UNIVERSITY OF SHEFFIELD

Application of layout optimisation to the design of additively manufactured metallic components

by

Christopher James Smith

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Department of Materials Science and Engineering

2016

Abstract

Additive manufacturing ('3D printing') techniques provide engineers with unprecedented design freedoms, opening up the possibility for stronger and lighter component designs. Such designs can be developed using structural topology optimisation, a method that computationally designs minimum mass components. Whilst many methodologies now exist this thesis focuses solely on a method known as layout optimisation where components are designed to meet a static strength target using a minimal amount of material. Although layout optimisation offers several advantages over the more predominant topology optimisation methods in the field such as the ability to optimise for strength rather than stiffness, high computational efficiency and the clarity of solutions, it hasn't received anywhere near as much research attention. And as with all topology optimisation methods there has been very little literature on validating the structural performance of solutions that have been additively manufactured, despite this manufacturing route being widely cited as an enabler of these methods.

A methodology for producing practical and structurally efficient component designs using layout optimisation was developed and applied to several 3-D problems including one real world problem from the Bloodhound land speed record project. Structural performance was assessed though physical load testing of specimens additively manufactured from titanium Ti-6Al-4V using the Electron Beam Melting (EBM) process. Once all the appropriate design and manufacturing considerations were included into the methodology the resulting load test specimens successfully achieved their strength target. Errors in dimensional accuracy and the presence of internal porosity were highlighted from X-Ray Computed Tomography (XCT) and laser scanning. The main source of the dimensional errors was identified experimentally and suitable process parameters that significantly reduce these errors were established. These process parameters are not just relevant to truss structures but to any reasonably complex component designs.

Acknowledgements

First and foremost I must express my gratitude towards Iain Todd and Matthew Gilbert who, brilliant in their own very different ways, complimented each other very well during my PhD research and provided me with invaluable guidance and opportunities throughout.

I am very grateful to the Advanced Metallic Systems Centre for Doctoral Training (CDT), which provided the funding and framework for my PhD and in particular to Claire Hinchliffe and Brad Wynne.

I must also thank my colleagues and friends Fatos Derguti, James Hunt, Everth Hernandez Nava, Lampros Kourtis, Sam Tammas-Williams, Jon Wright, Laura Asensio, Linwei He, Darren Lincoln, Sam Rigby, Rahi Rahbari, Duc Nguyen, Jack Hetherington, Ahmed Babiker, Sam Hawksbee, Johnny Reay.

A big thank you to the technical staff at the University in both the Quarrell Lab (Ian Watts and Kyle Arnold) and the heavy structures lab (Chris Todd and Paul Blackbourn) who have in one way on another contributed towards the experimental work in this thesis.

And finally I must thank my parents, who without their support and encouragement I would never have got this far. For that I dedicate this thesis to them.

Christopher Smith

Contents

List of Figures

List of Tables

Nomenclature

α	Load case number ($\alpha = 1, ..., p$)
δ	Measured curvature of truss member
δ_{max}	Maximum measured curvature of truss member
ϵ	Chapter 8 - Specified gap between nodes
ϵ_i	Virtual displacement of node i ($i = 1, ..., m$)
$\mathbf{B}$	A suitable ($3n \times m$) equilibrium matrix containing direction cosines
σ^+	Limiting tensile stress that can be sustained by the material
σ^-	Limiting compressive stress that can be sustained by the material
A	Measured cross-sectional area of truss member
a	Vector of element cross-sectional areas
A_0	Cross-sectional area of truss member from CAD definition
A_e	Measured error in cross-sectional area of truss member
a_i	Chapter 7 - Local plane ordinates of each vertex
a_i	The cross-sectional area of member i ($i = 1, ..., m$)
a_{min}	The minimum permitted cross-sectional area
$B.D$	Build Direction
b_i	Chapter 7 - Local plane ordinates of each vertex
B_s	Equilibrium matrix for rigid shell structure
C_0	Coordinates of the intersection between the measurement planes and the centre line of the truss members
D	Diameter of truss member
d_i	The vector between the equilibrium node 'e' of the rigid shell structure and the connection node 'j' of element 'i'
d_x	Directional matrix resolving force into the x direction
d_y	Directional matrix resolving force into the y direction

d_z	Directional matrix resolving force into the z direction
E	Chapter 7 - Beam energy density
E	Elastic modulus of the material
E_0	Beam energy density when unmodified by thickness function
E_f	Exponent Factor constant for thickness function
E_{def}	Beam energy density when using default thickness function parameters
f	A vector of applied loads and load cases
f_j^x	The x direction component of the live load applied to node j $(j = 1, ..., n)$
f_j^y	The y direction component of the live load applied to node j $(j = 1, ..., n)$
f_j^z	The z direction component of the live load applied to node j $(j = 1, ..., n)$
f_s	Vector of forces and moments acting on the rigid shell structure
h	Hatch spacing
I	Current set by 1-D thermal model (autocalc)
I_0	Second moment of area of truss member from CAD definition
I_c	Current set my current compensation algorithm
$I_{a,e}$	Measured error in second moment of area of truss member about the a axis
$I_{b,e}$	Measured error in second moment of area of truss member about the b axis
k	Chapter 4 - Constant for current compensation algorithm
k	Joint cost penalty
k_{eff}	The effective length factor
L	Chapter 8 - Length of arms of variable cross-section
L	Length of Hatch line
l	Vector of individual member lengths
L_0	Reference length for current compensation algorithm
l_i	The length of member i $(i = 1, ..., m)$
L_{max}	Chapter 8 - Maximum length of arms of variable cross-section
M_e	Measured mass error relative to CAD definition
M_s	Vector of moments acting on the rigid shell structure
N_x	Stress resultant in the x direction
N_xy	Stress resultant cross product
N_y	Stress resultant in the y direction

P	Beam Power
PRS	Powder Recovery System
q	A vector of member axial forces
q_i	The force in member i $(i = 1, ..., m)$
r	Chapter 4 - Radius of joint
r	Chapter 8 - Half distance between node j and nearest adjacent node
r	Chapter 3 - Fraction of total compressive load acing at a node to be used for stability forces
S	Original beam speed in turning point function
S_f	Speed Factor constant for thickness function
S_i	Updated beam speed determined from turning point function
T	Thickness of overhang
t	Layer thickness
T_f	Thickness Factor constant for thickness function
T_{max}	Maximum thickness the thickness function is active for
u	Vector containing the nodal displacements
u_i	Displacement of node i $(i = 1, ..., m)$
V	Modified beam velocity
V	Total volume of structure
V_0	Original beam velocity
V_∞	Estimate of volume with an optimised truss structure comprising an infinite number of members
V_i	The vector of member i $(i = 1, ..., m)$
V_{max}	Maximum adjusted beam speed determined by thickness function
W	Chapter 8 - Width of arms of variable cross-section
W	Virtual work done by the applied external forces
x_a	x ordinate of node ‘a’
x_b	x ordinate of node ‘b’
x_{lb}	Lower bound for x ordinate
x_{ub}	Upper bound for x ordinate
y_a	y ordinate of node ‘a’

y_b y ordinate of node 'b'

y_{lb} Lower bound for y ordinate

y_{ub} Upper bound for y ordinate

z_a z ordinate of node 'a'

z_b z ordinate of node 'b'

z_{lb} Lower bound for z ordinate

z_{ub} Upper bound for z ordinate

ν_j New position of node j (x,y.z)

ν_j^0 Initial position of node j (x,y,z)

$T_{j_D}^D$ Contains the coefficients $[T^x, T^y, T^z, T^c]$ for ν_{j_D} the nodal positions and constant term $[x, y, z, 1]^T$

AM Additive Manufacturing

AMF Additive Manufacturing Format file

BESO Bi-Directional Evolutionary Structural Optimisation

BREP Boundary-Representation

EBM Electron Beam Melting

ELI Extra Low Interstitial

ESO Evolutionary Structural Optimisation

FDM Fused Deposition Modelling

FE Finite Element

GA Genetic Algorithms

GCMMA Globally Convergent Method of Moving Asymptotes

HIP Hot Isostatic Pressing

KKT Karush-Kuhn-Tucker

LENS Laser Engineered Net Shaping

LENS Selective Laser Melting

LO Layout Optimisation

LP Linear Programming

MBB Messerschmitt-Bölkow-Blohm

MILP Mixed Integer Linear Programming

MMA Method of Moving Asymptotes

NDT Non-Destructive Testing

NURBS Non-Uniform Rational B-Splines

PTFE Polytetrafluoroehylene

SIMP Solid Isotropic Material with Penalisation

SLC Sliced Contour file

SLS Selective Laser Sintering

STL STereoLithography File

UTS Ultimate Tensile Strength

XCT X-Ray Computed Tomography

Chapter 1

Introduction

Additive manufacturing ('3D printing') techniques have matured rapidly in recent years, and are now starting to deliver on their promise of providing engineers with unprecedented design freedoms. One of these freedoms is the ability to produce truly weight optimised structural components with forms previously too complex to manufacture. However, to date there has been a limited range of tools available to engineers wishing to exploit these freedoms, with the result that many components produced using additive manufacturing techniques have had similar forms to those produced using conventional manufacturing methods (e.g. casting or machining).

Structural optimisation, a computational design method that aims to identify optimum forms, could be used to fully exploit the capabilities of additive manufacturing. Although a broad research field in itself with many methodologies actively being researched, this thesis focuses on those methods that determine the optimal topology as well as the shape. Structural topology optimisation aims to place material within a defined design domain so that it is used in the most efficient way. Problems are usually stated mathematically in the form of relationships and constraints relating the specifics of the problem such as applied forces, support points and domain boundaries to material properties such as elasticity, yield stress and density. The optimisation process will then try to minimise or maximise a property of interest which will be defined by the objective function. Typically, this will be to minimise compliance or total volume.

This thesis focuses on a specific topology optimisation method called layout optimisation, which determines the optimum size and position of truss elements to minimise overall mass for a given set of loads. To date this method has received significantly less attention than the minimum compliance formulations used in conjunction with finite element analysis, of which some have been commercialised. Layout optimisation potentially offers several advantages over these more established optimisation methodologies.

The thesis explores the use of additively manufacturing truss topologies produced using layout optimisation with the Electron Beam Melting (EBM) process. EBM is particularly suited for this application as it is capable of manufacturing complex geometries from high performance alloys with strong and predictable mechanical properties. The alloy to be used in this project is Titanium Ti-6Al-4V, an alloy commonly used in the aerospace and automotive industries for lightweight components. Like most other additive manufacturing methods, the efficacy of the EBM process for fabricating structures designed using topology optimisation has been little explored.

1.1 Project aims

This project aims to explore and develop the process of producing weight optimised truss structures designed using layout optimisation that can be immediately manufactured. The efficacy of both layout optimisation and the EBM process for designing and fabricating weight optimised structures respectively will be assessed. The structural performance of fabricated pieces will be assessed through physical load testing. The microstructure quality and dimensional accuracy of specimens made with the EBM process will also be characterised using 3-D scanning techniques to determine their effect (if any) on structural performance. Recommendations can then be made on taking this research forward.

1.2 Thesis outline

Chapter 2 - A review of the literature including a brief history of both the structural optimisation and additive manufacturing research fields up to and including the current state of the art.

Chapter 3 - Describes the 3-D layout optimisation formulation with multiple load cases used to design all specimens tested in this thesis. The formulation is applied to two benchmark problems (a 2-D cantilever and the Messerschmitt-Bölkow-Blohm (MBB) 2-D beam problem), both of which have known analytical solutions to demonstrate its accuracy.

Chapter 4 - Describes the process of transforming the result from layout optimisation into a solid model CAD definition that can be additively manufactured. Both the EBM hardware and its process algorithms are described in detail.

Chapter 5 - First case study: Describes the fabrication and physical load testing of the layout optimisation solution to a problem adapted from the cantilever benchmark problem. The internal defects of the fabricated pieces are characterised through X-Ray Computed Tomography (XCT)

Chapter 6 - Creation of an optimisation work-flow that addresses the issues identified from the first case study. The work-flow is applied to a 3-D problem adapted from the 2-D MBB problem and also a real-world problem in the form of an air-brake hinge provided by the Bloodhound Super-Sonic Car (SSC) team.

Chapter 7 - Characterisation of the dimensional accuracy of truss structures produced using the EBM process. Process modifications are made which significantly improve dimensional accuracy of the EBM process.

Chapter 8 - Overall discussion of the findings in this thesis, identifying the issues and suggesting possible solutions.

Chapter 9 - Concluding Remarks

Chapter 10 - Future Work

1.3 Publications

The work detailed in this thesis has been published, or is in consideration for publication in the following peer-reviewed journal papers and conference proceedings, and is reproduced herein:

- Smith, C J., Gilbert, M., Todd, I., & Derguti, F. (2016), 'Application of layout optimization to the design of additively manufactured metallic components', Structural and Multidisciplinary Optimization

- Smith C J., Derguti, F., Hernandez Nava, E., Thomas, M., Tammas-Williams, S., Guliziad, S., Fraser, D., & Todd, I. (2016) 'Dimensional accuracy of Electron Beam Melting (EBM) Additive Manufacture with regard to weight optimized truss structures', Journal of Materials Processing Technology

- Smith, C J., Gilbert, M., Todd, I. (2013), 'Utilizing additive manufacturing techniques to fabricate weight optimized components designed using structural optimization methods', 24th Annual Solid Freeform Fabrication Symposium (pp 879-894). Austin, Texas

Chapter 2

Literature review

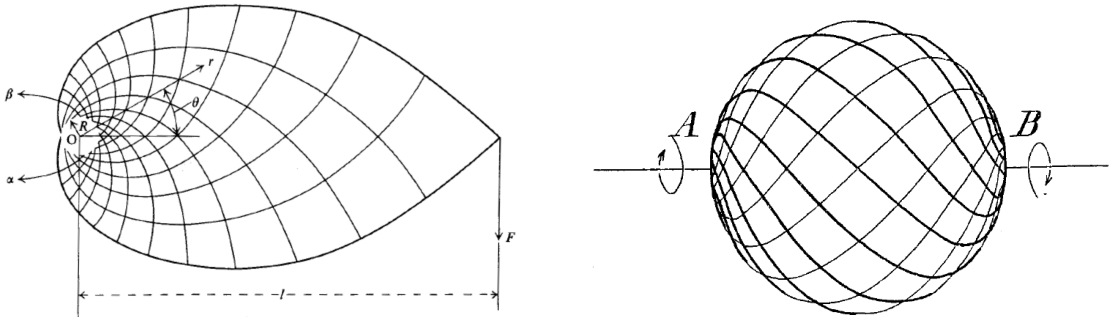
Literature relating to the structural optimisation and additive manufacturing research fields are reviewed separately in sections 2.1 and 2.2.

2.1 Structural optimisation

2.1.1 Origins & analytical solutions

Structural optimisation can be thought of as a means of determining the most efficient structure that transmits an applied force to supports. Typically, this will involve identifying the form that requires the least material to resist the predefined applied load(s) without exceeding the maximum allowable stress of the material. Traditionally engineers would use their experience and intuition to determine this through an iterative process of analysis and review. But a landmark study by Michell (1904) sought to identify the forms and volumes for truly weight optimised structures. Mitchell provided solutions for several simple problems (two of which are shown in Fig. 2.1). These continua-like truss structure are actually made up of an infinite number of joints with infinitesimally small members. The members are aligned with the principal compressive and tensile components in a strain field, which would be analogous to the slip lines in a two dimensional solid subject to plastic deformation. In fact the Hencky nets used to describe slips lines for perfect plastic flow can also be used to describe the form of Mitchell structures (Strang and Kohn, 1983). A key characteristic of these forms is that the compressive and tensile members are orthogonal to one another (as can be observed in Fig. 2.1) and thus do not counteract one another. Michell's criteria states that a framework of minimum volume will have all tensile and compressive members at their maximum respective allowable stresses. This can also be stated as: the virtual work associated with the applied force(s) on the frame should be at a maximum value, with all members within the allowable tensile and compressive strain limits for the material used.

Since then many authors have provided the solutions to similar problems, albeit several decades later. Prager (1977) for instance included a joint weight so that more practical layouts with a finite number of joints would result. Hemp (1973) derived the solution to another cantilever problem similar to that of Michell (1904) but with just two support points which is now often referred to as the Hemp cantilever in the literature. The solution to the Messerschmitt-Bölkow-Blohm (MBB) beam problem is shown in Fig. 2.3 (Lewiński et al., 1994). The solution to a cantilever problem similar to the Hemp cantilever but with



(a) (b)

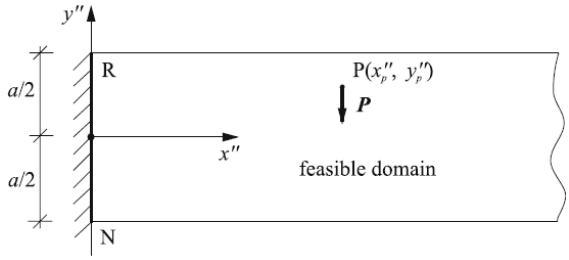
Figure 2.1: (a) Michell Cantilever and (b) Michell Sphere. (Michell, 1904)

a half strip domain is shown in Fig. 2.2. Graczykowski and Lewiński (2010) list solutions for different positions of the applied load P and ratios of tensile to compressive limiting stresses.

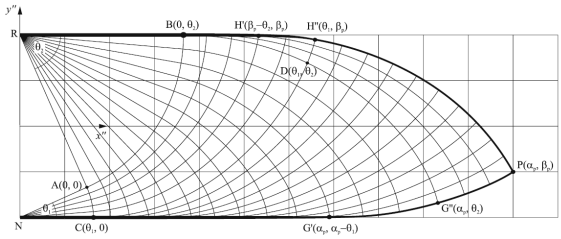
2.1.2 Numerical optimisation

Whilst analytical methods can provide exact solutions, the range of problem types that can be solved is very limited. To date many of these solutions are for simple 2-D problems. For more complicated problems much of the research effort has focused on numerical methods that iteratively determine an optimal solution based on a particular discretisation of the problem. Initial numerical methods through the 1950's could be classed as sizing problems, where the topology of discrete members would be defined *a priori* and the optimisation would be to size the members. Typically, these members would be either plates, shells or trusses. An example of the latter is shown in Fig. 2.4a. However, in 1964 a method was proposed that not only determined the size of truss elements but also their location (Dorn et al., 1964). In this sense the method could be described as a topology or layout optimisation method. A network of truss elements is defined at the beginning of the optimisation (much like that in Fig. 2.4a) which is referred to as the ground structure. The member cross-sectional areas are then sized much like the preceding methods in the literature, except that now a value of zero for the area can be taken. So in the case of Fig. 2.4a the non-bold members would likely have zero area and thus not be present in the final layout. Thus, the members defined in the ground structure are referred to as potential members, and it is likely that many of them will not form a part of the final solution. Because the problem is now discretised there will inevitably be a margin of error between the numerical solution and the exact solution (such as those determined analytically in Sec. 2.1.1). However, as might be expected, use of an increased number of members in the ground structure should result in a solution that is closer to the true optimum.

An extension to earlier methods, many of which were focused on using sizing design variables, was to instead use design variables that defined the shape of the boundaries between void and material. Based on the finite element (FE) method, the boundaries of a topology with a prescribed number of holes could be reshaped into a more optimum form. This method is referred to as structural shape optimisation, and is illustrated in Fig. 2.4b. How the material boundaries are defined will influence the quality of the result. Initially the positions of the nodes on the boundary were chosen as design variables but this would often lead to jagged boundaries which were not only undesirable from a design perspective

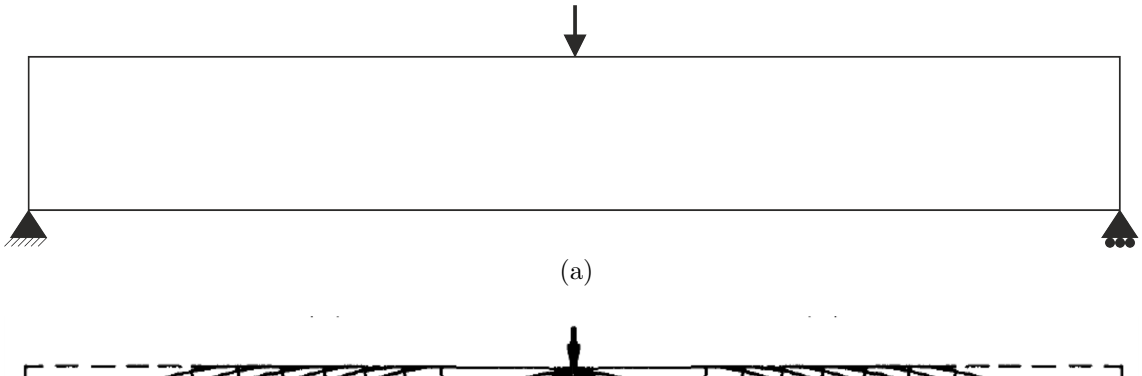


(a)

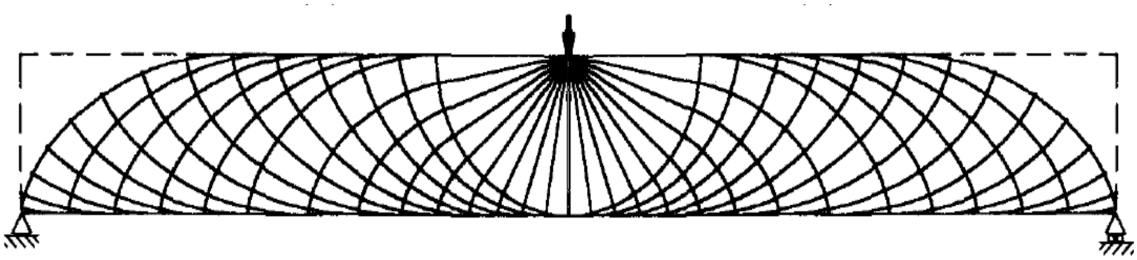


(b)

Figure 2.2: (a) Half strip Michell cantilever problem definition and (b) the solution for a specific location of force P and a ratio of tensile to compressive strength of five (Graczykowski and Lewiński, 2010)



(a)



(b)

Figure 2.3: (a) Beam problem definition and (b) Analytical solution (adapted from Lewiński et al. (1994))

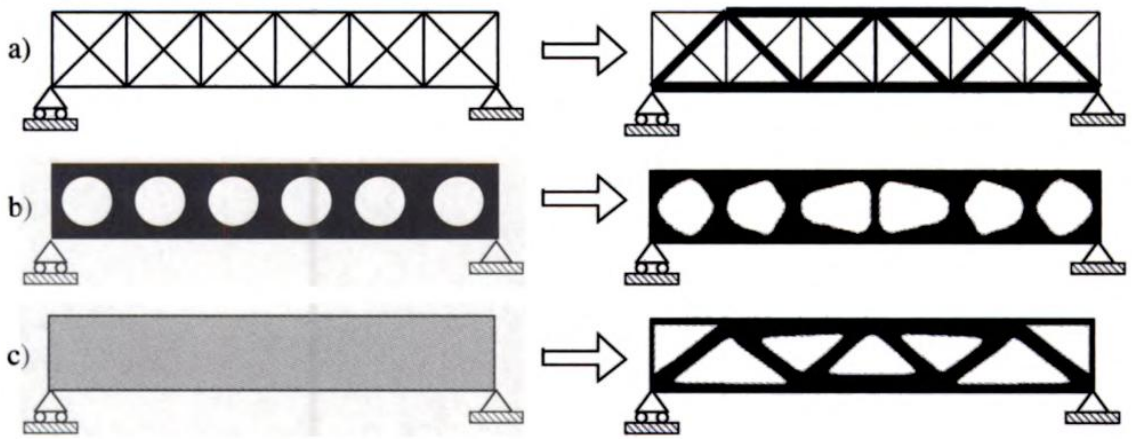


Figure 2.4: Three categories of optimisation: (a) Sizing (b) Shape (c) Topology. Adapted from Sigmund (2000)

but also affected the quality of mesh and hence the accuracy of the result. To attain smooth boundaries early efforts focussed on defining the boundaries using polynomial equations with coefficients as the design variables. However high order polynomials would result in oscillatory behaviour (Haftka and Grandhi, 1986). Splines such as B-splines and Non-Uniform Rational B-Splines (NURBS) would become the predominant method for defining smooth material boundaries (Seo et al., 2010). However, even with smooth boundaries a problem that still remains with shape optimisation is maintaining the mesh quality as the boundaries move (Seo et al. (2010); Haftka and Grandhi (1986)). Elements near the boundaries can become distorted and thus re-meshing is typically required to attain accurate results. Robust automatic re-meshing strategies remain an active area of shape optimisation research.

As well as challenges associated with re-meshing, difficulties arise when boundaries merge or split, requiring re-parametrisation of the boundaries. The level set method conceived in 1988 offers a potentially more robust method of determining the material boundaries by avoiding the need for re-meshing and re-parametrisation (Osher and Sethian (1988); Allaire et al. (2004); Brampton et al. (2012)). The boundaries between void and material are defined by isocontours of a level set function which is illustrated in Fig. 2.5. However like all shape optimisation methods the level set method cannot create new holes and thus as a standalone method cannot change the topology during the optimisation. For more detail on the level set method the reader is referred to a recent review by Dijk et al. (2013).

In 1988, in what can now be described as a landmark paper, the first continuum (finite element mesh) topology optimisation method was proposed by Bendsøe and Kikuchi (1988). Rather than changing the thickness of an element or moving the material boundaries, in the proposed homegenization approach the density of each finite element was varied. Elements in the mesh could take a value from 1 (full material) to zero (no material), where the boundaries could be defined between the 1 and 0 elements. Naturally it is desirable to allow elements to only take the values of 1 and 0 leading to well defined material boundaries, but this kind of problem is ill posed and lacks attainable solutions. To mathematically relax the problem the homogenization methods allows for the presence of intermediate densities by representing each element with a composite material. The topology of the composite is chosen *a priori* (i.e. a unit cell with a prescribed number of holes) and through homgenization theory a continuous function relating the density of the

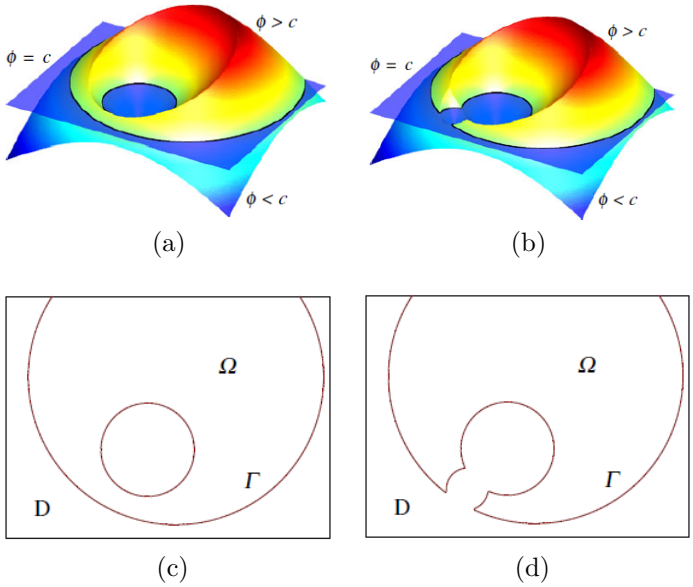


Figure 2.5: Example of a level set function (a) before and (b) after a design update with the respective isocontours defining the material boundaries (c) and (d). Dijk et al. (2013)

composite to the effective linear-elastic material properties is produced. The optimisation then works very much like a sizing optimisation problem where the density of each element is the design variable. The final topology can then be interpreted from the resulting density distribution.

Just one year after the homogenization approach came what would be one of the most widely used optimisation methodologies - the Solid Istoptropic Material with Penalization scheme (SIMP) by Bendsøe (1989). Similar to the homogenization method, the SIMP method is based on the formulation shown in (Eqns. 2.1 - 2.5) where the objective of the optimisation is to minimise compliance (product of the external forces F and nodal displacments U) by varying element density ρ_e (Eqn. 2.1). The difference between them is the material model i.e. how the density of the element relates to its stiffness. Rather than using composite unit cells, the SIMP method uses an artificial material model in which the element stiffness E is a product of the element density to a power p and the elastic modulus of the material E_0 (Eqn. 2.2). With a power of p equal to one the problem is linear and a global solution is guaranteed. However, with only a linear relationship between density and stiffness the final solution will likely include a substantial number of elements with intermediate densities. In SIMP intermediate densities have no practical meaning and are hence undesirable. To promote a 1-0 solution a power more than 1 must be applied which effectively makes elements with an intermediate density more costly (i.e. lower stiffness to density). Although this now makes the problem non-linear, which can lead to a local optimum solution, it does yield solutions with more well defined material boundaries.

$$\min_{\rho} = F^T U \tag{2.1}$$

s.t.

$$E(\rho_e) = \rho_e^p E_0 \tag{2.2}$$

$$\int_\Omega \rho_e \mathrm{d}\Omega \leq V \tag{2.3}$$

$$0 < \rho_{min} \leq \rho_e \leq 1 \tag{2.4}$$

$$KU = F \tag{2.5}$$

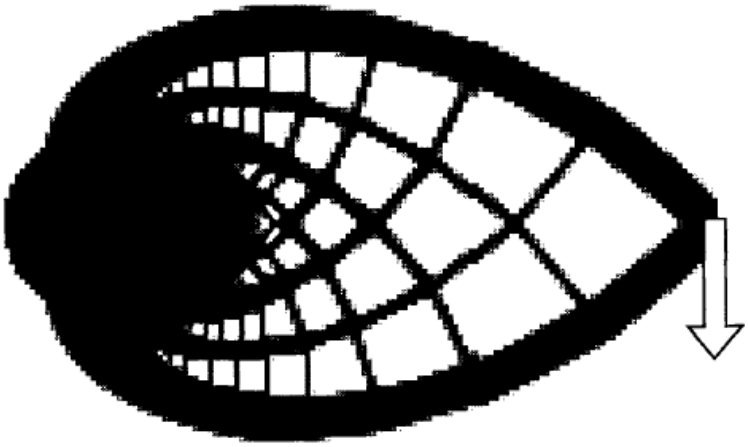


Figure 2.6: Solution to the Mitchell Cantilever problem using SIMP (Sigmund, 2000)

Both the homogenization and SIMP methods use a linear elastic FE analysis (Eqn. 2.5), which provides the nodal displacements for the following iteration.

The obvious solution to any minimum compliance formulation is a design domain (Ω) completely filled with material (i.e. every element density equal to one), so to prevent this a volume constraint must be enforced. In its simplest form can be set as a fraction of the design domain volume (2.3). Conceptually the density of each element varies between 0 and 1 although in reality a lower bound must be in place (ρ_{min}) to ensure density does not exactly equal zero, and lead to a singularity.

Although the intermediate densities in the Homogenization method have physical meaning the selection of the composite topology does influence the solution. Material with intermediate density in the SIMP method is artificial and is only present to allow a solution. However it has been shown for certain values of p that the intermediate density can be physically interpreted as a composite (Bendsøe and Sigmund, 1999). But generally it is desirable to have topologies of solid material and hence this is why SIMP with its penalisation of intermediate densities has become the more prevalent of the two methods.

The resulting topology from the SIMP method is heavily depended on the FE discretisation. It is often desirable to use a finer mesh to improve accuracy and produce higher resolution material boundaries (in the case of cell based approaches, illustrated Fig. 2.6 where a coarse mesh leads to non-smooth boundaries). With SIMP the use of a finer mesh will result in a lower compliance solution albeit one that is more geometrically complex (illustrated in Fig. 2.7). From a manufacturing/practical perspective this is a great limitation as relatively simple solutions can only result from a coarse FE mesh. SIMP is also prone to a phenomena known as 'checkerboarding' when using lower order finite elements. This is due to the fact that checkerboards (alternating 1 and 0 density elements in the

pattern of a chess or checkerboard - see Fig. 2.7) have an artificially high stiffness for certain discretisations. The latter problem can be largely overcome by the use of higher order elements. But both these issues can be avoided by the use of heuristic filtering techniques, which modifies the sensitivities to be based on the weighted average of local groups of elements (Sigmund and Petersson, 1998)

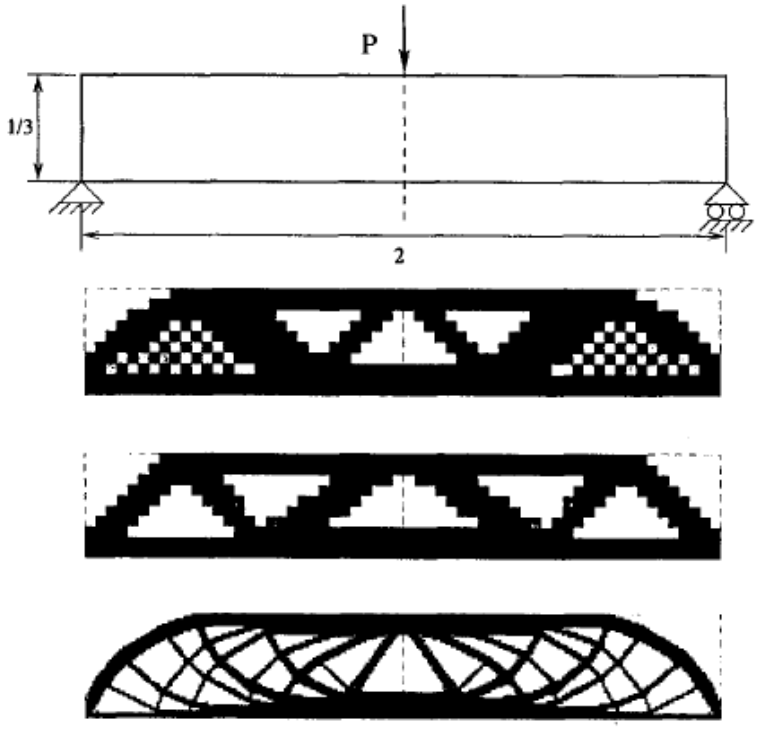


Figure 2.7: Solution to MBB problem with cell based methods highlighting the checkerboarding and mesh dependency issues. Adapted from (Sigmund and Petersson, 1998)

The update scheme for SIMP was the originally based on the Optimality Criteria (OC) method (more information can be found in (Bendsøe and Sigmund, 2003)). However the more flexible Method of Moving Aymptotes (MMA) method (Svanberg, 1987) and the Globally Convergent Method of Moving Asymptotes (GCMMA) (Svanberg, 2002) are generally preferred as they can handle a wider range of optimisation problems. The scheme terminates when the Karush-Kuhn-Tucker (KKT) conditions, which are first order criteria for optimality in non-linear programming, are met.

An alternative approach is the Evolutionary Structural Optimisation (ESO) method proposed by Xie and Steven (Xie, 1993). Unlike the homogenization and SIMP methods this method can be described as a discrete density method. Working on a 'hard kill' strategy, elements with low Von Misses stress are iteratively deleted until a final solution is arrived upon. One of the problems with this intuitive approach was that once an element was deleted it could not be considered as part of the solution in later iterations. This drawback was overcome with the Bi-Directional Evolutionary Structural Optimisation (BESO) (Xie et al., 1999) method which allows for elements to be added as well as deleted and works in both 2-D and 3-D with multiple load cases. Because the solutions from both the ESO and BESO methods do not have regions of intermediate density or grey areas. the boundaries are more easily defined. However, with coarse meshes some form of post-

processing smoothing will be required to arrive at a practical design solution. Like SIMP the ESO/BESO approaches are also prone to checkerboarding and mesh-dependency issues (Huang and Xie, 2010). Another issue with the ESO/BESO methods is that, unlike the SIMP method, there is no rigorous stopping criteria such as the KKT conditions (Sigmund and Maute, 2013).

2.1.3 Current state of the art

Continuum based methods - SIMP, ESO/BESO and levels sets

To date the SIMP method has enjoyed great success and has been implemented in several commercial FE based packages such as Optistruct, Ansys and NASTRAN (Rozvany, 2009). The ESO/BESO methods are also actively researched and have been implemented in the commercial FE package TOSCA. It has been argued by Huang and Xie (2007) that the ESO/BESO methods can be faster than SIMP as fewer elements are included in the FE analysis on subsequent iterations, although there does not seem to be any quantitative information to support this claim. But the ESO/BESO methods are criticised for their convergence characteristics and, as mentioned previously, have no rigorous stopping criteria. Originally based on a 'hard kill' approach where elements of low stress were deleted, the BESO method has been revised to use gradient information and sensitivity filters in a 'soft kill' approach (i.e. element stiffnesses are reduced rather than removed completely (Huang and Xie, 2010)). The authors include a Matlab code which is remarkably similar to the 99 line SIMP Matlab code (Sigmund, 2001). The sensitivity filters to control checkerboarding and mesh dependency are near identical. In fact it can be argued that the BESO method is now just a variation of SIMP that uses discrete variables (Sigmund and Maute, 2013)

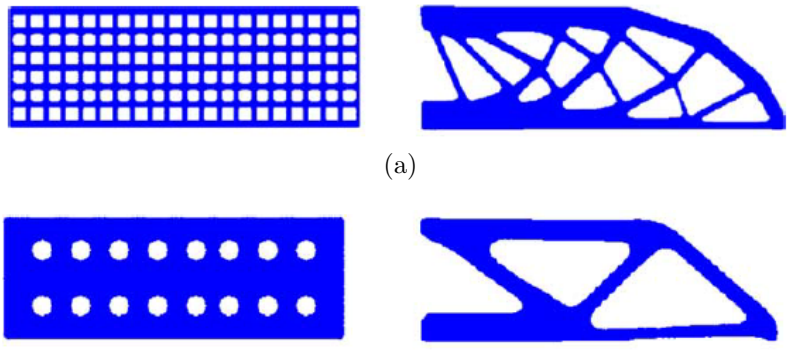


Figure 2.8: Two solutions for the half MBB problem obtained using the level set method from two different initial topologies (left) Yulin (2004).

Since the level set method cannot generate new holes during the optimisation, it cannot be classed as a topology optimisation method outright. But, as mentioned previously, the method can merge boundaries and effectively remove existing holes so some studies have made use of an initial topology with numerous, regularly spaced, holes (see Li et al. (2015), Villanueva and Maute (2014) & Yulin (2004)). Although seemingly effective this does make the solution dependent on the initial topology (see Fig. 2.8). And whilst it would seem intuitive to include as many holes as possible in the initial topology this does not necessarily address this issue and can introduce numerical issues (Dijk et al., 2013). Topological derivatives are now often used with the level set method to determine where

new holes need to be nucleated during the optimisation, but the solution is still influenced by the initial topology (Brampton et al. (2012); Dijk et al. (2013)).

Because level set approaches work on moving boundaries rather than changing element densities (either in a continuous or dicrete way), as is the case with SIMP and ESO/BESO, they do not suffer from numerical mesh related issues such as checkerboarding. But the moving boundaries are most advantageous in that clear material/phase boundaries are maintained throughout the optimisation, making them especially appealing for multi-material structural optimisation (e.g. Yulin (2004) & Villanueva and Maute (2014)) and problems where a well defined boundary is a necessity, such as when modelling fluid flow (Sigmund and Maute (2013)). This does not mean that the final topology is smooth (see Fig. 2.15), so much like the density methods a significant amount of post-processing is still likely to be required to produce a finalised design (Deaton and Grandhi, 2013).

It is not clear which of the three methods discussed here has a clear advantage for single material structural optimisation problems. All three of these methods, which are the centre of a substantial amount of the research effort in this field, are based on minimising compliance. There have been calls to place more emphasis on stress based approaches (Sigmund and Maute (2013); Deaton and Grandhi (2013)), which would minimise structural volume for a given stress limit or to minimise stress for a given volume. Another potential issue with minimum compliance problems is how multiple load cases are dealt with. Two prominent methods are to minimise the weighted sum of the compliances from all the load cases or to minimise the maximum value of all the load cases (known as min-max). The former requires the user to apply a weighting value to each load case as they see appropriate. The min-max formulation removes this heuristic requirement by always minimising the largest compliance value from all the load cases. However, issues have been noted using both of these formulations when designing structural components, such as the wing box ribs for the Airbus A380 (Krog et al., 2004).

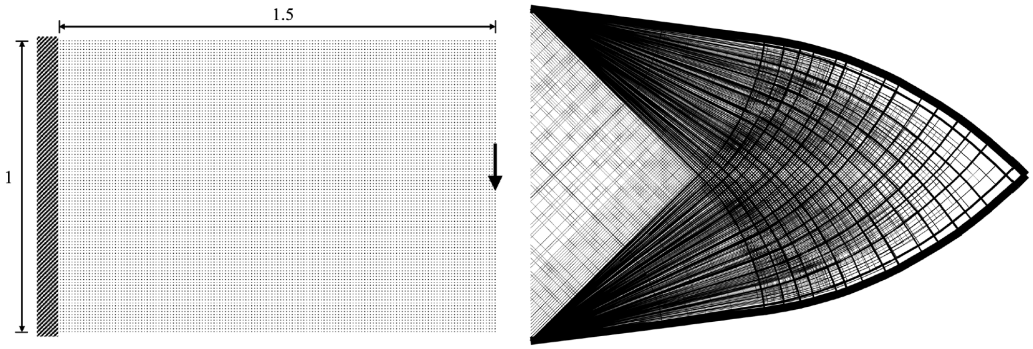
Plastic layout optimisation

Since Dorn et al. (1964) published the formulation for plastic layout optimisation (LO) much of the research field has focused on FE methods that minimise compliance. Recent work on layout optimisation has highlighted its potential as an alternative stress based topology optimisation method. Still using the ground structure approach proposed by Dorn et al. (1964), the method has been shown to be capable of solving much larger problems. This is due to advances made in the field of Linear Programming (LP), particularly with the advent of the interior point method which is very efficient for large problems. But even with more efficient LP algorithms, the computational cost of the ground structure approach becomes prohibitively expensive when using a large number of nodes (n) with the number of potential members for full connectivity given as $n(n-1)/2$.

To permit the solution of very large problems even on modest hardware, Gilbert and Tyas (2003) developed an algorithm that uses an adaptive ground structure approach. The algorithm involves starting the LP optimisation with a ground structure of nodes with adjacent connectivity rather than full connectivity, and then progressively adds more members as necessary in each iteration. Members are added from a pool of members that would be present in the fully connected ground structure. Determining which of these members to add is based on Michell's optimality criteria, which states that the virtual strain of all members must be within the limits shown in Eqn. 2.6 (where σ_i^- and σ_i^+ are then tensile and compressive strength). The virtual strains are easily computed from the

virtual displacements obtained from the dual LP problem. Any members in the pool that violate this condition are candidates to be included in the ground structure for the next iteration. Because member adding avoids analysing many of the potential members which would otherwise be excluded from the final topology it significantly improves the efficiency of the layout optimisation approach. The benefits of member adding are illustrated in Fig. 2.9 where a cantilever problem is solved using less than 0.1% of the 116,288,875 member variables that would have been required with a fully connected ground structure (Gilbert and Tyas, 2003). And yet the solution is within 0.038% of the analytical Michell type solution obtained by Zhou and Rozvany (1991). This ability to solve large scale problem efficiently makes layout optimisation ideal at finding very good estimates of exact analytical solutions, which are often well within 1% of the latter in the case of two-dimensional problems (e,g. Darwich et al. 2010a, Sokó and Lewiński 2010)

$$-\frac{1}{\sigma_i^-} \leq \epsilon_i \leq -\frac{1}{\sigma_i^+} \tag{2.6}$$



(a) (b)

Figure 2.9: A cantilever problem solved using layout optimisation yielding a solution that was just 0.038% greater than the analytical solution. With member adding only 215,103 members were required out of a potential 116,288,875 which would have been present in fully connected ground structure. Images taken from Gilbert and Tyas (2003)

The member adding approach has since been extended to 3-D problems with multiple load cases by including the member areas as well as the axial forces as variables (illustrated in Fig. 2.10 by Pritchard et al. (2005)). A constraint is then enforced on the areas such that the limiting stresses in all members are not exceeded for any load case. Unlike a single load case topology there will be redundant members for each load case (i.e. not fully stressed). As with 2-D problems, it has been demonstrated that with member adding, layout optimisation can solve large scale 3-D problems with multiple load cases (Gilbert et al., 2005).

A problem with extending the LO to 3-D problems is that instability may occur at nodes that are on the end of one or more members in compression. This of course would lead to local or overall failure of the structure. To promote structural stability a method proposed by Tyas et al. (2005) applies small nominal loads at each node which will require a solution that either braces unstable joints or avoids the formation of them all together. An illustration of this is shown in Fig. 2.11 where a simple problem with a single vertical load leads to a solution of two in-line compression members. Of course any deviation to the applied load or disturbance to the structure would result in collapse. Applying nominal loads in an orthogonal plane to these members would result in a more robust truss layout.

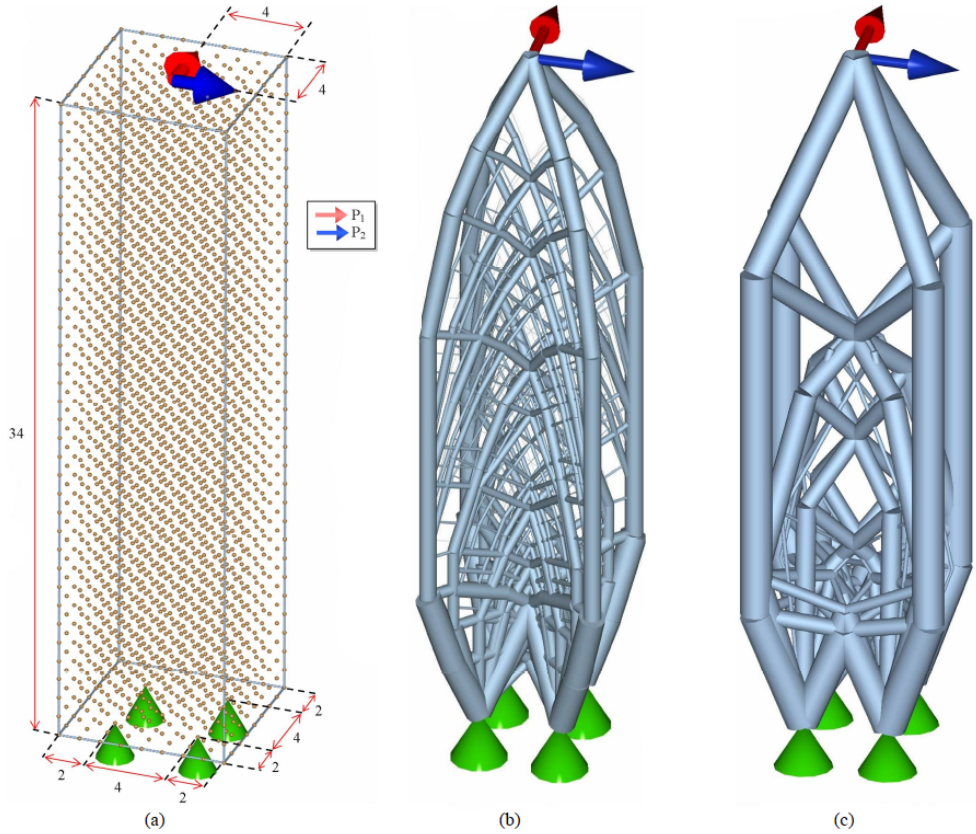


Figure 2.10: (a) 3D Prager type cantilever problem (b) basic solution (V = 1.191 m^3 and (c) solution with joint costs included V = 1.202m^3). Images taken from Pritchard et al. (2005)

These nominal loads are set as a percentage of the compressive forces applied at the node (typically 2%)

The resulting topology from layout optimisation is dependent on the initial ground structure, with more potential members producing a better but more geometrically complex solution. It will be undesirable, from a manufacturing perspective, to produce a design with too many members. To limit the geometric complexity of resulting topologies, the use of a joint length penalty was proposed by Pritchard et al. (2005) to make the occurrence of shorter members more costly. Based on the work by Parkes (1975), a constant length is added to each member length during the total volume calculation. Thus the optimisation should favour a solution with fewer longer members over one with many short members. Joint length penalties can be seen as analogous to the mesh independency filter used with the SIMP method (Sigmund and Petersson, 1998) to limit complexity. This is illustrated in Fig. 2.10c where a joint cost has simplified the resulting layout with a marginal increase in total volume.

As LO is formed as a LP problem finding a global optimum is guaranteed and for a single load case it has been proven that the resulting topology will also be the solution to the minimum compliance problem (Bendsøe and Sigmund, 2003). However non-linear behaviours, such as the moment capacity at rigid joints and local buckling of the members, are not accounted for during the optimisation.

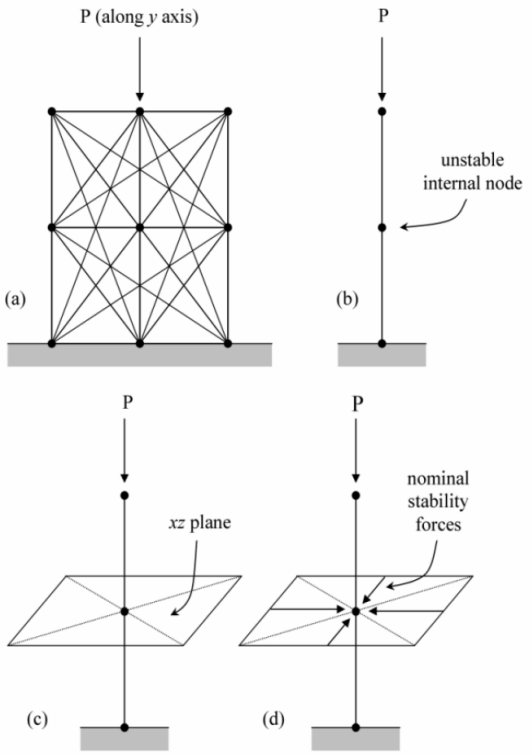


Figure 2.11: Illustration showing a simple problem with an unstable solution and how nominal forces are applied to prevent such a solution. (taken from Tyas et al. (2005))

Non-gradient methods

The methods discussed so far compute gradients on each iteration to be used as a basis for updating the design variables for the next (except the original ESO formulation). The use of gradient information results in a very efficient search of the design space and can thus converge on a solution in significantly fewer iterations than direct methods.

With optimisation problems with complicated constraints or physics it can be challenging implementing a gradient based method (for example enforcing a reliable stress constraint in the SIMP method has proven difficult (Sigmund and Maute, 2013)). This is because gradient based formulations can run into difficulties with functions that are not well behaved i.e. non-convex, non-smooth and discontinuous. Because non-gradient methods avoid much of the mathematics associated with gradient based methods they are far more flexible and much easier to implement. An example of this was the design of a satellite launch vehicle which accounted for the coupled parameters of propulsion, aerodynamics, structural mass and launch trajectory to minimise overall mass (Rafique et al., 2010). Solving a multidisciplinary optimisation problem such as this would be very challenging with gradient based methods.

But as well as multidisciplinary optimisation problems, non-gradient methods are being proposed as an alternative means to structural topology optimisation. In the case of Genetic Algorithms (GA), which are inspired by the process of evolution, every single possible solution to a particular problem is represented by a binary string (analogous to a DNA sequence). In a structural optimisation problem the ones and zeros in each solution string will correspond to the discrete densities of each finite element. During each iteration

the strings are cross-bred with one another and occasionally mutated. An FE analysis is then performed for each solution where a fitness function extracts improved solutions for the next iteration. This survival of the fitness process is repeated until the best solution is found.

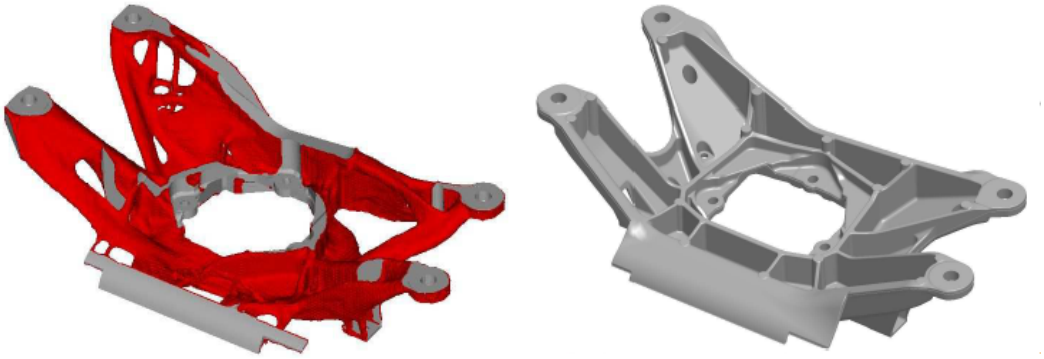
It is cited that because non-gradient methods use global search techniques and discrete variables that they are more likely to find the global optimum and the resulting solution will be one with well defined boundaries (i.e a 1-0 solution). The argument against them is that the computational cost far outweighs the benefits and there is still no guarantee in finding the global optimum (Sigmund, 2011). The immense computational cost of such methods is demonstrated with the optimisation of a 2-D cantilever beam by Wu and Tseng (2010). Finding a minimum compliance solution required 15,730 evaluations compared to 60 with SIMP. And as the number of function evaluations increases exponentially with the number of variables, the size of the problems that non-gradient methods can solve is severely limited. Thus even relatively recent literature on non-gradient methods for structural topology optimisation often use very coarsely discretised 2-D FE meshes (<1000 elements) e.g. Genetic Algorithms (Chen and Chiou (2013); Su et al. (2009)), Simulated Annealing (Bureerat and Limtragool, 2008), Particle Swarms (Luh et al., 2011) and Ant Colonies (Luh and Lin, 2009).

2.1.4 Applications of structural optimisation

As mentioned earlier, topology optimisation methods such as SIMP and ESO/BESO have been incorporated into several FE packages which are now actively being used to design components in the aerospace and automotive industries. Naturally finding details on industrial applications is challenging but case studies are published from user group conferences by optimisation software developers such as Altair.

