPs), were used successfully in filament-wound rocket motors by the end of the war [**14**], and eventually found broader use not only in aerospace applications in the 1950s, but also in commercial applications during the 1960s [**14**]. Because of their remarkable properties, PMCs have become more prevalent not only in aircraft construction but also in naval and land vehicles, both for civil and military applications [**14**].

Recent global trends promoting energy efficiency and greenhouse gas minimization have to bolstered the growth of composite materials in commercial applications. Because of their impressive specific properties, composites constitute an attractive solution towards mass reduction and increased fuel economy for aerospace and automotive vehicles [**27**]. By the same token, a greater focus on recycling has underlined a flaw prevalent in many composite materials. By nature, composites are an intimate mix of disparate constituent phases. As a consequence, they can be difficult to recycle. Some composites also contain phases which have undergone an irreversible chemical reaction like, for example, fibre-reinforced polymer matrix composites based on a thermosetting resin. Although strides are being made in the field of recycling and reuse of composites [**27**], their increased adoption in high-turnover applications such as consumer goods must be met with a commensurate regard for their full life-cycle.

1.4. Computer Aided Part Design Through Optimization

1.4.1 Fibre-Reinforced Composite Part Design Challenges

Like their natural counterparts, most structural composites are fibrous and therefore orthotropic. As a consequence, the design of a composite part must take into account not only the shape of a part, but also the locally orthotropic nature of the material; the local orientations of fibres dictates the overall strength of the part, its stiffness, and the stresses along different directions under a given load [**15**].

Biomimicry is the study of natural structures, systems and processes aiming at adapting these solutions to engineering problems. One such example is the use of lightweight laminate structures such as those used in trabecular bone towards the construction of aircraft components. However, further inspiration can be drawn from the process itself, through which such natural structures emerge; specifically, adaptation. As evidenced by the remodelling process in bones, biological structures can have the ability to adapt themselves continuously to better suit their purpose [**22**]. As a result, living organisms are well suited to their continued survival within the environment they inhabit. In contrast, synthetic structures are usually very limited in their adaptability. Therefore, the burden of matching the structure to its function falls on its designers, who must foresee a design to suit their purpose.

Evidently, prior experience as well as established models help inform the design process, as do solutions to similar problems in both accepted practice and the natural world. However, each new design is essentially an open loop process. Prototyping an evolving design through several generations can be prohibitively costly and time-consuming. The design of composite parts is further hindered by the fact that most composite materials are orthotropic. Therefore, predicting the behaviour of a part accurately requires more sophisticated models and additional time. Furthermore, as it is difficult to recycle many composite materials, a failed prototype cannot be salvaged easily.

1.4.2 Automated Computer Aided Design

The advent of inexpensive and powerful computers and sophisticated models for predicting material behaviour offers an additional design tool. By generating a computer model of a structure, one can predict its behaviour and improve upon it before the prototyping stage. Such simulations can shorten the design cycle and yield better structures. Computer aided design can be taken one step further; by defining a metric as well as a set of design parameters, an optimization algorithm informed by the computer model improve the design can rapidly and efficiently, without human intervention. Figure 1-6 shows a flowchart describing the method envisaged in this work. In this figure, light blue squares signify design parameters, the green square represents a basis for the optimization metric. The FEA solver, nonlinear optimization algorithm and optimization criteria are in blue, burnt orange and amber respectively.

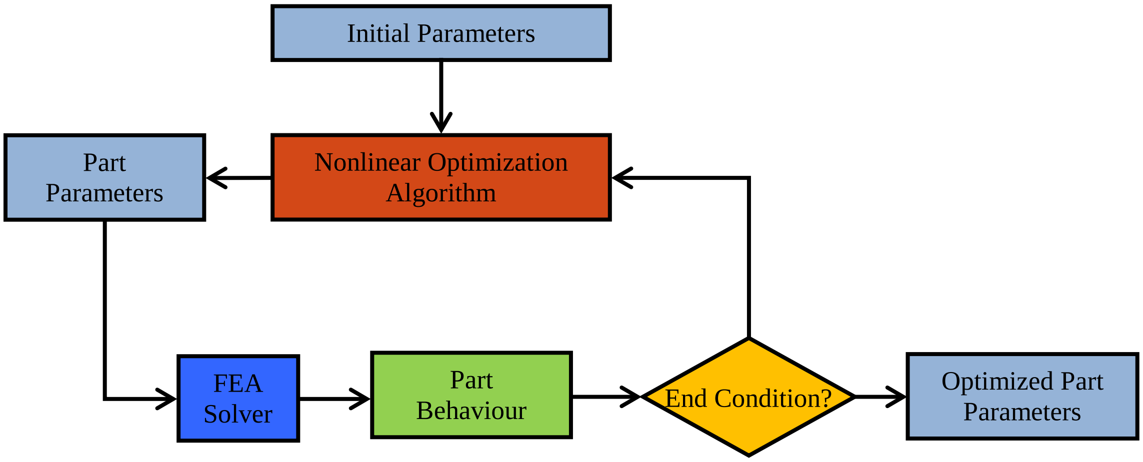


Figure 1-6: Automated design method flowchart

A corollary of this method is that the validity of any design obtained as a solution to the optimization problem is directly dependent upon the careful selection of an optimization algorithm, a valid description of the optimization problem as well as on an accurate model of the part being designed. Beyond providing an accurate model of the physical reality and specifying an efficient optimization algorithm, the task of the designer also consists in mathematically formulating the goals and constraints of the design problem.

The objective of this thesis is to explore further the concept of automated optimization of composite parts through the use of nonlinear multivariate optimization algorithms and finite element analysis. It is understood that the elastic response of fibre-reinforced composite materials is highly dependent on the concentration and orientation of its constituent fibres. In the scope of this thesis, the design goals reflected by the optimization functions which were implemented are modeled through a weighted sum of displacements at selected points on the parts being optimized. The design parameters consist of a field of values representing fibre orientations and volume fractions, which vary throughout the part. Simply put, the current implementation of the software program developed maximises the structural stiffness of a given part as a function of fibre orientations and volume fractions throughout that part.

1.5. Contribution and Overview

As mentioned in Section 1.2, an analogue to the software program developed in the scope of this thesis could not be found in the literature. Minimizing the displacements resultant from a given applied loading on a fibre-reinforced composite part by modifying the fibre orientations and volume fraction defined as a piecewise continuous field throughout the part by using a finite element solver and a library of optimization algorithms appears to be a novel concept.

However, the implementation of this method can be broken up into components which are based off existing practices in the fields of finite element method, infinitesimal strain theory and mathematical optimization. Namely, finite element method is used to model the mechanical behaviour of a fibre-reinforced composite domain of arbitrary geometry through the use of infinitesimal strain theory. The derivation of the reduced generalized node 4 node tetrahedron element for modeling elastic continua is given in Chapter 2; as is the derivation of Hooke's law for arbitrarily oriented fibre-reinforced materials through the off-axis transformation of the orthotropic elasticity matrix. Variations of this equation pertaining specifically to orthotropic materials and to off-axis fibre-reinforced composite materials are also given in this chapter. Chapter 2 then concludes with the calculation of the derivative of a function of the displacement vector with respect to parameters describing the fibre orientations and volume fractions throughout the part with the aim of enabling the use of gradient-based algorithms.

Naturally, in order to verify the calculations and algorithms of the FEA solution for fibre-reinforced composite parts implemented in the software program, convergence testing of the solver was conducted. To that effect, results obtained through the use of the FEA solver were compared to those of analytical equations for simple tension and for a beam loaded in bending over meshes of increasing resolution. These results are analyzed in more detail in Chapter 3.

Likewise, an analysis of the multivariate nonlinear optimization algorithms provided by the NLOpt library served to ascertain the function and practicality of optimizing 3D general fibre-reinforced composite parts, as done in this thesis. Chapter 4 provides additional information regarding the formulation of the optimization function and optimization constraints, followed by detailed descriptions and benchmarking of the specific optimization algorithms for a simple beam bending test case. To further prove the practicality of the method employed, three test cases featuring more complex geometries were run. These tests, which showcase practical usage of the software program, can be found in Chapter 5.

Unfortunately, the program which was developed is not without its limitations and simplifying assumptions. To that effect, a discussion of shortcomings and possible improvements of the FEA model and optimization framework is presented alongside the current capabilities of the software program are given in Chapter 6.

Chapter 2

Finite Element Method Formulation

2.1. Contents

This chapter describes the theory and approach followed in developing the finite element analysis used for predicting the properties of fibre-reinforced composite materials. Figure 2-1 illustrates the calculation steps involved in the FEA solution. The algorithms and equations pertaining to each step in the calculation of the nodal displacements as shown in this figure are therefore described herein.

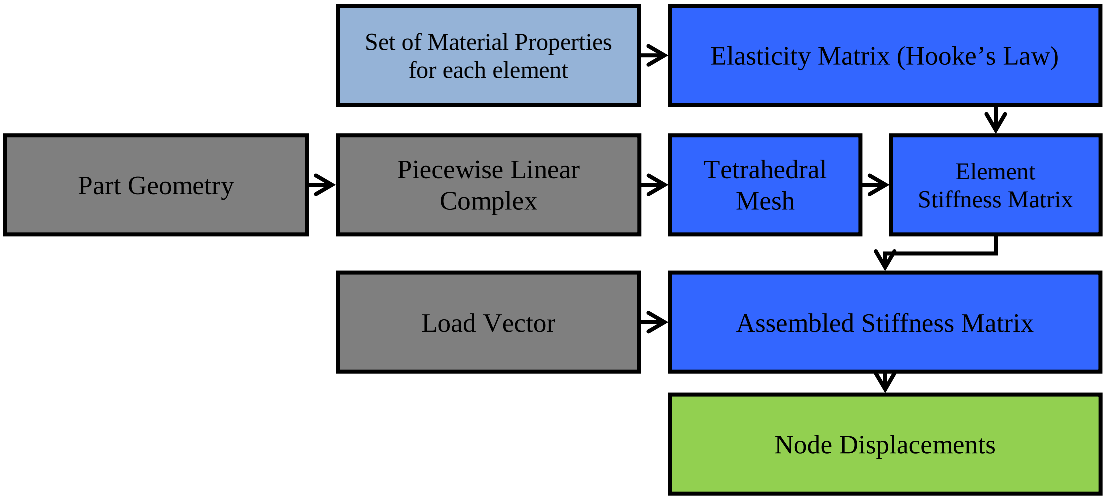


Figure 2-1: Flowchart describing FEA solution steps

Specifically, Section 2.3 introduces the finite element theory. Section 2.4 describes the geometry and basic properties of linear tetrahedra [**28**,**29**]. Section 2.5 introduces the reduced generalized node 4 node tetrahedron element (RGNTet4) [**30**,**31**,**32**] which was selected ultimately for the finite element model.

Section 2.7 details the derivation of Hooke's law in three dimensions, including the specific variations of the law pertaining to orthotropic materials and, further, to off-axis fibre-reinforced composite materials. Section 2.8 presents the derivation of equations towards obtaining the element stiffness matrix. Section 2.9 describes the discretization process central to FEM, its implementation through the use of the Tetgen tetrahedral mesh generator [**33**] and the calculation of an assembly matrix through the use of the finite element method.

Section 2.10 describes the methodology used for solving for displacements given, the assembled stiffness matrix and a vector which describes the loading condition imposed on the part, through the use of the Eigen library [**34**] and its sparse solver [**35**].

Section 2.11 constitutes a summary of the calculations steps performed by the software program which was developed in order to obtain the displacement vector from a given part geometry and loading condition. Section 2.12 addresses specific considerations regarding the optimization of multivariate functions involving finite element analysis solutions. Specifically, this section describes the calculation of the derivative of a function of the displacement vector with respect to parameters describing the fibre orientations and volume fractions throughout the part.

2.2. Introduction

The finite element method (FEM) is a mathematical tool aimed at calculating a numerical solution to integral or differential equations. Applications in engineering include heat transfer, fluid flow and, central to the context of this work, stress analysis and the structural behaviour of complex parts [**36**].

Engineering problems where behaviour that can be described by an integral or differential equation is queried over domains that feature complicated geometry are often impossible to solve using classical methods such as separation of variables. The basic premise of FEM is to divide such complicated problem spaces into finite elements, the behaviour of which is more easily modeled [**36**]. Of course, the accuracy depends on the number and type of elements used.

Through the definition of optimization functions based on the FEA simulation of elastic orthotropic continua over part domains, the code which was developed ultimately in this work enables the optimization of composite material parts. This entails the adequate simulation of on-axis and off-axis stiffness of orthotropic materials, as well as the ability to model variations of the fibre orientation and volume fraction throughout the domain. In order to converge more quickly to an optimized solution, the derivatives of the response to load with respect to each optimization parameter were implemented.

The FEM formulation which was selected for calculating the displacements, stresses and strains within a composite part in response to applied loads relies on the use of Reduced Generalized Node Tetrahedral Element with 4 nodes. This element is a subclass of generalized node tetrahedra and will henceforth be abbreviated as RGNTet4. Its formulation and the details of the FEA implementation used in this project are outlined in the following section.

2.3. Tetrahedron Geometry

Figure 2-2 shows a tetrahedron. This geometric solid features 4 planar triangular faces, 6 straight edges and 4 vertices. The vertices are numbered 1 to 4 conforming to referenced articles [**29**,**30**,**31**,**32**]. However, as per common practice in C++ which was used for programming the optimization routines and FEA model described herein, the array indices within the code start at 0.

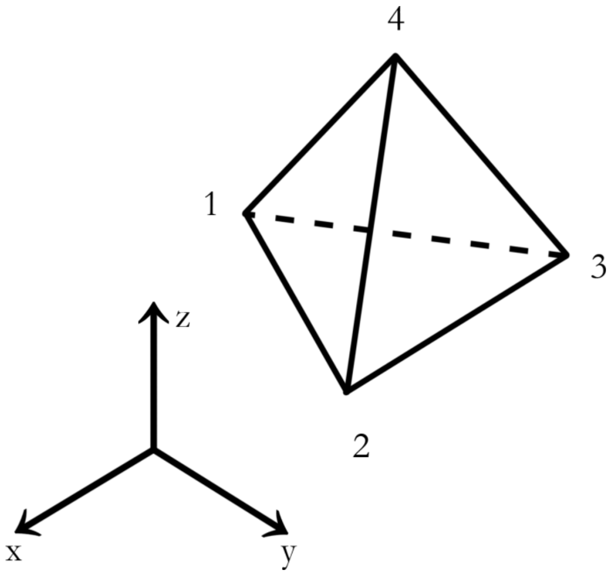


Figure 2-2: Tetrahedron

The tetrahedron is completely defined by giving the position of each vertex in Cartesian coordinates x, y and z relative to the global origin.

$$\vec{\mathbf{x}}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \text{ for } i = 1, 2, 3, 4 \tag{2.1}$$

The tetrahedron is attractive as a base for volume elements because any faceted volume, or polyhedron, can readily be subdivided into a collection of tetrahedra. This property enables the generation of 3D meshes from any geometry using solely tetrahedra. In comparison, polyhedral geometries cannot generally be subdivided into a finite number of well-formed hexahedra. For this reason, the FEM formulation which was developed uses tetrahedra exclusively as a base for volume elements.

2.4. Linear Tetrahedral Element

2.4.1 Linear Four-Node Tetrahedron and Tetrahedral Coordinates

In order to calculate the stress-strain behaviour over the element and eventually derive displacements at the nodes resulting from an applied force, a formalism that enables the description of such quantities and properties over the volume of the element is introduced.

In FEM parlance, a shape function relates the value of an element property being calculated over the volume of an element, such as a displacement or stiffness, to values of that property and/or other properties at the element's nodes. If the given function representing the property is linear the shape function may be represented by a shape function matrix, denoted as $\mathbf{N}$. For the linear tetrahedron, a shape function is defined by a linear interpolation between values at each of its four vertex nodes. This simple element is the basis for the RGNTet4 element which was selected ultimately in the FEM formulation used.

A common stepping stone towards defining a shape function is the use of tetrahedral coordinates. Though tetrahedral coordinates may be defined in several ways, the definition of tetrahedral coordinates in this work is consistent with that of so-called natural tetrahedral coordinates, and are defined as follows: the value of ζ_i is exactly one at vertex i and exactly zero at the opposite vertices and over the opposite face. As shown in Figure 2-2 for node $i = 1$, ζ_i varies linearly with distance from node 1 to the opposite face, along the normal to that face.

Table 2-1 describes the value of the tetrahedral coordinates at each vertex, or when $\vec{\mathbf{x}}$ is equal to $\vec{\mathbf{x}}_i$, with $i = 1, 2, 3$, and 4.

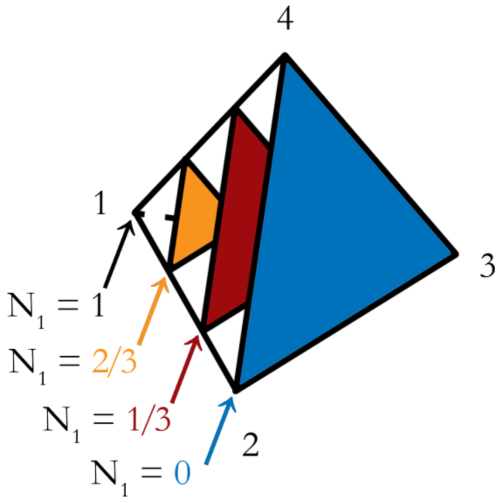


Figure 2-3: Tetrahedral coordinates example; adapted from [28]

Table 2-1: Tetrahedral coordinates at each node

	Node 1	Node 2	Node 3	Node 4
ζ_1	1	0	0	0
ζ_2	0	1	0	0
ζ_3	0	0	1	0
ζ_4	0	0	0	1

The sum of these four coordinates is equal to 1 at any point within the tetrahedron. This relation between ζ_1, ζ_2, ζ_3 and ζ_4 reduces the number of independent variables to three, fully defining the position of a point in 3D.

$$\zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 = 1 \quad \textbf{(2.2)}$$

2.4.2 Linear Interpolation using Tetrahedral Coordinates

Any linear function in x, y and z that takes value $F_i = F(x_i, y_i, z_i)$ at node i may be transformed to tetrahedral coordinates as follows:

$$F(\zeta_1, \zeta_2, \zeta_3, \zeta_4) = F_1\,\zeta_1 + F_2\,\zeta_2 + F_3\,\zeta_3 + F_4\,\zeta_4 \quad \textbf{(2.3)}$$

or, combining **(2.2)** and **(2.3)** in matrix form:

$$\begin{bmatrix} 1 \\ F \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ F_1 & F_2 & F_3 & F_4 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \quad \textbf{(2.4)}$$

2.4.3 Coordinate Transformations

Tetrahedral coordinates consist of a coordinate system local to the element that are well suited for describing linear functions within a tetrahedron. In order to facilitate a change of variables between tetrahedral coordinates and the global Cartesian coordinate system and vice-versa, the Jacobian matrix for tetrahedral coordinates, denoted **J**, is introduced.

The Jacobian matrix of a system of equations $\vec{\mathbf{f}}(\vec{\mathbf{x}}) = \left[f_1(\vec{\mathbf{x}})\ f_2(\vec{\mathbf{x}})\ \ \dots\ \ f_n(\vec{\mathbf{x}}) \right]^{\mathrm{T}}$, so named after German mathematician Carl Gustav Jacob Jacobi [**37**], is defined as the first order partial derivative of the set of equations where the first column corresponds to the derivative of that set with respect to x_1, the second column corresponds to the second derivative of that set with respect to x_2, and so on [**38**]:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \qquad \textbf{(2.5)}$$

The Jacobian determinant is defined as the determinant of the above matrix:

$$J = \det(\mathbf{J}) \qquad \textbf{(2.6)}$$

As the Cartesian coordinates x, y and z vary linearly throughout the tetrahedron, it is possible to obtain coordinates at any point from interpolation of the tetrahedral coordinates. Recalling equation **(2.4)**, the equation for the change of coordinates between the two coordinate systems is:

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \qquad \textbf{(2.7)}$$

Since equation **(2.7)** is a linear function of N_1, N_2, N_3 and N_4, the Jacobian matrix of the nodal coordinates described in equation **(2.1)** is equal to:

$$\mathbf{J} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \qquad \textbf{(2.8)}$$

This matrix has many interesting properties. For instance, its determinant J is exactly equal to 6Ω, with Ω being the volume of the element [**29**]. Furthermore, inverting equation **(2.7)** yields the relation for the tetrahedral coordinates as a function of $\vec{x}$:

$$\vec{\zeta} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} = \frac{1}{6\Omega} \begin{bmatrix} 6\,V_{01} & a_1 & b_1 & c_1 \\ 6\,V_{02} & a_2 & b_2 & c_2 \\ 6\,V_{03} & a_3 & b_3 & c_3 \\ 6\,V_{04} & a_4 & b_4 & c_4 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \mathbf{Q} \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} \qquad \textbf{(2.9)}$$

where V_{0i} , a_i, b_i and c_i for i = 1, 2, 3, 4 correspond to the factors of the inverse of the Jacobian matrix **Q**.

The geometric interpretation of V_{01}, V_{02}, V_{03} and V_{04} is that these quantities correspond to the volumes of tetrahedra formed by the origin at $\vec{\mathbf{x}} = [\,0\ 0\ 0\,]^{\mathrm{T}}$ and the faces opposite nodes 1, 2, 3 and 4 respectively. The signs of V_{01}, V_{02}, V_{03} and V_{04} depend on the direction of the normal to the faces of these tetrahedra. This interpretation has no impact on the FEM formulation but is included here for the sake of consistency with [**29**].

2.4.4 Linear Tetrahedron Shape Function

As shown in equation **(2.3)**, the heretofore mentioned tetrahedral coordinates may be used for relating coordinates or functions in x, y and z to their nodal instances $F_{\mathrm{i}} = F(x_i, y_i, z_i)$. Therefore, they could be used verbatim as a basic shape function. This is done, for instance, for calculating the displacements in the standard 4-node linear tetrahedron TET4 [**29**]:

$$\vec{\mathbf{u}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \sum_{i=1}^{4} \zeta_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} \qquad \mathbf{(2.10)}$$

where the vector $\vec{\mathbf{u}}$, whose components are u_i, v_i, and w_i, represents the displacements in x, y and z, respectively. By combining all four displacement vectors $\vec{\mathbf{u}}_{\mathbf{i}} = [\,u_i, v_i, w_i\,]^{\mathrm{T}}$ into a single column vector this relation can be expressed as a simple matrix multiplication:

$$\vec{\mathbf{u}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \left[\; \zeta_1 \mathbf{I} \mid \zeta_2 \mathbf{I} \mid \zeta_3 \mathbf{I} \mid \zeta_4 \mathbf{I} \;\right] \begin{bmatrix} \vec{\mathbf{u}}_1 \\ \text{—} \\ \vec{\mathbf{u}}_2 \\ \text{—} \\ \vec{\mathbf{u}}_3 \\ \text{—} \\ \vec{\mathbf{u}}_4 \end{bmatrix} \tag{2.11}$$

where **I** is a (3 x 3) identity matrix. The pipe operator **|** represents the concatenation of the vectors and matrices.

2.4.5 Cartesian and Tetrahedral Coordinate Derivatives

In order to express differential equations using tetrahedral elements, the derivative of a function expressed in tetrahedral coordinates with respect to Cartesian coordinates is of use. To this end, from linear relation **(2.7)** the partial derivative of the Cartesian coordinates with respect to tetrahedral coordinates can be calculated. Writing out the matrix multiplication in **(2.7)** as an explicit sum of products and differentiating it with respect to ζ_i,

$$\frac{\partial \vec{\mathbf{x}}}{\partial \zeta_i} = \sum_{j=1}^{4} \frac{\partial \zeta_j}{\partial \zeta_i} \vec{\mathbf{x}}_j \tag{2.12}$$

it follows that since

$$\frac{\partial \zeta_j}{\partial \zeta_i} \vec{\mathbf{x}}_j = 0 \text{ for } i \neq j \qquad \text{and} \qquad \frac{\partial \zeta_j}{\partial \zeta_i} \vec{\mathbf{x}}_j = 1 \qquad \text{for } i = j,$$

$$\frac{\partial \vec{\mathbf{x}}}{\partial \zeta_i} = \vec{\mathbf{x}}_i \text{ , for } i = 1, 2, 3, 4. \qquad \textbf{(2.13)}$$

Likewise, from **(2.9)** the partial derivative of the tetrahedral coordinates with respect to Cartesian coordinates can be calculated:

$$\frac{\partial \zeta_i}{\partial x} = \frac{a_i}{6\Omega} \qquad \frac{\partial \zeta_i}{\partial y} = \frac{b_i}{6\Omega} \qquad \frac{\partial \zeta_i}{\partial z} = \frac{c_i}{6\Omega} \qquad \textbf{(2.14)}$$

Finally, the partial derivative of a given function in tetrahedral coordinates with respect to Cartesian coordinates can be calculated using the chain rule:

$$\frac{\partial F}{\partial \mathrm{x}} = \sum_{i=1}^{4} \frac{\partial F}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial x} \qquad \frac{\partial F}{\partial \mathrm{y}} = \sum_{i=1}^{4} \frac{\partial F}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial y} \qquad \frac{\partial F}{\partial z} = \sum_{i=1}^{4} \frac{\partial F}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial z} \qquad \textbf{(2.15)}$$

Substituting **(2.14)** into **(2.15)** yields:

$$\frac{\partial F}{\partial \mathrm{x}} = \sum_{i=1}^{4} \frac{a_i}{6\Omega} \frac{\partial F}{\partial \zeta_i} \qquad \frac{\partial F}{\partial \mathrm{y}} = \sum_{i=1}^{4} \frac{b_i}{6\Omega} \frac{\partial F}{\partial \zeta_i} \qquad \frac{\partial F}{\partial z} = \sum_{i=1}^{4} \frac{c_i}{6\Omega} \frac{\partial F}{\partial \zeta_i} \qquad \textbf{(2.16)}$$

2.5. Higher Order Elements

2.5.1 Lagrange Interpolants and High Order Elements

The previous section describes an effective means to interpolate linearly a function over a tetrahedral domain. While the shape function described in equation **(2.11)** and the derivatives described in Section 2.4.5 can be used directly for calculating the linear tetrahedral element TET4 [**29**], this section will demonstrate the merits of using higher order elements. The use of higher order elements will then be explored as an enhancement to pure mesh refinement as a means of increasing the accuracy of a FEM solution. This analysis is then used to justify the final selection of element used in the present solution, considering the specifics of optimization problems examined as well as the evolution of the scope and method of the thesis.

As described in Section 2.4.2, linear interpolation represents the basis of the linear tetrahedron. However, linear interpolation is merely a specific case of Lagrange interpolants. This family of functions is used extensively as a basis for shape functions in FEM [**39**,**40**]. As described in Section 2.4.2, the method of linear interpolation describes a function using the values of that function at the nodes. A more accurate approximation of nonlinear functions can be made using higher order Lagrange polynomials as a basis for shape functions. Naturally, additional midpoint nodes and/or supplementary parameters at each node are required to fully describe higher order elements, resulting in an increase of their computational and analytical complexity.

The basic concept of Lagrange interpolants relies on the fact that a function can be approximated with the following sum of products, given a set of known points with which to interpolate [**39**,**41**,**42**]:

$$f(x) \approx P(x) = \sum_{j=1}^{n} P_j(x)$$

$$P_j(x) = f(x_j) \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} \tag{2.17}$$

Section 0 introduced mesh refinement as a central aspect of FEA. This approach, described in more practical terms in Section 2.9, implies that by refining a mesh and consequently increasing the number of function parameters, a more accurate solution to the differential equation being modeled can be obtained. As a consequence, a comparison can be made between increased accuracy obtained through mesh refinement, and increased accuracy achieved through the use of higher order elements.

Although the use a higher number of low order elements is better suited for describing small geometric features, and an increase in discretization resolution may ensure eventual convergence towards an adequate solution, an increase in the number of parameters or DoFs used for describing each discrete element will allow much faster convergence when the solution describes a smooth function. This is illustrated in Figure 3-3, which contrasts the accuracy obtained from mesh refinement and higher order interpolation polynomials for a 1D function, specifically the cosine function taken between $-\Pi/2$ and $\Pi/2$. From left to right, the columns in the figure represent successive piecewise refinement of the approximation, also-called mesh or h-refinements [**43**]. Successive rows in the figure represent successively higher order polynomial interpolants, or p-refinements [**43**]. In this example, the addition of midpoint nodes with which higher order Lagrange polynomials are formulated ensures much faster convergence than an equivalent mesh refinement consisting of additional piecewise interpolations. This fact is quantified in Table 2-2 **4.** , which lists the number of DoFs n and the coefficient of determination R^2 for a piecewise interpolation divided h times using Lagrange polynomials of order p.

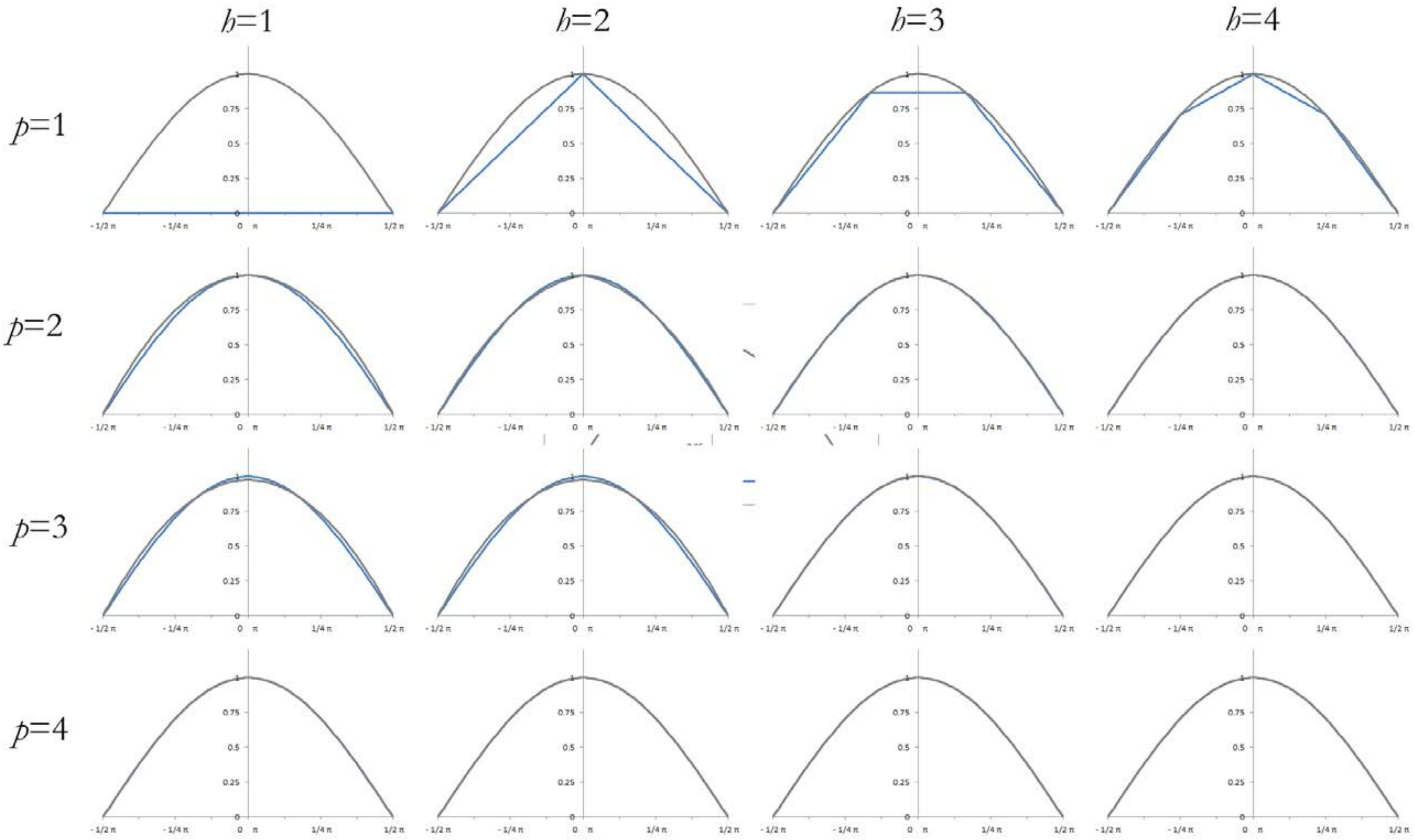


Figure 2-4: Comparison between piecewise approximations of the cosine function

Table 2-2: Comparison between piecewise approximations to the cosine function

	$h=1$	$h=2$	$h=3$	$h=4$
$p=1$	$n=2$ $R^2\approx0$	$n=3$ $R^2=0.958687$	$n=4$ $R^2=0.990288$	$n=5$ $R^2=0.996974$
$p=2$	$n=3$ $R^2=0.996725$	$n=5$ $R^2=0.999156$	$n=7$ $R^2=0.999847$	$n=9$ $R^2=0.999968$
$p=3$	$n=4$ $R^2=0.996725$	$n=7$ $R^2=0.999986$	n=10 $R^2=0.999999$	$n=13$ $R^2\approx1$
$p=4$	n=5 $R^2=999998$	$n=9$ $R^2\approx1$	$n=13$ $R^2\approx1$	$n=17$ $R^2\approx1$

Babuška and Rheinboldt's pioneering work in the field of hp-adaptive FEM formulations [**44**], further pursued by Rachowicz, Pardo and Demkowicz [**43**] outlines algorithms ensuring the most efficient combination of local mesh refinement and locally variable polynomial elements. However, although these remarkable algorithms deliver exponential convergence rates the heterogeneous adaptive nature of mesh refinement was deemed ill-suited to the application of automated optimization as this introduces variations in accuracy between optimization iterations, which would then have be accounted for and managed.

2.5.2 Generalized and Reduced Generalized Node Tetrahedra

Tian, Matsubara and Yagawa as well as Hua and To explored the use of additional nodal parameters rather than midpoint nodes in formulating higher order tetrahedral elements with only four nodes [**30**,**31**,**32**]. The generalized node tetrahedron developed by these authors is advantageous for three reasons. First, as this element is formed by the migration of the DoFs originally found at the midpoints to the four vertex nodes, the GNTet4 element is comparably accurate to ten node quadratic tetrahedra [**30**,**31**,**32**]. Second, mesh generation is simplified, eschewing the generation of midpoints. Finally, the nodal parameters which are represented correspond to meaningful physical quantities namely displacements, rotations and strains. This means that point forces, moments and strains can be applied directly at each node.

A more in-depth description of the reduced generalized node tetrahedron can be found in Section 2.5.3. Though this element does not offer full quadratic accuracy such as enabled by ten node tetrahedra or the generalized node tetrahedron described above, the RGNTet4 element is nonetheless superior in accuracy to the 4-node tetrahedron. DoFs associated to shear are omitted from this element, resulting in a reduction in accuracy from the generalized node tetrahedron. Ultimately, the RGNTet4 element was first selected for its ability to model point moments and rotations at its nodes. This was deemed essential to an approach considered initially for this work, which aimed at modeling fibrous composites using chains of 1D bar-beam elements representing orthotropic rovings suspended in a volume element mesh representing isotropic material or resin. Although this approach which parallels work done by Zhou, Sun and Wang [**45**] allows for a direct representation of roving paths and guarantees roving continuity, it also suffers from serious drawbacks when used for modeling impregnated fibres in a cured composite. Indeed, the 1D bar-beam elements considered can only account for the on-axis properties of the fibres and rovings, and they could not represent roving sections adequately without further modifications. In order to model the off-axis properties of impregnated and cured rovings, and of fibre-reinforced polymer composites properly, orthotropic volume elements were deemed superior and so the 1D elements were rejected from further development.

2.5.3 Reduced Generalized Node Shape Function

The Reduced Generalized Node Tetrahedral (RGNTET4) element is a particular class of higher order element which was chosen for the FEA formulation developed toward this thesis, not only because of its higher order accuracy but also because the nodal parameters that it employs are physically meaningful quantities. The generalized 4 node tetrahedral element and reduced generalized node tetrahedral elements were introduced in the previous section, and are explained in greater detail herein.

The response of a linear elastic body to a force involves one or a combination of four possible components: (a) rigid body translations, (b) rigid body rotations, (c) simple tension or compression, and (d) pure shear [**30**].

Figure 2-5 shows a point at $\vec{\mathbf{x}}$ displaced to $\vec{\mathbf{x}}'$ as a response to local displacements experienced near $\vec{\mathbf{x}}_i$. For clarity, the squares illustrate the displacement of the surrounding continuum for the four modes near a given point.

The reader should note that the prime operator ’ here is meant to designate the position after the transformation, and not a derivative.

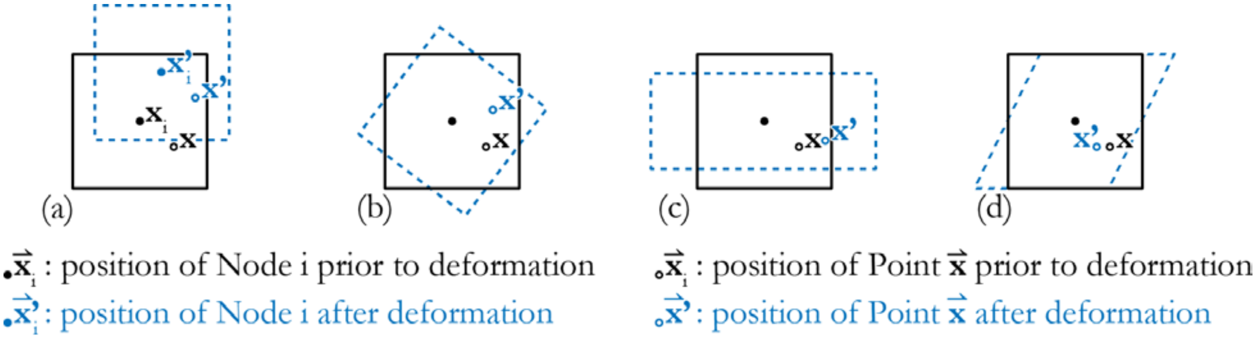


Figure 2-5: Four components of displacement
(a) rigid body translation; (b) rigid body rotation;
(c) simple tension/compression; and (d) pure shear;
adapted from [30], correction to (d)

General tetrahedral elements interpolate these displacements as follows. The displacement of point $\vec{\mathbf{x}}$ resulting from rigid body motion that occurs around node *i* can be expressed as:

$$(\vec{\mathbf{x}}' - \vec{\mathbf{x}})_{\mathrm{t}} = \vec{\mathbf{x}}_i' - \vec{\mathbf{x}}_i \tag{2.18}$$

where subscript t represents translation. More concisely:

$$\vec{\mathbf{u}}_{\mathrm{t}} = \vec{\mathbf{u}}_{\mathrm{t}\,i} \tag{2.19}$$

where $\vec{\mathbf{u}}_{\mathrm{t}}$ and $\vec{\mathbf{u}}_{\mathrm{t}\,i}$ are the displacements resulting from the translation of point at $\vec{\mathbf{x}}$ and node *i*, respectively. Likewise, the displacement of point $\vec{\mathbf{x}}$ following rigid body rotation about $\vec{\mathbf{x}}_i$ can be expressed as:

$$(\vec{\mathbf{x}}\,' - \vec{\mathbf{x}}\,)_{\mathrm{r}} = \begin{bmatrix} \tilde{x}_i \left(\cos\theta_{iy} + \cos\theta_{iz} - 2\right) + \tilde{z}_i \sin\theta_{iy} - \tilde{y}_i \sin\theta_{iz} \\ -\tilde{z}_i \sin\theta_{ix} + \tilde{\mathrm{y}}_{\mathrm{i}} \left(\cos\theta_{iz} + \cos\theta_{ix} - 2\right) + \tilde{x}_i \sin\theta_{iz} \\ \tilde{y}_i \sin\theta_{ix} - \tilde{x}_i \sin\theta_{iy} + \tilde{z}_i \left(\cos\theta_{ix} + \cos\theta_{iy} - 2\right) \end{bmatrix} \tag{2.20}$$

where the subscript r represents rotation and $\tilde{x}_i$, $\tilde{y}_i$ and $\tilde{z}_i$ represent the distances $(x - x_i)$, $(y - y_i)$ and $(z - z_i)$ from arbitrary point $\vec{x}$ to node i.

The previous equation which originates from the rotation transformation is valid for any angle θ_{ia}, where a indicates the rotation axis (x, y or z). For small rotations the equation can be linearized with little loss of precision:

$$(\vec{\mathbf{x}}\,' - \vec{\mathbf{x}}\,)_{\mathrm{r}} = \begin{bmatrix} 0 & \tilde{z}_i & -\tilde{y}_i \\ -\tilde{z}_i & 0 & \tilde{x}_i \\ \tilde{y}_i & -\tilde{x}_i & 0 \end{bmatrix} \begin{bmatrix} \theta_{ix} \\ \theta_{iy} \\ \theta_{iz} \end{bmatrix} \tag{2.21}$$

where the tilde notation, for example $\tilde{\mathrm{c}}_{\mathrm{i}}$, represents $(c - c_i)$ with c being x, y or z. More concisely we may denote the linearized rotation matrix in equation (2.21) as $F_i^{\boldsymbol{\theta}}(\vec{x}\,)$:

$$\vec{\mathbf{u}}_{\mathrm{r}} = \mathbf{F}_i^{\boldsymbol{\theta}}(\vec{\mathbf{x}}\,)\,\vec{\boldsymbol{\theta}}_i \tag{2.22}$$

where $\vec{\mathbf{u}}_{\mathrm{r}}$ is the displacement resultant from the rotation of the point at $\vec{\mathbf{x}}$ about node i, and $\vec{\boldsymbol{\theta}}_i$ is the 3 component column vector $[\theta_{\mathrm{ix}}\ \theta_{\mathrm{iy}}\ \theta_{\mathrm{iz}}]^{\mathrm{T}}$ containing the local rotations around axes x, y and z at node i.