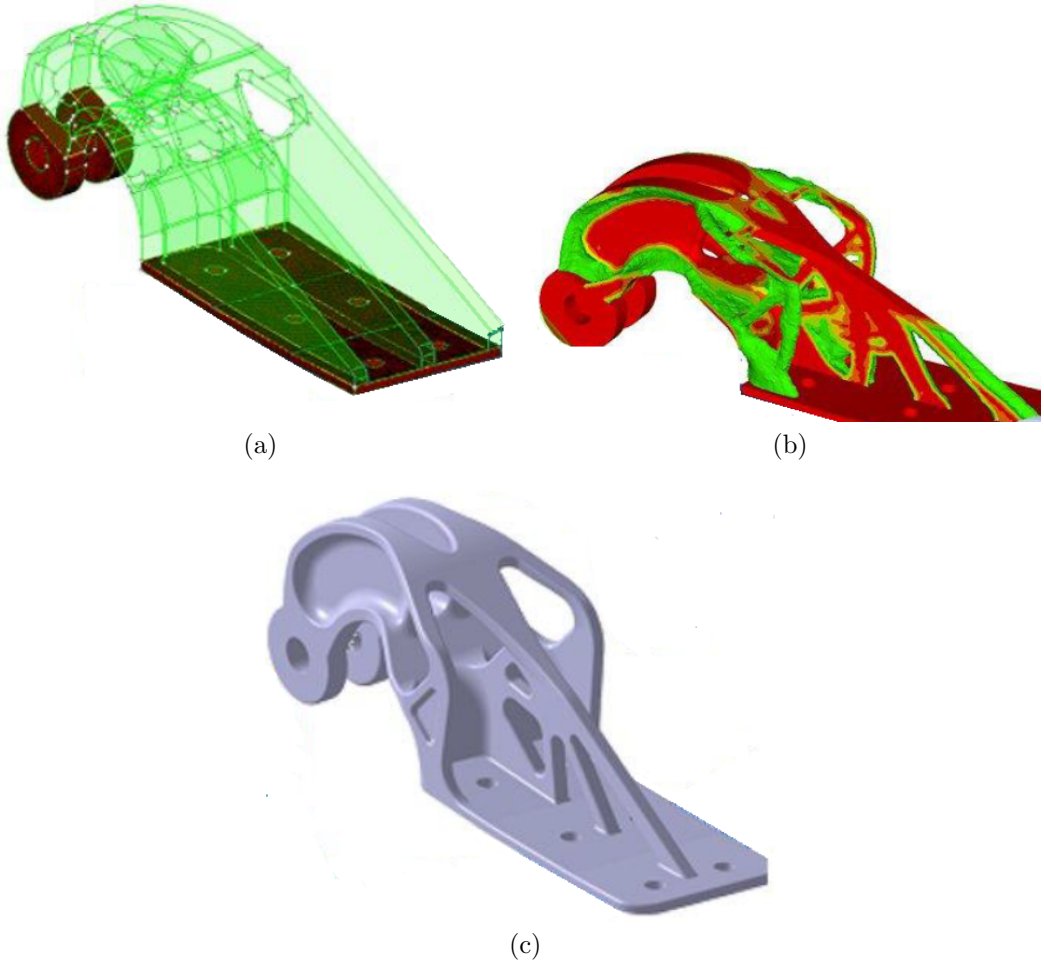
The problem with some these case studies is that it is difficult to ascertain the quantity of manual input that was required to produce a final design from the output from topology optimisation. As it can be seen in Fig. 2.12, the design of a gearbox mounting by Hougardy (2009) and Fig. 2.13, the design of a jet engine nacelle hinge by Tomlin and Meyer (2011), there has been substantial modifications made to the topology optimisation output to yield a final design. And whilst the resulting topology of the nacelle hinge is very elegant, it has clearly been heavily influenced by the shape of the design domain shown in Fig. 2.13a.

One reason for significant design modifications is that the resulting topology is not always suitable for the intended manufacturing route (as was the case with the gearbox mounting which was to be die cast). But the main reason is that the surfaces of the resulting topology are not smooth (partly due to the FE mesh resolution) and not of a quality required for an in-service component. This can be seen clearly in Fig. 2.14 which is a rendering of a bracket designed using the BESO method without any user refinement. This has been promoted as looking organic but in reality this part would have to undergo a similar transformation seen from Fig. 2.13b to 2.13c to be a feasible design. Thus, many authors have stated that minimum compliance FE methods such as SIMP produce solutions that act as starting point in the design cycle rather than the final design (Wieloch and Taslim (2004); Hougardy (2009); Kang et al. (2009)). The same can be seen with the Level Set method where in Brampton et al. (2012) it has been used to optimise the internal structure of a wing for a light aircraft. The resulting output shown in Fig. 2.15 would need a great deal of manual post-processing to produce a finalised design.



(a) (b)

Figure 2.12: Case study by Audi to reduce structural mass of a gearbox component with (a) the result obtained from Topology Optimisation and (b) the final component design (Hougardy, 2009).



(a) (b)

(c)

Figure 2.13: Case study by EADS to reduce the mass of a nacelle hinges for the Airbus A320. (a) The design domain, (b) the output from topology optimisation and (c) the final component design (Tomlin and Meyer, 2011).

Premature failures of components optimised for minimum compliance have been reported. Currently with the SIMP method there is not an efficient way to apply a robust stress constraint, and this can potentially complicate the optimisation problem (Sigmund and

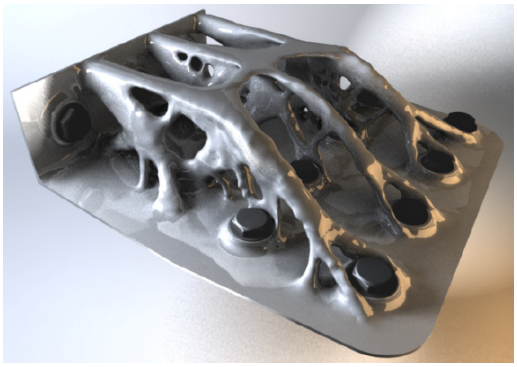


Figure 2.14: Bracket designed using the BESO method (Brackett et al., 2011).

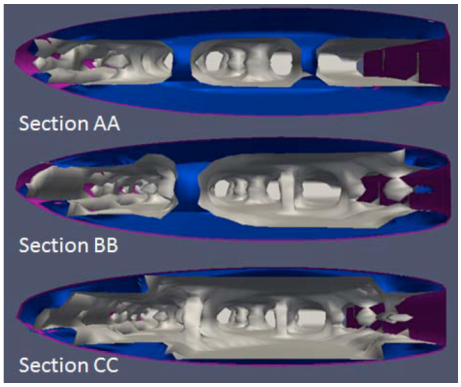


Figure 2.15: Solutions for the internal structure of light aircraft wing determined using the Level Set method. Grey indicates the internal structure (Brampton et al., 2012)

Maute (2013); Deaton and Grandhi (2013)). For some problems such as the nacelle hinge (Fig. 2.13) this did not present a problem because imposed displacement constraints resulted in low stresses in the component (Tomlin and Meyer, 2011). But for other such as the optimisation of wing box ribs for the Airbus A380 this did prove to be an issue (Krog et al., 2004). Another reason cited for the premature failures is poor modelling of the boundary conditions when optimising components locally (i.e. isolated from the system they will be integrated into) and an insufficient number of load cases to fully represent the those in service (Hougardy, 2009). Properly accounting for multiple load cases using the weighted sums and min-max SIMP formulation has also presented problems (Krog et al., 2004)

Cases presented by industry often include components that have been substantially modified post-optimisation and do not provide any data or conclusions from load testing. There are academic studies that have optimised simpler components and a few of these have manufactured and load tested the resulting forms with little modification. Dewhurst et al. (2003) manufactured an extruded 2-D MBB based on the analytical solution using CNC milling. The fabricated specimens reached the design target load. Several truss structure specimens optimised for minimum compliance using numerical shape and size optimisation were manufactured using several additive manufacturing methods by Cansizoglu et al. (2008). However these were found to have a higher compliance than expected which was at least partly attributable to various material and manufacturing issues. There are numerous studies detailing the design of additively manufactured components using topology optimisation in the literature, but many of these researchers have stopped short of fabricating the designs (e.g. Brackett et al. 2011, Aremu et al. 2010).

2.1.5 Conclusions of optimisation state of the art

- A great deal of the optimisation research field has focused on SIMP, ESO/BESO and level set methods which will mostly minimise compliance.
- SIMP and to a lesser degree ESO/BESO are the most prominent methods and have been implemented in commercial packages such as Hyperworks Optistruct.
- Issues can potentially arise with minimum compliance formulations when dealing with multiple load cases. Issues have been documented with both the weighted sums and min-max approaches.
- The outputs from SIMP and ESO/BESO and the Level Set methods usually require significant refinement to produce a feasible final design, even though the latter two produce solutions with clear boundaries. Shape optimisation and/or manual design input is usually used for this task.
- Non-gradient methods are cited as an alternative as they can be easily and robustly applied. However their direct search nature makes them very inefficient compared to their gradient based counterparts and hence greatly limits the size of problem that can be solved.
- Non-gradient methods are however suitable for multidisciplinary optimisation problems, where the design space is not fully understood and would make it challenging to implement gradient based approaches successfully.
- It has been observed by some that there is a gap in the research field with regard to stress based optimisation approaches. Several authors have advocated moving emphasis away from minimising compliance to minimising volume for a given stress limit.
- Layout optimisation also deals with multiple load cases rigorously and robustly by ensuring that all members in the final layout do not exceed the limiting stress for all load cases.
- As layout optimisation uses linear programming solvers the solution is guaranteed to be a global optimum. Duality can also be taken advantage of to reduce the problem size and hence allow very large problems to be solved.
- The use of discrete truss elements yields well defined designs that are potentially closer to a finalised design than those produced using FE meshes.
- The layout optimisation formulation discussed here can treat truss topologies only, which may not always be the most optimal for 3-D problems.
- This formulation also assumes pin jointed members and thus bending moments are neglected.
- For a single load case the resulting topology from plastic layout optimisation will be very similar to that for minimum compliance.
- For all optimisation methodologies there are very few cases where optimised designs have been physically load tested to assess their efficacy.
- Industrial case studies generally do not quantify the manual input required between the optimisation output and the final design.

2.2 Additive Manufacturing

2.2.1 Background on AM technologies

Additive manufacturing is based on the concept of decomposing a 3-D shape into a stack of 2-D cross-sections. The 3-D part is then realised layer by layer by fusing/binding material.

The first Additive Manufacturing technology was the stereolithography process developed by Hull (1986). Using a photosensitive liquid monomer/polymer a solid part could be created through photopolymerization by exposing regions to UltraViolet (UV) light. The solidified part is produced on a platform that moves vertically in a bath of resin. Starting at the surface the platform move down in decrements typically around 50μm (Halloran et al. (2011)), solidifying the part in layers. The remaining non-solidified liquid resin can be reused and different materials can be used in the process by draining the bath and filling with the new material at various stages. Imperfections with the process mainly occur on negative surfaces (regions being solidified with no solidified material beneath) where over-curing occurs i.e. resin not intended to be solidified. The viscosity of the liquid may also result in uneven layers.

Since Stereolithography additive manufacturing has become very diverse in terms of the material deposition strategies and binding mechanisms. Broadly speaking they can be split into the following categories:

- Powder Based
- Solid Deposition
- Liquid Based

Each category is discussed individually in Sec. 2.2.2 to 2.2.4. The current state of the art relating to the electron beam melting (EBM) process is then discussed in 2.2.5.

2.2.2 Powder based systems

The focus on this section will be on powder bed processes rather than blown powder systems such as Laser Engineered Net Shaping (LENS) as these processes are typically used for repair and/or modification of existing components. Powder bed systems operate on the same principle as stereolithography in that the parts are built in layers on a platform that moves down in small decrements. Instead of a liquid polymer/monomer a layer of fine powder particles material is deposited which is then sintered or melted using either a laser or electron beam heat source. The size of the powder particles vary on the material and process but are generally in the range of 5 to 100 μm (Król et al., 2013) in diameter and are deposited in layers of 100μm or less.

The most popular powder based processes are Selective Laser Sintering (SLS), Selective Laser Melting (SLM) and Electron Beam Melting (EBM). SLS systems do not melt the particles but rather fuse them together through a variety of mechanisms such as solid state sintering, partial melting or through the use of a binding material. As a result the parts are not fully dense making the process suitable for non-load bearing components only. Process parameters and part orientation also have a strong influence on the mechanical properties of the final part, particularly with Nylon-12 where attaining repeatable mechanical properties between builds or machines can be challenging (Caulfield et al. (2007);

Zarringhalam et al. (2006); Vasquez et al. (2011)). SLM and EBM do fully melt the powder particles resulting in parts which are more than 99% dense ((Król et al., 2013); (Vandenbroucke and Kruth, 2007)) and comparable in strength to bulk material.

The range of powder materials that can be used in the laser based process is much greater and diverse than that in the EBM process. This is because the EBM process is limited to materials with sufficient electrical conductivity (i.e. metallic based) to avoid an accumulation of negative charge in the powder bed. Laser systems can sinter/melt any material that absorbs laser wavelengths and as a result a far greater range of materials such as polymers, ceramics and metals can be used in the process.

Because laser systems can maintain smaller meltpools than EBM systems they are capapable of a higher feature resolution. Combined with the use of thinner layers (30 μm compared with 70μm) and slower scan speeds, laser systems also produce parts with a better surface finish (Rafi et al., 2013) (although parts from both processes normally require additional surface finishing). But as laser systems deflect the beam using galvanometers (mirrors steered by motors) the scan speed is fundamentally limited by the inertia of the mirror. This can limit part complexity particularly when melting layers where the beam has to jump between many disjoint cross-sections or shells, e.g. with lattices or truss structures. EBM systems on the other hand can deflect the beam near instantaneously though the use of electromagnetic coils. This permits very high scan speeds and the ability to sustain multiple melt pools simultaneously.

Because of the localised melting in powder bed processes there are high thermal gradients upon solidification. Whilst this is beneficial for producing fine microstructures which results in higher strength properties it also produces parts with residual stress due to the shrinkage during cooling. The problem manifests itself mostly in metal powder bed processes due to the higher melting temperatures and hence higher thermal gradients (Mercelis and Kruth (2006); Vrancken et al. (2013); van Belle et al. (2013)). The EBM system however can overcome this issue by uniformly preheating every layer through the near instantaneous deflection of the electron beam. In doing this much of the residual stress build up is eliminated through annealing (Al-Bermani et al., 2010). Residual stresses can cause dimensional errors through the part warping during the process and will also reduce the strength of the part ((Mercelis and Kruth, 2006); (Vrancken et al., 2013)). Generally larger parts with many negative surfaces (the downward facing surfaces of any overhanging features) will compound the issue of residual stresses leading to reduced strength properties and dimensional accuracy.

Whilst both systems require 'support structures' to provide a heat conduction path from a negative surface to the base plate, laser systems also require them to prevent part distortion. Negative surfaces of sufficient area will result in curling/warping which can interfere with the raking mechanism and hence prevent the process from reaching completion. Supports are used to effectively anchor the negative surface to the base plate and thus prevent significant distortion. Even with anchors warping may still occur and if the residual stresses are great enough then the anchors may break altogether (Mumtaz et al., 2011)

Heat treatments after the build can then be used to anneal the part and eliminate the residual stresses. But during heat treatment the part may distort, reducing the final dimensional accuracy. Attempts to reduce the residual stresses in SLM made part are being made by preheating the baseplate and powder bed during the build (Kempen et al., 2013). This usually involves the use of electrical resistance heaters place under the baseplate and around the powder bed as well as an Infra-red heater to preheat the top layer (Gibson and

Rosen, 2009). There is not much literature detailing how effective this preheating method is compared to the EBM process and most commercial systems do not incorporate these features.

2.2.3 Solid deposition systems

Fused Deposition Modelling (FDM) is one of the most prevalent AM technologies, partly due to its relative low cost. One of the main drawbacks with FDM is that it can lead to anisotropic material properties. Generally the strength properties are lower in the build direction and the properties are sensitive to the process parameters and deposition pattern (Ahn et al. (2002); Sood et al. (2010); Gibson and Rosen (2009)). The minimum feature size is determined by the nozzle diameter and dimensional inaccuracies may be introduced through shrinkage upon cooling (Sood et al. (2009); Rezaie et al. (2013)).

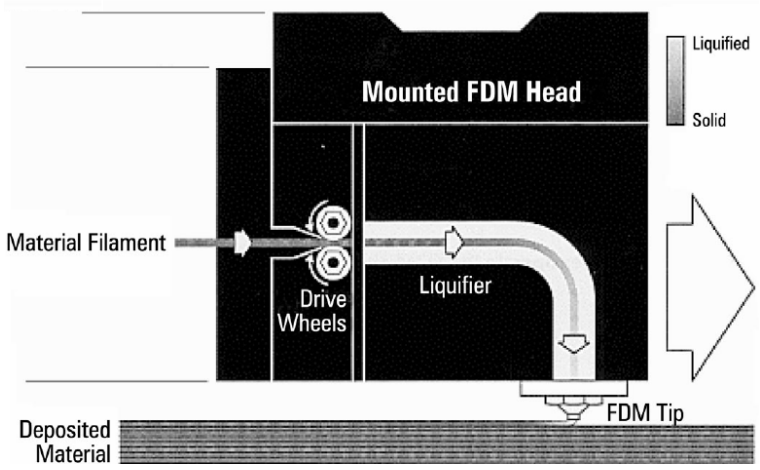


Figure 2.16: Schematic of FDM process (Ahn et al., 2002)

Wire deposition methods are actively being researched as an alternative method of manufacturing large complex engineering components. Rather than melt the feedstock as with FDM, solid wire is fed into a moving weld pool on existing deposited material using a tungsten inert gas (TIG), laser or plasma heat source. Using wire feed instead of powder deposition results in far higher build rates (e.g. 1.8kg/h for a plasma process (Martina et al., 2012) compared to 0.24kg/h for the EBM process (Arcam AB, 2013)) and far better control of impurities in the final part (Brandl et al. (2008); Brandl et al. (2011)). The main feature of the Wire Deposition approach is a manipulator arm that deposits and weld material. Because there is no need for an enclosed chamber that is kept under vacuum or filled with inert gas the only real limitation on size of the build envelope is the reach of the manipulator arm. The open setup also lends itself to incorporating additional stages during the build to improve part quality. One such feature is controlling the residual stresses and microstructure by cold rolling each deposited layer (Martina et al., 2013). Controlling the microstructure in powder-bed processes is much more challenging. The material properties achievable with wire deposition methods have been shown to be comparable to, or even slightly better than, those of wrought specification for Ti-6Al-4V ((Brandl et al., 2010); (Martina et al., 2013)).

Wire deposition processes can however suffer from anisotropy although this can be mostly alleviated by cold rolling during the process (Martina et al., 2013). The surface finish is

also worse than achievable with the powder bed methods. But by far the most significant disadvantage compared to powder based approaches is that they cannot produce components anywhere near as geometrically complex (Brandl et al., 2010). Nevertheless, they can produce parts more geometrically complex than is possible with conventional manufacturing routes and at sizes not possible with powder bed methods. In this respect they have their place for the production of large geometrically complex parts, e.g. of the scale of the airbus wing box Krog et al. (2004).

2.2.4 Liquid based systems

Liquid based systems such as those using photopolymer materials (e.g. stereolithography) and droplet deposition (inkjet head) systems can produce very high resolution parts with a high quality surface finish. This makes them especially suited for aesthetic or prototyping applications. Although there are methods of tailoring material properties, such as suspending ceramic powder particles in the liquid mediums (Halloran et al. (2011)), the material properties are generally inferior to those of powder bed processes. Thus these types of additive manufacturing processes will not be explored further in this literature review.

2.2.5 Electron beam melting (EBM)

Ti-6Al-4V grain structure formed

Titanium is an allotropic element meaning it can exist as two different crystal structures. Below a temperature of 882.5°C pure titanium exists as a hexagonal closed packed crystal structure which is referred to as alpha phase titanium. Above this temperature, which is called the beta transus, pure titanium exists as body centred cubic crystal structure or beta phase titanium. The versatility of titanium is derived from this phase change where alloying elements can be used to effectively control the degree of alpha and beta phase present in a titanium alloy. In essence this allows the tailoring of the titanium alloy to meet the required properties of the intended application. Alpha and beta alloys (α & β), particularly Ti-6Al-4V, are the most prominently used alloys in the aerospace industry (Boyer (1996); Peters et al. (2003)).

The solidification conditions in the EBM process, coupled with the solidification characteristics of Ti-6Al-4V, lead to the formation of coarse columnar beta grains. This is due to the limited supercooling characteristics of Ti-6Al-4V which makes it is very hard for new grains to nucleate ahead of the solid-liquid interface during solidification. As a result, after the initial grains have nucleated on a solid substrate (i.e. the base plate or surrounding powder at the edges of the part), a phenomena known as epitaxial grain growth becomes the dominant solidification mechanism (Kobryn and Semiatin (2001); Kobryn (2003)). This is growth of large columnar beta grains up through the build layers. The direction of the growth is parallel to that of the maximum thermal gradient which is perpendicular to the weld pool boundary (Porter and Easterling, 2001). Thus as the weld pool effectively moves up layer by layer, the columnar grains follow it. These coarse columnar grains orientated parallel to the build direction would be detrimental to the mechanical properties. Fortunately for titanium, the phase transformation at 882.5°C means that upon cooling these coarse beta grains transform into fine, seemingly randomly orientated alpha grains referred to as a Widmanstatten or basket-weave microstructure (Lutjering et al.,

2000). These very fine grains are why titanium components fabricated through EBM have excellent, near isotropic, static mechanical properties.

However many authors have speculated that because the orientation of the newly formed alpha grains is related to the prior beta grains (through the Burgers relationship (Burgers, 1934)), the alpha phase texture will be influenced by the parent beta phase (Kobryn and Semiatin (2001); Kobryn (2003)). This would then possibly affect other mechanical properties such as fatigue life (Kobryn, 2003). The Burgers relationship allows alpha grains to grow in 12 possible orientations from a parent beta grain; however it has been shown that the 12 variations are not necessarily selected equally (Stanford and Bate (2004); Humbert et al. (2006)). With only several of the 12 possible orientations being selected, a far less randomised texture would result, having undesirable effects on the mechanical properties. Fortunately for the titanium EBM process, it has been shown that variant selection does not occur (Antonysamy et al. (2012); Al-Bermani et al. (2010)). Thus, the effect of the highly textured, columnar, beta grains on the final texture formed after the phase transformation is dramatically reduced. Whilst this phase transformation is the reason Ti-6Al-4V parts exhibit near isotropic strength properties it was observed by Hrabe and Quinn (2013) that the elongation of horizontally orientated test specimens was 30% higher than those with a vertical orientation. This is likely to be due to the prior beta grains.

Kobryn (2003) and Bontha et al. (2006) have characterised the occurrence of columnar beta grains which has been related to the thermal gradient and solidification velocity. It is suggested that if the thermal gradient around the meltpool is reduced and/or the solidification velocity is increased then the more desirable equiaxed prior beta grains would form instead. Modifying the EBM process to achieve this would involve changing the electron beam parameters.

As the EBM process preheats each layer the resulting microstructure is heavily influenced by the preheat temperature. Al-Bermani et al. (2010) found that between 626 and 678°C there was negligible difference in the yield strength of as-built specimens. But increasing the preheat temperature further to 700°C resulted in a 5.2% reduction in yield strength. The reason for the reduction was observed to be due to a coarsening of the microstructure.

Internal porosity

It is well documented that parts made from Ti-6Al-4V using the EBM process will contain internal porosity. Fatigue cracks in polished fatigue specimens produced using the EBM process have been shown to mostly nucleate at pores, suggesting that internal porosity is currently the dominant microstructure feature in determining fatigue life (Edwards et al. (2013); Antonysamy (2012); Chan et al. (2012)). The fatigue life of these specimens has been shown to vary between samples by several orders of magnitude, with the locations of these pores having a strong effect. These pores are spherical ($<100\mu m$) and their origin is due to trapped gas from the powder manufacturing process. However larger defects can form as a result of process parameters that lead to insufficient melting (Bauereißet al., 2014). Standard Hot Isostatic Press (HIP) routines have been found to significantly reduce, perhaps even eliminate, internal porosity. High cycle fatigue properties of HIPed EBM components have been shown to be far more consistent and actually exceeded those of wrought standard (AMS 4928) (Brandl et al. (2008); Brandl et al. (2011)). HIPing is however an expensive process and can lead to a reductions in tensile strength due to grain coarsening (Al-Bermani et al., 2010).

Surface finish and dimensional accuracy

As expected the as-built surface produced from EBM will reduce fatigue life (Chan et al., 2012). But it has also been shown that the static properties such as yield strength and Ultimate Tensile Strength (UTS) are adversely affected too. Khalid Rafi et al. (2012) provides tensile data for Ti-6Al-4V Extra Low Interstitial (ELI) powder for specimens tested in their as-built state and polished. The yield strength and UTS and elongation were found to be 10 and 9% lower respectively for the former. The tensile test data for the machined specimens was largely consistent with those performed by Brandl et al. (2011).

Lower dimensional accuracy has been observed when manufacturing solid parts with the EBM process compared to machining by Cooke and Soons (2010). More specifically, it has been found that topology optimised truss structures are fabricated with undersized truss members by Cansizoglu et al. (2008). Members at an angle to the vertical build direction were found to be most affected. Topology optimised structures are especially sensitive to manufacturing defects and dimensional errors as there is little redundant material in the design. For manufacturing processes such as milling and etching these dimensional errors can be random. Probabilistic methods developed by Schevenels et al. (2011) attempt to make topology optimised designs more robust to random dimensional errors that occur when manufacturing planar structures by milling or etching.

2.2.6 Additive manufacturing File types - STL, SLC and AMF

STeroLithography or Standard Tessellation Language is a file format used to describe the surface of a 3-D object. Developed in 1987 by 3-D systems, the file format uses a triangular mesh to describe the 3-D profile. The files which can be read with any text editor detail the vertices and normal vectors of each triangular element. Because of its simplicity and thus universal compatibility with all manner of 3-D priniting software and CAD packages it is still the predominant file type for additive manufacturing. But its simplicity also contributes to many of its drawbacks. The first is the file is very inefficient for describing complex geometries, becoming prohibitively large with finer detail. The faceted nature of the described geometry makes it very hard to capture curved geometry without using a very fine mesh that results in a larger file size. STL files describing complex geometries are likely to be generated in a CAD package or with an algorithm and it is almost certain that these files will contain errors.

STL files can be thought of as error free when they present a single well defined and 'water-tight' surface, i.e. containing no holes or gaps. For complex geometries it will be high unlikely that a generated STL file will be completely error free (Szilvśi-Nagy and Mátyási, 2003). A well defined mesh will have all triangles connected at the edges forming a single surface. Each edge should border only one other edge from a neighbouring element. If there is a slight gap to a neighbouring edge because of a precision error or there is no there is no neighbouring edge at all (a hole), then these constitute bad or naked edges. The mesh surface describes a solid using vectors normal to each element to form a boundary representation (BREP), which is the boundary between solid and non-solid. A common error in STL files is that the normals are not coherent, which complicates distinguishing between a solid and non-solid. Fortunately this issue can usually be isolated to relatively few elements which have inverted normals. There will be cases where two or more triangles intersect each other and/or overlap. Whilst the latter is not much of an issue for additive manufacture any intersecting triangles must be removed or trimmed to form good edges. Finally, a seemingly single surface maybe composed of several surfaces or shells. If these

are closed and error free then these will not present an issue for additive manufacture, although it may be more desirable to union the shells into a single surface.

There are numerous additive manufacturing software packages that can automatically fix these errors but as the complexity of the STL file increases the likelihood of success decreases. At a certain degree of complexity an STL file will be too large to create a build file for additive manufacturing (notably the Arcam EBM control software). Another file format also developed my 3-D systems instead represents a 3-D BREP with a stack of 2-D slices with polyline contours. Called the 'sliced contour' (SLC) it is the preferred choice for complex or intricate topologies such as lattices because of its smaller memory requirements and the relative ease with which errors can be fixed. However, this file type is not as flexible for manipulating parts or generating support structures in preparation for additive manufacture.

With an ever increasing number of additive manufacturing processes ASTM committee F42 was created in 2009 to pave the way for a new universal data exchange format for additive manufacturing (Picariello, 2009). This would culminate in the creation of the 'additive manufacturing format' (AMF), initially dubbed STL 2.0, and covered by ISO and ASTM standard ISO/ASTM52915-13. The new file format still uses a mesh of triangular elements but also permits other properties such as colour, material and texture to be defined. This is especially useful for AM processes that use multiple materials or colours. For the EBM process this is not relevant but the AMF format does have several attributes that are beneficial. The first is that lattice structures can be defined more efficiently by specifying the repeating unit cell and the pattern rather than the whole structure. The AMF format also has a provision for curved elements, permitting a more accurate and memory efficient representation of curved surfaces. Finally, the contents of the file are stored much more efficiently. It does however still use a mesh to define BREPs which presumably means it is still susceptible to errors. Also, its relatively recent creation and slow uptake by many CAD packages means that currently the STL file format is still the predominant data exchange format.

2.2.7 Conclusions from additive manufacturing state of the art

- Metallic powder bed processes such as SLS/SLM and EBM can produce geometrically complex parts with high strength and consistent material properties, thus making them wells suited for realising designs produced using structural topology optimisation.
- SLM is capable of making metallic parts with a finer resolution than EBM but the resulting parts (particularly large parts) will have residual stresses affecting the strength properties.
- Preheating each layer in EBM eliminates the majority of these residual stresses but increasing the preheat temperature reduces the tensile strength properties of the fabricated component.
- Whilst wire deposition cannot produce parts with the same geometric complexity, it is fast and can produce parts which are much larger than powder bed processes. This could make it suitable for large optimised components such as the wing box ribs from the Airbus A380 case study.
- The static strength properties of EBM parts are affected by the as-built surface finish.

- Prior beta columnar grains forming during solidification of Ti-6Al-4V potentially may affect fatigue life but seem to have little effect on the static mechanical properties (except for elongation).
- EBM parts will contain numerous internal pores with a diameter of 200μm or less. Whilst these significantly affect fatigue life they should have a negligible effect on static strength. HIPing routines have been found to close internal pores, resulting in fatigue life comparable to wrought grade parts.
- Larger pores may form if insufficient energy is used during melting. If large enough these internal voids could adversely affect the static strength properties of fabricated pieces.
- Errors in the dimensional accuracy of components manufactured using EBM have been reported and could adversely affect the structural performance of topology optimised specimens.
- The STL file format, despite its shortcomings, remains the most popular data exchange format used in the additive manufacturing industry.

2.3 Literature review summary

An observation made by several prominent authors of the optimisation literature is that stress based approaches have been neglected in favour of methods that minimise compliance, and that the focus should be shifted towards the former. Layout optimisation not only addresses this but also offers several other advantages over minimum compliance methods, such as a rigorous and robust method of handling of multiple load cases and yielding designs that potentially require less manual refinement. However the fact that the standard implementation can only generate pin-jointed trusses, which may be in unstable equilibrium with applied loads, mean that it needs further investigation to see whether it can be rendered practically useful.

Another striking gap in the optimisation literature is the lack of physical load testing of fabricated optimised components. Whilst many authors cite additive manufacturing as being a viable route to fabricate optimised designs there are a surprisingly few authors who actually go as far as this stage. And of those who have, even fewer have physically load tested the fabricated specimens. This deprives the research field of two important pieces of knowledge; the efficacy of said optimisation method being assessed and the capability of the additive manufacturing technology to fabricate topology optimised designs. Additive manufacturing has become very broad as a field but many of the technologies themselves are far from mature, and hence have their own shortcomings. How these shortcomings affect how well they can capture the intricate features of topology optimised structures is a research area that has received relatively little attention.

Chapter 3

Plastic layout optimisation

3.1 Formulation

Numerical layout optimisation provides a powerful and efficient means of identifying the optimum topology of discrete truss structures. For a given set of load cases and support conditions, layout optimisation can be used to determine a minimum volume truss topology. With the basic single load case 'plastic' formulation (Dorn et al. (1964)) the resulting structure will be fully stressed when the design load is applied; the design solution obtained will also be the same as the topology derived using the minimum compliance formulation. However, for practical problems the use of multiple load cases is generally necessary to ensure robustness.

Layout optimisation is based on the ground structure approach, in which nodes distributed across the design space are interconnected by potential members (or 'truss bars'). The objective of the optimisation (Eqn. 3.1a) is to find which of these members makes up a structure of minimum volume and will be in static equilibrium (Eqn. 3.1b) when subjected to the applied loads whilst not exceeding the tensile or compressive stress of the material (Eqn. 3.1c). The optimisation will vary the internal forces (and thus the cross-sectional areas) of the potential members to achieve this. Those members that take a value of zero are removed from the final solution.

$$\min \; V = \mathbf{l}^{\mathrm{T}}\mathbf{a} \tag{3.1a}$$

subject to

$$\mathbf{B}\mathbf{q}^{\alpha} = \mathbf{f}^{\alpha} \tag{3.1b}$$

$$\left.\begin{array}{l} a_i \geq q_i^{\alpha}/\sigma^{+} \\ a_i \geq -q_i^{\alpha}/\sigma^{-} \end{array}\right\} \quad \begin{array}{l} i = 1, ..., m; \\ \alpha = 1, ..., p \end{array} \tag{3.1c}$$

where V is the total volume of the structure; l is a vector of individual member lengths, $\mathbf{l}^{\mathrm{T}} = \{l_1, l_2, ...l_m\}$, where l_i is the length of member i $(i = 1, ..., m)$, $\mathbf{a}$ is a vector of member cross-sectional areas, $\mathbf{a}^{\mathrm{T}} = \{a_1, a_2, ..., a_m\}$, where a_i is the cross-sectional area of member i $(i = 1, ..., m)$. Also, $\mathbf{B}$ is a suitable $(3n \times m)$ equilibrium matrix containing direction cosines; $\mathbf{q}^{\mathrm{T}}$ is a vector of member axial forces, $\mathbf{q}^{\mathrm{T}} = \{q_1, q_2, ..., q_m\}$, where q_i is the force in

member i ($i = 1, ..., m$); $\mathbf{f}^{\alpha}$ is a vector of applied loads and load cases, where α represents an individual load case ($\alpha = 1, ..., p$) and $\mathbf{f}^{\alpha \mathrm{T}} = \{f_1^x, f_1^y, f_1^z, f_2^x, f_2^y, f_2^z, ..., , f_m^z\}^{\alpha}$where f_j^x, f_j^y, f_j^z are the x, y and z direction components of the live load applied to node j ($j = 1, ..., n$). Finally, σ^+ and σ^- are respectively the limiting tensile and compressive stresses that can be sustained by the material. This type of problem can be solved using linear programming (LP).

A simple layout optimisation problem is shown in Fig. 3.1, which consists of a rectangular design domain with two fully constrained support points and a single tip load, i.e. a cantilever problem. The design domain has been populated with 21 nodes which, with full connectivity, leads to a ground structure of 210 potential members. Out of these potential members only 10 are present in the final solution.

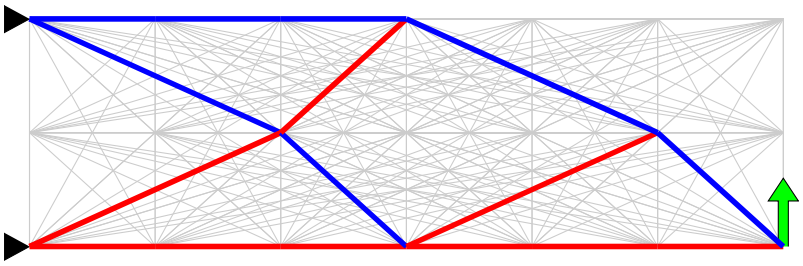


Figure 3.1: 2-D cantilever problem with a single load. The red and blue lines indicate tensile and compressive members respectively. The grey lines are the potential members in the ground structure.

3.1.1 Equilibrium matrix

The plastic layout optimisation assumes pin jointed truss elements and thus only axial forces are considered in the equilibrium constraint. For a single truss member joining two nodes 'a' and 'b' (Fig. 3.2), the corresponding equilibrium matrix is shown in eqn. 3.4 which resolves the internal member force into the global x, y and z components at the nodes (based on the element vector and length calculated in eqns. 3.2 & 3.3 respectively). This matrix is easily expanded for all members in the ground structure by including the external force components for each additional node in the RHS and resolving terms for each additional member in the LHS and B matrix respectively.

$$V_i = (x_b - x_a, y_b - y_a, z_b - z_a) \tag{3.2}$$

$$l_i = ||V_i|| \tag{3.3}$$

$$\begin{bmatrix} \frac{V_{i,x}}{l_i} & \frac{V_{i,y}}{l_i} & \frac{V_{i,z}}{l_i} & -\frac{V_{i,x}}{l_i} & -\frac{V_{i,y}}{l_i} & -\frac{V_{i,z}}{l_i} \end{bmatrix}^T \begin{bmatrix} q_i \end{bmatrix} = \begin{bmatrix} f_{a,x} \\ f_{a,y} \\ f_{a,z} \\ f_{b,x} \\ f_{b,y} \\ f_{b,z} \end{bmatrix} \tag{3.4}$$

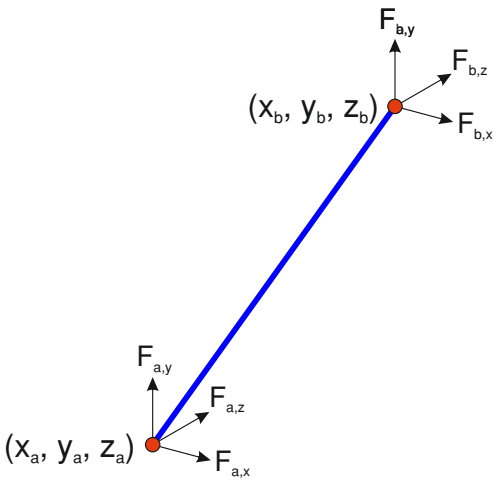


Figure 3.2: Truss notation

3.1.2 Extending the basic formulation to consider stability and joint costs

Solutions for layout optimisation are likely to contain numerous short members if a sufficiently fine nodal discretisation is used. To penalise the presence of short members the formulation can be modified slightly so that each member length is extended by a constant value. The implementation of this simply requires that $\mathbf{l^T}$ in Eqn. 3.1a is replaced with $\mathbf{l_j^T}$ which is a vector of the member lengths plus a constant, $\mathbf{l_j^T} = \{l_1 + k, l_2 + k, ...l_m + k\}$. This will make short members more costly in terms or volume relative to longer members, prompting the optimisation to prefer the latter. The joint cost k can be varied until the desired result is achieved.

Resulting layouts from the layout optimisation procedure may be found to be in unstable equilibrium with the applied loads. The use of multiple load cases will avoid solutions that are mechanisms but there may still be nodes with many compressive elements that are unstable. A method to avoid unstable nodes and produce more robust designs is to apply small disturbing forces at every node (Tyas et al. 2005). This should promote additional bracing members at these nodes or avoid their formation altogether. The implementation involves the addition of six load cases which apply nominal nodal loads in both the positive and negative x, y and z directions (as shown in 3.13). The magnitude of these forces ($\mathbf{f}^x$, $\mathbf{f}^y$ & $\mathbf{f}^z$) is determined by resolving the compressive member forces acting on the nodes using the direction matrices (d_x, d_y & d_z) which is then multiplied by a constant r. A recommended range for this constant is between 1 and 2%.

$$\min \; V = \mathbf{l_j}^{\mathrm{T}}\mathbf{a} \tag{3.5}$$

$$\mathbf{Bq}^{\alpha} = \mathbf{f}^{\alpha} \tag{3.6}$$
$$\mathbf{Bq}^{p+1} = \mathbf{f}^{x} \tag{3.7}$$
$$\mathbf{Bq}^{p+2} = -\mathbf{f}^{x} \tag{3.8}$$
$$\mathbf{Bq}^{p+3} = \mathbf{f}^{y} \tag{3.9}$$
$$\mathbf{Bq}^{p+4} = -\mathbf{f}^{y} \tag{3.10}$$
$$\mathbf{Bq}^{p+5} = \mathbf{f}^{z} \tag{3.11}$$
$$\mathbf{Bq}^{p+6} = -\mathbf{f}^{z} \tag{3.12}$$

$$\left.\begin{aligned} \mathbf{f}^{x} &\geq rq^{\alpha}d_{x} \\ \mathbf{f}^{y} &\geq rq^{\alpha}d_{y} \\ \mathbf{f}^{z} &\geq rq^{\alpha}d_{z} \end{aligned}\right\} \quad \alpha = 1, ..., p \tag{3.13}$$

$$\left.\begin{aligned} a_i &\geq q_i^{\alpha}/\sigma^{+} \\ a_i &\geq -q_i^{\alpha}/\sigma^{-} \end{aligned}\right\} \quad \begin{aligned} &i = 1, ..., m; \\ &\alpha = 1, ..., p+6 \end{aligned} \tag{3.14}$$

3.1.3 LP algorithms and duality

Two prominent linear programming algorithms are the simplex and interior point (IP) methods. The difference between these algorithms lies in the way in which they search for the optimum solution, and can be demonstrated using a simple linear programming problem shown in eqn. 3.15. Here the objective is to simply maximise the sum of two variables (x_1 and x_2) subject to the linear constraints in eqn. 3.16. The problems are plotted in Figs. 3.3a & 3.3b showing the search paths taken by the simplex and interior point algorithms respectively.

The feasible region bounded by the inequality constraints forms a polytope, which for the two variable example here is a polygon. For a three variable problems this would be a polyhedron and for increasing variables the polytope will have the equivalent number of dimensions, which would be of course impossible to visualise.

The simplex method is the more intuitive of the two procedures and works on the concept that the optimum solution will lie on a turning point, which is the intersection of two or more linear constraints. The simplex algorithm in its simplest form will start at a known vertex of the feasible space bounded by the constraints and then evaluate the objective function of the neighbouring vertices. In the case of Fig. 3.3a the starting point is (0,0) and there are two paths the algorithm can choose (0,4) and (6,0). Obviously the latter yields a higher value from the objective function (to maximise y) so the first iteration will be to (6,0). Subsequent iterations are then performed using this logic, ensuring the path that maximises the objective function is taken. When it is no longer possible to yield an increase in the objective function the algorithm is said to have converged on the optimum solution.

The interior point method, so called because it searches within the feasible space, is a widely used method both linear and non-linear optimisation problems. The edges of the polytope are represented by continuous barrier functions which reduce the objective function to negative infinity at the edges. These are usually logarithmic functions as their

influence on the objective function will only be in close proximity to the polytope edges without creating any discontinuities.

When comparing the two methods it is important that the nature of the particular problem is considered. Generally the simplex will perform well (and perhaps even better than IP methods) for small and medium size problems. However it will often perform poorly for large scale problems in comparison to the interior point method. This is because an increasing number of variables will result in a polytope with more edges. In fact in the worst case the simplex method exhibits exponential time complexity, which means for a given number of variables $\mathbf{n}$ the time taken to find a solution is in the order of $\mathbf{2^{nP}}$, where $\mathbf{P}$ is a power that is dependent on the problem. The interior point method however is characterised as a polynomial time method with a much more favourable time complexity in the order of $\mathbf{n^P}$.

As the interior point method uses logarithmic barrier functions the solution will not lie exactly on the edge of the polytope. Thus for the plastic layout optimisation problem no member areas will take a value of exactly zero. This necessitates the need for a filter to remove members with a cross-sectional area close to zero from the final layout. The simplex method will find an exact solution but it too will likely produce a layout that contains members with a very small area. Although these are part of a valid solution rather than just residuals, in reality their presence has no significant impact on the final layout and thus should also be filtered (e.g. <0.1 mm diameter).

$$\max \; y = x_1 + x_2 \tag{3.15}$$

Subject to:

$$\begin{aligned} x_1 + 5x_2 &\leq 20 \\ 3x_1 + 6x_2 &\leq 30 \\ 4x_1 + 3x_2 &\leq 27 \\ 5x_1 + 2x_2 &\leq 30 \end{aligned} \tag{3.16}$$

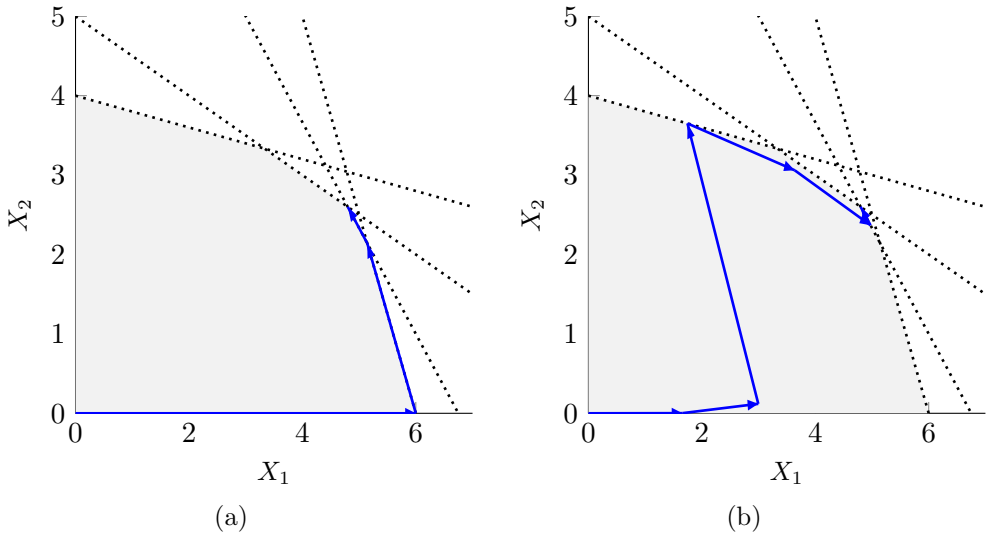


Figure 3.3: Search paths taken by (a) the simplex and (b) the interior point methods

All linear programming problems can be converted from their original formulation (referred to as the primal) to an equivalent problem referred to as the dual. The interpretation of the

dual depends on the particular linear programming problem considered. For the volume minimisation problem in eqn. (3.1a) the dual is to maximise the virtual work W done by the applied external forces f. This is shown in eqn. (3.17) where a constraint on the maximum permissible strain for each member i can also be derived (eqn. ()3.18)). The solution obtained from the dual is the nodal displacements u.

$$\max \; W = f^T u^\alpha \tag{3.17}$$

subject to

$$-\frac{1}{\sigma^-} \leq B_i^T u_i \leq \frac{1}{\sigma_i^+} \tag{3.18}$$

3.1.4 Utilizing duality to reduce the problem size - 'member adding'

The use of a fully connected ground structure becomes computationally expensive for large-scale problems. This is because as the number of nodes n in a problem grows, the number of potential members quickly becomes large (since there will be $n(n-1)/2$ connections in a fully connected ground structure). As the majority of potential members will not be represented in the final design (i.e. will have zero area), representing all potential members becomes rather inefficient. Even in the simple example shown in Fig. 3.1 out of the 210 members in the ground structure only 10 are present in the final solution)

Instead of using a fully connected ground structure a method originally developed by Gilbert and Tyas (2003) starts with a comparatively sparse ground structure (e.g. with only adjacent nodes interconnected). Additional members are then adaptively added in an iterative solution procedure. The Michell-Hemp optimality criteria (which imposes limits on the virtual strain ϵ_i which can be experienced by each potential member i), is used to judge whether to admit additional members to the ground structure:

$$-\frac{1}{\sigma} \leq \epsilon_i \leq -\frac{1}{\sigma}, \quad i = 1, ..., m \tag{3.19}$$

Where σ is the limiting material stress. The virtual strains are easily calculated from the nodal displacement values in the dual LP problem. To ensure the problem size does not grow too rapidly by admitting all potential members that violate this criteria, only members most violating the above criteria are added initially in the procedure (Gilbert and Tyas 2003). The procedure terminates when no potential members violate the criteria, ensuring that the solution obtained is the same as would have been obtained using a fully connected ground structure. The procedure was extended to allow multiple load cases and unequal limiting tensile and compressive stresses to be handled by Pritchard et al. (2005).

3.2 Practical considerations

The buckling response of solid circular bars is intrinsically non-linear and thus not considered in the linear programming optimisation; instead it is considered in a post-processing step. The relevant Euler buckling relationship, which must be satisfied for all compressive

members, is as follows:

$$a_i \geq \sqrt{\frac{4q_i^-(k_{eff}l_i)^2}{\pi E}} \qquad i = 1, ..., m \tag{3.20}$$

where a_i, q_i^- and l_i are respectively the area, compressive force and length of member i and k_{eff} is the effective length factor. Finally, E is the elastic modulus of the material.

Any members that violate this criterion are resized.

As layout optimisation may yield a solution with very thin members that are not suitable for manufacture, a simple minimum area constraint (6.5) must also be enforced. Any member below a set threshold value is resized to this value:

$$a_i \geq a_{min} \qquad i = 1, ..., m \tag{3.21}$$

where a_{min} is the minimum permitted cross-sectional area.

3.2.1 Including rigid shell structures

In addition to truss bar members, rigid shell structures can also be included to facilitate the modelling of more realistic scenarios. For example, a component to be optimised will often have to integrate into a larger assembly of components, and, to model component interfaces, rigid shell structures can conveniently be employed. Triangular elements are used to form the rigid shell structure in this formulation, with the locations of these prescribed prior to the optimisation (Fig. 3.4). A connection point is created at the centroid of each triangular element forming the rigid shell structure, which can accept multiple truss connections as well as the application of external forces. Because the shell structure is rigid, equilibrium needs only to be considered at a single node. All external forces and moments applied at other nodes can be converted into a single set of forces and moments applied at this node. Local equilibrium is then enforced at this node in Eqn. 3.23 where d_i is the vector between the equilibrium node 'e' and the connection node 'j' of element 'i' to the shell.

$$d_i = (x_j - x_e, y_j - y_e, z_j - z_e) \tag{3.22}$$

$$\begin{bmatrix} \frac{V_{i,x}}{l_i} \\ \frac{V_{i,y}}{l_i} \\ \frac{V_{i,z}}{l_i} \\ \frac{V_{i,x}}{l_i}(d_{i,y} + d_{i,z}) \\ \frac{V_{i,y}}{l_i}(d_{i,x} + d_{i,z}) \\ \frac{V_{i,z}}{l_i}(d_{i,x} + d_{i,y}) \end{bmatrix} \begin{bmatrix} q_i \end{bmatrix} = \begin{bmatrix} F_{s,x} \\ F_{s,y} \\ F_{s,z} \\ M_{s,x} \\ M_{s,y} \\ M_{s,z} \end{bmatrix} \tag{3.23}$$

This can then be expanded to ensure global equilibrium is satisfied by including all elements that connect to the rigid shell structure. This global matrix B_s is then combined with that for truss equilibrium in Eqn. (3.1b) to form the overall global equilibrium matrix shown in Eqn. (3.24). F_s are the forces and moments acting on the rigid shell structure. Note that as there are no deformable elements there are no terms in the inequality matrix for the rigid shell structure.

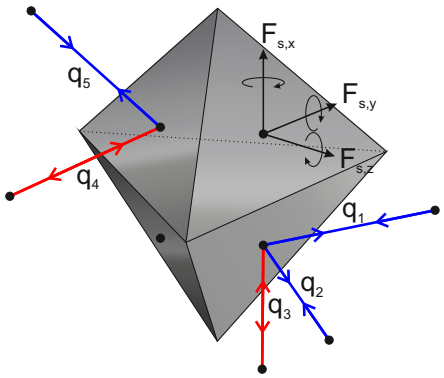


Figure 3.4: Member forces and applied external forces and moments acting on a rigid shell structure are resolved at a single node

$$\begin{bmatrix} \mathbf{B} \\ \mathbf{B_s} \end{bmatrix} \left[\mathbf{q}^{\alpha}\right] = \begin{bmatrix} \mathbf{f}^{\alpha} \\ \mathbf{f_s}^{\alpha} \end{bmatrix} \tag{3.24}$$

3.3 Application of layout optimisation to benchmark problems

As discussed in Chapter 2, layout optimisation can provide a very good estimate for the exact solution for 2-D problems obtained analytically. Because the analytical solutions for 2-D problems are continua like truss structures, with an infinite number of joints, the solution yielded from layout optimisation should be closer when a larger number of potential members are considered. By performing multiple optimisations with increasing numbers of potential members, a very good estimate of the volume with an infinite number of members can be obtained using the power law extrapolation scheme described by Darwich et al. (2010a).