Generalized node elements also contain components representing tension/compression and shear at node *i*. In-depth treatment of the generalized node is available in [**30**,**31**,**32**]. The reduced generalized node element only factors translations and rotations as described above. As the reduced generalized node element was selected ultimately towards the FEA solver implemented in this thesis, it is discussed more thoroughly below. Let the reduced generalized node local displacement vector $\overrightarrow{\mathbf{u_{ei}}}$ be arranged as follows:

$$\overrightarrow{\mathbf{u}_{\mathbf{e}i}} = \left[\begin{array}{ccc|ccc} u & v & w & \theta_{ix} & \theta_{iy} & \theta_{iz} \end{array} \right]^{\mathrm{T}} = \left[\begin{array}{c|c} \overrightarrow{\mathbf{u}_{\mathrm{t}i}} & \vec{\boldsymbol{\theta}}_i \end{array} \right]^{\mathrm{T}} \qquad \textbf{(2.23)}$$

where u, v and w are the translational displacements in x, y and z of node i, and θ_{ix}, θ_{iy} and θ_{iz} are the rotational displacements about axes x, y and z at node i. The pipe operator | represents the concatenation of the two vectors.

From linearity, the shape function relating displacement $\vec{\mathbf{u}} = \left[\begin{array}{ccc} u & v & w \end{array} \right]^{\mathrm{T}}$ of a point within a four node tetrahedron (i = 1, 2, 3, 4) is the superposition of the displacements caused by the translation and rotation of the nodes, or:

$$\vec{\mathbf{u}} = \sum_{i=1}^{4} \zeta_i \left[\begin{array}{c|c} \mathbf{I} & \mathbf{F}_i^{\boldsymbol{\theta}}(\vec{\mathbf{x}}) \end{array} \right] \overrightarrow{\mathbf{u}_{\mathbf{e}i}} \qquad \textbf{(2.24)}$$

where $\vec{\mathbf{u}}$ is a column vector describing the displacements in x, y and z taking into account nodal rotations, ζ_i are the tetrahedral coordinates, **I** is a (3 x 3) identity matrix and $F_i^{\boldsymbol{\theta}}(\vec{x})$ is the (3 x 3) displacement function of point $\vec{\mathbf{x}}$ due to the rotation of the point about node i. Furthermore,

$\overrightarrow{\mathbf{u}_{\mathbf{e}\,i}} = \left[\, u \;\; v \;\; w \mid \theta_{ix} \;\; \theta_{iy} \;\; \theta_{iz} \,\right]^{\mathrm{T}}$ is the reduced generalized nodal displacement vector, which describes the displacements and rotations at node i. Figure 2-6 illustrates the DoFs, at each node for the RGNTet4.

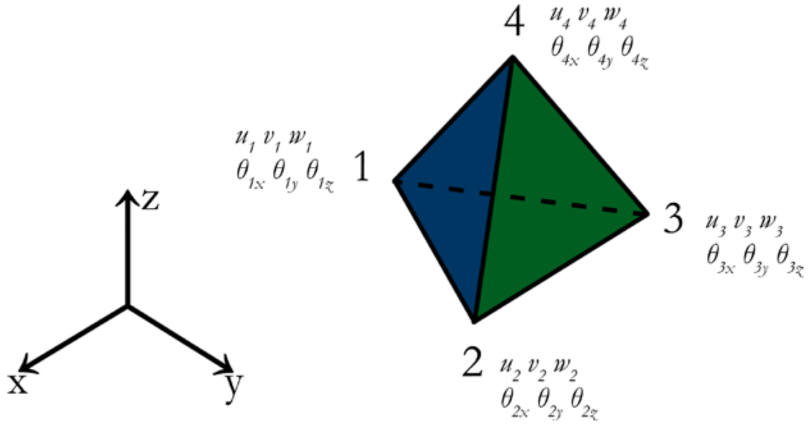


Figure 2-6: RGNTet4 element and its degrees of freedom

Concatenating the terms summed up in **(2.24)** into a single matrix multiplication between a 3x24 shape matrix **N** and a 24 DoF RGN element displacement vector $\overrightarrow{\mathbf{u}_{\mathbf{e}}} = \left[\, \overrightarrow{\mathbf{u}_{\mathbf{e}\,1}} \mid \overrightarrow{\mathbf{u}_{\mathbf{e}\,2}} \mid \overrightarrow{\mathbf{u}_{\mathbf{e}\,3}} \mid \overrightarrow{\mathbf{u}_{\mathbf{e}\,4}} \,\right]$ leads to:

$$\vec{\mathbf{u}} = \left[\, \zeta_1 \left[\mathbf{I} \,\middle|\, \mathbf{F}_1^{\boldsymbol{\theta}}(\,\vec{\mathbf{x}}\,) \right] \middle| \zeta_2 \left[\mathbf{I} \,\middle|\, \mathbf{F}_2^{\boldsymbol{\theta}}(\,\vec{\mathbf{x}}\,) \right] \middle| \zeta_3 \left[\mathbf{I} \,\middle|\, \mathbf{F}_3^{\boldsymbol{\theta}}(\,\vec{\mathbf{x}}\,) \right] \middle| \zeta_4 \left[\mathbf{I} \,\middle|\, \mathbf{F}_4^{\boldsymbol{\theta}}(\,\vec{\mathbf{x}}\,) \right] \right] \begin{bmatrix} \overrightarrow{\mathbf{u}_{\mathbf{e}\,1}} \\ \text{—} \\ \overrightarrow{\mathbf{u}_{\mathbf{e}\,2}} \\ \text{—} \\ \overrightarrow{\mathbf{u}_{\mathbf{e}\,3}} \\ \text{—} \\ \overrightarrow{\mathbf{u}_{\mathbf{e}\,4}} \end{bmatrix} \qquad \textbf{(2.25)}$$

2.6. Element Strain Field

2.6.1 Introduction

Thus far, an effective way of modeling arbitrary linear functions such as displacements has been presented using the linear, general and reduced generalized node tetrahedra. However, a closer look at strain theory is warranted in order to link the displacements at the nodes of the elements examined earlier to the constitutive equation which will describe the elastic behaviour of these elements.

2.6.2 Strain Tensors

A strain is defined as a measure of deformation representing the relative displacement of particles in a body relative to a reference length. This quantity is of great importance in solid mechanics. The following sections will describe its use in the finite element formulation implemented in this work.

The finite strain theory is used for dealing with deformations where both rotations and strains are arbitrarily large. Such formulations are nonlinear and far more complex to deal with than those entailed by infinitesimal strain theory. For small deformations, the latter theory coupled with the engineering definition of strain suffice for describing displacements in the material. Consequently, the infinitesimal strain theory was used in this thesis, under the assumption that the strains and rotations undergone by the materials being modeled are sufficiently small.

Components of engineering strain are dependent on variations in the local displacement field. They are defined by the following relation [**46**]:

$$\boldsymbol{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{2.26}$$

where u_i are components of the element displacement field $\vec{\mathbf{u}} = \left[\, u \;\; v \;\; w \,\right]^{\mathrm{T}}$ and x_i are directions x, y and z. Examination of the subscripts reveals that equation **(2.26)** describes a normal strain when subscripts i and j are equal. Otherwise, equation **(2.26)** describes a shear strain. Reordering the sum in Equation **(2.26)** gives the equivalence $\varepsilon_{ij} = \varepsilon_{ji}$. As a consequence, the strain tensor is symmetric.

2.6.3 Strain Vector Definition

The infinitesimal strain tensor describes the infinitesimal deformations of a continuum in 3D. It is commonly written as a (3 X 3) tensor:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \gamma_{xy}/2 & \gamma_{xz}/2 \\ \gamma_{yx}/2 & \varepsilon_{yy} & \gamma_{yz}/2 \\ \gamma_{zx}/2 & \gamma_{zy}/2 & \varepsilon_{zz} \end{bmatrix} \tag{2.27}$$

where $\boldsymbol{\varepsilon}$ is the (3 X 3) infinitesimal strain tensor, ε_{xx}, ε_{yy} and ε_{zz} are the normal engineering strains in the x, y and z directions and γ_{xy} , γ_{xz} , γ_{yx} , γ_{yz} , γ_{zx} and γ_{zy} are the engineering shear strains which correspond to 2 ε_{ij} in the previous notation. As the strain tensor is symmetric, it can be shown that:

$$\gamma_{xy} = \gamma_{yx} \qquad \gamma_{xz} = \gamma_{zx} \qquad \gamma_{yz} = \gamma_{zy} \tag{2.28}$$

Therefore, the local state of strain in the material can be described by 6 parameters, arranged here as a column vector:

$$\vec{\boldsymbol{\varepsilon}} = [\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}]^{\mathrm{T}} \tag{2.29}$$

This notation is convenient as it will allow for the straightforward calculation of stress and strain transforms which can then be used for describing off-axis orthotropic materials such as those being modeled in this thesis, as discussed in Section 2.7.4.

2.6.4 Strain Field Calculation

Sections 2.4.4 and 2.5.3 described the shape functions for the linear tetrahedral and the reduced generalized node tetrahedral elements, respectively. These functions provide the means to obtain the displacements within an element $\vec{\mathbf{u}}$ as a function of a set of nodal DoFs $\mathbf{u_e}$. In turn, using the definition of strain in **(2.26)** the strain field can be calculated from this local displacement field:

$$\vec{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{D}\,\vec{\mathbf{u}} \qquad \textbf{(2.30)}$$

where **D** is called the linear strain operator. Combining equations **(2.11)** or **(2.25)** with equation (2.30), the strain at a point in the element is expressed as a function of tetrahedral coordinates:

$$\vec{\boldsymbol{\varepsilon}} = \mathbf{D}\,\mathbf{N}\,\vec{\mathbf{u}}_{\mathbf{e}} \qquad \textbf{(2.31)}$$

The product of **D** and **N** is **B**, which represents the (6 x 24) matrix relation between the element displacement vector $\vec{\mathbf{u}}_{\mathbf{e}}$ and the strain at a given point within the element:

$$\vec{\boldsymbol{\varepsilon}} = \mathbf{B}\,\vec{\mathbf{u}}_{\mathbf{e}} \qquad \textbf{(2.32)}$$

After applying the derivative rules in **(2.14)**, **B** can be evaluated explicitly using the following sum:

$$\mathbf{B} = \sum_{i=1}^{4} \zeta_i \left[\mathbf{b}_{i,1} \,\middle|\, \mathbf{b}_{i,2} \,\middle|\, \mathbf{b}_{i,3} \,\middle|\, \mathbf{b}_{i,4} \right] \qquad \textbf{(2.33)}$$

with:

$$\mathbf{b}_{i,j} = \frac{1}{6\,\Omega}\begin{bmatrix} a_i & 0 & 0 \\ 0 & b_i & 0 \\ 0 & 0 & c_i \\ b_i & a_i & 0 \\ 0 & c_i & b_i \\ c_i & 0 & a_i \end{bmatrix}\left[\mathbf{I}\,\middle|\,\mathbf{F}_i^{\boldsymbol{\theta}}(\,\vec{\mathbf{x}}\,)\right] \qquad \textbf{(2.34)}$$

where ζ_i is the tetrahedral coordinate corresponding to node i. The product of N_i with the (6 x 6) block $\mathbf{b}_{i,j}$ describes the strain as a function of the displacement of node j. From equations **(2.33)** and **(2.34)**, matrix **B** can be evaluated explicitly as a function of tetrahedral coordinates.

2.7. Material Elasticity Matrix

2.7.1 Hooke's Law in Three Dimensions

In order to calculate the element stiffness matrix in Section 2.8, it is necessary to develop beforehand a complete definition of the elastic behaviour of the material being modelled. Perhaps one of the most used and best understood constitutive relations is Hooke's law, which relates stresses and strains at a given point within a linearly elastic material and thereby provides the foundation for the mathematical theory of linear elasticity [**47**]. Specifically, Hooke's law is a linear relation between each stress-tensor component σ_{ij} and all strain tensor components ε_{kl} [**47**]:

$$\vec{\boldsymbol{\sigma}}_{ij} = \sum_{k=1}^{3}\sum_{l=1}^{3} c_{ijkl}\,\varepsilon_{ijkl} \qquad i,j \in \{1, 2, 3\} \qquad \textbf{(2.35)}$$

By rearranging the components of the strain tensor as done in **(2.29)**, and likewise for the stress tensor, relation **(2.35)** can be expressed in matrix form [**47**]:

$$\vec{\boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} & C_{1,5} & C_{1,6} \\ C_{2,1} & C_{2,2} & C_{2,3} & C_{2,4} & C_{2,5} & C_{2,6} \\ C_{3,1} & C_{3,2} & C_{3,3} & C_{3,4} & C_{3,5} & C_{3,6} \\ C_{4,1} & C_{4,2} & C_{4,3} & C_{4,4} & C_{4,5} & C_{4,6} \\ C_{5,1} & C_{5,2} & C_{5,3} & C_{5,4} & C_{5,5} & C_{5,6} \\ C_{6,1} & C_{6,2} & C_{6,3} & C_{6,4} & C_{6,5} & C_{6,6} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \mathbf{E}\,\vec{\boldsymbol{\varepsilon}} \qquad \mathbf{(2.36)}$$

where **E** is known as the elasticity matrix and $\vec{\boldsymbol{\sigma}}$ is the stress vector.

From symmetry of the elasticity matrix and the strain tensor it can be shown that the general stiffness matrix, equation **(2.36)** can have at most 21 independent parameters; however, most engineering materials can be modeled using fewer parameters.

2.7.2 Hooke's Law for Isotropic Materials

Materials for which measured properties such as the modulus are independent of orientation are called isotropic. Glass, elastomers and certain polymers have an amorphous microstructure; as a consequence their behaviour is generally isotropic. Furthermore, several metallic alloys have a polycrystalline microstructure in which the crystallographic orientations and grain aspect ratios are random. Although each crystalline grain may behave in an anisotropic manner, the stochastic distribution of grain orientations yields a material where properties along a given direction represent an average of the directional properties across the random distribution of the microstructures.

In the case of isotropic materials, the elasticity matrix described in equation **(2.36)** can be simplified to [**29**]:

$$\vec{\boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{bmatrix} = \frac{\mathrm{E}}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2-\nu \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \mathbf{E}\ \vec{\boldsymbol{\varepsilon}} \qquad \textbf{(2.37)}$$

where E is Young’s modulus and ν is Poisson’s ratio.

2.7.3 Hooke’s Law for Orthotropic Materials

The orthotropic material compliance matrix **S** takes the form:

$$\vec{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{1}{\mathrm{E_x}} & -\frac{\nu_{\mathrm{yx}}}{\mathrm{E_y}} & -\frac{\nu_{\mathrm{zx}}}{\mathrm{E_z}} & 0 & 0 & 0 \\ -\frac{\nu_{\mathrm{xy}}}{\mathrm{E_x}} & \frac{1}{\mathrm{E_y}} & -\frac{\nu_{\mathrm{zy}}}{\mathrm{E_z}} & 0 & 0 & 0 \\ -\frac{\nu_{\mathrm{xz}}}{\mathrm{E_x}} & -\frac{\nu_{\mathrm{yz}}}{\mathrm{E_y}} & \frac{1}{\mathrm{E_z}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\mathrm{G_{xy}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\mathrm{G_{yz}}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\mathrm{G_{zx}}} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{bmatrix} = \mathbf{S}\,\vec{\boldsymbol{\sigma}} \qquad \textbf{(2.38)}$$

The on-axis orthotropic elasticity matrix $\mathbf{E'}$ is obtained by taking the inverse of the compliance matrix:

$$\vec{\boldsymbol{\sigma}} = \mathbf{S}^{-1}\,\vec{\boldsymbol{\varepsilon}} = \mathbf{E'}\,\vec{\boldsymbol{\varepsilon}} \qquad \textbf{(2.39)}$$

Naturally, because the moduli and Poisson’s ratios are different along the three principal axes of an orthotropic material, the elasticity matrix is dependent on the orientation of the material. Equations **(2.38)** and **(2.39)** are valid only when the material is aligned along x. To calculate the elastic response of an arbitrarily aligned orthotropic material, transformation matrices must first be calculated to express the resulting stresses and strains as a function of the on-axis elasticity matrix.

2.7.4 Off-Axis Orthotropic Materials Transformation

This section deals with the derivation of transformation matrices for both stresses and strains, needed for defining off-axis orthotropic elasticity matrices. Details regarding the transformation matrices used to account for the composite material orientation as a field across the optimized part are given in Section 2.7.7.

A change of orthonormal coordinate system in 3D can be represented by a transformation matrix [**48**]:

$$\vec{\mathbf{x}}\,' = \mathbf{A}\,\vec{\mathbf{x}} \tag{2.40}$$

where $\vec{\mathbf{x}}$ and $\vec{\mathbf{x}}\,'$ represent a 3D position vector in the original and transformed coordinate systems, respectively. As mentioned in Section 2.5.3, the prime operator ’ is meant to designate the position after the transformation, and not a derivative. The reverse transformation can be obtained by inverting the relation:

$$\vec{\mathbf{x}} = \mathbf{A}^{-1}\,\vec{\mathbf{x}}\,' \tag{2.41}$$

Matrices describing distance-preserving transformations such as translations, rotations and reflections are orthogonal and therefore have the following property [**49**]:

$$\mathbf{A}^{-1} = \mathbf{A}^{\mathrm{T}} \tag{2.42}$$

Using the notation used in **(2.26)** and **(2.35)**, equations **(2.40)** and **(2.42)** take the form:

$$x'_i = \sum_{n=1}^{3} A_{ni} x_i \tag{2.43}$$

$$x_i = \sum_{n=1}^{3} A_{in} x'_n \tag{2.44}$$

where x_i and x'_i are the directions x, y and z for i and $n = 1, 2$ and 3. Deriving equation **(2.43)** with respect to x_i yields:

$$\frac{\partial x'_n}{\partial x_i} = A_{ni} \tag{2.45}$$

From the chain rule a function in x_i can be derived through:

$$\frac{\partial f(\vec{\mathbf{x}})}{\partial x_i} = \sum_{n=1}^{3} \frac{\partial f(\vec{\mathbf{x}})}{\partial x'_n} \frac{\partial x'_n}{\partial x_i} = \sum_{n=1}^{3} \frac{\partial f(\vec{\mathbf{x}})}{\partial x'_n} A_{ni} \tag{2.46}$$

Similarly, displacements can be transformed via:

$$u_i = \sum_{m=1}^{3} A_{im} u'_m \tag{2.47}$$

where u_i and u'_m are components of the 3D displacement vector before and after the transformation, respectively. Recalling equation **(2.26)** and substituting **(2.47)** yields:

$$\varepsilon_{ij} = \sum_{m=1}^{3} \frac{1}{2} \left(\frac{\partial}{\partial x_j} \left(A_{im} u'_m \right) + \frac{\partial}{\partial x_i} \left(A_{jm} u'_m \right) \right) \tag{2.48}$$

As **A** is constant with respect to x_i this relation is simplified to:

$$\varepsilon_{ij} = \sum_{m=1}^{3} \frac{1}{2} \left(A_{im} \frac{\partial u'_m}{\partial x_j} + A_{jm} \frac{\partial u'_m}{\partial x_i} \right) \tag{2.49}$$

Finally, considering **(2.46)** and u'_m as the function being derived in x_i and x_j ,

$$\varepsilon_{ij} = \sum_{m=1}^{3} \sum_{n=1}^{3} \frac{1}{2} \left(A_{im} \frac{\partial u'_m}{\partial x'_n} A_{nj} + A_{jm} \frac{\partial u'_m}{\partial x'_n} A_{ni} \right) \tag{2.50}$$

Reverting to matrix form and inverting this relation leads to the stress transformation described in [**49**]:

$$\boldsymbol{\varepsilon}' = \mathbf{A}\, \boldsymbol{\varepsilon}\, \mathbf{A}^{\mathrm{T}} \tag{2.51}$$

where ε and ε' correspond to the strain tensor before and after the change of coordinates respectively, and **A** represents the (3 x 3) transformation matrix which describes this change.

As noted in **(2.30)**, since the stress tensor is symmetric it can be represented by a 6-parameter column vector, and consequently it is possible to calculate a single (6 x 6) transformation matrix to describe a distance-preserving change of coordinates by calculating each of its columns sequentially [**49**]:

$$\vec{\boldsymbol{\varepsilon}}\,' = \mathbf{A}_{\varepsilon}\, \vec{\boldsymbol{\varepsilon}} \tag{2.52}$$

where $\vec{\boldsymbol{\varepsilon}}$ and $\vec{\boldsymbol{\varepsilon}}\,'$ are the 6 parameter strain vectors congruent with **(2.30)** before and after the change of coordinates respectively, and $\mathbf{A}_{\varepsilon}$ is the (6 x 6) strain transformation matrix which relates them to each other. The matrix $\mathbf{A}_{\varepsilon}$ can be written out explicitly as [**49**]:

$$\mathbf{A}_\varepsilon = \begin{bmatrix} A_{11}\,A_{11} & A_{12}\,A_{12} & A_{13}\,A_{13} & A_{12}\,A_{13} & A_{11}\,A_{13} & A_{11}\,A_{12} \\ A_{21}\,A_{21} & A_{22}\,A_{22} & A_{23}\,A_{23} & A_{22}\,A_{23} & A_{21}\,A_{23} & A_{21}\,A_{22} \\ A_{31}\,A_{31} & A_{32}\,A_{32} & A_{33}\,A_{33} & A_{32}\,A_{33} & A_{31}\,A_{33} & A_{31}\,A_{32} \\ 2\,A_{21}\,A_{31} & 2\,A_{22}\,A_{32} & 2\,A_{23}\,A_{33} & A_{22}\,A_{33} + A_{23}\,A_{32} & A_{21}\,A_{33} + A_{23}\,A_{31} & A_{21}\,A_{32} + A_{22}\,A_{31} \\ 2\,A_{11}\,A_{31} & 2\,A_{12}\,A_{32} & 2\,A_{13}\,A_{33} & A_{12}\,A_{33} + A_{13}\,A_{32} & A_{11}\,A_{33} + A_{13}\,A_{31} & A_{11}\,A_{32} + A_{12}\,A_{31} \\ 2\,A_{11}\,A_{21} & 2\,A_{12}\,A_{22} & 2\,A_{13}\,A_{23} & A_{12}\,A_{23} + A_{13}\,A_{22} & A_{11}\,A_{23} + A_{13}\,A_{21} & A_{11}\,A_{22} + A_{12}\,A_{21} \end{bmatrix}$$

(2.53)

Similarly it can be shown that the stress transform for a change of coordinates described by the matrix **A** equates to [**49**]:

$$\boldsymbol{\sigma}' = \mathbf{A}^{T}\boldsymbol{\sigma}\,\mathbf{A} \qquad \textbf{(2.54)}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ are the (3 x 3) stress tensors before and after the change of coordinates, respectively. As done in equation **(2.52)**, this operation can be performed as a single matrix product between the original 6-parameter stress vector and a (6 x 6) transformation matrix:

$$\vec{\boldsymbol{\sigma}}' = \mathbf{A}_{\boldsymbol{\sigma}}\vec{\boldsymbol{\sigma}} \qquad \textbf{(2.55)}$$

where $\vec{\sigma}$ and $\vec{\sigma}'$ are the 6 parameter stress vectors congruent with **(2.36)** before and after the change of coordinates respectively, and $\mathbf{A}_\sigma$ is the (6 x 6) transformation matrix which relates them to each other. Explicitly, $\mathbf{A}_\sigma$ takes the form [**49**]:

$$\mathbf{A_\sigma} = \begin{bmatrix} A_{11}\,A_{11} & A_{12}\,A_{12} & A_{13}\,A_{13} & 2\,A_{12}\,A_{13} & 2\,A_{11}\,A_{13} & 2\,A_{11}\,A_{12} \\ A_{21}\,A_{21} & A_{22}\,A_{22} & A_{23}\,A_{23} & 2\,A_{22}\,A_{23} & 2\,A_{21}\,A_{23} & 2\,A_{21}\,A_{22} \\ A_{31}\,A_{31} & A_{32}\,A_{32} & A_{33}\,A_{33} & 2\,A_{32}\,A_{33} & 2\,A_{31}\,A_{33} & 2\,A_{31}\,A_{32} \\ A_{21}\,A_{31} & A_{22}\,A_{32} & A_{23}\,A_{33} & A_{22}\,A_{33} + A_{23}\,A_{32} & A_{21}\,A_{33} + A_{23}\,A_{31} & A_{21}\,A_{32} + A_{22}\,A_{31} \\ A_{11}\,A_{31} & A_{12}\,A_{32} & A_{13}\,A_{33} & A_{12}\,A_{33} + A_{13}\,A_{32} & A_{11}\,A_{33} + A_{13}\,A_{31} & A_{11}\,A_{32} + A_{12}\,A_{31} \\ A_{11}\,A_{21} & A_{12}\,A_{22} & A_{13}\,A_{23} & A_{12}\,A_{23} + A_{13}\,A_{22} & A_{11}\,A_{23} + A_{13}\,A_{21} & A_{11}\,A_{22} + A_{12}\,A_{21} \end{bmatrix} \tag{2.56}$$

To obtain the transformation matrix $\mathbf{A}_\sigma^{-1}$ which performs the inverse change of coordinates, one must first invert relation **(2.54)** recalling that **A** is orthogonal **(2.42)**:

$$\boldsymbol{\sigma} = \mathbf{A}\,\boldsymbol{\sigma}'\,\mathbf{A}^{\mathrm{T}} \tag{2.57}$$

Similarly, this expression can be rewritten with 6 parameter stress vectors as done in **(2.55)**:

$$\vec{\boldsymbol{\sigma}} = \mathbf{A}_{\boldsymbol{\sigma}}^{-1}\,\vec{\boldsymbol{\sigma}}\,' \tag{2.58}$$

Explicitly, $\mathbf{A}_\sigma^{-1}$ takes the following form, which is identical to **(2.56)** but with the terms of matrix **A** transposed [**50**]:

$$\mathbf{A}_\sigma^{-1} = \begin{bmatrix} A_{11}\,A_{11} & A_{21}\,A_{21} & A_{31}\,A_{31} & 2\,A_{21}\,A_{31} & 2\,A_{11}\,A_{31} & 2\,A_{11}\,A_{21} \\ A_{12}\,A_{12} & A_{22}\,A_{22} & A_{32}\,A_{32} & 2\,A_{22}\,A_{32} & 2\,A_{12}\,A_{32} & 2\,A_{12}\,A_{22} \\ A_{13}\,A_{13} & A_{23}\,A_{23} & A_{33}\,A_{33} & 2\,A_{23}\,A_{33} & 2\,A_{13}\,A_{33} & 2\,A_{13}\,A_{23} \\ A_{12}\,A_{13} & A_{22}\,A_{23} & A_{32}\,A_{33} & A_{22}\,A_{33} + A_{32}\,A_{23} & A_{12}\,A_{33} + A_{32}\,A_{13} & A_{12}\,A_{23} + A_{22}\,A_{13} \\ A_{11}\,A_{13} & A_{21}\,A_{23} & A_{31}\,A_{33} & A_{21}\,A_{33} + A_{31}\,A_{23} & A_{11}\,A_{33} + A_{31}\,A_{13} & A_{11}\,A_{23} + A_{21}\,A_{13} \\ A_{11}\,A_{12} & A_{21}\,A_{22} & A_{31}\,A_{32} & A_{21}\,A_{32} + A_{31}\,A_{22} & A_{11}\,A_{32} + A_{31}\,A_{12} & A_{11}\,A_{22} + A_{21}\,A_{12} \end{bmatrix} \tag{2.59}$$

Notably, this relation is exactly equal to $\mathbf{A}_{\varepsilon}{}^{\mathrm{T}}$, therefore:

$$\mathbf{A}_{\boldsymbol{\sigma}}^{-1} = \mathbf{A}_{\boldsymbol{\varepsilon}}^{T} \tag{2.60}$$

2.7.5 Hooke's Law for Off-Axis Orthotropic Materials

Given equations **(2.39)**, **(2.52)** and **(2.60)** which describe Hooke's law for orthotropic materials, the strain coordinate transform as well as inverse stress coordinate transform, the off-axis orthotropic stiffness matrix can be derived as follows. The stresses and strains oriented about an arbitrary coordinate system at a point in a given orthotropic material are described by vectors $\vec{\boldsymbol{\sigma}}'$ and $\vec{\boldsymbol{\varepsilon}}'$ respectively. Prior to an off-axis transformation, Hooke's law relates these two tensors are related through the on-axis stiffness tensor **E**':

$$\vec{\boldsymbol{\sigma}}' = \mathbf{E}'\,\vec{\boldsymbol{\varepsilon}}' \tag{2.61}$$

This relation can be expressed as a function of on-axis stresses and strains through the change of coordinate transformations described in **(2.52)** and **(2.60):**

$$\mathbf{A}_{\boldsymbol{\sigma}}\vec{\boldsymbol{\sigma}} = \mathbf{E}'\mathbf{A}_{\boldsymbol{\varepsilon}}\vec{\boldsymbol{\varepsilon}} \tag{2.62}$$

Multiplying both sides by $\mathbf{A}_{\sigma}^{-1}$ and applying **(2.60)** yields Hooke's law for off-axis orthotropic materials:

$$\vec{\sigma} = \mathbf{A}_{\varepsilon}^{T} \mathbf{E}' \mathbf{A}_{\varepsilon} \vec{\varepsilon} \tag{2.63}$$

Explicitly, the off-axis transformed elasticity matrix **E** describes the relation between stresses and strains as a function of the coordinate transform given by the matrix **A**:

$$\mathbf{E} = \mathbf{A}_{\varepsilon}^{T} \mathbf{E}' \mathbf{A}_{\varepsilon} \tag{2.64}$$

2.7.6 Hooke's Law for Fibre-Reinforced Composite Materials

A common approach used for modeling the microstructure of fibre-reinforced structures consists in making the boundaries between physical volumes occupied by rovings and physical volumes occupied exclusively by surrounding reinforcing matrix coincide with finite element faces [**2**,**3**,**4**,**5**,**6**,**7**]. Material properties corresponding to the rovings and neat matrix can be applied to elements which are located within the envelope of either roving or matrix, respectively. However, the mesh resolution required for achieving such intricate geometrical detail for meshes at mesoscopic scales whilst retaining adequate element aspect ratios makes the problem intractable computationally in the context of the optimization scheme described in Sections 2.12 and 4.1. Therefore, another approach was retained towards mesh division, where parts analyzed would be segmented into meshes the topology and geometry of which are independent of the material structure.

Instead, a field of vectors was used for representing concisely the constitutive properties of the composite material throughout the optimized part, specifying a square symmetric material in the plane normal to local vector orientation. To this end, each element in the discretized mesh was assigned a vector. The direction of each vector within the field represents the mean orientation of the material at that location. Furthermore, the Euclidian norm of these vectors represents the volume fraction of fibres at that location. Faced with a similar problem, Kim and Swan utilized a comparable method to model accurately the mechanical behaviour of a composite weave at the micro and meso scales [**51**]. The focus in Kim and Swan's work was to calculate most accurately the mechanical properties over the volume being modeled. In comparison, the voxel method implemented in the scope of this thesis aims instead at reducing the computational complexity resultant from adaptive meshing. Further, the voxel method ensures that the optimization metric based on the part behaviour calculated by FEA as a function of the material structure will be free of discontinuities resultant from remeshing of the solution domain. Additional information concerning the vectorial representation of material properties and the selection of material parameters can be found in Section 4.1. Details about the decomposition of the vector components and their impact on the fibre-reinforced composite modeled therewith follow.

Section 2.7.5 derives equations which describe the off-axis orthotropic material stiffness matrix as a function of the on-axis stiffness matrix **E**' and the orthogonal transformation matrix **A**. In the context of an off-axis fibre-reinforced composite material stiffness matrix, the purpose of the transformation matrix is to align the matrix along the fibre direction specified by the vector $\vec{\mathbf{p}}_e$. Therefore, **A** represents the transformation matrix which corresponds to a rotation between the on-axis longitudinal fibre direction along the x-axis, and the direction given by vector $\vec{\mathbf{p}}_e$.

To calculate **A**, the unit vector $\overrightarrow{\lambda_p}$ which is aligned along $\overrightarrow{\mathbf{p}_{\mathrm{e}}}$ is first calculated:

$$\overrightarrow{\lambda_{\mathbf{p}}} = \frac{\overrightarrow{\mathbf{p}_{\mathrm{e}}}}{\left|\overrightarrow{\mathbf{p}_{\mathrm{e}}}\right|} = \frac{\overrightarrow{\mathbf{p}_{\mathrm{e}}}}{\sqrt{p_{ex}^2 + p_{ey}^2 + p_{ez}^2}} \tag{2.65}$$

Taking the cross product of the unit vector $\overrightarrow{\lambda_p}$ with a unit vector aligned along the x-axis yields the axis about which the system must be rotated:

$$\vec{\boldsymbol{\omega}} = \vec{\mathbf{i}} \times \overrightarrow{\lambda_{\mathbf{p}}} \tag{2.66}$$

where $\vec{\boldsymbol{\omega}}$ is the axis about which the material is rotated and $\vec{\mathbf{i}}$ is a unit vector aligned along the positive x-axis.

Because both $\vec{\mathbf{i}}$ and $\overrightarrow{\lambda_p}$ are unit vectors, their dot product corresponds directly to the cosine of the angle about which the material is to be rotated:

$$\omega_c = \vec{\mathbf{i}} \cdot \overrightarrow{\lambda_{\mathbf{p}}} = \left|\vec{\mathbf{i}}\right| \left|\overrightarrow{\lambda_{\mathbf{p}}}\right| \cos(\theta) = \cos(\theta) \tag{2.67}$$

Therefore, this angle corresponds to:

$$\theta = \arccos\left(\vec{\mathbf{i}} \cdot \overrightarrow{\lambda_{\mathbf{p}}}\right) \tag{2.68}$$

The rotation matrix $\mathbf{A}_\mathrm{R}$ associated with axis $\vec{\boldsymbol{\omega}}$ and angle θ can be calculated with Rodrigues' rotation equation [**52**]:

$$\mathbf{A_R} = \mathbf{I} + [\boldsymbol{\omega}]_\times \sin(\theta) + ([\boldsymbol{\omega}]_\times)^2 (1 - \cos(\theta)) \qquad \textbf{(2.69)}$$

where $[\boldsymbol{\omega}]_\times$ corresponds to the cross product matrix, a function of the vector $\vec{\boldsymbol{\omega}}$ which performs the cross product operation between $\vec{\boldsymbol{\omega}}$ and any $\mathcal{R}^3$ vector multiplied with it:

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & -0 \end{bmatrix} \qquad \textbf{(2.70)}$$

where ω_x, ω_y and ω_z are the components of vector $\vec{\boldsymbol{\omega}}$. As vector $\vec{\boldsymbol{\omega}}$ is a unit vector, the term $([\boldsymbol{\omega}]_\times)^2$ can be expressed instead as a function of the tensor product $\vec{\boldsymbol{\omega}} \cdot \vec{\boldsymbol{\omega}}^\mathrm{T}$:

$$([\boldsymbol{\omega}]_\times)^2 = \begin{bmatrix} -\omega_z^2 - \omega_y^2 & \omega_y \omega_x & \omega_z \omega_x \\ \omega_y \omega_x & -\omega_z^2 - \omega_x^2 & \omega_z \omega_y \\ \omega_z \omega_x & \omega_z \omega_y & -\omega_y^2 - \omega_x^2 \end{bmatrix} = \mathbf{I} - \vec{\omega} \cdot \vec{\omega}^\mathrm{T} \qquad \textbf{(2.71)}$$

Substituting this equation in (2.69) yields the alternative form:

$$\mathbf{A_R} = \mathbf{I} \cdot \cos(\theta) + [\boldsymbol{\omega}]_\times \sin(\theta) + \vec{\boldsymbol{\omega}} \cdot \vec{\boldsymbol{\omega}}^{\mathrm{T}}(1 - \cos(\theta)) \qquad \textbf{(2.72)}$$

Then, substituting **(2.68)** into **(2.72)** yields the rotation matrix which aligns the material along the axis $\overrightarrow{\lambda_{\mathbf{p}}}$:

$$\mathbf{A_R} = \mathbf{I} \cdot \omega_c + [\boldsymbol{\omega}]_\times \sqrt{1 - \omega_c^2} + \vec{\boldsymbol{\omega}} \cdot \vec{\boldsymbol{\omega}}^{\mathrm{T}}\left(1 - \omega_{\mathrm{c}}\right) \qquad \textbf{(2.73)}$$

where ω_c is the dot product given in **(2.67)**.

2.7.7 Composite Elasticity Matrix

As the reinforcing fibres in a composite material typically have much higher stiffness than the surrounding matrix material, the volume fraction of fibres in a composite material is a critical factor in its overall stiffness. The elastic moduli E_x, E_y and E_z as well as the shear moduli G_{xy} G_{yz} and G_{zx} described in Section 2.7.3 are a function of the fibre volume fraction. The calculation of these properties and that of the volume fraction itself is described herein.

The fibre volume fraction at any given point within the part is also represented by the vector field; this will be described in Section 4.1. Specifically, the length of parameter vector $\vec{\mathbf{p}}$ at said point, denoted $|\vec{\mathbf{p}}|$, is scaled by the maximum volume fraction assumed possible for the part being optimized to describe the volume fraction in the volume surrounding that vector:

$$\mathrm{v_f} = \mathrm{v_{f\,max}} \cdot |\vec{\mathbf{p}}| \qquad \textbf{(2.74)}$$

Practical limits on fibre volume fraction in fibre-reinforced composites arising from manufacturing concerns, denoted here as $v_{f\,max}$, typically range from 0.5 to 0.7, depending on the nature of the reinforcement, the degree to which the fibres are aligned with each other and the manufacturing processes selected for the construction of the part [**53**]. The theoretical limit on v_f for aligned hexagonal packed cylindrical fibres is approximately 0.906 but this is never reached in practice.

In the case of a unidirectional fibre-reinforced composite loaded along the fibre axis, the cross-section of fibres bearing a given on-axis stress of magnitude σ_f corresponds to A_f. This area corresponds to the product of the total area A times the fibre volume fraction v_f. Likewise, the cross-section of matrix bearing a given on-axis stress of magnitude σ_m corresponds to the product of the total area A times the volume fraction of the surrounding matrix v_m . Assuming that no voids are present in the composite and given the definition of stress, the magnitude of the on-axis forces exerted by the fibres and matrix respectively correspond to:

$$F_f = \sigma_f\,(A\,v_f) \tag{2.75}$$

$$F_m = \sigma_m\,(A\,v_m) \tag{2.76}$$

Summing these components and dividing by the total cross-sectional area A yields the average stress on the entire cross-section:

$$\sigma_c = \frac{F_f + F_m}{A} = \sigma_f v_f + \sigma_m v_m \tag{2.77}$$

Because the fibres are substantially stiffer than the matrix, they also carry a proportion of the load that is much greater than v_f. Therefore, σ_f is significantly larger than σ_m and σ_c.

Given that both the fibrous component and the matrix component of the fibre-reinforced composite deform equally when the material is loaded along the fibres, Hooke's law as applied to the fibre cross-section and matrix cross-section leads to:

$$\sigma_f = E_f \varepsilon_x \tag{2.78}$$

$$\sigma_m = E_m \varepsilon_x \tag{2.79}$$

respectively. Substituting these relations in **(2.77)** yields:

$$\sigma_c = (E_f v_f + E_m v_m) \varepsilon_{on\text{-}axis} = E_x \varepsilon_x \tag{2.80}$$

Therefore, the on-axis modulus E_x is obtained through the rule of mixture:

$$E_x = E_f v_f + E_m v_m \tag{2.81}$$

As the total cross-sectional area is the sum of the fibre and matrix areas, their volume fractions add to 1. Therefore, equation **(2.81)** simplifies to:

$$E_x = E_f v_f + E_m(1 - v_f) \tag{2.82}$$

While equation **(2.82)** is a simple and accurate representation of the on-axis Young's modulus for a unidirectional fibre-reinforced composite without voids, it cannot be applied to off-axis or transversally loaded fibre-reinforced composites. While the simpler approach to modeling the transverse modulus is to invert the rule of mixtures applied to compliances as done for springs in series, a more accurate model of the transverse modulus is given by Tsai and Hahn [**54**] with empirical corrections for the micromechanical structure:

$$E_y = \left(\frac{vf}{E_f} + \frac{(1-v_f)}{E_m} - v_f \cdot (1-v_f) \frac{\nu_f^2 \cdot \left(\frac{E_m}{E_f} \right) + \nu_r^2 \cdot \left(\frac{E_f}{E_m} \right) - v_f \cdot \nu_r}{E_f v_f + E_m (1 - v_f)} \right)^{-1} \quad \textbf{(2.83)}$$

where ν_f corresponds to the Poisson's ratio of the fibres and ν_m corresponds to the Poisson's ratio of the matrix. For an orthotropic material, the major Poisson's ratio corresponds to the ratio of strain in the on-axis axial direction ε_x to the strain in the on-axis transverse direction, ε_y or ε_z :

$$\nu_{xu} = - \varepsilon_x/\varepsilon_u \quad \textbf{(2.84)}$$

This property can also be calculated using the rule of mixtures:

$$\nu_{xu} = \nu_f \, v_f + \nu_m \, (1 - v_f) \quad \textbf{(2.85)}$$

The minor Poisson's ratios correspond to the ratio of strain in an on-axis transverse direction, ε_y or ε_z, to the strain in the on-axis axial direction:

$$\nu_{ux} = - \varepsilon_u/\varepsilon_x \quad \textbf{(2.86)}$$

This property is a function of the major Poisson's ratio and the on-axis longitudinal and transverse moduli:

$$\nu_{ux} = (E_y / E_x) \ , \ \nu_{zx} = (E_z / E_x) \tag{2.87}$$

Combining the properties calculated above with the general orthotropic stiffness matrix derived in Section 2.7.3, and applying the coordinate transform and the rotation matrix derived in Section 2.7.4 yields the expression for an off-axis elasticity matrix of the composite material. This is then used in conjunction with matrix **B** expressed by equation **(2.33)** for calculating the element stiffness matrix as outlined in the next section.