This scheme is here applied to two benchmark cases which will be explored later in this thesis; the cantilever in a half strip domain and the Messerschmitt-Bölkow-Blohm (MBB) beam. From numerous optimisations, V_∞, k and α are determined by minimising the sum of the squares of the errors.

$$V_{n_s} = V_\infty + k\left(\frac{1}{n_s}\right)^{-\alpha} \tag{3.25}$$

3.3.1 Cantilever with half strip domain

The analytical solutions for the half strip cantilever problem (shown in Fig. 3.5 are determined in Graczykowski and Lewiński (2010) for various positions of the applied load, aspect ratio of the design domain and ratio of tensile to compressive strength. The analytical volume V is computed in Eqn. 3.26, where P is the applied vertical load, a is the width of the design domain and σ_T is the limiting tensile strength. The values for the ratio V/V_0 is tabulated in Graczykowski and Lewiński (2010) for various combinations

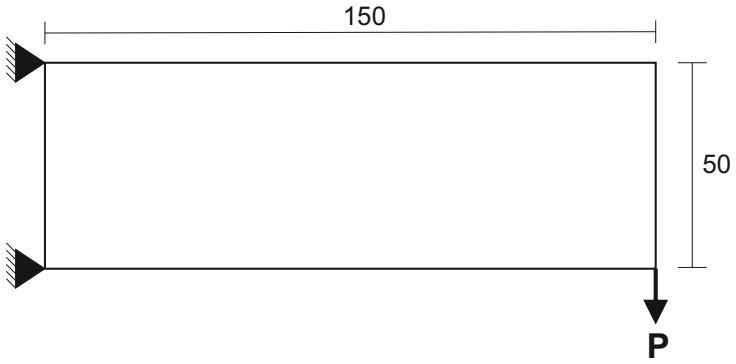


Figure 3.5: The half strip cantilever problem. All dimensions are in mm

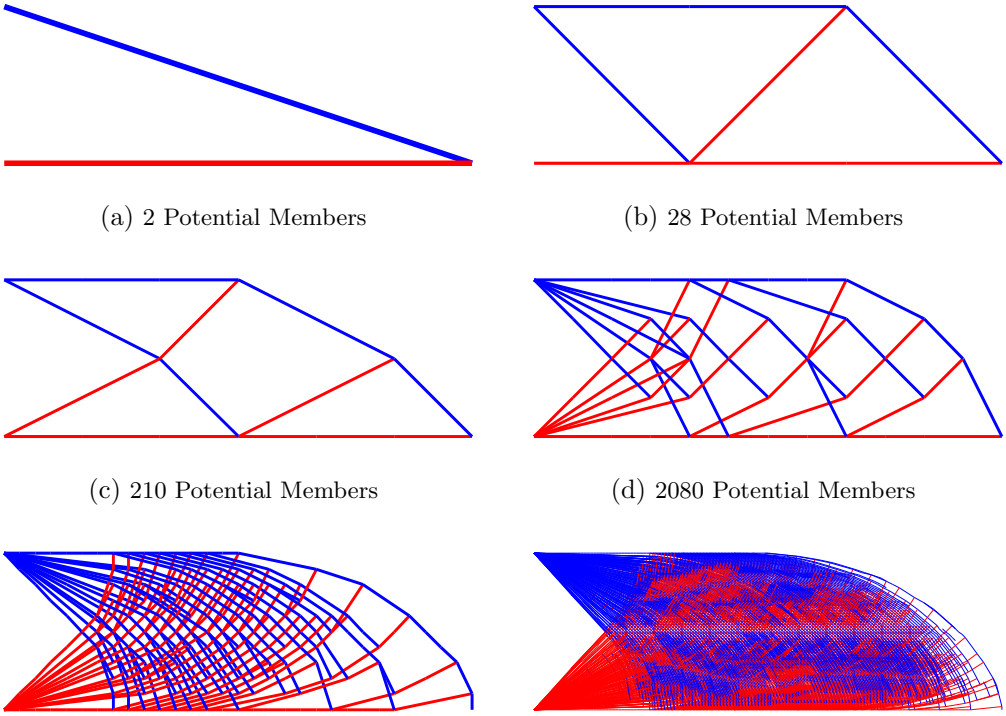
Table 3.1: Optimisations performed for the cantilever problem using a square nodal grid of varying density. The layouts of the first six entries are shown in Fig. 3.6

Nodal Spacing (mm)	No. of Nodes	No. of Potential Members	Volume cm^3	Volume V/V_∞	Error to analytical ($\Delta\%$)
-	3	2	112.82	1.3725	37.23%
50.00	8	28	89.07	1.0835	8.34%
25.00	21	210	86.10	1.0474	4.73%
12.50	65	2080	84.00	1.0218	2.17%
5.00	341	57,970	82.84	1.0077	0.76%
1.00	7701	29,648,850	82.28	1.0009	0.08%
0.83	11041	60,946,320	82.28	1.0009	0.07%
∞	-	-	82.21	1.0000	-0.01%

of aforementioned variables. For the problem shown in Fig. 3.5 the value for the ratio (V/V_0) is 13.8455970. With an applied load of 100kN and a limiting tensile stress of 842 MPa (the justification for this stress value is covered in Chapter 4) the exact volume is 82.2185096 cm^3

The truss layouts obtained using layout optimisation with increasingly more potential members are shown in Fig. 3.6 with the corresponding volumes listed in Table 3.1. The power law curve fitted to this data using the least squares method is shown in Fig. 3.7 whereby through extrapolation, the volume for an infinite number of potential members was determined as 82.2062662 cm^3. The difference of this to the exact analytical solution is just 0.01%.

$$V = \left(\frac{V}{V_0}\right)\frac{Pa}{\sigma_T} \tag{3.26}$$



(a) 2 Potential Members

(b) 28 Potential Members

(c) 210 Potential Members

(d) 2080 Potential Members

(e) 57,970 Potential Members

(f) 29,648,850 Potential Members

Figure 3.6: Layouts obtained for the cantilever problem from optimisations performed using a square nodal grid of varying density. Tensile and compressive members are red and blue respectively. Refer to Table 3.1 for volumes

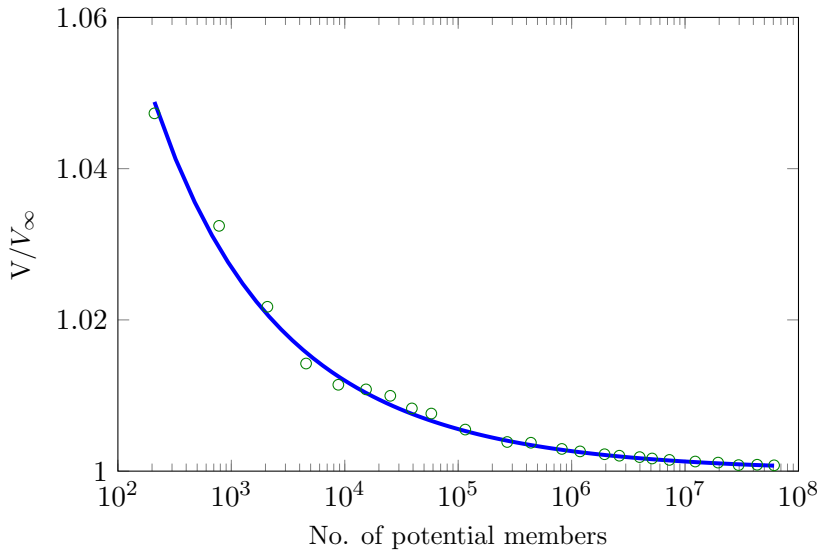


Figure 3.7: Relative volume against no. of potential members

3.3.2 MBB beam

The analytical solutions for the MBB problem (shown in Fig. 3.8) are determined in Lewiński et al. (1994) for various aspect ratios of the design domain. The analytical

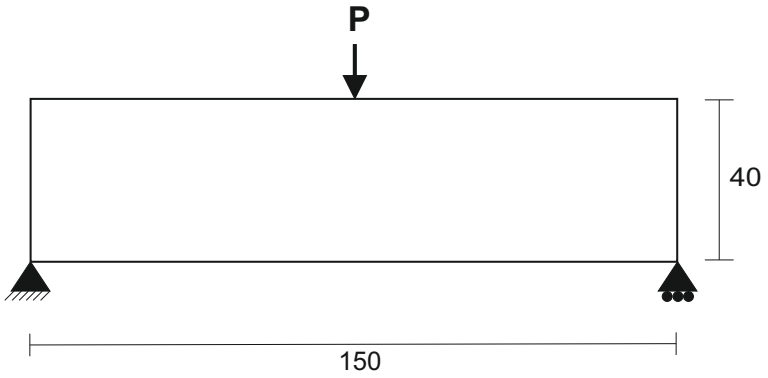


Figure 3.8: The MBB problem. All dimensions are in mm

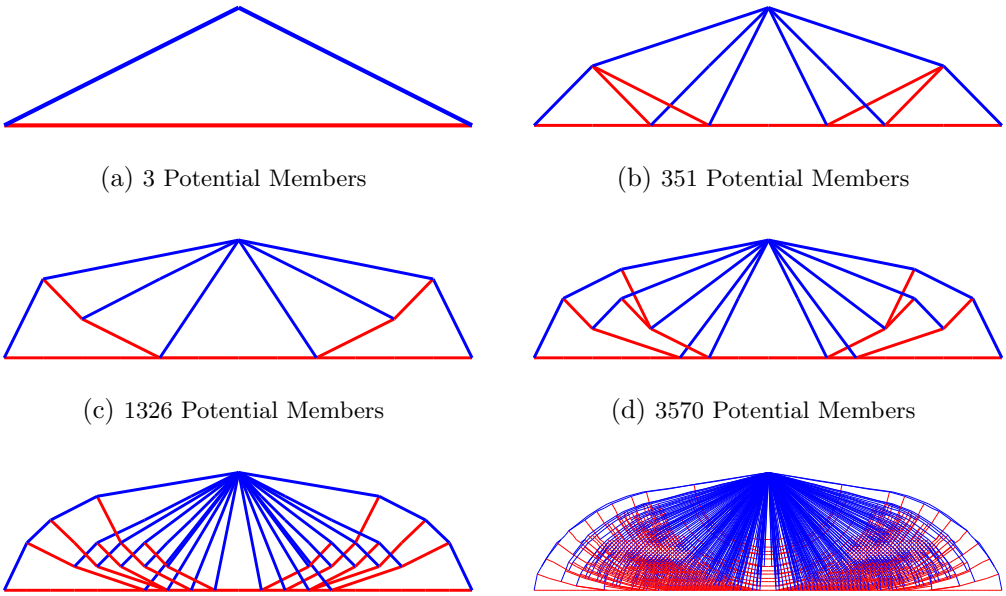
Table 3.2: Optimisations performed for the MBB problem using a square nodal grid of varying density. The layouts of the first six entries are shown in Fig. 3.9

Nodal Spacing (mm)	No. of Nodes	No. of Potential Members	Volume cm^3	Volume V/V_∞	Error to analytical ($\Delta\%$)
-	3	3	42.76	1.1919	19.21%
20.00	27	351	38.00	1.0594	5.96%
13.33	52	1326	37.41	1.0429	4.31%
10.00	85	3570	36.72	1.0236	2.38%
8.00	126	7875	36.40	1.0147	1.49%
1.33	3751	7,033,125	35.94	1.0020	0.22%
0.80	10251	52,536,375	35.90	1.0009	0.11%
∞	-	-	35.87	1.0000	0.02%

volume is calculated using Eqn. 3.27 where P, r and σ_T are the applied load, aspect ratio and limiting tensile stress respectively. $\hat{\Phi}$ is the dimensionless weight and is calculated based on the dimension of the design domain. For the problem in Fig. 3.8 the value of $\hat{\Phi}$ is 7.5497517. Thus with an applied load of 100kN and a limiting stress of 842 MPa the analytical solution for the MBB problem here is 35.8658039 cm^3.

$$V = \frac{\hat{\Phi} P r}{\sigma_T} \tag{3.27}$$

The layouts and corresponding volumes obtained using layout optimisation with an increasing number of potential members are shown in Fig. 3.9 and Table 3.2 respectively. The curve obtained using the least square method is shown in Fig. 3.10, where V_∞ was found to be 35.8725426 cm^3, just 0.02% different to the exact analytical solution.



(a) 3 Potential Members

(b) 351 Potential Members

(c) 1326 Potential Members

(d) 3570 Potential Members

(e) 7875 Potential Members

(f) 7,033,125 Potential Members

Figure 3.9: Layouts obtained for the MBB problem from optimisations performed using a square nodal grid of varying density. Tensile and compressive members are red and blue respectively. Refer to Table 3.2 for volumes.

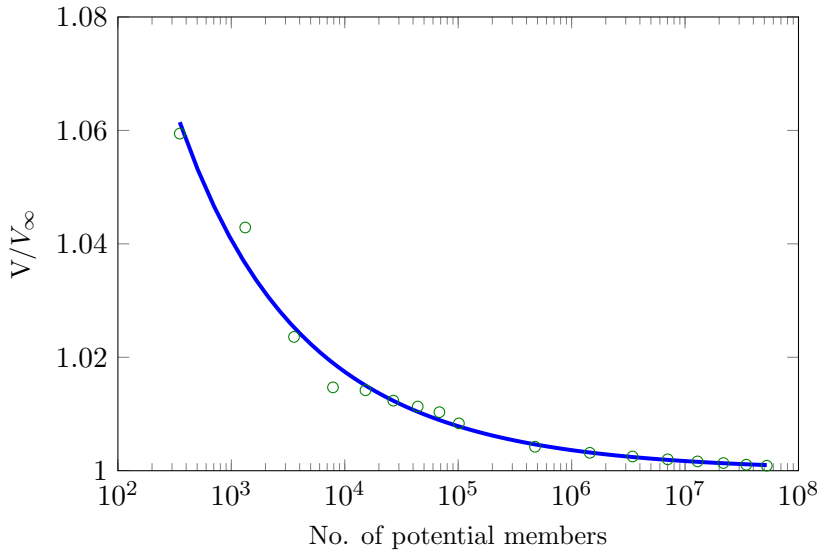


Figure 3.10: Relative volume against no. of potential members

3.4 Chapter conclusions

- The layout optimisation formulation detailed herein can solve large scale 3-D problems with multiple load casing and will yield a structure composed of truss members.
- Complexity of the resulting layout can be limited by imposing a heuristic joint cost penalty which makes short members more costly in the optimisation.
- Rigid shell elements can be included in the problem to facilitate the modelling of

more realistic scenarios.

- Small disturbing forces can be applied as additional load cases to the nodes to promote a layout that is more likely to be stable when in equilibrium.
- Layout optimisation solutions to the cantilever and MBB benchmark problems with very small margin of error to those obtained analytically (<0.02%), which is consistent with what is claimed in the literature.
- Although these very complex layouts cannot be manufactured directly, they do provide a useful reference volume for benchmarking more practical layouts.
- Practical layouts, which may be more heuristic to derive, can be deemed acceptable if within a predefined percentage of the reference volume. This has been set at 20% for the cases explored in later chapters of this thesis.
- The 2-D cantilever and MBB problems explored herein will form the basis for the case studies in the later chapters.
- This thesis made use of an existing C++ program developed at the University of Sheffield to solve layout optimisation problems using the MOSEK APS (2014) interior point programming library. The problem initialisation, post processing and manipulation of the solution data and the creation of the data required to automatically generate a solid CAD model were performed using a MATLAB script which is included in the appendix.

Chapter 4

Realisation of layout optimised structures through EBM additive manufacture

4.1 Creation of a 3-D solid model

The resulting line model produced using layout optimisation will first need to be transformed into a 3-D solid model. This will simply involve lofting a cross-section with an area equal to that determine in the optimisation along the element lengths . Surfaces are defined using Non-Uniform Rational B-Splines (NURBS). A single solid can then be formed using boolean operations which will find the intersection of two or more solids, trim the excess and then effectively stitch them together. The result of this is shown in Fig. 4.1b.

However, given that numerous elements will often converge at a node in a layout optimisation solution, there will often be a significant amount of overlapping in the vicinity of a joint. To avoid increased stresses at such locations additional material must be introduced. Eqn. (6.6a) is evaluated at each node to calculate the radius of the joint in proportion to the area and the number and orientation of the elements connected to the node. This is shown schematically in Fig. 6.4 and implemented in Fig. 4.1c.

A NURBS model offers a great deal of flexibility for converting to various CAD file types. For additive manufacturing the standard file format is SteroLithography (STL), a triangular mesh surface representation. The conversion from NURBS to STL is shown in Figs. 4.1c & 4.1d respectively.

$$r = \sqrt{\frac{1}{\pi}\left(a_i + \sum_{k=1}^{m} a_{j,k}\left\{\hat{i}\cdot\hat{j_k}\right\}\right)} \tag{4.1}$$

Where $\hat{i}$ and a_i is the unit vector and area of the master member respectively, and $\hat{j_k} = \{j_1, j_2, ...j_m\}$ and $a_{j,k} = \{a_{j,1}, a_{j,2}, ...a_{j,m}\}$ are the unit vectors and areas of all the other members connected to the node. This equation is evaluated with every member as the 'master' member, with the largest value for the radius being selected.

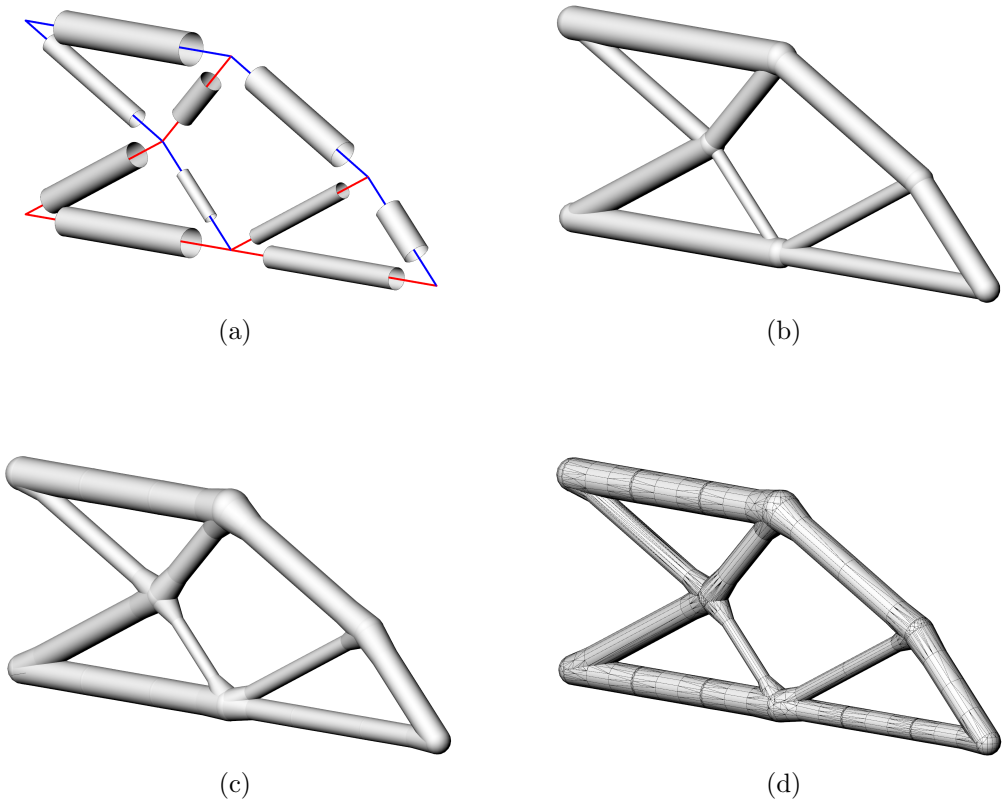


Figure 4.1: Generation of a solid model from a line model. (a) Surface lofting (b) Single solid surface after boolean operation (c) Single surface with expanded joints (d) Conversion to an STL mesh

4.2 File preparation for additive manufacture

All additive manufacturing methods require a 3-D model to be decomposed into a stack of 2-D layers at a prescribed spacing in a process referred to as slicing. For the EBM process the spacing (or layer thickness) is typically 50 or 70μm. Most slicing programs will require that the input solid model is in the Stereoligrophy (STL) format and that all surfaces are closed, well defined and error free (as discussed in Sec. 2.2.6). This is to ensure that during slicing each layer will have closed, well defined polylines (with the normals points inwards) that represent the boundaries of the region(s) that are to be melted in each layer. But before slicing several procedures must be performed which a discussed in the following subsections.

4.2.1 Placement, orientation and scaling

The first procedure is to decide the optimal position and orientation of the part in the build envelope. This could be to produce parts of the highest quality or to minimise process time. Like most additive manufacturing processes, the weakness of the EBM process is the fabrication of features with negative surfaces. These are surfaces whose normals point downwards e.g. the bottom surface of an overhanging feature. To achieve the best overall surface quality it is necessary to orientate a part such that the total negative surface area is minimised. For truss structures this will involve having all the truss members as closely

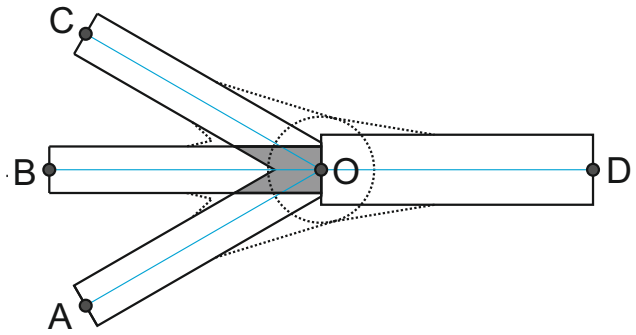


Figure 4.2: Schematic of joint expansion

aligned with the vertical build direction as possible. As shown in Fig. 4.3 the best way to orientate the cantilever structure derived in Chapter 3 was vertically.

Once the part position and orientation has been decided it will need to be scaled to compensate for shrinkage during cooling. The scaling factors recommended by the manufacturer for 70μm layers are 1.0086 in the build plane and 1.0093 in the build direction.

4.2.2 Incorporation of support structures

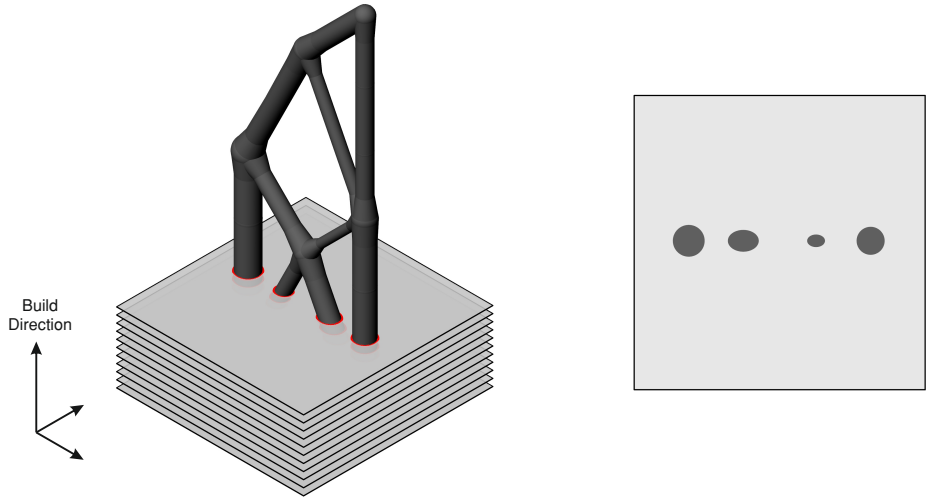
No matter how well the orientation is optimised to minimise the total negative surface area there are likely to be truss members that are at a significant angle to the build direction. The EBM process struggles to fabricate features with negative surface as there is no solid material beneath to aid with heat conduction. The reduced thermal conductivity of the sintered powder results in an accumulation of heat at these regions. Support structures can be placed at negative surfaces to improve the heat transfer away from such features. These are vertical thin wall (~0.5 mm thick) or pin structures (~0.8 mm diameter) which are melted with a high beam velocity to produce a high degree of porosity. The porosity allows them to be easily removed by hand or with pliers post-build although they will leave small stubs on the surface.

4.3 The EBM machine

Although there are many EBM machine models provided by Arcam they are all based on the same template which is shown schematically in Fig. 4.4. The hardware setup can be split into two parts; the electron gun column where the electron beam is generated and controlled and the build chamber where the powdered material is melted. The column and the build chamber are kept under vacuum initally at 1×10^{-6} and 1×10^{-4} mbar respectively. Helium is pumped into the chamber before the electron beam is activated raising the chamber pressure to approximately 1×10^{-3} mbar. The purpose of the helium is to moderate the accumulation of charge in the powder bed caused by the electron beam.

4.3.1 Electron gun column

The electron beam is generated from a tungsten filament which operates at a voltage of 60kV and a current of around 11A, which maintains the filament at an operating temperature of 2200°C. The electron gun assembly is water cooled to keep the temperature



(a) (b)

Figure 4.3: (a) Decomposing 3-D solid model into 2-D slices and (b) one of the layers with the regions to be melted shaded.

below 60°C. The filament component (which is a circular composite plate with the tungsten filament embedded) sits inside a metallic cup structure with the filament itself protruding out of a small hole. This metallic cup, also known as the grid cup, is charged during operation and regulates the emission from the filament by acting as a field barrier. To increase the emission from the filament the voltage of the grid cup is reduced to a value below 1kV, little or no emission.

Once through the grid cup the electrons are channelled and accelerated through a copper anode before heading into the lower column. It is here where three sets of electromagnetic coils are positioned circumferentially with the purpose of influencing the astigmatism, focus and deflection of the electron beam (illustrated in Fig. 4.4). The control software manages these parameters to achieve the desired preheat or melting characteristics. After this stage the electrons pass into the build chamber.

4.3.2 Build chamber and powder deposition

All parts are built on a stainless steel plate which itself sits on top of the build table that can move vertically up and down. Powder is deposited on the start plate by a raking mechanism that fetches from powder stockpiles on either side of the build chamber. The powder stockpiles themselves are gravity fed through a slot at the base of two hoppers. The fetch position of the rake, which is a prism shaped bar with flexible overlapping teeth at the base, is calibrated in such a way that as the rake moves into the powder stockpile the desired amount passes over the apex. The rake then moves back to the other side of the build chamber with the teeth ensuring the fetched powder is evenly distributed 0n the start plate as it passes over. Between the powder stockpiles and the build table on each there are powder sensors that measure the amount of powder that has been fetched. The control software will automatically change the fetch positions as the build progresses based on data from these sensors, particularly as the hoppers begin to empty.

To ensure an even powder distribution on the build area multiple passes are made by the rake for each layer (typically three). But if reduction in total build time is desired

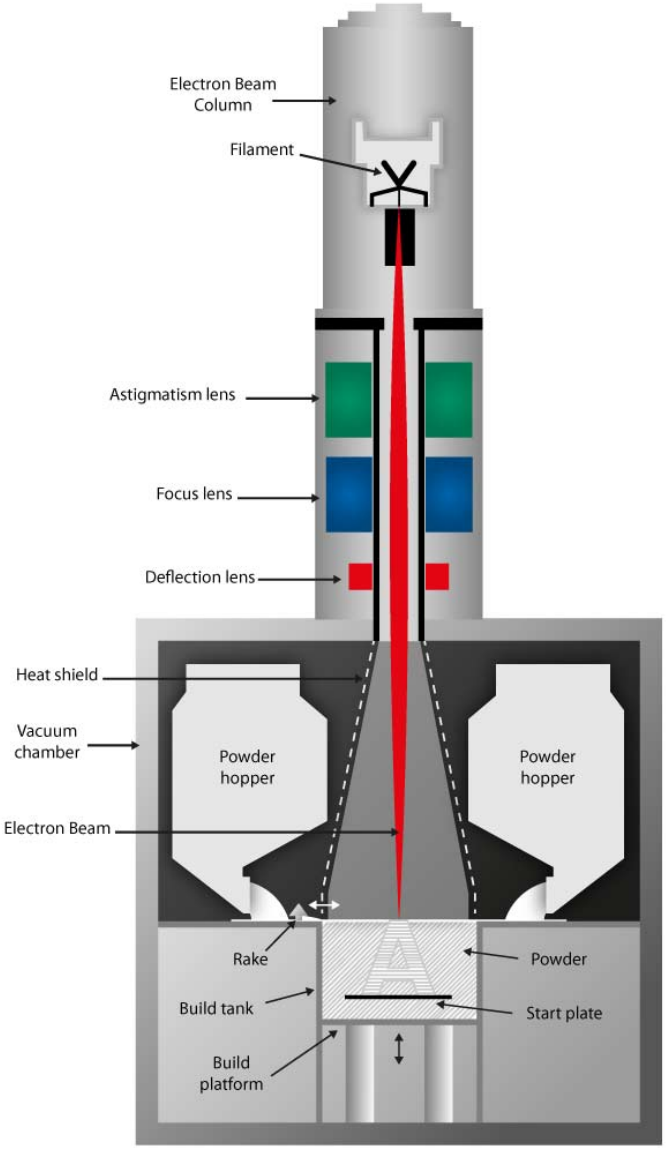


Figure 4.4: Schematic of the Arcam EBM hardware showing both the build chamber and electron gun

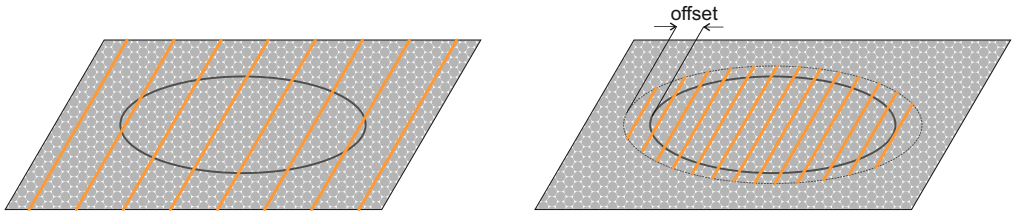
then this can be reduced to just a single pass. The raking systems is very sensitive to the flowability of the powder which itself is a reflection of the powder quality. Powder with particles sizes and shapes that vary greatly as well as the inclusion of contaminants will result in poor flowability. The titanium Ti-6Al-4V powder to be used for fabricating specimens in the later chapters has a particle size range of 40-105 μm.

At the start of the build the start plate is preheated by the electron beam to the desired layer temperature (typically 720°C for Ti-6Al-4V). The build table then drops by decrement equal to the layer thickness (typically 50 or 70μm) where powder is then evenly distributed over the start plate. The beam melts the regions of the layer and the table is dropped by the same decrement and the process is repeated.

4.4 The EBM Process

The EBM process is a hot process that maintains each layer at a target temperature which is achieved by solving a one dimensional thermal model in real time. This model is used to balance the energy input (from preheating and melting) with the energy lost by conductance through the powder bed. The model takes into account the total preheat area, melt area and layer thickness.

4.4.1 Preheating



(a) (b)

Figure 4.5: (a)Preheat I and (b) preheat II phases.

Once the powder deposition stage has completed the entire layer is preheated in two phases - preheat I and preheat II. Preheat I uniformly heats the entire layer to 720°C and sinters the powder. Whilst this is beneficial for relieving residual stresses during the build its real purpose is to prevent a phenomena known as 'smoking'. Powder particles exposed to an electron beam will acquire a negative charge which in turn causes them to be repelled by the electron beam. Puffs of dust or smoke observed during the build is when a region of powder bed has become airborne because of this repulsion. If these powder particles end up in the column then electron gun will be compromised and the build will fail. Sintering the powder in combination with the ionising effect of the helium present under the controlled vacuum conditions helps reduce the occurrence of smoking. The sintering performed by the preheat I phase will permit the beam to scan across the layer with a reduced risk of smoking.

Whilst the first preheat phase will allow the electron beam to scan across the layer smoking may still occur when the beam power is increased for melting. A further preheating stage, preheat II, is thus performed to sinter the powder further. The preheat II phase will act around regions to be melted (the melt area plus and offset distance) which is Shown in Fig. 4.5b.

The degree of preheating does influence the final mechanical properties, particularly yield strength Al-Bermani et al. (2010). Increased preheat temperatures (particuarly above 700°C) have been shown to result in reduced yield strength due to a coarsening of the grains in the microstructure.

4.4.2 Melt

The melt stage is split into two phases; contouring and hatching. The contour phase will melt the perimeter of the cross-section(s) with two or more concentric passes (offset from one another by a set distance). The purpose of this phase is to improve the surface

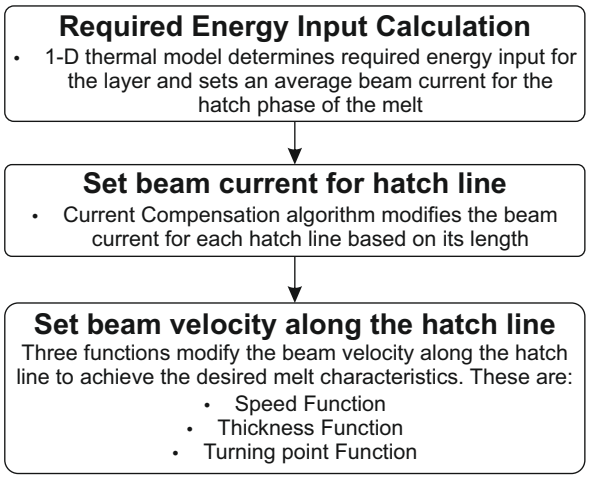


Figure 4.6: Steps involved for calculating the beam current and velocity for the hatch melt phase.

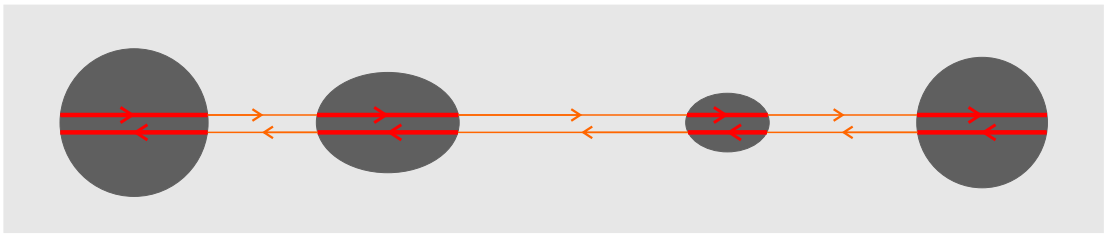
quality which is best achieved by using a small melt pool sustained with a low beam velocity. As this can be quite time consuming the process takes advantage of the near instantaneous deflection of the electron beam by sustaining a number of melt pools (up to 50) simultaneously (as illustrated in Fig. 4.9a). The majority of the melt area will be performed by the hatching phase which melts the cross-section using a back and forth raster pattern as illustrated in Fig. 4.9b.

Whilst the contour phase of the melt is performed with using a constant beam velocity and current throughout the build, the hatch will vary these two parameters based on the thermal history of the melt area. Four separate functions determine the value of these two parameters based on the desired melt pool size, geometry of the of cross-section to be melted and the geometry of the part. These are discussed in detail in the following sub-sections and the whole EBM control process is summarised in Fig. 4.6.

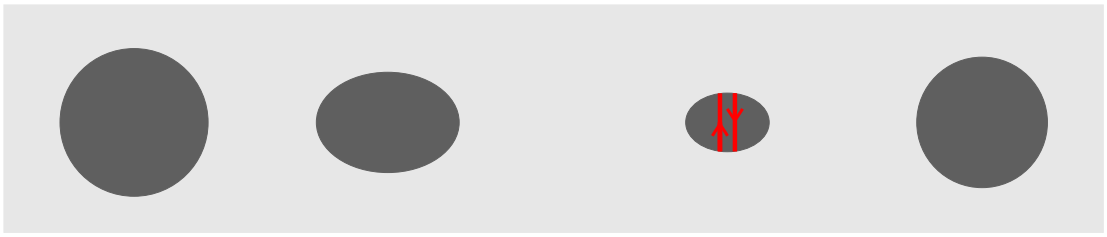
Current compensation algorithm

$$I_c = I\left(1 + \frac{k\,(L - L_0)}{L_0}\right) \tag{4.2}$$

Hatch lines of differing length will have an effect on the melt pool size. This is illustrated with the example shown in Fig. 4.7 which is the slice from Fig. 4.3b with hatching in two different directions. The hatch line length in Fig. 4.7a is much longer than that Fig. 4.7b and thus these two melting paths will have very different thermal histories. Because the longer hatch line will take more time to complete one back and forth route, extra energy must be provided to compensate for the heat dissipation in this time. This is the purpose of the current compensation algorithm, which will set the beam current for a given hatch line length (L) based on the linear relationship in Eqn. 4.2. L_0 and k are the reference length and a constant respectively and take the default values of 45mm and 0.9. The current compensation function is plotted in Fig. 4.8 for these default values and a layer current of 0.9mA.



(a)



(b)

Figure 4.7: (a) Schematic diagram of the EBM process melting overhanging features for (a) a ledge type overhang and (b) truss element. Region where the speed adjustments made by the thickness function is labelled.

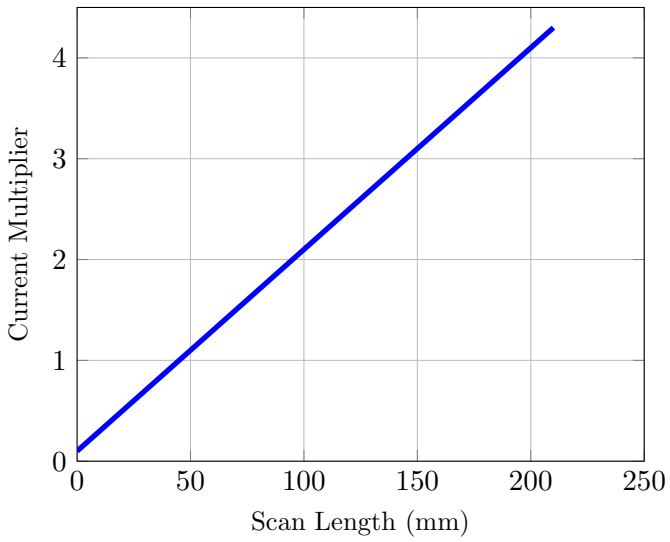
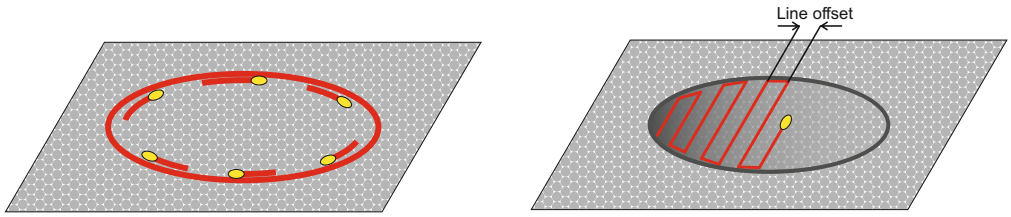


Figure 4.8: Current compensation function for the standard parameters provided by Arcam

Speed function

The purpose of the speed function in the Arcam EBM process is to maintain a constant melt pool size regardless of the supplied current by the electron beam. The speed of the beam is adjusted based on a non-linear relationship between beam current and speed which was derived by Arcam using Finite Element modelling. The beam speed for two beam current values (5 and 20mA) is shown for a range of speed function presets in Fig. 4.10. Increasing the speed function reduces the melt pool size.



(a) (b)

Figure 4.9: (a) Contouring and (b) hatch phase of the melt.

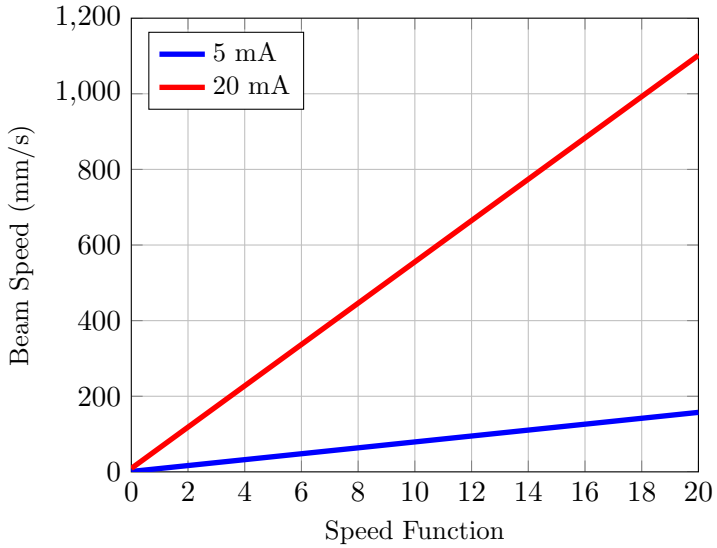


Figure 4.10: Resulting beam speeds with speed function presets 1-20 for 5 and 20 mA

Thickness Function

To account for the lower conductivity of sintered powder beneath negative surfaces the beam energy density must be reduced. With the Arcam EBM process this is achieved by increasing the beam velocity rather than reducing the beam power as the latter would require beam current changes at a rate that the system is not capable of attaining. The beam velocity V is altered as per the inverse exponential function of the thickness T in Eqn. 4.3.

$$\frac{V}{V_0} = 1 + \frac{S_f}{exp\left(E_f(T - T_f)\right) + 1} \tag{4.3}$$

T is defined as the vertical distance between the current layer and the negative surface beneath it. For a ledge overhang (Fig. 4.11a) the velocity will be maximum V_{max} at the negative surface ($T = 0$) and then gradually reduce in subsequent layers until the maximum preset thickness T_{max} is reached where the velocity will return to the original value of V_0. For an angled specimen (Fig. 4.11b) the thickness will vary across each layer and thus so will the velocity. The distance L from the edge that the thickness function is active (darker region shown in Fig. 4.11b) for a specimen at an angle of θ to the build direction is easily calculated using Eqn. 4.4.

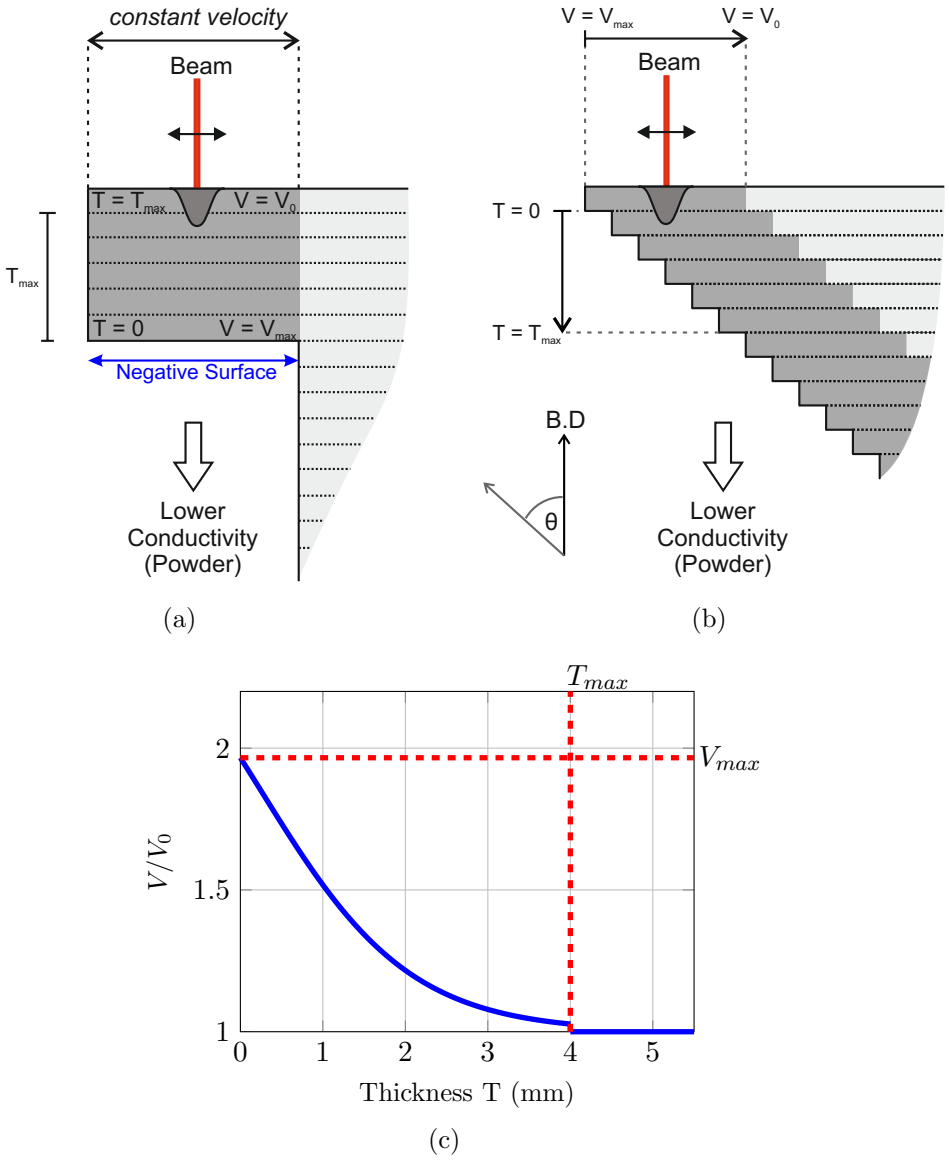


Figure 4.11: Schematic diagram of the EBM process melting overhanging features for (a) a ledge type overhang and (b) truss element at an angle θ to the build direction (B.D). Darker regions show where the thickness function is active. (c) Shows the relative velocity profile with the default thickness function parameters.

$$L = \frac{T_{max}}{tan(\theta)} \tag{4.4}$$

Three tuning parameters known as the Speed Factor (S_f), Exponent Factor (E_f) and Thickness factor (T_f) can be changed to adjust the profile. The Speed Factor influences V_{max} and the Exponent factor influences the steepness of the profile between $T = 0$ and $T = T_{max}$. The Thickness factor influences the coordinates of the inflexion point of the function.

Turning point function

$$S_i = S\left(1 + C_e exp\left(-\left(S\left(E_f d - E_{F2} S\right)\right)^2\right)\right) \tag{4.5}$$

With the back and forth raster pattern of there will be a great deal of heat accumulation at the regions where the beam reverses direction (as illustrated in Fig. 4.9b). The resulting heat accumulation at these turning points there will be a variation in the size of the melt pool as the beam is melting in close proximity to the region that has just been melted on the previous pass. To reduce the heat input at the turning points the beam speed is increased based on Eqn. 4.5 based on the distance d from the edge. S is the beam speed and C_e, E_f and E_{f2} are constants that can be adjusted. For the default settings of $C_e = 0.75$, $E_f = 0.012$ and $E_{f2} = 1 \times 10^{-5}$ the turning point function is plotted for a beam speed of 1000 mm/s.

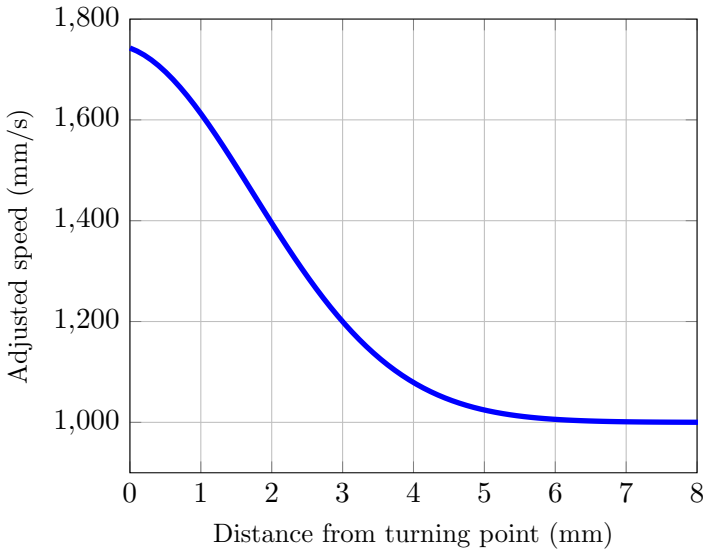


Figure 4.12: Turning point function plot for the standard parameters provided by Arcam

4.5 Post-build

Once the final layer has completed the high voltage unit is powered down and the part is allowed to cool. To assist cooling but in doing so avoiding oxidisation of the titanium powder helium is pumped into the chamber. The time taken for cooling will vary with part size but it is recommended that the chamber is not ventilated with air until the thermocouple beneath the start plate reads a value below 100°C. The part is then extracted and placed in the Powder Recovery System (PRS) to remove the sintered powder that the part is encased in. This apparatus uses high pressured air (6 bar) with titanium particles to sandblast away the sintered powder. If designed correctly the support structures can be removed by hand or with pliers.

The sintered powder recovered in the PRS is easily recycled by using a pneumatic sieve that breaks down sintered powder particles and removes any contaminants.

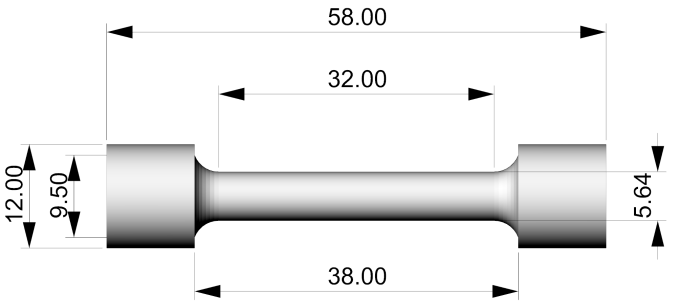


Figure 4.13: Dimensions (in mm) of tensile test specimen

4.6 Material properties of Ti-6Al-4V from EBM process

As noted in the literature the as-built surface finish from the EBM process results in a reduction in strength in comparison to specimens with a machined finish. As all structural specimens were to be tested without any surface finishing it was necessary to determine the strength properties for as-built specimens. Six specimens were fabricated from Titanium Ti-6Al-4V Extra Low Interstitial (ELI) powder, which is supplied by Arcam and would be the powder used for fabricating all layout optimised structures. The dimensions of the tensile test specimens are shown in Fig. 4.13.

As mentioned in Sec. 4.2.1 the ideal build orientation for truss structures is one where the members are closely aligned to the vertical build direction. It has been observed in the literature that the tensile strength properties are slightly lower in the vertical build direction (z) than the horizontal directions along the layer plane (x and y). This slight anisotropy has been attributed to the prior beta grain structure that forms during solidification, which was discussed in Chapter 2. The tensile specimens were thus orientated vertically so that the lower bound for strength properties could be determined.

The results from tensile testing are shown in Table 4.1 which can be compared with the results from a similar study found in the literature (Table 4.2). Although the strength data obtained is slightly higher than that of a similar study in the literature the two can be considered to be in good agreement when considering the variation (2.3% difference between the vertical UTS values). The more conservative estimate of 842 MPa for the Ultimate Tensile Strength from the literature was chosen as the limiting stress for all optimisation problems considered in the following chapters.

Table 4.1: Tensile properties of specimens tested with as-built surface finish

Condition	Sample	Yield Strength (MPa)	UTS (MPa)	Elongation (%)	Area Reduction (%)
As-built	1	-	881	14	39
	2	-	852	8	36
	3	-	851	18	36
	Average	-	861	13	37
Machined	1	818	929	12	49
	2	836	949	14	50
	3	834	944	13	47
	Average	829	941	13	49

Table 4.2: Mechanical properties of Ti-6Al-4V ELI samples tested in as-built condition Khalid Rafi et al. (2012)

Condition	Orientation	Yield Stress (MPa)		UTS (MPa)		Elongation (%)	
		Avg	SD	Avg	SD	Avg	SD
As-Built	Vertical	782	5.10	842	13.8	9.9	1.02
	Horizontal	844	21.6	917	30.5	8.8	1.42
Machined	Vertical	869	7.2	928	9.8	9.9	1.7
	Horizontal	899	4.7	978	3.2	9.5	1.2

4.7 Chapter conclusions

- The standard process parameters provided by the manufacturer Arcam for the functions that determine beam energy input will be used for fabricating layout optimised truss structures. These have been developed with the aim of producing fully dense, geometrically accurate parts with minimal internal defects such as porosity.
- However as these have been developed for general purpose applications they may not be optimised for fabricating truss structures. Thus the presence of internal defects and dimensional accuracy of of the fabricated truss structure will need to be investigated, with the view of modifying these parameters if necessary.
- The ultimate tensile strength of specimens fabricated from Ti-6Al-4V ELI powder was 8.5% lower than those that were machined before testing, which is largely consistent with what is stated in the literature.
- The results from Khalid Rafi et al. (2012) for as-built and machined specimens with the same powder were only 2.2 and 1.4% lower than the values determined herein.
- The more conservative value of 842 MPa will be used as the limiting stress in all optimisation problems investigated in the following chapters.

Chapter 5

Application to the cantilever benchmark problem

5.1 Introduction

This chapter describes the first case study of the application of layout optimisation to a simple 3-D cantilever problem, which has been extended from the 2-D problem explored in Chapter 3. A reference volume was determined using the extrapolation method also detailed in Chapter 3, and was used to determine a practical layout for the 3-D cantilever problem that is within an acceptable margin (20%) of this value. The resulting truss layouts was then fabricated from titanium Ti-6Al-4V ELI powder using the EBM process. X-ray Computed Tomography (XCT) was used to characterise any internal defects such as porosity, which the EBM process is known to produce. These defects are typically ∼100 μm in diameter which should have a greater influence of fatigue life rather that static strength, provided the overall fraction of the pores very low (typically ¡0.2% fro EBM parts (Tammas-Williams et al., 2015)). But the potential for larger defects exists if there is insufficient melting at any stage during manufacture. These defects could compromise the load bearing capacity of the entire structure and thus must be identified. The fabricated truss specimens were then load tested to establish their ultimate load bearing capacity and assess how close this is to the target load.

5.2 Problem Definition

The cantilever benchmark problem explored in Chapter 3 has been extended to the 3-D problem shown in Fig. 5.1. The depth of the design domain has been set equal to the original width of 50 mm. The load (of 100 kN) has been centrally placed along the depth dimension. Four support constraints are now in place of the original two in the planar problem.

As discussed in Chapter 4, and as with the analytical solutions computed in Chapter 3, the limiting tensile and compressive strength was set at 842 MPa. The elastic modulus which was used for the buckling analysis was set at 113.8 GPa (standard for Ti-6Al-4V) and the column effective length factor K was assumed to be 0.699.

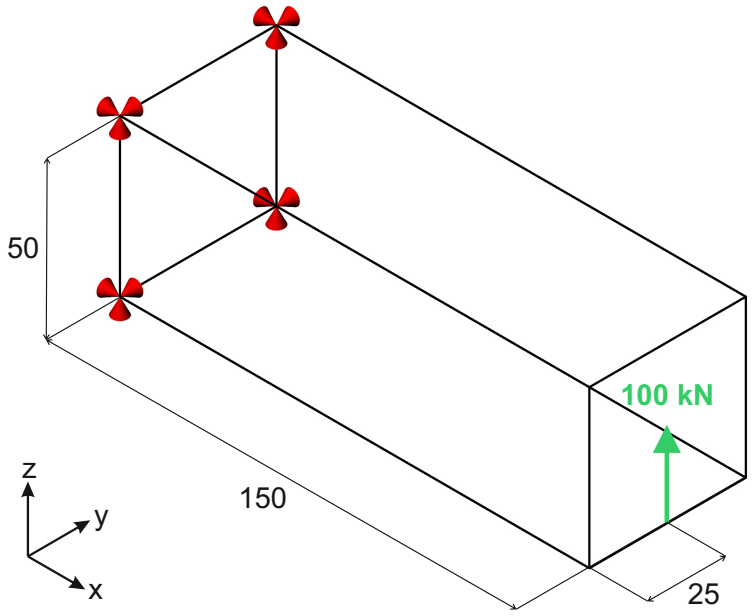


Figure 5.1: The 3-D cantilever problem definition. All dimensions are in mm

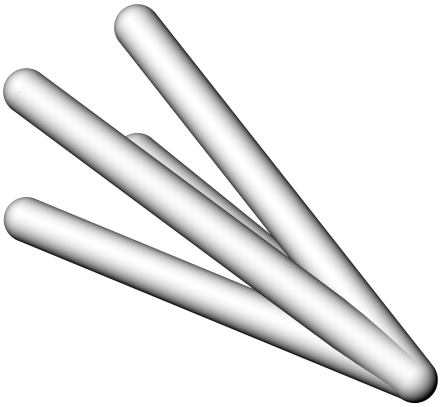
5.3 Determining a reference volume

As outlined in Chapter 3 a very good approximation for the true optimum volume can be obtained from extrapolating the trend between volume and the number of potential members to infinity. This value can then act as a reference for determining a layout that is both practical and within an acceptable percentage of this value, which has been set at 20%. The same procedure is performed here for the problem definition described in Section 5.2. Table 5.1 lists a selection of volumes used in the extrapolation procedure. Renderings of some of these truss layouts are shown in Fig. 5.2.

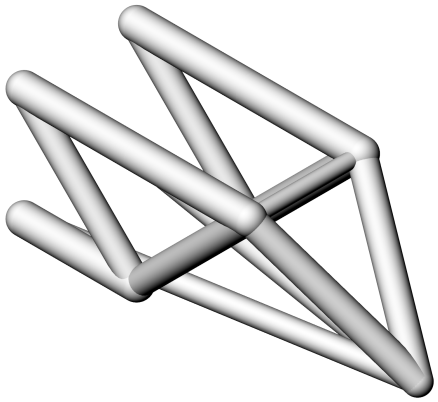
A reference volume of 83.97 cm^3 was determined, 2.1% higher than that of the 2-D analytical solution from Chapter 3.

Table 5.1: Selected volumes resulting from layout optimisation with differing nodal densities

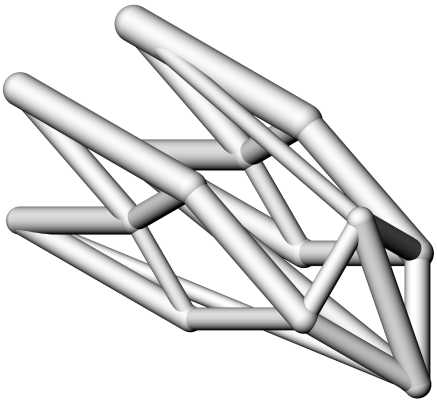
Nodal Spacing (mm)	No. of Nodes	No. of Potential Members	Volume cm^3	Volume V/V_∞
-	5	4	115.80	1.3790
50	16	120	92.54	1.1020
25	63	1953	91.15	1.0855
12.5	325	52,650	87.31	1.0397
10	576	165,600	86.56	1.0308
6.25	2025	2,049,300	85.52	1.0184
2.78	19855	197,100,585	84.43	1.0054
∞	-	-	83.97	1.0000



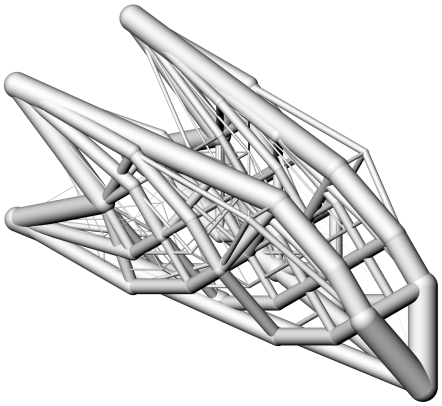
(a) 5 nodes (4 potential members)



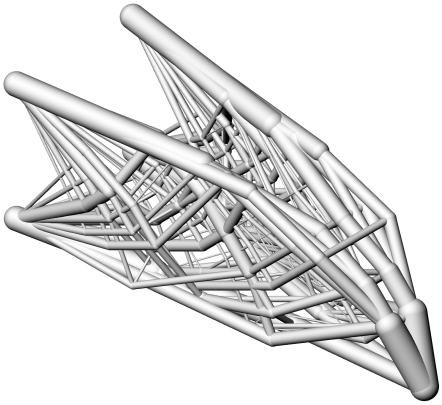
(b) 16 nodes (120 potential members)



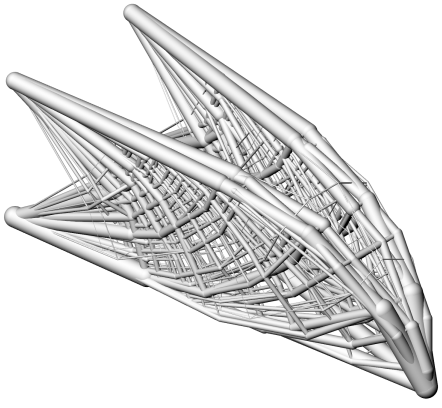
(c) 63 nodes (1953 potential members)



(d) 325 nodes (52,650 potential members)



(e) 576 nodes (165,600 potential members)



(f) 2025 nodes (2,049,300 potential members)

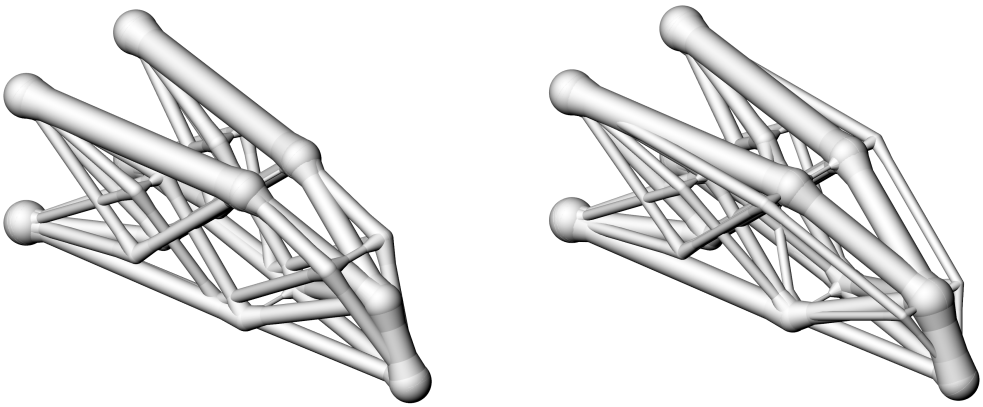
Figure 5.2: Resulting layouts for the 3-D cantilever problem with an increasing number of potential members. The volumes of these layouts are presented in Table 5.1

5.4 Determining a practical layout

With a finer nodal discretisation and hence more potential members the resulting solutions are closer to the true optimum volume. But many of these solutions contain numerous slender members that are both susceptible to buckling and would prove difficult to manufacture. To simplify the topologies without increasing the volume significantly, a joint cost penalty has been introduced. Several values for the joint cost k were applied to the cantilever optimisation problem to assess their influence on the resulting volume before and after members had been resized for buckling and minimum area considerations. These are shown in Table 5.2, where it can be observed that a joint cost of two or three results in a much simpler layout with only a 1.5% increase in volume over the layout with no joint cost penalty. After considering minimum area and buckling constraints, this actually becomes a reduction of 1.7%. A rendering of this layout is shown in Fig. 5.3a which when compared to the counterpart with no joint cost in Fig 5.2d is visually much simpler only having one fifth of the number of members.

Another issue discussed in Chapter 3 is that layout optimised structures can be in unstable equilibrium. A modified formulation that applies six disturbing forces at each node can be implemented to theoretically yield a more robust layout. Another optimisation was performed using the stability formulation, with disturbing forces set at 2% of the total compressive load acting on each node and a joint cost of two. The resulting layout is shown in Fig. 5.3b which has a volume that is just 0.3% higher than that without the use of the stability formulation. However, once the many slender bracing members that have been introduced are resized for buckling and minimum area constraints the increase in volume is actually 3.9%. With a volume that is 11% larger than the reference volume, this layout was selected for fabrication and will henceforth be referred to as Cantilever 2.

The layout involving just four potential members (shown in Fig. 5.2a) was also fabricated. This very simple layout can be seen as analogous to a conventionally designed part for this particular problem. As no members required resizing the final volume of this layout was 115.8 cm^3, 24.1% larger than the Cantilever 2 layout. This layout will henceforth be referred to as Cantilever 1.



(a) (b)

Figure 5.3: Solution using a joint cost k of 2, (a) without the stability formulation and (b) with the stability formulation

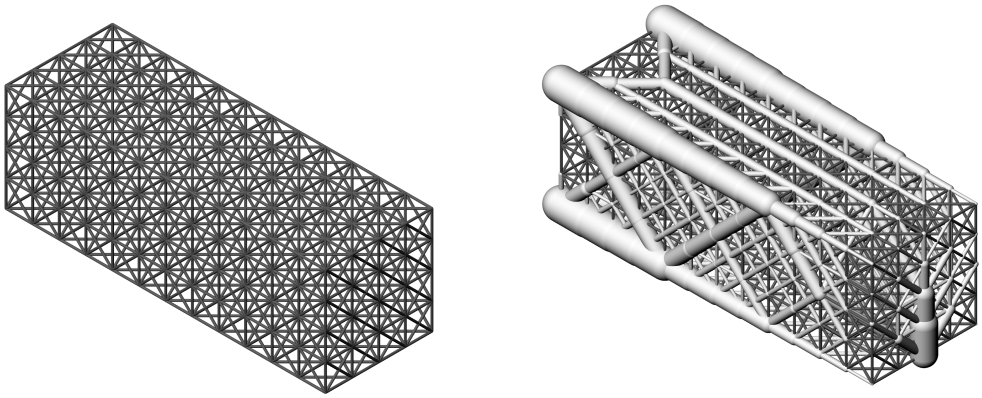
Table 5.2: Resulting volumes with various joint cost penalties k, relative to the reference volume ($V_\infty = 83.97$ cm^3). (*) denotes the result yielded when using the stability formulation

Joint Cost	No. of Members	Volume (V/V_∞)	
		Original	After resizing
0	202	1.0397	1.0876
1	145	1.0449	1.1155
2	39	1.0556	1.0693
3	39	1.0556	1.0693
4	45	1.0578	1.0789
5	51	1.0582	1.1089
2*	64	1.0590	1.1107

5.4.1 Incorporation of a lattice structure

As mentioned previously layout optimised structures may be unstable at equilibrium i.e. cannot sustain any load(s) that deviate even slightly from those considered in the optimisation. As a result one or more nodes may become unstable during deformation leading to a buckling or collapse failure. The stability formulation discussed in the previous chapter aims to prevent the occurrence of unstable structures by applying small disturbing loads at each node. This should promote the occurrence of bracing members or the avoidance of unstable frame geometries altogether. A potential alternative is to include a fixed lattice structure into the optimisation problem. The thin members of the lattice could then provide structural support for all of the nodes of the main structure.

A lattice based solution was also considered by placing a minimum member area constraint on those members that connected adjacent nodes. The area set corresponded to a diameter of 1 mm with the resulting lattice shown in Fig. 5.4a. With the exception of the minimum area constraint the optimisation problem was the same as before and the resulting layout is shown in Fig. 5.4b.This layout has a total volume of 119.3 cm^3 and is referred to as Cantilever 3.



(a) (b)

Figure 5.4: (a) The lattice structure used in the optimisation and (b) the resulting layout.

5.5 Load Testing Arrangement

Load testing was performed using a universal testing machine in an asymmetric three point flexural bend test arrangement (which was easier to setup than a true cantilever arrangement). Each test specimen rested on two pairs of titanium blocks at the tip and at the base. The surfaces in contact with the test specimen had grooves that were 0.5 mm larger in radius than that of the circular contact points on the specimen. This offset was in place to account for the presence of a Polytetrafluoroethylene (PTFE) sheet between the specimen and the grooved blocks. PTFE sheet was also used between the grooved and non-grooved titanium block supports. A further grooved titanium block (with the same enlarged radius) was used as an interface between the specimen and the loading platen of the universal testing machine, again with the contact surfaces lined with PTFE sheet. All contact surfaces were polished.

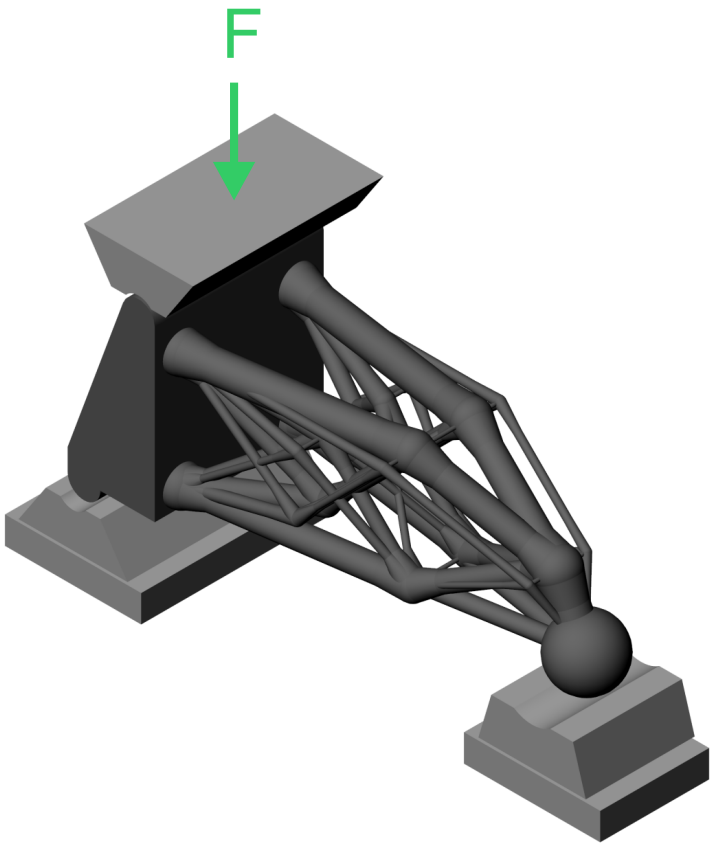


Figure 5.5: The cantilever problem definition: overview.

To achieve an equivalent tip load of 100 kN the corresponding load applied by the testing machine would have needed to be 720 kN.

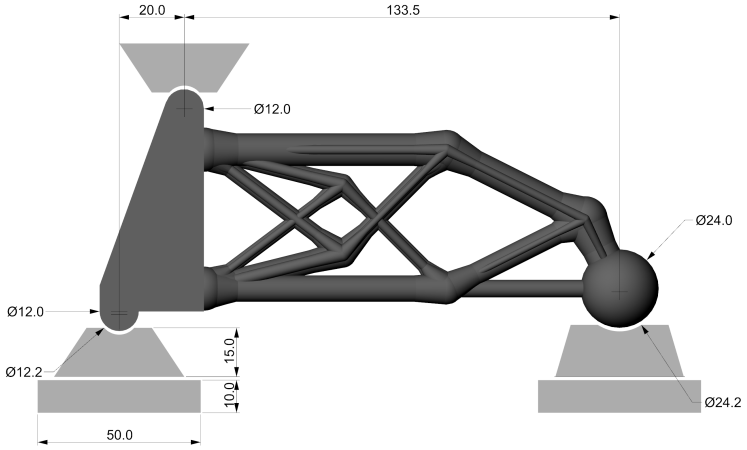


Figure 5.6: The cantilever problem definition: dimensions (all in mm).

5.6 Fabrication

All specimens were fabricated using an Arcam S12 EBM machine using the standard process parameters detailed in Chapter 4. At the time of fabrication the machine that was originally intended to manufacture each specimen became unavailable indefinitely. The alternative route was to use another Arcam S12 EBM machine which had a smaller build envelope, only permitting builds of up to 180 mm in height. As each specimen was 205 mm in height a scaling factor of 0.85 was applied to the length dimension in all three Cartesian axes. The truss areas were therefore reduced by a factor of $0.852^2 = 0.7225$.

The dimensions for the load test arrangement in Fig. 5.6 account for the scaling down of the three specimens.

5.7 Assessing internal defects using X-ray Computed Tomography (XCT)

In medicine X-ray Computed Tomography (XCT or CT in medicine) scans are commonly used for diagnosis by providing non-intrusive internal imaging of a patient's anatomy. During a CT scan a patient is subject to a low-level of x-ray radiation. The technology is based on the principle that materials of higher density will more readily absorb x-rays. So by positioning a patient between an x-ray source and a sensor array, which can then rotate around the patient, the various internal tissues of a patient can be visualised in 3-D. These can then be isolated into differing tissue types such as bones, specific organs etc. In industry the same concept is applied for Non-Destructive Testing (NDT) of components, although without patient comfort or health concerns. As defects, voids, inclusions and any other anomalies will have a different density to that of the bulk material their size, location and shape can be computed. This makes it an effective tool for quality control and assessing material quality.

As reported in the literature the EBM process does produce parts that have a certain degree of porosity. The general conclusions from porosity studies are that the majority of the pores will be relatively small ($\sim$100 μm in diameter) and will only be an issue for fatigue life and not the static mechanical properties. However, the porosity distribution in more

geometrically complex specimens is seldom investigated and the potential for larger melt defects is known to occur with the EBM process if there is a lack of fusion between powder particles. With many thin truss members present in an optimised structure, a significantly sized pore or colony or pores could influence the load bearing capacity of such members. XCT scans were therefore performed on several optimised truss specimens to highlight internal defects and to assess whether they would affect the structural performance of the specimen.

5.7.1 XCT setup and image reconstruction

During an XCT scan a cone of X-rays is emitted from a source which then pass through the sample and are then incident of a detector. The specimen is placed on a turntable that allows the sample to rotate through 360° during the scan (shown in Fig. 5.7 for the Cantilever 3 specimen). By measuring the energy of x-rays absorbed as the sample rotates a complete picture of the internal material density distribution in the sample can be reconstructed in the form of 2-D image stacks. The closer a sample is to the source the better the image resolution, although at the cost of a more limited field of view. For larger specimens a higher resolution can be achieved by positioning the sample closer to the source and performing multiple scans to capture the entire volume. This was the case for both cantilever specimens which were scanned in two parts.

A 3-D rendering of the scanned part and any internal features of interest is then created from the image slices through image processing. A colour scale is assigned to the magnitude of energy absorbed. With a greyscale regions with the highest absorption would appear white and regions with the lowest absorption would appear black. This is shown in fig. 5.8 where the material boundaries are shown with a light blue line and a single pore within a truss member is shown with a purple curve. By lofting surfaces between the curves in each layer a complete 3-D representation can be generated. The process of thesholding to find boundaries can be subjected as material density changes are not always clearly defined. For the analyses performed here the thresholding value was determined using the Otsu method (Otsu, 1979).

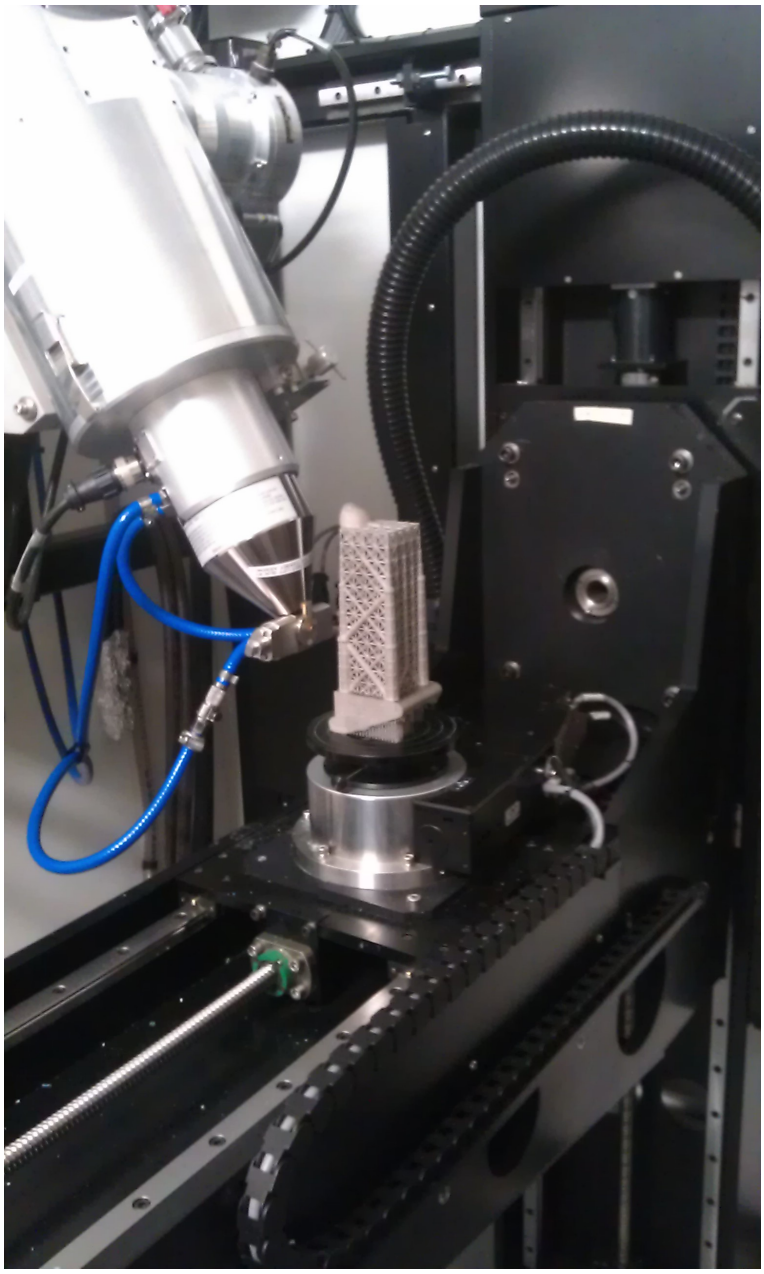


Figure 5.7: Scanning setup in the Nikon Metris Custom Bay, shown here with the cantilever 3 specimen.

5.8 Results

5.8.1 XCT

The two specimens were scanned at the Henry Moseley X-Ray imaging facility at the University of Manchester. Using the Nikon Metric Custom Bay a resolution of 38.9 μm was achieved for both specimens using a 1 mm copper filter and a silver detector.

The distribution of porosity size, which was quantified using an effective diameter, is shown in the histograms in Figs. 5.9b & 5.10b for the cantilever 2 and 3 specimens respectively.

Table 5.3: Equivalent Diameter measurements of detected pores for Cantilevers 1 & 2.

Specimen	Equivalent Diameter (mm)		
	Mean	Max	SD
2	0.150	0.312	0.0492
3	0.704	2.67	0.473

The average effective diameter and standard deviation for both specimens are shown in Table 5.3. The reconstructed isosurfaces for both specimens are shown in Figs. 5.9a & 5.10a where large apparent defects can be observed in the Cantilever 3 specimen.

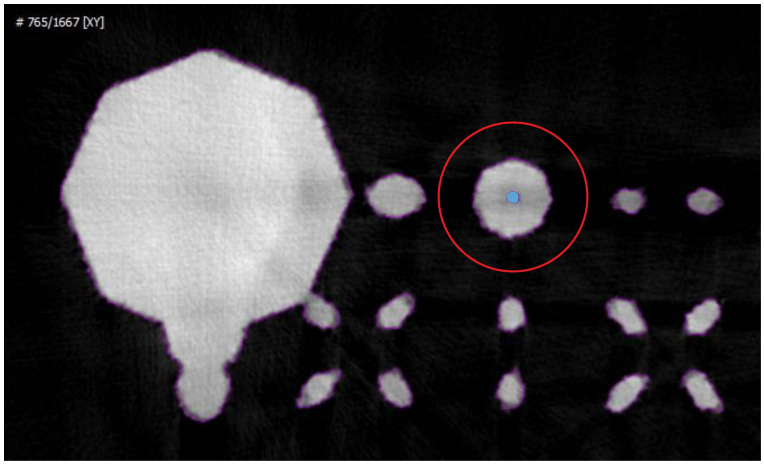
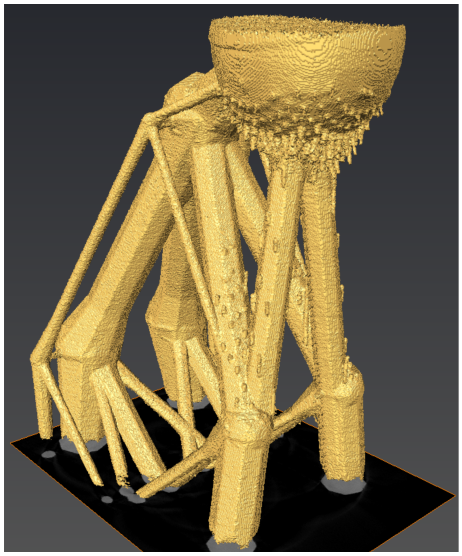
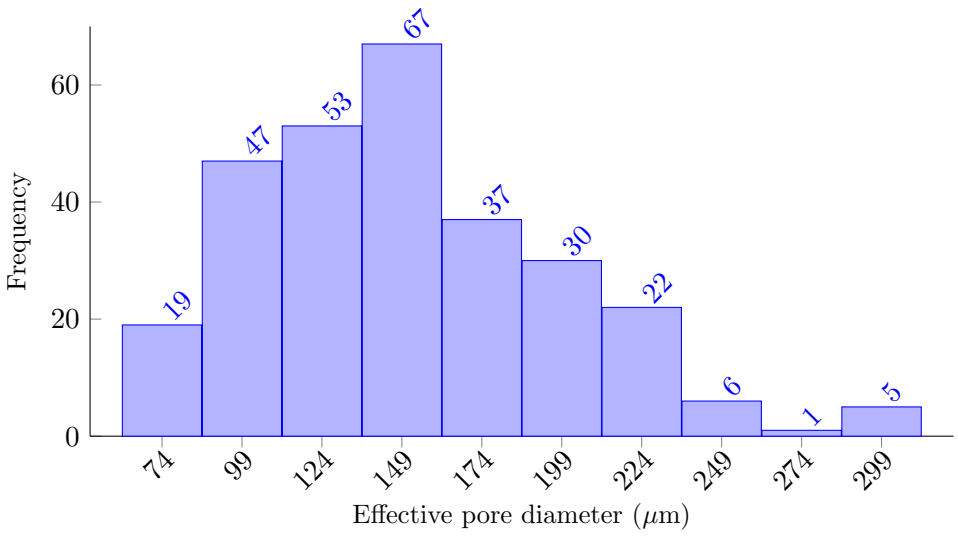


Figure 5.8: Single tomographic image slice on Cantilever 3 specimen. Material boundaries are shown with purple outline. A single large pore (circled) is shown in blue.

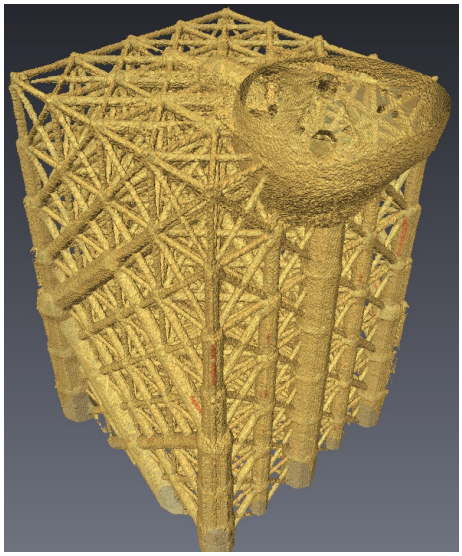


(a)

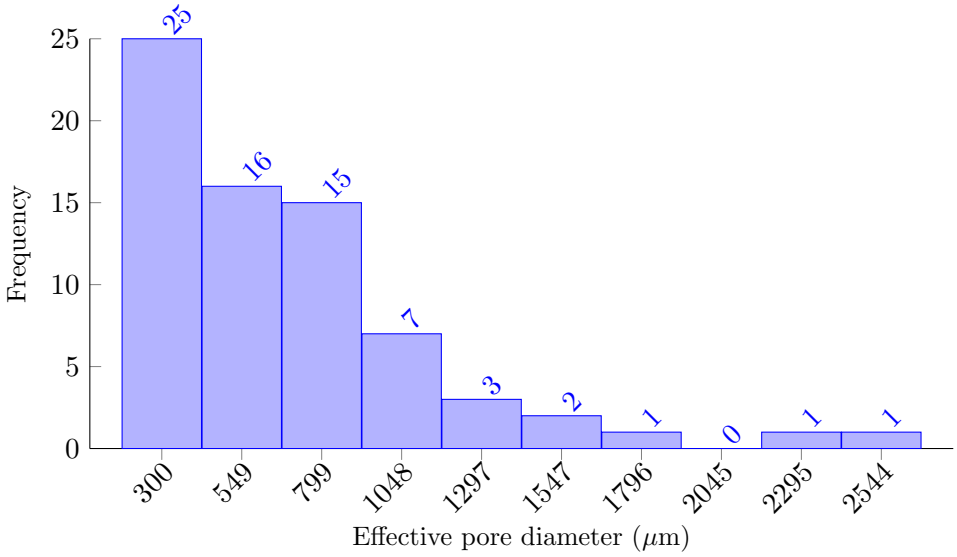


(b)

Figure 5.9: (a) Isosurfaces reconstructed from XCT data for Cantilever 2 and (b) Histogram plot for the porosity size distribution.



(a)



(b)

Figure 5.10: (a) Isosurfaces reconstructed from XCT data for Cantilever 3 with large pores highlighted in red, and (b) Histogram plot for the porosity size distribution.

5.8.2 Load Testing

During load testing all three test specimens were found to fail prematurely. The load displacement curves are shown in Fig. 5.11, with the recorded failure loads shown in Table 6.5. Unlike the other two specimens, Cantilever 1 did not fail in fracture but rather by bending (as seen by the deformation of the two lower members in tension) which the resulted in the buckling of the top two members in compression (Fig. 5.12). Cantilever 2 failed in two stages, starting with a failure of two cross-link members and then followed by fractures at five truss intersections. These two failures are respectively labelled (1) and (2) in Fig. 5.14. Cantilever 3 suffered a very sudden failure at the interface between the specimen and the base structures. A cup and cone fracture surface was observed at the location of two of the four large members at the base (shown in Fig. 5.15) suggesting failure in tension.

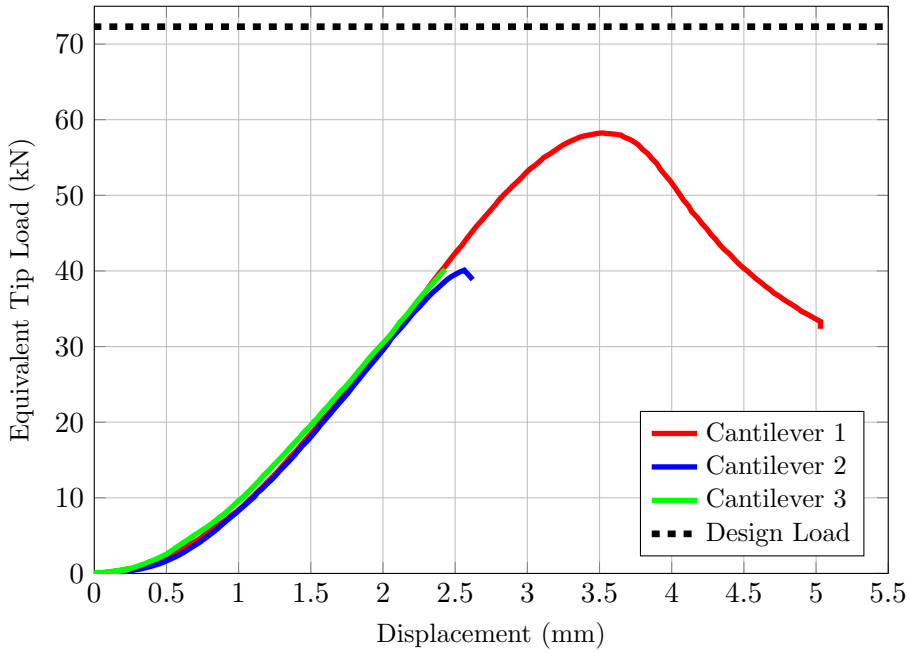
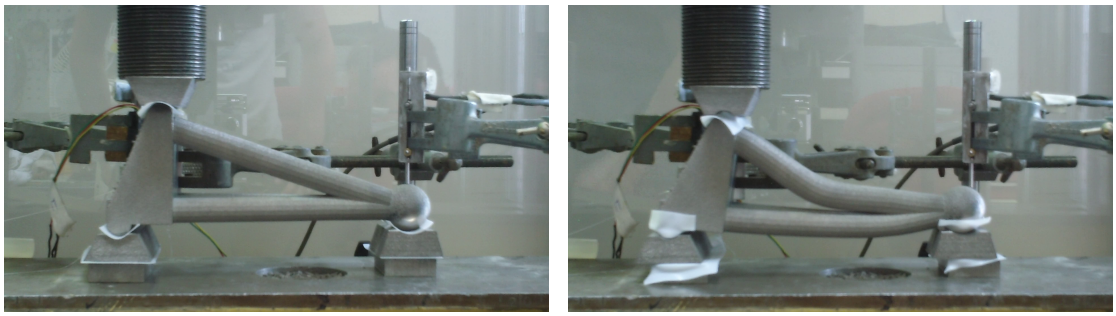


Figure 5.11: Load-Displacement curves for all three cantilever specimens. Dashed line denotes target load.

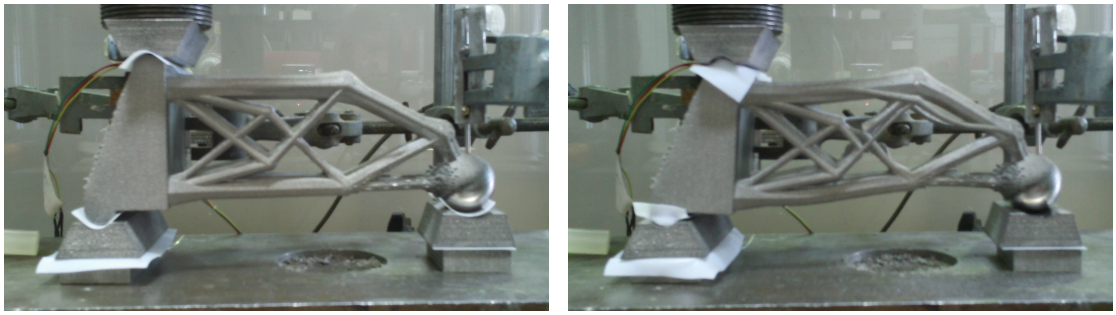
Table 5.4: Volume, mass and load test results for all the fabricated beam specimens. † Total mass is the truss mass plus the ancillary structures required for load testing and after the 0.852 scaling factor had been applied. $V_\infty = 83.97$ cm^3

Specimen	Line Model (resized members) volume (cm^3)	Solid Model (truss only) V/V_∞	Solid Model (truss only) Mass (g)	Solid Model (total mass †) (g)	Fabricated Mass (g)	Ultimate Load (kN)
1	115.8	1.281	476.8	707.2	677.4	58.2
2	93.3	1.019	379.1	631.3	598.2	40.1
3	119.3	1.254	466.5	721.1	681.4	40.2



(a) (b)

Figure 5.12: Cantilever 1 at (a) the beginning and (b) the end of the load test.



(a) (b)

Figure 5.13: Cantilever 2 at (a) the beginning and (b) the end of the load test.

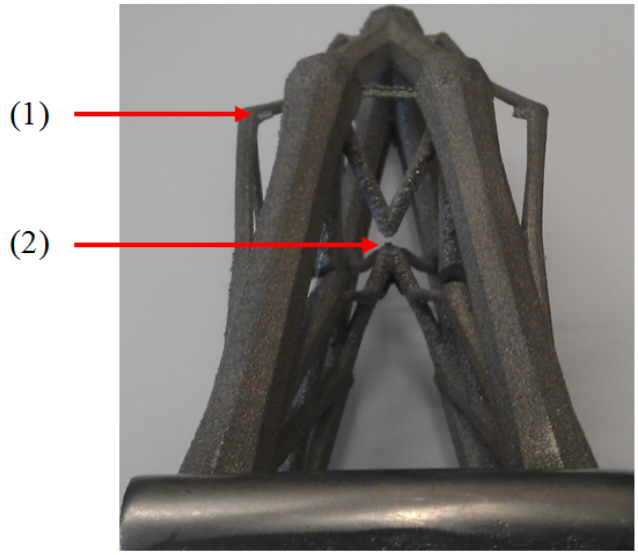


Figure 5.14: Order of fracture failures for Cantilever 2

Figure 5.15: Cantilever 3 post-testing with fracture surfaces shown.

5.9 Discussion

5.9.1 XCT

XCT analysis found internal porosity in both the Cantilever 2 and 3 specimens. The average pore size in the Cantilever 2 specimen was 150 μm, similar to the 100 μm stated in the literature. The largest pores detected were 312 μm in diameter. These pores were mostly spherical.

The pores detected in the Cantilever 3 specimen were significantly larger with an average effective diameter of 704 μm and a maximum of 2.67 mm. Rather than the spherical pores observed in the Cantilever 2 specimen these pores appeared to be similar to the tunnel defects that have been observed in the literature. Thus the use of an effective diameter is somewhat misleading as these defects had a cylindrical profile which, for most of the pores detected, had a diameter of $\sim$300 μm and a length up to 10 mm. Pores of this size will almost certainly influence the static mechanical properties.

An observation of the occurrence of these tunnel defects is that their centre axis was always collinear with the centre line of a truss member. In addition to this they always appeared in pairs about the single plane of symmetry of the Cantilever 3 layout. This would suggest that these pores may have been erroneously identified from artefacts created by the reconstruction process of the XCT data. During this process the contrast of the image is adjusted to account for a phenomenon known as beam hardening, which occurs when low energy photons are rapidly attenuated as the beam passes through the material. But in compensating for this it is possible that the reconstructed image will contain artefacts, which are features that can be mistaken for pores or other internal defects (Kruth et al., 2011).

5.9.2 Load Testing

Recalling that the specimens were fabricated at 85% of their originally intended size, it might be assumed that the load bearing capacity would be 72.3% of the original design value (72.3% = 0.852^2 x 100%), considering area reduction alone. With a revised target load of 72.3 kN the Cantilever 1,2 and 3 specimens only reached 80.5, 55.5 and 55.6% of this load respectively. But additional factors make this simplistic assumption incorrect. Firstly, during the post-processing phase all members were checked to see if they were susceptible to buckling, and that they had diameters above the minimum threshold of 1.5 mm. Any members in violation of one or other of these criteria had their cross-sectional area adjusted. However these measures were not performed after scaling, and as a consequence there were members present in Cantilever 2 that violated both criteria. This will have likely reduced the load bearing capacity but as the failure of the specimen was due to several fractures rather than buckling this may not have had a significant effect.

There were two more serious issues that almost certainly did contribute to the premature failure of Cantilever 2. Firstly, at the locations of the first failure (the cross-links labelled (1) in Fig. 5.14) there were several line elements overlapping collinearly in the solution. When these were converted to cylinders much of volume of each of these members was overlapping with the others. These members should have been merged into one member with the cross-sectional area being the sum of those of the merged members. The sum of the cross-sectional areas of the three overlapping members (which appeared as one in the fabricated piece) at the failed cross-link was 0.622 mm^2. But the actual cross-link area

was just 0.367 mm^2, 58% of what it should have been. i.e. a failure to identify overlapping members in the post-processing phase led to a unintentional stress concentration at each of these locations. Secondly, at five cross-over locations (labelled (2) in Fig. 5.14) there were also stress concentrations. Extra material was added at joints to avoid stress concentrations but the simple post-processing routine employed only performed this measure at nodes, omitting to add material at the five cross-overs indicated in Fig. 5.16.

Unlike the other two specimens, Cantilever 1 failed through bending which induced a buckling failure of the top two members in compression (as can be observed in Fig. 5.12). There was a significant amount of overlap of the members at the tip which would have resulted in a significant bending moment. This can be observed in Fig. 5.12) where the lower two members in tension have bent. The deformation caused by bending then triggered buckling in the top two members in compression. Hence the use of Euler buckling as a method for checking against buckling failures was not completely valid due to this this large bending moment being neglected.

In the case of Cantilever 3, this suffered a very abrupt failure at the interface between the specimen and the base structure. The fracture surfaces shown in Fig. 14 show a cup and cone failure on two of the four largest members at the base. As these members would have been in tension it is possible that the bending moment capacity of the specimen had been exceeded during the test. A simple shear force and bending analysis predicted that the Von Mises stress at failure was below the yield stress of Ti-6Al-4V. Though note that this analysis did not take into account the deformed geometry of the structure at failure. It is possible that the deformed shape of the structure combined with the large applied load from the universal testing machine (341 kN) could have markedly altered the distribution of internal forces. Upon inspection of the fracture surface no tunnel defect voids were observed. It should also be mentioned that the background lattice, although increasing the volume significantly over Cantilever 2 (by 27.9%), did not increase the load bearing capacity.

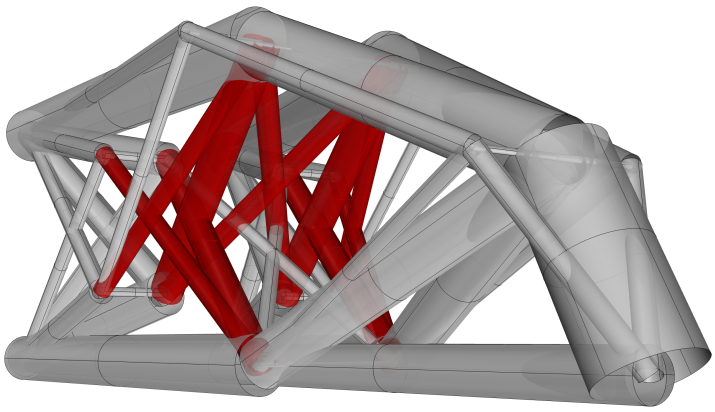


Figure 5.16: Intersecting members that failed during the load testing of Cantilever 2.

5.10 Chapter Conclusions

- All three specimens failed below the target load.
- Cantilever 1 failed prematurely because the buckling capacity of the members was

not accurately determined. The classical Euler buckling criteria used had likely over-simplified the behaviour of these elements.

- The presence of overlapping and/or intersecting members which were not dealt with in the post-processing phase (by introducing additional material) have been identified as the main cause of the premature failure of Cantilever 2. This highlights the need for additional post-processing routines to rectify this.
- Cantilever 3 seemingly failed because its bending load capacity had been exceeded. Changes in geometry coupled with the excessive load required for the three point flexural test arrangement was probably a significant factor. This load test arrangement will not be used is subsequent load tests.
- X-Ray Computed Tomography (XCT) revealed that pores were present in both the specimens which were scanned (Cantilevers 2 and 3). The pores in Cantilever 2 were consistent with the small spherical pores that have been observed in the literature, although some of these were significantly larger.
- Much larger tunnel like defects were detected in Cantilever 3 although it is highly likely that these were artefacts created during the image reconstruction process. No tunnel defects were observed on the fracture surface.

Chapter 6

Secondary optimisation

The contents of this chapter encompasses the content published in the journal article 'Application of layout optimization to the design of additively manufactured metallic component'. Several sections from the article including the introduction and methodology have been omitted as these have been covered in the previous chapters.

6.1 Introduction

The previous chapter outlined the process of transforming a line structure to a 3-D solid model. Because these layouts had overlapping members which were not accounted for when generating the solid model, there were local regions of elevated stress in the fabricated specimens during testing. This was ultimately responsible for the premature failures observed. Chapter 6 details the creation of an optimisation work-flow which includes steps to modify the layout so that the degree to which truss members overlap when transformed to a 3-D continua is minimised.

The main steps in the work-low are summarized below, with the first four steps identical those carried out in Chapter 5:

1. Determine a reference volume, V_0, for the component in question by performing one or more high resolution layout optimisation runs, taking account of the extent of the design domain and the loads and boundary conditions, but no account of 'practical' constraints.

2. Determine a practical layout by performing further layout optimisation runs (these may involve runs with various practical constraints included, or simply the use of lower nodal resolutions if fewer elements are required in the final design).

3. Perform post-processing steps as necessary (e.g. impose minimum area or buckling / overall stability constraints, if not explicitly enforced in step 2), to obtain an updated design and associated volume V.

4. Check structural efficiency $e = V_0/V$. Repeat from step 2 if efficiency e is below an acceptable threshold.

5. Obtain a continuum model by converting the line-structure using geometrical rules. (Recheck structural efficiency if desired.)

A more detailed view of the workflow is shown in Fig. 6.1. The workflow is then applied to the design of a simple and a somewhat more complex component (Section 6.3), which are then fabricated using the electron beam melting (EBM) additive manufacturing technology and, in the case of the simpler component, load-tested to failure (Section 6.4).

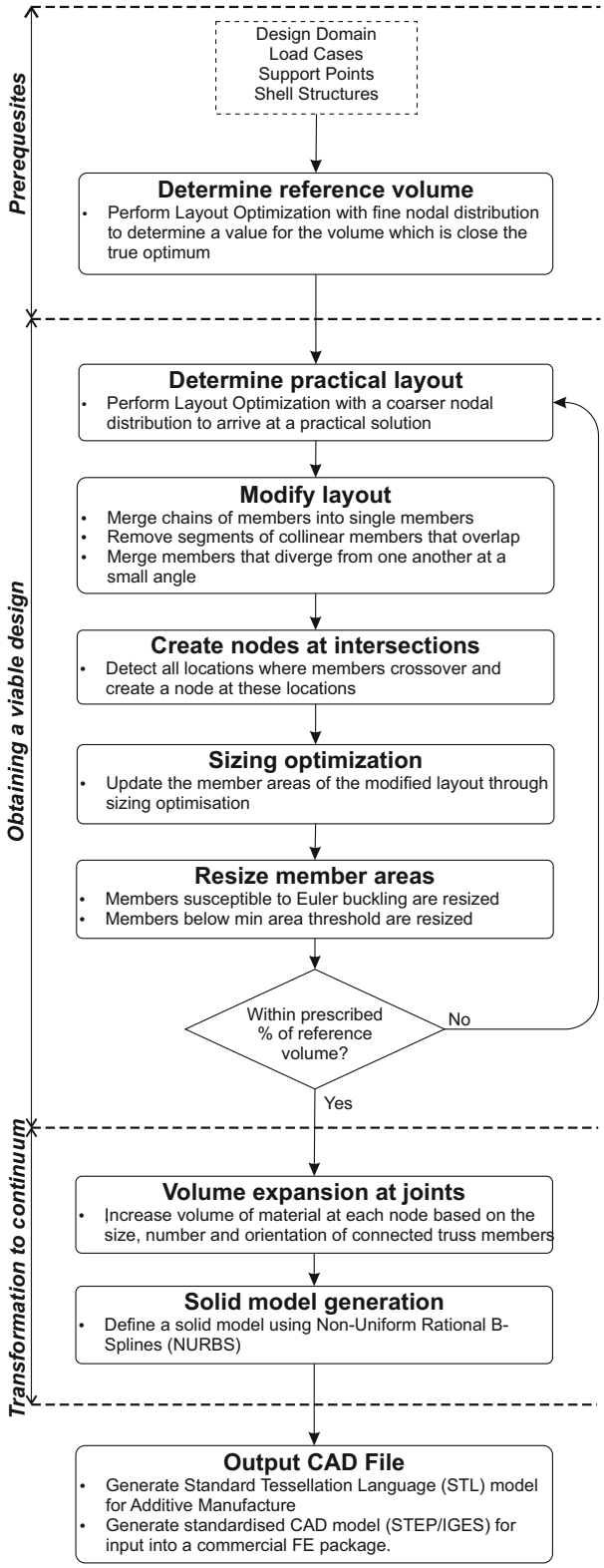


Figure 6.1: Flow chart showing the proposed design workflow, from layout optimisation to finalized CAD model

6.2 Transformation of line structure into a continuum

The output from the layout optimisation procedure will be a frame comprising a series of one-dimensional line elements. In order for this to be realized physically, it must be transformed into a 3-D solid model. Various means of achieving this are possible; here the emphasis has been on simplicity and ease of implementation. Should the basic approach prove to be attractive then clearly the level of sophistication can in due course be increased.

Thus, for sake of simplicity, it was decided that each line element would be transformed into a solid circular element (i.e a simple cylinder). Steps must be taken to address situations where two or more members, once expanded, occupy some of the same volume. This can occur at joints and in regions where members overlap. Also, at this stage, members which are too thin to be effectively fabricated using the chosen additive manufacturing process, or which are susceptible to buckling, need to be resized. These stages, together with the final form generation, constitute steps in the post-processing workflow, which will now be outlined in sequence.

6.2.1 Modify layout

Collinearity

Chains of collinear elements, as shown between nodes A and B in Fig. 6.2a, should be identified and then merged. This is primarily because the buckling analysis described in Section 6.2.4 requires these to be merged into a single element. This can be achieved by searching for nodes that have only two connections (with the same normalized direction vector), and then eliminating them (i.e. nodes C, D and E in Fig. 6.2a).

A collinear member that overlaps with a section of, or the entirety of, another member must also be identified and modified (e.g. see Fig. 6.2b, where member AB overlaps with member CD). (Otherwise, when the line elements are converted to cylinders, there will be significant overlap which, if left unchecked, will result in high local stresses.) Considering the example shown in Fig. 6.2b, Eq. (6.1a) can be used to identify members that are collinear and Eq. (6.1b) and Eq. (6.1c) can be used to identify if these collinear members pass through an end node of another member, and hence overlap.

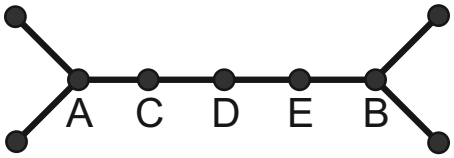
$$\left| \frac{(i \cdot j)\,(i \cdot v_{AC})}{|i|^2\,|j|\,|v_{AC}|} \right| = 1 \tag{6.1a}$$

$$0 < \frac{v_{AC}, v_{AD}}{i} < 1 \tag{6.1b}$$

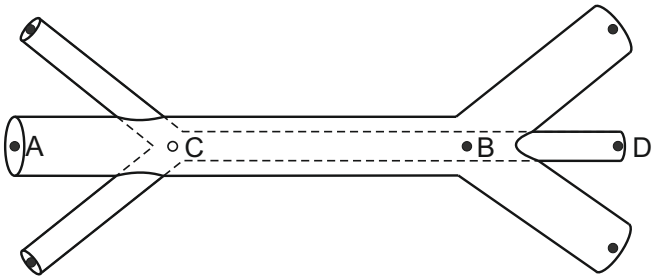
$$0 < \frac{v_{CA}, v_{CB}}{j} < 1 \tag{6.1c}$$

where i and j are the vectors of members AB and CD respectively. Also, v_{AC}, v_{AD}, v_{CA} and v_{CB} are vectors interconnecting the nodes denoted by the given subscript.

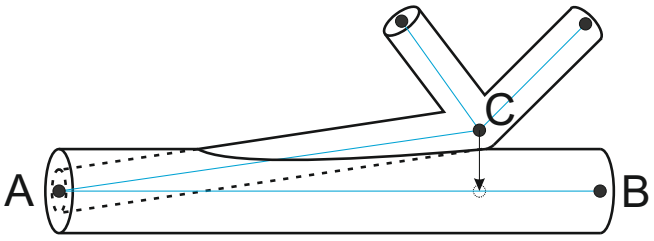
The members are then split at the nodes they pass through, and overlapping segments deleted.



(a)

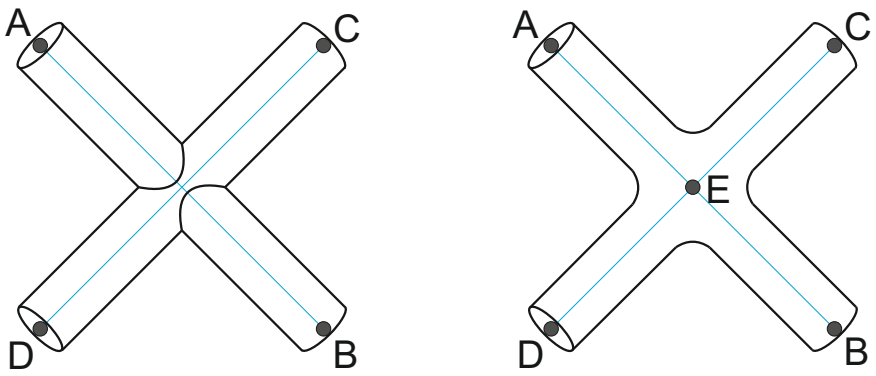


(b)



(c)

Figure 6.2: (a) Collinear overlapping members, (b) chain of members between nodes A and B, (c) members that overlap significantly



(a) (b)

Figure 6.3: (a) illustration of two members (AB and CD) that crossover and (b) node created at this intersection permitting use of the joint expansion routine detailed in sec. 6.2.5

Overlapping at nodes

A pair of members which intersect a common node at a small subtended angle will often overlap over a large part of their length (Fig. 6.2c). A way of addressing this is to simply merge the two members. Equation (6.2) determines if the perpendicular distance from one member to the end node of a second member is less than the sum of the radii of the two members. If so, and if the two members are the same length, then the two end nodes are merged and the resulting node placed midway between them. If they have different lengths then the node joined by the shorter element is moved onto the longer element in a direction perpendicular to it. Thus in the case shown in Fig. 6.2c, node C would be moved onto element AB, which would then be split into two elements, AC and CB. This algorithm is performed for all the members at each joint, with the longest member taken as j.

$$\left| j - i \left(\frac{j \cdot i}{|i|^2} \right) \right| \leq r_i + r_j \tag{6.2}$$

Considering the example shown in Fig. 6.2c, i and j would be the vectors of elements AC and AB respectively, which have radii r_i and r_j.

6.2.2 Create nodes at intersections

Layout optimisation will frequently generate topologies that include members that crossover one another, presenting two issues: (i) because there is no explicit node at a crossover location (which now forms a joint) the simple buckling analysis that will be presented in section 6.2.4 will not provide accurate results; (ii) the joint expansion routine (detailed in section 6.2.5) will not be applied at all. The simplest way of addressing both these issues is to find all crossover locations and to then add explicit nodes at these locations, thereby splitting adjoining elements at the newly created node.

Determining which members intersect with one another and the coordinates of the intersection can readily be found. Thus, considering the example depicted in Fig. 6.3, (6.3a)

and (6.3b) can be evaluated:

$$n_{Ei} = n_A + i \left[\frac{(j \times v_{CA}) \cdot (j \times i)}{(j \times i) \cdot (j \times i)} \right] \tag{6.3a}$$

$$n_{Ej} = n_C + j \left[\frac{(i \times v_{CA}) \cdot (j \times i)}{(j \times i) \cdot (j \times i)} \right] \tag{6.3b}$$

where n_A and n_C are the (x, y, z) coordinates of nodes A and C and where i and j are the vectors of elements AB and CD. Also, v_{CA} is the vector interconnecting nodes C and A and n_{Ei} and n_{Ej} are the coordinates of node E. If these two coordinates are the same then the members do crossover, and a new node should be created at this location. Calculations of this type should be undertaken for each pair of members in the frame.

6.2.3 Sizing optimisation

After modifying the layout of members, a secondary sizing optimisation can be performed to update the member cross-sectional areas. The same layout optimisation formulation and design parameters are used, but now using the updated layout as the ground structure. Because the problem size is very small this optimisation step adds an insignificant amount of time to the overall post-processing workflow.

6.2.4 Resize member areas

The buckling response of solid circular bars is intrinsically non-linear and thus it is convenient to consider buckling as a post-processing step, rather than during the main optimisation phase. The relevant Euler buckling relationship, which must be satisfied for all compressive members, is as follows:

$$a_i \geq \sqrt{\frac{4q_i^-(k_{eff}l_i)^2}{\pi E}} \qquad i = 1, ..., m \tag{6.4}$$

where a_i, q_i^- and l_i are respectively the area, compressive force and length of member i and k_{eff} is the effective length factor. Finally, E is the elastic modulus of the material.

Any members that violate this criterion are resized. The choice of a suitable effective length factor (k_{eff}) is here investigated experimentally - see Section 6.3.2.

A simple minimum area constraint (6.5) must also be enforced during the post-processing workflow. This is because many additive manufacturing processes will not effectively capture truss bars below a given diameter.

$$a_i \geq a_{min} \qquad i = 1, ..., m \tag{6.5}$$

where a_{min} is the minimum permitted cross-sectional area.

6.2.5 Volume expansion at joints

Given that numerous elements will often converge at a node in a layout optimisation solution, there will often be a significant amount of overlapping in the vicinity of a joint.

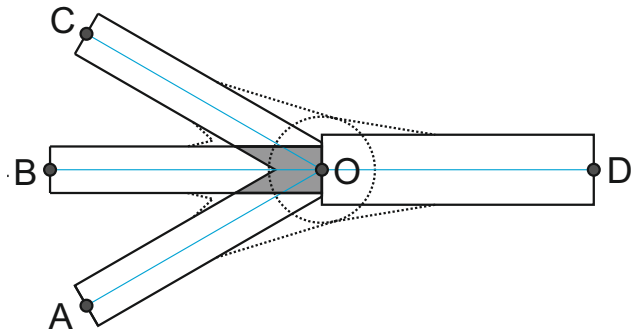


Figure 6.4: Schematic of joint expansion

To avoid high localized stresses at such locations additional material must be introduced. Equation (6.6a) is evaluated at each node to calculate the radius of the joint in proportion to the area and the number and orientation of the elements connected to the node.

$$r = \sqrt{\frac{1}{\pi}\left(a_i + \sum_{k=1}^{m} a_{j,k}\left\{\hat{i}\cdot\hat{j}_k\right\}\right)} \tag{6.6a}$$

Where $\hat{i}$ and a_i is the unit vector and area of the master member respectively, and $\hat{j_k} = \{j_1, j_2, ...j_m\}$ and $a_{j,k} = \{a_{j,1}, a_{j,2}, ...a_{j,m}\}$ are the unit vectors and areas of all the other members connected to the node. This equation is evaluated with every member as the 'master' member, with the largest value for the radius being selected.