2.8. Element Stiffness Matrix

The element stiffness matrix relates the displacements observed at the element's nodes to the forces acting thereupon. This matrix can be derived from the minimum potential energy principle.

2.8.1 Minimum Potential Energy Principle

The minimum potential energy principle states that a body will deform to a state which minimizes the total potential energy contained therein. This state represents the equilibrium condition for the structure. Concisely, this principle can be expressed at equilibrium using the following equation [**55**]:

$$\delta\Pi = \delta(U_i + U_e) = 0 \tag{2.88}$$

where Π is the total potential energy which is the sum of the elastic strain energy stored in the body, also known as the internal potential energy and labelled U_i, and the potential energy of the applied forces, also known as the external potential energy and labelled U_e.

In conservative systems, when no energy dissipation occurs, the potential energy stored in the system upon loading is equal to the work W_e exerted on the system by the external forces. Therefore, at equilibrium and with $-U_e = W_e$, equation (2.88) can be rewritten as:

$$\delta\Pi = \delta(U_i - W_e) = 0 \tag{2.89}$$

Because Π is a function of both the strain function and the displacement function, it is known as a functional. It follows that at equilibrium, equation **(2.89)** is equivalent to:

$$\delta U_i = \delta W_e \tag{2.90}$$

2.8.2 Element Stiffness Matrix Derivation

The right-hand side of equation **(2.90)** can be expanded to consider body forces acting over element volume Ω_e and identified by the vector function $\vec{\mathbf{b}}$, surface tractions acting over element surface S_e and identified by the vector function $\vec{\mathbf{s}}$, and point loads identified by the set of vectors $\vec{\mathbf{f}}_{\mathrm{p}} = \left[f_x, f_y, f_z, m_x, m_y, m_z\right]^{\mathrm{T}}$ [**28**]:

$$\delta W_e = \int_{\Omega_e} \delta \vec{\mathbf{u}}^{\mathrm{T}} \, \vec{\mathbf{b}} \, \mathrm{d}\Omega_e + \int_{S_e} \delta \vec{\mathbf{u}}^{\mathrm{T}} \, \vec{\mathbf{s}} \, \mathrm{d}S_e + \sum_{p=1}^{n} \delta \vec{\mathbf{u}}_e^{\mathrm{T}} \, \vec{\mathbf{f}}_p \qquad \textbf{(2.91)}$$

Recalling that for the reduced generalized node element the nodal displacement vector's components $\vec{\mathbf{u}}_{\mathbf{e}\,i} = \begin{bmatrix} u & v & w & \theta_{ix} & \theta_{iy} & \theta_{iz} \end{bmatrix}^T$ correspond to displacements along the axes x, y and z and to rotations about these axes. Likewise, from the definition of strain energy, the left-hand side of equation **(2.90)** can be expanded into [**28**]:

$$\delta U_i = \int_{\Omega_e} \delta \vec{\boldsymbol{\varepsilon}}^{\mathrm{T}} \, \vec{\boldsymbol{\sigma}} \, \mathrm{d}\Omega_e \qquad \textbf{(2.92)}$$

Then, applying the minimum potential energy principle to a tetrahedral element, setting $n = 4$ for the 4 nodes of the tetrahedron and neglecting body forces and surface tractions from the analysis, equation **(2.90)** becomes:

$$\delta U_i = \int_{\Omega_e} \delta \vec{\boldsymbol{\varepsilon}}^{\mathrm{T}} \; \vec{\boldsymbol{\sigma}} \; \mathrm{d}\Omega_e = \sum_{\mathrm{p}=1}^{4} \delta \vec{\mathbf{u}}_{\mathrm{p}}^{\mathrm{T}} \; \vec{\mathbf{f}}_{\mathrm{p}} = \delta W_e \qquad \textbf{(2.93)}$$

Using $\vec{\mathbf{f}}_{\mathrm{e}} = \left[\, \vec{\mathbf{f}}_1 \,\middle|\, \vec{\mathbf{f}}_2 \,\middle|\, \vec{\mathbf{f}}_3 \,\middle|\, \vec{\mathbf{f}}_4 \,\right]^{\mathrm{T}}$ and $\vec{\mathbf{u}}_{\mathrm{e}} = \left[\, \vec{\mathbf{u}}_1 \,\middle|\, \vec{\mathbf{u}}_2 \,\middle|\, \vec{\mathbf{u}}_3 \,\middle|\, \vec{\mathbf{u}}_4 \,\right]^{\mathrm{T}}$ as the column vectors describing the forces and displacements at each of the nodes respectively, this equation can be rewritten as:

$$\int_{\Omega_e} \delta \vec{\boldsymbol{\varepsilon}}^{\mathrm{T}} \; \vec{\boldsymbol{\sigma}} \; \mathrm{d}\Omega_{\mathrm{e}} = \delta \vec{\mathbf{u}}_{\mathrm{e}}^{\mathrm{T}} \; \vec{\mathbf{f}}_{\mathrm{e}} \qquad \textbf{(2.94)}$$

Substituting equation **(2.36)** expressing Hooke's law in 3D and equation **(2.32)** describing the definition of engineering strain as well as its transpose yields:

$$\int_{\Omega_e} \delta \vec{\mathbf{u}}_{\mathrm{e}}^{\mathrm{T}} \mathbf{B}^{\mathrm{T}} \mathbf{E} \, \mathbf{B} \, \vec{\mathbf{u}}_{\mathrm{e}} \, \mathrm{d}\Omega_e = \delta \vec{\mathbf{u}}_{\mathrm{e}}^{\mathrm{T}} \; \vec{\mathbf{f}}_{\mathrm{e}} \qquad \textbf{(2.95)}$$

As $\delta \vec{\boldsymbol{u}}_{\boldsymbol{e}}^{T}$ is independent of the infinitesimal volume $\mathrm{d}\Omega_{\mathrm{e}}$ it can be eliminated from both sides of the equation. This yields the expression for the element stiffness matrix $\mathbf{K}_{\mathrm{e}}$:

$$\vec{\mathbf{f}}_{\mathrm{e}} = \int_{\Omega_e} \mathbf{B}^{\mathrm{T}} \mathbf{E} \; \mathbf{B} \, \mathrm{d}\Omega_e \, \vec{\mathbf{u}}_{\mathrm{e}} = \mathbf{K}_{\mathrm{e}} \; \vec{\mathbf{u}}_{\mathrm{e}} \qquad \textbf{(2.96)}$$

and therefore:

$$\mathbf{K}_\mathrm{e} = \int_{\Omega_e} \mathbf{B}^\mathrm{T} \mathbf{E} \ \mathbf{B} \, \mathrm{d}\Omega_e \qquad \textbf{(2.97)}$$

Substituting equation **(2.34)** into this equation, matrices $\mathbf{B}$ and $\mathbf{B}^\mathrm{T}$ in the integrand can be expressed as a sum of products over tetrahedral coordinates ζ_1, ζ_2, ζ_3 and ζ_4:

$$\mathbf{K}_\mathrm{e} = \sum_{i=1}^{4} \sum_{j=1}^{4} \int_{\Omega_e} \zeta_i \left[\mathbf{b}_{i,1} \middle| \mathbf{b}_{i,2} \middle| \mathbf{b}_{i,3} \middle| \mathbf{b}_{i,4} \right]^\mathrm{T} \mathbf{E} \ \zeta_j \left[\mathbf{b}_{j,1} \middle| \mathbf{b}_{j,2} \middle| \mathbf{b}_{j,3} \middle| \mathbf{b}_{j,4} \right] \mathrm{d}\Omega_e \qquad \textbf{(2.98)}$$

$\mathbf{E}$ may either be taken as a constant over the element volume, in which case taking the constants out of the integral yields:

$$\mathbf{K}_\mathrm{e} = \sum_{i=1}^{4} \sum_{j=1}^{4} \left[\mathbf{b}_{i,1} \middle| \mathbf{b}_{i,2} \middle| \mathbf{b}_{i,3} \middle| \mathbf{b}_{i,4} \right]^\mathrm{T} \mathbf{E} \left[\mathbf{b}_{j,1} \middle| \mathbf{b}_{j,2} \middle| \mathbf{b}_{j,3} \middle| \mathbf{b}_{j,4} \right] \int_{\Omega_e} \zeta_i \ \zeta_j \ \mathrm{d}\Omega_e \qquad \textbf{(2.99)}$$

Otherwise, taking $\mathbf{E}$ to be linear over the element and expressing it in terms of tetrahedral coordinates yields:

$$\mathbf{K}_\mathrm{e} = \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} \left[\mathbf{b}_{i,1} \middle| \mathbf{b}_{i,2} \middle| \mathbf{b}_{i,3} \middle| \mathbf{b}_{i,4} \right]^\mathrm{T} \mathbf{E}_k \left[\mathbf{b}_{j,1} \middle| \mathbf{b}_{j,2} \middle| \mathbf{b}_{j,3} \middle| \mathbf{b}_{j,4} \right] \int_{\Omega_e} \zeta_i \ \zeta_j \ \zeta_k \ \mathrm{d}\Omega_e \qquad \textbf{(2.100)}$$

2.8.3 Analytical Integration

Because the integrands in **(2.99)** and **(2.100)** are both linear polynomials, integration over the element's volume in tetrahedral coordinates can be done analytically with the following equation [**29**]:

$$\int_{\Omega_e} \zeta_1^i \zeta_2^j \zeta_3^k \zeta_4^l \, \mathrm{d}\Omega_e = \frac{i!\,j!\,k!\,l!}{(i+j+k+l+3)!} 6\,\Omega_e \qquad \textbf{(2.101)}$$

where ζ_i for $i = 1, 2, 3, 4$ are the tetrahedral coordinates in the integrand. Factorial products p, q, r and s vary depending on i, j and k in **(2.99)** and **(2.100)**.

2.9. Discretization and Assembled Stiffness Matrix

2.9.1 Mesh Discretization

Tetrahedral elements were chosen for partitioning the object. Since tetrahedral elements are simplexes they can be used in modeling any concave or convex shape efficiently. The quality of a tetrahedral mesh is function of the aspect ratios of the tetrahedra which make it up, as well as of the resolution of the mesh. The aspect ratio of a shape is a measure of the ratio of the maximum to minimum lengths, and it must be kept as small as possible to reduce inaccuracies in the FEA calculation. Likewise, by reducing the maximum tetrahedron volume permitted in the discretized mesh, a more precise model of the part's behaviour may be obtained at the cost of computational complexity.

The Delaunay tetrahedralization algorithm is a common method used for segmenting a volume into tetrahedral elements whilst ensuring that these elements have appropriate sizes and aspect ratios. An implementation of this algorithm is provided in the Tetgen library, which was relied upon for generating the meshes used in this work. Figure 2-7 shows an example volume defined by a set of polyhedral facets, along with Tetgen's tetrahedral output mesh. A thorough description of the Delaunay tetrahedralization algorithm and its implementation in Tetgen is given in [**33**].

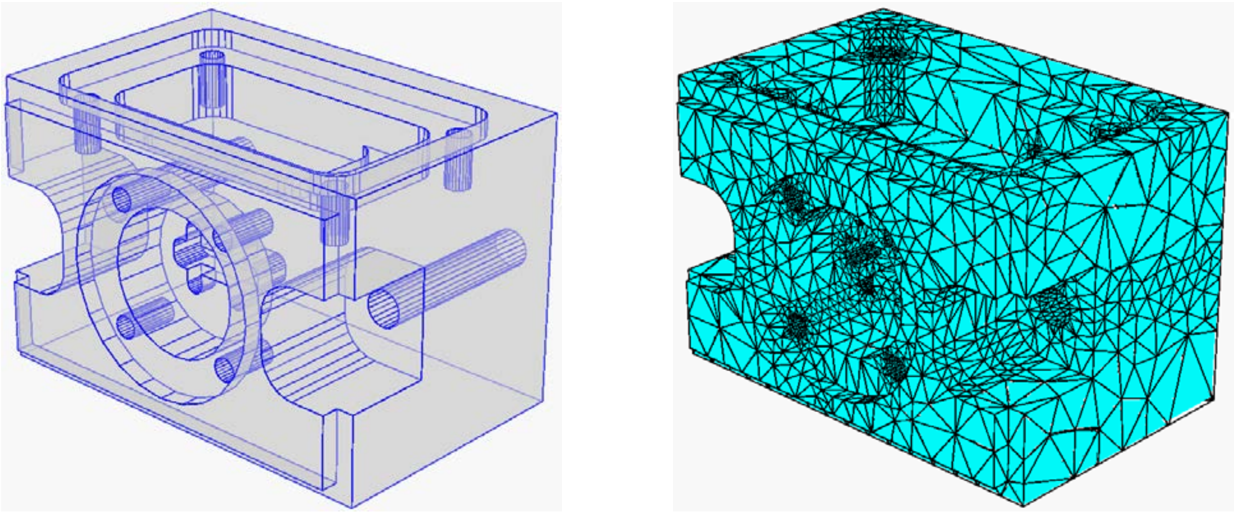


Figure 2-7: Polyhedral domain and tetrahedral mesh thereof [33**]**

2.9.2 Physical Meaning and Description

Where the element stiffness matrix relates the forces present at the nodes of an elastic element to the displacements imposed upon them, the global assembled stiffness matrix does the same over the entire discretized mesh. The assembled stiffness matrix, labelled $\mathbf{K}_a$, is a $(p \times n - c) \times (p \times n - c)$ square matrix, where p, n and c represent the number of DoFs per element, the number of elements and the number of DoFs constrained, respectively.

As each given node may be shared by several elements, the forces generated at these nodes as a result of a displacement add up by superposition. As a result, entries in the global stiffness matrix can receive contributions from several elements. Similarly, when two or more DoFs are linked by constraining their displacements, their entries in the stiffness matrix represent the sum of the stiffness coupling contributions to each DoF. Given a row/column index representing this linked set, element contributions to each of these DoFs are therefore redirected to a common, linked DoF index. Finally, as certain DoFs can be eliminated by a fixed constraint, the rows and columns corresponding to these DoFs are also omitted from the matrix. Element contributions to these rows and columns are removed from the matrix.

2.9.3 Calculation

2.9.3.1 Node Indices

The first step in calculating the entries of the global assembled stiffness matrix is to convert the local element indices which range from 1 to 4 for 4 node tetrahedra, to global indices, which range from 1 to the total number of nodes in the mesh. As global node indices can be stored easily with the node data during mesh discretization this is a straightforward procedure.

2.9.3.2 Degree of Freedom Indices

Likewise, the local indices corresponding to each DoF of an element must be mapped to their global equivalents. For instance, the RGN tetrahedron has 24 local DoFs corresponding to 4 nodes with 6 DoFs per node. On the other hand, a mesh consisting of a collection of such tetrahedra would have a number of DoFs equivalent to 24 times the number of nodes therein.

Therefore a map between DoF indices numbered 1 to 24 locally must point to its corresponding global integer.

The final DoF map must also account for boundary conditions imposed on certain DoFs. As a consequence, this map is not a straightforward equivalence as the nodal index map was. As described in the previous section, every local DoF index may either point to a unique global DoF index, to a shared global index corresponding to linked DoFs or to a marker indicating that the DoF is fixed and therefore not present in the matrix. Such a map can be constructed as a C++ array which is first initialized with an ascending set of integers, then modified and resized as needed once constraints are put in place.

2.9.3.3 Element Contributions

Stiffness contributions from every element in the mesh must be accounted for in the assembly matrix. Iterating over these elements, every element stiffness matrix has a set of local DoFs which in turn correspond to the row and column indices in the assembled stiffness matrix. As described in the previous section, a map between the local DoF indices which point to rows and columns in the element stiffness matrices, and the corresponding global DoF indices which point to rows and columns in the assembled stiffness matrix, is first constructed. Given these equivalences and starting from an assembly matrix initialized to zero, the construction of the global stiffness matrix consists simply in compounding, or summing, the matrix entries from every element stiffness matrix to the appropriate entry in the global assembly matrix.

2.10. Solving for the Displacement Vector

Given the linear relation between displacements and forces expressed through the assembled stiffness matrix, one can solve for the displacements by solving the equation for $\vec{\mathbf{u}}_{\mathrm{a}}$:

$$\vec{\mathbf{u}}_{\mathrm{a}} = \mathbf{K}_{\mathrm{a}}^{-1} \vec{\mathbf{f}}_{\mathrm{a}} \tag{2.102}$$

where $\vec{\mathbf{f}}_{\mathrm{a}}$ is the assembled force vector which describes the forces acting along each of the independent DoFs, $\mathbf{K}_{\mathrm{a}}$ is the assembled stiffness matrix and $\vec{\mathbf{u}}_{\mathrm{a}}$ is the assembled displacement vector which describes the displacements corresponding to each of the independent DoFs.

This can be accomplished in a straightforward manner by multiplying both sides of the equation by the inverse of the assembled stiffness matrix $\mathbf{K}_{\mathrm{a}}^{-1}$. As the stiffness matrix is usually sparse, it is far more efficient to compute its LU factorization and solve for the displacement vector therewith.

2.10.1 LU Factorization

The LU factorization of a matrix aims at expressing this matrix as a product of a lower triangular matrix and an upper triangular matrix [**56**]:

$$\mathbf{A} = \mathbf{L}\,\mathbf{U} \tag{2.103}$$

where **A** is the (m x n) matrix being factorized, **L** is a (m x m) lower triangular matrix and **U** is a (m x n) upper triangular matrix.

Using relation **(2.103)** it is possible to solve the linear equation quickly:

$$\mathbf{A}\,\vec{\mathbf{x}} = \vec{\mathbf{b}} \tag{2.104}$$

using the pair of equations:

$$\mathbf{U}\,\vec{\mathbf{x}} = \vec{\mathbf{y}} \tag{2.105}$$

$$\mathbf{L}\,\vec{\mathbf{y}} = \vec{\mathbf{b}} \tag{2.106}$$

Not only does this method lend itself very well to the solution of sparse matrix equations, but once **L** and **U** have been computed any number of equations involving the same matrix **A** can be solved using the equations described in **(2.105)** and **(2.106)**.

The Eigen linear algebra library includes a sparse matrix solver which was used for computing the LU factorization of $\mathbf{K}_a$ [**57**]. In turn, this solver calls a user-specified implementation of the basic linear algebra subprograms (BLAS) routines, which perform the basic linear algebra operations rapidly at hardware level. Two BLAS solvers were tested; first the GotoBLAS library [**58**], and secondly AMD’s Accelerated Parallel Processing Math Libraries [**59**]. While the former detects the hardware architecture and effectively tailors the library given the host central processing unit (CPU) architecture, the latter is a direct implementation for AMD devices and makes use of multithreading and graphical processing unit (GPU) resources when available. Ultimately, AMD’s proprietary libraries performed better and were selected towards the final build of the software program.

2.11. Summary of Calculation Steps

For convenience the following section summarizes the algorithm and equations implemented in the scope of this thesis, towards calculating the deformation of a fibre-reinforced composite part in response to a given load.

2.11.1 Reduced Generalized Node Element Stiffness Matrix Calculation

The following subsections describe the calculation of the element stiffness matrix $\mathbf{K}_e$ for reduced generalized node tetrahedra.

2.11.1.1 Element Stiffness Integrand Calculation

The (6 x 24) matrix **B** relates the strain at a point described by the tetrahedral coordinates ζ_1, ζ_2, ζ_3 and ζ_4 to the nodal displacement vector $\vec{\mathbf{u}}_{\mathbf{e}}$. This matrix $\mathbf{B}_k$ must be evaluated at each vertex of the tetrahedron ($k = 1, 2, 3$ and 4) where $\zeta_i = 1$ if $i = k$, and $\zeta_i = 0$ otherwise.

$$\mathbf{B} = \sum_{i=1}^{4} \zeta_i \left[\mathbf{b}_{i,\,1} \,\middle|\, \mathbf{b}_{i,\,2} \,\middle|\, \mathbf{b}_{i,\,3} \,\middle|\, \mathbf{b}_{i,\,4} \right] \tag{2.33}$$

Each matrix block $\mathbf{b}_{i,j}$ can be calculated directly as a function of the linearized rotation matrix about node *i*, $\mathbf{F}_j^{\boldsymbol{\theta}}(\vec{\mathbf{x}}_i)$, and the components of the inverse of the Jacobian matrix **Q**:

$$\mathbf{b}_{i,j} = \frac{1}{6\Omega}\begin{bmatrix} a_i & 0 & 0 \\ 0 & b_i & 0 \\ 0 & 0 & c_i \\ b_i & a_i & 0 \\ 0 & c_i & b_i \\ c_i & 0 & a_i \end{bmatrix}\left[\mathbf{I}\,\middle|\,\mathbf{F}_i^{\boldsymbol{\theta}}(\vec{\mathbf{x}})\right] \tag{2.34}$$

The components a_i , b_i and c_i can be taken directly from **Q,** which is evaluated by taking the inverse of the Jacobian matrix:

$$\mathbf{Q} = \frac{1}{6\Omega}\begin{bmatrix} 6V_{01} & a_1 & b_1 & c_1 \\ 6V_{02} & a_2 & b_2 & c_2 \\ 6V_{03} & a_3 & b_3 & c_3 \\ 6V_{04} & a_4 & b_4 & c_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix}^{-1} = \mathbf{J}^{-1} \tag{2.9}$$

$\mathbf{F}_j^{\boldsymbol{\theta}}(\vec{\mathbf{x}}_i)$ is a (3 X 3) anti-symmetric matrix which describes the deformation of the elastic continuum of the element as a result of nodal rotations. It is a function of the coordinate vectors which correspond to nodes *i* and *j*, $\vec{\mathbf{x}}_i$ and $\vec{\mathbf{x}}_j$, respectively.

$$\mathbf{F}_j^{\boldsymbol{\theta}}(\vec{\mathbf{x}}_i) = \begin{bmatrix} 0 & (z_i - z_j) & -(y_i - y_j) \\ -(z_i - z_j) & 0 & (x_i - x_j) \\ (y_i - y_j) & -(x_i - x_j) & 0 \end{bmatrix} \tag{2.21}$$

The elastic behaviour of each element is described by a (6 x 6) elasticity matrix $\mathbf{E_e}$ which relates the strains present in an elastic continuum to the stresses acting upon it. This matrix depends on the type of material being modeled. Details involved its calculation appear in Section 2.7.

Although the optimizations performed assume that the elasticity matrix is uniform throughout each element with discontinuities at the interfaces separating the elements, the software developed also allows assigning a unique elasticity matrix to each pairing of local node and tetrahedron, thereby interpolating the stiffness matrix between these nodes.

2.11.1.2 Element Stiffness Integration

The (24 x 24) RGN element stiffness matrix $\mathbf{K}_e$ relates the displacements at the nodes of a tetrahedron to the forces acting upon them. It is calculated by taking the integral of $\mathbf{B}^T\ \mathbf{E}\ \mathbf{B}$ over the element volume. Since the integrand can be expressed as a polynomial in terms of tetrahedral coordinates, this can be accomplished symbolically using the following equations.

If $\mathbf{E}$ is constant over the volume of the tetrahedron:

$$\mathbf{K}_e = \sum_{i=1}^{4} \sum_{j=1}^{4} \left[\mathbf{b}_{i,1} \middle| \mathbf{b}_{i,2} \middle| \mathbf{b}_{i,3} \middle| \mathbf{b}_{i,4} \right]^T \mathbf{E} \left[\mathbf{b}_{j,1} \middle| \mathbf{b}_{j,2} \middle| \mathbf{b}_{j,3} \middle| \mathbf{b}_{j,4} \right] \int_{\Omega_e} \zeta_i\ \zeta_j\ \mathrm{d}\Omega_e \qquad \textbf{(2.99)}$$

Alternatively, if $\mathbf{E}$ is represented as a linear interpolation between nodes $k = 1, 2, 3, 4$:

$$\mathbf{K}_\mathrm{e} = \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} \left[\mathbf{b}_{i,\,1} \middle| \mathbf{b}_{i,\,2} \middle| \mathbf{b}_{i,\,3} \middle| \mathbf{b}_{i,\,4}\right]^\mathrm{T} \mathbf{E}_k \left[\mathbf{b}_{j,\,1} \middle| \mathbf{b}_{j,\,2} \middle| \mathbf{b}_{j,\,3} \middle| \mathbf{b}_{j,\,4}\right] \int_{\Omega_e} \zeta_i \, \zeta_j \, \zeta_k \, \mathrm{d}\Omega_e \tag{2.100}$$

The integral can be evaluated directly using the following equation:

$$\int_{\Omega_e} \zeta_1^i \, \zeta_2^j \, \zeta_3^k \, \zeta_4^l \, \mathrm{d}\Omega_e = \frac{i!\,j!\,k!\,l!}{(i+j+k+l+3)!} 6\,\Omega_e \tag{2.101}$$

with exponents p, q, r and s depending on the selection between equations **(2.99)** and **(2.100)** and on the values of i, j and k therein.

2.12. Gradient Calculation

2.12.1 Efficient Optimization of Large Parameter Spaces

Thus far, the FEA model described in this section can be used for predicting many aspects of the mechanical behaviour of composite parts of complex geometry. The mechanical behaviour of such parts is represented by material properties such as the elasticity matrix, which in turn depends on material parameters such as the fibre volume fractions and fibre orientations throughout the part. A more detailed description of the parameters being optimized in this work, their representation and the optimization process can be found in Section 4.1. At this point, suffices to say that a set of material parameters is defined at each element or node in order to characterize the state of the material constituting it, and that the resultant part behaviour is then quantitatively evaluated as a basis of the function to be optimized.

In test cases performed in this work, optimizations involved mesh geometries which could easily exceed hundreds of elements and consequently, hundreds of local material parameters. A variety of nonlinear optimization algorithms were tested which aimed at minimizing the deformation of parts at selected points in response to given loading conditions. These algorithms and their performance are described thoroughly in Section 4.2. Many of the optimization algorithms which were examined make use of the gradient of the optimization metric with respect to the set of optimization variables. As a general rule, although calculating this gradient is computationally demanding, the gradient-based optimization algorithms performed much better given this additional information than their derivative-free counterparts because of the number of parameters involved in the optimizations performed.

Calculating the gradient vector involves successive application of the chain and product rules for derivatives. Although this can be accomplished readily using automated tools such as ADOL-C [**60**], the derivations were calculated and programmed manually. To this end, Section 2.12 lists the successive derivation steps involved in the derivation of the gradient of the optimization function with respect to each component of a composite material property vector $\vec{\mathbf{p}}_{\mathrm{e}}$, initially described in Section 2.7.7.

2.12.2 Derivative of the Assembled Stiffness Matrix

The selected metric $f_{opt}(\vec{\mathbf{u}}_{\mathrm{a}})$ is taken to be a weighted sum of the coefficients of the part's deformation vector $\vec{\mathbf{u}}_{\mathrm{a}}$:

$$f_{opt} = \vec{\mathbf{w}} \cdot \vec{\mathbf{u}}_{\mathrm{a}} \tag{2.107}$$

Because the weighing row vector $\vec{w}$ does not vary with p_i, the derivative of f_{opt} depends solely on the derivative of $\vec{\mathbf{u}}_\mathrm{a}$ with respect to one of the material parameters p_i:

$$\frac{\partial f_{opt}}{\partial p_i} = \vec{\mathbf{w}} \cdot \frac{\partial \vec{\mathbf{u}}_\mathrm{a}}{\partial p_i} \qquad \mathbf{(2.108)}$$

Since $\vec{\mathbf{u}}_\mathrm{a}$ is a function of the inverse of the assembled stiffness matrix $\mathbf{K}_\mathrm{a}^{-1}$ and of the load vector $\vec{\mathbf{f}}_\mathrm{a}$, and as the load vector is constant with respect to $\mathrm{p_i}$, the derivative of the displacement vector $\frac{\partial \vec{\mathbf{u}}_\mathrm{a}}{\partial p_i}$ is a function of the derivative of the inverse of the stiffness matrix:

$$\vec{\mathbf{u}}_\mathrm{a} = \mathbf{K}_\mathrm{a}^{-1} \vec{\mathbf{f}}_\mathrm{a} \qquad \mathbf{(2.102)}$$

$$\frac{\partial \vec{\mathbf{u}}_\mathrm{a}}{\partial p_i} = \frac{\partial \mathbf{K}_\mathrm{a}^{-1}}{\partial p_i} \vec{\mathbf{f}}_\mathrm{a} \qquad \mathbf{(2.109)}$$

$\frac{\partial \mathbf{K}_\mathrm{a}^{-1}}{\partial p_i}$ can then be solved as follows. Starting from the definition of the matrix inverse, and taking its derivative:

$$\mathbf{K}_\mathrm{a}^{-1}\mathbf{K}_\mathrm{a} = \mathbf{I} \qquad \mathbf{(2.110)}$$

$$\frac{\partial\left(\mathbf{K}_\mathrm{a}^{-1}\mathbf{K}_\mathrm{a}\right)}{\partial p_i} = \frac{\partial \mathbf{I}}{\partial p_i} = \mathbf{0} \qquad \mathbf{(2.111)}$$

Expanding the left-hand side:

$$\frac{\partial\left(\mathbf{K}_{\mathrm{a}}^{-1}\right)}{\partial p_i}\mathbf{K}_{\mathrm{a}} + \mathbf{K}_{\mathrm{a}}^{-1}\frac{\partial\left(\mathbf{K}_{\mathrm{a}}\right)}{\partial p_i} = \mathbf{0} \tag{2.112}$$

Then, subtracting $\mathbf{K}_{\mathrm{a}}^{-1}\frac{\partial\left(\mathbf{K}_{\mathrm{a}}\right)}{\partial p_i}$ from both sides of the equation and multiplying by $\mathbf{K}_{\mathrm{a}}^{-1}$ yields the expression for the derivative of the matrix inverse:

$$\frac{\partial\left(\mathbf{K}_{\mathrm{a}}^{-1}\right)}{\partial p_i} = \mathbf{K}_{\mathrm{a}}^{-1}\frac{\partial\left(\mathbf{K}_{\mathrm{a}}\right)}{\partial p_i}\mathbf{K}_{\mathrm{a}}^{-1} \tag{2.113}$$

Substituting this result in equation **(2.109)** the derivative of the displacement vector can be expressed as:

$$\frac{\partial \vec{\mathbf{u}}_{\mathrm{a}}}{\partial p_i} = \mathbf{K}_{\mathrm{a}}^{-1}\frac{\partial \mathbf{K}_{\mathrm{a}}}{\partial p_i}\mathbf{K}_{\mathrm{a}}^{-1}\vec{\mathbf{f}}_{\mathrm{a}} \tag{2.114}$$

Recalling equation (2.102) this can be simplified to:

$$\frac{\partial \vec{\mathbf{u}}_{\mathrm{a}}}{\partial p_i} = \mathbf{K}_{\mathrm{a}}^{-1}\frac{\partial \mathbf{K}_{\mathrm{a}}}{\partial p_i}\mathbf{K}_{\mathrm{a}}^{-1}\vec{\mathbf{u}}_{\mathrm{a}} \tag{2.115}$$

Finally, substituting this equation into (2.108):

$$\frac{\partial f_{opt}}{\partial p_i} = \vec{\mathbf{w}} \cdot \mathbf{K}_{\mathrm{a}}^{-1} \frac{\partial \mathbf{K}_{\mathrm{a}}}{\partial p_i} \mathbf{K}_{\mathrm{a}}^{-1} \vec{\mathbf{u}}_{\mathrm{a}} \tag{2.116}$$

$\vec{\mathbf{w}}\,\mathbf{K}_{\mathrm{a}}^{-1}$ and $\vec{\mathbf{u}}_{\mathrm{a}}$ can be stored and reused to calculate the derivative of $\mathrm{f}_{\mathrm{opt}}$ with respect to all optimization variables in the gradient. Specifically, the vector $\vec{\lambda}$ is defined as:

$$\vec{\lambda} = \left(\left(\mathbf{K}_{\mathrm{a}}^{-1} \right)^{T} \vec{\mathbf{w}}^{\mathrm{T}} \right)^{\mathrm{T}} \tag{2.117}$$

and can be solved using the same method as the deformation vector $\vec{\mathbf{u}}_{\mathrm{a}}$ in Section 2.10.1.

Therefore, given the derivative of the assembled stiffness matrix $\mathbf{K}_{\mathrm{a}}$ with respect to each parameter p_i, the gradient can be calculated term by term using the following equation:

$$\frac{\partial \mathrm{f}_{\mathrm{opt}}}{\partial p_i} = \vec{\lambda}\, \frac{\partial \mathbf{K}_{\mathrm{a}}}{\partial p_i} \vec{\mathbf{u}}_{\mathrm{a}} \tag{2.118}$$

for each optimization variable p_i in the assembled parameter vector $\vec{\mathbf{p}}_{\mathrm{a}}$. This method, which accelerates the calculation of the gradient by sidestepping the requirement for the solution of the linear system in (2.116) is called the adjoint method; it is explained in greater detail in [**61**].

As explained in Section 2.9, each coefficient in the assembled stiffness matrix $\mathbf{K}_{\mathrm{a}}$ is calculated by taking the sum of components from contributing element stiffness matrices $\mathbf{K}_{\mathrm{e}}$:

$$K_a[i,j] = \sum_{e_i=1}^{n} K_e\left[h\left(e_i, i\right), h\left(e_i, j\right)\right] \tag{2.119}$$

where $K_a[i,j]$ represents a component of the assembled stiffness matrix at row index i and column index j, e_i represents the index of one of the elements contributing to $\mathbf{K}_a$ at indices i and j, and where $K_e\left[h\left(e_i, i\right), h\left(e_i, j\right)\right]$ corresponds to the contribution of one of the indices of the element stiffness matrix. $h\left(e_i, i\right)$ represents a function which relates the global stiffness matrix i to the corresponding index of the contribution from the element stiffness matrix.

Furthermore, since every optimization variable contributes to describing the local material properties of the part, only one element stiffness matrix is affected by a change in any single optimization variable. Therefore, it follows from equation (2.119) that the derivative of the assembled stiffness matrix with respect to an optimization variable can be calculated much in the same way as described in Section 2.9, but instead of obtaining stiffness contributions from element stiffness matrices for all the elements in the mesh, the derivative of these contributions with respect to an optimized variable is calculated instead, only for the elements whose properties are described by the optimized variable in question.

2.12.3 Derivative of the Element Stiffness Matrix

As described in Section 2.8, the symbolic solution to the element stiffness matrix integral described in equations **(2.99)** and **(2.100)** can be expressed as a sum of products **(2.101)**.

The matrix **B** constructed by the concatenation of matrix blocks $\mathbf{b}_{i,j}$ depends solely on the shape of the element. As such, it is independent of the material properties assigned to the element. Therefore, the derivative of the element stiffness matrix with respect to a material parameter of the element takes the following forms:

If **E** is constant over the volume of the tetrahedron:

$$\frac{\partial \mathbf{K}_{\mathrm{e}}}{\partial p_m} = \sum_{i=1}^{4} \sum_{j=1}^{4} \left[\mathbf{b}_{i,1} \middle| \mathbf{b}_{i,2} \middle| \mathbf{b}_{i,3} \middle| \mathbf{b}_{i,4}\right]^{\mathrm{T}} \frac{\partial \mathbf{E}}{\partial p_m} \left[\mathbf{b}_{j,1} \middle| \mathbf{b}_{j,2} \middle| \mathbf{b}_{j,3} \middle| \mathbf{b}_{j,4}\right] \int_{\Omega_e} \zeta_i \zeta_j \, \mathrm{d}\Omega_e \tag{2.120}$$

Alternatively, if **E** is represented as a linear interpolation between nodes k = 1, 2, 3, 4:

$$\frac{\partial \mathbf{K}_{\mathrm{e}}}{\partial p_m} = \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} \left[\mathbf{b}_{i,1} \middle| \mathbf{b}_{i,2} \middle| \mathbf{b}_{i,3} \middle| \mathbf{b}_{i,4}\right]^{\mathrm{T}} \frac{\partial \mathbf{E}_k}{\partial p_m} \left[\mathbf{b}_{j,1} \middle| \mathbf{b}_{j,2} \middle| \mathbf{b}_{j,3} \middle| \mathbf{b}_{j,4}\right] \int_{\Omega_e} \zeta_i \zeta_j \zeta_k \, \mathrm{d}\Omega_e \tag{2.121}$$

It should be noted that the change in notation between equation **(2.118)** and equations **(2.120)** and **(2.121)** from p_i to p_m was done to avoid confusion between the index i which denotes both the parameter index in the former equation, and a component of the shape matrix **B** in the latter equation.

For a composite material, the derivative of the elasticity matrix **E** with respect to a material parameter p_m depends on the definition of that material parameter. As highlighted in Section

2.7.7, the material properties of the discretized mesh are described by a set of $\mathcal{R}^3$ parameter vectors of which p_m would be a component. Equation **(2.65)** gives the direction of each of these vectors, which represent the mean orientation of the material at that location. As described by equation **(2.74)**, the length of each vector which is constrained between 0 and 1. The length of each vector represents the ratio of the volume fraction of fibres at that location to the maximum volume fraction enabled by the manufacturing process.

Given that the composite materials simulated are orthotropic, the orientation of the material must be accounted for. The off-axis material transform given in **(2.64)** describes the behaviour of an off-axis orthotropic material. As both $\mathbf{A}_\varepsilon$ and **E'** are dependent upon every component of the parameter vector, the derivative of this matrix is given by the product rule:

$$\frac{\partial \mathbf{E}}{\partial p_m} = \left(\frac{\partial \mathbf{A}_\varepsilon^{\mathrm{T}}}{\partial p_m} \mathbf{E}' \mathbf{A}_\varepsilon \right) + \left(\mathbf{A}_\varepsilon^{\mathrm{T}} \frac{\partial \mathbf{E}'}{\partial p_m} \mathbf{A}_\varepsilon \right) + \left(\mathbf{A}_\varepsilon^{\mathrm{T}} \mathbf{E}' \frac{\partial \mathbf{A}_\varepsilon}{\partial p_m} \right) \quad \textbf{(2.122)}$$

The strain transform matrix $\mathbf{A}_\varepsilon$ is a function of the rotation matrix **A**, which in turn describes the material's orientation from the reference axis along the x-direction, as described in equation **(2.53)**. Then, since the components of this matrix are comprised of products of the (3 x 3) transform matrix **A** calculated by equations **(2.66)** to **(2.73)**, the derivative of this matrix can be calculated term by term using the product rule:

$$\begin{aligned}\frac{\partial \mathbf{A}_{\mathrm{R}}}{\partial p_m} &= \mathbf{I}\cdot\frac{\partial \omega_c}{\partial p_m} \\ &+ \frac{\partial[\boldsymbol{\omega}]_{\times}}{\partial p_m}\sqrt{1-\omega_c^2} + [\boldsymbol{\omega}]_{\times}\frac{\partial\sqrt{1-\omega_c^2}}{\partial p_m} \\ &+ \frac{\partial\vec{\boldsymbol{\omega}}}{\partial p_m}\cdot\vec{\boldsymbol{\omega}}^{\mathrm{T}}\left(1-\omega_c\right) + \vec{\boldsymbol{\omega}}\cdot\frac{\partial\vec{\boldsymbol{\omega}}}{\partial p_m}^{\mathrm{T}}\left(1-\omega_c\right) - \vec{\boldsymbol{\omega}}\cdot\vec{\boldsymbol{\omega}}^{\mathrm{T}}\frac{\partial\omega_c}{\partial p_m}\end{aligned} \tag{2.123}$$

Following the chain rule, the derivative of ω_c with respect to matrix parameter $\mathrm{p_m}$ is:

$$\frac{\partial \omega_c}{\partial p_m} = \vec{\mathbf{i}} \bullet \frac{\partial \overrightarrow{\boldsymbol{\lambda}_{\mathbf{p}}}}{\partial p_m} \tag{2.124}$$

As stated earlier, p_m must be one of the three components of the parameter vector $\vec{\mathbf{p}}_{\mathrm{e}}$. Therefore, the derivative of the normalized parameter vector, labeled $\overrightarrow{\boldsymbol{\lambda}_{\mathbf{p}}}$, is dependent upon which coordinate is described by p_m. The three possible cases are described here in matrix form:

$$\begin{bmatrix} \dfrac{\partial \overrightarrow{\boldsymbol{\lambda}_{\mathbf{p}}}}{\partial p_{ex}} \\ \dfrac{\partial \overrightarrow{\boldsymbol{\lambda}_{\mathbf{p}}}}{\partial p_{ey}} \\ \dfrac{\partial \overrightarrow{\boldsymbol{\lambda}_{\mathbf{p}}}}{\partial p_{ez}} \end{bmatrix} = \frac{1}{|\vec{\mathbf{p}}_{\mathrm{e}}|}\mathbf{I} + \frac{1}{|\vec{\mathbf{p}}_{\mathrm{e}}|^3}\vec{\mathbf{p}}_{\mathrm{e}}\,\vec{\mathbf{p}}_{\mathrm{e}}^{\mathrm{T}} \tag{2.125}$$

where **I** represents a (3 X 3) identity matrix and $|\vec{p}_e|$ is the Euclidian norm of the element parameter vector $\vec{p}_e$. Therefore, the derivatives represented by equation **(2.124)** correspond to the first column of the (3 X 3) matrix in **(2.125)**, and can be written explicitly as:

$$\begin{bmatrix} \dfrac{\partial \vec{\lambda}_{\mathbf{p}}}{\partial p_{ex}} \\ \dfrac{\partial \vec{\lambda}_{\mathbf{p}}}{\partial p_{ey}} \\ \dfrac{\partial \vec{\lambda}_{\mathbf{p}}}{\partial p_{ez}} \end{bmatrix} = \begin{bmatrix} -\dfrac{p_{ex}^2}{\left(p_{ex}^2+p_{ey}^2+p_{ez}^2\right)^{3/2}} + \dfrac{1}{\sqrt{p_{ex}^2+p_{ey}^2+p_{ez}^2}} \\ -\dfrac{p_{ey}p_{ex}}{\left(p_{ex}^2+p_{ey}^2+p_{ez}^2\right)^{3/2}} \\ -\dfrac{p_{ez}p_{ex}}{\left(p_{ex}^2+p_{ey}^2+p_{ez}^2\right)^{3/2}} \end{bmatrix} \tag{2.126}$$

The derivative cross product matrix $[\boldsymbol{\omega}]_\times$ is also dependent on the component of the parameter vector being derived. Being that $\vec{\boldsymbol{\omega}}$ is itself a cross product of $\vec{\lambda}_p$ and the unit vector aligned along the x direction, equation (2.70) can be rewritten as:

$$[\boldsymbol{\omega}]_\times = \frac{1}{|\vec{\mathbf{p}}_e|} \begin{bmatrix} 0 & -p_{ey} & -p_{ez} \\ p_{ey} & 0 & 0 \\ p_{ez} & 0 & 0 \end{bmatrix} \tag{2.127}$$

and its derivatives are:

$$\frac{\partial[\boldsymbol{\omega}]_\times}{\partial p_{ex}} = -\frac{p_{ex}}{|\vec{\mathbf{p}}_e|^3} \begin{bmatrix} 0 & -p_{\mathrm{ey}} & -p_{\mathrm{ez}} \\ p_{\mathrm{ey}} & 0 & 0 \\ p_{\mathrm{ez}} & 0 & 0 \end{bmatrix} \tag{2.128}$$

$$\frac{\partial[\boldsymbol{\omega}]_{\times}}{\partial p_{e\,y}} = \frac{1}{\left|\vec{\mathbf{p}}_e\right|}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{p_{e\,y}}{\left|\vec{\mathbf{p}}_e\right|^3}\begin{bmatrix} 0 & -p_{\mathrm{e\,y}} & -p_{\mathrm{e\,z}} \\ p_{\mathrm{e\,y}} & 0 & 0 \\ p_{\mathrm{e\,z}} & 0 & 0 \end{bmatrix} \tag{2.129}$$

$$\frac{\partial[\boldsymbol{\omega}]_{\times}}{\partial p_{e\,z}} = \frac{1}{\left|\vec{\mathbf{p}}_e\right|}\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} - \frac{p_{e\,z}}{\left|\vec{\mathbf{p}}_e\right|^3}\begin{bmatrix} 0 & -p_{\mathrm{e\,y}} & -p_{\mathrm{e\,z}} \\ p_{\mathrm{e\,y}} & 0 & 0 \\ p_{\mathrm{e\,z}} & 0 & 0 \end{bmatrix} \tag{2.130}$$

Substitution of these identities into equation (2.123) yielded the explicit code for calculating the derivative of the rotation matrix.