Note that because additional material is introduced at joints, the volumes of proposed layouts which include large numbers of joints may be significantly increased in this step, potentially rendering the design unviable. In other words, the proposed procedure relies on the initial layout optimisation solution being practical, and amenable to post-processing using the procedure described. If this is not the case the post-processed volume may exceed the reference volume by a wide margin.

6.2.6 Solid model generation

The final form of the component is conveniently generated using Non-Uniform Rational B-Splines (NURBS), which provide a mathematical representation of analytical and free-form geometries and are the industry standard for CAD exchange formats such as STEP and IGES. Each member is created by extruding a circular cross-section along the length of the line element, with the area being set to that determined from the layout optimisation process. At each joint a sphere is created with a radius calculated using Eq. (6.6a). At a set distance from the joint node, each truss member is expanded using a linear loft operation so that the end radius is equal to that of the sphere at the joint. This is shown schematically by the dotted line in Fig. 6.4. Boolean operations are then performed to create a single 'watertight' surface for the whole geometry.

6.2.7 Output CAD file

A STereoLithography (STL) file can then be derived in preparation for additive manufacture and a STEP, IGES or other standard CAD file can be created for input into a commercial finite element analysis package. Here the NURBS based geometries were created

using Rhinoceros V5.0 (developed by Robert McNeel & Associates) and the Grasshopper plug-in (also developed by Robert McNeel & Associates).

6.3 Component design examples

The layout optimisation workflow outlined in Section 6.2 was applied to two case study problems: a single point loaded beam and an air-brake hinge for the Bloodhound Supersonic Car (SSC) project, with the former being fabricated and load tested.

6.3.1 Material properties

As in the previous chapter, all specimens were to be fabricated from Ti-6Al-4V 'extra low interstitials' (ELI) supplied by Arcam and tested in their as-built state (i.e. with no surface finishing). As all specimens in this study were to be built vertically a value of 842 MPa was selected as the limiting stress in the optimizations. An elastic modulus of 113.8 GPa (standard for Ti-6Al-4V) was used in the subsequent buckling analyses.

6.3.2 Example 1: beam subject to point load

The Messerschmitt-Bölkow-Blohm (MBB) beam is a 2-D benchmark optimisation problem often referred to in the literature, consisting of two simple supports and an elevated load applied at a location midway between the supports. To better assess the capabilities of the layout optimisation method, this standard problem has been adapted to 3-D, as illustrated in Fig. 6.5.

Problem definition

The problem definition shown in Fig. 6.5 consists of a cuboidal domain with four support locations at the corners of the base. A half-cylindrical rigid shell structure (radius = 7.5mm and length = 20mm) discretized using 266 triangular elements is located centrally in the design domain but with the flat surface coplanar with the top surface of the design domain. Each of these triangular elements could accept multiple truss connections. The 100kN load was applied as a pressure on the flat surface of the rigid shell structure.

This arrangement meant that, even when a single load case was involved, the probability of encountering a solution that was in unstable equilibrium with the applied loading was low. This avoided the need to consider multiple load cases (or the use of a stability formulation, such as the one described by Tyas et al. (2005)) in this preliminary study.

Nodes used for the layout optimisation were generated spatially in the design domain using cubic grids of various densities. These nodes, together with those positioned on each triangle forming the rigid shell structure, were used to create the ground structure for each optimisation.

Design candidates

Volumes from optimizations performed with cubic nodal grids of differing densities were established, a selection of which are shown in Table 6.1. From these a reference volume

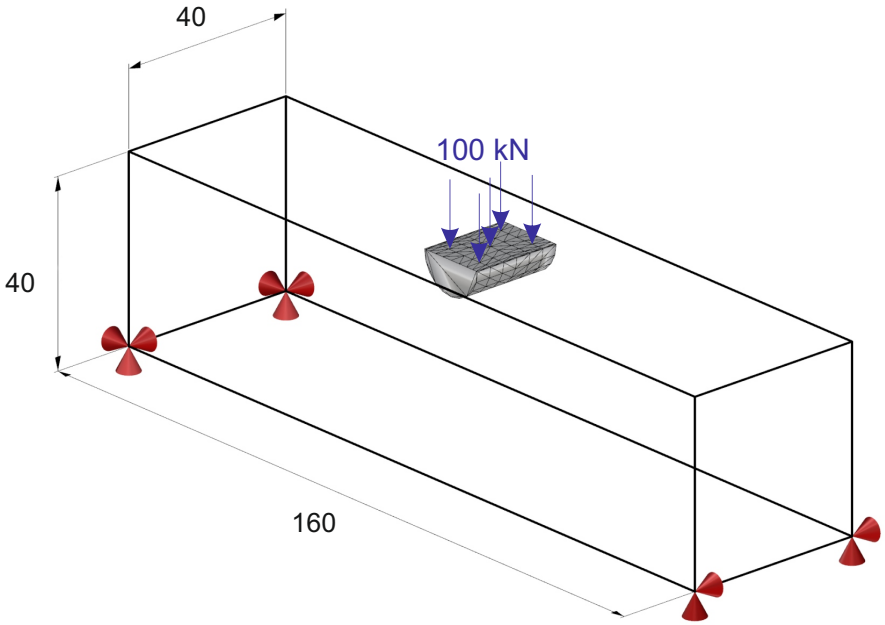


Figure 6.5: Example 1: problem definition (all dimensions are in mm). Constrained support directions are indicated

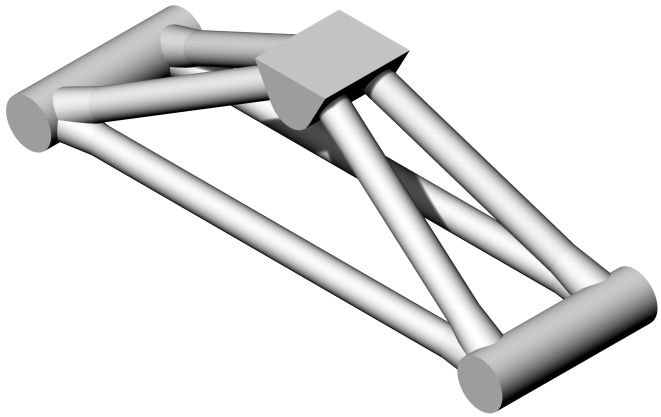
V_0 of 33.06cm^3 was determined, using the power law extrapolation scheme described by Darwich et al. (2010b). (Note that this implicitly assumes that a truss form will be optimal, or near-optimal.)

After resizing members for buckling and minimum area considerations, the solutions obtained when using an initially coarse nodal discretization (8 nodal divisions across the span length) were found to lie within approximately 10% of the reference volume. In contrast solutions obtained when using more nodes (e.g. 20 nodal divisions) were found to be heavier after member resizing, and considerably more complex (e.g. see layout shown in Fig. 6.6c). It was therefore considered that the simpler structure shown in Fig. 6.6b made a more suitable design candidate for this preliminary study.

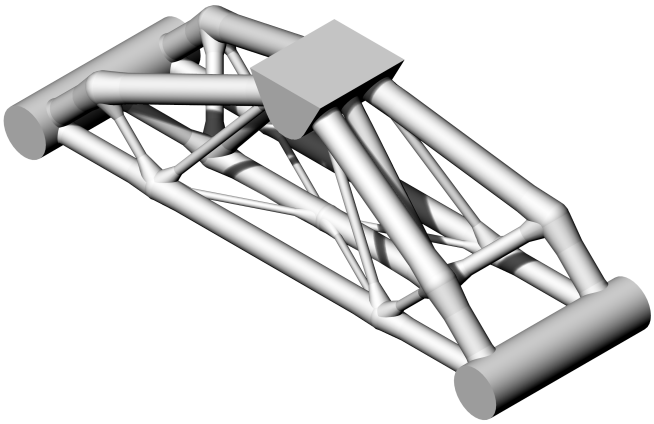
Final designs

A total of nine examples of the 20mm nodal spacing design (shown in Fig. 6.6b) were fabricated; three identical specimens (for the purposes of repeatability) for each assumed member end condition shown in Table 6.1. The main difference between each group is the cross-sectional area of the innermost inclined compressive members, which were deemed susceptible to buckling for all three assumed joint conditions.

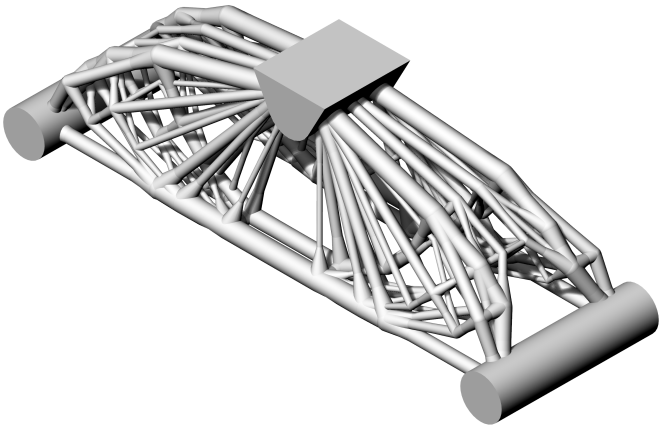
Additionally, a simple six element truss design Fig. 6.6a was also designed to act as a benchmark (first entry in Table 6.1). Again three specimens of this design were fabricated. Note that for the problem considered the benchmark is actually quite competitive, being less than 10 percent heavier than the optimised design. However, for more complex design problems this difference can be expected to be greater.



(a)



(b)



(c)

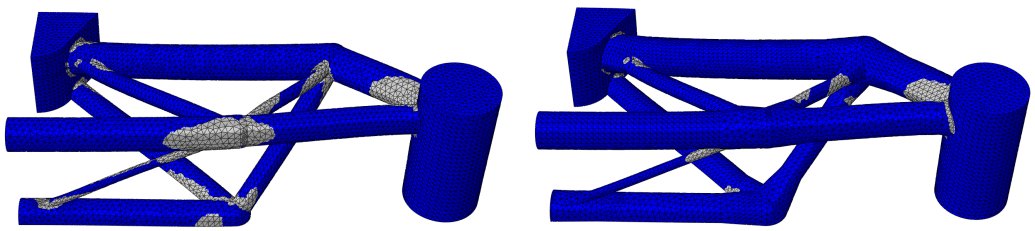
Figure 6.6: Example 1: candidate designs, (a) benchmark , (b) obtained using a nodal spacing of 20mm (89096 potential members) and used in testing, and (c) obtained using a nodal spacing of 8mm (499213 potential members) which was rejected for being too complex

Table 6.1: Example 1: Volumes of line models resulting from layout optimisation with differing nodal densities. (Volumes after the members have been resized to account for buckling are shown for different assumed end conditions; A - Fixed-Fixed, B - Fixed-Pinned and C - Pinned-Pinned.)

.

Nodal spacing (mm)	No. of nodes†	No. of potential members	Vol.		Vol. after member resizing					
					A		B		C	
			(cm^3)	Δ%	(cm^3)	Δ%	(cm^3)	Δ%	(cm^3)	Δ%
-	8	68,724	38.1	15.2	38.1	15.2	38.1	15.2	38.76	17.2
20	81	89,096	35.5	7.4	35.6	7.7	36.07	9.1	36.98	11.9
8	756	499,213	34.17	3.4	37.9	14.6	39.55	19.6	43.48	31.5
4	4961	11,541,432	33.61	1.7						
2	35721	530,708,402	33.32	0.8			*not resized*			
∞‡	-	-	33.06	0.0						

† Excluding nodes in the rigid shell structure (266 in total).
‡ Obtained via extrapolation.



(a) (b)

Figure 6.7: Example 1: (a) elasto-plastic finite element analysis of the fixed-pinned optimised specimen without joint expansion, and (b) with joint expansion. Highlighted are the regions of plastic strain at 80% of the design load (80kN).

6.3.3 Finite element analysis verification

A simplified elasto-plastic finite element analysis was performed for the 'fixed-pinned' optimised geometry with and without the joints expanded (see Section 6.2.5). The specimens with and without expansion at the joints were meshed using 89138 and 104494 10-noded tetrahedral elements respectively, as shown in Fig. 6.7a and Fig. 6.7b; in both cases quarter symmetry was assumed. At the time of the study strain hardening data was not available for as-built titanium manufactured using the EBM process. Thus an elastic-perfectly plastic model was assumed, with an elastic modulus of 113.8 GPa, yield stress of 842 MPa and Poisson's ratio of 0.342. It is evident from Fig. 6.7a and Fig. 6.7b that stress concentrations at the joints were reduced by using the joint expansion algorithm, though not entirely eliminated.

6.3.4 Example 2: Bloodhound SSC air-brake hinge

The Bloodhound project aims to build a supersonic car that will break the land speed record, and also reach the landmark top speed of 1000 mph (1600 km/h). The predominant braking system is in the form of two deployable air-brakes positioned on each side of the car. Each air-brake is attached to the car with four hinges, as shown in Fig. 6.8a. The load exerted on each hinge is related to the aerodynamic drag experienced by the air-brake during deployment. Naturally the Bloodhound team wish to keep the structural weight of the car to a minimum and are thus investigating the use of optimisation to reduce the mass of this and other components. This provides an opportunity to apply layout optimisation to a practical engineering problem.

Problem definition

The original design and load cases (Table 6.2) were provided by the Bloodhound team. The design domain shown in Fig. 6.8b is based on the topology of the original design. The size of the slot into which the hinge retracts is based on the original design; it is imperative that no material is placed outside the volume of the original design. The volume of the original design in the region to be optimised is 189cm^3. The hinge is subjected to five load cases, derived from various stages in deploying the air-brake (Table 6.2).

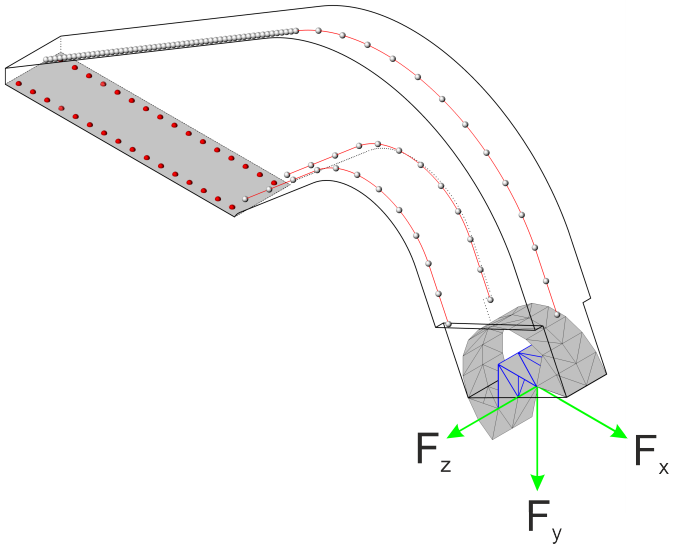
The Bloodhound team require the hinge to resist the loads shown in Table 6.2 without yielding or failure. A factor of safety (FoS) of 2.4 on ultimate strength was specified.

The initial optimizations were performed using a regular cubic nodal grid, as used for Example 1. Potential members created between these nodes that crossed any of the design domain boundaries (Fig. 6.8b) were omitted from the ground structure. Nodes shown on the shaded plane in Fig. 6.8b were subject to a displacement constraint in all three Cartesian directions. A rigid shell structure composed of 78 triangular elements was placed at the tip to represent the connector ring (outer diameter = 50mm; inner diameter = 24.5mm). Loads were applied directly to the shell structure, on the shell elements shown in Fig. 6.8b.

The use of a cubic nodal distribution in the curved design domain proved to be problematic. Because the nodes were not placed in a way that reflected the shape of the design domain the optimiser would yield overly complicated layouts with many very thin members. This problem is especially exacerbated when using relatively coarse nodal distributions. To place the nodes in a more ideal distribution with the goal of yielding simpler layouts, a parametric model was created to allow nodal locations to be modified in a straightforward and interactive manner. The parametric model creates splines that follow the profile of the design domain (illustrated in Fig. 6.8b). The number and relative spacing of these splines could be defined across the breath and height of the design domain. Nodes could then be created along these splines at a specified spacing. Although this approach is devised based on engineering judgement, a safeguard is that the resulting volume can always be compared with the reference volume.

After a number of iterations, involving varying the nodal distribution using the parametric model, a final design was arrived upon, as shown in Fig. 6.9(a). The resulting volume was somewhat higher than the reference volume obtained via extrapolation (using increasingly fine nodal discretizations and the extrapolation scheme described by Darwich et al. (2010b)); respectively 13.8% and 17.9% heavier before and after member resizing, as indicated on Table 6.3. However, this was nevertheless deemed reasonable.

(a)

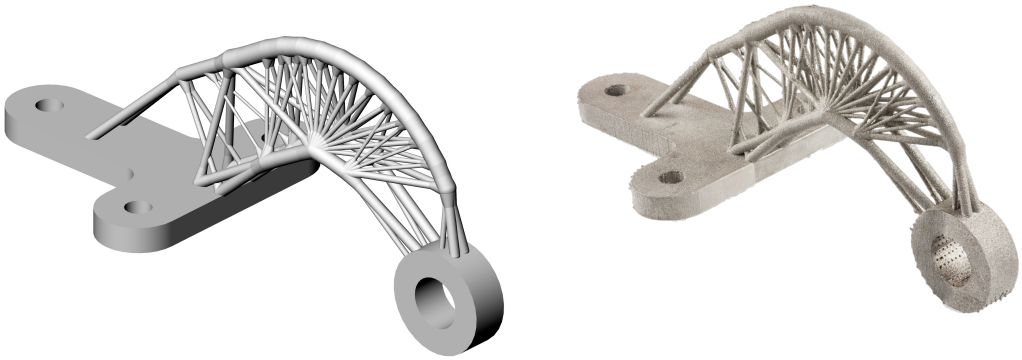


(b)

Figure 6.8: Example 2: (a) original air-brake hinge design and (b) problem definition for the present study, derived from (a)

Table 6.2: Example 2: Load cases considered for Bloodhound SSC air-brake hinge (prior to FoS of 2.4 being applied)

Load Case	F_x (N)	F_y (N)	F_z (N)
1	5500	7550	0
2	4171	-2981	500
3	3675	765	500
4	2440	6399	600
5	-4545	12272	500



(a) (b)

Figure 6.9: Example 2: (a) final design for airbrake hinge (volume = 163.9 cm^3), (b) final design additively manufactured from Ti-6Al-4V using the EBM process.

Table 6.3: Example 2: optimisation performed using cubic nodal grids of varying densities. The entry marked with a (*) is the final design produced using a nodal distribution created with the parametric model. A fixed-pinned assumption was used for the buckling analysis during post-processing.

Nodal spacing	No. of nodes	No. of potential members	Vol. (cm^3)	Δ%	Vol. after member resizing (cm^3)	Δ%
Parametric*	157	13,994	34.00	13.8	35.21	17.9
12.5	340	36,824	33.64	12.6	34.81	16.5
5	1022	2,988,133	31.37	5.0		
3.5	8420	21,720,412	30.80	3.1		*not resized*
∞†	-	-	29.87	-		

† Obtained via extrapolation.

Table 6.4: Example 1: design and measured masses of the benchmark and 'fixed-fixed' optimised designs

Sample	STL mass (g)	Measured mass (g)	Δ%
Benchmark	284.5	269.2	5.4
Optimised	273.6	259.8	5.0

6.4 Additive manufacture and load testing of components

6.4.1 Manufacture

All specimens were manufactured from gas atomized Ti-6Al-4V ELI powder using an Arcam EBM S12 system. Layers were deposited at a thickness 70 μm and the standard Arcam Ti-6Al-4V melt, wafer support and preheat themes for were used. In addition to the standard process themes a modified pin support theme based on the standard 'Nett' theme was used to produce porous pin supports.

Initial Build Study

In order to determine how accurately the EBM process would fabricate the two beam designs a preliminary build study was carried out. The Example 1 benchmark and fixed-fixed optimised designs were fabricated in the same build with the mass and truss diameters measured upon completion. The mass of each specimen was found to be approximately 5% lighter than the predicted mass of the STL definitions they were built from (shown in Table 6.4). The measured truss diameters were also below those defined in the respective STL files. This discrepancy has been documented previously when using the EBM process to fabricate truss structures (Cansizoglu et al. 2008). As the length of the truss bars are dimensionally correct, it would seem that the 5% volume reduction is the result of undersized truss areas. Therefore a pragmatic solution to this was to increase the member cross-sectional areas by 5%.

6.4.2 Example 1: beam subject to point load

Fabrication

After the preliminary build study all 12 specimens (incorporating a 1.05 area scale factor) were built in two batches. The specimen arrangement and the support structures utilized for the builds is shown in Fig. 6.10. Each build took 40 hours to complete on an Arcam S12 machine.

Standard scaling factors were also applied to the whole sample geometry, as provided by the manufacturer Arcam (1.0068 in x- and y-directions and 1.0093 in the z-direction). These account for shrinkage during cooling.

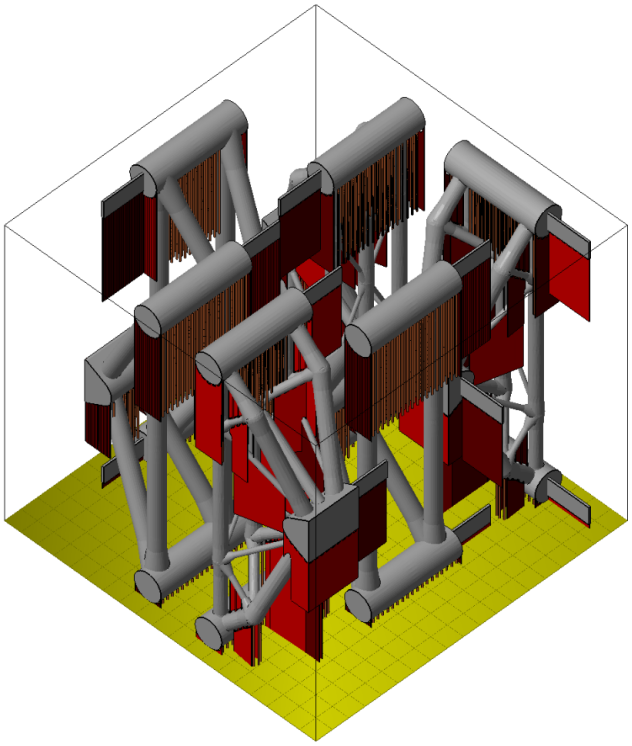


Figure 6.10: Example 1: arrangement of specimens during manufacture. The wafer and pin supports structures required by the EBM process to alleviate heat accumulation are shown in red. The build envelope is 190×190×180mm.

Load testing arrangement

All the fabricated specimens were load tested using a universal testing machine in the arrangement shown in Fig. 6.11. Each specimen was placed on two sets of supporting blocks, each set having one grooved and one flat support. All contact surfaces were polished and lined with PTFE sheets. The grooved support provided more contact surface with the cylindrical ends of the specimens, helping distribute the reaction load whist allowing rotation. The flat support permits the grooved support to translate horizontally with minimal resistance, effectively replicating a roller support. The load was applied centrally on each specimen using a polished steel ball coupled to the cross head of the testing machine. The cross head applied the load at a constant displacement rate of 0.167 mm/min.

Load testing results

The measured masses and results from the load tests for all twelve specimens are shown in Table 6.5. The specimens designated by 'O' are the benchmark designs and the load-displacement curves for these specimens are shown in Fig. 6.13 (it is evident that all the specimens narrowly failed to carry the design load of 100kN). The optimised specimens, designated A, B and C are those analysed assuming fixed-fixed, fixed-pinned and pinned-pinned end conditions respectively in the buckling analysis. The load displacement curves for these specimens are shown in Fig. 6.14.

It is evident that most of the specimens designated B and C were able to carry the design load of 100kN, whereas, as expected, specimens designated A did not.

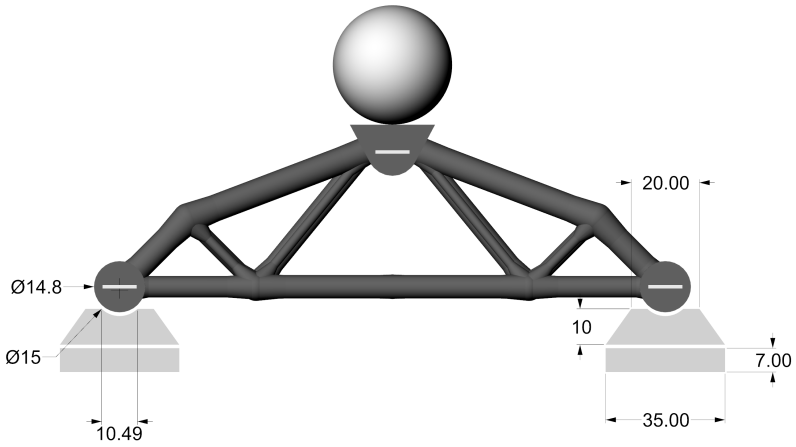


Figure 6.11: Example 1: load testing arrangement. (All dimensions are in mm)

Table 6.5: Example 1: volume, mass and load test results for all the fabricated beam specimens.

Specimen	Line Model (resized members) volume (cm^3)	Solid Model (truss only) Volume (cm^3)	Solid Model (truss only) Mass (g)	Solid Model (total mass †) (g)	Fabricated Mass (g)	Ultimate Load (kN)
O1					284.6	85.9
O2	38.10	41.15	182.3	284.5	283.2	91.6
O3					281.8	95.3
A1					273.4	81.6
A2	35.60	38.06	168.6	273.6	268.8	80.3
A3					269.4	79.4
B1					260.2	93.2
B2	36.07	38.24	169.4	266.8	258.8	102.4
B3					258.6	100.9
C1					260.6	101.7
C2	36.98	39.20	173.7	270.0	262.8	98.6
C3					262.4	107.2

† Total mass is the truss mass plus that of the ancillary structural elements required for load testing.

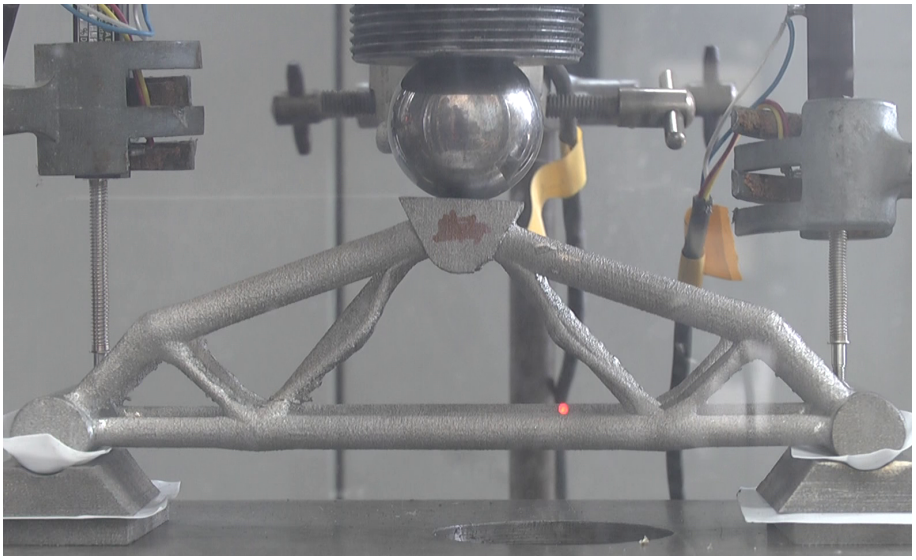


Figure 6.12: Example 1: specimen A1 at ultimate load. The specimen failed through the buckling of the four central compressive members

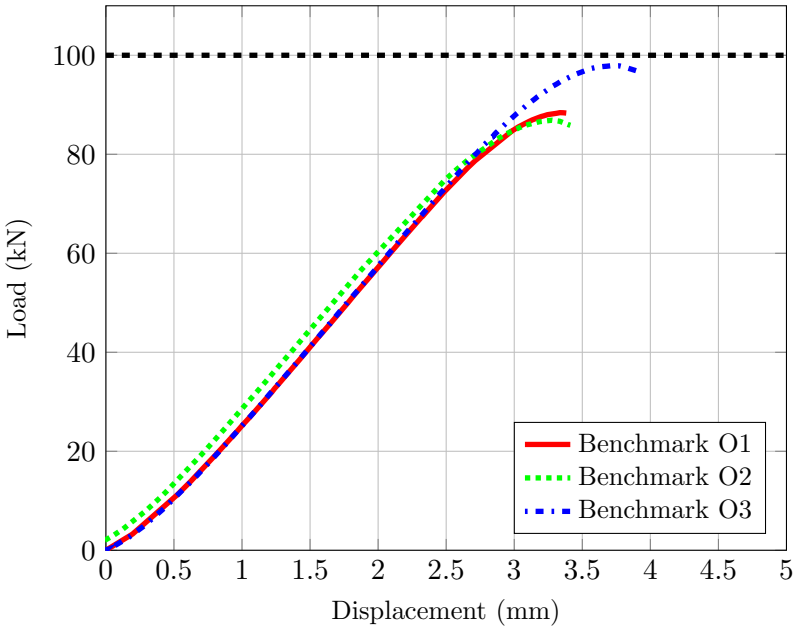


Figure 6.13: Example 1: Load-Displacement curves for benchmark samples (O1 - O3)

6.4.3 Example 2: Bloodhound SSC air-brake hinge

Fabrication of Bloodhound SSC air-brake hinge

To demonstrate the design could be realised physically, the Bloodhound SSC air-brake hinge was also fabricated, using the same approach as used for the beam specimens. The fabricated design is shown in Fig. 6.9(b).

The volume and mass of the solid model and fabricated component are shown in Table 6.6. The volume of the original design replaced by the truss structure was 189cm^3. However, as the original part was designed to be fabricated using aluminium 7075 T7 alloy rather than titanium Ti-6Al-4V, this volume should be scaled to obtain the volume which would be occupied by a structurally equivalent titanium component (i.e. scaling by the ratio of the ultimate strengths of aluminium 7075 T7 alloy and titanium Ti-6Al-4V, 505MPa and 842MPa respectively). This gives a scaled volume of 113cm^3. With a volume of 35.27cm^3, the truss structure therefore consumes 69% less material compared with the original component.

6.5 Discussion

6.5.1 Optimisation methodology

This study has demonstrated that layout optimisation can be used as a part of a workflow to automatically produce 3-D CAD models, ready for additive manufacture. Physical load testing of beam specimens designed using this workflow, and then fabricated using the EBM process, generally met or slightly exceeded the required load capacity. The beam problem was selected for its simplicity for this exploratory study and as such the potential for mass reduction was limited (approx. 7% compared with the benchmark design in this case). The difference in the measured strength-to-weight ratio of the benchmark and optimised specimens was a more significant 18% increase in the case of the latter. For the more complex Bloodhound SSC airbrake hinge problem the volume reduction was however much more significant (approx. 69%).

It is imperative to properly account for buckling in the proposed workflow. When checking the optimised designs, the specimens designed assuming 'fixed-pinned' and 'pinned-pinned' end support conditions in the buckling calculations generally met or slightly exceeded the design load. Those designed using the 'fixed-fixed' assumption (samples A1-3) failed through buckling at around 80% of the design load, with the buckling failure of specimen A2 shown in Fig. 6.12. This might suggest that at least the 'fixed-pinned' end support condition should be used in future when considering buckling. However, all the three of the benchmark specimens (i.e. samples O1-3) which were analysed using the 'fixed-pinned' end conditions failed through buckling at approx. 91% of the design load. This indicates that more a rigorous frame buckling instability analysis should in future be undertaken.

When increasingly fine nodal discretizations are used, layout optimisation can be used to provide an estimate of the likely mathematical optimum solution, with this then providing a reference volume (V_0) for future design studies. However, there is a need to identify design solutions which are more practical (i.e. which do not contain numerous thin members, which are difficult to manufacture using additive manufacturing techniques and/or which are susceptible to buckling). This can be achieved by ensuring the optimisation formulation

includes appropriate manufacturability and/or buckling constraints. Alternatively, as in this study, coarse nodal discretizations can be used to obtain simpler design solutions, with checks then performed retrospectively. Provided the volume of the resulting component is within an acceptable of margin of the reference volume then this, or indeed any other strategy, can be justified. The main drawback of the strategy adopted here is that some iterations were required in order to obtain a viable design solution. However, the need for iteration could potentially be reduced through the use of a geometry optimisation rationalization step (this involves adjusting the positions of nodes, which in turn leads to removal of many thin elements (He and Gilbert 2015). This could potentially replace the sizing optimisation step described in section 6.2.3.

The present study made use of solid circular cross-sections to keep the solid model generation and fabrication stages as straightforward as possible. However, in future the use of more structurally efficient cross-sections could be used (e.g. cruciform sections, which are more buckling resistant). This will often mean that the volume does not increase following the main optimisation.

The finite element analysis highlighted that stress concentrations will occur at the joints if material is not added to compensate for overlapping elements, as expected. The simple volume expansion algorithm used in this study did successfully address this problem, though a more rigorous volume conservation algorithm could be applied at the joints, potentially reducing the volume of material added in this step.

Although the workflow presented here is comparatively simple, the results from the beam load tests, and the significant mass reduction achieved in the case of the Bloodhound SSC airbrake hinge, serve to demonstrate the potential of the method. A key potential benefit of the method is that the process of transforming the optimisation result into a feasible CAD model is simpler than when using many existing continuum based methods (these methods often require significant manual interpretation and/or post-processing to produce a feasible design, with smooth and well defined surfaces).

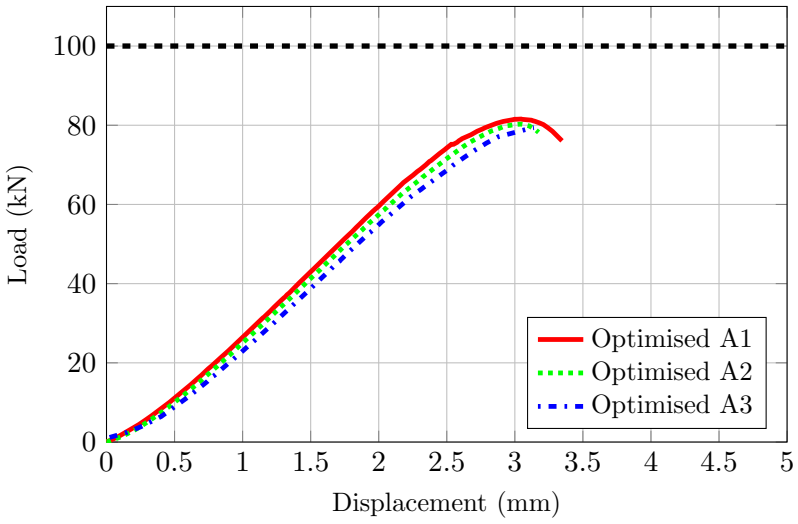
6.5.2 Additive manufacturing

As mentioned earlier, a scaling factor of 1.05 was applied to all element areas of each specimen to compensate for an issue with the EBM process which led to many of the truss bars on the fabricated specimens being undersized. This scaling factor was, however, applied globally, whereas the degree to which the truss bars were undersized was found to vary locally. For example, only truss bars not aligned to the build direction (i.e. not vertical) appeared to be affected and the general trend was that bars with a greater angle to the build direction were affected more. However, rather than developing more specific scaling factors, it is probably more worthwhile to develop more robust EBM process themes, tailored for truss type structures.

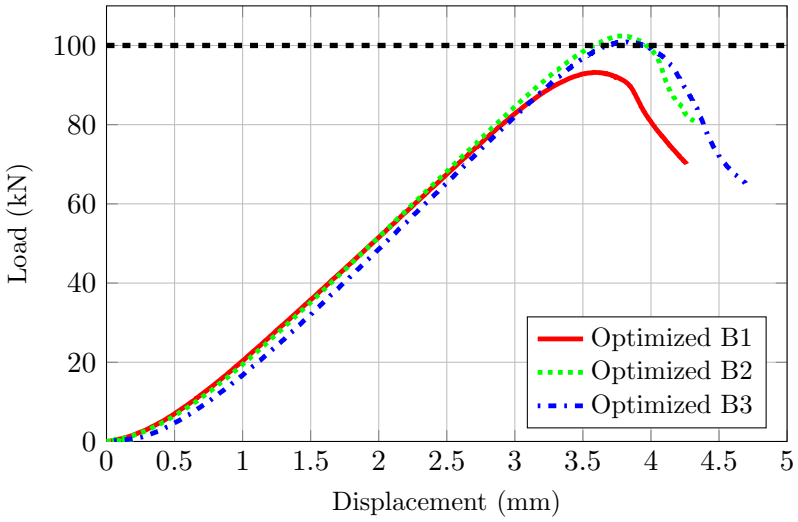
6.6 Conclusions

- A workflow in which layout optimisation is used to automatically design components suitable for fabrication via additive manufacture has been developed.

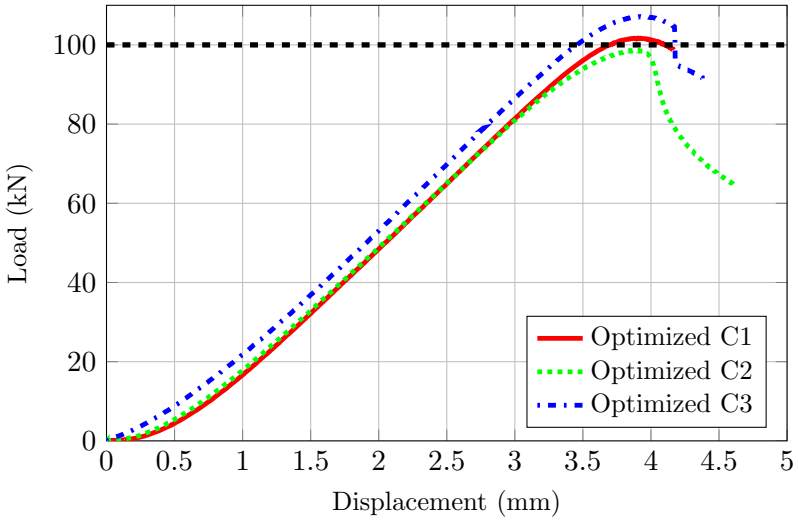
- Beam specimens designed using the proposed workflow, and fabricated using titanium Ti-6Al-4V, generally met or slightly exceeded the target design load capacity.



(a)



(b)



(c)

Figure 6.14: Example 1: load-displacement curves for samples (a) A1-3 (fixed-fixed), (b) B1-3 (fixed-pinned) and (a) C1-3 (pinned-pinned)

Table 6.6: Example 2: Volume and mass data of final Bloodhound SSC air-brake hinge design.

Line model (resized members) volume (cm^3)	Solid Model (truss only) volume (cm^3)	Solid Model (truss only) mass (g)	Solid model (total†) mass (g)	Fabricated mass (g)
35.21	35.27	156.25	726.09	704.4

†Total mass is the truss mass plus the plate and ring structure which were not included as part of the optimisation.

- A reference volume, derived by performing a series of increasingly fine resolution layout optimisation runs, was used to quantify the structural efficiency of the components produced. The two designed components were at least 80% efficient (i.e. were within 20% of their respective reference volumes).
- The volumes of the designed components were also compared with those of simpler benchmark components, of the sort that would be designed by manual means. Although the volume of the optimised beam was just 7% lower than the benchmark, the optimised Bloodhound SSC airbrake hinge component was a far more significant 69% lower than the benchmark. This highlights the potential for applying the proposed workflow to real-world engineering problems.
- The workflow lays a foundation on which further developments can be made. Potential extensions, which should result in greater mass reduction, and/or less need for manual intervention by the user, have been outlined.
- Whilst the Electron Beam Melting (EBM) additive manufacture process employed was successfully able to fabricate all the designs described, various dimensional accuracy issues were encountered which need to be addressed in the future.

Chapter 7

Characterizing the dimension accuracy of the EBM process

The contents of this chapter encompasses the content published in the journal article 'Dimensional accuracy of Electron Beam Melting (EBM) Additive Manufacture with regard to weight optimized truss structures'. However several sections from the article including the introduction and methodology have been omitted as these have been covered in the previous chapters.

7.1 Introduction

In the previous chapters it has been observed that the EBM process manufactures truss structures with undersized members. Dimensional errors have been observed in the literature with regard to the EBM process; Cooke and Soons (2010) found that both EBM and laser processes incurred dimensional errors at least an order of magnitude higher than their machined counterparts. Specifically Cansizoglu et al. (2008) also observed that the EBM process tends to fabricate truss structures with undersized members, particularly those at an angle to the build direction. The observed error seems to mostly occur in regions with no prior melted material beneath them, generally referred to as *negative surfaces*. These features are generally problematic for metallic powder bed AM processes as the lower thermal conductivity of the powder beneath them results in a great deal of heat accumulation leading to part distortion. Even with the use of porous support structures, which increase the thermal conductivity beneath negative surfaces, this particular issue still seems to persist.

The in-service performance implications of undersized members in fully stressed truss structures are potentially very serious. A decrease in cross-sectional area will result in a reduced axial load capacity of the member. Additionally, with material lacking on only one side of the member the second moment of area will be reduced and an effective curvature introduced in an otherwise linear member. Both of these will reduce the resistance to buckling. Just one undersized member can compromise the performance of an entire structure. Thus developing design methods that reduce the sensitivity of topology optimised structures to manufacturing defects has become an active area of research. Schevenels et al. (2011), have for example, made use of probabilistic methods to produce more robust topology optimised designs that are less sensitive to the random dimensional errors that occur during manufacturing. The dimensional errors observed for components manufac-

tured by the EBM process do not appear entirely random however and it is therefore likely that these could instead be significantly reduced through process modifications rather than through statistical approaches.

When modifying the process parameters of the EBM process to maintain accuracy it is, however, important to assess the quality of the resulting microstructure, particularly with regard to the presence of internal voids and porosity. In common with laser powder bed AM, the EBM process is known to produce parts with a certain degree of porosity, but these pores are usually small (<100 μm) and their origins are known to be related to trapped gas during the manufacture of the powder or the result of lack of fusion (Tammas-Williams et al., 2015). Their presence is still significant enough to affect fatigue life (as shown experimentally by Edwards et al. (2013)) and can be removed by Hot Isostatic Pressing (HIPing). Larger voids can present themselves if insufficient energy is directed into the melt pool, as modelled and shown experimentally by Bauereißet al. (2014).

To understand these issues affecting complex truss structures, this investigation seeks to probe the capability limits of the EBM process through the manufacture of simple truss specimens. As follows:

1. Quantify the dimensional accuracy for arbitrary truss structures fabricated using the default process parameters as provided by the manufacturer (Arcam) to act as a benchmark. The dimensional accuracy is assessed for members of several diameters over a range of angles relative to the build direction.

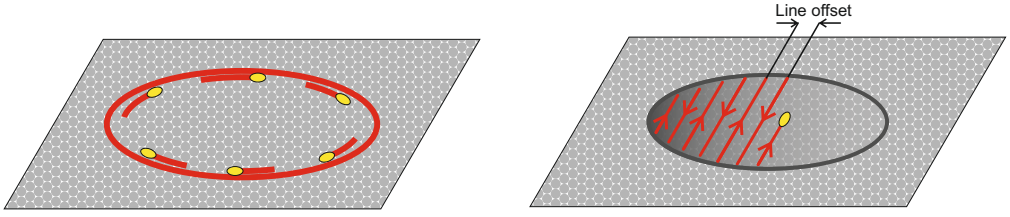
2. The dimensions to be quantified are those directly related to the structural performance of truss structures i.e. cross-sectional area, second moment of area and curvature of each truss member.

3. Attempt to improve dimensional accuracy for single truss members at the largest angle to the build direction explored in the benchmark build by systematic modification of the beam process parameters.

4. Apply the process parameters yielding the best results to the benchmark experiment in step 1 and assess changes in dimensional accuracy that result.

5. Assess the quality of the microstructure (presence of voids and porosity) produced using the modified process parameters.

7.2 EBM process and the thickness function



(a) (b)

Figure 7.1: (a) Contour and (b) hatch stage of the melt.

After powder deposition, the EBM process will preheat (and sinter) the present layer to a specified target temperature (720°C in this case) by scanning the beam across the entire layer at high velocity. The amount of energy required to maintain this temperature is determined by solving a one dimensional thermal model that balances the energy input (based on the area to be melted) with the energy lost by conductance through the powder bed. Once preheating has completed the part cross-section(s) are melted in two stages referred to as *contouring* and *hatching*. *Contouring*, which is used to improve the surface finish of the part, melts the perimeter of the part cross-section using a constant beam power and velocity (as illustrated in Fig. 7.1a). Typically this is done using two passes concentrically offset from one another (referred to as the inner and outer *contouring*). The near instantaneous deflection of the electron beam is used to maintain multiple melt pools simultaneously to reduce the time spent on this stage.

The majority of the melting is performed using *hatching*, which can follow or precede *contouring*. *Hatching* employs a back and forth raster pattern as shown in Fig. 7.1b which, unlike *contouring*, does not use a constant beam power and velocity throughout the build. These two parameters are instead varied to attain consistent melt properties in the face of changing thermal conditions during melting. In total four analytical expressions are evaluated to determine the beam power and velocity to achieve this, which were discussed in detail in Chapter 4. Of these it is the thickness function, which alters the beam energy input when melting in proximity to a negative surface, that is the area of interest for this chapter. Only this function will be modified in this chapter to characterize its effect on dimensional accuracy.

As discussed in Chapter 4, the thickness function reduces the beam energy input by increasing the beam velocity rather than reducing the power, as the latter would require beam current changes at a rate that the system is not capable of attaining. The beam velocity V is altered as per the inverse exponential function of the thickness T in Eqn. 7.1. T is defined as the vertical distance between the present layer and the negative surface beneath it. For a ledge overhang (Fig. 7.2a) the velocity will be maximum V_{max} at the negative surface ($T = 0$) and will then gradually reduce in the subsequent layers until the maximum preset thickness T_{max} is reached, upon which the velocity will return to its original value of V_0. For an angled specimen (Fig. 7.2b) the thickness will vary across each layer and thus so will the velocity. The distance L from the edge that the thickness function is active (darker region shown in Fig. 7.2b) for a specimen at an angle of θ to the build direction is easily calculated using Eqn. 7.2.

$$\frac{V}{V_0} = 1 + \frac{S_f}{exp\left(E_f(T - T_f)\right) + 1} \tag{7.1}$$

$$L = \frac{T_{max}}{tan(\theta)} \tag{7.2}$$

Three tuning parameters known as the Speed Factor (S_f), Exponent Factor (E_f) and Thickness factor (T_f) can be changed to adjust the profile, but for this study only the former two are of interest. The Speed Factor influences V_{max} and the Exponent factor influences the steepness of the profile between $T = 0$ and $T = T_{max}$. The Thickness factor and T_{max} will remain unchanged.

$$E = \frac{P}{Vht} \tag{7.3}$$

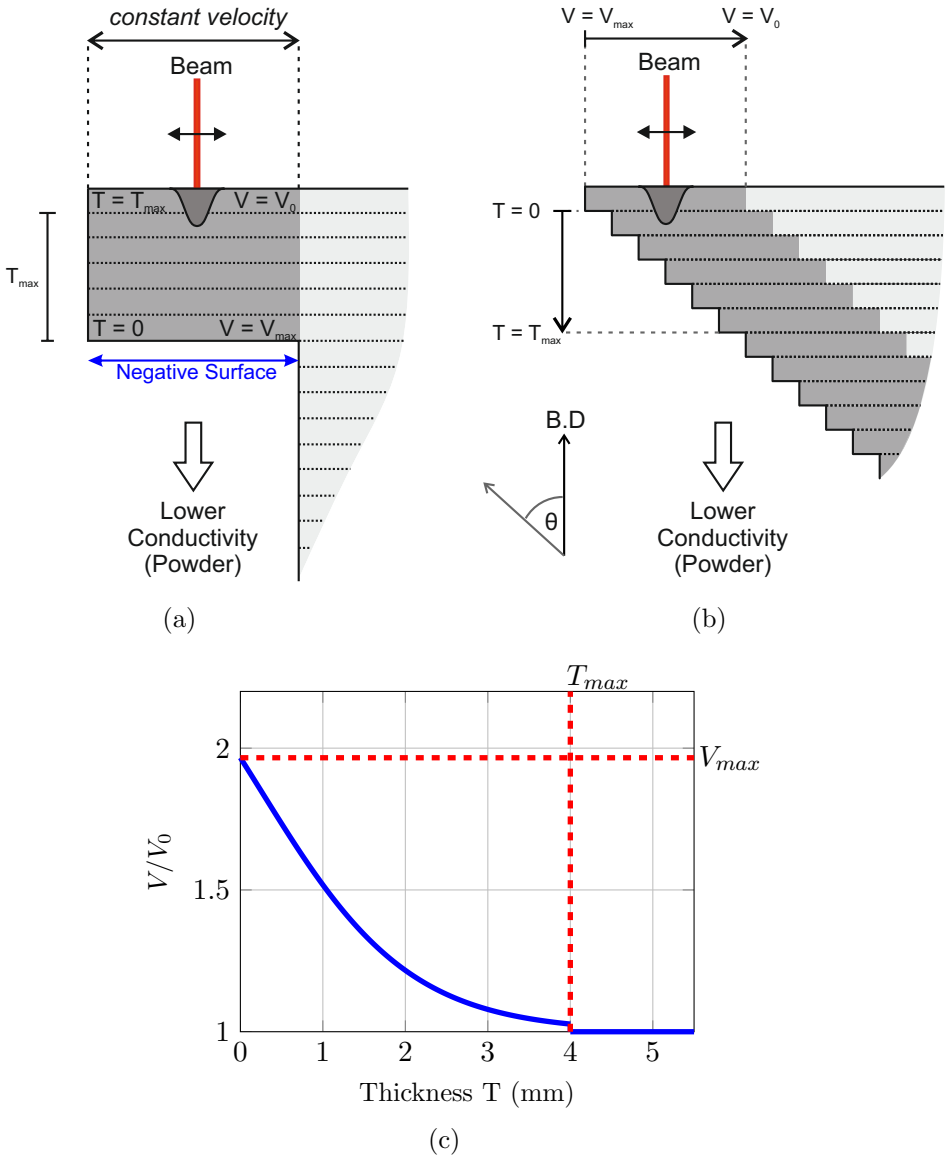


Figure 7.2: Schematic diagram of the EBM process melting overhanging features for (a) a ledge type overhang and (b) truss element at an angle θ to the build direction (B.D). Darker regions show where the thickness function is active. (c) Shows the relative velocity profile with the default thickness function parameters.

The resulting energy input changes from modifying the thickness function velocity profile would be quantified using the applied energy density parameter shown in Eqn. 7.3. This has previously been used in the literature to benchmark changes made to key process attributes of powder bed systems i.e. beam power, beam velocity, line offset (spacing between hatch lines) and layer thickness (Thijs et al. (2010); Attar et al. (2014) ; Tammas-Williams et al. (2015)). Changes to the thickness function will only result in changes to the beam velocity, this expression can be reduced to $E \propto V$. Comparisons between different parameter sets would be performed using an average applied energy density across the region the thickness function is active (darker region in Fig. 7.2b), which would be calculated from the average velocity between $T = 0$ and $T = T_{max}$. This can be obtained by integrating Eqn. 7.1 between these two limits and is shown in Eqn. 7.4. The average

energy densities for each modified thickness function parameter set can then be normalised against that of the default (E_{def}) in Eqn. 7.5. For the default process parameters the average velocity ratio $(V/V_0)_{def}$ is 1.32

$$\left(\frac{V}{V_0}\right)_{avg} = 1 + S_f + \frac{S_f}{E_f T_{max}} log_e \left(\frac{exp(-E_f T_f) + 1}{exp(E_f T_{max} - E_f T_f) + 1}\right) \tag{7.4}$$

$$\left(\frac{E}{E_{def}}\right)_{avg} = \frac{(V/V_0)_{avg}}{(V/V_0)_{def,avg}} \tag{7.5}$$

7.3 Experimental

7.3.1 Experimental builds

All three experimental builds detailed herein were performed using an Arcam A1 EBM machine using Ti-6Al-4V ELI powder and a layer thickness of 50 μm.

First build - benchmark experiment

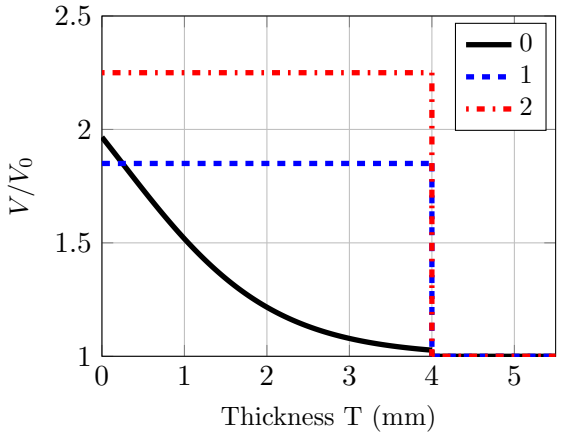
To quantify the dimensional accuracy using default process parameters (listed in Table 7.1) a build comprising fifteen variations of the truss structure shown in Fig. 7.4a was first manufactured to provide a reference data set. Five angles to the build direction were investigated (θ = 20, 30, 40, 50 and 60°) each with three different truss diameters (D = 2, 5 and 10 mm), to observe the effects of both angle and diameter on dimensional accuracy which was quantified through measurement of mass. The lowest (θ = 20°) and highest (θ = 60°) angle specimens at 2 and 10 mm diameters were also assessed using laser scanning and X-ray Computed Tomography (XCT), allowing quantification of the member areas, second moment of areas and member curvature (detailed in sec. 7.3.2).

Second build - process modification experiment

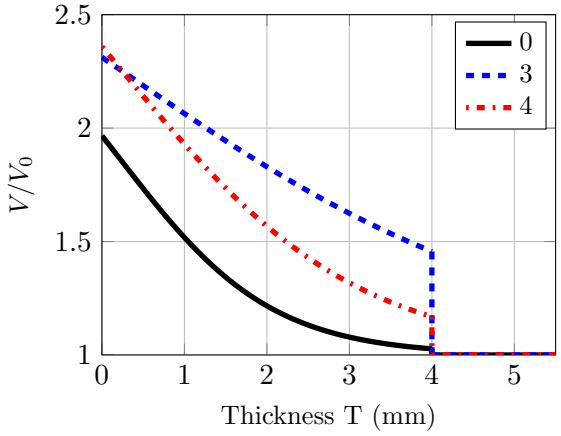
The second build contained groups of individual truss members with diameters of 2 and 10 mm at a constant angle to the build direction, which in this case was set to the maximum presented angle of 60°. An additional member with a square cross-section of side length 10 mm was also included in each group to assess the effect of geometry (if any). Groups of these three individual members would be manufactured for each of the thickness function

Table 7.1: The default and four modified parameter sets investigated

Parameter set	Process Parameters				$\left(\frac{E}{E_0}\right)_{avg}$
	S_f	E_f	T_f	T_{max} (mm)	
0 (default)	1.7	1.1			1.00
1	1.7	0			0.71
2	2.5	0	0.25	4	0.59
3	2.5	0.4			0.71
4	2.5	0.7			0.80



(a)



(b)

Figure 7.3: Velocity ratio plots for (a) Step functions '1' and '2' and (b) Inverse exponential functions '3' and '4'. Default profile is denoted '0'

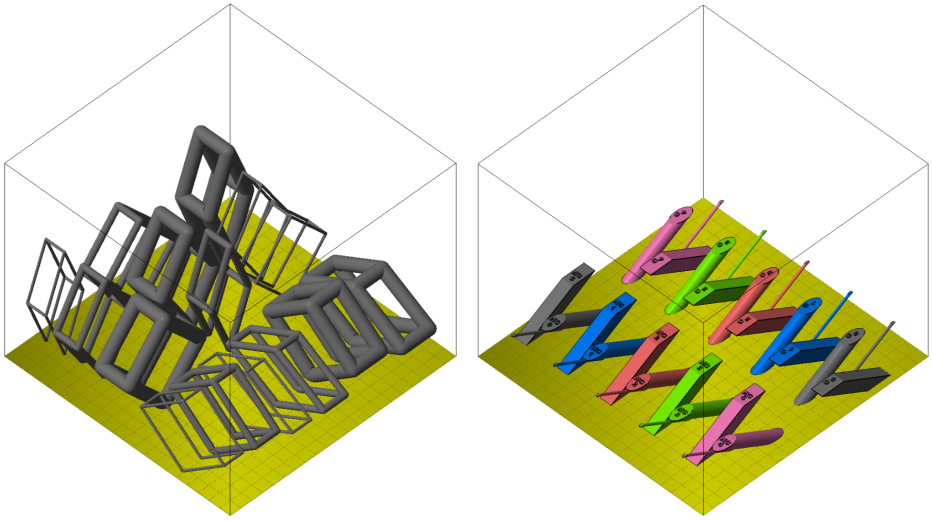
parameter sets shown in Table 7.1 which include the default parameters ('0' - plotted in Fig. 7.2c) and four modified parameter sets that result in a lower average beam energy density. Parameter sets '1' and '2' investigate the use of a constant elevated velocity profile (Fig. 7.3a) whereas '3' and '4' investigate an inverse exponential velocity profile (Fig. 7.3b).

Because the thickness function is only applied in *hatching*, an additional five groups with the same parameters but without the use of *contouring* were also be fabricated. The prefix 'C' and 'NC' denotes specimens made with and without the use of *contouring* respectively. This build with all 10 groups is shown in Fig. 7.4b. The fabrication accuracy would be quantified through measurement of mass.

Third build - application of modified process parameters to benchmark truss structure

The modified parameters that resulted in the best dimensional accuracy were then applied to two of the benchmark truss structures (with $\theta = 60°$ and D = 2 and 10 mm). In addition a third set with the default parameters but with *contouring* disabled were also studied.

Mass measurements and 3D scanning were performed so that any changes in dimensional accuracy resulting from these changes in process conditions could be compared with the results from the benchmark experiment.



(a) (b)

Figure 7.4: (a) Benchmark experiment and (b) process modification experiment.

7.3.2 Assessing dimensional accuracy from 3D scanning of fabricated specimens

To quantify the cross-sectional area, second moment of area and curvature along each member for selected specimens, laser scanning and XCT techniques were employed. A reasonably accurate 3-D surface representation could be created from each of these methods which could then aligned with that of the CAD model (a STereoLithography STL file). Once aligned, multiple planes perpendicular to the member centre lines of the CAD model were generated at equally spaced points along each member (shown schematically in Fig. 7.5b). The intersection of each of these planes with the scanned model would form closed polyline curves representing the member cross-sections (an example is shown in Fig. 7.5c). The coordinates of the polyline vertices were then used to calculate the area, second moment of area and the centroid of the cross-section. The latter would be used to calculate the linear distance between the centroid and the member centre line of the CAD model (labelled d in Fig. 7.5d), which would be the measure of effective curvature.

The coordinates of the vertices were first transformed from their global coordinates into the local plane coordinates 'a' and 'b' using Eqns. 7.6a & 7.6b. This transformation ensures that the 'b' axis is aligned in such a way that it bisects the positive and negative surface of the member from the CAD model (illustrated in Fig. 7.5d). The area of each cross-section can then be calculated using Eqn. 7.7. Following this the error in area and second moment of area about the 'b' axis can be calculated in Eqns. 7.8 & 7.9 respectively. The 'b' axis was chosen as the loss of material is on the negative surface hence the second moment of area about this axis should be the most affected. The curvature along the element δ, defined as the linear distance between the centroid of the scanned cross-section

and the member centre line of the CAD model, is calculated in Eqn. 7.10.

$$a_i = (v_i(x,y,z) - C_0(x,y,z)) \cdot (e_k \times \hat{x}) \tag{7.6a}$$

$$b_i = (v_i(x,y,z) - C_0(x,y,z)) \cdot (e_k \times \hat{z}) \tag{7.6b}$$

$$A = \frac{1}{2} \sum_{i=1}^{N} (a_i b_{i+1} - a_{i+1} bi) \tag{7.7}$$

$$A_e = \frac{1}{A_0} (A - A_0) \tag{7.8}$$

$$I_{b,e} = \frac{1}{12 I_0} \left[\left(\sum_{i=1}^{N} \left(a_i - a_i a_{i+1} + a_{i+1}^2 \right) (a_i b_{i+1} - a_{i+1} b_i) \right) - 12 I_0 \right] \tag{7.9}$$

$$\delta = \frac{1}{6A} \left[\left(\sum_{i=1}^{N} (a_i + a_{i+1}) (a_i b_{i+1} - a_{i+1} b_i) \right)^2 + \left(\sum_{i=1}^{N} (b_i + b_{i+1}) (a_i b_{i+1} - a_{i+1} b_i) \right)^2 \right]^{\frac{1}{2}} \tag{7.10}$$

C_0 are the coordinates of the intersection between the planes and the member centre lines of the CAD model, a_i and b_i are the local plane ordinates of each vertex i ($i = 1, ..., n$) which are calculated from the global coordinates v_i, e_k is the vector of the member centre line and $\hat{x}$ and $\hat{z}$ are unit x (1,0,0) and z (0,0,1) vectors, A is area of the resulting cross-section, A_0 and $I_{0,b}$ are the area and second moment of area (about local b axis) of the CAD model respectively, A_e and $I_{e,b}$ are the errors in area and second moment of area relative to that of the CAD model.

The 10 mm diameter specimens were scanned using a Metris LC15 laser scanner. The 2 mm diameter specimens were scanned using the Nikon Metris Custom Bay XCT apparatus at the University of Manchester Henry Mosely facility. All analyses used 100 intersecting planes per member.

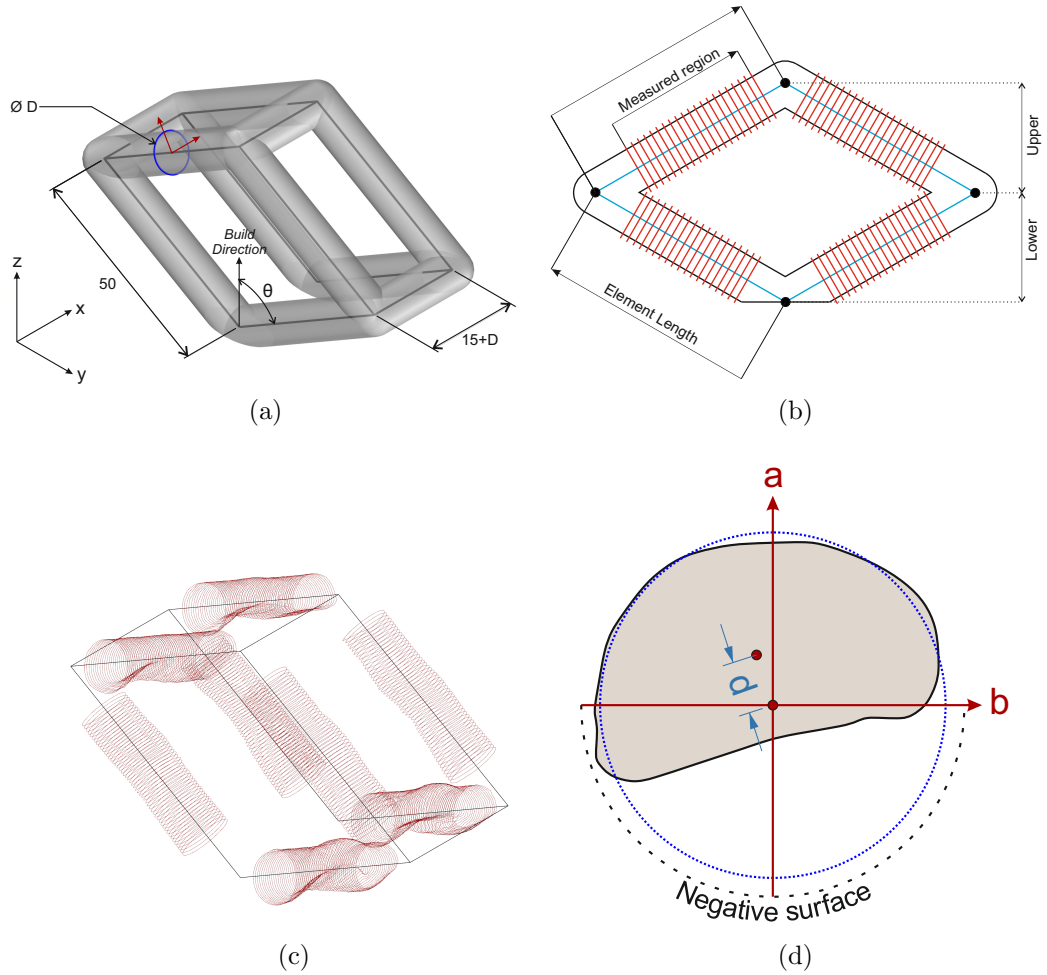


Figure 7.5: (a) Dimensions of the benchmark truss structures (b) Schematic showing the location on the measuring planes which are normal to the member centre lines of the CAD model (c) an example showing the polyline cross-sections obtained from these planes intersecting the scanned model and (d) the local coordinate system for each plane. The shaded region shows an example polyline cross-section and the dotted outline shows the cross-section of the CAD model.

7.3.3 Metallurgical evaluation

To investigate the effect of process parameter adjustment on the internal structure of the experimental builds, selected specimens would be sectioned with a low speed precision cut-off wheel and prepared for metallurgical evaluation. Samples would initially be planar ground using water-lubricated SiC paper which would then be followed by the use of a 9 μm diamond suspension for the final grinding step. Final polishing of the specimens would be performed using colloidal silica suspension.

The size-distribution of internal features (either gas porosity of lack of fusion defects) was assessed optically by examining four regions of interest per condition to provide a total field of analysis of 16.8 mm^2 An automated thresholding procedure in the image analysis software ImageJ was employed to quantify the number and size (Feret Length) of defects.

7.4 Results

7.4.1 Benchmark experiment

The measured mass error (M_e) for all fifteen specimens are presented in Fig. 7.6 and the results of the 3D scanning analysis are given in Table 7.2. Here the mean average error and standard deviation of the area (A_e) and second moment of area ($I_{b,e}$) of all eight members for each structure has been calculated. The average of the maximum curvature (δ_{max}) of all 8 members per specimen is also shown.

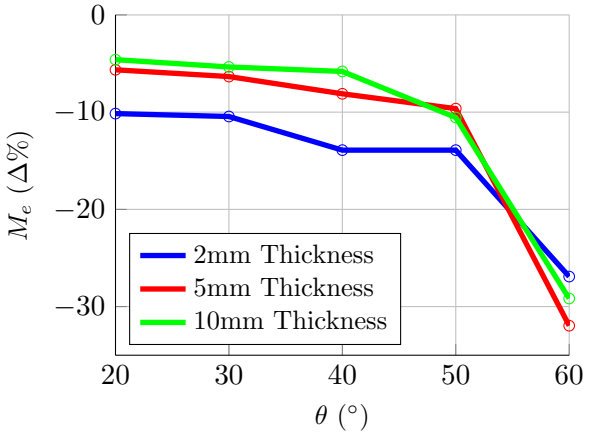


Figure 7.6: Mass error (M_e) of fabricated specimens relative to their respective CAD models

Table 7.2: Area error (A_e), second moment of area error ($I_{b,e}$) and maximum curvature (δ_{max}) averaged for all 8 members of each of the benchmark truss structures. μ and σ denote the mean and standard deviation respectively.

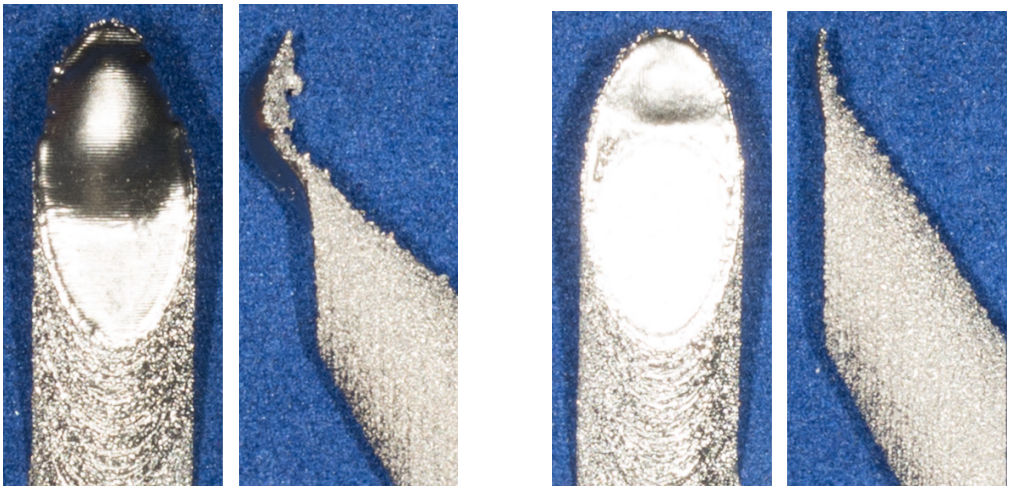
Specimen		Mass error	Area error A_e (Δ%)		2^{nd} moment of area error $I_{b,e}$ (Δ%)		Curvature δ_{max}
θ (°)	D (mm)	M_e (Δ%)	μ	σ	μ	σ	(mm)
20	2	-10.2	-8.55	3.67	-14.2	5.76	0.176
	10	-4.60	3.64	1.81	-7.58	1.79	0.0739
60	2	-41.2	-40.2	11.3	-48.5	11.4	0.423
	10	-27.1	-25.5	11.3	-51.0	17.6	1.23

7.4.2 Process modification experiment

The mass error (M_e) for each of the three singular members in the ten sets are given in Table. 7.3. Photographs taken of the 10 mm diameter circular members from the C0 and C1 sets are presented in Fig. 7.7.

Table 7.3: The resulting mass errors measured from each of the three geometries fabricated using the parameter sets shown in Table 7.1. The prefix 'C' denotes parameters sets that used *contouring* 'NC' denotes those that did not. An average mass error for all three geometries for each parameter set is also shown

	Label	$\left(\frac{E}{E_0}\right)_{avg}$	M_e (Δ%)			Average
			Circular		Square	
			10mm	2mm	10mm	
Contouring Enabled	C0	1.00	-35.9	-33.4	-30.5	-33.3
	C1	0.71	-28.6	-46.5	-23.7	-32.9
	C2	0.59	-16.7	-30.5	-14.9	-20.7
	C3	0.71	-24.7	-34.9	-18.7	-26.1
	C4	0.80	-28.2	-39.2	-23.0	-30.1
Contouring Disabled	NC0	1.00	-14.3	-5.9	-12.3	-10.8
	NC1	0.71	-4.7	-8.8	-3.0	-5.5
	NC2	0.59	-4.9	-10.3	-3.7	-6.3
	NC3	0.71	-12.7	-11.7	-14.5	-13.0
	NC4	0.80	-19.2	-13.2	-16.5	-16.3



(a) (b)

Figure 7.7: Photographs of two 10 mm diameter circular truss members fabricated with (a) the default parameter set - C0 and (b) C1 parameter set, viewed from the top (left) and side (right)

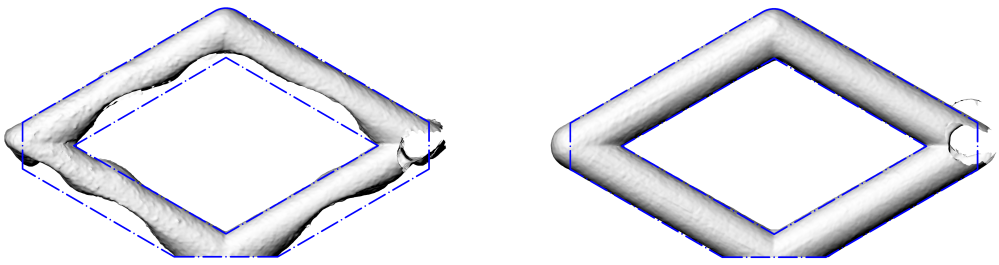
.

7.4.3 Application of modified process parameters to benchmark truss structure

The two best parameter sets (as defined by those causing the lowest deviation in mass - NC1 and NC2) from Sec. 7.4.2 as well as the default parameters without *contouring* (NC0) were then used to manufacture the "benchmark" truss structure ($\theta = 60°$ and diameter D = 2 and 10 mm). The resultant average mass error, area error, second moment of area error and maximum curvature for each specimen are given in Table 7.4. The variation of latter three parameters along the member lengths (grouped for the upper and lower members) is shown in Figs. 7.9 & 7.10 for the 2 and 10 mm diameter specimens respectively.

Table 7.4: The measured errors in mass (M_e), cross-sectional area (A_e), second moment of area error ($I_{b,e}$) and maximum curvature (δ_{max}) for the 2 and 10 mm diameter benchmark truss structures for each process parameter set. Note that M_e, A_e, $I_{b,e}$ and δ_{max} have been averaged over all 8 members for each specimen.

D (mm)	Parameter set	M_e (Δ%)	A_e (Δ%)		$I_{b,e}$ (Δ%)		δ_{max} (mm)
			μ	σ	μ	σ	
2	C0	-41.2	-40.2	11.3	-48.5	11.4	0.423
	NC0	-9.0	-2.72	9.71	15.9	12.9	0.256
	NC1	-5.4	2.36	7.66	23.9	9.24	0.206
	NC2	-10.5	-0.39	4.23	18.6	14.3	0.281
10	C0	-27.1	-25.5	11.3	-51.0	17.6	1.23
	NC0	-17.2	-18.4	10.5	-41.5	18.0	0.613
	NC1	-5.5	6.35	3.26	6.18	9.51	0.139
	NC2	-2.8	9.12	1.40	13.8	4.10	0.108



(a) (b)

Figure 7.8: Side profile of scan for the 10 mm diameter benchmark truss structure fabricate using (a) default C0 process parameters (b) NC2 process parameters. Dashed outline shows the boundaries of the CAD model

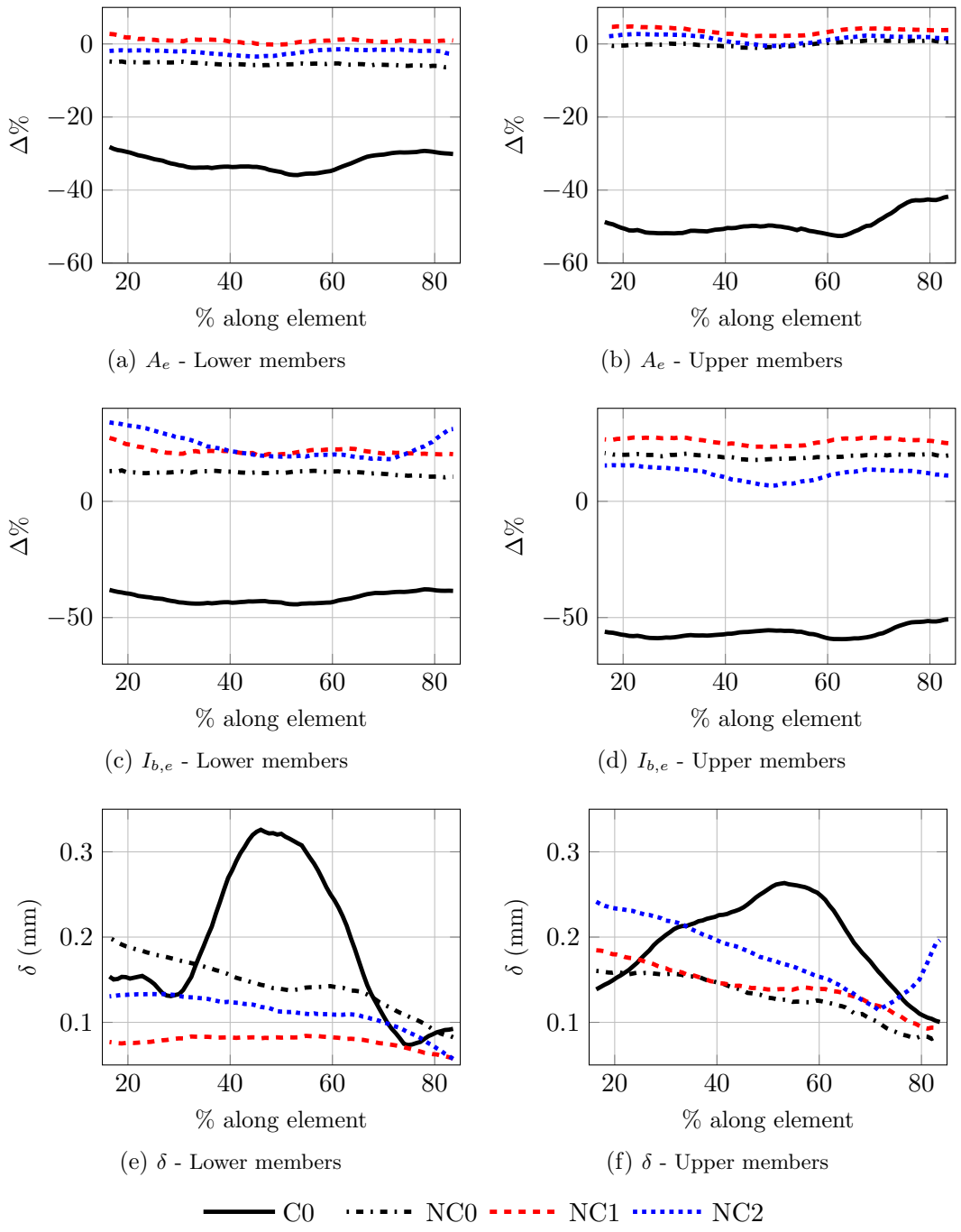


Figure 7.9: The variation of the cross-sectional area error (A_e), second moment of area error ($I_{b,e}$) and curvature (δ) along the member lengths for the lower and upper members (denoted in Fig. 7.5d) of the 2 mm diameter benchmark truss structure. Shown in each plot is the average curve for the four members in each of these two groups.

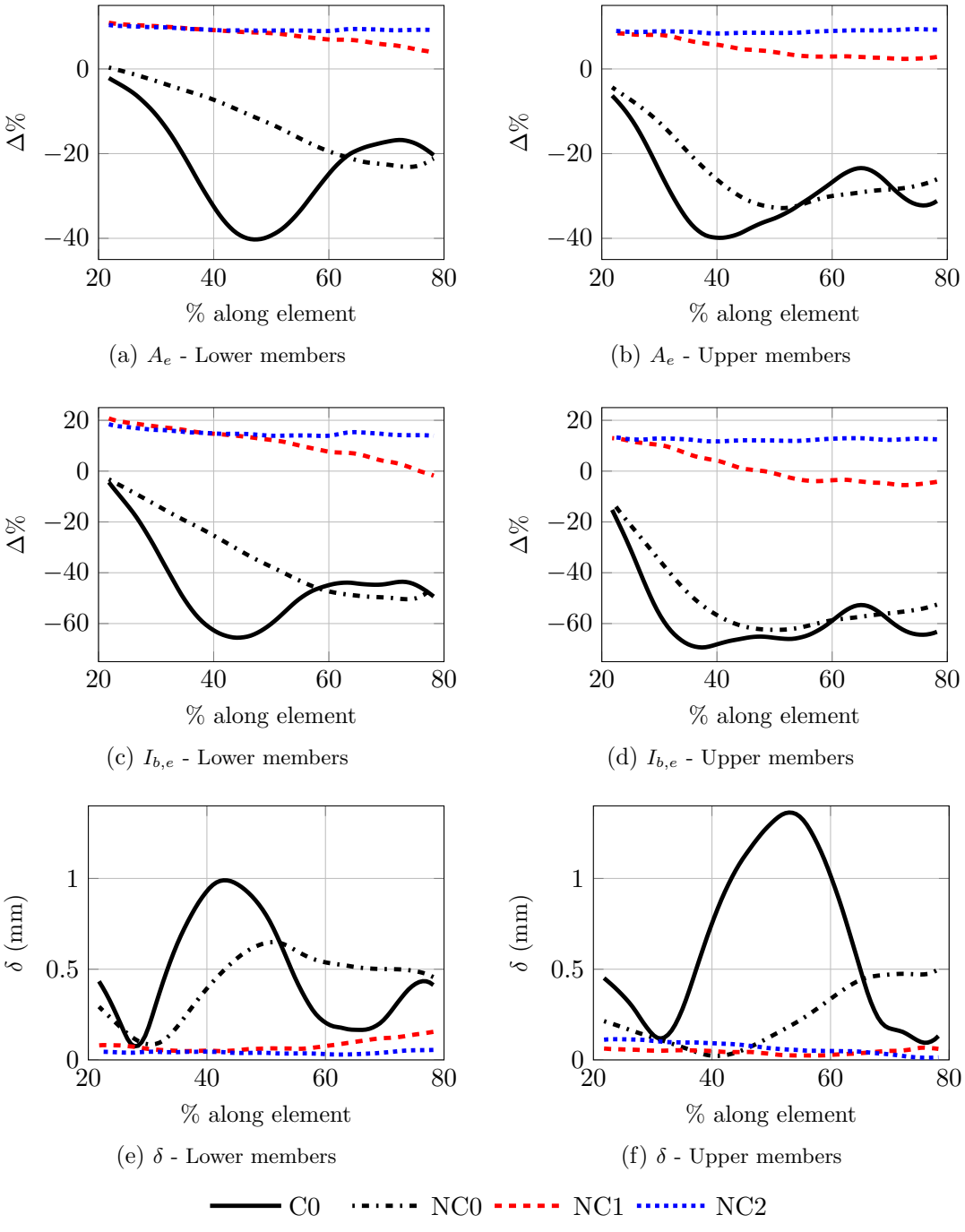


Figure 7.10: The variation of the cross-sectional area error (A_e), second moment of area error ($I_{b,e}$) and curvature (δ) along the member lengths for the lower and upper members (denoted in Fig. 7.5d) of the 10 mm diameter benchmark truss structure. Shown in each plot is the average curve for the four members in each of these two groups.

7.4.4 Porosity

The NC1 and NC2 cylindrical specimens (10 mm diameter) that were fabricated in the experiment detailed in section 7.4.2, were sectioned at the mid-point on a plane perpendicular to the member centreline and then prepared for optical microscopy as outlined in section 7.3.3. The purpose of this analysis was to assess whether adjustments to the processing strategy to minimize mass loss also influenced the population of internal defects. Example micrographs illustrating the typical internal structure of specimen NC1 and NC2 are given Fig. 7.11. In both cases, large and irregular-shaped lack of fusion defects were observed along with sub-100 μm spherical pores, with the latter probably originating from the gas atomised feedstock powder.

Fig. 7.12 is an empirical cumulative probability plot showing the size distribution of internal defects sampled from specimens NC1 and NC2. Data was collected from equal areas of analysis from both specimens. Although the defects sampled from specimen NC1 are marginally larger (in terms of the distribution of Feret Lengths) than those sampled from NC2 on the whole, the number density of features observed per unit area of analysis (N_A) within NC1 is considerably smaller than in specimen NC2 (8.5 mm^{-2} versus 33.7 mm^{-2}).

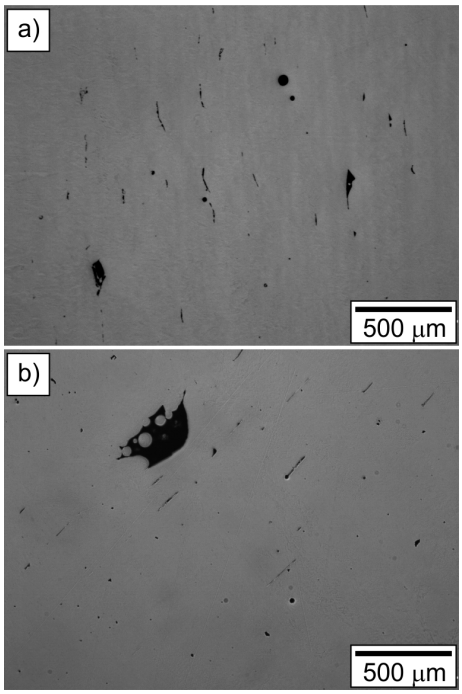


Figure 7.11: Photographs obtained through optical microscopy for (a) the NC1 and (b) NC2 specimens

.

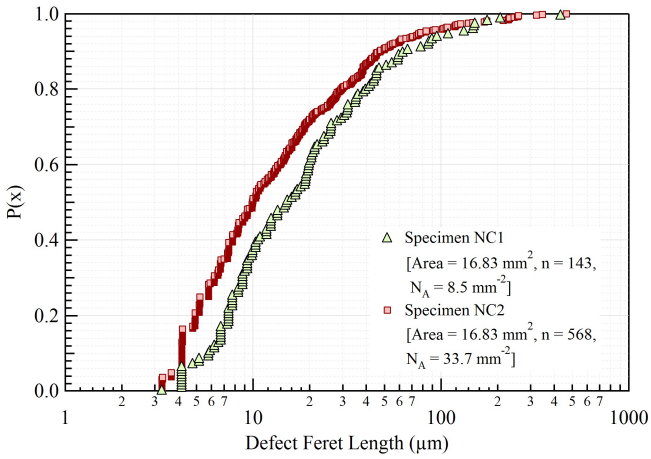


Figure 7.12: Empirical cumulative probability plot showing the size distribution of defects sampled from specimens NC1 and NC2. The total area of analysis for each specimen is A = 16.83 mm^2, with n = total number of observed defects and N_A = calculated number density of defects per unit area of analysis.

7.5 Discussion

7.5.1 Overview

The EBM process will as a matter of course fabricate truss structures with dimensional errors if the beam energy input in close proximity to negative surfaces is not modified adequately, all fifteen truss structures in the benchmark experiment had undersized members. The magnitude of these errors increase with a greater relative orientation of the truss members to the build direction as shown in Fig. 7.6), however it has been demonstrated in this investigation that even for the worst observed case (60° to the build direction) the degree of inaccuracy can be mitigated significantly when the beam energy density is suitably reduced (-41.2% to -5.4% in one case with regard to the mass error).

The reason for the dimensional inaccuracy can be seen in the photographs of two specimens in Fig. 7.7. In Fig. 7.7a a single member has been fabricated using the default process parameters and this has resulted in a significant degree of distortion which has caused material to rise above what would have been the final layer during the build. The consequence of this "peeling up" effect is that powder will not have been properly deposited at this region in the subsequent layers. With the amount of heat input reduced, (Fig. 7.7b) it can be seen that the degree of part distortion is significantly lower and from the side there is visibly less material loss on the negative surface.

7.5.2 Benchmark experiment

The degree of the distortion becomes more prominent the greater the angle of the member orientation to the build direction, but as can be seen from Table 7.2 even members at a relatively small angle of 20° are still prone to this issue. In the case of the 2 mm diameter truss structure the average cross-sectional area and second moment of area errors were -8.6 and -14.2% which would result in an *equivalent* reduction in tensile/compressive strength and buckling capacity. At 60° however, these errors increase to -40.2 and -48.5 % for the 2 mm specimens. The 10 mm specimens at 20° are less compromised (A_e = -3.64 $I_{b,e}$ =

-7.6%) whilst, in contrast to the 2 mm diameter trusses, at 60° this becomes even more significant (A_e = -25.5 $I_{b,e}$ = -51%) and a large effective curvature is introduced (δ_{max} = 1.23 mm).

What compounds these large errors in area, second moment of area and member curvature is that they are not consistent along the member lengths and would be hard to compensate for using scaling factors alone. This was especially true for the 10 mm diameter specimens where in Fig. 7.10 all three error measures (for C0) fluctuate significantly along the members lengths. This almost sinusoidal fluctuation can also be observed directly from the visual representation obtained from the 3D scan as shown in Fig. 7.8a.

7.5.3 Process modification experiment

The data in Table 7.3 reveals that a reduction in the average beam energy density resulted in a reduced mass error, but what is more striking is that samples fabricated without the use of *contouring* were significantly more accurate than those fabricated with both *contouring* and *hatching*. This is most obvious with the NC0 parameters where, still using the default thickness function velocity profile, the average mass error for the three geometries was reduced from -33.3% to -10.8%. Even more noticeable is that the 2 mm diameter specimen was fabricated most accurately when using the default thickness function profile without *contouring*. Although this is perhaps not too surprising as the contour stage in these specimens is melting a far larger proportion of the area, and the thickness function does not apply to this stage of the melting process. With no compensation for the lower conductivity of the powder beneath negative surfaces, the contour stage is depositing an excess energy into these regions.

Of the two types of velocity profile explored, those with a constant velocity yielded the better results than those employing the velocity scaling function given in Eqn 7.1. In fact the NC1 and NC3 specimens had the same average energy density but the former using a constant velocity profile proved the more accurate of the two. This highlights the importance of the beam velocity profile as well as the overall quantity of energy input.

7.5.4 Application of modified process parameters to benchmark truss structure

Both the constant velocity profiles NC1 and NC2 were applied to the fabrication of two variations of the benchmark truss structures (θ = 60° for D = 2 & 10mm) with very promising results. First of all it can be seen in Table 7.4 that the average for all three error measures has been dramatically reduced over the default C0 parameters. For the 10mm specimens a reduction from -25.5% to +6.35% was found when using the NC1 parameters. As with the experiment discussed in section 7.5.3 most of the improvement with the 2 mm diameter specimens were achieved through disabling *contouring*, with marginal differences between the NC0, NC1 and NC2 parameters.

Whilst NC1 resulted in slightly lower average errors for the 10 mm diameter specimens, it can be seen in Fig. 7.10 that these error measures were more consistent along the member lengths for NC2. What is also noticeable is that both the NC1 and NC2 produce oversized members. As the second moment of area is proportional to the cross-sectional area squared, this error is particularly large (13.8% for NC2), but this is most probably caused by the lack of a proper offset in place between the *hatching* path and the boundary of the cross-section of the part i.e. the beam is melting material outside of the perimeter

defined by the CAD model. Because the error is consistent along the member lengths a simple remedy for this would be the implementation of such an offset, this being a trivial matter to achieve.

The effective curvature of the 2 mm diameter members was difficult to determine in an absolute manner as the specimens may have been deformed when being manipulated for scanning. This could potentially be the reason for the great variation in Figs. 7.9e & 7.9f, nevertheless the effective curvature has been approximately halved for all three non-contour parameter sets NC0, NC1 and NC2 as compared with their *contoured* equivalents C0, C1 and C2. For the sturdier 10 mm samples, when compared to the default C0 parameters the curvature has been reduced by 89 and 91% for NC1 and NC2 respectively, and as can be seen in Figs. 7.10e & 7.10f the curvature along the length of the members is very low.

The improvements in dimensional accuracy are most easily observed in Fig. 7.8 where the side profiles of the laser scanned C0 and NC2 specimens are shown with an overlay of the CAD model.

7.5.5 Porosity

Although the NC1 specimen had a decreased defect population density versus NC2 the size of the features are still unacceptably large. For both of these parameter sets there is clearly insufficient heat input per unit volume of material on the powder bed to guarantee a fully dense part. Further modifications to these parameters would therefore aim to determine the energy density that yields an effective compromise between high dimensional accuracy and a low defect population density. Referring back to Eqn. 7.3 this would involve further modification to the beam velocity and/or changes to the line offset (distance between hatch lines). Changes to the T_{max} variable should also be investigated as this determines the proportion of the melt area for which the thickness function is active (as per Eqn. 7.2) and this was also where the large lack of fusion defects commonly appeared.

It should be noted that the defect population data acquired from both specimens was sampled from two-dimensional planar cross-sections. The use of Micro XCT analysis, which could consider the whole volume, would considerably increase the reliability of the data.

The pores seen with the NC1 and NC2 parameters could be easily eliminated (or at least significantly reduced) by Hot Isostatic Pressing. Even parts with the more commonly observed sub 100 μm pores would require this step to ensure acceptable fatigue life, but it must be accepted that HIP routines do result in a slight reduction in tensile strength due to coarsening of the Widmanstatten alpha lath structure (Al-Bermani et al., 2010).

7.6 Conclusions

- All truss members at an angle to the build direction fabricated with the default process parameters provided by the manufacturer Arcam were undersized. This issue is exacerbated as the angle to the build direction increases

- Compensating for these errors using linear scaling factors would be ineffective as magnitude of the errors vary considerably along the truss members

- The reason for this inaccuracy was observed to be an excessive beam energy density at regions in proximity to negative surfaces. This resulted in part distortion that compromised the powder deposition and melting stages in the subsequent layers.
- Reducing the beam energy density by increasing the beam velocity of the hatch melt stage when melting in proximity to a negative surface and discarding the contour stage during melting resulted in significant improvements in dimensional accuracy.
- But in reducing the energy density there was an increase in the number and size of pores present due to lack of fusion in localised regions.
- These pores could be eliminated using HIP routines, a step that would also be required to eliminate the sub 100 μm spherical pores more commonly observed in EBM parts.
- Further refinement of the process modifications detailed herein would seek to find the energy input that is an acceptable balance between good dimensional accuracy and low defect population density

Chapter 8

Discussion

It has been demonstrated that layout optimisation can be used to produce practical layouts that can then be made ready for additive manufacture. The process of transforming the raw numerical result into a finalised CAD model would appear to be far less involved than that associated with mesh based approaches such as SIMP and ESO/BESO. Layout optimisation can solve very large problems that provide a very good approximation of the analytical solutions for Michell type truss structures. Although these layouts are too complicated to manufacture, the resulting volume serves as a reference point for benchmarking more practical layouts where no such analytical solutions exists. This reference volume can be seen as analogous to the Carnot efficiency of a heat engine, a theoretical maximum value for the achievable thermal efficiency. Practical solutions were then sought that were within a reasonable margin of this reference point (within 20%; i.e. 80% structurally efficient)

The load test performance of the optimised specimens for the initial case study, the cantilever problem, failed at just 55% of the target load. These solutions (particularly Cantilever 2) comprised many overlapping members and cross-over joints that were not properly accounted for in the transformation from a line structure to a solid model, likely leading to elevated regions of stress during load testing. The creation of a work-flow based around layout optimisation, which modified the layout to prevent the occurrence of elevated regions of stress, proved to be successful for the second case study - the MBB beam problem. The majority of these specimens reached the target load. The work-flow was then applied to the Bloodhound air-brake hinge problem, where the potential of layout optimisation for producing minimal mass solutions for real world problems was demonstrated, with a mass reduction of 69% over the original, conventionally designed, part.

For all three cases explored in this thesis, determining a layout that satisfied equilibrium requirements, was feasible to fabricate and within a reasonable margin of the reference volume proved to be an iterative procedure requiring some manual input. The majority of the manual input was in determining a suitable nodal distribution that would yield a sensible solution. The work detailed in this thesis lays the groundwork for further development that could reduce the amount of manual input required. It also allows for further extensions to the optimisation formulation and an enhanced work-flow that could result in even more efficient, practical layout optimised structures. Potential avenues of further research and development of this optimisation methodology will be discussed in the following sections. Issues that presented themselves in relation to the additive manufacturing process, such as internal porosity, surface finish and dimensional errors are also discussed.

8.1 Yielding practical layouts

8.1.1 Use of Mixed Integer Linear Programming (MILP)

The work-flow detailed in Chapter 6 applied geometric rules to eliminate overlapping members as a post-optimisation procedure. But the use of a Mixed Integer Linear Programming (MILP) formulation would permit these rules as well as a minimum member area constraint to be implemented in the primary optimisation problem. Each member would be assigned a binary value (representing true or false) for each of the geometric and the minimum area constraints. If a solution has any member that violates these constraints then it will not be valid and another solution will then be sought in the optimisation. MILP is not as efficient as the LP formulation and so would come at a cost of longer solution times and size of the problem that can be solved. So any implementation of it would probably involve it being activated in the latter iterations of the LP problem. Whether an MILP formulation would add much value over the work-flow that will be proposed next, involving the use of a geometry optimisation step, would need to be assessed.

8.1.2 Use of geometry optimisation

All the case studies in this thesis made use of a much coarser discretisation to yield layouts that were much simpler but are still within a reasonable margin of the derived reference volume. The problem with using fewer nodes is that their relative positions within the design domain has a greater influence on the solution, making the procedure of finding a practical and structurally efficient layout much more iterative and heuristic. This was most obvious with the air-brake hinge problem, which required a parametric model (in place of a cubic nodal grid) that would place nodes along splines that followed the profile of the design domain. Geometry optimisation could potentially eliminate this limitation by including the nodal positions as variables in the optimisation problem.

The use of geometry optimisation with a fully connected ground structure would be very computationally expensive so a much more efficient implementation would be to use the result from layout optimisation as the starting point. In the work-flow described in Chapter 6, geometry optimsation would be used in place of the sizing optimisation step.

Geometry optimisation formulation

He and Gilbert (2015) have proposed a method suitable for treating trusses comprising $\mathbb{N} = \{1, 2, ..., n\}$ nodes for load cases $\alpha = \{1, 2, ..., F\}$, with the formulation shown in Eqns. 8.1. In this formulation the objective function (8.1a) and both the static equilbrium and limiting stress constraints (8.1b) remain the same as in the layout optimisation problem. But, with the addition of the three position variables (x, y, z) for each node, the problem is now non-linear and potentially non-convex. Because of this there is no guarantee that the solution will be a global optimum. But it can be argued that this is not an issue as the starting layout is one derived from layout optimisation for which a global optimum is guaranteed (for a given numerical discretisation) and the geometry optimisation step can only result in an objective function (total volume) that is either the same or lower.

Several extra constraints are also included to maintain the robustness of the optimisation problem. Eqn. (8.1c) restricts the maximum permissible movement of the nodes on each

iteration to prevent the occurrence of zero length bars. The position of node 'j' ($\boldsymbol{\nu}_j$) relative to its initial position ($\boldsymbol{\nu}_j^0$) cannot be more than the half distance (r) between node 'j' and the nearest adjacent node and a specified gap (ϵ). Eqn. (8.1d) ensures that the nodes in proximity to the design domain boundaries (in the subset $\mathbb{N}^D$) do not move outside of the boundaries (8.1e). For a 2-D problem the design domain can be represented with a series of line equations in the form $T^x x + T^y y + T^c = 0$. These can then be applied as an inequality in the form $T^x x + T^y y + T^c \geq 0$, with the sign of the coefficients determining which side of the line (i.e. the direction of the normal) is the inside of the design domain. But for a 3-D problem the design domain would have to be represented by a series of surface equations, which would be much more complicated to implement. Because the design domain for the cantilever problem is a simple box, line constraints can still be used to represent the edges. $\mathbf{T}_{j_D}^D$ contains the coefficients $[T^x, T^y, T^z, T^c]$ for $\boldsymbol{\nu}_{j_D}$ the nodal positions and constant term $[x, y, z, 1]^T$.

The problem is solved using the interior point method with the stopping criteria based on the Karush–Kuhn–Tucker conditions.

$$\min_{x,y,z,a,q^\alpha} \quad V = \mathbf{l}^T\mathbf{a} \tag{8.1a}$$

s.t.

$$\left.\begin{aligned} &\mathbf{B}\mathbf{q}^\alpha = \mathbf{f}^\alpha \\ &\sigma^-\mathbf{a} \leq \mathbf{q}^\alpha \leq \sigma^+\mathbf{a} \end{aligned}\right\} \text{for all } \alpha \in \mathbb{F} \tag{8.1b}$$

$$\left\|\boldsymbol{\nu}_j - \boldsymbol{\nu}_j^0\right\|_2^2 \leq (r-\epsilon)^2 \quad \text{for all } j \in \mathbb{N} \tag{8.1c}$$

$$\mathbf{T}_{j_D}^D \boldsymbol{\nu}_{j_D} \geq \mathbf{0} \quad \text{for all } j_D \in \mathbb{N}^D \tag{8.1d}$$

$$\mathbf{a} \geq \mathbf{0} \tag{8.1e}$$

$$\mathbf{x}^{lb} \leq \mathbf{x} \leq \mathbf{x}^{ub} \tag{8.1f}$$

$$\mathbf{y}^{lb} \leq \mathbf{y} \leq \mathbf{y}^{ub} \tag{8.1g}$$

$$\mathbf{z}^{lb} \leq \mathbf{z} \leq \mathbf{z}^{ub} \tag{8.1h}$$

Application to the cantilever problem

To assess the effectiveness of geometry optimisation, a very preliminary investigation was performed using the Cantilever 2 layout, which was chosen because it was the simplest problem explored in this thesis. The layout was modified as per the work-flow detailed in Chapter 6 to remove overlapping members and expand the joints at all cross-over points. The load in the sizing optimisation step was reduced from 100 kN to 50 kN to facilitate the use of a true cantilever loading test scenario (shown in Fig. 8.2). It can be seen in Table 8.1 that the layout produced using the work-flow (denoted modified) now comprises more members than the original for Chapter 5 (because of members being split at detected cross-over points) and has a total truss volume that IS much closer to that of the resulting solid model, indicating that there is a much lower degree of overlap of the members. It should be noted that because of the lower applied load of 50 kN, the reduction in member areas has resulted in more members being resized to resist buckling in both the original and modified layouts, leading to a larger total volume relative to the reference volume.

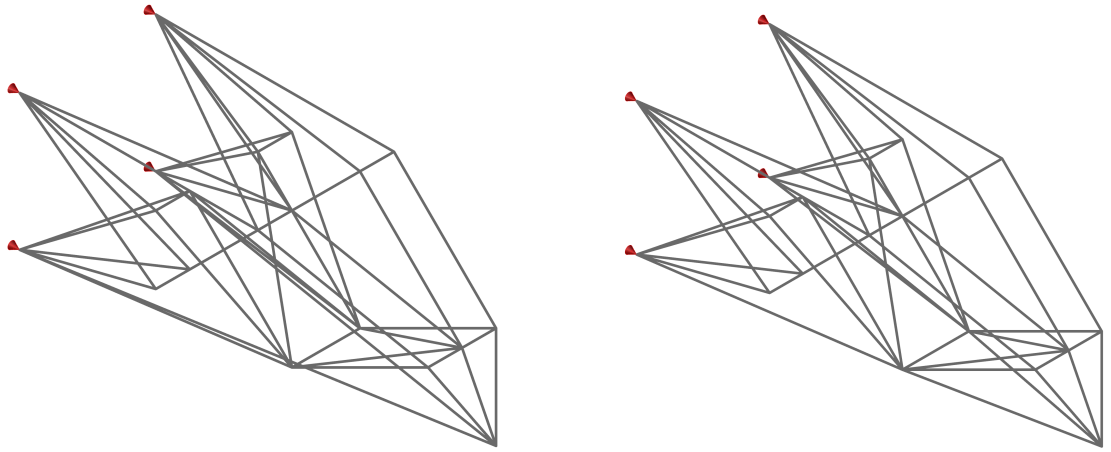
The modified layout was used as the starting layout for the geometry optimisation, which, apart from the applied load, used the same material properties and disturbing forces as the original problem in Chapter 5. It can be seen in Table 8.1 that geometry optimisation

has simplified the layout (56 members as opposed to 83) and had also reduced the volume slightly (which becomes more significant once minimum area and buckling considerations have been accounted for). The original, modified and geometry optimised layouts for the cantilever 2 solution are shown in Fig. 8.1. Upon load testing two instances of the modified layout and one of the geometry optimised layout it can be seen if Fig. 8.3 that the ultimate loads obtained were very similar, with the geometry optimised specimen marginally higher. It is also noticeable that whilst the specimens performed better than those in Chapter 5, they still failed below the target load. All specimens failed through buckling of one or more members (see Fig. 5.13) which may have been due to taking insufficient account of frame stability considerations and/or the dimensional errors known to occur within the EBM process. Both of these potential causes will be discussed in more detail in the following sections.

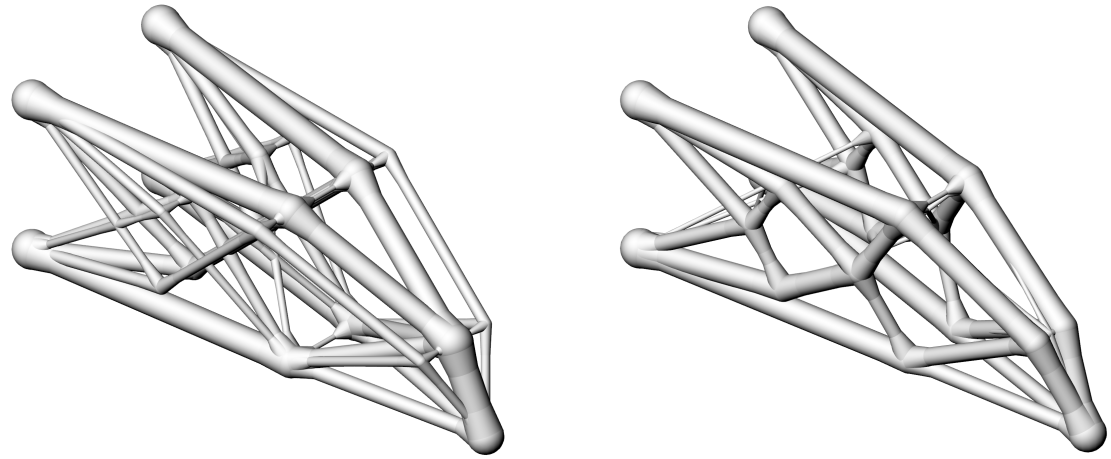
Whilst this was a very preliminary investigation it does highlight the potential value of including geometry optimisation as part of the optimisation work-flow detailed herein. Being already non-linear, the geometry optimisation formulation could also be extended to include bending moments which could potentially eliminate many of the very slender members. Because of the pin jointed assumption some of these very slender members are only present to satisfy the static equilibrium constraint. But the reality is that the rigid joints would have some bending capacity that would possibly obviate the need for some of these slender members.

Table 8.1: Relative volumes for the original cantilever 2 specimen (Chapter 5), the modified layout yielded from the work flow detailed in Chapter 6 and the geometry optimised layout. $V_\infty = 41.99$ cm^3

Specimen	No. of Members	Relative Volume V/V_∞		
		Optimised	Resized	Solid Model
Layout Optimised (original)	64	1.0542	1.2439	1.1485
Layout Optimised (modified)	83	1.0561	1.2199	1.2014
Geometry Optimised	56	1.0235	1.0508	1.0450



(a) (b)



(c) (d)

Figure 8.1: Cantilever 2, (a) original layout, (b) modified layout with diverging members merged, (c) resulting solid model, (d) corresponding model obtained after geometry optimisation

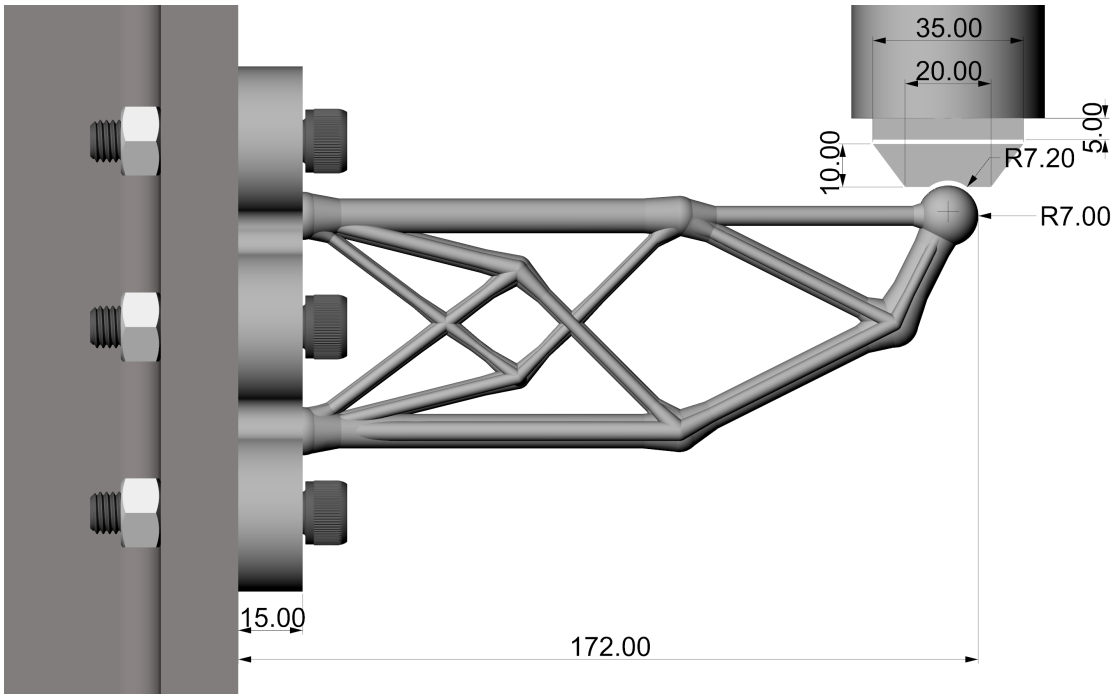


Figure 8.2: Load test arrangement. All dimensions are in mm

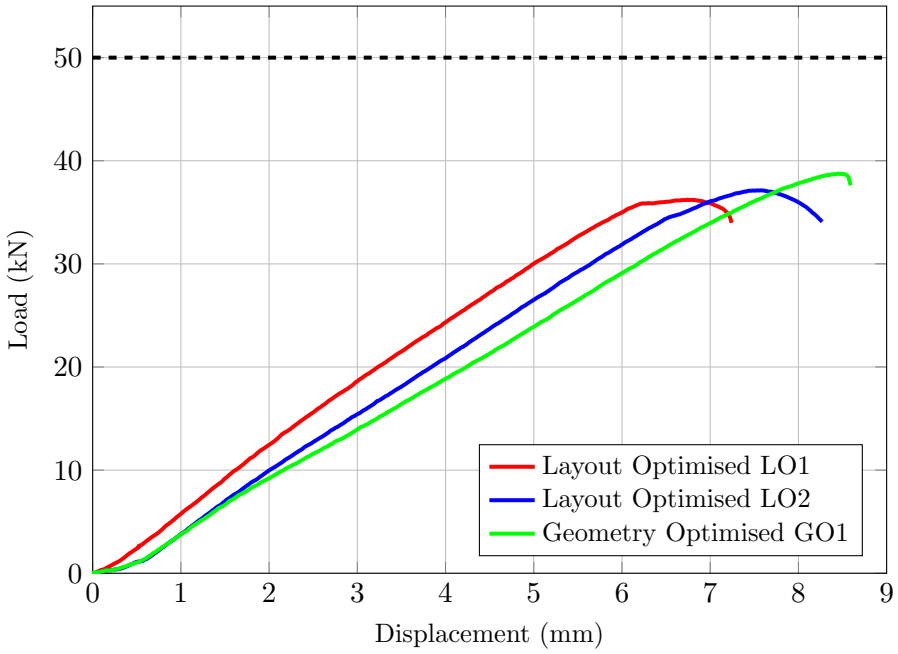
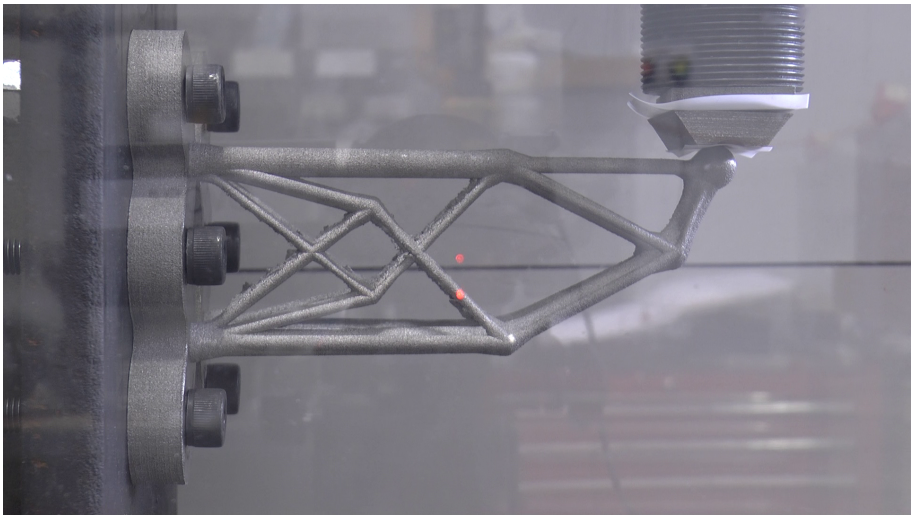
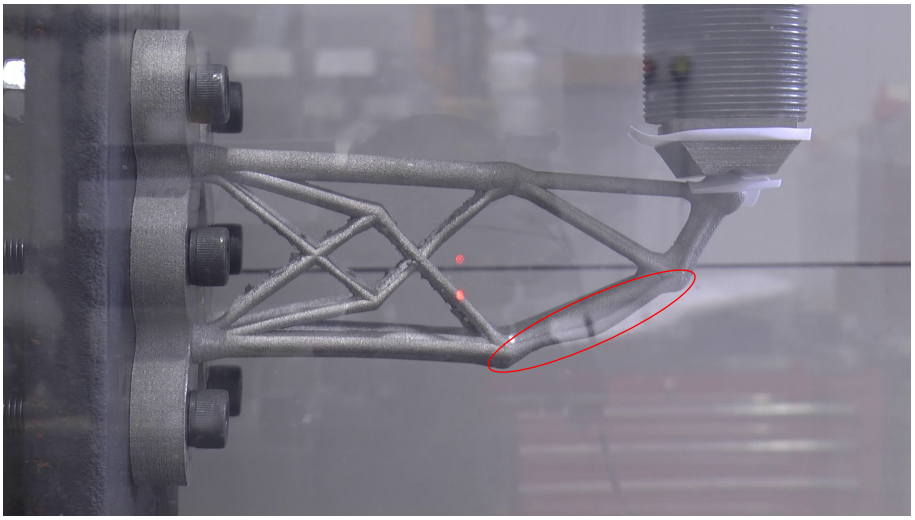


Figure 8.3: Load-Displacement curves for all three specimens load tested



(a)



(b)

Figure 8.4: Photographs taken (a) the start and (b) the end of the load test. The cause of failure is a member that has buckled (circled in red)

8.2 Joint stability and buckling

For all specimens a simple Euler buckling analysis was performed after the optimisation to identify and resize members that would buckle before a tensile or compressive failure. But this analysis assumed an ideal case where the joints are either fixed or pinned and are stable whereas in reality the joints themselves are likely to become unstable if they are not sufficiently braced. Although the stability formulation aims to promote the presence of bracing members or the formation of a layout that is intrinsically stable, it is possible that the magnitude of the disturbing forces for the cantilever case at least (2% of the compressive load acting on the joint), were not large enough.