The final step in the calculation of the elasticity matrix for an off-axis composite material is the calculation of on-axis orthotropic stiffness matrix parameters as a function of the fibre volume fraction. The equations pertaining to these properties are found in Section 2.7.7. Their derivatives with respect to the modulus of the element parameter vector $\vec{\mathbf{p}}_{\mathrm{e}}$ are as follows. From equation (2.81):

$$\frac{\partial E_x}{\partial p_{e\,i}} = \frac{\partial E_x}{\partial v_f}\frac{\partial v_f}{\partial p_{e\,i}} = \left(E_f - E_m\right)\frac{\partial v_f}{\partial p_{e\,i}} \tag{2.131}$$

Likewise, deriving equation (2.83):

$$\frac{\partial\left(E_y\right)}{\partial\left(p_{e,\,i}\right)} = - \frac{\dfrac{1}{E_f} - \dfrac{\left(1-v_f\right)\left(\dfrac{\nu_f^2 E_r}{E_f} + \dfrac{\nu_r^2 E_f}{E_r} - \nu_f \nu_r\right)}{E_f vf + E_f\left(1-v_f\right)} + \dfrac{v_f\left(\dfrac{\nu_f^2 E_r}{E_f} + \dfrac{\nu_r^2 E_f}{E_r} - \nu_f \nu_r\right)}{E_f vf + E_f\left(1-v_f\right)} - \dfrac{v_f\left(1-v_f\right)\left(\dfrac{\nu_f^2 E_r}{E_f} + \dfrac{\nu_r^2 E_f}{E_r} - \nu_f \nu_r\right) E_f}{\left(E_f vf + E_f\left(1-v_f\right)\right)^2}}{\left(\dfrac{v_f}{E_f} + \dfrac{1-vf}{E_r} - \dfrac{v_f\left(1-v_f\right)\left(\dfrac{\nu_f^2 E_r}{E_f} + \dfrac{\nu_r^2 E_f}{E_r} - \nu_f \nu_r\right)}{E_f vf + E_f\left(1-v_f\right)}\right)^2}$$

(2.132)

Chapter 3

Verification and Testing of FEA Model

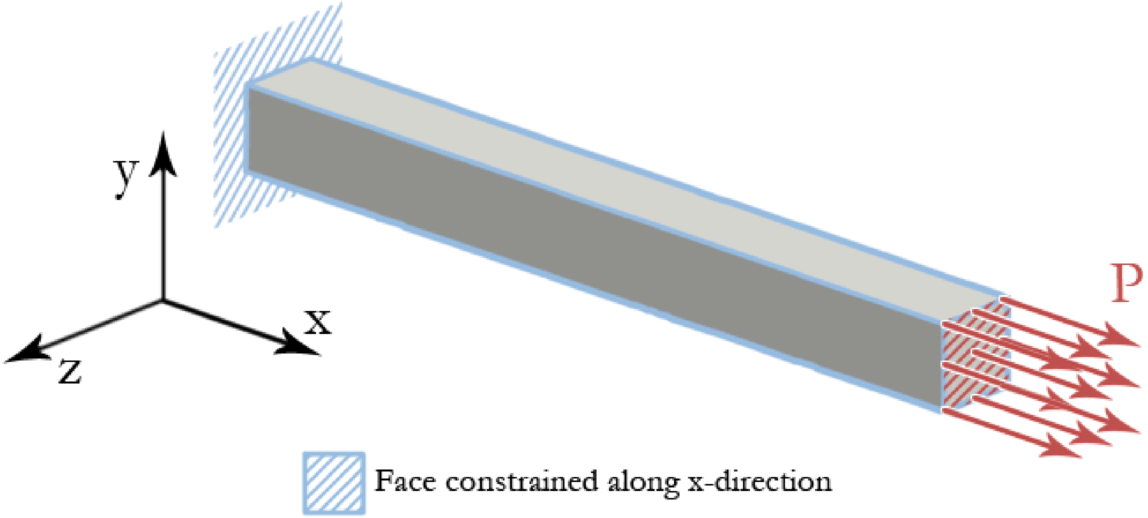
Several equations exist that predict the elastic deformation of simple parts as a function of material properties and loading. The accuracy of the FEA model which was implemented can be verified readily using analytical results obtained using such equations.

3.1.1 Note on Pre/Post-Processing

The following 3 chapters contain renderings of simulation and optimization results, obtained using the software program which was developed towards this thesis. In order to generate and format these renderings, a number of other software programs were used. Appendix A highlights the data flow and implicit methodology used for initializing the software program and for analyzing its output. Appendix B contains examples of the configuration files used in initializing the program. Though the options currently set in the examples given pertain more to Section 5.1.2.3, the options relevant to the cases used in this chapter are visible nonetheless.

3.1.2 Bar Loaded Axially

As illustrated in



, the first test case consisted of a rectangular prism, or bar, subjected to axial tension. A face of this bar, the normal of which pointed along x was constrained along the x-direction only. The opposite face was subjected to a uniformly distributed force along the x-direction. The displacements along the x-direction of the nodes within the loaded face were constrained to be equal. Table 3-1 lists the main dimensions and material properties of the part.

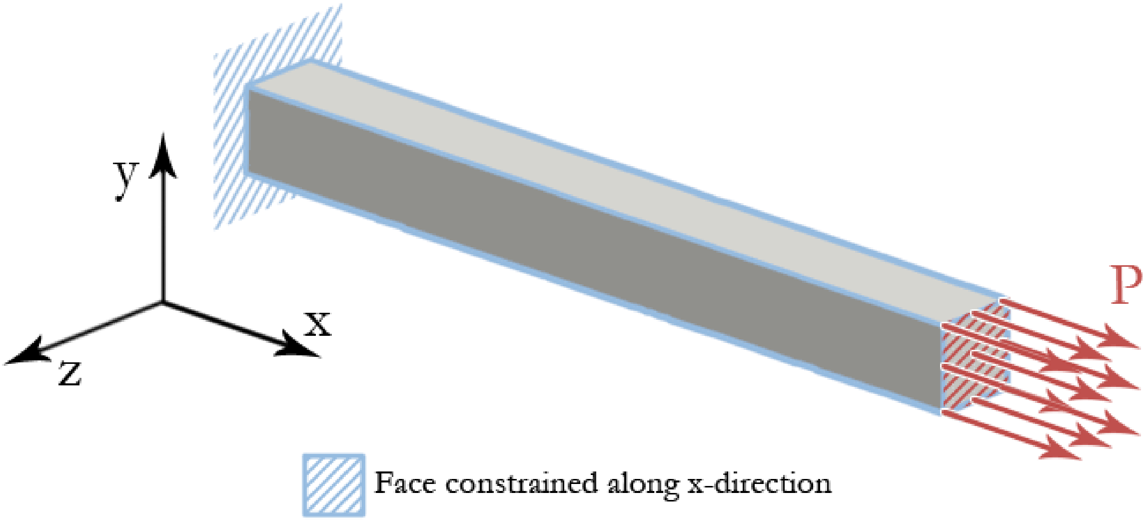


Figure 3-1: Bar geometry with axial loading

Table 3-1: Bar geometry and material properties

Length (L) along x-direction	0.1m
Width (w) along y-direction	0.01m
Thickness (t) along z-direction	0.01m
Modulus of elasticity (isotropic):	50×10^9 Pa

3.1.2.1 Symbolic Solution

From Hooke's law, it is anticipated that the elastic deformation of such a bar will occur along the x-direction with a magnitude of:

$$d_x = (P_x L_x)/(E A_{yz}) \quad \textbf{(3.1)}$$

where d_x is the axial deformation, P_x is the load applied along the x-direction, L_x is the length of the bar, E_x is the elastic modulus of the material and A_{yz} is the cross-sectional area of the specimen normal to the applied load. Setting $P = 1$N with other quantities as stated below yields an expected elastic deformation of:

$$d_x = (1[\mathrm{N}] \times 0.1[\mathrm{m}])/(50 \times 10^9 [\mathrm{N\ m^{-2}}] \times 0.01^2 [\mathrm{m^2}]) = 2.0 \times 10^{-8}[\mathrm{m}] \quad \textbf{(3.2)}$$

Unsurprisingly, since the stresses and strains are constant throughout the part the superlinear RGN tetrahedron also yields this exact result, even at low mesh resolutions as shown in Table 3-2.

Table 3-2: Convergence of FEA solver for bar under simple tension

	Displacement at loaded end [m]	Error [%]
Symbolic solution	2.0×10^{-8}	-
FEA solution, 48 elements	2.0×10^{-8}	0
FEA solution, 381 elements	2.0×10^{-8}	0
FEA solution, 2070 elements	2.0×10^{-8}	0

3.1.3 Cantilevered Beam With End Load

The second test case examined featured the same geometry as the first. In this trial, displacements at one face were constrained along the x-direction and y-direction, and the rotations of the nodes at the same end were locked around the z-direction. The opposite face of the beam was loaded with a 1N distributed force oriented along the y-direction. The displacements of the nodes at the loaded end along the y-direction were constrained to be equal. This loading mode resulted in a deflection along the y-direction caused in part by the shear strain of the beam and also by the bending moment generated. Figure 3-2 shows the geometry of the loaded parts as well as its loading mode.

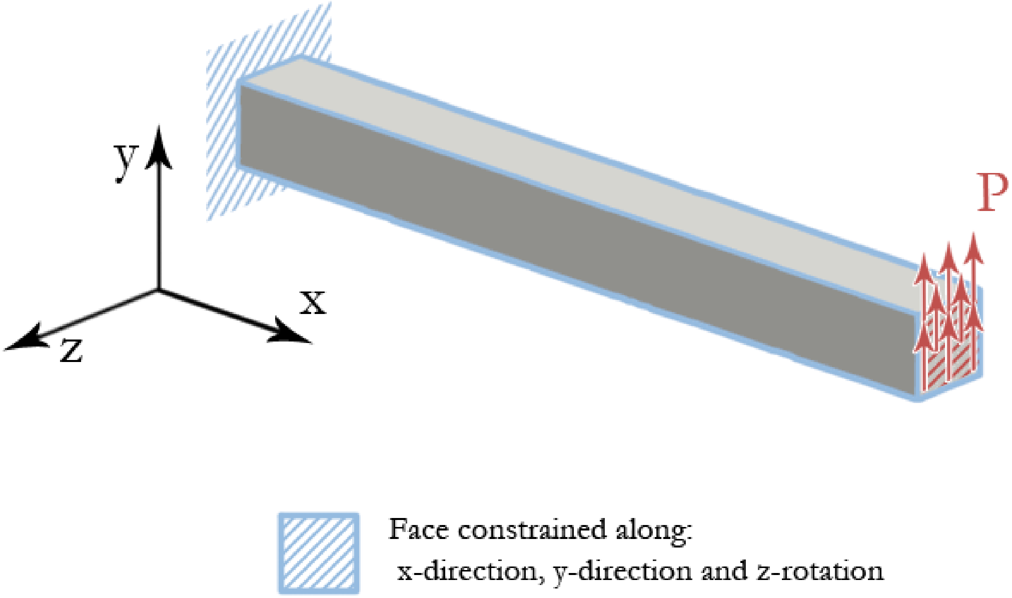


Figure 3-2: Bar geometry with shear loading

3.1.3.1 Symbolic Solution

For a cantilever beam thus loaded, the strain energy method predicts a displacement [**62**]:

$$d_y = \frac{P_y L^3}{3 E_x I_y} \left(1 + 3 \frac{E_x}{10\, G_{xy}} \frac{t^2}{L^2} \right) \qquad \textbf{(3.3)}$$

where d_y is the magnitude of the deflection at the loaded end, P_y is the load applied along the y-direction, E is the elastic modulus of the material and I_y is the second moment of inertia about y, the specimen's cross-section axis normal to the applied load which, for a rectangular section, is equal to:

$$I_y = w \cdot t^3/12 \quad \mathbf{(3.4)}$$

where w is the width of the part along the z-direction and t is the thickness of the part along the y-direction.

Setting $P = 1$ N with other quantities, as stated above yields an expected elastic deformation of 8.0624×10^{-6} metres. Due to the shear stresses and strains generated by this loading mode, a quadratic element such as the GN tetrahedron would be needed for achieving an exact solution. However, given a sufficient number of elements the superlinear RGN tetrahedron converges toward the symbolic solution, as shown in Figure 3-3 shows the deformation calculated by the FEA solver at each resolution, with displacements magnified by a factor of 10000.

Table 3-3: Convergence of FEA solver for beam subjected to end-loading

	Displacement at loaded end [m]	Error [%]
Symbolic solution	8.0624×10^{-6}	-
FEA solution, 48 elements	5.6075×10^{-6}	30.45%
FEA solution, 381 elements	7.0970×10^{-6}	11.97%
FEA solution, 2070 elements	7.6122×10^{-6}	5.58%
FEA solution, 19257 elements	7.9046×10^{-6}	1.96%
FEA solution, 76503 elements	7.9377×10^{-6}	1.55%

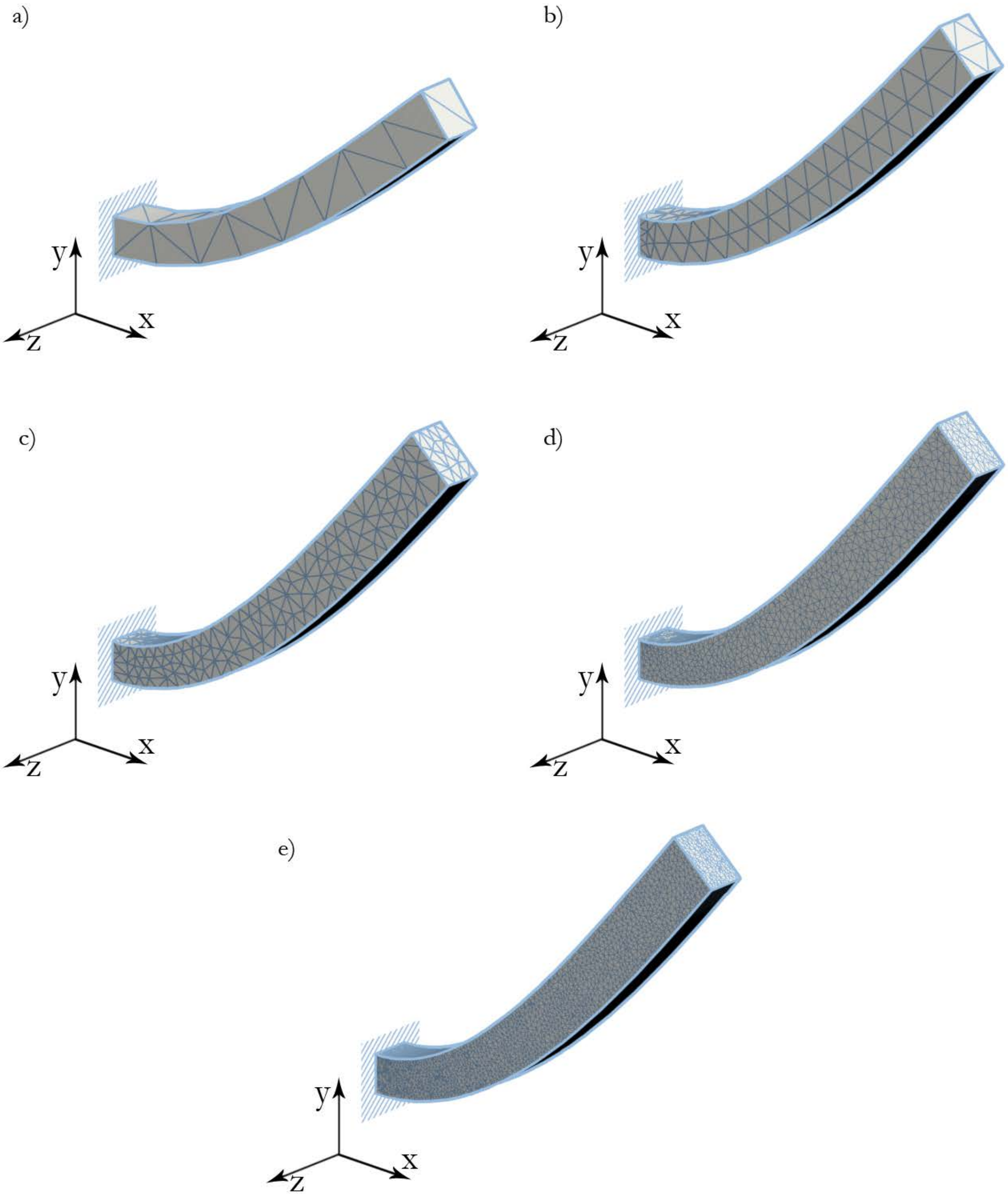


Figure 3-3: Illustration of bending response for various mesh sizes
a) 48 elements, b) 381 elements, c) 2070 elements, d) 19257 elements, e) 76503 elements; displacements magnified by 10000.

Chapter 4

Iterative Optimization

4.1. Approach

4.1.1 Optimization Problem Definition

An optimization problem is defined as follows. Given a set of variables, a set of constraints and an objective function, the goal of optimization is to find either the minimum or maximum value of the objective function. The objective function or metric is a function of the optimization variables, and the variables must lie within the space bound by the constraints.

4.1.2 Selection of Metric and Optimization Variables

Implicitly or explicitly, the definition of an objective metric which measures the value or suitability of a system is a prerequisite to the design of any such system. This metric can be viewed ultimately as a function of the criteria which inform the design process. Therefore, any designer must formulate the objective metric, consider the set of design constraints, and implement an iterative process which compares a set of possible designs.

Given that the selection of design criteria is open-ended by nature a simple objective metric was chosen in this thesis. As a first proof of concept, a weighted sum of nodal displacements along the loaded section of the part was chosen as the sole optimization metric to test the optimization program.

The specific test cases examined centered on the optimization of a fibre-reinforced composite material for a given part geometry and loading. Under these conditions, the optimization metric examined is a function of the displacement vector calculated by the FEA solver. More precisely, the optimization metric consisted of a scalar representing the displacement of the part at the point of loading and along the loading direction. This quantity was chosen as a metric because it is a direct result of the FEA solution, and is easiest to calculate. The program developed for this thesis could be adapted eventually to consider further factors such as failure indices, total part weight and cost, for example.

Depending on the optimization algorithm, the magnitude of the optimization metric may also affect convergence speed and stability, where a larger scaling on the metric induces higher confidence intervals or gradients and consequently increases convergence speed. As the nodal displacements are relatively small given the generally high stiffness of composite materials, a uniform scaling on the optimization metric was applied to improve convergence speed without the need to alter the optimization algorithms examined.

As described in Section 2.7.6, the set of parameters which describe an element's material properties are represented by a single vector, the orientation of which represents the mean orientation of the fibres at that element, and the length of which represents the average fibre volume fraction in the element. The ensemble of these element parameter vectors makes up the parameter space of the optimization problem.

The effect of fibre volume fraction on the elastic behaviour of composite materials is well understood as described in Section 2.7.6. As the metrics being optimized in this work are purely functions of stiffness and are devoid of additional constraints for which fibre volume fraction could be limited, any optimum solution implies that the maximum allowable fibre volume fraction will be achieved throughout the part. Including this parameter in the software program and enabling its optimization still has merits, however. First, its inclusion allows for the eventual consideration of useful criteria such as part weight and the relief of stress concentrations by reducing the modulus and consequent stresses in overloaded sections of a part or by stiffening other sections. In addition, the inclusion of fibre volume fractions into optimization routines as secondary parameters contributes to the verification of the optimization results, provided that the software program converges towards the given maximum fibre volume fraction.

Although the geometry of a part is also a critical parameter of its design and should therefore be included in the optimization process, variations of the geometry are beyond the scope of this thesis. The test cases of the model and optimization algorithm provided herein were conducted using static geometries. Likewise, although the selection of materials as well as their relative prevalence in different parts of a structure is crucial to the design process, this aspect was omitted from the optimization problems examined for the purposes of simplicity. Instead, fibre and matrix material properties as well as loading conditions were inputted directly as constants in the test cases.

The part’s deformation is calculated through the use of the FEA model as a function of the composite’s volume fraction and orientation of its reinforcing fibres, throughout the part. An optimized part design may involve material properties which vary throughout its volume. To that effect, a vector field was defined which describes the varying fibre orientations and volume fractions throughout the part and the consequent elastic response as a function thereof. Two versions of this method were implemented in the software program. The first implementation utilizes a single vector per element while the second implementation defines a material property vector at each node. While the first method assumes constant material properties throughout each element, the second interpolates the elastic response resulting from the fibre descriptions at each element node. Because the first method based on elements requires fewer variables than the second method based on nodes, it was selected for benchmarking and proof of concept. The vector field describing the composite properties in turn dictates the orthotropic elasticity matrices of each element and, as a consequence, the elastic response of the elements they describe.

These optimization variables, also chosen for their relative simplicity, are bounded by a set of constraints. Specifically, an upper bound was set on the fibre volume fraction to reflect the either theoretical limit on fibre packing or the limit of a facility’s manufacturing ability. Although additional constraints would ideally be set on fibre orientations and volume fractions to prevent discontinuities in the model in the case where long continuous fibre-reinforced composites are to be simulated, this step was also omitted in the interest of obtaining general solutions towards objectives imposed on a part’s design. As a consequence, the optimized designs examined in this work are not aimed at continuous fibre composites, as fibre continuity was not accounted for in the optimization process. These considerations are discussed briefly in Section 6.1.3. Similarly, though most part designs should also account for additional constraints such as failure indices, such considerations could be accounted for by the addition of nonlinear constraints using Lagrange multipliers [**63**], but were neglected for the sake of simplicity.

4.1.3 Rigorousness of a Solution to a Non-Convex Problem

It is not uncommon for mathematical functions to have more than one local optimum. Such non-convex problems may have a number of local optima. Therefore, in the context of optimization, arriving at such a local optimum cannot guarantee that this is also the global optimum as there may be a superior solution as of yet undiscovered by the optimization algorithm. Hence a distinction should be made between global optimization algorithms which scour the problem space for a global optimum, and local optimization algorithms which only seek any local optimum. However, by nature the global optimization algorithms tend to take much longer to arrive at a solution than local optimization algorithms.

Although the NLOpt library used for minimizing the displacement function in the software developed provides global optimization algorithms, these algorithms are only practical for problems with a relatively small number of parameters which would in turn be insufficient at describing in detail the fibre orientations throughout a part. Consequently, the focus of the benchmarking and analysis performed in this thesis was to test the limits of the local optimization algorithms with a large number of material design parameters. Given that even the local optimization algorithms are very time consuming with the number of parameters specified, ensuring global convergence for all but the simplest of cases seemed impractical.

4.2. Algorithm Descriptions and Benchmarking

4.2.1 Introduction

The NLOpt suite of nonlinear multivariate optimization algorithms collated and adapted under a common interface in C/C++ by Steven G. Johnson [**63**] was used for minimizing the displacements yielded by the FEA solution presented in Chapter 2. The following section compares the accuracy and convergence speed of the algorithms employed in this application for a variety of test cases. It should be noted that the performance of each algorithm depends heavily on the type of optimization problem. Consequently, each of the algorithms provided by NLOpt was benchmarked in order to ascertain their suitability towards the automated design of composite parts using FEA as envisaged in this thesis.

A complete description of each of the algorithms tested is beyond the scope of this work. A cursory description of each algorithm is provided however, along with references to sources which better describe them.

As outlined in Section 2.12, the algorithms provided by NLOpt which make use of the derivatives of the optimized function tend to converge far more rapidly, even for the smaller test cases examined below. These so-called derivative or gradient-based algorithms are typically based on Newton's method [**64**], whereby the roots of the gradients of the function being optimized are found using its second order derivatives, or Hessian matrix. As in the case of large problems it is often intractable to calculate the Hessian for each optimization step, an approximation of this matrix is often made resulting in a so-called quasi-Newton algorithm [**65**].

Alternatively, some approximation of the function being optimized can be generated simply by successive evaluation of that function at different test points, then employing this approximation to minimize the function. Indeed, NLOpt also provides a suite of derivative-free optimization algorithms which employ this paradigm and which were also included in the benchmarks. Typically, the models generated by derivative-free algorithms are less accurate than those which employ the gradient of the function being optimized, as the path of steepest descent is not readily apparent at each iteration without the benefit of a gradient. Descriptions and benchmarks of the derivative-free algorithms provided by the NLOpt library are also included, despite their inferior performance, even for the simplest test cases attempted.

4.2.2 Problem Parameters

The first test case consisted of a cantilevered beam of fibre-reinforced polymer composite, with geometry and loading condition identical to those described in Section 3.1.3. The optimization problem consisted in minimizing the deflection at the free end of the beam given a load imposed upon it, Figure 4-1. Table 4-1 lists the geometry and properties of the loaded part.

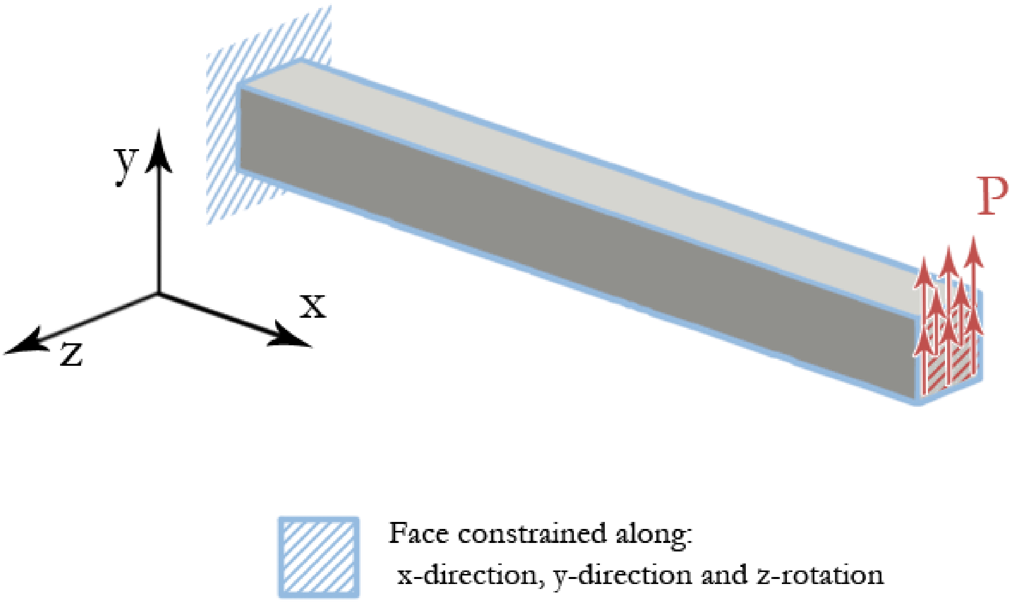


Figure 4-1: Beam geometry with shear loading

Table 4-1: Beam geometry and material properties

Length (L) along x-direction	0.1m
Width (w) along y-direction	0.01m
Thickness (t) along z-direction	0.01m
Fibre Modulus of elasticity	50×10^9 Pa
Resin Modulus of elasticity	1.5×10^9 Pa
Poisson's ratio of fibres	0.3
Poisson's ratio of resin	0.3

The optimization variables consisted of the fibre orientation and volume fraction in each element of the finite element mesh. Section 2.7.6 describes the way in which these parameters are defined. Specifically, the fibre volume fraction was limited to a value of 0.906 and initialized to 0.1% of that value at the start of each optimization run. A very small fibre volume fraction throughout was selected as a start point to the optimization runs because it approaches the worst case scenario. Volume fractions of 0 were avoided to sidestep numerical instabilities in the derivatives of the rotation matrix described in Equation **(2.120)**. The maximum value constraint for the fibre volume fraction corresponds to the theoretical maximum assuming hexagonal packing of uniform cylindrical fibres. Fibres were originally aligned along the y-axis in every element.

All optimization tests were executed using a personal computer running an AMD Phenom II x4 995 quad core 64-bit central processing unit (CPU) with 6GB of random access memory (RAM) and an ATI Radeon HD4890 graphics processing unit (GPU). Optimization benchmarks were measured using seconds instead of iteration counts as the time necessary to perform each iteration could vary depending on the algorithm used. Since every optimization run was performed on the same computer, the results presented herein should be consistent with each other and should provide an indication of the performance of the software program which was developed in a realistic setting.

Benchmarks were first carried out on a multitude of NLOpt algorithms at low FEA mesh resolution, therefore featuring comparatively few optimization variables. Specifically, the beam was first discretized into 48 tetrahedral elements, each of which bounded a volume whose fibre orientation and volume fraction were given by a 3D vector, for a total of 144 independent optimization variables. Each optimization run was allowed a maximum of 5 minutes to converge, and the results of these tests were compared against the results of an FEA simulation where the fibres were aligned longitudinally along the x-direction.

The higher resolution mesh used in the second wave of testing for this geometry consisted of 381 elements, each of which was described by a 3 parameter vector resulting in 1143 independent variables per test. Testing of the simplified variants of the truncated Newton algorithm was not carried out at this resolution because little difference was found between them and the full-featured truncated Newton algorithm in previous tests.

Tests were then carried out at higher resolutions on the subset of algorithms which had successfully converged to within 1% of the optimal value given by the on-axis test within the prescribed 2 hour time limit with the medium resolution mesh. The two algorithms examined in this manner were the Low-Storage Broyden-Fletcher-Goldfarb-Shannon (LBFGS) algorithm and the rank-2 Shifted Limited-Memory Variable-Metric (VAR2). These algorithms were each given 24 hours to optimize a simulation with 2070 elements and 6210 variables.

The results of these tests are discussed in more detail for each algorithm in the following sections. Graphs of the optimization runs performed using the cantilevered beam at various resolutions are provided after the description of each algorithm. These graphs show the deflection calculated by the FEA solver at each iteration of the run in blue, as compared to the lower bound corresponding to the deflection calculated by the FEA solver for the beam with all fibres aligned longitudinally along the x-direction, in gray. As described in Section 3.1.3 the solutions calculated by FEA solver depend on the resolution of the mesh, and eventually converge to an accurate value. As a consequence, the baseline graphed corresponds to the deflection of a longitudinally aligned fibre-reinforced part calculated using a mesh of the corresponding resolution.

4.2.3 Method of Moving Asymptotes

This algorithm described extensively in [**66**] uses the gradient of the optimized function and constraints in addition to quadratic penalty terms to form a conservative local approximation of the optimization function. The resulting approximation is convex by design, and a candidate point for the next optimization iteration can be obtained through its minimization by using a dual method. The optimization function and gradient is evaluated at the candidate point to determine if the approximations were indeed conservative. Should this be the case, the algorithm then proceeds to the next iteration starting at the candidate point. Otherwise, a more conservative approximation is obtained by increasing the quadratic penalty term.

The author claims that the MMA algorithm can be applied to problems with as many as 10^4 to 10^5 variables [**66**]. In practice however, this algorithm failed to converge to a value under 564% of that which was obtained with all fibres aligned longitudinally along the x-direction for a cantilevered beam described using 144 variables, which is the lowest resolution test which was attempted. Figure 4-2 shows the results of the optimization run on the 48 element mesh. It should be noted that this algorithm seems to be highly sensitive to the magnitude of the load being applied and consequently to the magnitude of the function being optimized. Because the FEA model used describes an elastic displacement response which scales linearly with the magnitude of a given load, this suggests that the starting quadratic penalty terms which are hard-coded in the implementation of this algorithm are ill-suited to the very small displacements which are being minimized.

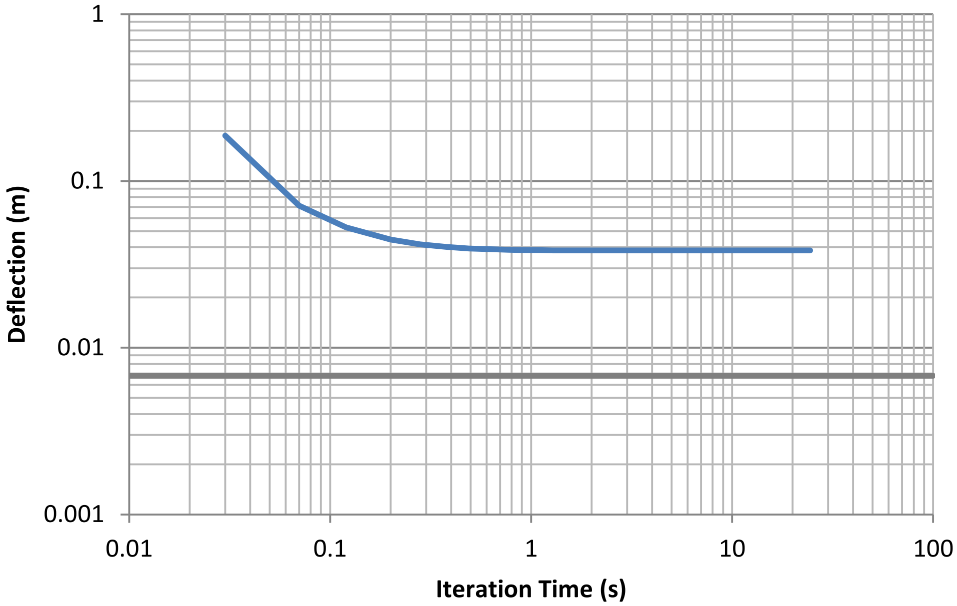


Figure 4-2: Cantilevered beam optimization; MMA 48 element test case

Unfortunately, the algorithm fared even worse with 1143 variables when the optimization of a 381 element mesh was attempted. Similarly, Figure 4-3 shows the results of the 381 element beam optimization run with its corresponding longitudinally aligned lower bound.

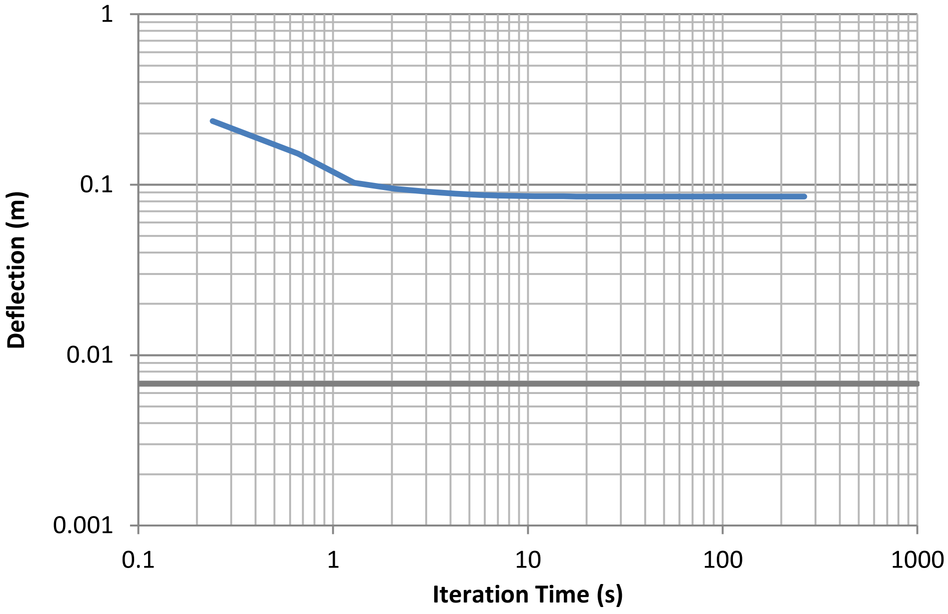


Figure 4-3: Cantilevered beam optimization; MMA 381 element test case

4.2.4 Sequential Least Squares Quadratic Programming

The Sequential Least Squares Quadratic Programming (SLSQP) algorithm aims at minimizing the optimized function by utilizing its value and its gradient for sequentially constructing and optimizing a series of quadratic approximations [**67**]. This sequence of constrained least squares problems is then solved to converge toward an optimum value to the function given. Of special note is the context in which this program was introduced [**67**]. Indeed, much like the work attempted in this thesis Kraft utilized this optimization method in conjunction with a model which numerically solved partial differential equations (PDE), in his case describing the equations of motion for industrial robotics. Much as in the present work, the nonlinear optimization routine and the PDE model were fed back into each other, yet remained distinct and independent in order to maximize the flexibility of the resulting program.

When compared with the other optimization routines, the SLSQP algorithm quickly converges to a displacement function under 200% of that which is obtained with all fibres aligned longitudinally for a cantilevered beam described using 144 variables. However, it stalled shy of achieving displacements smaller than 119% of those achieved by aligning all fibres in the longitudinal direction. Higher resolution tests involving 1143 variables also showed the algorithm quickly achieving displacements under 200% of the displacements calculated for the longitudinal fibre alignment baseline, and subsequently stalling at 136% of the longitudinal values. As a consequence, though the algorithm may be well-suited at seeding slower algorithms with a good preliminary guess, it was deemed too coarse for the purposes of this work.

Figure 4-2 shows the deflection calculated by the FEA solver at each iteration of the 48 element beam optimization run. Similarly, Figure 4-5 shows the results of the 381 element beam optimization run.

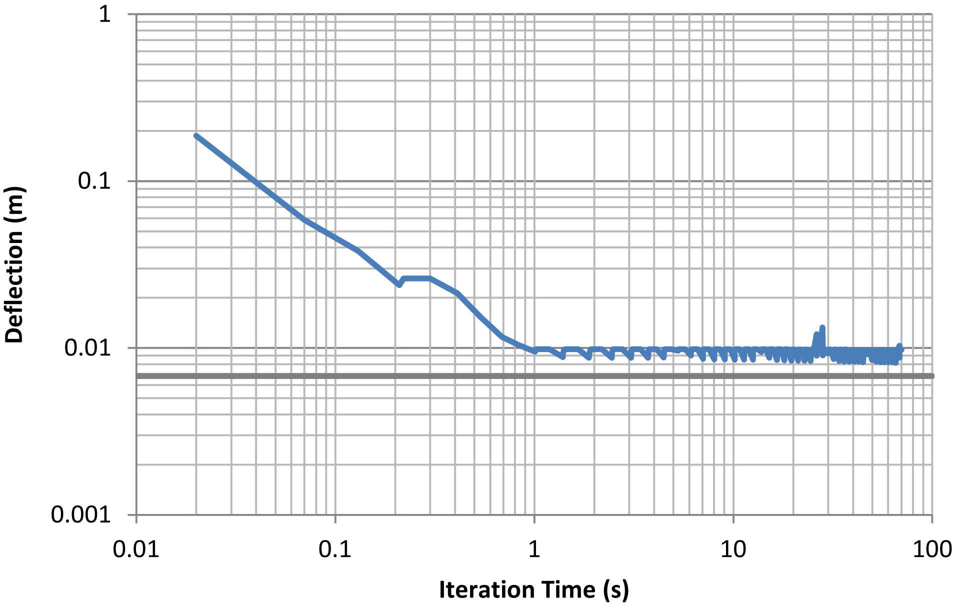


Figure 4-4: Cantilevered beam optimization; SLSQP 48 element test case

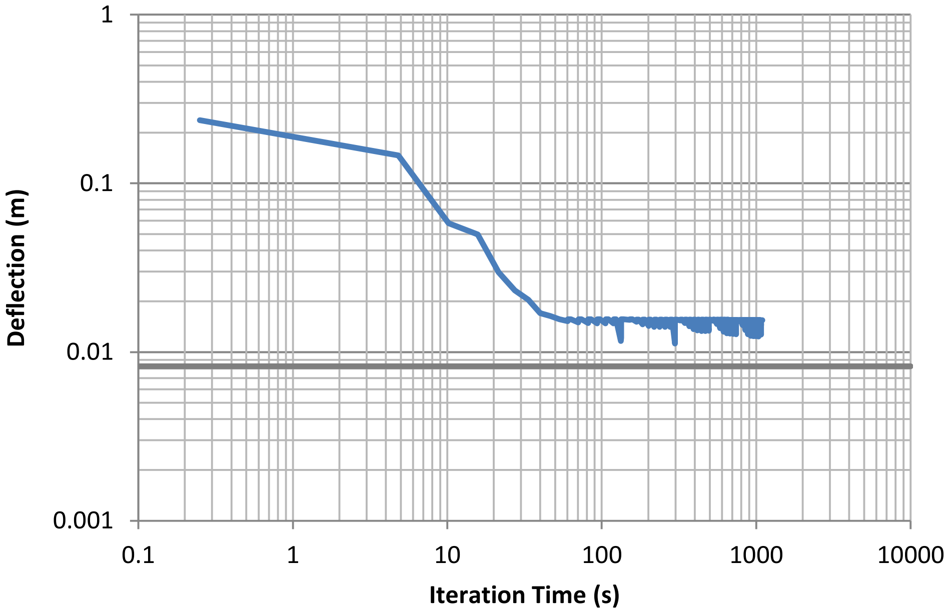


Figure 4-5: Cantilevered beam optimization; SLSQP 381 element test case

4.2.5 Low Storage Broyden-Fletcher-Goldfarb-Shannon

As described in [**68**] the LBFGS algorithm is a quasi-Newton method. Such methods attempt to find a local minimum to the function being optimized by finding zeros to the function's gradient. To that effect, quasi-Newton algorithms construct a quadratic approximation of the second order derivatives of the function represented by a Hessian matrix. Hessian matrices of a function are square matrices, the size of which corresponds to the number of variables in the function being optimized. Quasi-Newton methods are of great utility when the Hessian matrix is a large sparse matrix, as is often the case in problems with a large number of variables such as those examined in this work. In particular, the LBFGS algorithm further approximates the Hessian as a function of a set of previous optimization test points and their gradients.