Because layout optimisation does not account for non-linear changes in geometry due to deformation, the internal forces will deviate from those predicted when the truss structure is subject to large deformations, leading to significant bending moments at the joints. The pin jointed assumption will become less valid with increasing deformation of the truss structure. For the MBB cases the vertical displacement at the loading point was 3 mm at the ultimate load whereas the equivalent value for the cantilever specimens was two to three times larger. Therefore frame instability was potentially a more significant issue for the cantilever case.

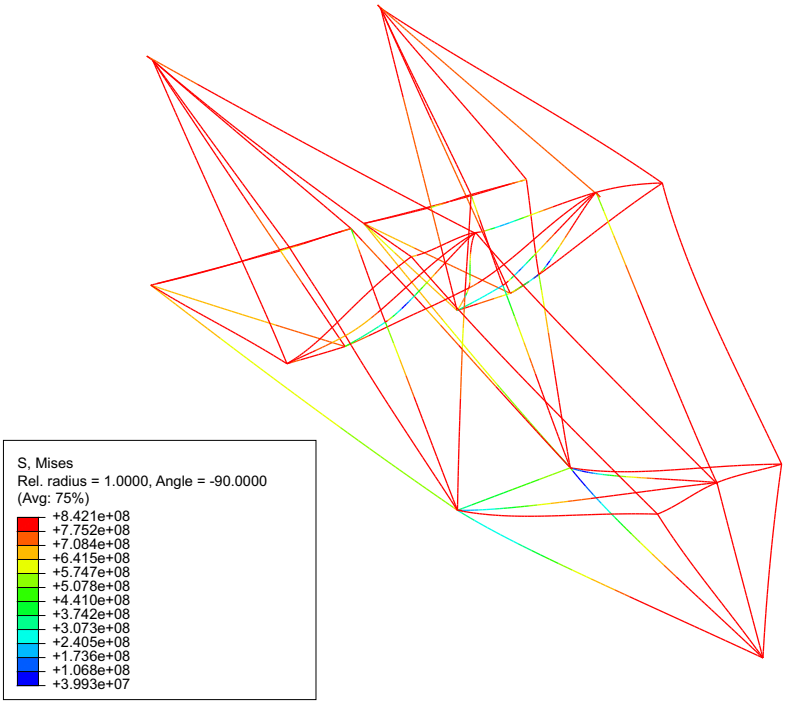
8.2.1 Finite element analysis

To gain a better insight into the effect of bending moments and non-linear changes in geometry a finite element analysis for both layouts in the previous chapter was performed. This analysis made use of two node beam elements of length 1 mm and an elasto-plastic material model. Two types of analysis were performed; a standard static analysis and a RIKS analysis, which is typically employed for problems where there is likely to be buckling or complete collapse of the structure. The latter takes into account non-linear changes in geometry are taken into account by updating the stiffness matrix based on the displacement of the nodes.

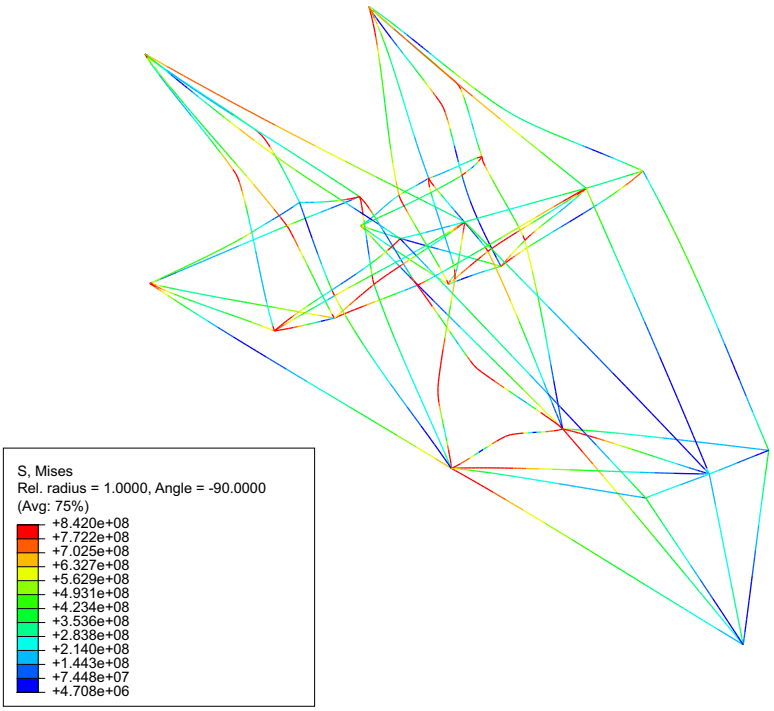
It can be seen in Figs. 8.6a & 8.7a that the static analysis predicts a failure load close to that of the target, with the slight discrepancy likely due to the inclusion of shear forces at the joints. The RIKS analyses (Figs. 8.6b & 8.7b) however predicted much lower failure loads, which for the geometry optimised layout (39.0 kN) was very close to that of the load test result (38.8 kN). These analyses should only be viewed as a preliminary investigation as further experimental testing and finite element analyses would be required to assess the validity of the modelling results. However, it does highlight the need for either larger disturbing forces in the stability formulation or more rigorous frame stability considerations in the optimisation formulation.

8.2.2 Use of cross-sections more resilient to buckling

All the case studies explored in the preceding chapters made use of members with a solid circular cross-section for the sake of simplicity. But in terms of buckling resilience this type of cross-section is very inefficient. All truss layouts in the preceding chapters required the cross-sectional area of several or more members to be increased as to avoid premature failure through buckling, which of course increased the overall mass of these structures. The problem with solid circular cross-sections is that they have a relatively low second moment of area about the principal axes, that is the material is not distributed very far



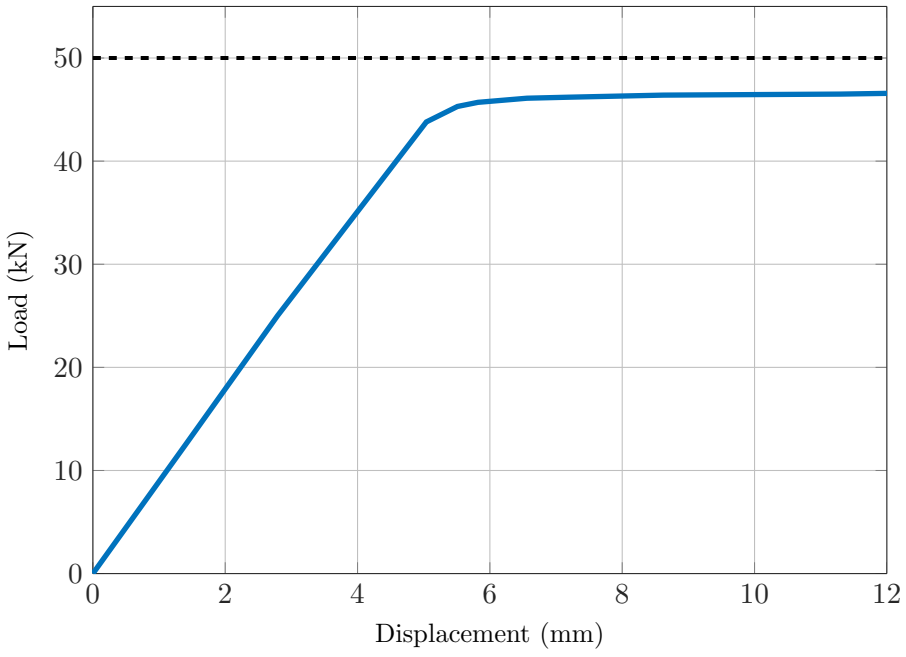
(a) Static analysis



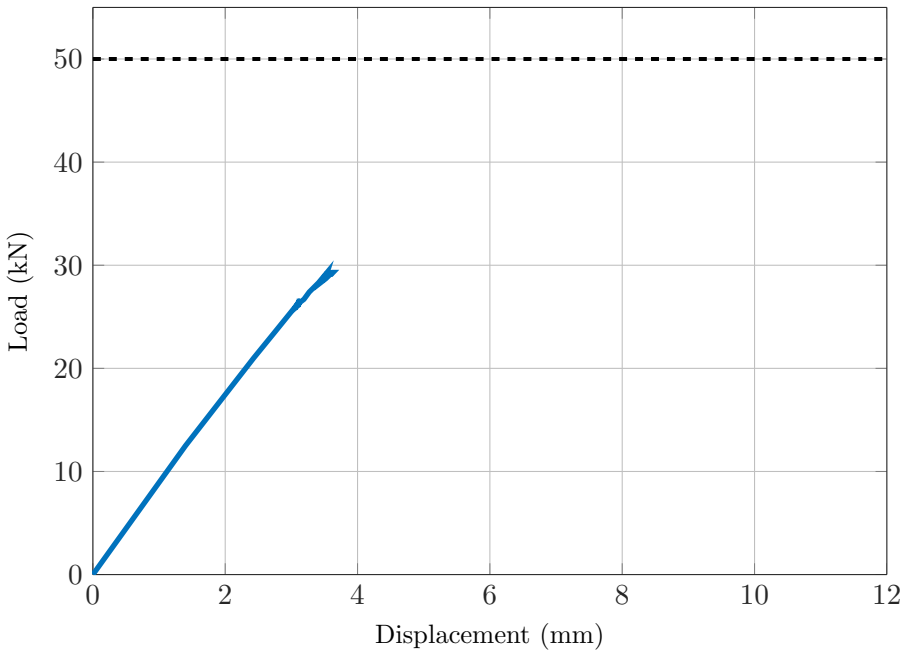
(b) RIKS analysis

Figure 8.5: FEA results for the layout optimised specimen showing the deformed structure. Von Mises stresses are stated in Pa.

from the centroid. Typically shapes that have a larger distribution of material from the



(a) Static analysis

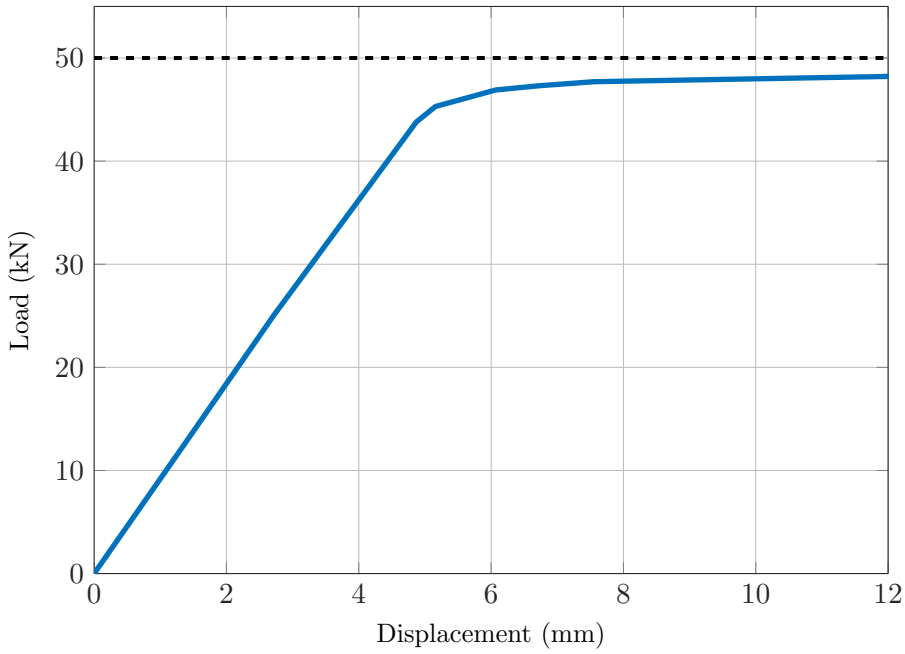


(b) RIKS analysis

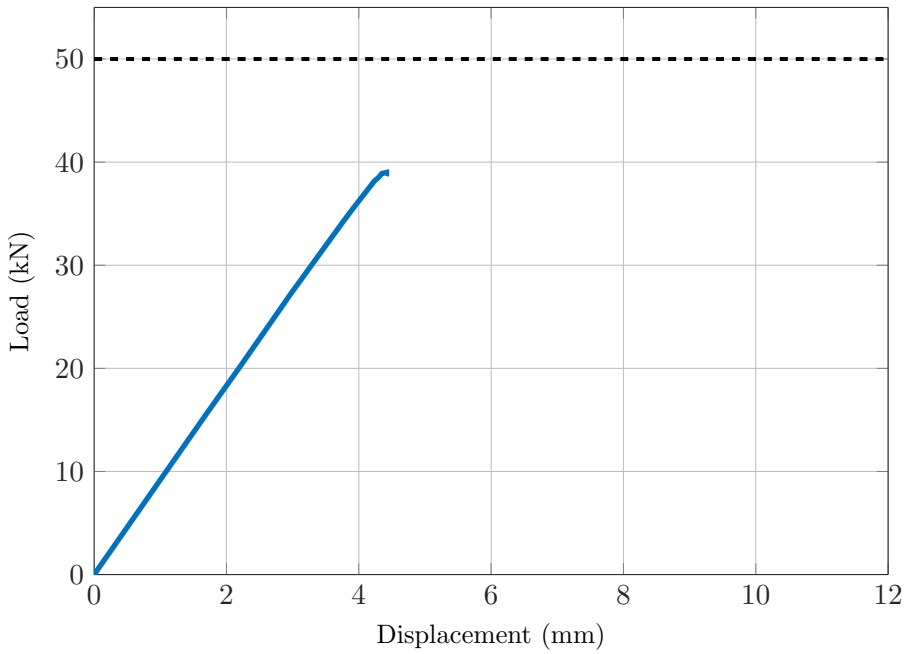
Figure 8.6: Load displacement curves for both types of analysis for the layout optimised specimens

centre (such a I-beams) or hollow members are used for beam like structures subject to large bending and/or axial compressive loads. Incorporating such members into the layout optimisation method proposed thus far is a relatively simple extension.

Although inefficient at resisting buckling, circular members are advantageous for compatibility at the joints. And as buckling instability is greatest at the centre of the member, a sensible solution would be to have members that are circular at the ends and transition to a more buckling resilient cross-section towards the centre. Whilst hollow sections are very



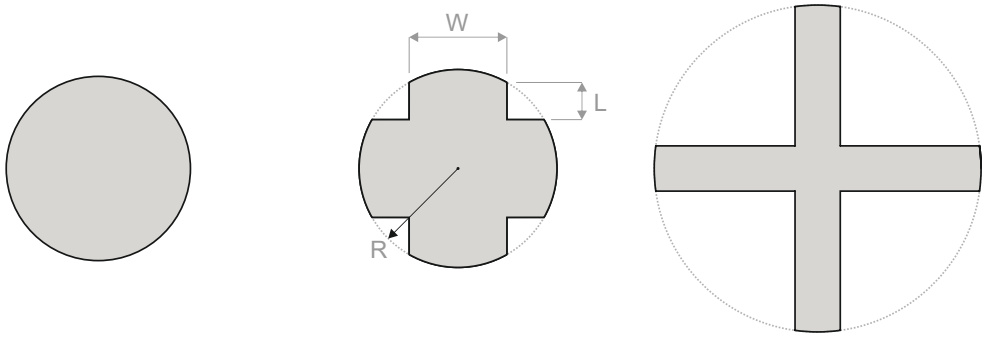
(a) Static analysis



(b) RIKS analysis

Figure 8.7: Load displacement curves for both types of analysis for the geometry optimised specimens

efficient for buckling resilience they would, if additively manufactured, likely present issues with powder removal post-build. Instead a simple but effective cross-section would be that in the shape of a cross, which would transform from a circle at the member ends and then fully open out into a cross in the centre as illustrated in Fig. 8.8. A 3-D rendering of this is shown in Fig. 8.9 and the implementation of these members to the MBB beam case from Chapter 6 is shown in Fig. 8.10.



(a) S_1 (b) S_2 (c) S_3

Figure 8.8: Transition of cross-section from (a) the start of the element through to (c) the mid-point. Section tags correspond to those in Fig. 8.9

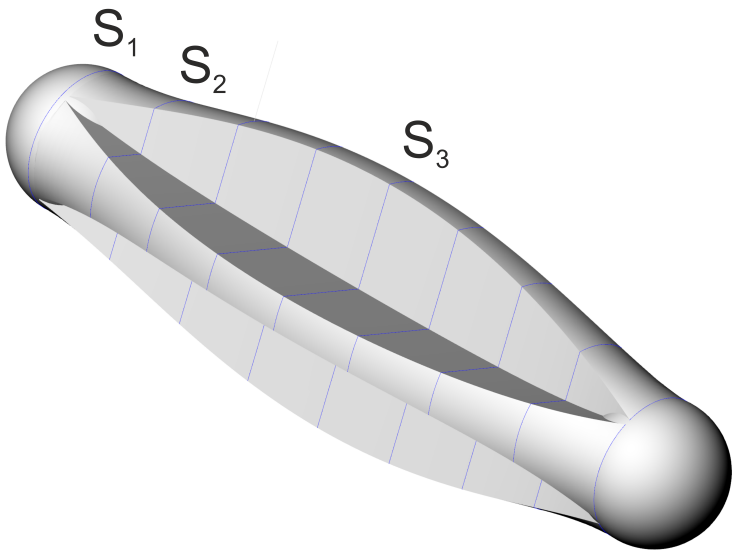


Figure 8.9: Overlay of hexagonal shell elements (shaded) on six truss elements at equal angular spacings

The equation for determining the width W of the arms of this cross-section for a given arm length L and member area A is calculated by numerically solving Eqn. 8.2. The length L is calculated using Eqn. 8.3 which varies the length from zero at the ends of element ($x* = 0$ & 1) to a maximum value (L_{max}) at the member centre ($x^* = 0.5$), based on a profile function $P(x^*)$ which itself varies between zero and one. Determining the optimum profile function of x^*, which is the position along the element as a fraction of the element length, is an area of future research. The members in Figs. 8.9 & 8.10 made use of the parabolic profile function in Eqn. 8.4. L_{max} can be set manually or as with the members in Figs. 8.9 & 8.10 set so that the arm width W at L_{max} is not lower that the minimum feature size, which has been 1.5mm for all case studies in the preceding chapters.

For the member shown in Fig. 8.9 the second moment of area is 555.0mm^4. For a member with circular cross-section of the same area (radius 3.87 mm) the second moment of area is 175.3mm^4, only 32% of the former.

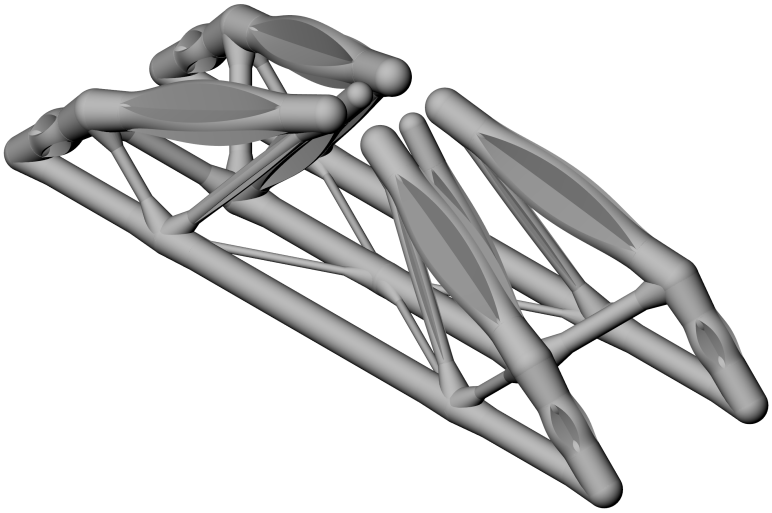


Figure 8.10: The MBB solution with modified compressive members

$$4tan^{-1}\left(\frac{W}{2L+W}\right)\left(\frac{W^2}{2}+WL+L^2\right)+2WL-A=0 \tag{8.2}$$

$$L = L_{max}P(x^*) \tag{8.3}$$

$$P(x^*) = 1-2(x^*)^2 \tag{8.4}$$

8.3 Inclusion of shell elements

For 3-D problems a topology comprising of both truss and shell elements may be more optimal than one with trusses only. Determining a suitable stress analysis for shell elements that can be incorporated into a layout optimisation problem would be challenging. A method proposed by Pavlović (1984a) makes use of a statically determinant truss structure to derive the stress resultants and deformation characteristics of membrane shells. Membrane shell theory considers in-plane forces only and is thus at most only applicable for very thin shells. Pavlović (1984a) surmises that a shell element placed at the node of a truss structure will exert the same virtual work as the axially loaded truss elements it is replacing. From the truss element forces acting on a node the stress resultants of the shell (which are constant) can be determined.

An example is shown in Fig. 8.11 for the hexagonal mesh arrangement with the stress results N_x, N_y and N_{xy} calculated in eqn. 8.3 for the truss member forces (T_1, T_2...T_6) and the shell area a. An example of this mesh pattern is shown in Fig. 8.12, which was then applied to the two cases shown in Fig. 8.13.

Of course the applicability of membrane shells are limited to cases where bending moments can be neglected. In Pavlović (1984b) the method was extended to include bending moments using a similar theory. The method uses a two coincident surfaces to represent the mid-plane of the shell; one that accounts for the in-plane forces only (membrane theory

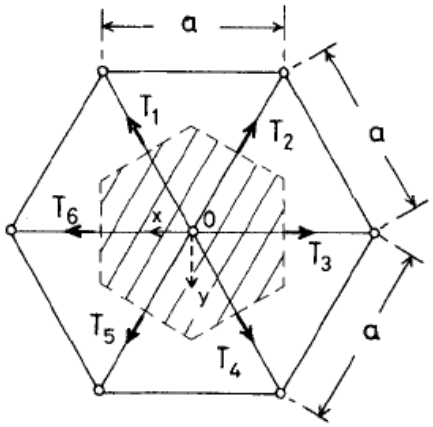


Figure 8.11: Overlay of hexagonal shell elements (shaded) on six truss elements at equal angular spacings (Pavlović, 1984a)

as before) and a second that accounts for bending and twisting moments as well as out of plane shear forces but no in-plane forces. This decomposition approach is of course more complicated and would be harder to implement. How valid either of these approaches are for a layout optimisation problems is a possible area of future research.

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \frac{1}{a} \begin{bmatrix} \frac{1}{4\sqrt{3}} & \frac{1}{4\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{4\sqrt{3}} & \frac{1}{4\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{4\sqrt{3}} & \frac{3}{4\sqrt{3}} & 0 & \frac{3}{4\sqrt{3}} & \frac{3}{4\sqrt{3}} & 0 \\ -\frac{1}{4} & \frac{1}{4} & 0 & -\frac{1}{4} & \frac{1}{4} & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} \tag{8.5}$$

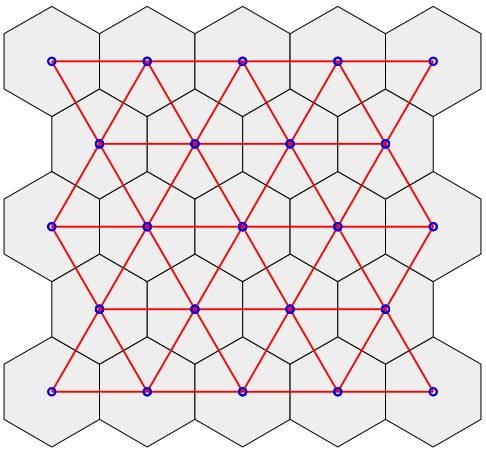
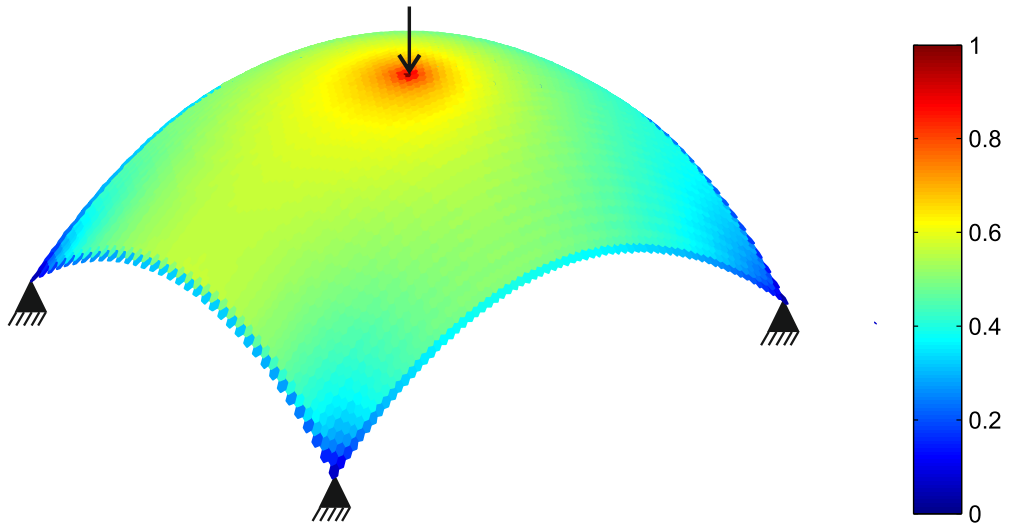


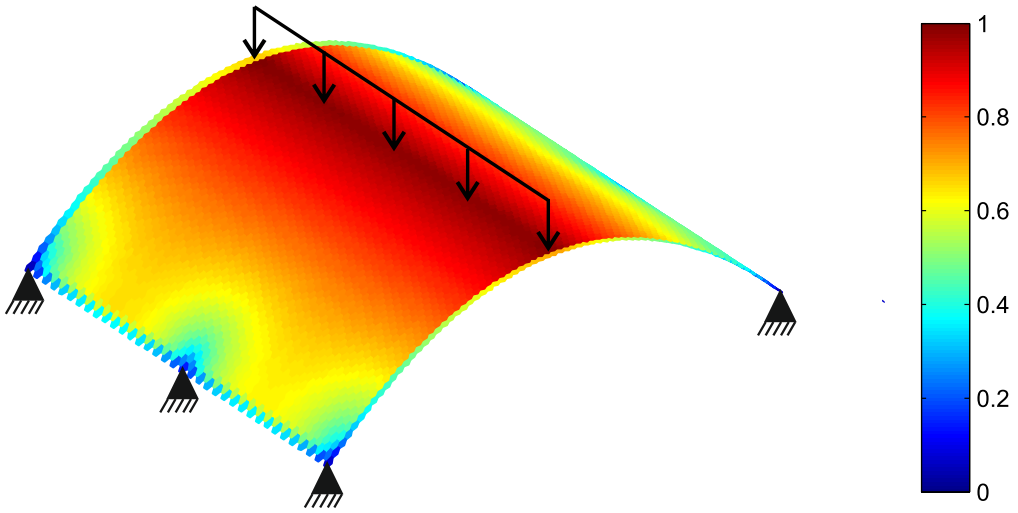
Figure 8.12: Shell overlay scheme used with truss elements shown in red

8.3.1 Refining solid model generation

Although the rudimentary joint expansion route (detailed in Chapter 4) was shown through FEA to reduce the plastic strain at the joints in Chapter 6, it did not completely alleviate



(a)



(b)

Figure 8.13: (a) Spherical dome with central point load and four fully constrained support points and (b) Cylindrical dome with central line load along the length and six fully constrained support points. Both examples used 14735 truss elements represented by 5006 shell elements

the problem. And it is also likely that in some instances there was too much additional material placed at the joints. A more refined method could be the use of a true volume conserving algorithm like that proposed by Decaudin et al. (1994), which is illustrated in Fig. 8.14.

The concept of this method is that any overlapping area is projected from a central point H to the boundary of the original shapes, creating a new shape that is a fusion of the two. The equation for the 2-D case is shown in Eqn. 8.6 and has been evaluated for the two circles shown in Fig. 8.14. This equation is evaluated at a range of discrete angles and defines the vertices of the polygon. The position of the projection point is important as this will determine the position of the centroid of merged shape. Ideally this should be

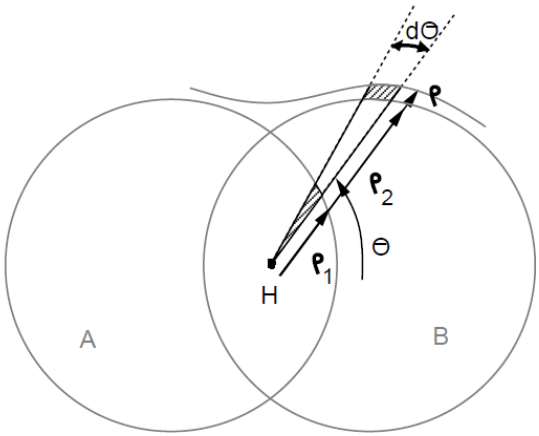


Figure 8.14: Concept of area conservation for two intersecting circles

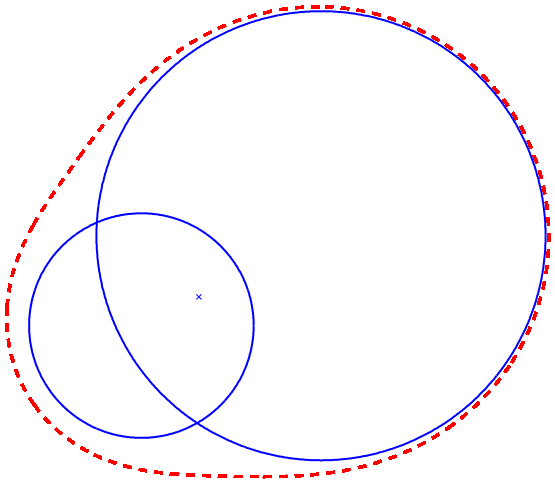


Figure 8.15: Concept of area conservation for two intersecting circles

the same as the average of the centroids of the cross-sections being merged. Additional shapes are merged sequentially. i.e. Eqn. 8.6 would be evaluated for the merged shape and the next one.

The method is easily extended to 3-D using Eqn. 8.7, which in its simplest implementation could be used to define the vertices of a polygonal mesh. A more sophisticated approach could use it to define the control points of a NURBS defined surface. But as this paper is from 1994 it is more than likely that more sophisticated methods have been developed since, though the basic concept would be the same.

$$\rho(\theta) = \sqrt{\rho_1(\theta)^2 + \rho_2(\theta)^2} \tag{8.6}$$

$$\rho(\theta, \phi) = \sqrt[3]{\rho_1(\theta, \phi)^3 + \rho_2(\theta, \phi)^3} \tag{8.7}$$

Use of shape optimisation and isogeometric methods

Perhaps a more rigorous way to address this problem from a structural perspective is to apply one of the shape optimisation methods discussed in Chapter 2. It is documented in the literature that topology optimisation methods such as SIMP are usually followed by a shape optimisation step to refine the final output structure. But before this is possible the output must first be converted into a smooth and well defined surface, which can be a laborious process. Because the output from the work-flow described in Chapter 6 is a NURBS model with well defined surfaces, transferring to a shape optimisation should be relatively straightforward process. Whether this be a FE based or isogeometric shape optimisation would be an area of further research.

8.4 Accuracy, defects and surface finish of EBM

A big issue with the fabrication of truss structures with the EBM process is that many members will be undersized, reducing overall load capacity. This may have been at least partly responsible for the frame instability seen with the cantilever specimens. The source of this issue was identified in Chapter 7 as excessive beam energy input at regions in close proximity to a negative surface. Negative surfaces are problematic for most AM processes so it is likely this issue may not be unique to the EBM process. Significant improvements in dimensional accuracy were achieved when reducing the energy input through using a hatch only melt with a faster beam velocity. But this came at a cost of increased internal porosity in the regions where these modified parameters were active.

Internal porosity is generally present with in parts made with all metallic powder bed processes and it can be said there are two schools of thought with regard to dealing with it. The first and most straight forward method is to accept porosity will be present and, provided the pores are relatively small, post-process the parts using Hot Isostatic Pressing (HIP). HIP routines have been shown in the literature to significantly increase fatigue life of specimens fabricated using EBM. The second is to modify the parameters of the EBM process to reduce and perhaps eliminate porosity from EBM parts. It has been found that the use of a higher beam energy density results in a lower density of pores. But this is of course contrary to the requirement for improved dimensional accuracy. But it should also be noted that other parameters, such as a higher beam focus, which would not increase the energy density, have been shown to also result in a reduction in pore density. Determining a compromise between low internal porosity and high dimensional accuracy on negative surfaces will of course be an area of further research.

The as-built surface finish of the EBM process has shown to reduce the tensile strength in both the tests performed herein and in those covered in the literature. All layout optimised design specimens were tested in their as-built state as surface finishing such complex geometries would prove very challenging. And this highlights a possible limitation with EBM and many other additive manufacturing processes. If a high surface finish quality cannot be produced, either from the process or from a post-process surface treatment, then the application of these parts will be limited. Whilst a reduction in static strength properties is acceptable and can be accounted for, the same cannot be said for the fatigue properties which have been shown in the literature to be greatly influenced by the as-built surface quality. Surface treatments such as vibratory polishing, electro-polishing and abrasive flow machining (abrasive particles suspended by a liquid catalyst) have the potential to improve the surface quality of complex truss structures produced using EBM.

An area of future work would be to assess the measured improvement in surface roughness and strength properties for single truss members and entire truss structures using these methods.

Chapter 9

Conclusions

- A work-flow based around layout optimisation has been created which can solve large 3-D problems with multiple load cases and result in a solid model defined by NURBS or an STL mesh ready for additive manufacture. The process of going from an optimisation result to a finalised CAD model can be seen as far less laborious than the prevalent mesh based approaches, involving methods such as SIMP and ESO/BESO.
- Structural performance of specimens designed using this work-flow and manufactured using the EBM process was validated through physical load testing. The MBB beam specimens successfully reached their target load upon physical testing.
- The potential of the layout optimisation work-flow for industrial applications was demonstrated with the air-brake hinge case, where a mass reduction of 69% was achieved over the conventionally designed original part.
- Frame stability is likely to be an issue with the current formulation for structures likely to experience significant deformation, such as seen with the cantilever case. The stability formulation and choice of member cross-section used herein will need be reviewed for such cases.
- The work-flow can be further developed to include additional steps. Of these discussed, geometry optimisation is particularly promising as a subsequent step to layout optimisation as it has the potential to reduce the complexity of the layout optimisation result and reduce the total volume in the process. This may obviate the need to manually specify the nodal distribution, which is used as an input for layout optimisation.
- A scaling factor of 1.05 was applied to the cross-sectional areas of all truss specimens fabricated to account for an apparent dimensional issue with the EBM process. Upon quantifying these dimensional errors with 3D scanning techniques it was found that the error is not consistent, indicating that a global scaling factor is not appropriate. Dimensional errors may have contributed to the frame stability issues observed with the cantilever case.
- The source of these dimensional errors was identified and suitable modifications were made to the process parameters that not only reduced these errors significantly but also greatly reduced the need for support structures. Samples were built accurately at 60° to the vertical without support structures, suggesting that limitations on part

orientation could be reduced or eliminated. Further research on developing these process parameters should be an area of further research.

- X-ray Computed Tomography (XCT) identified spherical gas pores in the region of (150μm) which would present an issue for fatigue life but should not have had a significant effect on static strength properties.
- Larger tunnel like defects with diameters of $\sim$300 μm and lengths of up to 10 mm were detected in one specimen but it is highly likely that these were anomalies generated in the image reconstruction process. No such defects were observed at the fracture surfaces.
- Whilst the as-built surface finish and internal porosity did not present an issue here they will when fatigue life is of importance. How these two issues are addressed, be it through post-processing steps such as Hot Isostatic Pressing (HIP) and surface treatments, or via improvements to the EBM process itself, will be an important area of research.

Chapter 10

Future Work

The following have been identified as areas of further work and are summarised from Chapter 8:

- Investigate if the use of members with varying cross-section with length can both reduce overall volume and increase frame stability. A relationship relating the dimensions of these members to buckling strength will need to be derived.
- Assess the use of higher disturbing forces in the stability analysis to improve both local and global buckling stability. This could be performed initially using a FEA RIKS analysis and then later validated with physical load testing. The incorporation of members with a varying cross-section should also be included in these analyses.
- Investigate the use of a more rigorous joint expansion algorithm (in the solid model generation step), considering both total volume and the stress distribution at the joints. The use of a volume conservation algorithm similar to that detailed in Chapter 8 or the use of shape optimisation should be explored.
- Assess the effectiveness of geometry optimisation for the MBB beam and, in particular, the air-brake hinge problem. The latter problem required considerable manual effort and iteration to determine a suitable spatial distribution of nodes. Geometry optimisation could potentially provide an alternative and more rigorous approach.
- Further characterize the mechanical properties, internal porosity and dimensional accuracy for truss members at a range of angles and diameters fabricated using the modified process parameters detailed in Chapter 7.
- The cantilever and MBB truss layouts should be fabricated again so that the 3D scanning analysis detailed in Chapter 7 can be performed to determine dimensional errors for each member. The results from such analyses would provide insight as to what degree the premature failures (for the cantilever case) could be attributed to dimensional errors.
- These layouts could then be manufactured using the modified process parameters so that they too can be assessed using the 3-D scanning analysis and then load tested.
- Explore the use of incorporating shell elements into the layout optimisation formulation using the truss analogies described in Chapter 8.

Appendix

Listing 10.1: Problem setup for C++ based optimiser

```
% COPT FILE GENERATOR

% --------------------- INPUTS -------------------------
obj_type = 5 ; % 1 - Abaqus Gen, 2 - Meshlab gen, 3 -
   Grasshopper Gen, 4 - read from node_list, 5 - Use existing
    frame file

dup_node_planes = 0 ; % Duplicates planes of nodes in y
   direction 0 - no, 1 - yes
domain_width = 25 ;
no_node_planes = 3 ; % node planes imported from Abaqus
FoS = 1 ; % Factor of Safety
node_input = 1 ; % 0 - generate nodes in COPT, 1 - Use nodes
   from existing Frame file, 2 - Generate nodes in Matlab
   defined in input file. 3 - For BloodHound using abaqus
   mesh
shell_or_point = 1 ; % 1 - Point Loads or 2 - Shell loads
active_load_cases = 1 ; % Load Cases to use
forces = [1 0 0 50] ; % Forces - [ LoadCase F1 F2 F3 ]
Yc = 0.933 ; % Compressive Yield (GPa)
Yt = 0.933 ; % Tensile Yield (GPa)
surf_nodes = 0 ; %surface nodes
max_l = 1000 ; % Maximum potential member length (mm)
node_s = 1000 ; % Nodal Spacing (mm)
off_surf_nodes = 0 ; % offset surface nodes
stability = 0 ; % Stability on-off (1-0)
min_area = 1.7671455 ; % Minimum truss area permitted in post
   -processing
joint_cost = 0 ; % Joint costs
y_symmetry = 0 ; % Symmetry on or off

output = 'Can2members' ; % Name of COPT file to be generated
directory = pwd ;
domain = 'Domain.obj' ;
support = 'supports.obj' ;
shell_file = 'Shell1' ; % exclude extension .obj
solid_files = {[shell_file '.obj'],'0.00000001','my_mat','1'}
    ; % solid files [ Filename  Thickness  Material  Rigid
```

```
   :(1-0) ]

% ------------- GROUND STRUCTURE IMPORT ----------------

if obj_type < 2 ;

    fid = fopen('shell_lower.obj') ;
    delimiter = ' ' ;
    loop = 1 ;

    switch obj_type
        case 1
            E = textscan(fid,'%s %f %f %f',1,'HeaderLines',2)
                ;
        case 2
            V = textscan(fid,'%s %f %f %f',1,'HeaderLines'
               ,11) ;
            E = textscan(fid,'%s %f %f %f %f %f %f',1) ;
    end

    node_list(1,1:3) = [E{2} E{3} E{4}] ;
    tolerance = 2 ;

    while 1
        switch obj_type
            case 1
                E = textscan(fid,'%s %f %f %f',1) ;
            case 2
                V = textscan(fid,'%s %f %f %f',1) ;
                E = textscan(fid,'%s %f %f %f %f %f %f',1) ;
        end

        if sum([isempty(E{2}) isempty(E{3}) isempty(E{4})]) >
            0
            break
        else
            node_list(loop+1,1:3) = [E{2} E{3} E{4}] ;
        end
        loop = loop + 1 ;
    end

    fclose(fid) ;
    node_list = sortrows(node_list,1) ;
    node_list = nodesort(node_list, tolerance) ;
end

if obj_type == 3
    node_list(:,2:4) = importdata('GHOutput.csv',',') ;
    node_list(:,1) = 1:1:size(node_list,1) ;
end
```

```
77
78  if obj_type == 4
79      node_list_imp = importdata('C:\Users\Owner\Documents\
            Advanced Metallic Systems DTC\Personal Work\Models\
            Pure Matlab Codes\Frame Post Processor\frame_nodes.txt
            ',' ') ;
80      node_list(:,2:4) = node_list_imp(:,2:4) ;
81      node_list(:,1) = 1:1:size(node_list,1) ;
82  end
83
84  if obj_type == 5
85      FrameFile = 'Examples\Can2.txt' ;
86      delimiter = ' ' ;
87      [elem_list, node_list, pre_vol, nodeListForces,
            elem_forces, elem_comp_forces] = FrameImpv2a(FrameFile
            ,delimiter,1) ; % Importing the data from the previous
             optimisation
88  end
89
90  forces(:,2:4) = FoS * forces(:,2:4) ;
91  forces = forces(active_load_cases,:) ;
92
93  switch shell_or_point
94      case 1
95          load_point = [150 0 0] ; % X Y Z of Force
96          % Point loads - [ LoadCase x y z F1 F2 F3 ]
97          force_point(:,1) = forces(:,1) ;
98          force_point(:,2) = load_point(1) ;
99          force_point(:,3) = load_point(2) ;
100         force_point(:,4) = load_point(3) ;
101         force_point(:,5:7) = forces(:,2:4) ;
102         force_surf = 0 ;
103     case 2
104         force_surf = {} ;
105         force_point = [] ;
106         for k = 1 : size(forces,1)
107             force_ind = forces(k,1) ;
108             force_2 = forces(k,3) ;
109             force_3 = forces(k,4) ;
110             % Force on surface - [ LoadCase Filename
                    Magnitude minArea ]
111             force_surf = [ force_surf
112                            {num2str(k),[shell_file 'F2.obj'],
                                num2str(force_2),'0.001'}
113                            {num2str(k),[shell_file 'F3.obj'],
                                num2str(force_3),'0.001' }] ;
114         end
115 end
116
117 % ------- Mesh Refinement and Coordinate mapping--------
```

```
118
119 if dup_node_planes == 1 ;
120     max_2 = max(node_list(:,2)) - min(node_list(:,2)) ;
121     max_3 = max(node_list(:,3)) - min(node_list(:,3)) ;
122     max_4 = max(node_list(:,4)) - min(node_list(:,4)) ;
123
124     [X,I(1)] = max([max_2 max_3 max_4]) ; % Will be the x
        coordinates
125     [Y,I(2)] = min([max_2 max_3 max_4]) ; % Will be y
126     z_index = ((I(1) ~= [1 2 3]) & (I(2) ~= [1 2 3])) * [1 2
        3]' + 1 ;
127     new_node_list(:,1) = 1:1:size(node_list,1) ;
128     new_node_list(:,2) = node_list(:,I(1) + 1) ;
129     new_node_list(1:size(node_list,1),3) = 0 ; % Set this to
        zero as values will be essentially zero
130     new_node_list(:,4) = node_list(:,z_index) ;
131
132     node_list = [] ;
133     no_nodes = size(new_node_list,1) ;
134
135     y_offset = linspace(-0.5*domain_width, 0.5*domain_width,
        no_node_planes) ;
136
137     for k = 1:no_node_planes
138         node_list((k - 1) * no_nodes + 1 : k * no_nodes, 1) =
              k * no_nodes + 1 : (k + 1) * no_nodes ;
139         node_list((k - 1) * no_nodes + 1 : k * no_nodes, 2) =
              new_node_list(1:no_nodes,2) ;
140         node_list((k - 1) * no_nodes + 1 : k * no_nodes, 3) =
              y_offset(k) ;
141         node_list((k - 1) * no_nodes + 1 : k * no_nodes, 4) =
              new_node_list(1:no_nodes,4) ;
142
143     end
144 end
145
146 % ----------------------FILE GEN ------------------------
147
148 if node_input > 0
149     node_s = 10000 ;
150 end
151
152 fid = fopen([directory '\' output '.copt'],'w') ;
153
154 fprintf(fid,'%s\n','<!DOCTYPE COPTML>') ;
155 fprintf(fid,'%s\n','<project') ;
156 fprintf(fid,'%s%f%s\n','        compressive_strength="',Yc,'"
    ') ;
157 fprintf(fid,'%s%f%s\n','        tensile_strength="',Yt,'"') ;
```

```
158 fprintf(fid,'%s%i%s\n','         surface_nodes="',surf_nodes,'
    "') ;
159 fprintf(fid,'%s%f%s\n','         max_potential_member_length="
    ',max_l,'"') ;
160 fprintf(fid,'%s%f%s\n','         nodal_spacing="',node_s,'"')
    ;
161 fprintf(fid,'%s%i%s\n','         offset_surface_nodes="',
    off_surf_nodes,'"') ;
162 fprintf(fid,'%s%i%s\n','         stability="',stability,'"') ;
163 fprintf(fid,'%s%f%s\n','         minimum_member_area="',
    min_area,'"') ;
164 fprintf(fid,'%s%i%s\n','         joint_cost="',joint_cost,'"')
     ;
165 fprintf(fid,'%s%i%s\n','         symmetry_y="',y_symmetry,'"')
     ;
166 fprintf(fid,'%s\n','         move_loaded_nodes_to_surface="0"'
    ) ;
167 fprintf(fid,'%s\n','         use_nodes_from_file="1"') ;
168 fprintf(fid,'%s\n','>') ;
169 fprintf(fid,'%s%s%s\n','  <design_space file="',domain,'"/>')
     ;
170 fprintf(fid,'%s%s%s\n','  <support_surface file="',support,'
    "/>') ;
171
172 if shell_or_point == 2
173     for k = 1:size(solid_files,1)
174         fprintf(fid,'%s%s%s%s%s%s%s%s%s\n','  <solids file="'
            ,solid_files{k,1},'" t="',solid_files{k,2},'" mat
            ="',solid_files{k,3},'" is_rigid="',solid_files{k
            ,4},'"/>') ;
175     end
176 end
177
178 if shell_or_point == 2
179     % Write the shell loads to COPT file
180     for k = 1:size(force_surf,1)
181         fprintf(fid,'%s%s%s%s%s%s%s%s%s\n','  <load type="
            force_on_surface" load_case="',force_surf{k,1},'"
            solid_file="',force_surf{k,2},'" magnitude="',
            force_surf{k,3},'" min_area_for_nde="',force_surf{
            k,4},'"/>') ;
182     end
183 end
184
185 % Write the Point loads to COPT file
186
187 if size(force_point,2) > 1
188     for k = 1 : size(force_point,1) ;
189         fprintf(fid,'%s%i%s%f%s%f%s%f%s%f%s%f%s%f%s\n','  <
            load load_case="',force_point(k,1),'" x="',
```

```
            force_point(k,2),'" y="',force_point(k,3),'" z="',
            force_point(k,4),'" lx="',force_point(k,5),'" ly="
            ',force_point(k,6),'" lz="',force_point(k,7),'"/>'
            ) ;
    end
end

% Write potential members to COPT file
if obj_type >= 5
    for k = 1:size(elem_list,1)
        fprintf(fid,'%s%i%s%i%s\n','  <member na="N',
           elem_list(k,2),'" nb="N',elem_list(k,3),'"/>') ;
    end
end

% Write all the nodes to the COPT file

for k = 1:size(node_list,1)
    fprintf(fid,'%s%s%s%f%s%f%s%f%s\n','  <node name="N',
       num2str(k),'" load_case="1" x="',node_list(k,2),'" y="
       ',node_list(k,3),'" z="',node_list(k,4),'"/>') ;
end

fprintf(fid,'%s','project>') ;
fclose(fid) ;
scatter3(node_list(:,2),node_list(:,3),node_list(:,4))
```