When applied to the test cases in this work, LBFGS performed very well for both large and small problem sizes, outperformed only by the Shifted Limited-Memory Variable-Metric class of algorithms for the medium resolution test case featuring 381 elements. Figure 4-6 shows the deflection calculated by the FEA solver at each iteration of the 48 element beam optimization run.

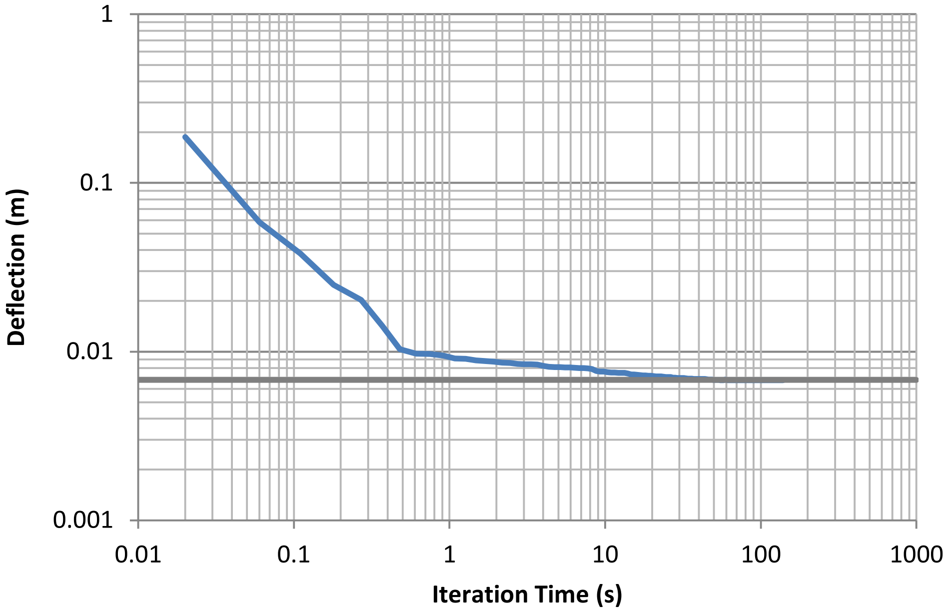


Figure 4-6: Cantilevered beam optimization; LBFGS 48 element test case

Similarly, Figures 4-7 and 4-8 show the results of the 381 and 2070 element beam optimization runs, respectively.

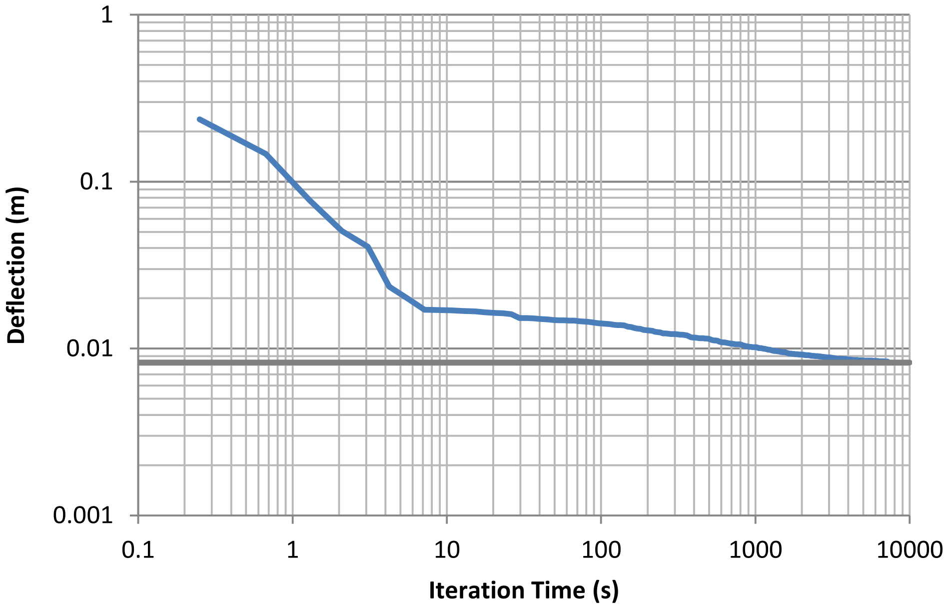


Figure 4-7: Cantilevered beam optimization; LBFGS 381 element test case

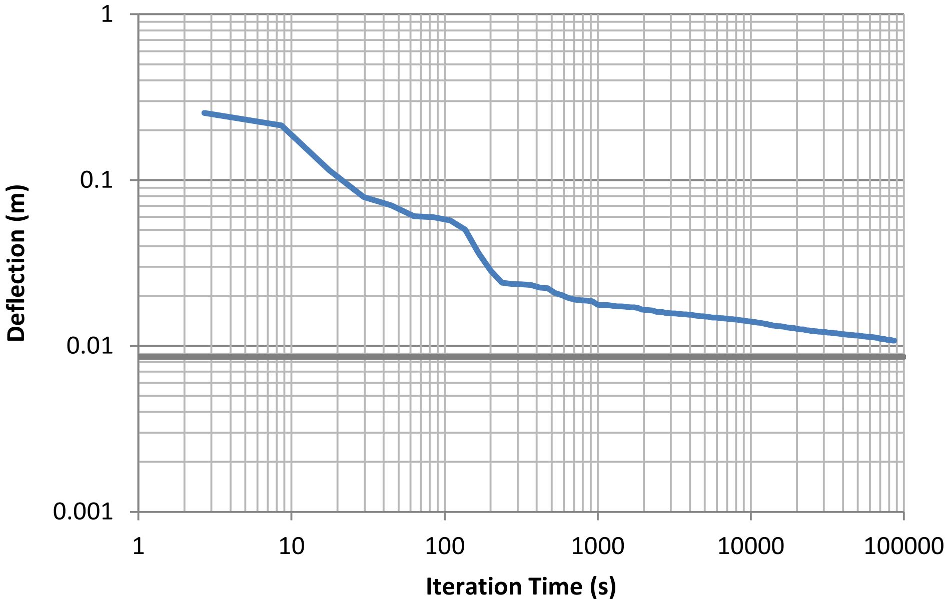


Figure 4-8: Cantilevered beam optimization; LBFGS 2070 element test case

4.2.6 Preconditioned Truncated Newton

Like the LBFGS algorithm described in the previous section this method, described by Dembo and Streihaug [**69**], is a quasi-Newton method. In this algorithm the computationally expensive task of solving exactly the Newton equations is eschewed until the algorithm nears the local minimum or the residual describing the accumulated inaccuracy of the more efficient truncated preconditioned conjugate gradient (PCG) algorithm [**65**] exceeds some specified value. As the more computationally efficient truncated PCG algorithm which is nested within Newton iterations allows for quicker convergence given an accurate approximation to the Hessian provided by the outer Newton iterations, this method often converges more quickly than a pure Newton algorithm. The NLOpt implementations of this algorithm include versions with and without the conjugate gradient conditioner and without steepest-descent restarting [**63**].

When applied to the test cases performed in this work, the Preconditioned Truncated Newton method performed very well for both small and large problem sizes. However, it failed to converge as quickly as the LBFGS and Shifted Limited-Memory Variable-Metric algorithms, especially when nearing the minimum. Figure 4-9 shows the optimization run for the 48 element mesh with preconditioning and restarting enabled. Tests were performed on all four variants of the TNEWTON algorithm provided by NLOpt, but no appreciable difference was found when applying the LBFGS preconditionner or restarting options.

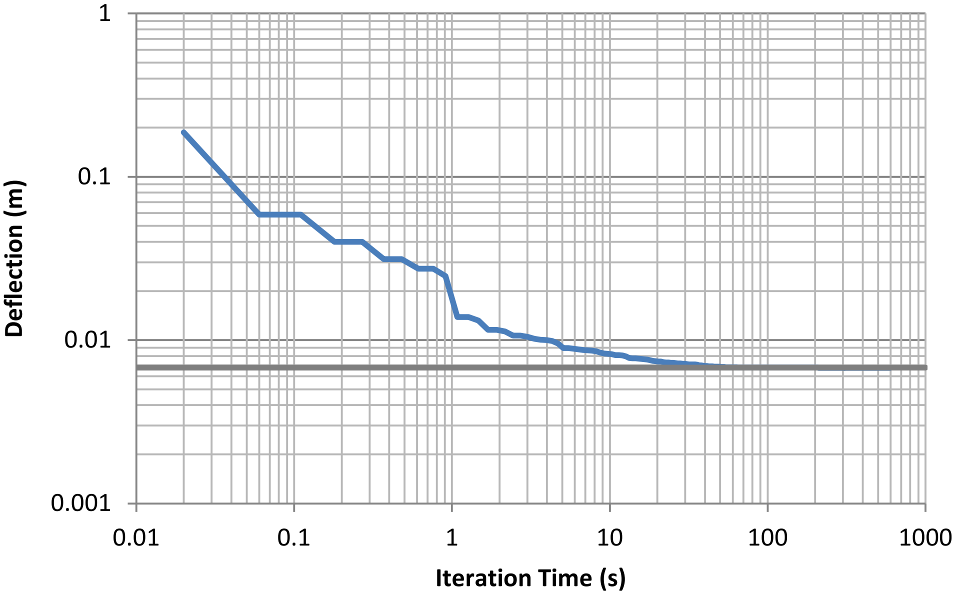


Figure 4-9: Cantilevered beam optimization; TNEWTON_PRECOND_RESTART 48 element test case

Given that no differences in performance were found between the simplified variants of the truncated Newton algorithm and the preconditioned and restarting versions, further testing on the full-featured variant of the TNEWTON algorithm was performed for the 381 element mesh. The algorithm performed very well at medium resolutions, outperformed only by LBFGS and the VAR1 and VAR2 algorithms. Figure 4-10 shows the results of this optimization run, where it is seen that convergence slows down as the algorithm nears the optimum.

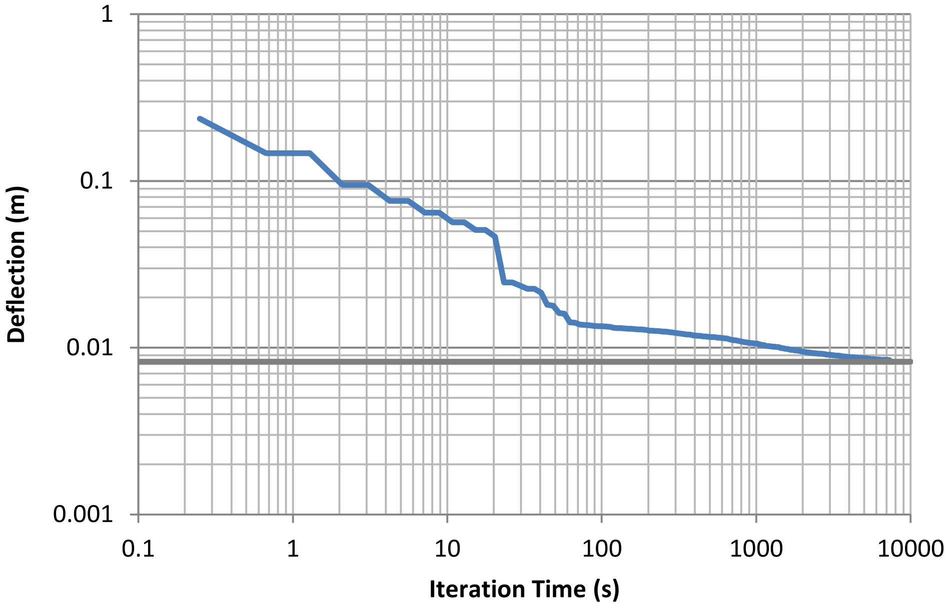


Figure 4-10: Cantilevered beam optimization;
TNEWTON_PRECOND_RESTART 381 element test case

4.2.7 Shifted Limited-Memory Variable-Metric

Like the LBFGS algorithm, the Shifted Limited-Memory Variable-Metric SLMVM algorithm is a truncated-Newton method which replaces the use of an explicit Hessian in favour of a more compact vector-based approximation [**70**]. The manner in which the Hessian approximation is constructed from the values and gradients of the optimized function from a set of recent iterations is a refinement of what is used in the LBFGS algorithm.

NLOpt provides an implementation of the shifted limited-memory variable-metric algorithm in of type 1 (VAR1) and type 2 (VAR2), where the type of method describes the rank, or number of independent columns and rows, of the matrix used to update the Hessian. Both these algorithms were benchmarked using the cantilevered beam problem with 48 and 381 element meshes. In these low and medium test cases, these algorithms converged very quickly when compared with the other algorithms tested. Figures 4-11 and 4-12 show the results of the 48 element tests for both the VAR1 and VAR2 algorithms respectively, whereas Figures 4-13 and 4-14 show the results of the 381 element tests for both algorithms.

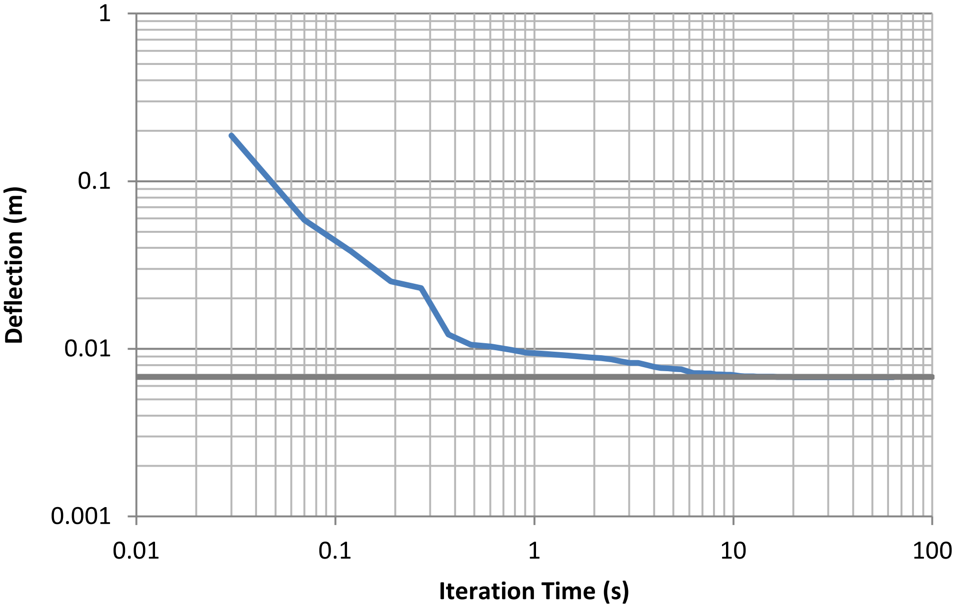


Figure 4-11: Cantilevered beam optimization; VAR1 48 element test case

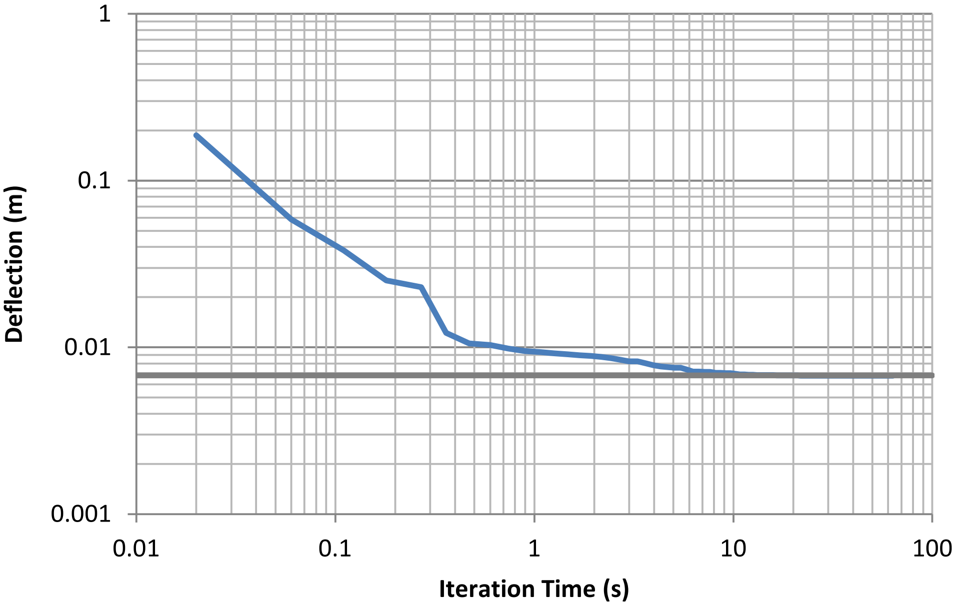


Figure 4-12: Cantilevered beam optimization; VAR2 48 element test case

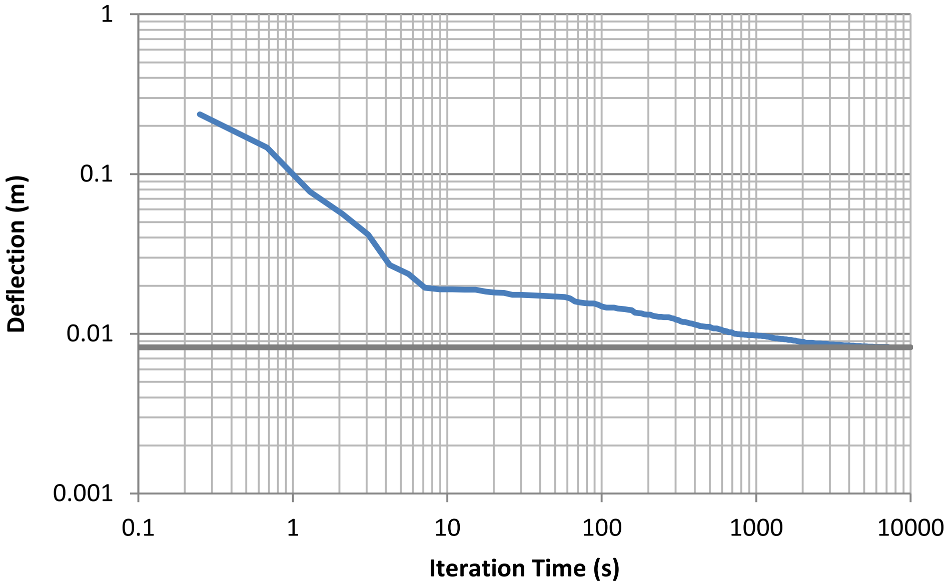


Figure 4-13: Cantilevered beam optimization; VAR1 381 element test case

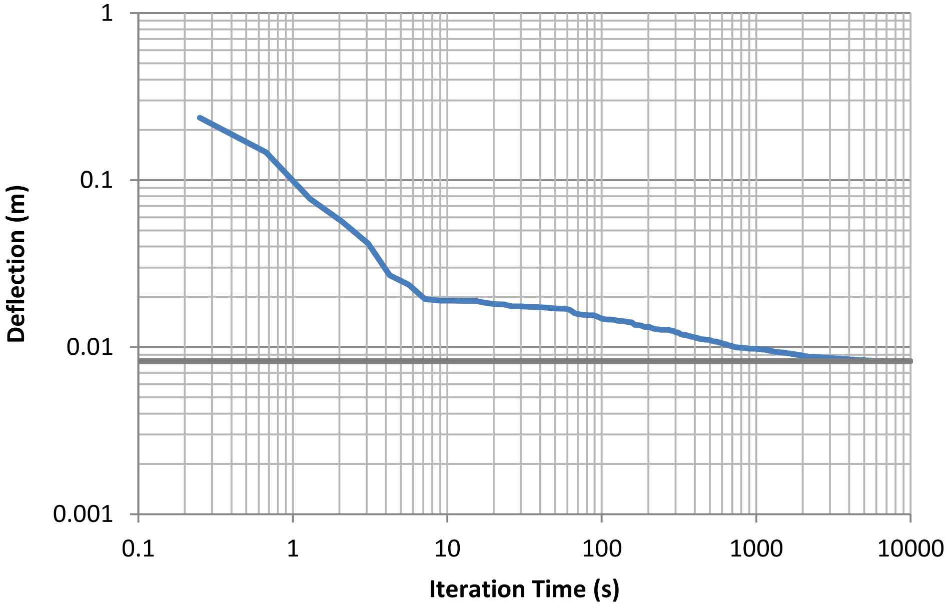


Figure 4-14: Cantilevered beam optimization; VAR2 381 element test case

The VAR1 and VAR2 algorithms performed similarly on both the 48 and 381 element meshes. Since these algorithms surpassed LBFGS at medium resolution, the rank 2 version of this algorithm was compared with the LBFGS algorithm described above using the 2070 element mesh. In this test case, LBFGS outperformed VAR2.

4.2.8 Constrained Optimization BY Linear Approximations

As described in [**71**] by Powell, the COBYLA algorithm constructs a linear approximation of the function being optimized over a range of test points in the optimization domain. At each iteration, test points are compared and moved successively to converge towards an optimum. Tests performed by Powell indicate that the Constrained Optimization BY Linear Approximations (COBYLA) algorithm is well suited to problems with as many as 100 variables. When testing the algorithm on the 48 element test case, however, convergence towards expected minima was much slower than with the derivative based algorithms described above. Figure 4-15 shows the results of this optimization run. As COBYLA is a derivative-free algorithm, the effect of each variable with respect to the optimization function must be ascertained independently. As COBYLA builds its linear approximation of the function being optimized, convergence begins slowly. To construct the linear approximation, the algorithm first alters one variable at a time and retains the value which best minimizes the function being optimized.

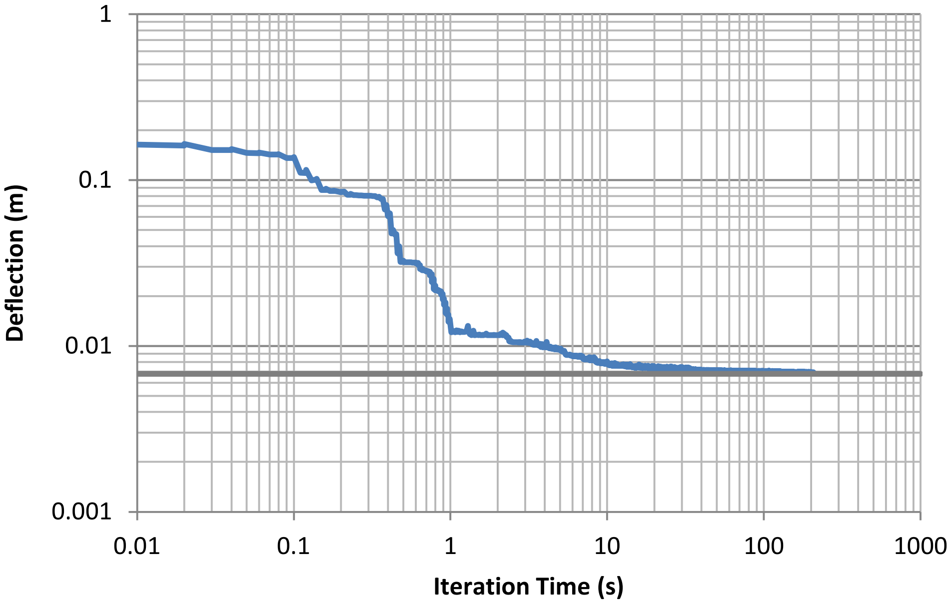


Figure 4-15: Cantilevered beam optimization; COBYLA 48 element test case

Figure 4-16 shows the record of the optimization run using COBYLA on the 381 element beam. Owing to the much larger number of parameters, the plateau corresponds to the successive evaluations of the optimization function where most of the optimization variables are near the non-optimal starting values during which time the simplex is being constructed. After approximately 100 seconds, the algorithm begins to converge much more rapidly, but evidently fails to compare to the convergence speed of the gradient-based algorithms described above.

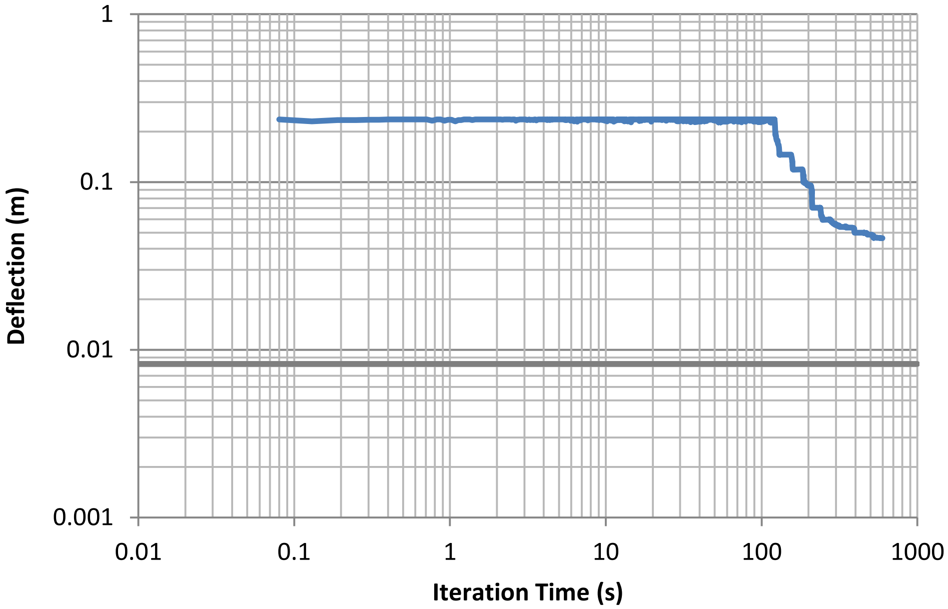


Figure 4-16: Cantilevered beam optimization; COBYLA 381 element test case

4.2.9 Bound Optimization BY Quadratic Approximation

Similar to COBYLA, BOBYQA interpolates the function being optimized over a set of test points. However, BOBYQA makes use of quadratic rather than linear interpolation [**72**]. Figures 4-17 and 4-18 show the optimization runs using the BOBYQA algorithm on the 48 and 381 element beam tests. As BOBYQA is the quadratic equivalent to COBYLA, it is unsurprising that it inherits COBYLA's convergence characteristics. The plateau seen with the 48 element case is more prevalent owing to the nature in which BOBYQA constructs its initial approximation to the function being optimized. Indeed, each parameter is evaluated first at the midpoint of the bounds given to the algorithm, and then reset to zero, resulting in very small differences in the deflections observed.

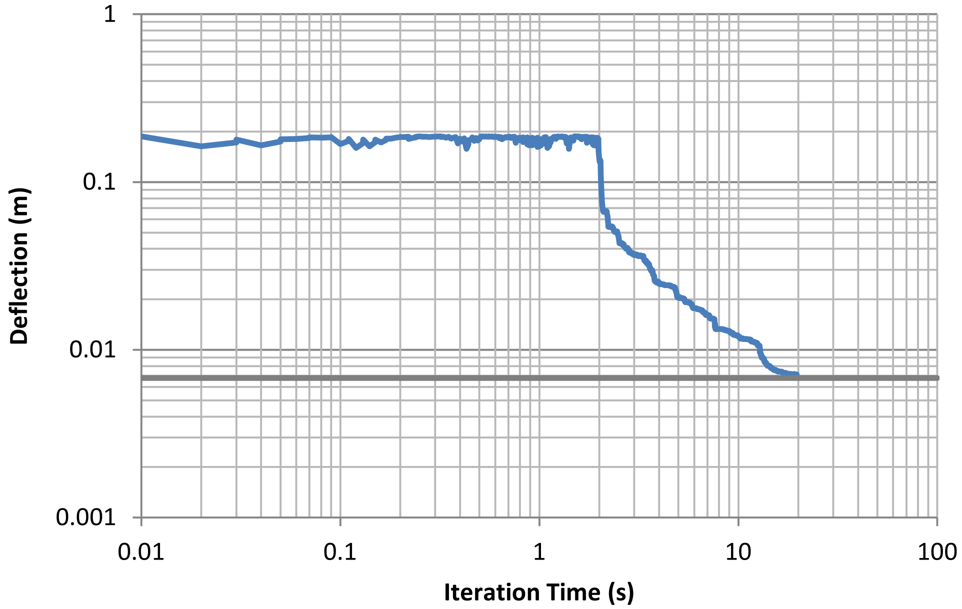


Figure 4-17: Cantilevered beam optimization; BOBYQA 48 element test case

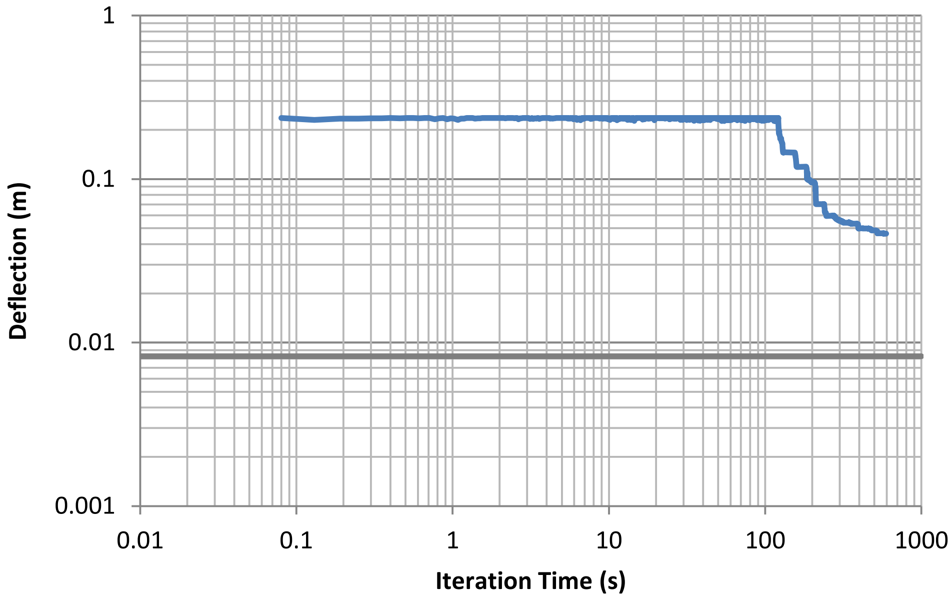


Figure 4-18: Cantilevered beam optimization; BOBYQA 381 element test case

4.2.10 Principal Axis

As described in [**73**] the principal axis method generates a number of search directions corresponding to the number of DoFs describing the function being optimized. The function is then optimized along each successive search direction. A quadratic approximation of the test function is then used to reorient the search directions and the next iteration begins anew.

As with the other derivative-free algorithms, PRAXIS must evaluate the optimization function multiple times to inform its initial search direction. In particular, PRAXIS begins by performing a series of line searches over every optimization variable, each of which has a very small impact on the optimized function. This explains the slow convergence characteristic at the beginning of the optimization run seen for both the 48 element mesh and the 381 element mesh in Figures 4-19 and 4-20 respectively. The algorithm then re-evaluates suitable search directions from these preliminary results, resulting in faster convergence until the algorithm approaches the optimum and convergence finally slows down. The figures also show sharp peaks in the function being optimized, resulting from numerical instability in the solution of the displacements and thought to be more prominent because of the very large number of iterations needed for convergence. Section 6.1.5 details the methods used to account for such numerical instabilities.

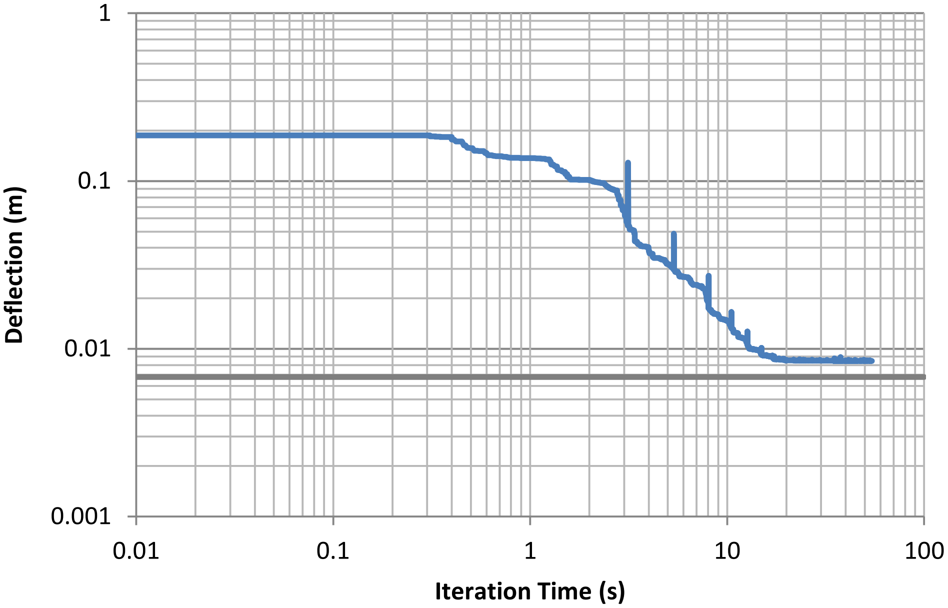


Figure 4-19: Cantilevered beam optimization; PRAXIS 48 element test case

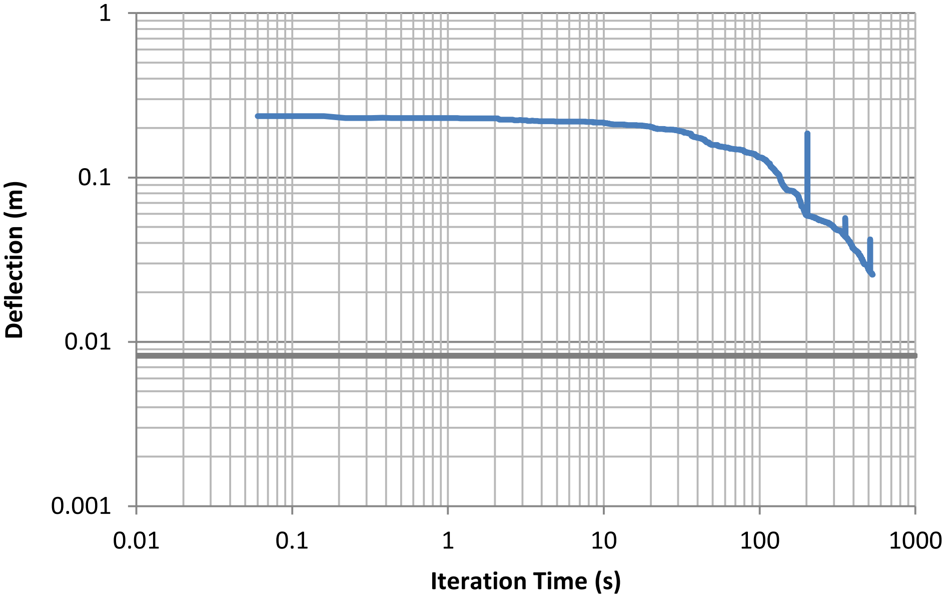


Figure 4-20: Cantilevered beam optimization; PRAXIS 381 element test case

4.2.11 Subplex

This method described by Rowan [**74**] is an extension of the Nelder-Mead Simplex, itself similar in nature to the COBYLA method wherein a simplex of points is created in order to approximate linearly the function being optimized. In this case however, the function is being optimized in a sequence of low-dimensional subspaces which can be optimized more rapidly. Because of the large number of optimization variables being changed at once, the initial convergence rate is limited until a larger subset of the part volume is assigned reasonable fibre volume fractions. As a consequence the optimization curves for this algorithm, shown in Figures 4-21 and 4-22 for the 48 and 381 element cases respectively, resemble COBYLA's.

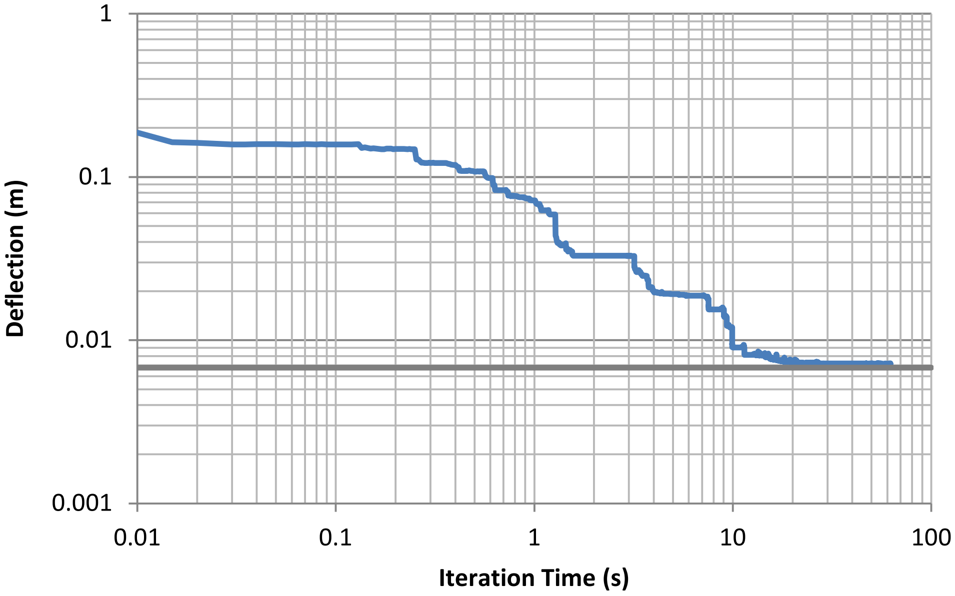


Figure 4-21: Cantilevered beam optimization; SUBPLEX 48 element test case

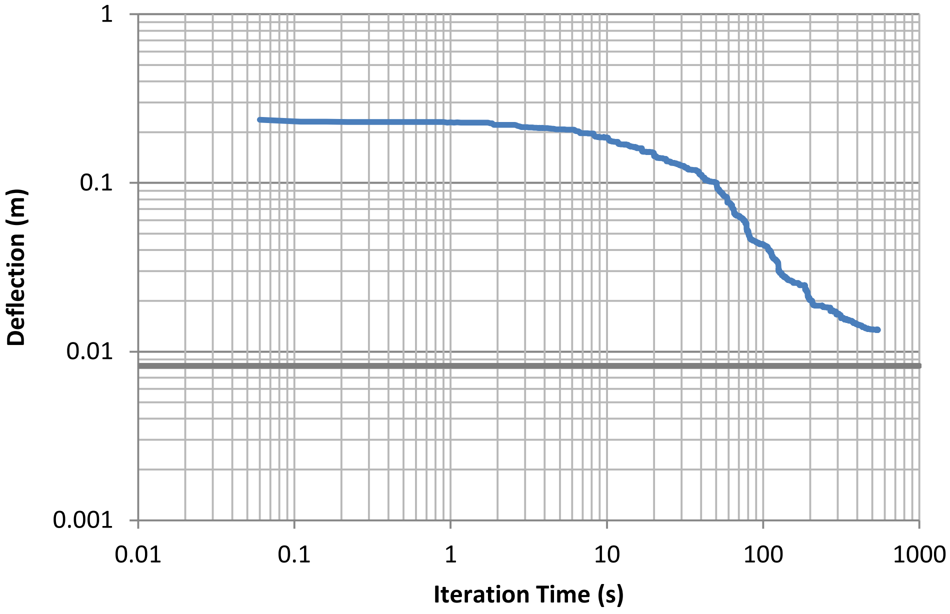


Figure 4-22: Cantilevered beam optimization; SUBPLEX 381 element test case

4.2.12 Algorithm Analysis and Comparison

The displacement at the loaded end in each optimization run was compared against the theoretical lower bound given by the FEA simulation with the same mesh, at the maximum fibre volume fraction and with all fibres aligned longitudinally along the beam. This section summarizes the results of the benchmarks performed, as well as the results of the on-axis simulation against which each optimization run was compared.

Because each algorithm may differently evaluate convergence criteria imposed through NLOpt's interface, a uniform means of grading their convergence characteristics was needed. To that effect, the optimization results were compared with the results for the bending of a beam with the same geometry and its fibres aligned longitudinally, as the longitudinal fibre alignment case is known to be the global optimum for the simple geometry and loading presented in this chapter.

To ascertain both the efficiency and the accuracy of the algorithms, the time it took for each algorithm to come within sequentially tighter margins of this global optimum is compared in Tables 4-2 and 4-5 for the 48 and 381 element test cases respectively. Cells in the optimization result tables are colour-coded to illustrate convergence times and solution accuracy, where greener hues indicate the fastest convergence times and graded towards yellow for slower convergence times. Cells were left blank and coloured red when the algorithm failed to converge within the given threshold. Times to convergence for the three thresholds are each separated in columns. Furthermore, the values of deflection corresponding to the three tolerance thresholds are found in Tables 4-3 and 4-6 for the 48 and 381 element test cases respectively.

To further measure the performance of the algorithms, the final value upon the termination each optimization run and the corresponding time it took to achieve these results are given in Tables 4-4, 4-7 and 4-8, for the 48, 381 and 2070 element test cases respectively. Again, the table cells are colour-coded, in this case with gradients from red to yellow to green to indicate final values which approach the global minimum most closely.

Observing the results found in Table 4-2, it is immediately apparent that most derivative-based algorithms perform much better than the derivative-free algorithms. Indeed, LBFGS, TNEWTON, VAR1 and VAR2 converged far more quickly than BOBYQA, COBYLA, PRAXIS and SBPLX, none of which managed to approach within 1% of the theoretical minimum given by the longitudinally aligned simulation performed in Section 3.1.3 and indicated in Table 4-3. It should be noted that MMA and SLSQP terminated before achieving an adequate solution to the optimization problem.

Table 4-2: Cantilevered beam optimization results, 48 element test case

Algorithm	Time for convergence to relative margins [s]			Time at finish [s]
	100%	**10%**	**1%**	
LBFGS	0.48	13.36	45	141.02
MMA				24.47
SLSQP	0.69			69.94
TNEWTON	1.28	27.2	80.65	599.77
TNEWTON PRECOND	1.27	27.07	80.1	597.92
TNEWTON PRECOND RESTART	1.47	18.76	49.36	597.31
TNEWTON RESTART	1.48	18.88	49.66	599.06
VAR1	0.37	5.88	12.01	63.52
VAR2	0.36	5.85	11.98	63.36
BOBYQA	7.67	15.66		19.67
COBYLA	1.29	14.43		207.09
PRAXIS	10.43			54.29
SBPLX	9.32	17.26		62.19

Table 4-3: Relative margins of longitudinally aligned deflection global minimum, 48 element test case

Value at relative margin [m]			
100%	**10%**	**1%**	**Exact Value (0%)**
0.013598	0.00748	0.00687	0.00679901

Observing the final values of deflection after the optimization, it can be seen that LBFGS, TNEWTON, VAR1 and VAR2, which converged relatively quickly, were also the most effective at finding the minimum given enough time, managing to reduce the deflection at the tip to values marginally smaller than the expected global minimum corresponding to the case where all fibres are exactly aligned longitudinally.

Table 4-4: Final value of optimization function and optimization time,
48 element test case

Algorithm	Time at finish [s]	Final Value [m]	Difference from longitudinal minimum
LBFGS	141.02	0.00675879	-0.59%
MMA	24.47	0.0383728	464.39%
SLSQP	69.94	0.00814009	19.72%
TNEWTON	599.77	0.00676074	-0.56%
TNEWTON PRECOND	597.92	0.00676071	-0.56%
TNEWTON PRECOND RESTART	597.31	0.00676036	-0.57%
TNEWTON RESTART	599.06	0.00676043	-0.57%
VAR1	63.52	0.0067601	-0.57%
VAR2	63.36	0.0067601	-0.57%
BOBYQA	19.67	0.00712137	4.74%
COBYLA	207.09	0.00693231	1.96%
PRAXIS	54.29	0.00846199	24.46%
SBPLX	62.19	0.00718275	5.64%

Figure 4-23 shows the optimization curves for every algorithm tested using the 48 element mesh. The exponential convergence of the gradient-based methods towards the optimum on-axis value is readily apparent, as are their superior convergence characteristics when compared to the derivative-free methods.

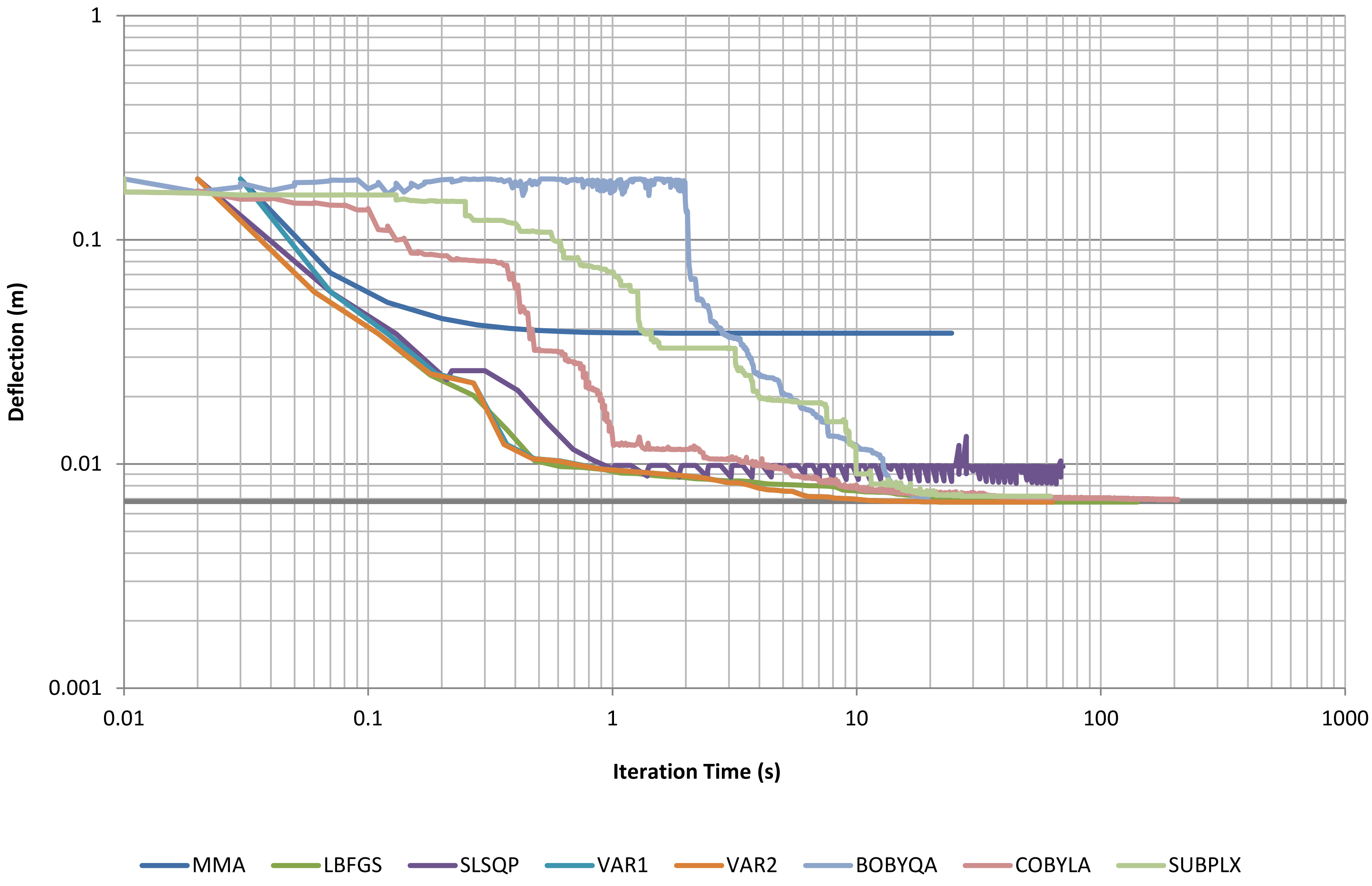


Figure 4-23: Cantilevered beam optimization; algorithm comparison 48 element test case

Convergence times were found to be much higher as the mesh size was increased from 48 elements to 381 elements. This is due to the increase in the number of dimension of the optimization problem, and also to the increase in the time required for the FEA solver to calculate the part's deflection upon each optimization iteration. Ultimately, it is evident that the derivative-free algorithms BOBYQA, COBYLA, PRAXIS and SBPLX performed much worse than the derivative-based algorithms at this medium resolution.

Table 4-5: Cantilevered beam optimization results, 381 element test case

Algorithm	**Time to convergence [s]**			**Time at finish [s]**
	200%	**110%**	**101%**	
LBFGS	20.34	2339.2	🟥	7196.27
MMA	🟥	🟥	🟥	262.62
SLSQP	46.46	🟥	🟥	1100.69
TNEWTON PRECOND RESTART	52.93	3093.0	🟥	7283.48
VAR1	67.24	1817.3	5402.1	7215.29
VAR2	67.17	1808.6	5389.4	7203.24
BOBYQA	🟥	🟥	🟥	595.38
COBYLA	69.15	🟥	🟥	7170.38
PRAXIS	🟥	🟥	🟥	531.12
SBPLX	312.74	🟥	🟥	542.19

Table 4-6: Relative margins of longitudinally aligned deflection global minimum, 381 element test case

Value at relative margin [m]			
100%	**10%**	**1%**	**Exact Value (0%)**
0.016465	0.00906	0.00831	0.00823267

Table 4-7: Final value of optimization function and optimization time, 48 element test case

Algorithm	Time at finish [s]	Final Value [m]	Difference from longitudinal minimum
LBFGS	7196.27	0.00835326	1.46477
MMA	262.62	0.0853649	936.9042
SLSQP	1100.69	0.0112109	126.17575
TNEWTON PRECOND RESTART	7283.48	0.00842161	2.295
VAR1	7215.29	0.00825266	0.24281
VAR2	7203.24	0.00825266	0.24281
BOBYQA	595.38	0.0463556	463.06885
COBYLA	7170.38	0.0144365	75.35623
PRAXIS	531.12	0.0255821	210.7388
SBPLX	542.19	0.0134467	63.3334

The final test cases examined were nearing usable limits of the program and hardware. At 2070 elements, the two quickest optimization algorithms tested at lower resolutions reached the nominal prescribed limit of roughly 24 hours before achieving less than 25% over the longitudinal baseline value. Given additional time, the algorithms would likely have approached much closer to the global minimum.

Table 4-8: Final value of optimization function and optimization time, 2070 element test case

Algorithm	Time at finish [s]	Final Value [m]	Difference from longitudinal minimum
LBFGS	86605.9	0.0107754	25.26%
VAR2	84410.2	0.0219145	154.74%

Chapter 5

Applications

5.1. Software Program Features and Use

5.1.1 Introduction

The simulations and optimizations presented in previous chapters were aimed at ascertaining the accuracy and convergence speed of the software program which was developed. To that effect, the meshes which were tested consisted of simple bars and beams in order to enable comparisons between the results obtained from well-established equations. With the aim of demonstrating the software program in a more realistic application, the two cases examined in this chapter focus on more complex geometries which highlight the intended use of the program. An outline of the design philosophy, features and limitations of the program follow the two test cases.

5.1.2 Test Cases

Findings regarding actual use of the software tool described in Chapter 3 and Chapter 4 help delineate the practical computational limits of the software program which was developed in its present implementation. In light of these limitations, for fibre orientation problems mesh sizes of approximately 300 to 1500 elements represent adequately the elastic behaviour of simple parts whilst converging towards an optimum solution regarding fibre orientations and volume fractions within a reasonable timeframe. Denser meshes upwards of 5000 elements were deemed to be necessary to characterize thin or complex parts using tetrahedra with a reasonable aspect ratio.

As the program's convergence rate is dependent on the number of elements and optimization variables, both of which are linked currently, the size of the parts examined is inversely proportional to the resolution at which they are modeled and optimized. Parts with larger volumes and coarser resolutions can be optimized as quickly as smaller parts with finer resolution.

The geometries presented in this section were meshed using Tetgen then fed into the software program's optimizer. As the LBFGS algorithm described in Section 4.2.5 was the best suited at solving the test case in Chapter 4, it was selected to demonstrate the program's utility in this chapter. The optimization cases were then left to run until successive iterations could not produce values reduced by at least 0.0001% (1×10^{-6}) of the previous minimum, or until a specified time had elapsed. The following sections describe each part, loading and optimization criterion in addition their performance relative to this metric.

The fictitious materials specified in the optimizations presented in this chapter were assigned credible values derived from the moduli of carbon fibres and epoxy resins; properties are listed in Table 5-1. For optimizations performed in this chapter the maximum fibre volume fraction was fixed at a reasonable value of 60% [**53**]. Contrary to optimizations performed in Chapter 4, as there is no gain in reducing the fibre volume fraction with respect to stiffness optimization and since the program has already been proven to converge automatically towards high volume fraction values, the fibre volume fraction is not a variable; instead it was set to the maximum value specified.

Table 5-1: Material properties for proof of concept optimizations

Modulus of elasticity, fibres	50×10^9 Pa
Modulus of elasticity, resin	1.5×10^9 Pa
Poisson’s ratio, fibres	0.3
Poisson’s ratio, resin	0.3
Fibre volume fraction (fixed)	0.6

5.1.2.1 Double S-Bracket

The first geometry which was optimized consists of a 42 x 44 mm plate sectioned so as to feature a reticulated path between the two opposing sides, as shown in Figure 5-1.

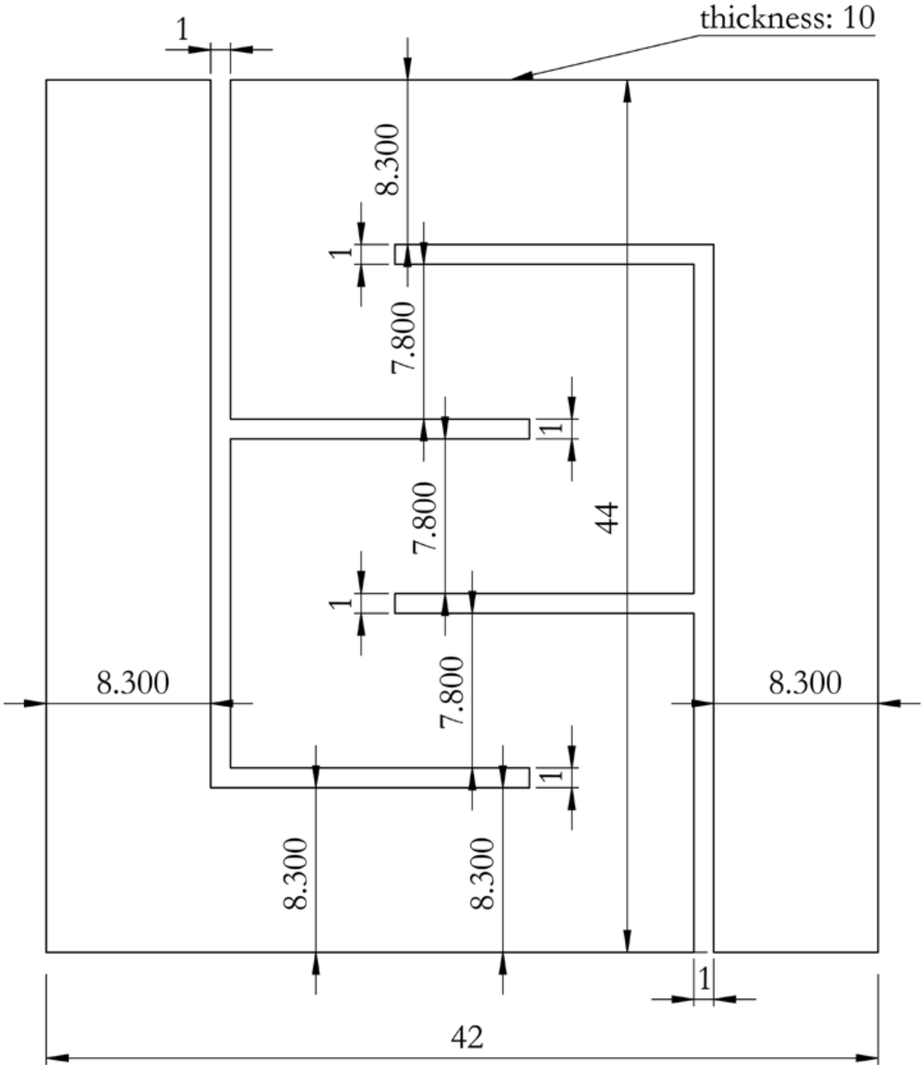


Figure 5-1: Double S-bracket – Geometry in mm

Deformations along the y-direction of the nodes lying on the face opposite to the constrained face were set to be equal, and these nodes were loaded along the y-direction with a uniformally distributed load of 1N. An isometric view of the part including the constraints and the chosen loading mode can be found in Figure 5-2.

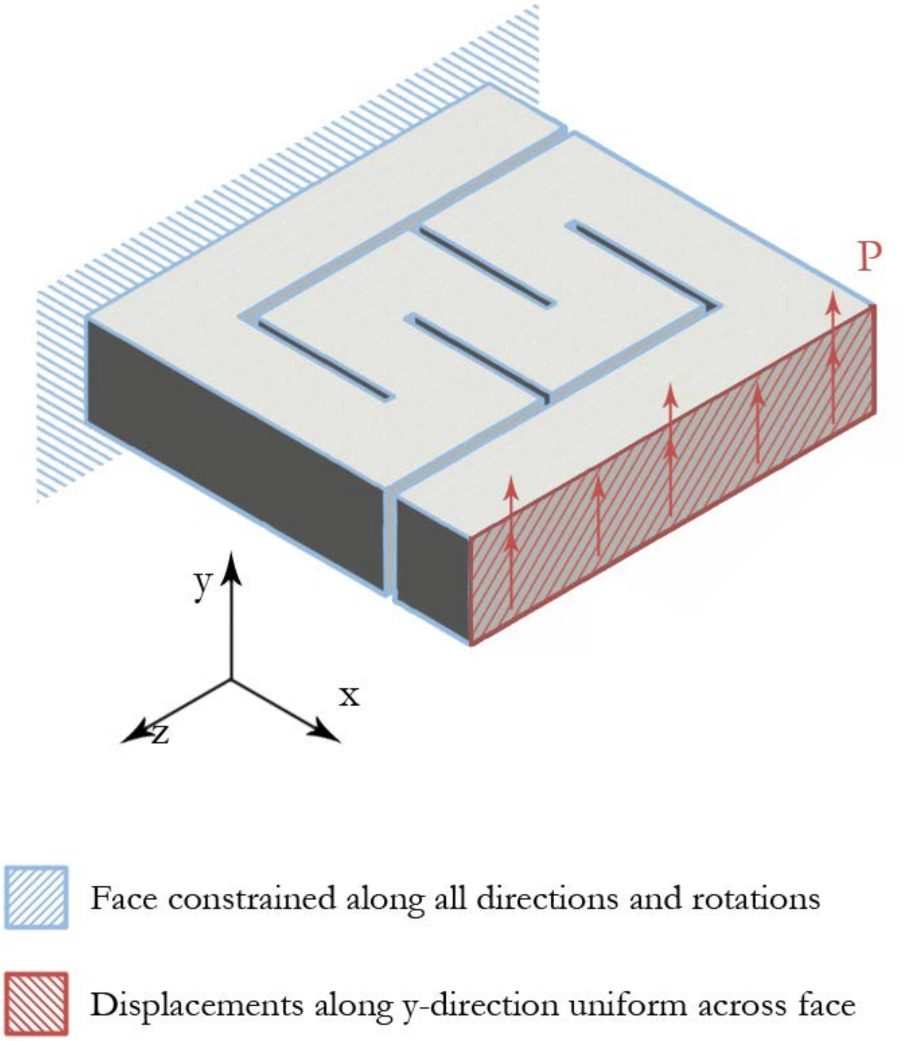


Figure 5-2: Double S-bracket – Constraints and loading mode

This geometry was created using the 3D modeling software Solidworks and exported as an STL file which was imported into Tetgen. A description of Tetgen can be found in Section 2.9.1. Each element's fibre orientation was initially set collinear with the x-axis. Figure 5-3 shows the part's deformation under load before and after optimization, as well as the outer edges and faces of the mesh generated by Tetgen. Through the optimized alignment of fibres in each element, deformations resulting from the applied loading were reduced by 48% from those seen with the initial configuration.

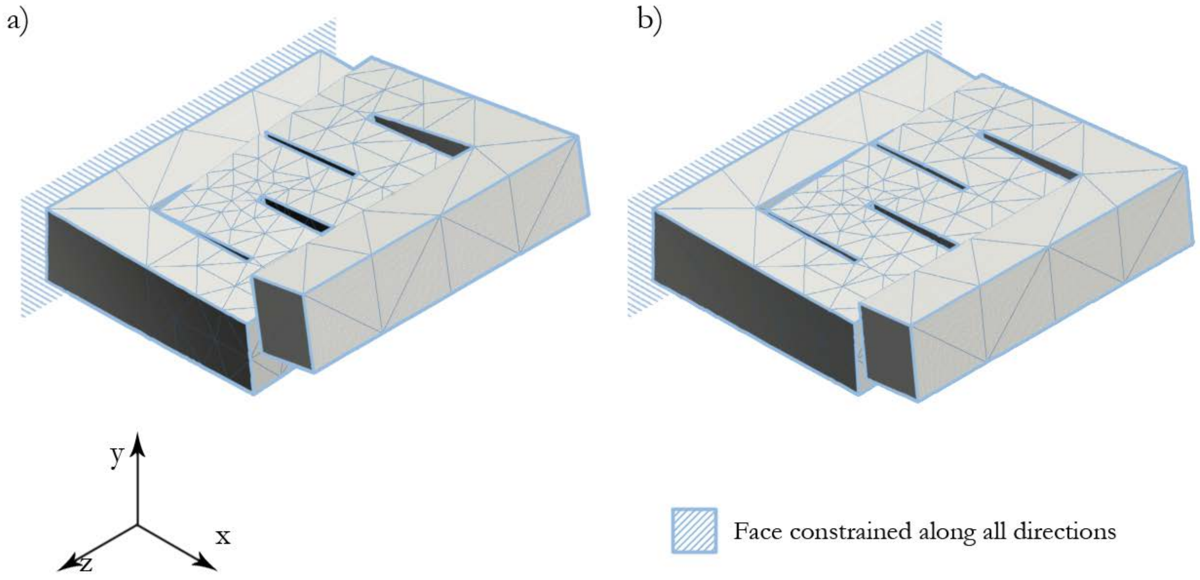


Figure 5-3: Double S-bracket – Optimization snapshots
Deformation x 1000;
a) First optimization iteration at 0 seconds, b) Final optimization iteration at 7133 seconds

The optimization routine was run with the parameters specified for a maximum of 2 hours. Figure 5-4 plots the deformation at the loaded end as a function of optimization time. As with previous test cases using the LBFGS algorithm, the software program converged asymptotically towards a local minimum deflection. In this case, a minimum deflection of 4.49×10^{-6} m was attained along the y-axis at the loaded nodes.

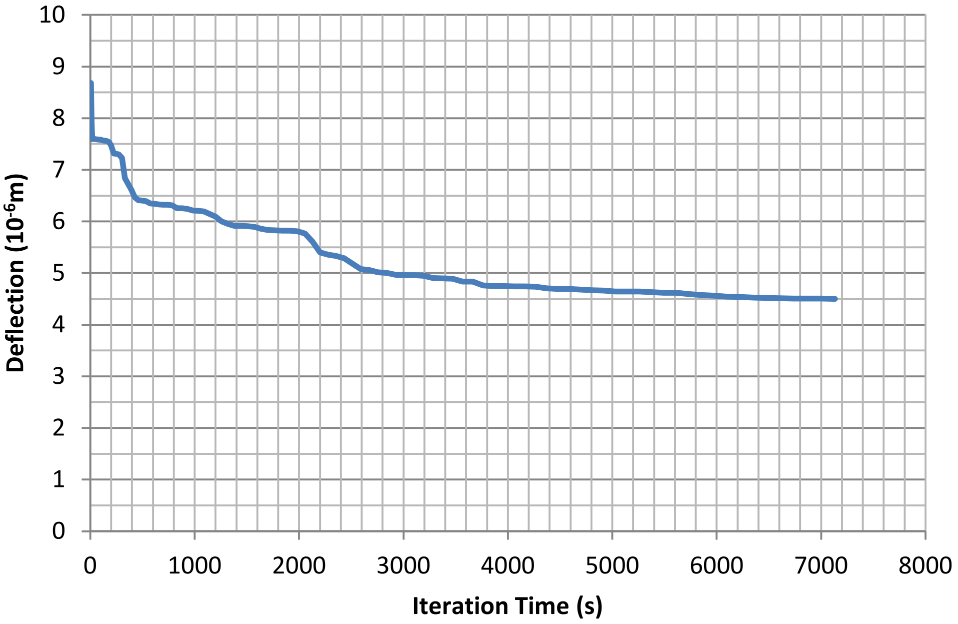


Figure 5-4: Double S-bracket – Displacement at loaded end vs. optimization time

Figure 5-5 shows top views of the part with vectors representing fibre orientations at each element, at different times during optimization. The rate of change of fibre orientations during optimization is generally proportional to the derivative of the displacement at the loaded end with respect to that change. The magnitude of these derivative is dependent on several factors. For instance in Figure 5-5, larger elements and therefore elements where vectors describe larger volumes and displacements for a given strain are clearly favoured by the optimization program. Likewise, elements under high stresses and therefore where strains are also relatively important, also see their fibres reorient quickly. Finally, elements at the base of beam sections seem to converge the quickest due not only to the high stresses resulting from large bending moments but also to the large induced deflections at the far end of the beams for a given strain. The mechanical properties of some elements were of little consequence to the metric being optimized, namely the elements along the top left and bottom right corners of the part in Figure 5-5 which are under very little stress from the loads imposed. For this reason, the orientations of the fibres in these elements remained essentially constant throughout optimization.

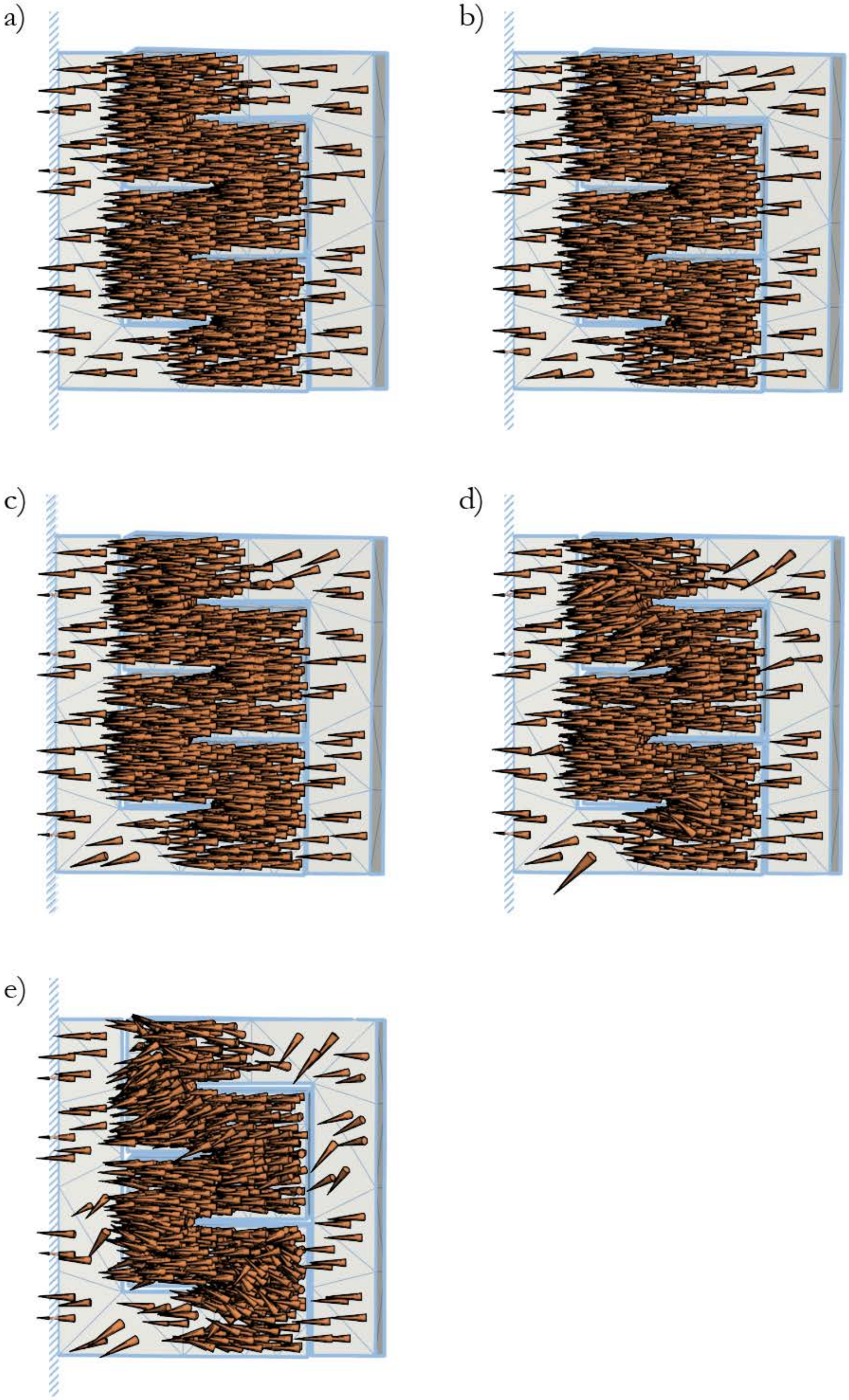


Figure 5-5: Double S-Bracket – Top view featuring parameter vector overlay Deformation x 2500, Vectors showing fibre orientation; a) 0 seconds, b) 7 seconds, c) 71 seconds, d) 713 seconds, e) 7133 seconds.

5.1.2.2 Torsion Bracket

The second geometry examined consisted of a series of connected beams, each of which is oriented along different orthogonal axes and with 20mm chamfers in each interior corner. The length of each beam segment is 100mm.

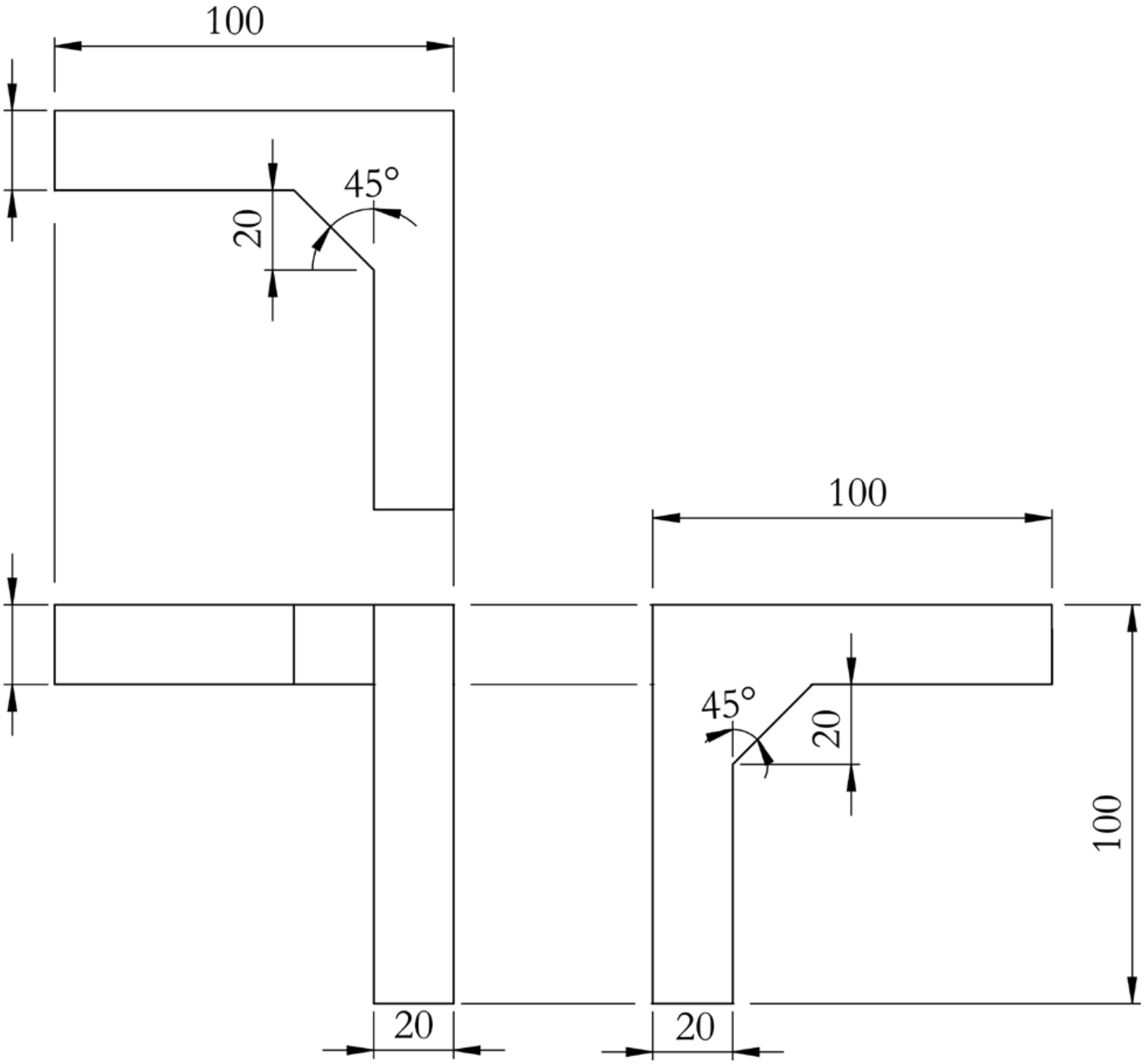


Figure 5-6: Torsion bracket with chamfered corners – Geometry in mm

The part was subjected to a load of 1N along the negative z-direction. Figure 5-7 shows an isometric view of the constraints and loading subjected to the optimization process.

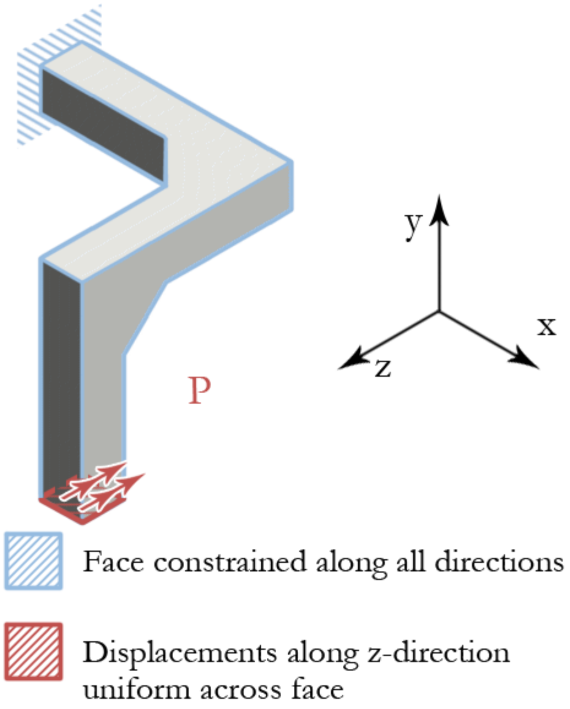


Figure 5-7: Torsion bracket with chamfered corners – Constraints and Loading Mode

As with the double-S bracket geometry described in the previous section, the part geometry described in this section was created using the 3D modelling software Solidworks and the resulting STL file was inputted into Tetgen, this time resulting in a 309 element mesh. In turn, this mesh was used for the optimization routine with each element thereof seeded with material with fibres aligned along the x-axis.

The optimization routine was run with the specified parameters for a maximum of 2 hours or until the changes in the optimization metric incurred by each iteration approached 1×10^{-6} of the current metric value, as was the case after approximately 35 minutes. Figure 5-8 shows the displacement at the loaded end of the part throughout the optimization process. The optimization routine reduced the displacement at the loaded end successfully by up to 66% from the initial iteration. Like in the previous case and those conducted in Section 4.2.5 on the square slender beam using the LBFGS algorithm, the optimization routine converged asymptotically towards a minimum. As the mesh being optimized had fewer elements than the previous case the optimization algorithm converged more quickly towards an optimum.

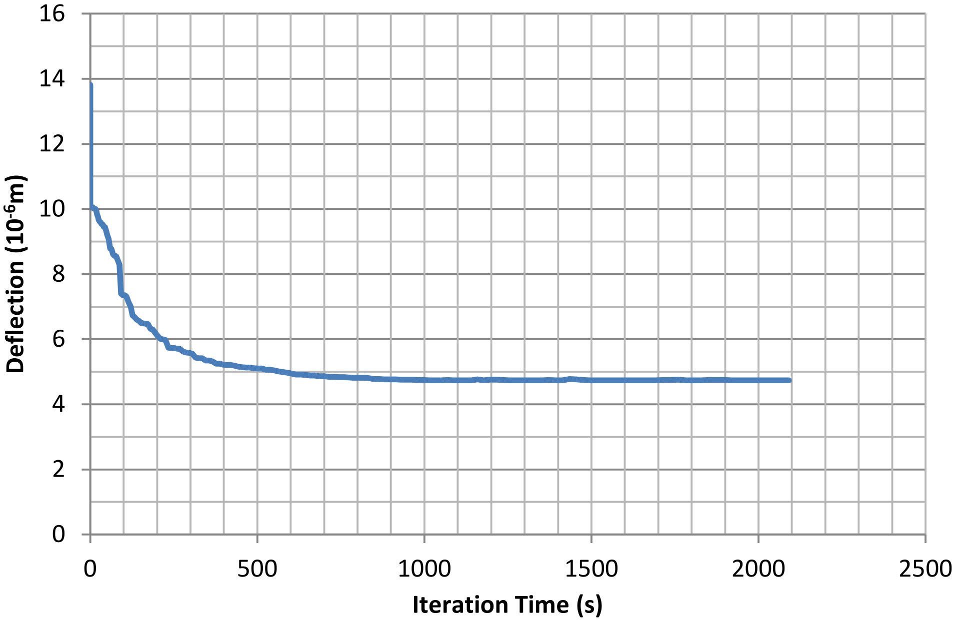


Figure 5-8: Torsion bracket with chamfered corners – Displacement at Loaded End vs. Optimization Time

Figure 5-9 shows isometric views of the part with vectors representing fibre alignment at each element, at different times during optimization. As with the previous section, the rate of change in fibre orientations is proportional to the impact on the deflection at the loaded end. The final result depicted in Figure 5-9 e) shows elements at the fixed end of the loaded part having fibres aligned in order to best resist torsion and bending loads resulting from oading. The elements comprising the second beam section aligned along the z-direction have fibres aligned along the z-direction to resist bending in the y-z plane. Interestingly, the fibres near the loaded end of the beam are not well aligned to resist bending in the y-z plane. The misalignment is a result of the fact that this final beam is stiffened more thoroughly by the gusset joining it to the intermediate section aligned along the z-direction than the other two beams. Therefore, the alignment of the fibres in these elements should have a smaller impact on the deflection of the part given this loading mode and produce smaller derivatives with regards to the optimization value.

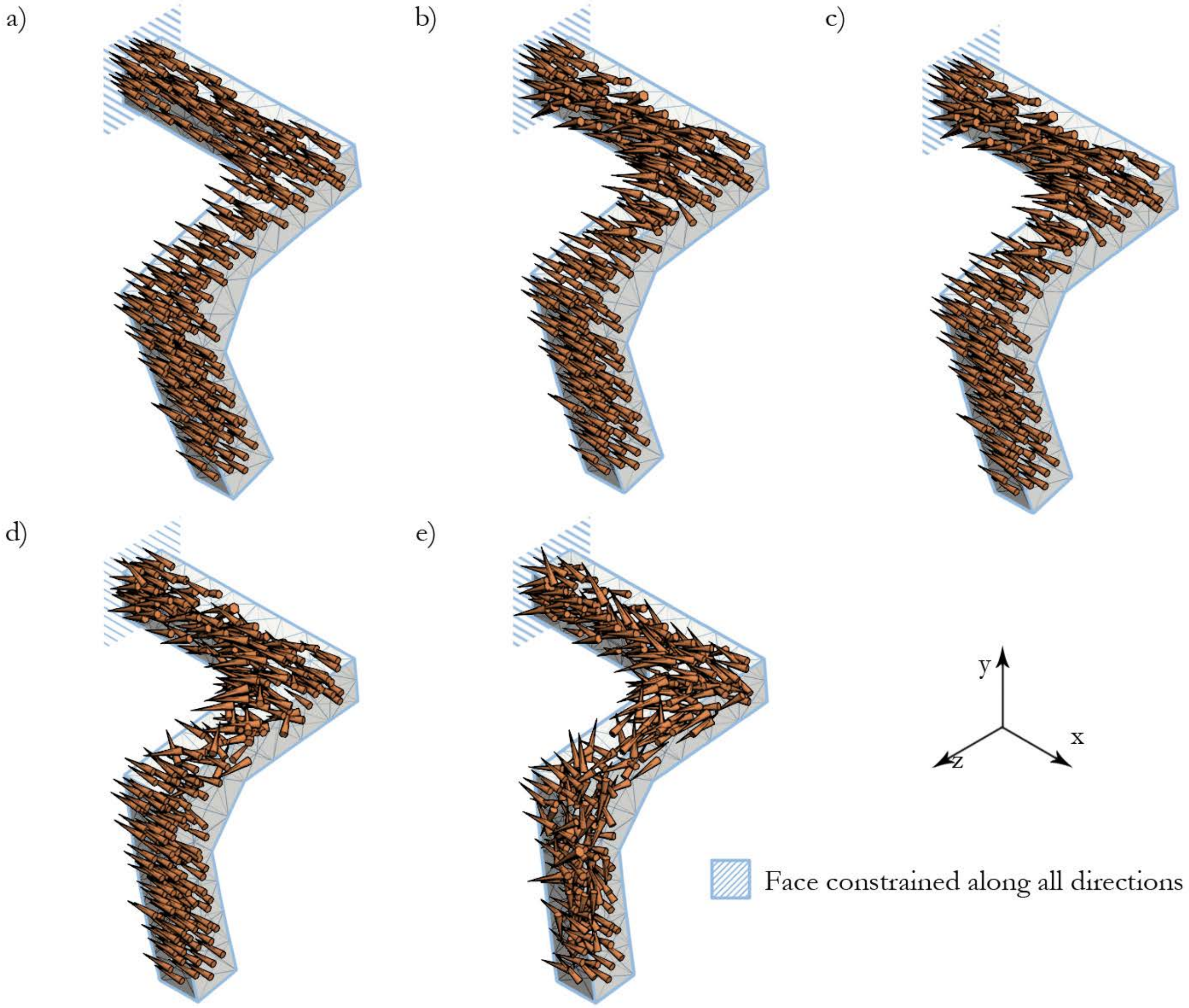


Figure 5-9: Torsion bracket with chamfered corners – Optimization Snapshots featuring parameter vector overlay Deformation X 2500, Vectors showing fibre orientation; a) 0 seconds, b) 2 seconds, c) 20 seconds, d) 206 seconds, e) 2067 seconds.

5.1.2.3 Stepped Shell

The third geometry was based on a +/-45° carbon fibre laminated part constructed previously, which is shown in Figure 5-10.

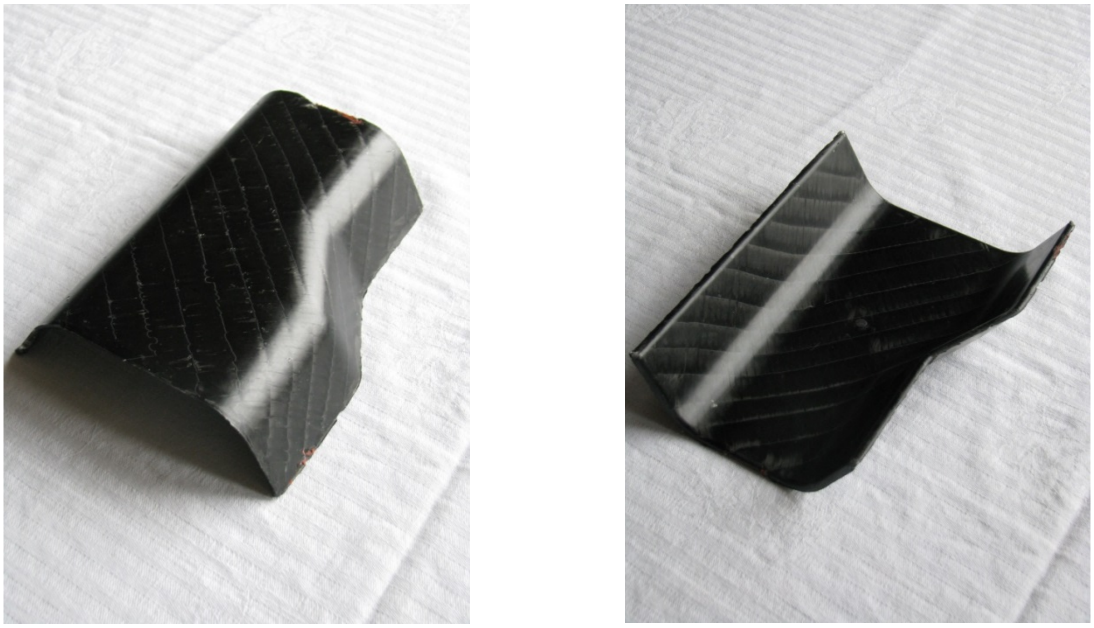


Figure 5-10: Stepped shell – Reference part

This stepped shell is defined by the geometry shown in Figure 5-11. The part was recreated in Solidworks and dimensioned in inches to conform with the pre-existing part and mesh. Figure 5-12 shows the part dimensioned in mm.

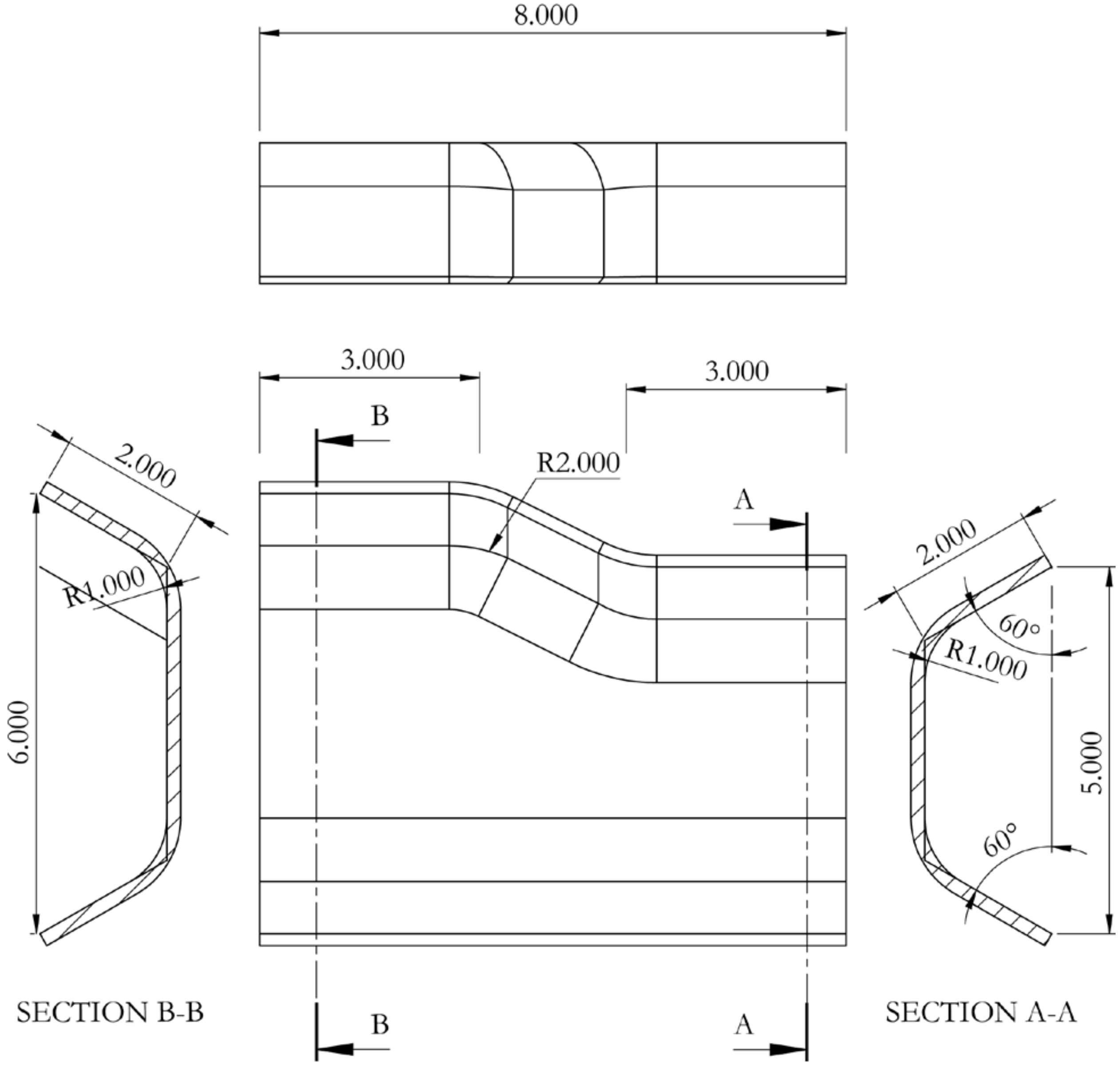


Figure 5-11: Stepped shell – Geometry in inches

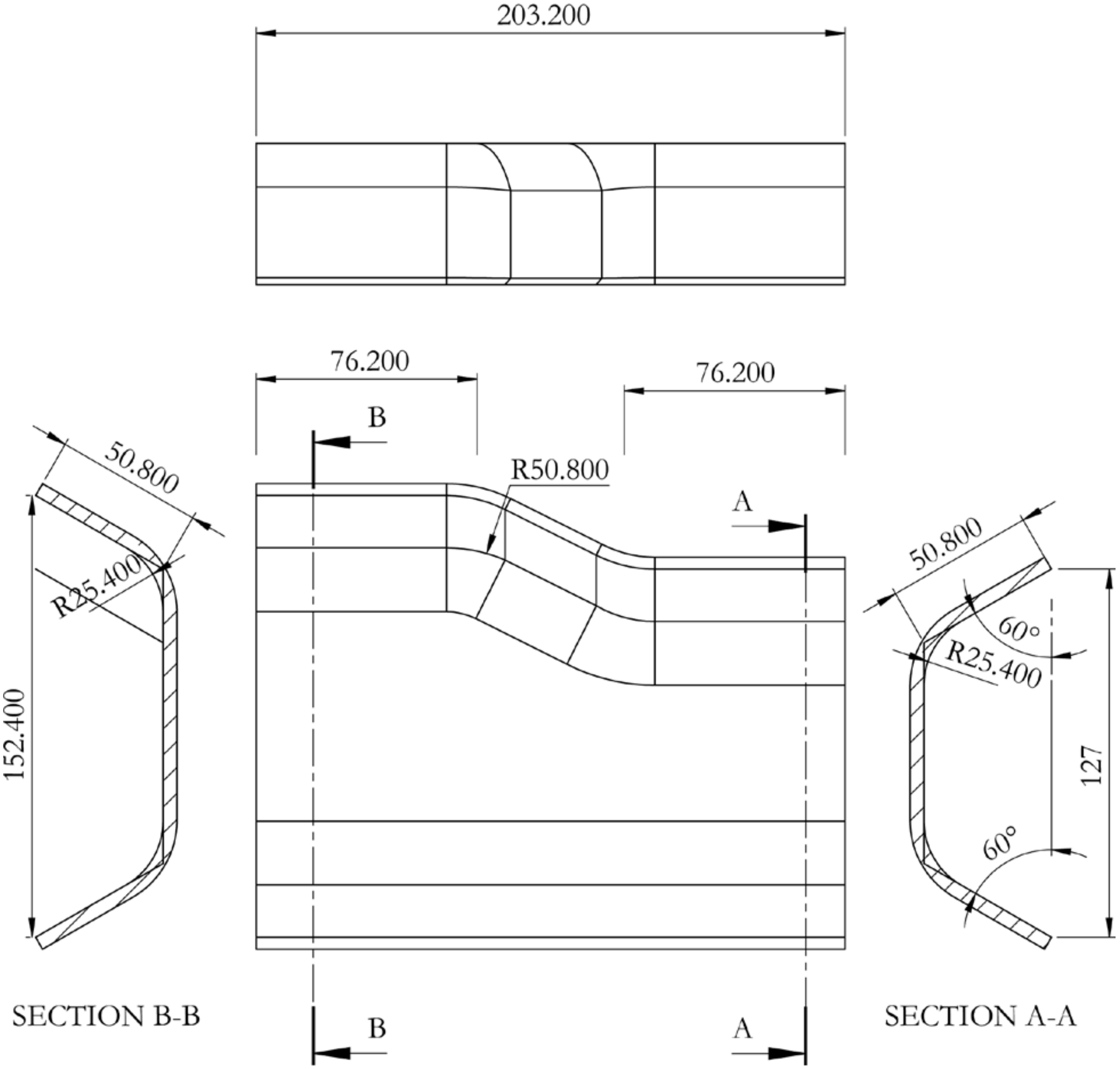


Figure 5-12: Stepped shell – Geometry in millimetres

The meshes were exported from Solidworks in STL format and post-processed for input into Tetgen and the optimization software program.

Two optimization cases were considered. In the first optimization case, a point load of 1000 Newtons was applied in the negative y-direction on a node near the center of the piece, as shown in Figure 5-13. The optimization routine was run with these parameters for a maximum of 12 hours, with initial fibre orientations aligned along the z-direction and an optimization metric consisting of the displacement at the loaded node along the load direction.

In the second case optimized fibre orientations from case 1 were kept, and a point load of 3000 Newtons was applied in the positive x-direction along the edge of the part in addition to the 1000 Newton load mentioned above. Nodes along the bottom-most edges of the part were constrained from moving along the y-direction in both optimization cases, and nodes coincident with the rearmost x-y plane were constrained from moving along the x-direction in the second case. The optimization routine was then run for an additional 12 hours for this case, with an optimization metric consisting of the sum of the displacement at the loaded nodes weighted by the relative magnitude of the loads applied, specifically $f_{opt} = d_1 + 3 \cdot d_2$, where d_1 is the displacement at the node loaded by P_1 and d_2 is the displacement at the node loaded by P_2.

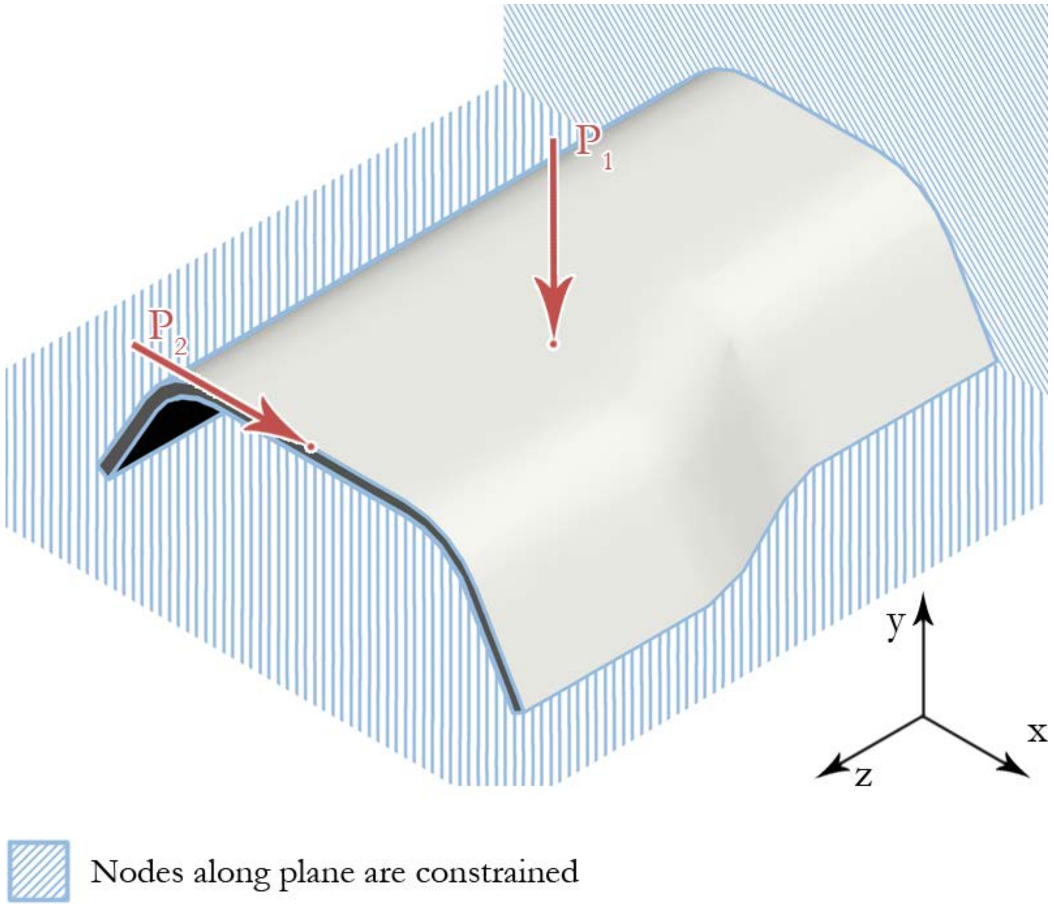


Figure 5-13: Stepped shell – Constraints and loading mode

Figure 5-14 shows optimization metrics and the displacements at the loaded nodes as a function of optimization time for both optimization cases. Through the optimized alignment of fibres in each element, the software program converged asymptotically towards a local minimum deflection of 8.46×10^{-3} m along the y-axis at the loaded node, resulting in a reduction of 58% from that seen in the initial configuration for case 1. Likewise the optimization metric based on deformations resulting from the two applied loads in case 2 was reduced by 32%.

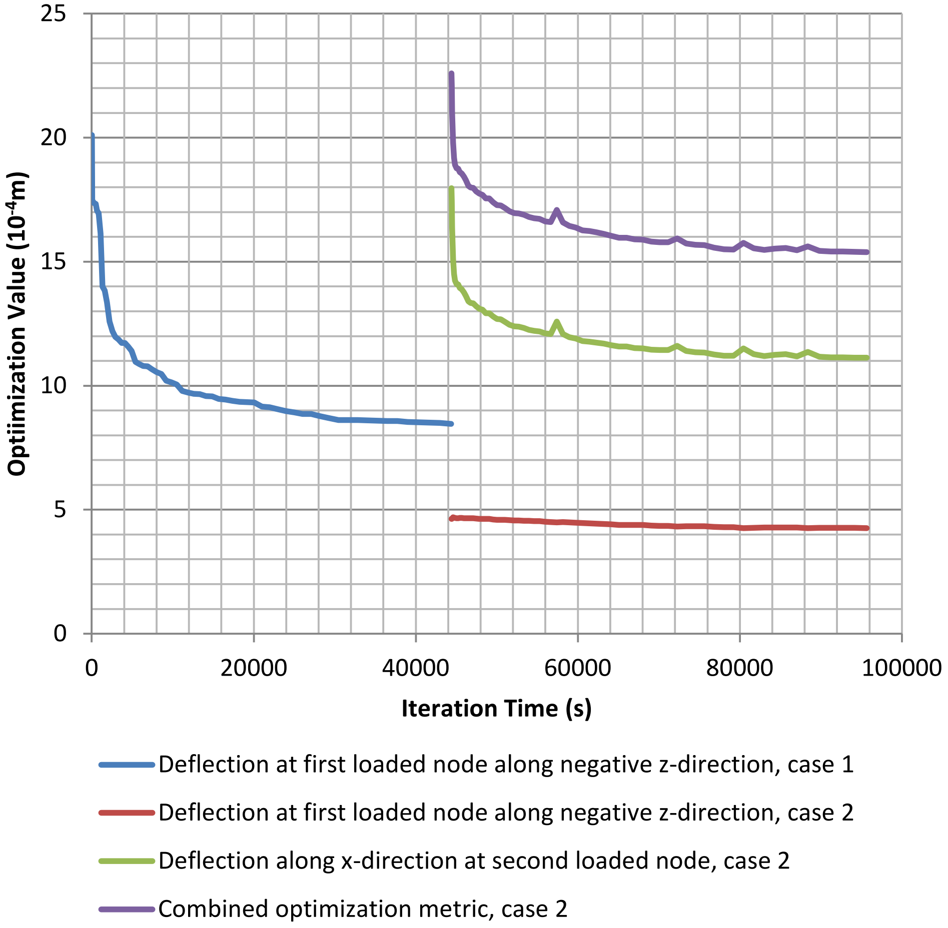


Figure 5-14: Stepped shell – Displacements at loaded nodes and optimization metrics vs. optimization Time

Figure 5-15 shows isometric views of the part with vectors representing fibre alignment at each element, at different times for optimization during the first load case. As seen in previous sections, the rate of change in fibre orientations is proportional to its impact on the optimization metric. The final result shown in Figure 5-15 (f) illustrates that fibre orientations near the loaded node align so as to reduce the displacement of the node by stiffening the part's transverse bending stiffness.

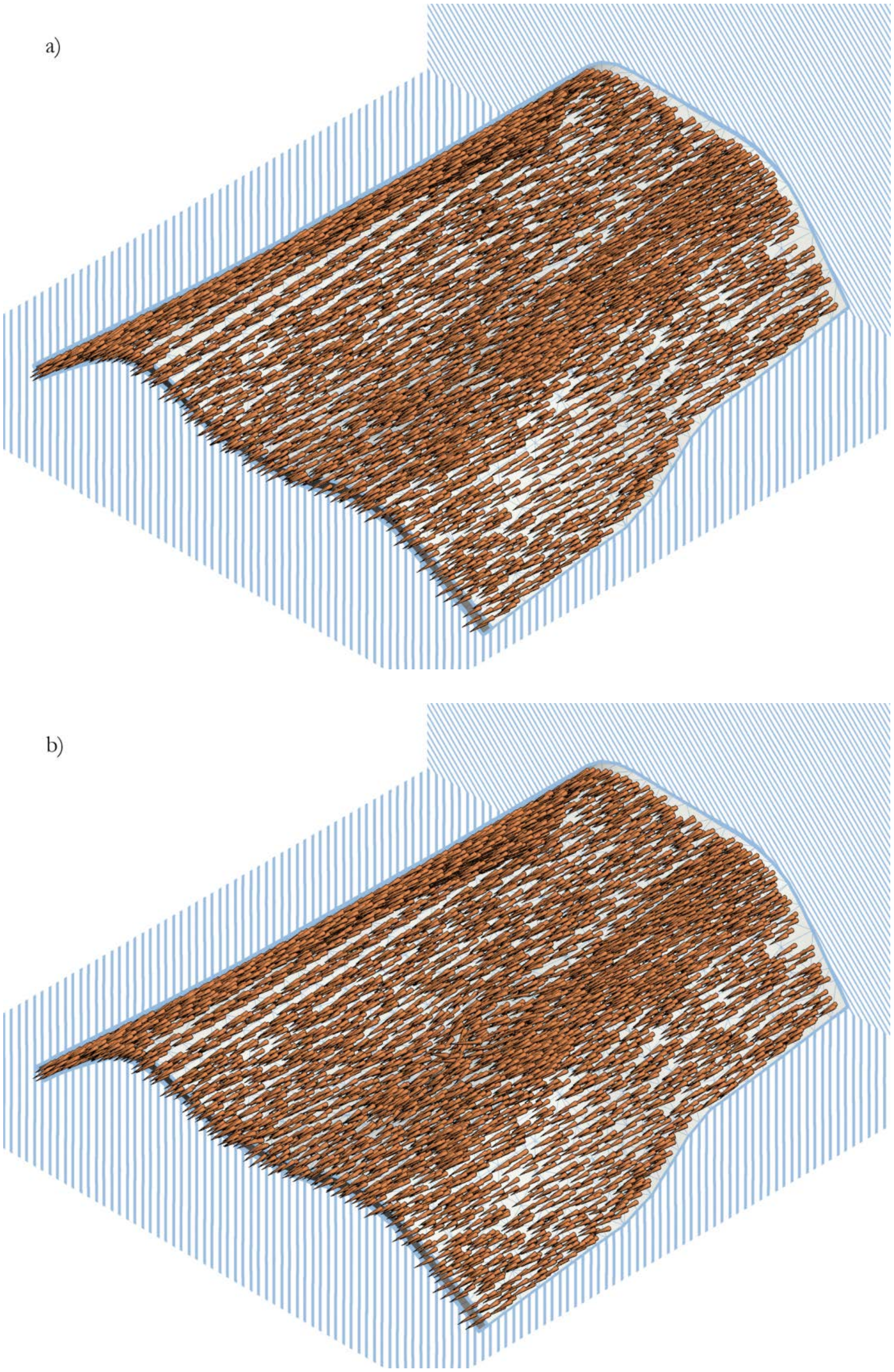


Figure 5-15: Stepped shell, load case 1 – Optimization Snapshots featuring parameter vector overlay Deformation X 10, Vectors Showing Fibre Orientation;
a) 0 seconds, b) 4 seconds.

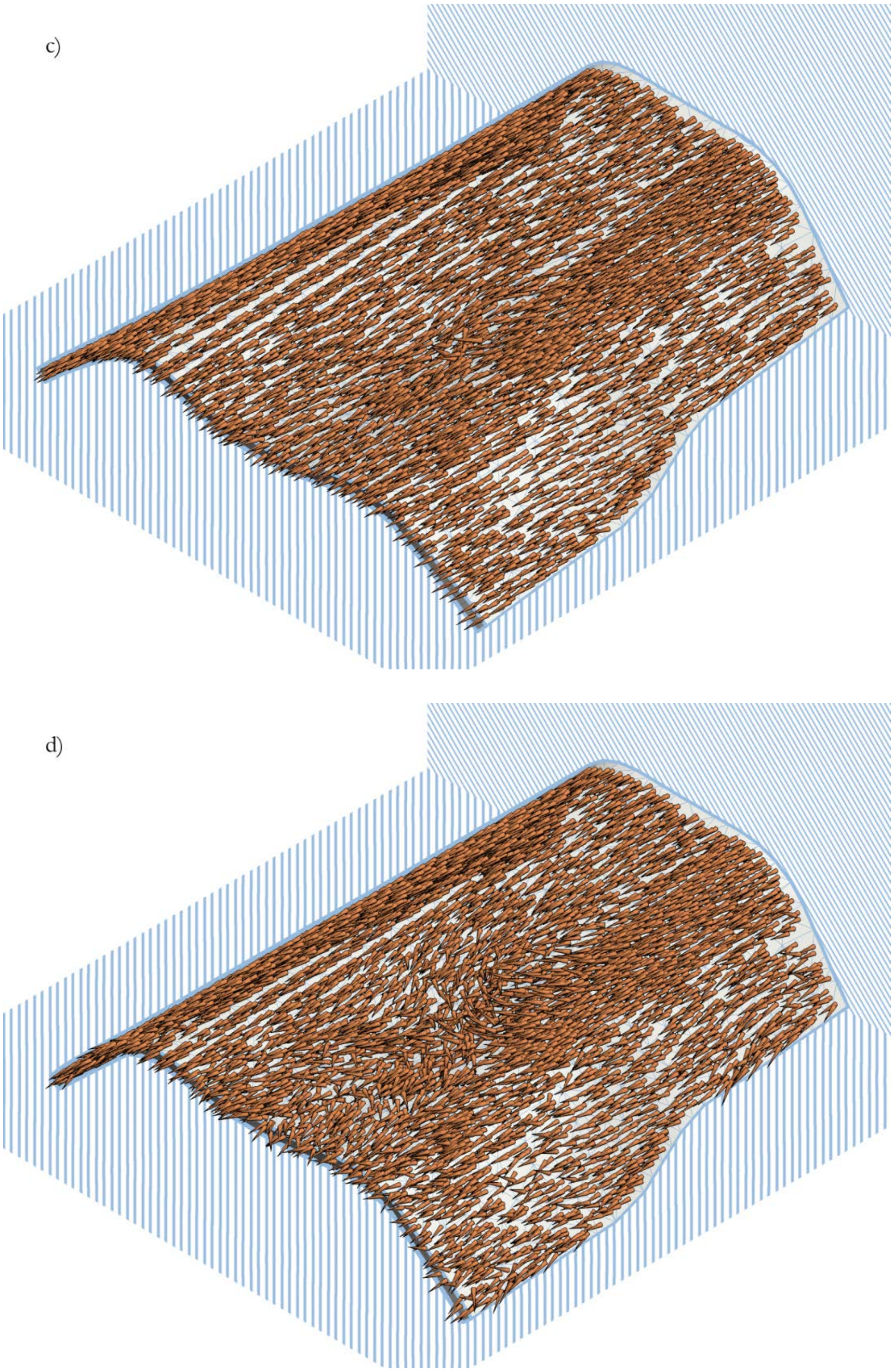
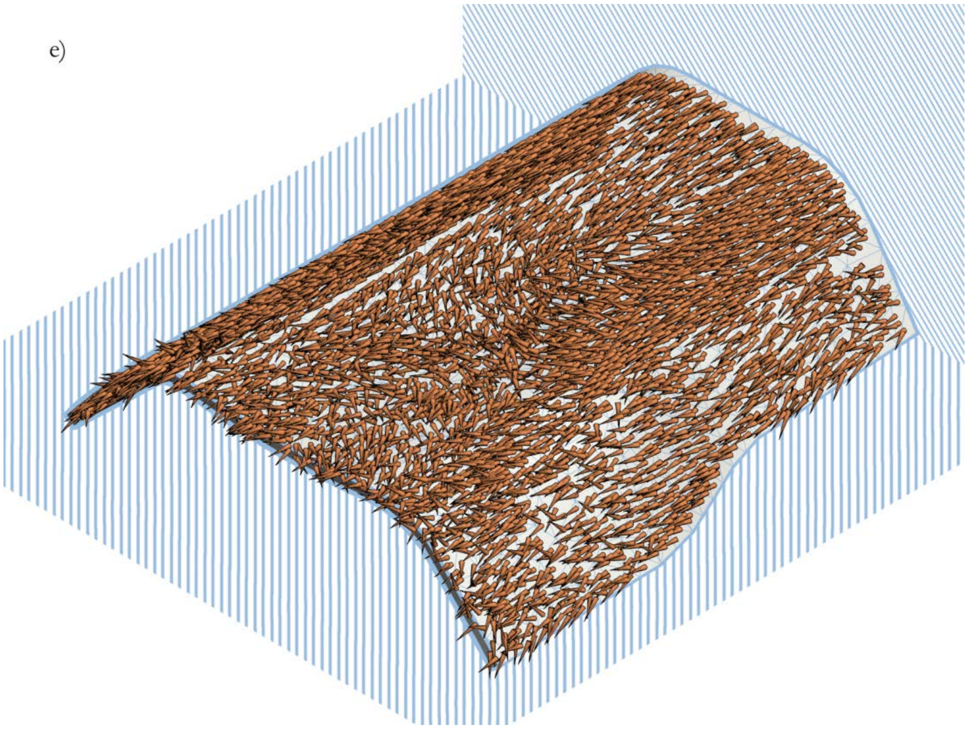


Figure 5-15: Stepped shell, load case 1 – Optimization Snapshots featuring parameter vector overlay Deformation X 10, Vectors Showing Fibre Orientation; c) 41 seconds, d) 416 seconds.

e)



f)

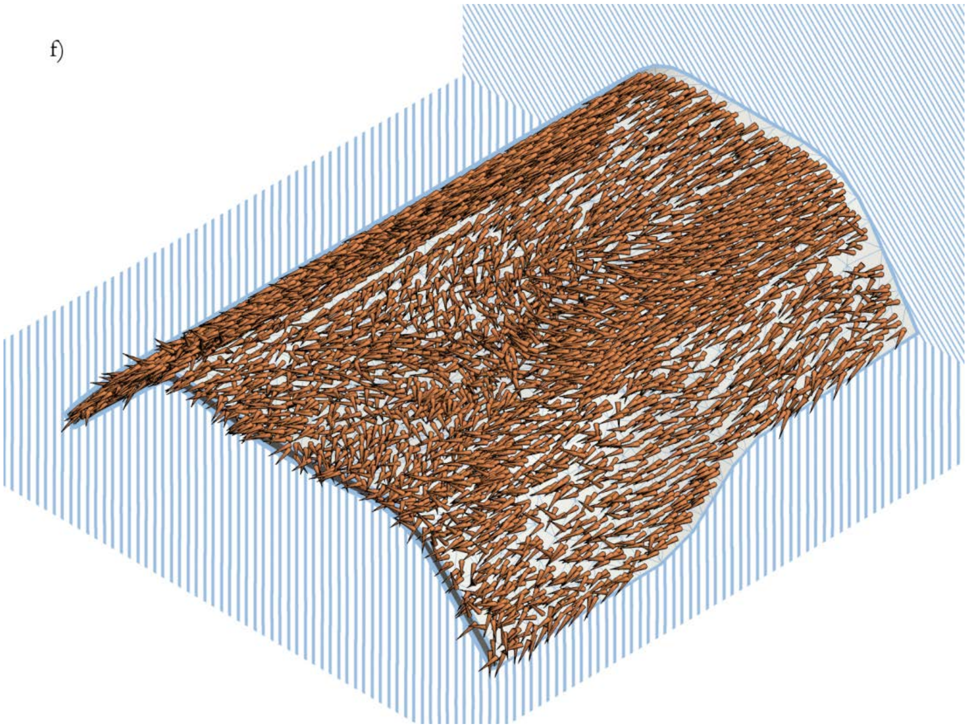


Figure 5-15: Stepped shell, load case 1 – Optimization Snapshots featuring parameter vector overlay Deformation X 10, Vectors Showing Fibre Orientation; e) 4163 seconds, f) 41635 seconds.

Figure 5-16 shows isometric views of the part with vectors representing fibre alignment at each element, at different times during optimization during the second load case. As pictured, in this second case the fibre orientations are initialized from the final result of the first optimization case. Given the additional load and boundary conditions for case 2, the software program reorients the fibres to reduce the weighted sum of displacements at the loaded nodes. The final result shown in Figure 5-16 (f) illustrates that the fibre orientations remain mostly unchanged with the largest changes occurring near the newly constrained end, where fibres become more aligned along the longitudinal direction of the part in order to better counteract the bending load imposed at the other end.

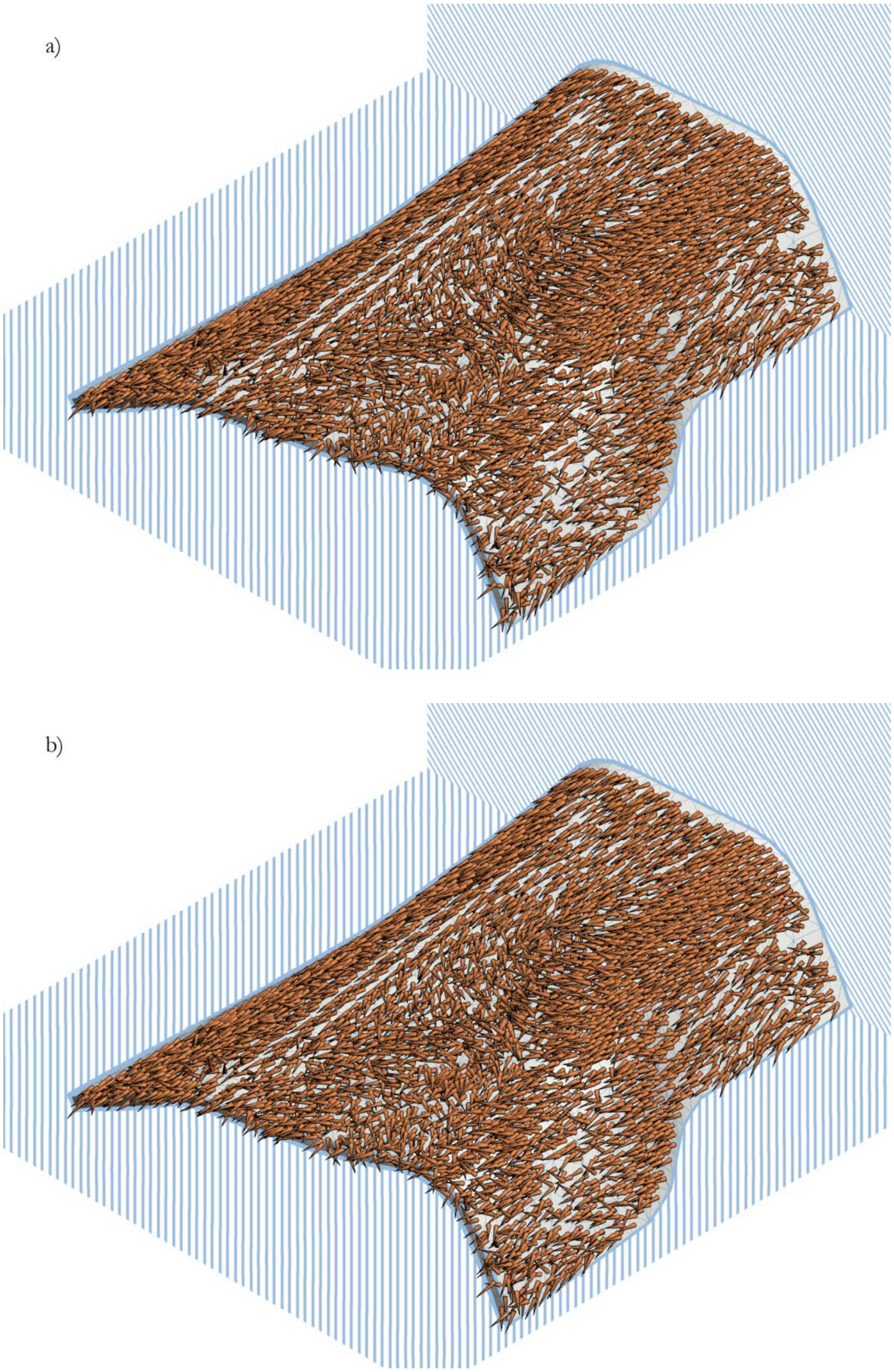


Figure 5-16: Stepped shell, load case 2 – Optimization Snapshots featuring parameter vector overlay Deformation X 10, Vectors Showing Fibre Orientation; a) 0 seconds, b) 4 seconds.

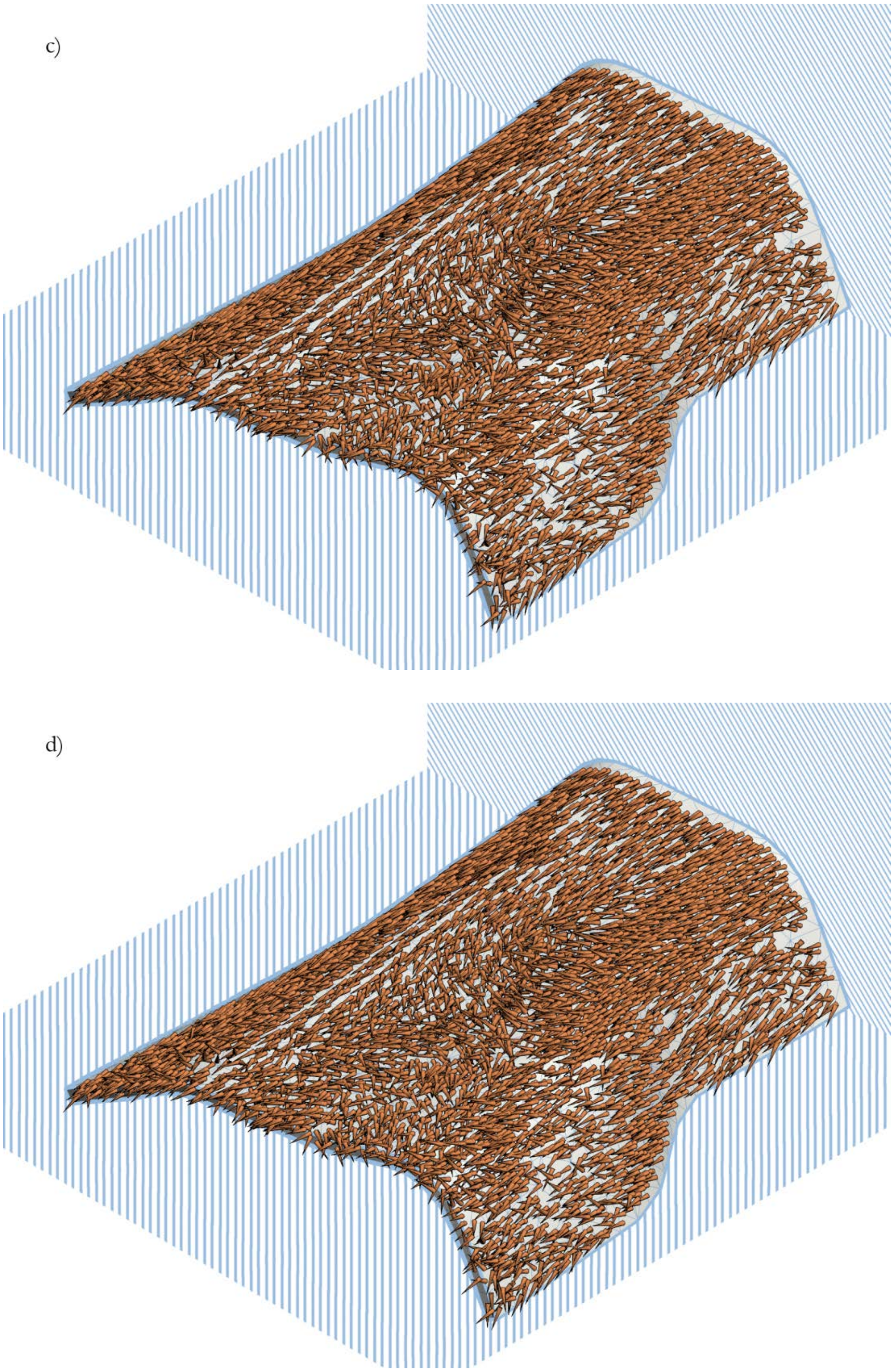


Figure 5-16: Stepped shell, load case 2 – Optimization Snapshots featuring parameter vector overlay
featuring parameter vector overlay
Deformation X 10, Vectors Showing Fibre Orientation; c) 41 seconds, d) 416 seconds.

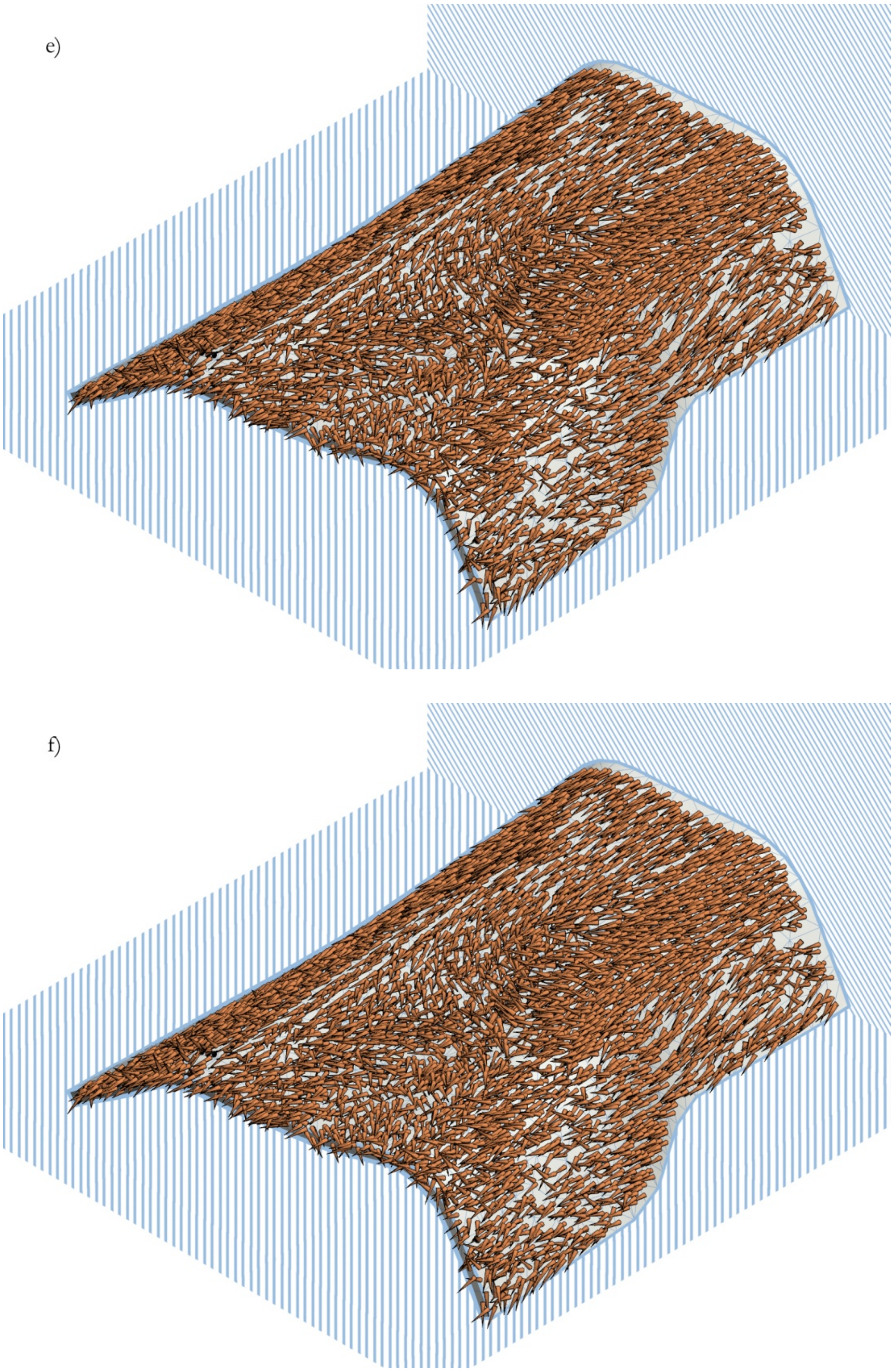


**Figure 5-16: Stepped shell, load case 2 – Optimization Snapshots
featuring parameter vector overlay
featuring parameter vector overlay
Deformation X 10, Vectors Showing Fibre Orientation;
e) 4163 seconds, f) 41635 seconds.**

Chapter 6

Conclusion

6.1. Software Program Improvements

6.1.1 Mesh Size

The scale of problems suited for optimization through the use of the software program is currently limited by memory and computation power. In practice problems of the form described in this work with more than a few thousands of elements are impractically time consuming. Furthermore, the current meshing library used was not coded to generate meshes with more than roughly 40000 elements.

The first issue will be readily solved by exploiting advances in computer technology to make larger problems tractable within reasonable timeframes. Exploitation of parallel computing algorithms and graphical processing unit computation for use in matrix inversion was attempted using ViennaCL's implementation of the open computing language (OpenCL) [**75**]. However, the GPU installed in the current testing hardware was incompatible with double-precision floating point numbers, which made this method much less precise than Eigen's BLAS-based sparse matrix operations [**35**] using AMD's Accelerated Parallel Processing Math Libraries (APPML) [**59**].

6.1.2 Materials

The software program which was developed in the course of this work focuses on general 3D fibre-reinforced materials. Though the program can also model general isotropic and orthotropic materials, specific models for additional classes of materials could be added for applying the program to more design problems.

6.1.3 Material and Manufacturing Considerations

An open-ended approach was taken in the scope of this work regarding the resulting structure of the optimized 3D fibre-reinforced composite parts. The current version of the software program developed is capable of imposing basic manufacturing constraints. Most notably, for simple plates and shell parts the fibre directions specified as optimization variables can be constrained to a given plane, limiting the optimized results to those which are amenable for construction using traditional 2D fibre deposition processes such as hand layup and sprayup, manual prepreg layup, fibre placement, automated tape laying and filament winding [**76**]. In addition, the maximum allowable fibre volume fraction can be set easily to any level in order to reflect manufacturing constraints. On the other hand, the methods described cannot at this time account for cases where sharp changes or divergences in fibre directions would result in geometries where the radius of curvature would be smaller than the minimum radius permissible for the fibres used, or necessitate the use of discontinuous sections spliced together. The current implementation also fails to account for reductions in volume fraction necessarily resulting from the splitting of fibres along two directions.

Given these current limitations, the general approach to composite part optimization could be modified readily to describe existing material structures and processes using the same material models and optimization algorithms but with far fewer independent parameters, and meshes and constraints better suited at describing the orientations throughout composite created using these processes.

Alternatively, the application of a general open-ended optimization routine may inform a designer on a more advantageous reinforcement construction method as of yet undiscovered, provided that sensible constraints are imposed on the optimization problem to reflect the physical reality of the materials being modeled. In effect, the solutions determined by the current software program will serve as a guideline around which both the manufacturing process and the final part design must be married, accounting for the divergences in fibre direction in three ways. First, the formulation of the optimization problem and the implementation of constraints imposed upon the optimization run can ensure that any given solution to this problem yields parts which can be manufactured using currently available processes. Second, after the optimization run, post processing of an optimized design can be performed in order to determine the feasibility of its manufacture and to adjust the solution to retain most of the desirable properties while reflecting practical considerations. Third, given an optimized design, a novel or adapted manufacturing technique could be implemented to achieve its manufacture. Beyond the basic constraints described in this work, more development is needed on this front to realize fully the usefulness of the software program which was developed.

6.1.4 Element Types

While tetrahedral elements are well suited at representing parts with low aspect ratios, high aspect ratio parts such as rods and plates require a prohibitively large amount of elements in order to ensure that element aspect ratios are favourable for accurate calculation. Not only does this increase the time required for calculating the part's response to given loads but it also increases the number of variables in the optimization problem. Because of this, the software developed could be amended to include models for plate and beam elements, enabling the designer to better balance generality and computational efficiency. Though the groundwork for coding such elements is already in place, more work would be needed to implement fully these elements and integrate them with the optimization routines.

6.1.5 Numerical Stability

The solution of the linear system describing the nodal displacements as a function of the forces applied on the mesh as described in Section 2.10 was performed using Eigen's sparse solver, which in turn makes use of iterative methods. By nature, these methods are not exact, but approach the correct solution numerically within a specified precision commensurate with the number of iterations spent on calculation. Because of slight variations in accuracies and roundoff errors from iteration to iteration, the FEA solver introduces a certain amount of noise to the optimization metric. The current implementation of the software program compensates for this noise by estimating the error term and applying a penalty to inaccurate solutions. This could have the effect of slowing or preventing the convergence of certain optimization routines. The sensitivity of these optimization routines to such noise could be established in order to better judge their suitability towards the optimization of material structures, and to direct improvements toward the speedy solution of the finite element model without compromising accuracy or convergence speed.

6.1.6 Additional Optimization Parameters

To truly take advantage of an automated design tool, design parameters describing as many aspects of the final part should be accounted for in the optimization process. The current implementation of the software program currently operates on a static geometry. Additional optimization parameters could include modifications to the geometry of the part being optimized, resulting in a design where both the material and the shape of the part are optimized in unison. Furthermore, a model describing parts consisting of more than one material could be added.

The strength and weight of a part are both critical factors in most mechanical designs. Currently, these factors are not represented in the software program which was developed, as the selection of appropriate failure models for composite materials and their proper application is a complex and controversial endeavour. The ability to add these to the optimization functions and constraints being optimized by the program would greatly broaden its application.

Parameters and optimization metrics reflecting other aspects of a design should also be included in the full design of a part. The dynamic analysis of a part, aiming at accounting for vibrations and shock loading, the aerodynamic properties of a part as well as the aesthetics and economics of any design should also be considered in a complete design, and would be useful additions to the software program. Such holistic optimization schemes, called multidisciplinary design optimization (MDO) are described by Bo and Costin [**77**], Chattopadhyay and McCarthy [**78**], Thanedar, et al. [**79**], Haftka et al. [**80**], Sobieszczanski-Sobieski and Tulinius [**81**], and Venkayya [**82**] in the context of composite material sciences.

6.2. Summary of Work

This work presents a general approach towards the optimization of the stiffness of fibre-reinforced composite parts through the variation of fibre orientations and volume fractions and modeled using finite element analysis. A description of the finite element formulation implemented was given, along with the derivations of the elasticity matrices for isotropic, orthotropic and fibre-reinforced materials. The properties of fibre-reinforced composite materials were described using a vector representation, and then derived with respect to fibre orientation and volume fraction. In turn, the deformation of a fibre-reinforced composite part subjected to a given load was expressed as a function of its material properties and calculated using the finite element method. The derivative of any given deformation with respect to a change in fibre orientation or volume fraction within a subset of the part was calculated. Using these models in conjunction with the NLOpt library, a framework for the minimization of the deformation of a part in response to a given load was created.

This software program was tested using simple test cases in order to ascertain the accuracy of the model. The software was proven to converge to expected results for simple tension and cantilevered beam bending test cases.

Further testing was performed on each optimization algorithm made available through NLOpt to ascertain their convergence speed in the context of the minimization of deformations predicted by an FEA model. Several algorithms proved to be suitable for the optimization of a low-resolution mesh of 48 elements for a beam bending case, converging to within 1% or closer of the global minimum given by longitudinally aligned fibres.

The more successful algorithms were tested using a denser mesh of 381 elements with correspondingly more optimization parameters in order to ascertain the dependence of convergence speed on the number of optimization parameters and FEA degrees of freedom. Unsurprisingly, convergence times increased considerably as the mesh size was augmented. Furthermore, the optimization algorithms which make use of the gradient of the optimization function converge much faster for large problem spaces. Similarly, the algorithms best suited for large problem spaces were again tested with a denser mesh of 2070 elements. Of the two algorithms best suited at large scale optimization, namely VAR2 and LBFGS, the latter was found to be most effective algorithm for the optimization of large problem spaces such as those examined in this thesis, converging to 10% and 25% of the global minimum for the 381 element and 2070 element meshes respectively.

As the limiting stopping criterion for the denser meshes was in both cases the allotted time, namely 3 hours and 24 hours for the 281 element mesh and 2070 element meshes respectively, the LBFGS algorithm may be able to more closely approach the global minimum given more time.

The software program was then put to use in optimizing more complex parts in a more realistic scenario. Using the software program and the LBFGS algorithm, reductions of 48%, 66%, 58% and 32% were achieved in the displacement at the loaded end of the parts examined from a first-guess design, for the four proof of concept cases examined. A discussion of the merits and weaknesses of the current implementation of the software program followed.

In summary, the use of optimization algorithms in conjunction with FEA modeling is a promising approach to aid in the design of complex parts. Such an approach was tested and found to be effective at minimizing displacements given a loading condition and material constraints. The addition of additional constraints, parameters and functions to the optimization problem can allow for a more nuanced design process, and should be considered for additional development of this idea.

Bibliography

[1] S.V. Lomov et al., "Meso-FE modelling of textile composites: Road map, data flow and algorithms," *Composites Science and Technology*, vol. 67, no. 9, pp. 1870-1891, July 2007.

[2] De Carvalho, N.V., S.T. Pinho, and P. Robinson, "Reducing the domain in the mechanical analysis of periodic structures, with application to woven composites," *Composites Science and Technology*, vol. 71, no. 7, pp. 969-979, May 2011.

[3] S.V. Lomov, A.V. Gusakov, G. Huysmans, A. Prodromou, and I. Verpoest, "Textile geometry preprocessor for meso-mechanical models of woven composites," *Composites Science and Technology*, vol. 60, no. 11, pp. 2083-2095, August 2000.

[4] S.V. Lomov, G. Huysmans, and I. Verpoest, "Hierarchy of textile structures and architecture of fabric geometric models," *Textile Research Journal*, vol. 71, no. 6, pp. 534-543, June 2001.

[5] S.V. Lomov et al., "Textile composites: Modelling strategies," *Composites - Part A: Applied Science and Manufacturing*, vol. 32, no. 10, pp. 1379-1394, October 2001.

[6] F Robitaille, B.R Clayton, A.C Long, B.J Souter, and C.D. Rudd, "Geometric modelling of

industrial preforms: Woven and braided textiles," *Proceedings of the Institution of Mechanical Engineers Part L: Journal of Materials: Design and Applications*, vol. 213, no. 2, pp. 69-83, 1999.

[7] F Robitaille, B.R Clayton, A.C Long, B.J Souter, and C.D Rudd, "Geometric modelling of industrial preforms: Warp-knitted textiles," *Proceedings of the Institution of Mechanical Engineers Part L: Journal of Materials: Design and Applications*, vol. 214, no. 2, pp. 71-89, 2000.

[8] C. Qian, L.T. Harper, T.A. Turner, S. Li, and N.A. Warrior, "Establishing size effects in discontinuous fibre composites using 2D finite element analysis," *Computational Materials Science*, vol. 64, pp. 106-111, November 2012.

[9] Ping Tan, Liyong Tong*, and G.P. Steven, "Behavior of 3D orthogonal woven CFRP composites. Part II. FEA and analytical modeling approaches," *Composites Part A: Applied Science and Manufacturing*, vol. 31, no. 3, pp. 273-281, March 2000.

[10] R.S Thomson, P.J Falzon, A Nicolaidis, K.H Leong, and T Ishikawa, "The bending properties of integrally woven and unidirectional prepreg T-sections," in *10th International Conference on Composite Structures*, Melbourne, 1999, pp. 781-787.

[11] N. Tolosana, M. Carrera, R.G. De Villoria, L. Castejon, and A. Miravete, "Numerical analysis of three-dimensional braided composite by means of geometrical modeling based on machine emulation," *Mechanics of Advanced Materials and Structures*, vol. 19, no. 1-3, pp. 207-215, January 2012.

[12] S.V. Lomov, G. Perie, D.S. Ivanov, I. Verpoest, and D. Marsal, "Modeling three-dimensional fabrics and three-dimensional reinforced composites: Challenges and solutions," *Textile Research Journal*, vol. 81, no. 1, pp. 28-41, January 2011.

[13] G. Fang and J. Liang, "A review of numerical modeling of three-dimensional braided textile composites," *Journal of Composite Materials*, vol. 45, no. 23, pp. 2415-2436, November 2011.

[14] Daniel B. Miracle and Steven L. Donaldson, "Introduction to Composites," in *ASM Handbook, Volume 21, Composites*.: ASM International, 2001, vol. 21, ch. 1, pp. 39-66.

[15] D Hull and T. W. Clyne, "General Introduction," in *An Introduction to Composite Materials*. Cambridge, United States of America: Cambridge University Press, 1996, ch. 1, pp. 1 - 2.

[16] D Hull and T. W. Clyne, "Strength of Composites," in *An Introduction to Composite Materials*. Cambridge, United States of America: Cambridge University Press, 1996, ch. 8, pp. 158-207.

[17] P. K. Mallick, "Mechanics," in *Fiber-Reinforced Composites*, L.L. Falkner and S. B. Menkes, Eds. New York, United States of America: Marcel Dekker, Inc., 1988, ch. 3, pp. 73-167.

[18] L. T. Harper, T. A. Turner, R. B. Martin, and N. A. Warrior, "Fibre Alignment in Directed Carbon Fibre Preforms - A Feasibility Study," *Journal of Composite Materials*, vol. 43, no. 1, pp. 57-72, January 2009.

[19] David W. Green, Jerrold E. Winandy, and David E. Kretschmann, "Orthotropic Nature of Wood," in *Wood Handbook: wood as an engineering material*. Almonte, Canada: Algrove Publishing Limited, 2002, ch. 4, pp. 63 - 64.

[20] Richard Pallardy and Shiveta Singh, "Xylem," *Encyclopædia Britannica. Encyclopædia Britannica Online*, pp. http://www.britannica.com/EBchecked/topic/650951/xylem, February 2013.

[21] Roberta Farrell. (2013, April) The University of Waikato. [Online]. http://sci.waikato.ac.nz/farm/content/plantstructure.html

[22] Susan Standring, "Bone," in *Gray's Anatomy, 39th Edition*. Philadelphia, USA: Elsevier Churchill Livingstone, 2005, ch. 6, pp. 87 - 103.

[23] Stephen, SW Composites Ward and Lawrence, Concurrent Technologies Corporation Gintert, "Analysis of Sandwich Structures, Introduction," in *ASM Handbook, Volume 21, Composites*.: ASM International, 2001, ch. 3, p. 749.

[24] Jim Kindinger and Hexcel Composites, "Lightweight Structural Cores, Sandwich Structures," in *ASM Handbook, Volume 21, Composites*.: ASM International, 2001, ch. 2, p. 458.

[25] J. Jayasinghe, S. Jones, and A. Boyde, "Three-dimensional photographic study of cancellous bone in human fourth lumbar vertebral bodies," *Anatomy and Embryology*, vol. 189, no. 3, pp. 259-274, March 1994.

[26] Hobdell M. H. and A. Boyd, "Microradiography and Scanning Electron Microscopy of

Bone Sections," *Zeitschrift für Zellforschung und Mikroskopische Anatomie,* vol. 94, no. 4, pp. 487-494, October 1968.

[27] Nicholas J., Visteon Corporation Gianaris, "Introduction to Recycling and Disposal of," in *ASM Handbook, Volume 21, Composites*.: ASM International Handbook, 2001, p. 2297.

[28] Michel Labrosse, "Interpolation functions - Integration formulas," in *MCG 4102/5108 Finite Element Analysis*. Ottawa, Canada: Department of Mechanical Engineering, University of Ottawa, 2012, pp. 33-45.

[29] Carlos A. Felippa, "The Linear Tetrahedron," in *Advanced Finite Elements for Solids, Plates and Shells*. Colorado, United States of America: Department of Aerospace Engineering Sciences, University of Colorado at Boulder, 2011, ch. 16, pp. 1-22.

[30] Rong Tian and Genki Yagawa, "Generalized nodes and high-performance elements," *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING,* no. 64, pp. 2039-2071, August 2005.

[31] Xiao Hua and Cho W Solomon To, "Simple and efficient tetrahedral finite elements with rotational degrees of freedom for solid modeling," *Journal of Computing and Information Science in Engineering,* vol. 7, no. 4, pp. 382-393, December 2007.

[32] Rong Tian, Hitoshi Matsubara, and Genki Yagawa, "Advanced 4-node tetrahedrons," *International Journal for Numerical Methods in Engineering,* vol. 68, no. 12, pp. 1209-1231, December 2006.

[33] Hang Si. (2011, January) A Quality Tetrahedral Mesh Generator and a 3D Delaunay

Triangulator. [Online]. http://wias-berlin.de/software/tetgen/files/tetgen-manual.pdf

[34] Gaël Guennebaud and Benoît Jacob. (2013, January) Eigen v3. [Online]. http://eigen.tuxfamily.org

[35] Gaël Guennebaud and Benoît Jacob. (2013, January) Eigen v3. [Online]. http://eigen.tuxfamily.org/dox/TutorialSparse.html

[36] Jacob Fish and Ted Belytshchko, "Introduction," in *A First Course in Finite Elements*. West Sussex, England: John Wiley & Sons Ltd, 2007, ch. 1, p. 1.

[37] Erwin Kreyszig, "Change of Variables in Double Integrals. Jacobian," in *Advanced Engineering Mathemetics*. Hoboken, United States of America: John Wiley & Sons, 2006, ch. 10.3, pp. 436-438.

[38] Eric W. Weisstein. (2013, February) MathWorld--A Wolfram Web Resource. [Online]. http://mathworld.wolfram.com/Jacobian.html

[39] Jacob Fish and Ted Belytschko, "Approximation of Trial Solutions, Weight Functions and Gauss Quadrature for One-Dimensional Problems," in *A First Course in Finite Elements*. Chichester, England: John Wiley & Sons, 2007, ch. 4, pp. 77-92.

[40] Jacob Fish and Ted Belytschko, "Approximations of Trial Solutions, Weight FUnctions and Gauss Quadrature for Multidimensional Problems," in *A First Course in Finite Elements*. Chichester, England: John Wiley & Sons, 2007, ch. 7, pp. 151-186.

[41] Erwin Kreyszig, "Interpolation," in *Advanced Engineering Mathematics*. Hoboken, United

States of America: John Wiley & Sons, 2006, ch. 19.3, pp. 797-808.

[42] Branden Archer and Eric W. Weissten. (2013, Februray) Mathworld. [Online]. http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html

[43] W. Rachowicz, D. Pardo, and L. Demkowicz, "Fully automatic hp-adaptivity in three dimensions," *Computer methods in applied mechanics and engineering*, no. 195, pp. 4816-4842, August 2005.

[44] Ivo M. Babuska and W.C. Rheinboldt, "Adaptive Approaches and Reliability Estimations in Finite Element Analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 17-18, no. 3, pp. 519-540, March 1979.

[45] Guangming Zhou, Xuekun Sun, and Youqi Wang, "Multi-chain digital element analysis in textile mechanics," *Composites Science and Technology*, vol. 64, no. 2, pp. 239-244, February 2004.

[46] Michael A. Slawinski, "Rotation tensor and rotation vector," in *Waves and Rays in Elastic Continua*. Singapore, Singapore: World Scientific Publishing, 2007, ch. 1.5, pp. 25-26.

[47] Michael A. Slawinski, "Formulation of stress-strain equations: Hookean solid," in *Waves and Rays in Elastic Continua*. Singapore, Singapore: World Scientific Publishing, 2007, ch. 3.2, pp. 65-70.

[48] Michael A. Slawinski, "Orthogonal transformations," in *Waves and Rays in Elastic Continua*, World Scientific Publishing, Ed. Singapore, Singapore, 2007, ch. 5.1, pp. 93-94.

[49] Michael A. Slawinski, "Transformation of coordinates," in *Waves and Rays in Elastic Continua*. Singapore, Singapore: World Scientific Publishing, 2007, ch. 5.2, pp. 94-99.

[50] Michael A. Slawinski, "Condition for material symmetry," in *Waves and Rays in Elastic Continua*. Singapore, Singapore: World Scientific Publishing, 2007, ch. 5.3, pp. 100-102.

[51] Hyung Joo Kim and Colby C. Swan, "Voxel-based meshing and unit-cell analysis of textile composites," *International Journal for Numerical Methods in Engineering*, vol. 56, no. 7, pp. 977-1006, February 2003.

[52] Serge Belongie. (2013, February) Mathworld. [Online]. http://mathworld.wolfram.com/RodriguesRotationFormula.html

[53] Hoebergen and Arlen, "Vacuum Infusion," in *Committee, ASM International Handbook*.: ASM International, 2001, ch. 4, pp. 1184-1120.

[54] Stephen W. Tsai and H. Thomas Hahn, "Micromechanics," in *Introduction to Composite Materials*. Westport, United States of America: Technomic Publishing Co., Inc., 1980, ch. 9, pp. 379-433.

[55] Michel Labrosse, "Linear elasticity - Principle of virtual work," in *MCG 4102/5108 Finite Element Analysis*. Ottawa, Canada: University of Ottawa, 2012, ch. 2, pp. 13-19.

[56] David C. Lay, "Les factorisations de matrices," in *Algèbre Linéaire, Théorie Exercies & Applications, 3e edition*, Micheline Citta-Vantesmsche, Ed. Bruxelles, Belgium: De Boeck, 2004, ch. 2.5, pp. 134-142.

[57] Gaël Guennebaud and Benoît Jacob. (2013, Februrary) Eigen v3. [Online]. http://eigen.tuxfamily.org/dox/TutorialSparse.html

[58] Kazushige Goto. (2011, April) GotoBLAS2. [Online]. http://www.tacc.utexas.edu/tacc-projects/gotoblas2

[59] Advanced Micro Devices Corporation. (2013, April) Accelerated Parallel Processing Math Libraries (APPML). [Online]. http://developer.amd.com/tools/heterogeneous-computing/amd-accelerated-parallel-processing-math-libraries/

[60] A. Griewank A. Walther, "Getting started with ADOL-C," *Combinatorial Scientific Computing* , pp. 181-202, 2012. [Online]. https://projects.coin-or.org/ADOL-C

[61] Steven G. Johnson, "Notes on Adjoint Methods for 18.335," in *Introduction to Numerical Methods*. Massachusetts, United States of America: MIT, 2012.

[62] E. P. Popov, "Beam Deflections by Direct Integration," in *Engineering Mechanics of Solids*, Marcia Horton, Ed. Upper Saddle River, United States of America: Prentice Hall, Inc., 1998, ch. 14, pp. 582-624.

[63] Steven G. Johnson. (2013, January) The NLopt nonlinear-optimization package. [Online]. http://ab-initio.mit.edu/nlopt

[64] Eric W. Weisstein. (2013, March) MathWorld--A Wolfram Web Resource. [Online]. http://mathworld.wolfram.com/NewtonsMethod.html

[65] Jonathan Richard Shewchuk, *An Introduction to the Conjugate Gradient Method Without*

the Agonizing Pain, 1st ed. Pittsburgh, United States of America: School of Computer Science, Carnegie Mellon University, 1994.

[66] Krister Svanberg, "A Class pf Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 555–573, January 2002.

[67] Dieter Kraft, "Algorithm 733: TOMP - Fortran Modules for Optimal Control Calculations," *ACM Transactions on Mathematical Software*, vol. 20, no. 3, pp. 262-281, September 1994.

[68] D.C., Nocedal, J. Liu, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 503-528, August 1989.

[69] Dembo Ron S. and Trond Streihaug, "Truncated-Newton Algorithms for Large-Scale Unconstrained Optimization," *Mathematical Programming*, vol. 26, no. 2, pp. 190-212, June 1983.

[70] J. Vlček and L Lukšan, "Shifted limited-memory variable metric methods for large-scale unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 186, no. 2, pp. 365-390, February 2006.

[71] M.J.D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in optimization and numerical analysis*, Oaxaca, 1994, pp. 51-67.

[72] M.J.D. Powell, "The BOBYQA algorithm for bound constrained optimization without

derivatives," Cambridge University, Cambridge, Report NA2009/06, 2009.

[73] R. P. Brent, *Algorithms for Minimization without Derivatives*. Toronto, Ontario: General Publishing Company, Ltd., 2002.

[74] Thomas Harvey Rowan, "Functional Stability Analysis of Numerical Algorithms," University of Texas at Austin, Austin, PhD Thesis 1990.

[75] Karl Rupp. (2013, April) ViennaCL. [Online]. http://viennacl.sourceforge.net

[76] ASM International Handbook Committee, "Manufacturing Processes," in *ASM Handbook, Volume 21, Composites*.: ASM International, 2001, ch. 4, pp. 1010-1575.

[77] Ping Wang Bo and Daniel Costin, "Optimum design of a composite structure with three types of manufacturing constraints," *AIAA journal*, vol. 30, no. 6, pp. 1667-1669, June 1992.

[78] A. Chattopadhyay and T.R. McCarthy, "Multidisciplinary optimization and design variable sensitivity of helicopter rotor blades using a composite box beam model," *Composites Engineering*, vol. 3, no. 7-8, pp. 585-599, 1993.

[79] P.B. Thanedar, J.S. Arora, G.Y. Li, and T.C. Lin, "Robustness, generality and efficiency of optimization algorithms for practical applications," *Structural Optimization*, vol. 2, no. 4, pp. 203-212, December 1990.

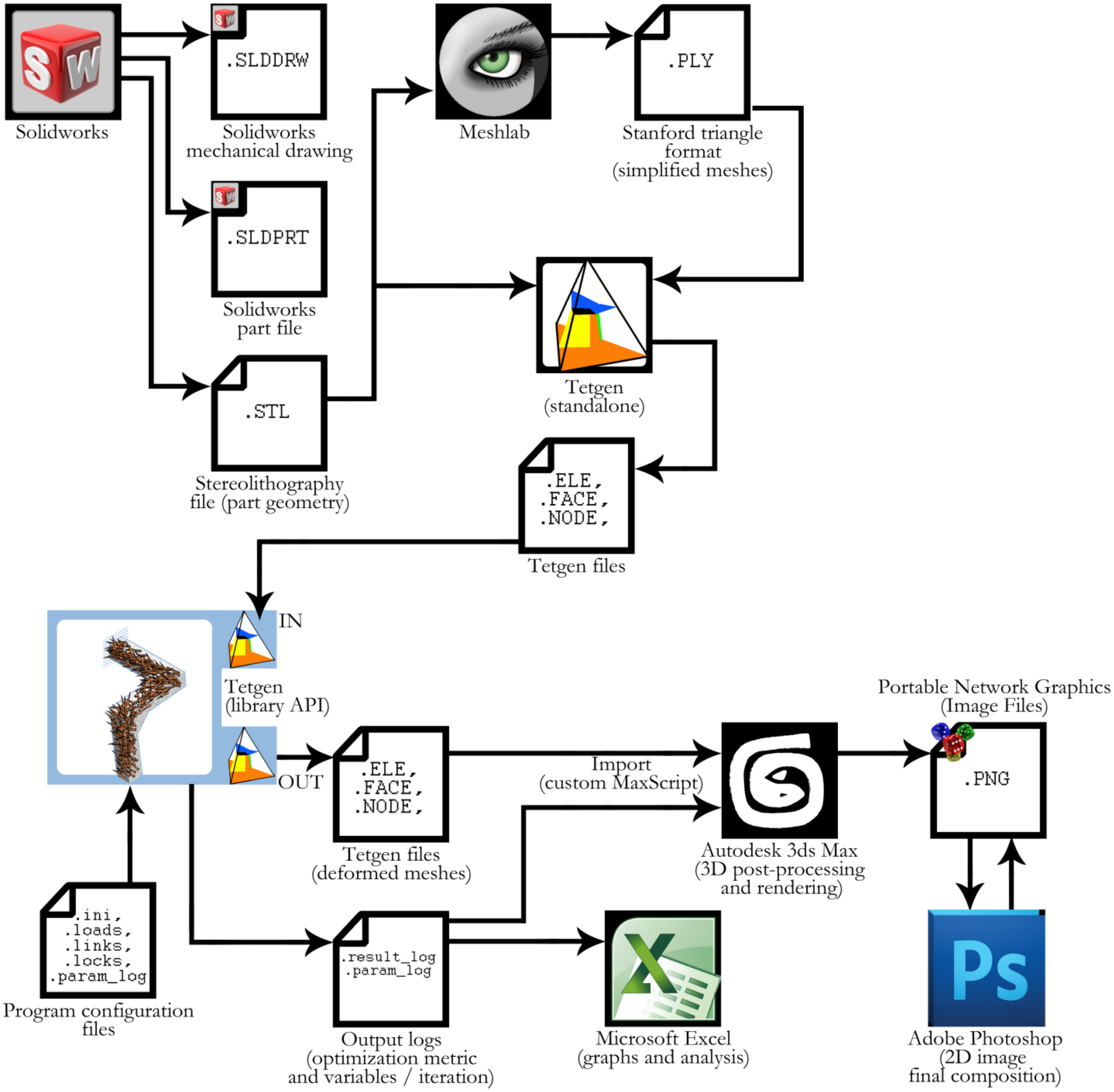
[80] R.T. Haftka, B. Grossman, W.M. Eppard, P.J. Kao, and D.M. Polen, "Efficient optimization of integrated aerodynamic-structural design," *International Journal for Numerical Methods in Engineering*, vol. 28, no. 3, pp. 593-607, March 1989.

[81] Jaroslaw, Tulinius, Jan Sobieszczanski-Sobieski, "MDO can help resolve the designer's dilemma," *Aerospace America*, vol. 29, no. 9, pp. 32-35, 63, September 1991.

[82] V.B. Venkayya, "Optimality criteria: A basis for multidisciplinary design optimization," *Computational Mechanics*, vol. 5, no. 1, pp. 1-21, January 1989.

Appendix A

Data Flow Diagram



Appendix B

Example of configuration files
(Configuration used for case 2 in Section 5.1.2.3)

The first configuration file contains the general configuration options relevant for any optimization run.

.ini file:

```
#### A config file

### This section deals with the command line interface input/output
[command line interface]
      message = Drape test file loaded!
      cout_precision = 3

### The following lists out the different materials used later on
## The names must match their respective sections
[isotropic material list]
      default_iso_volume_material

[general orthotropic material list]
      default_ortho_volume_material

[classic composite material list]
      default_composite_volume_material

## Special! Uses materials from the classic composite material list
## Identify which classic composite material you want to parametrize
[vector volume from classic composite]
      default_composite_volume_material



### The following describes each material's properties
      [default_iso_volume_material]
            ## Young's modulus - Stiffness (Pa)
            E = 50.0E9
            ## Poisson ratio
            nu = 0.3

      [default_ortho_volume_material]
            ## Young's modulus - Stiffness in the x direction (Pa)
            E_x = 50.0E9
```

```
        ## Young's modulus - Stiffness in the y direction (Pa)
        E_y = 1.5E9
        ## Young's modulus - Stiffness in the z direction (Pa)
        E_z = 50.0E9
        ## Major poisson ratio x
        nu_yz = 0.3
        ## Minor poisson ratio y
        nu_zx = 0.3
        ## Minor poisson ratio z
        nu_xy = 0.3

    [default_composite_volume_material]
        ## Young's modulus - Stiffness in the x direction (Pa)
        E_f = 50.0E9
        ## Young's modulus - Stiffness in the y direction (Pa)
        E_r = 1.5E9
        ## Major poisson ratio
        nu_f = 0.3
        ## Minor poisson ratio
        nu_r = 0.3
        ## Maximum volume fraction
        #max_vf = 0.906
        max_vf = 0.6

        ## Sets whether the material will be parametrized as a vector
        ## (orientation and volume fraction can be changed / get
optimized in sim)
        #as_vector = true
        as_unscaled_vector = true
        ## if as_vector is set to true:
            ## Starting value
            ##(important if it is not overwritten prior to first
iteration)
            vector_x = 0.999
            vector_y = 1.0E-3
            vector_z = 1.0E-3
            ## Register immediately (set to false if this is only a
prototype)
            register_parameter = false

[mesh arguments]
    ## message to be printed before mesh creation
    message_pre = Mesh initialization has begun

    ## message to be printed after mesh creation
    message_post = Mesh initialization complete

    ## quality correponds to the aspect ratio of the tetrahedra
    ## generated by tetgen. Tetgen can produce down to qualities of sqrt(2)
    ## (lower is better). No value, or a value of 0 removes this constraint
    #quality = 1.414214

    ## resolution corresponds to the minimum volume of a tetrahedron
    ## (lower is better) No value, or a value of 0 removes this constraint
    #resolution = 1e-6

    ## the material properties for elements whose properties are not
```

```
defined
      ## in a material properties configuration file (see below)
      default_mp_from = classic composite material list
      default_mp_id = default_composite_volume_material

      ## If this option is set to true, then each tetrahedron assigned
      ## the default material gets its own copy of it
      ## (useful if parameters are optimized independently)
      #clone_materials = true
      ## If clone_materials is set to false, and it is desired that
      ## the sim optimize a parameter or vector globally, register_parameter
      ## must be set to true on that material
      clone_materials = true

      ## mesh creation type corresponds to the way Arachne builds a mesh
      ## options currently supported are:
      ## 0: a brick (size dimensions (in metres) are required)
      ## 1: from a file (a root filename is required)

      ### brick type:
      ###
      #mesh_creation_type = 0
      #     ## brick dimensions (m)
            mesh_size_x = 0.1
            mesh_size_y = 0.01
            mesh_size_z = 0.01

      ### from files
      ###
      mesh_creation_type = 1
            ## file details:
            ## name of the file set being opened
            mesh_filename_root = drape_test_sw.2

            ## stl: stl file
            ## ply: ply file
            ## poly: smesh file
            ## tetmesh: ele, face and node files
            ## more details found in the tetgen manual
            mesh_file_type = tetmesh

[constraint arguments]
      ## constraint load method

      ## for an object clamped and oriented along an axis,
      ## #
      ## otherwise, we can load the constraints from a file on a node-by node
basis
      init_method = 1
            file_name = drape_test.multiload

[load and parameter arguments]
      ## this method applies the load to the first clamped dof
      ## (in the case of constraint init_method = 0, this corresponds
      ## to plain_load_type at the free end of the brick
      #load_init_method = 0
```

```
      ## load magnitude (Newtons)
      ## note: certain NLOpt algorithms have built-in constraints
      ## that require a minimum displacement in order to converge
      ## therefore, a large enough load must be provided
      #load_magnitude = 1.0

      ## this method gets the load from a file
      load_init_method = 1
      load_file_name = drape_test

      ## this method applies the default parameter vector to each parameter
      ## (as specified, for example, in the materials section)
      #param_init_method = 0

      ## otherwise, the starting parameters can be loaded from the last
      ## line of a file (useful for resuming from a saved optimization)
      param_init_method = 1
      param_file_name = drape_test_cont


[optimizer arguments]
      ## message to be printed before mesh creation
      message_pre = Mesh optimization has begun

      ## message to be printed after mesh creation
      message_post = Mesh optimization complete

      ###### Selection of algoritm to use

      #### Local optimization algoritms (storage free)
      ## Method of moving asymptotes
      #algorithm_selection = NLOpt_LD_MMA
      #algorithm_selection = NLOpt_LD_SLSQP

      #### Local optimization algorithms (storage size can be defined)
      algorithm_selection = NLOpt_LD_LBFGS
      #algorithm_selection = NLOpt_LD_TNEWTON_PRECOND_RESTART
      #algorithm_selection = NLOpt_LD_TNEWTON_PRECOND
      #algorithm_selection = NLOpt_LD_TNEWTON_RESTART
      #algorithm_selection = NLOpt_LD_TNEWTON
      #algorithm_selection = NLOpt_LD_VAR1
      #algorithm_selection = NLOpt_LD_VAR2
      ## Vector storage size (MB) -- relies on heuristic if 0 or not defined,
default is 10MB
      #vector_storage_size = 4000

      #### Local optimization algoritms (derivative free, slow)
      #algorithm_selection = NLOpt_LN_COBYLA
      #algorithm_selection = NLOpt_LN_BOBYQA
      #algorithm_selection = NLOpt_LN_PRAXIS
      #algorithm_selection = NLOpt_LN_SBPLX

      #### Global optimization algorithms
      #algorithm_selection = NLOpt_GN_ORIG_DIRECT_L
      #algorithm_selection = NLOpt_GN_DIRECT_L

      #algorithm_selection = NLOpt_G_MLSL_LDS
```

```
	### see NLOpt manual for more on these parameters
	## relative tolerance corresponds to ftol_rel
	relative_tolerance = 1.0E-6

	### see NLOpt manual for more on these parameters
	## absolute tolerance corresponds to ftol_abs
	#absolute_tolerance = 1.0E-3

	## time_limits correspond to maxtime
	time_limit_hours = 24
	time_limit_minutes = 0
	time_limit_seconds = 0

	## iteration_limit corresponds to max_eval
	#iteration_limit = 10000
	#iteration_limit = 1

[sub optimizer arguments]
	#algorithm_selection = NLOpt_LD_LBFGS
	#vector_storage_size = 4000

[output arguments]
	## optimized parameter logging
	## 0: do not log
	## 1: log each iteration
	## 2: log successful iterations
	## 3: log final iteration (does not log after unexpected termination)
	param_logging = 1

	## output filename for the optimized parameters
	param_out_file_name = log/drape_test

	## result logging
	## (as above)
	result_logging = 1
	## output filename for the result data
	result_out_file_name = log/drape_test


	## mesh response logging
	## (as above)
	mesh_response_logging = 1

	## mesh response verbosity
	## 0 : do not log
	## 1 : export node file only
	## 2 : export all tetgen files
	mesh_response_verbosity = 2

	## best for slim bar beam test case
	##  note  that  load  from  drape_test.loads  is  currently  1000,  so  total
deformation per Newton is 25000!
	mesh_response_displacement_scale = 10

	## best for bending slim beam test case
	#mesh_response_displacement_scale = 1E1
```

```
## output filename for the mesh response
resp_out_file_name = log/drape_test
```

Next, if loads are to be specified at specific nodes, the following file contains a set of node indices and the magnitude of the forces or moments at each DoF to apply at each loaded node.

*.loads file:

```
1188 0.0 0.0 1E3 0.0 0.0 0.0
630 0.0 3E3 0.0 0.0 0.0 0.0
```

Next , the following file can be used to set immobility constraints on a node-by-node basis. Each line in this file contains a node index followed by a number of DoFs to fix on that node and a list of DoF.

*.locks file:

```
18 6 0 1 2 3 4 5
19 3 2 3 4
52 6 0 1 2 3 4 5
53 3 2 3 4
59 3 2 3 4
237 3 2 3 4

(…)

80 3 1 3 5
638 3 1 3 5
639 3 1 3 5
702 3 1 3 5
703 3 1 3 5
767 3 1 3 5
768 3 1 3 5
1140 3 1 3 5
1141 3 1 3 5
1162 3 1 3 5
1163 3 1 3 5
1288 3 1 3 5
1289 3 1 3 5
```

```
(…)
```

Next , the following file can be used to set equality constraints on node-DoF combinations. Each line in this file contains 2 node indices followed by a number of DoFs to link together (set to be equal), followed by the list of DoF pairs to link together.

*.links file:

```
630 629 6 0 0 1 1 2 2 3 3 4 4 5 5
```