Listing 10.2: Initialisation code for post-processing

```
%         ** FRAME IMPORT AND POST PROCESSING CODE **
%                       ** FramePP **


% CODE imports Frame File from COPT and runs post-processing
   modules to
% produce a final, practical design

% Output is a text file for reading into Rhino via a
   Grasshopper network

% ---------------- PROBLEM DEFINITION ------------------

clf, clear, clc
bigNumber = 1e6 ;

prob_example = 5 ; % 1 - New Problem, 2 - MBB examples, 3 -
   Bloodhound

switch prob_example

    case 1 % New Example
```

```
20
21          FrameFile = 'Examples\Sphere.txt' ;
22          delimiter = ' ' ;
23          [elem_list, node_list, pre_vol, nodeListForces,
               elem_forces, elem_comp_forces] = FrameImpv2a(
               FrameFile,delimiter) ; % Importing the data from
               the previous optimisation
24
25          %FEA ABAQUS
26          applied_force = [100 5 0 0 0 -100] ; % [X Y Z Fx Fy
               Fz] of Applied load(s)
27          % applied_force(:,4:6) = applied_force(:,4:6) * 0.8 ;
28          transpose = [] ; % Swap z and y axis for abaqus
               analysis
29          analysis_type = 3 ; % 1 - Elastic Linear Peturbation,
                2 - Static, General & 3 - Static, RIKS
30          NLGEOM = 0 ; % Account for non-linear geometry
               changes
31          sup_type = [1 1 1 0 0 0] ; % [Sx Sy Sz Rx Ry Rz ; Sx
               Sy Sz Rx Ry Rz ; ....] % Row entry for each
               support group.
32          load_type = 'POINT' ; % Surface of point
33          shell_def = [] ; % Surface X-Y coordinate, must be in
                \Surface Definitions folder
34          elem_type = 1 ; % For Abaqus FEA  (1) T3D2 - Truss
               element axial forces only (2) B31 2 node Linear
               beam elements (3) B32 3 node Quadratic Beam
               elements
35          global_seed = 1 ; % Seeding distance (mm) for meshing
                in Abaqus (only applicable for beam elements)
36
37          material_file = 'Ti64AsBuilt.txt' ; % Materials file,
                set at [] if linear elastic. machinedTi64.txt -
               Ti64AsBuilt.txt - Ti64perfectplastic.txt
38          sup_node_ind = (abs(node_list(:,2)) == 0 ) ; % Find
               supported nodes - SPECFIC TO MBB's!!
39          sup_nodes = node_list(sup_node_ind==1,1) ;
40          con_nodes = [] ;
41
42
43      case 2 % MBB example
44
45          FrameFile = 'Examples\
               MBB2_842MPA_correct_axial_forces.csv' ;
46          delimiter = ',' ;
47          %FrameFile = 'Examples\MBB_160mm_point.txt' ;
48          %delimiter = ' ' ;
49          [elem_list, node_list, pre_vol, nodeListForces,
               elem_forces, elem_comp_forces] = FrameImpv2a(
               FrameFile,delimiter,1) ; % Importing the data from
```

```
            the previous optimisation
50          %[elem_list, node_list] = ElementMerger(elem_list,
               node_list) ;
51
52          %FEA ABAQUS
53          applied_force = [0 40 0 0 -100 0] ; % [X Y Z Fx Fy Fz
               ] of Applied load(s)
54          % applied_force(:,4:6) = applied_force(:,4:6) * 0.8 ;
55          transpose = [] ; % Swap z and y axis for abaqus
               analysis
56          analysis_type = 3 ; % 1 - Elastic Linear Peturbation,
               2 - Static, General & 3 - Static, RIKS
57          NLGEOM = 1 ; % Account for non-linear geometry
               changes
58          sup_type = [0 1 1 0 0 0 ; 0 1 1 0 0 0] ; % [Sx Sy Sz
               Rx Ry Rz ; Sx Sy Sz Rx Ry Rz ; ....] % Row entry
               for each support group.
59          load_type = 'SURFACE' ; % Surface of point
60          shell_def = 'MBBShell4.txt' ; % Surface X-Y
               coordinate, must be in \Surface Definitions folder
61          elem_type = 2 ; % For Abaqus FEA  (1) T3D2 - Truss
               element axial forces only (2) B31 2 node Linear
               beam elements (3) B32 3 node Quadratic Beam
               elements
62          global_seed = 1 ; % Seeding distance for meshing (mm)
               in Abaqus (only applicable for beam elements)
63
64          material_file = 'Ti64perfectplastic.txt' ; %
               Materials file, set at [] if linear elastic.
               machinedTi64.txt - Ti64AsBuilt.txt -
               Ti64perfectplastic.txt
65          sup_node_ind1 = (node_list(:,2) > 79 ) + (node_list
               (:,3) < 0.1) ; % Find supported nodes - SPECFIC TO
               MBB's!!
66          sup_node_ind2 = (node_list(:,2) <- 79 ) + (node_list
               (:,3) < 0.1) ;
67          sup_nodes = [node_list(sup_node_ind1==2,1)' ;
               node_list(sup_node_ind2==2,1)'] ;
68
69          con_nodes_ind = (abs(node_list(:,2)) < 7.6) + (abs(
               node_list(:,4)) < 10.1) + (node_list(:,3) > 32) ;
               % Find Shell Connected nodes - SPECFIC TO
               BLOODHOUND PROBLEM!!
70          con_nodes = node_list(con_nodes_ind==3,1) ;
71
72      case 3 % Bloodhound example
73
74          FrameFile = 'C:\Users\Owner\Documents\Advanced
               Metallic Systems DTC\ALM Commerical builds\
               Bloodhound\Optimisations\Node Gen Grasshopper\
```

```
        Iteration 1 - One Load Case\frame.txt' ;
75      FrameFile = 'Examples\BloodhoundCubic340Nodes.txt' ;
76      delimiter = ' ' ;
77      [elem_list, node_list, pre_vol, nodeListForces,
        elem_forces, elem_comp_forces] = FrameImpv2a(
        FrameFile,delimiter) ; % Importing the data from
        the previous optimisation
78
79      %FEA ABAQUS
80      applied_force = [260.713 0 2.30448 -10.9 0 29.4528] ;
         % [X Y Z Fx Fy Fz] of Applied load(s)
81      % applied_force(:,4:6) = applied_force(:,4:6) * 0.8 ;
82      transpose = 'xzy' ; % Swap z and y axis for abaqus
        analysis
83      analysis_type = 2 ; % 1 - Elastic Linear Peturbation,
         2 - Static, General & 3 - Static, RIKS
84      NLGEOM = 0 ; % Account for non-linear geometry
        changes
85      sup_type = [1 1 1 0 0 0] ;  % [Sx Sy Sz Rx Ry Rz ; Sx
         Sy Sz Rx Ry Rz ; ....] % Row entry for each
        support group.
86      load_type = 'SURFACE' ; % Surface of point
87      shell_def = 'BHShell1.txt' ; % Surface X-Y coordinate
        , must be in \Surface Definitions folder
88      elem_type = 1 ; % For Abaqus FEA  (1) T3D2 - Truss
        element axial forces only (2) B31 2 node Linear
        beam elements (3) B32 3 node Quadratic Beam
        elements
89      global_seed = 1 ; % Seeding distance (mm) for meshing
         in Abaqus (only applicable for beam elements)
90
91      material_file = 'Ti64perfectplastic.txt' ; %
        Materials file, set at [] if linear elastic.
        machinedTi64.txt - Ti64AsBuilt.txt -
        Ti64perfectplastic.txt
92      sup_node_ind = (node_list(:,2) < 150 ) + (node_list
        (:,4) < 12.6) ; % Find supported nodes - SPECFIC
        TO BLOODHOUND PROBLEM!!
93      sup_nodes = node_list(sup_node_ind==2,1)' ;
94
95      con_nodes_ind = (node_list(:,2) > 230) + (node_list
        (:,4) < 30) ; % Find Shell Connected nodes -
        SPECFIC TO BLOODHOUND PROBLEM!!
96      con_nodes = node_list(con_nodes_ind==2,1) ;
97
98  case 4 % Cantilever
99
100     FrameFile = 'Examples\can3_members_NOPP.txt' ;
101     delimiter = ' ' ;
102     %FrameFile = 'Examples\MBB_160mm_point.txt' ;
```

```
103         %delimiter = ' ' ;
104
105         [elem_list, node_list, pre_vol, nodeListForces,
               elem_forces, elem_comp_forces] = FrameImpv2a(
               FrameFile,delimiter,1) ; % Importing the data from
                the previous optimisation
106         %[node_list, elem_list, elem_forces, elem_comp_forces
               ] = linweiGeoOpt() ;
107         %FEA ABAQUS
108         applied_force = [150 0 0 0 0 50] ; % [X Y Z Fx Fy Fz]
                of Applied load(s)
109         % applied_force(:,4:6) = applied_force(:,4:6) * 0.8 ;
110         transpose = [] ; % Swap z and y axis for abaqus
               analysis
111         analysis_type = 3 ; % 1 - Elastic Linear Peturbation,
                2 - Static, General & 3 - Static, RIKS
112         NLGEOM = 0 ; % Account for non-linear geometry
               changes
113         sup_type = [1 1 1 1 1 1] ; % [Sx Sy Sz Rx Ry Rz ; Sx
               Sy Sz Rx Ry Rz ; ....] % Row entry for each
               support group.
114         load_type = 'POINT' ; % Surface of point
115         shell_def = [] ; % Surface X-Y coordinate, must be in
                \Surface Definitions folder
116         elem_type = 2 ; % For Abaqus FEA  (1) T3D2 - Truss
               element axial forces only (2) B31 2 node Linear
               beam elements (3) B32 3 node Quadratic Beam
               elements
117         global_seed = 1 ; % Seeding distance for meshing (mm)
                in Abaqus (only applicable for beam elements)
118
119         material_file = 'Ti64AsBuilt.txt' ; % Materials file,
                set at [] if linear elastic. machinedTi64.txt -
               Ti64AsBuilt.txt - Ti64perfectplastic.txt -
               Ti64perfectplastic842.txt
120         sup_node_ind1 = (node_list(:,2) < 0.01 ) ; % Find
               supported nodes - SPECFIC TO MBB's!!
121         sup_nodes = node_list(sup_node_ind1==1,1)' ;
122
123         con_nodes_ind = [] ;
124         con_nodes = [] ;
125
126     case 5 % Geo Optimised input
127
128         forceScale = 50 ;
129         limitStress = 0.6825 ;
130
131         elemFile = 'Examples\Can 2 Geo Optimised 2016\elems.
               txt' ;
```

```
132         nodeFile = 'Examples\Can 2 Geo Optimised 2016\nodes.
               txt' ;
133         forceFile = 'Examples\Can 2 Geo Optimised 2016\
               InternalForces.csv' ;
134         delimiter = ' ' ;
135         elem_list(:,2:4) = importdata(elemFile,delimiter) ;
136         elem_list(:,4) = elem_list(:,4) * forceScale /
               limitStress ;
137         elem_list(:,1) = (1:1:size(elem_list,1))' ;
138         node_list(:,2:4) = importdata(nodeFile,delimiter) ;
139         node_list(:,1) = (1:1:size(node_list,1))' ;
140         elem_forces = importdata(forceFile,',') ; % Delimiter
                different here!
141         [maxForce,maxForceInd] = max(abs(elem_forces),[],2) ;
142         minForce = min(elem_forces,[],2) ;
143         elem_forces = elem_forces( sub2ind( size(elem_forces)
               , (1:size(elem_forces,1))', maxForceInd ) ) *
               forceScale ; % Extracts maximum force (either
               tensile or compressive) for each load case
144         elem_comp_forces = minForce .* (minForce < 0)  *
               forceScale ; % Creates a vector of compressive
               loads for buckling analysis (tensile forces set to
                zero)
145
146         applied_force = [150 0 0 0 0 50] ; % [X Y Z Fx Fy Fz]
                of Applied load(s)
147         % applied_force(:,4:6) = applied_force(:,4:6) * 0.8 ;
148         transpose = [] ; % Swap z and y axis for abaqus
               analysis
149         analysis_type = 3 ; % 1 - Elastic Linear Peturbation,
                2 - Static, General & 3 - Static, RIKS
150         NLGEOM = 1 ; % Account for non-linear geometry
               changes
151         sup_type = [1 1 1 1 1 1] ; % [Sx Sy Sz Rx Ry Rz ; Sx
               Sy Sz Rx Ry Rz ; ....] % Row entry for each
               support group.
152         load_type = 'POINT' ; % Surface of point
153         shell_def = [] ; % Surface X-Y coordinate, must be in
                \Surface Definitions folder
154         elem_type = 2 ; % For Abaqus FEA  (1) T3D2 - Truss
               element axial forces only (2) B31 2 node Linear
               beam elements (3) B32 3 node Quadratic Beam
               elements
155         global_seed = 1 ; % Seeding distance for meshing (mm)
                in Abaqus (only applicable for beam elements)
156
157         material_file = 'Ti64perfectplastic931.txt' ; %
               Materials file, set at [] if linear elastic.
               machinedTi64.txt - Ti64AsBuilt.txt -
               Ti64perfectplastic.txt -  Ti64perfectplastic842.
```

```
            txt
158         sup_node_ind1 = (node_list(:,2) < 0.01 ) ; % Find
               supported nodes - SPECFIC TO MBB's!!
159         sup_nodes = node_list(sup_node_ind1==1,1)' ;
160
161         con_nodes_ind = [] ;
162         con_nodes = [] ;
163
164 end
165
166 sigmaT = 0.8 ;
167 sigmaC = 0.8 ;
168 area = 1 ;
169 min_dia = 2 ; % Minimum diameter of truss bars in mm
170 buck_K = 0.699 ; % Column effective length factor for Euler
       Buckling analysis. Fixed-Fixed 0.5, Fixed-Free 0.699 and
       Free-Free 1.0
171 E_mod = 113.8 ; % Modulus of elasticity in GPa
172 poisson = 0.342 ; % Poisson ratio
173 density = 4.43 ; % Material density in g/cc
174 forceScale = 1 ; % For cantilver 2 this is set at 0.5: 100kN
       --> 50kN
175 areaScale = 1 ;
176 stabilityCoeff = 0.02 ;
177
178 problemType = 2 ; % PROBLEM TYPE: 1 - Secondary Optimsation,
       2 - Post-processing
179 FrameDir = 'frame_mod.txt' ;
180 bucklingCheck = 1 ; % Thicken buckling members
181 minAreaCheck = 1 ; % Increase member areas if below minimum
       threshold
182 cross_over_check = 1 ; % Split at crossovers
183 abaqusAnalysis = 1 ; % Enable abaqus analysis
184 %FrameFile = 'examples\can2.txt' ;
185 % delimiter = ' ' ;
186
187 % ------------- Morphing Members Settings --------------
188
189 noLoftPoints = 11 ;
190 morphOffset = 8.5 ; % Offset from member ends in mm
191 maxMemDia = 100 ; % Maximum expanded diameter as a fraction
       of member length
192 morphType = 2 ; % Members to morph 1) All members with a
       compression force in at least one load case 2)
       Predominantly compression members only
193
194 % --------------- DATA IMPORT FROM COPT ----------------
195
196 switch problemType
197
```

```
198     case 1
199
200         [elem_list, node_list] = ElementMerger(elem_list,
                node_list) ;
201         [elem_list, node_list] = collinear(elem_list,
                node_list) ;
202         %[elem_list, node_list] = overlapAnalysisv2(elem_list
                , node_list) ;
203         %             if sum(elem_list_orig(:,1) == elem_list
                (:,2)) == size(elem_list,1) && sum(elem_list_orig
                (:,2) == elem_list(:,3)) == size(elem_list,1)
204         %                 break
205         %             end
206
207         FrameCreate(elem_list,node_list,FrameDir) ; % Create
                a file in same format as frame.txt but with
                modified elements
208
209     case 2
210
211         if cross_over_check == 1 ;
212
213             [elem_list, node_list, elem_forces,
                    elem_comp_forces] = crossoverSkewLines(
                    elem_list, node_list, elem_forces,
                    elem_comp_forces) ;
214
215         else
216             node_list(:,5) = 0 ;
217
218         end
219
220         for k = 1 : size(elem_list,1) % Recalculate element
                lengths
221             elem_list(k,5) = norm(node_list(elem_list(k,2)
                    ,2:4) - node_list(elem_list(k,3),2:4)) ;
222         end
223
224         elem_list(:,4) = elem_list(:,4) * areaScale ;
225         postProcessing(elem_list,node_list,elem_forces,
                elem_comp_forces,min_dia,buck_K,E_mod,density,
                bucklingCheck,minAreaCheck, noLoftPoints,
                morphOffset, maxMemDia, morphType)
226
227         if abaqusAnalysis == 1
228             AbaqusLineFE(node_list, elem_list, E_mod, poisson
                    , analysis_type, NLGEOM, sup_type,
                    applied_force,elem_type,global_seed,load_type,
                    sup_nodes,con_nodes,transpose,shell_def,
                    material_file)
```

```
229             FrameVisualisation(elem_list,node_list)
230         end
231 end
```

Listing 10.3: Function that imports and organises text data from C++ based linprog optimiser

```
1  function [elem_list, node_list, volume, nodeListForces,
       elem_forces, elem_comp_forces] = FrameImpv2a(filename,
       delimiter,plotOption)
2
3  % --------------- COPT FRAME IMPORTER ------------------
4
5  % THIS FUNCTION IMPORTS THE FRAME.TXT FROM COPT AND GENERATES
       A NODE AND
6  % ELEMENT LIST
7
8  % --------------------IMPORT DATA ----------------------
9
10 truss_data = importdata(filename,delimiter,1) ;
11 truss_data = truss_data.data ;
12
13 x1 = truss_data(:,1) ;
14 y1 = truss_data(:,2) ;
15 z1 = truss_data(:,3) ;
16
17 x2 = truss_data(:,4) ;
18 y2 = truss_data(:,5) ;
19 z2 = truss_data(:,6) ;
20
21 bar_length = truss_data(:,7) ;
22
23 Ft = truss_data(:,10) ; % Tensile force of element
24 Fc = truss_data(:,11) ; % Compressive force of element
25 F_flag = truss_data(:,8) ; % Force flag: 1 - Tension or 2 -
       Compression
26
27 A = truss_data(:,12) ; % Truss bar CS area
28
29 tolerance = 0.01 ;
30
31 % ---------- GENERATE NODE AND ELEMENT LISTS -----------
32
33 no_elem = size(truss_data,1) ;
34
35 index_nde_one(1:no_elem) = 1 ;
36 index_nde_two(1:no_elem) = 2 ;
37 element = 1:1:no_elem ;
38
39 comb_elem_list = [element' x1 y1 z1 index_nde_one' ; element'
       x2 y2 z2 index_nde_two'] ;
40
```

```
41 comb_elem_list = sortrows(comb_elem_list,2) ;
42
43 node = 1 ;
44 y_ind = 1 ;
45 z_ind = 1 ;
46
47 for k = 1 : size(comb_elem_list,1)
48
49     if k < size(comb_elem_list,1)
50
51         curr_x = comb_elem_list(k, 2) ;
52         next_x = comb_elem_list(k+1, 2) ;
53
54     end
55
56     if k == size(comb_elem_list,1) || next_x - curr_x >
       tolerance
57
58         y_sort = sortrows(comb_elem_list(y_ind:k,:),3) ;  %
           creating and sorting a subset of y values for the
           set of x values
59
60         z_ind = 1 ;
61
62         for j = 1:size(y_sort,1)
63
64
65             elem = comb_elem_list(j,1) ;
66             index = comb_elem_list(j,5) ;
67
68             if j < size(y_sort,1)
69
70                 curr_y = y_sort(j, 3) ;
71                 next_y = y_sort(j+1, 3) ;
72
73             end
74
75
76             if j == size(y_sort,1) || next_y - curr_y >
                tolerance
77
78                 z_sort = sortrows(y_sort(z_ind:j,:),4) ; %
                    Sorting the z values from the subset of y
                    values
79
80                 z_ind ;
81                 j ;
82                 F = 0 ;
83
84                 for h = 1 : size(z_sort,1)
```

```
85
86                  elem = z_sort(h,1) ;
87                  index = z_sort(h,5) ;
88
89
90                  elem_list(elem,1) = elem ;
91                  elem_list(elem,index+1) = node ;
92                  elem_list(elem,4) = A(elem) ;
93
94                  forces = [Ft(elem) Fc(elem)] ; % Forces
                       in element from COPT
95                  [elem_force,tenOrComp] = max(abs(forces))
                        ; % Determine whether element is in
                       tension or compression by taking
                       largest value of the two
96                  [elem_force,comp] = min(forces) ;
97
98                  elem_forces(elem,1) = forces(tenOrComp) ;
                        % Set the element force as that
                       determined in above
99                  elem_comp_forces(elem,1) = forces(comp) ;
                        % Compression forces in elements -
                       for buckling analysis later
100
101                 if abs(F) < abs(Ft(elem)) || abs(F) < abs
                       (Fc(elem)) % Determine member with
                       maximum axial force for stability calc
                        during secondary optimisation
102
103                     if abs(Ft(elem)) > abs(Fc(elem))
104
105                         F = Ft(elem) ;
106
107                     else
108
109                         F = Fc(elem) ;
110
111                     end
112
113                 end
114
115                 if h < size(z_sort,1)
116
117                     curr_z = z_sort(h, 4) ;
118                     next_z = z_sort(h+1, 4) ;
119
120                 end
121
122
```

```
123                     if h == size(z_sort,1) || next_z - curr_z
                            > tolerance
124
125                         if index < 2
126
127                             truss_index = 1:3 ;
128
129                         else
130
131                             truss_index = 4:6 ;
132
133                         end
134
135                         node_list(node,1) = node ;
136                         node_list(node,2:4) = truss_data(elem
                               ,truss_index) ;
137                         nodeListForces(node,1) = F ;
138                         node = node + 1 ;
139                         F = 0 ;
140
141                         z_ind = j + 1 ;
142
143
144                     end
145                 end
146             end
147         end
148         y_ind = k + 1 ;
149     end
150 end
151
152 if size(node_list,1) ~= size(unique(comb_elem_list(:,(2:4)),'
       rows'),1)
153
154     error('Number of extracted nodes is incorrect')
155
156 else
157
158     disp('Correct number of nodes extracted')
159
160 end
161
162 volume = sum( A .* bar_length) ; % Calculate the volume of
       truss structure
163
164 % ---------------------- PLOTTING ----------------------
165
166 if strcmp(plotOption,'noPlot') == 0
167
168     figure
```

```
169     hold on
170
171     for k = 1:size(elem_list,1)
172
173         node1 = elem_list(k,2) ;
174         node2 = elem_list(k,3) ;
175
176         plot_x = [node_list(node1,2) node_list(node2,2)] ;
177         plot_y = [node_list(node1,3) node_list(node2,3)] ;
178         plot_z = [node_list(node1,4) node_list(node2,4)] ;
179
180         plot3(plot_x,plot_y,plot_z,'b')
181
182     end
183
184     for k = 1:size(node_list,1)
185
186         node_x = node_list(k,2) ;
187         node_y = node_list(k,3) ;
188         node_z = node_list(k,4) ;
189
190         scatter3(node_x,node_y,node_z,'xr')
191
192     end
193
194 end
```

Listing 10.4: Function that resizes members and generates CAD data for solid model generation

```
1  function [] = postProcessing(elem_list,node_list,elem_forces,
       elem_comp_forces,min_dia,buck_K,E_mod,density,
       bucklingCheck,minAreaCheck, noLoftPoints, morphOffset,
       maxMemDia, morthType)
2
3  % ------------------ NODAL EXPANSION -------------------
4
5  elem_areas = elem_list(:,4) ;
6  elem_lengths = elem_list(:,5) ;
7
8  all_nodes(:,1) = [elem_list(:,1) ; elem_list(:,1)] ; % All
       the element numbers
9  all_nodes(:,2) = [elem_list(:,2) ; elem_list(:,3)] ; % All
       the nodes
10
11 all_nodes(:,3) = [elem_areas ; elem_areas] ; % All the
       element areas
12 all_nodes = sortrows(all_nodes,2) ; % Sort list into
       ascending node number order
13
14 node_counter = 1 ;
15 area_counter = 1 ;
```

```
16 node_area_exp = [] ;
17
18 for k = 1 : size(all_nodes,1) ;
19
20     node = all_nodes(k,2) ; % Node
21     elem = all_nodes(k,1) ; % Element
22     orig_area(area_counter) = all_nodes(k,3);  % Element area
23
24     X = node_list(node,2) ; % Coordinates of prime node
25     Y = node_list(node,3) ;
26     Z = node_list(node,4) ;
27
28     sec_node = elem_list(elem,2:3) * (elem_list(elem,2:3) ~=
          node)' ;
29     sec_node_coor(area_counter,1:3) = node_list(sec_node,2:4)
           ;
30
31     if k == size(all_nodes,1) || all_nodes(k+1,2) > all_nodes
          (k,2)
32
33         if size(sec_node_coor,1) > 2
34             new_area = nodal_expansion(X, Y, Z, sec_node_coor
                  , orig_area) ;
35             node_area_exp = [node_area_exp ; node new_area] ;
                   % Create list of expanded node "areas"
36
37             if sum(new_area<orig_area) > 0
38                 error('New expanded area is less than one or
                       more of the truss areas')
39             end
40
41         else
42             node_area_exp = [node_area_exp ; node node_list(
                  node,5)] ;
43         end
44
45         orig_area = [] ;
46         sec_node_coor = [] ;
47         area_counter = 1 ;
48
49     else
50
51         area_counter = area_counter + 1 ;
52
53     end
54
55 end
56
57 % ------------ BUCKLING & MIN AREA MEASURES ------------
58
```

```
59 tot_mem_changed = 0 ;
60 buck_mem_change = 0 ;
61 min_area_enfor = 0 ;
62 orig_vol = 0 ;
63 new_vol = 0 ;
64
65 figure
66 axis equal
67
68 for k = 1:size(elem_list,1)
69
70     % First assess if members have sufficient buckling
       resistance then
71     % modify if nessary
72     elem = elem_list(k,1) ; % Element number
73     node1 = elem_list(elem_list(:,1) == elem,2) ; % Node 1 of
       element
74     node2 = elem_list(elem_list(:,1) == elem,3) ; % Node 2 of
       element
75     elem_area = elem_areas(k) ;  % CS area of element
76     bar_length = elem_lengths(k)  ; % Truss length
77     comp_force = elem_comp_forces(k) ; % Compressive force in
       truss - for multiple load cases
78
79     colour = 'r' ;
80
81     orig_vol = orig_vol + elem_areas(k) * elem_lengths(k) ; %
       Volume of truss element before PP measures enfoced
82
83     elem_area_exp(k,1) = elem ;
84     elem_area_exp(k,2) = elem_areas(k) ;
85
86     if bucklingCheck==1 && comp_force < 0 % i.e. in
       compression
87
88         E_buck_A = 2 * buck_K * bar_length * sqrt(abs(
           comp_force) / (E_mod * pi)) ; % Calculate area
           required for buckling resistance
89         colour = 'b' ;
90
91         if elem_area < E_buck_A
92
93             elem_areas(k) = E_buck_A ; % Area in mm^2
94
95             buck_mem_change = buck_mem_change + 1 ;
96
97             colour = 'g' ;
98
99         end
100
```

```
101     end
102
103     nodeA = node_list(elem_list(k,2),2:4) ;
104     nodeB = node_list(elem_list(k,3),2:4) ;
105     plot3([nodeA(1) nodeB(1)],[nodeA(3) nodeB(3)],[nodeA(2)
            nodeB(2)],colour)
106     hold on
107     axis off
108     % Next increase CS area of any members that are below the
            minimum area
109     % threshold
110
111     min_area = 0.25 * pi * min_dia ^ 2 ;
112
113     if minAreaCheck == 1 && elem_areas(k) < min_area
114
115             elem_areas(k) = min_area ;
116
117             min_area_enfor = min_area_enfor + 1 ;
118
119     end
120
121     node1_exp = node_area_exp(node_area_exp(:,1) == node1,2)
            ;
122     node2_exp = node_area_exp(node_area_exp(:,1) == node2,2)
            ;
123
124     if elem_areas(k) > node1_exp % Expand the node volumes to
            account for are expansion due to buckling and min
            area steps
125
126         node_area_exp(node_area_exp == node1,2) = elem_areas(
                k) ;
127
128     end
129
130      if elem_areas(k) > node2_exp
131
132          node_area_exp(node_area_exp == node2,2) = elem_areas
                (k) ;
133
134      end
135
136     new_vol = new_vol + elem_areas(k) * elem_lengths(k) ; %
            New volume of truss element
137
138 end
139
140
141
```

```
orig_mass = orig_vol * density / 1000 ; % Original mass in
   grams
new_mass = new_vol * density / 1000 ; % Mass after post-
   processing in grams
disp(['Total members: ' num2str(k)])
disp(['Members modified for increased buckling resitance: '
   num2str(buck_mem_change)])
disp(['Minmum area enforced on: ' num2str(min_area_enfor) '
   members'])
disp(['Mass increase from ' num2str(orig_vol) ' mm3' ' to '
   num2str(new_vol) ' mm3'])
disp(['Mass increase from ' num2str(orig_mass) ' g' ' to '
   num2str(new_mass) ' g'])

elem_list(:,4) = elem_areas ; % Update element list with new
   areas

fid = fopen('frame_edit.txt','w') ;
fidMemMorph = fopen('memberMorphData.txt','w') ;

for k = 1:size(elem_list,1)

    elem = elem_list(k,1) ; % Element number
    elem_area = elem_list(k,4) ;  % CS area of element
    bar_length = elem_list(k,5) ; % Truss length
    elem_forces(k) = elem_forces(k) ; % Axial force in truss
    comp_force = elem_comp_forces(k) ;

    node1 = (elem_list(elem,2)) ;
    node2 = (elem_list(elem,3)) ;

    A_node1 = node_area_exp((node_area_exp(:,1)==node1),2) ;
       % Area at node 1
    A_node2 = node_area_exp((node_area_exp(:,1)==node2),2) ;
       % Area at node 2

    if A_node1 < elem_area || A_node2 < elem_area

        error(['Area at element ' num2str(elem) ' is more
           than area at nodes ' num2str(node1) ' and/or '
           num2str(node2)])

    end

    X = [node_list(node1,2) node_list(node2,2)] ;
    Y = [node_list(node1,3) node_list(node2,3)] ;
    Z = [node_list(node1,4) node_list(node2,4)] ;

    print = [X(1) Y(1) Z(1) X(2) Y(2) Z(2) A_node1 A_node2
       elem_area] ;
```

```
180
181     fprintf(fid, '%d %d %d %d %d %d %d %d %d\n',print) ;
182
183     [memberMorphData] = memberMorph(elem_area, A_node1,
            A_node2, node_list(node1,2:4), node_list(node2,2:4),
            noLoftPoints, morphOffset, maxMemDia, comp_force,
            elem_forces(k), min_dia, morthType) ;
184     elemMorph = [] ; % Element number
185     elemMorph(1:size(memberMorphData,1),1) = k ;
186     memberMorphData = [elemMorph memberMorphData] ;
187     fprintf(fidMemMorph, '%i %d %d %d %d %d %d %d %d %d\n',
            memberMorphData') ;
188
189     plot3(X,Y,Z)
190     hold on
191
192 end
193
194 fclose(fid) ;
195 fclose(fidMemMorph)
196
197 fid = fopen('frame_nodes.txt','w') ;
198
199 for k = 1:size(node_area_exp,1)
200
201     node = node_area_exp(k,1) ;
202     X = node_list(node,2) ;
203     Y = node_list(node,3) ;
204     Z = node_list(node,4) ;
205     A = node_area_exp(k,2) ;
206
207     print = [node X Y Z A] ;
208
209     fprintf(fid, '%d %d %d %d %d\n',print) ;
210
211 end
212
213 fclose(fid) ;
```

Listing 10.5: Identifies and splits all members that intersect

```
1  function [elem_list, node_list, elem_list_forces,
       elem_comp_forces] = crossoverSkewLines(elem_list,
       node_list, elem_list_forces, elem_comp_forces)
2
3  % ---------------- CrossoverSkewLines ------------------
4
5  % Function identifies lines AND volumes that interesect.
       Volumes are deemed
6  % to interesect when the the elements lines are within a
       distance that is
```

```
% equal to the sum of the element radii

cross_elems = [] ;
crossPt =[] ;
tolerance = 1e-3 ;

node_list(:,5) = 0 ; % Add in extra column in node_list for
   nodal areas (expansion at joints)

skewXover = 0 ; % No. of skew line cross-overs
lineXover = 0 ; % No. of line intersections

skewAreas = [] ; % Area expansion for nodes on skew lines

for k = 1 : size(elem_list,1)

    elem1NodeA = elem_list(k,2) ; % Node A of Element 1
    elem1NodeAXYZ = node_list(elem1NodeA,2:4) ; % (x,y,z) of
       Elem 1 Node A
    elem1NodeB = elem_list(k,3) ; % Node A of Element 1
    elem1NodeBXYZ = node_list(elem1NodeB,2:4) ; % (x,y,z) of
       Elem 1 Node B
    elem1Vec = node_list(elem1NodeA,2:4) - node_list(
       elem1NodeB,2:4) ; % Vector of element 1
    elem1Rad = sqrt(elem_list(k,4) / pi) ; % Radius of Elem 1

    for j = k + 1 : size(elem_list,1)

        elem2NodeA = elem_list(j,2) ; % Node A of Element 2
        elem2NodeAXYZ = node_list(elem2NodeA,2:4) ; % (x,y,z)
            of Elem 2 Node A
        elem2NodeB = elem_list(j,3) ; % Node A of Element 2
        elem2NodeBXYZ = node_list(elem2NodeB,2:4) ; % (x,y,z)
            of Elem 2 Node B
        elem2Vec = node_list(elem2NodeA,2:4) - node_list(
           elem2NodeB,2:4) ;  % Vector of element 2
        elem2Rad = sqrt(elem_list(j,4) / pi) ; % Radius of
           Elem 2

        cosine = dot(elem1Vec,elem2Vec)/(norm(elem1Vec)*norm(
           elem2Vec)) ; % Calculate cosine of two vectors, if
            |cosine| == 1 then they are parallel

        if abs(cosine) > 1 - tolerance % If elements are
           parallel

            %warning('There area parallel elements with
               overlapping volumes')
            dist = inf ;

```

```
45          else % Following calculation is for non-parallel
                elements
46
47              nA = dot(cross(elem2Vec,elem1NodeAXYZ-
                    elem2NodeAXYZ),cross(elem1Vec,elem2Vec)) ;
48              nB = dot(cross(elem1Vec,elem1NodeAXYZ-
                    elem2NodeAXYZ),cross(elem1Vec,elem2Vec)) ;
49              d = dot(cross(elem1Vec,elem2Vec),cross(elem1Vec,
                    elem2Vec)) ;
50
51              elem1Pt = elem1NodeAXYZ + (nA/d)*(elem1Vec) ; %
                    Closest point on Element 1
52              elem2Pt = elem2NodeAXYZ + (nB/d)*(elem2Vec) ; %
                    Closest point on Element 2
53              dist = norm(elem1Pt - elem2Pt) ; % Distance
                    between closest points
54
55              onElems = [((elem1Pt - elem1NodeBXYZ) / elem1Vec)
                     ((elem2Pt - elem2NodeBXYZ) / elem2Vec)] ;
56
57              if sum(onElems < 1 - tolerance) == 2 && sum(
                    onElems > tolerance) == 2
58                  if dist < tolerance % Lines intersect
59
60                      cross_elems = [cross_elems ; k, j] ;
61                      crossPt = [crossPt; elem1Pt] ;  % Add
                            location of intersection (elem1Pt =
                            elem2Pt)
62                      skewAreas = [skewAreas ; 0] ;
63                      lineXover = lineXover + 1 ;
64
65                  elseif dist < elem1Rad + elem2Rad % If
                        elements are not parallel
66
67                      cross_elems = [cross_elems ; k, 0 ; 0, j]
                             ;
68                      crossPt = [crossPt; elem1Pt ; elem2Pt] ;
69                      skewXover = skewXover + 1 ;
70                      node_coor = 0.5*(elem1Pt + elem2Pt) ; %
                            Mid point between closest points acts
                            as "joint node"
71                      sec_node_coor = [elem1NodeAXYZ ;
                            elem1NodeBXYZ ; elem2NodeAXYZ ;
                            elem2NodeBXYZ] ; % [X Y Z] of the
                            nodes of both elements
72                      elem_areas = [elem_list(k,4) ; elem_list(
                            k,4) ; elem_list(j,4) ; elem_list(j,4)
                            ] ; % Areas of split elems
73
```

```
74                      Exp_area = nodal_expansion(node_coor(1),
                           node_coor(2), node_coor(3),
                           sec_node_coor, elem_areas) ; %
                           Calculate expanded area using same
                           method as line intersections
75                      skewAreas = [skewAreas ; Exp_area ;
                           Exp_area] ; % Add areas to list to be
                           indexed into node_list
76                      %Perform area expansion
77                  end
78              end
79          end
80
81
82
83      end
84
85  end
86
87  if lineXover + skewXover > 0
88      [node_list elem_list elem_list_forces elem_comp_forces] =
            creatCrossoverPts3D(node_list, elem_list,
          cross_elems, crossPt, 0.01, skewAreas,
          elem_list_forces, elem_comp_forces) ;
89  end
90
91  figure
92  hold on
93  axis off
94
95  disp(['Number of line intersections cross-overs: ' num2str(
       lineXover)])
96  disp(['Number of skew line cross-overs: ' num2str(skewXover)
       ])
97
98  for k = 1:size(elem_list,1)
99
100     if elem_list_forces(k) > 0
101         forceColour = 'r';
102     else
103         forceColour = 'b';
104     end
105
106     delta = 0.1 ;
107     node1 = elem_list(k,2) ;
108     node2 = elem_list(k,3) ;
109
110     X = [node_list(node1,2) node_list(node2,2)] ;
111     Y = [node_list(node1,3) node_list(node2,3)] ;
112     Z = [node_list(node1,4) node_list(node2,4)] ;
```

```
113
114     plot_x = [node_list(node1,2) node_list(node2,2)] ;
115     plot_y = [node_list(node1,3) node_list(node2,3)] ;
116     plot_z = [node_list(node1,4) node_list(node2,4)] ;
117
118     text(X(1) + delta, Y(1) + delta, Z(1) + delta,num2str(
            node1))
119     text(X(2) + delta, Y(2) + delta, Z(2) + delta,num2str(
            node2))
120
121     plot3(plot_x,plot_y,plot_z,forceColour)
122     scatter3(X,Y,Z,'xr')
123
124 end
```

Listing 10.6: Function that creates new nodes at the intersection points of member pairs

```
1  function [node_list elem_list elem_list_forces
       elem_comp_forces]=  creatCrossoverPts3D(node_list,
       elem_list, cross_elems, crossPt, mergeradius, skewAreas,
       elem_list_forces, elem_comp_forces)
2  % create crossover nodes
3  % break up connections
4  % new nodes have only [x,y,z] information
5  % new connections will copy original connection information
6  % mergeradius: default 1e-3
7
8  if nargin <5
9    mergeradius=1e-3;
10 end
11
12 % algorithm:
13 % create all new nodes
14 % for each connection, find all crossovers
15 % break up connection
16
17 no_nodes=size(node_list,1);
18 num=size(crossPt,1); % Number of crossover nodes
19 new_node = zeros( num,size(node_list,2) );
20 new_node(:,1:4) = [(no_nodes+1:no_nodes+num)' crossPt];
21 node_list = [node_list ; (no_nodes+1:no_nodes+num)', crossPt,
       skewAreas] ; % Add in crossover nodes to node list
22
23 xElems=unique(cross_elems(:,1:2)) ;  % List of elements that
       intersect
24 xElems(xElems==0) = [] ; % Remove zero
25 new_elem_num=0;
26
27 for i=1:length(xElems)
28     elem = xElems(i) ;
```

```
29      Ptlst = find( cross_elems(:,1)== elem | cross_elems(:,2)
           == elem) ; % Find elem index in the cross_elems list
30      Ndlst = Ptlst + no_nodes ; % Index for node in updated
           node list
31      % add existing nodes:
32      Ndlst=[Ndlst(:); elem_list(elem,2:3)'] ;
33      %find a correct order, based on distance to an existing
           node
34      Nddist = ( node_list(Ndlst,2)-node_list(elem_list(elem,2)
           ,2)).^2 + ( node_list(Ndlst,3)-node_list(elem_list(
           elem,2),3)).^2+...
35          ( node_list(Ndlst,4)-node_list(elem_list(elem,2),4))
               .^2;
36      Nddist = [Nddist , Ndlst(:)];
37      Nddist=sortrows(Nddist,1) ;
38      Ndlst=Nddist(:,2) ;
39
40      %create new connections for line intersection
41      for j=1 : length(Ndlst)-1
42
43          new_elem_num=new_elem_num+1;
44          new_elem(new_elem_num,:)= [0 Ndlst(j), Ndlst(j+1),
               elem_list(elem, 4:end)];
45          new_elem_list_forces(new_elem_num) = elem_list_forces
               (elem) ;
46          new_elem_comp_forces(new_elem_num) = elem_comp_forces
               (elem) ;
47
48      end
49  end
50
51  new_elem=new_elem(1:new_elem_num,:);
52  delCnlst=xElems;
53  elem_list(delCnlst,:)=[] ;
54  elem_list_forces(delCnlst,:)=[] ;
55  elem_comp_forces(delCnlst,:)=[] ;
56  elem_list=[elem_list;new_elem];
57  elem_list_forces = [elem_list_forces ; new_elem_list_forces']
        ;
58  elem_comp_forces = [elem_comp_forces ; new_elem_comp_forces']
        ;
59  elem_list(:,1) = 1:size(elem_list,1) ;
60
61  % % merge nodes
62  % [node_list elem_list]=LineConnection.mergeCloseNd3D(
       node_list,elem_list,mergeradius);
63  end
```

Listing 10.7: Function that determines the radius of each joint

```
function [new_area] = nodal_expansion(X, Y, Z, sec_node_coor,
    orig_area)

% -- Function expands the cross sectional area at nodes based
    on the number
% and orientation of elements joined to the node

% sec_node_coor --> [X2 Y2 Z2]

for k = 1:size(sec_node_coor,1)

    Vx1 = X - sec_node_coor(k,1) ;
    Vy1 = Y - sec_node_coor(k,2) ;
    Vz1 = Z - sec_node_coor(k,3) ;

    new_area = orig_area(k) ;

    for j = 1:size(sec_node_coor,1) - 1

        if k + j > size(sec_node_coor,1)

            index = k + j - size(sec_node_coor,1) ;

        else

            index = k + j ;

        end

        Vx2 = X - sec_node_coor(index,1) ;
        Vy2 = Y - sec_node_coor(index,2) ;
        Vz2 = Z - sec_node_coor(index,3) ;

        norm_v1 = max(abs([Vx1 Vy1 Vz1])) ;
        norm_v2 = max(abs([Vx2 Vy2 Vz2])) ;

        dot_p = dot(([Vx1 Vy1 Vz1] / norm_v1),([Vx2 Vy2 Vz2]
           / norm_v2)) ;

        %sca_pro = abs(dot_p / (norm([Vx1 Vy1 Vz1] / norm_v1)
           )^2) ;

        sca_pro = dot_p / (norm([Vx1 Vy1 Vz1] / norm_v1))^2 ;

        if sca_pro < 0

            sca_pro = 0 ;

        end

```

```
47          new_area = new_area + sca_pro * orig_area(index) ;
48
49      end
50
51      exp_area(k) = new_area ;
52
53  end
54
55  new_area = max(exp_area) ;
```

Listing 10.8: Function that identifies and modifies collinear members that either partially or completely overlap

```
1   function [elem_list, node_list] = collinear(elem_list,
        node_list)
2
3   tolerance = 1e-6 ; % Tolerance for numerical errors
4
5   elem_create = [] ;
6   no_created_elems = 0 ;
7   modified_elems = [] ;
8   duplicate_elements = [] ;
9   for e1 = 1 : size(elem_list,1)
10      for e2 = e1+1 : size(elem_list,1)
11
12          vec_e1 = node_list(elem_list(e1,2),2:4) - node_list(
                elem_list(e1,3),2:4) ; % Vector of element 1
13          vec_e2 = node_list(elem_list(e2,2),2:4) - node_list(
                elem_list(e2,3),2:4) ; % Vector of element 2
14
15          cosine = dot(vec_e1,vec_e2)/(norm(vec_e1)*norm(vec_e2
                )) ; % Calculate cosine of two vectors, if |cosine
                | == 1 then they are parallel
16
17          if abs(cosine) > 1 - tolerance % If elements have the
                 same direction
18
19              nodes_match = [(elem_list(e1,2) ~= elem_list(e2
                    ,2:3)) (elem_list(e1,3) ~= elem_list(e2,2:3))]
                     ;
20
21              if sum(nodes_match) == 2 % If the two elements
                    are duplicates (same start and end nodes)
22
23                  duplicate_elements = [duplicate_elements ; e1
                        ] ;
24                  %error('Duplicate elements')
25
26              else
27
28                  nodes = [elem_list(e2,2) elem_list(e2,3)] ;
```

```
29                  nodes_uni = [(elem_list(e1,2) ~= elem_list(e2
                        ,2)) (elem_list(e1,2) ~= elem_list(e2,3))]
                         ;
30
31                  nodes = nodes(nodes_uni) ;
32
33                  vec_e1e2 = node_list(elem_list(e1,2),2:4) -
                        node_list(nodes(1),2:4) ; % Vector between
                         node on element 1 and node on element 2
34
35                  cosine_e1e2 = dot(vec_e1,vec_e1e2)/(norm(
                        vec_e1)*norm(vec_e1e2)) ;  % Calculate
                        cosine of two vectors, if |cosine_e1e2| ==
                         1 then they are collinear
36
37                  if abs(cosine_e1e2) > 1 - tolerance % If
                        element are collinear
38
39                      if sum(nodes_match) == 4 % If two
                            elements dont share a node
40
41                          elem1_nodeA = elem_list(e1,2) ;
42                          elem1_nodeB = elem_list(e1,3) ;
43                          elem2_nodeA = elem_list(e2,2) ;
44                          elem2_nodeB = elem_list(e2,3) ;
45
46                          vec_e1Ae2A = (node_list(elem1_nodeA
                                ,2:4) - node_list(elem2_nodeA,2:4)
                                ) / vec_e1 ; % (Elem 1 node A -->
                                Elem 2 node A) / (Elem 1 node A
                                --> Elem 1 node B)
47                          vec_e1Ae2B = (node_list(elem1_nodeA
                                ,2:4) - node_list(elem2_nodeB,2:4)
                                ) / vec_e1 ; % (Elem 1 node A -->
                                Elem 2 node B) / (Elem 1 node A
                                --> Elem 1 node B)
48
49                          vec_e2Ae1A = (node_list(elem2_nodeA
                                ,2:4) - node_list(elem1_nodeA,2:4)
                                ) / vec_e2 ; % (Elem 2 node A -->
                                Elem 1 node A) / (Elem 2 node A
                                --> Elem 2 node B)
50                          vec_e2Ae1B = (node_list(elem2_nodeA
                                ,2:4) - node_list(elem1_nodeB,2:4)
                                ) / vec_e2 ; % (Elem 2 node A -->
                                Elem 1 node B) / (Elem 2 node A
                                --> Elem 2 node B)
51
52
```

```
53                      overlap_vec = [vec_e1Ae2A ;
                           vec_e1Ae2B ; vec_e2Ae1A ;
                           vec_e2Ae1B ] ;
54                      overlap_nodes = [elem2_nodeA ;
                           elem2_nodeB ; elem1_nodeA ;
                           elem2_nodeB] ;
55
56                      if sum(overlap_vec > 0 & overlap_vec
                           < 1) == 2
57
58                          elems = [e1 e2] ;
59                          elem_sel = elems([(sum(
                               overlap_vec(1:2)>0) == 2) (sum
                               (overlap_vec(3:4)>0) == 2)]) ;
60                          nodes_ind = [(sum(overlap_vec
                               (1:2)>0) == 2) (sum(
                               overlap_vec(1:2)>0) == 2) (sum
                               (overlap_vec(3:4)>0) == 2) (
                               sum(overlap_vec(3:4)>0) == 2)]
                                ;
61                          node_origin = elem_list(elem_sel
                               ,2) ;
62                          end_node = elem_list(elem_sel,3)
                               ;
63
64                          node_order = [overlap_vec(
                               nodes_ind,1) overlap_nodes(
                               nodes_ind,1) ; 1 end_node ] ;
65                          node_order = sortrows(node_order
                               ,1) ; % Sort so that the nodes
                                are in distance order from
                               origin node
66                          elem_list(elem_sel,2:3) = [
                               node_origin node_order(1,2)] ;
67                          elem_list(elems(elems~=elem_sel)
                               ,2:3) = [node_order(1,2)
                               node_order(2,2)] ;
68                          elem_create = [elem_create ; 0 [
                               node_order(2,2) node_order
                               (3,2)] 0 ] ;
69
70                          modified_elems = [modified_elems
                               ; e1 ; e2] ;
71
72
73                      end
74
75                  else % they share a node
76
```

```
77                  elem1_nodes = elem_list(e1,2:3) ; %
                        Node A and B of element 1
78                  shared_node = elem1_nodes([(sum(
                        nodes_match(1:2)) < 2) (sum(
                        nodes_match(3:4)) < 2)]) ; % Node
                        share by both elements
79                  elem1_sec_node = elem_list(e1,logical
                        ([0 (elem_list(e1,2:3) ~=
                        shared_node) 0])) ; % Find
                        unshared node of elem1
80                  elem2_sec_node = elem_list(e2,logical
                        ([0 (elem_list(e2,2:3) ~=
                        shared_node) 0])) ; % Find
                        unshared node of elem2
81
82                  elem1_vec = node_list(elem1_sec_node
                        ,2:4) - node_list(shared_node,2:4)
                         ;
83                  elem2_vec = node_list(elem2_sec_node
                        ,2:4) - node_list(shared_node,2:4)
                         ;
84
85                  elem1_overlap = elem1_vec / elem2_vec
                         ;
86                  elem2_overlap = elem2_vec / elem1_vec
                         ;
87
88                  if elem1_overlap > 0 && elem1_overlap
                         < 1 % If element 1 is shorter
                        than element 2
89
90                      elem_list(e2,logical([0 (
                            elem_list(e2,2:3) ==
                            shared_node) 0])) =
                            elem1_sec_node ;
91                      modified_elems = [modified_elems
                            ; e2] ;
92
93                  elseif elem2_overlap > 0 &&
                        elem2_overlap < 1 % If element 2
                        is shorter than element 1
94
95                      elem_list(e1,logical([0 (
                            elem_list(e1,2:3) ==
                            shared_node) 0])) =
                            elem2_sec_node ;
96                      modified_elems = [modified_elems
                            ; e1] ;
97
98                  end
```

```
99
100                     end
101
102                 end
103
104             end
105
106         end
107
108     end
109
110 end
111
112 if size(elem_create,1) > 0
113
114     elem_create(:,1) = size(elem_list,1) + 1 : size(elem_list
        ,1) + size(elem_create,1) ;
115     elem_list = [elem_list ; elem_create] ; % Add new entries
          to element list
116
117 end
118
119 if size(modified_elems,1) + size(duplicate_elements,1) > 0
120
121     disp([num2str(size(modified_elems,1)) ' collinear
        elements modified, ' num2str(size(elem_create,1)) '
        elements were created and ' num2str(size(
        duplicate_elements,1)) ' duplicate elements were
        removed.'])
122
123 else
124
125     disp('No collinear elements present')
126
127 end
128 figure
129 hold on
130
131 for k = 1 : size(elem_list,1)
132
133     elem = elem_list(k,1) ;
134     nodeA = elem_list(k,2) ;
135     nodeB = elem_list(k,3) ;
136
137     if sum(elem_list(k,1) == modified_elems) == 1
138         colour = 'g' ;
139
140     %elseif sum(elem_list(k,1) == elem_create(:,1))
141         %colour = 'r' ;
142
```

```
143     else
144         colour = 'b' ;
145 
146     end
147     plot3(node_list([nodeA nodeB],2),node_list([nodeA nodeB
            ],3),node_list([nodeA nodeB],4),colour)
148     scatter3(node_list(nodeA,2),node_list(nodeA,3),node_list(
            nodeA,4))
149     scatter3(node_list(nodeB,2),node_list(nodeB,3),node_list(
            nodeB,4))
150 end
151 
152 
153 if size(duplicate_elements,1) > 0
154 
155     figure
156     hold on
157     title('Duplicate Elements')
158     for k = 1 : size(elem_list,1)
159 
160         elem = elem_list(k,1) ;
161         nodeA = elem_list(k,2) ;
162         nodeB = elem_list(k,3) ;
163 
164         if sum(elem_list(k,1) == duplicate_elements) > 0
165             colour = 'g' ;
166 
167             %elseif sum(elem_list(k,1) == elem_create(:,1))
168             %colour = 'r' ;
169 
170         else
171             colour = 'b' ;
172 
173         end
174         plot3(node_list([nodeA nodeB],2),node_list([nodeA
                nodeB],3),node_list([nodeA nodeB],4),colour)
175         scatter3(node_list(nodeA,2),node_list(nodeA,3),
                node_list(nodeA,4))
176         scatter3(node_list(nodeB,2),node_list(nodeB,3),
                node_list(nodeB,4))
177     end
178     elem_list(duplicate_elements,:) = [] ; % Remove duplicate
            elements
179 end
```

Listing 10.9: Function that modifies layout of members that overlap due to there being a small angle between them

```
1 % Overlap Function check for any members that diverge at a
      small angle to
2 % each other and merges them
```

```
3
4  function [elem_list, node_list] = overlapAnalysisv2(elem_list
       , node_list)
5
6  A = 1 ;
7
8  elem_list_orig = elem_list ;
9  node_list_orig = node_list ;
10
11 node_colours = cell(size(node_list,1),1) ;
12 for k = 1 : size(node_list,1)
13     node_colours{k} = 'b' ;
14 end
15 elem_colours = cell(size(elem_list,1),1) ;
16 for k = 1 : size(elem_list,1)
17     elem_colours{k} = 'b' ;
18     elem_mod_colours{k} = 'b' ;
19 end
20
21
22 for k = 1:size(elem_list,1)
23
24     nodeA = node_list(elem_list(k,2)) ;
25     nodeB = node_list(elem_list(k,3)) ;
26
27     elem_lengths(k,1) = sqrt((node_list(nodeA,2) - node_list(
           nodeB,2))^2 + (node_list(nodeA,3) - node_list(nodeB,3)
           )^2 + (node_list(nodeA,4) - node_list(nodeB,4))^2) ; %
            calculate element lengths and add as new column in
           elem_list
28
29 end
30
31 all_nodes = [elem_list(:,[1 2]) ; elem_list(:,[1 3])] ;
32 elem_del = [] ;
33 modded_elems = [] ;
34
35 for k = 1:size(node_list,1) % 1 : size(node_list,1)
36
37     node = node_list(k,1) ;  % Node of joint
38
39     local_elems = all_nodes(all_nodes(:,2) == node,1) ; %
           Find all elements that connect to joint
40
41     if size(local_elems,1) > 1 % If more than two connections
            at joint then perform analysis
42
43         checked_elems = 1 ;
44
45         while checked_elems <= size(local_elems,1)
```

```
46
47              local_elems = sortrows([local_elems -elem_lengths
                   (local_elems(:,1))],2) ; % Sort elements in
                   descending order based on length
48              elem_mod = local_elems(1,1) ; % Element to modify
49              elem = local_elems(1,1) ; % Largest Vector
50              large_nodeB = elem_list(elem,2:3) * (elem_list(
                   elem,2:3) ~= node)' ; % B node of large
                   element
51              large_vec = node_list(large_nodeB,2:4) -
                   node_list(node,2:4) ; % Vector of large
                   element
52              local_elem_del = 1 ; % List of local elements to
                   exclude on next loop
53
54              for j = 1 : size(local_elems,1) - 1
55
56
57                  elem2 = local_elems(j+1,1) ; % secondary
                       vector
58                  elem_rad = sqrt(elem_list(elem,4)/pi) + sqrt(
                       elem_list(elem2,4)/pi) ; % Combined radii
                       of both elements
59
60                  sec_nodeB = elem_list(elem2,2:3) * (elem_list
                       (elem2,2:3) ~= node)' ; % B node of
                       secondary element
61
62                  sec_vec = node_list(sec_nodeB,2:4) -
                       node_list(node,2:4) ; % Vector of
                       secondary element
63
64                  dist = norm(sec_vec - large_vec * dot(sec_vec
                       ,large_vec)/norm(large_vec)^2) ; %
                       Determine perpendicular distance from
                       large element to B node of secondary
                       element
65
66                  if dist < elem_rad && dot(large_vec,sec_vec)
                       > 0 % Overlap and less than 90 degs to
                       each other
67
68                      sca_proj = dot(sec_vec,large_vec)/norm(
                           large_vec)^2 ; % Scalar projection of
                           second onto large element
69
70                      new_node = sca_proj * large_vec ; % new
                           coordinates of secondary element B
                           node
71                      node_colours{sec_nodeB} = 'r' ;
```

```
72                      node_list(sec_nodeB,2:4) = new_node +
                           node_list(node,2:4) ; % Update
                           node_list with new coordinates of
                           secondary element B node
73
74                      elem_list(elem_mod,logical([0 (elem_list(
                           elem_mod,2:3) == node) 0])) =
                           sec_nodeB ; % Update element list with
                            shortened large element connected at
                           newly created node and original nodeB
75                      elem_colours{elem_mod} = 'g' ; % Mark
                           elements in red that have been
                           modified
76                      elem_colours{elem2} = 'g' ;
77
78                      modded_elems = [modded_elems ; elem_mod ;
                            elem2] ;
79
80                      local_elem_del = [local_elem_del ; j+1] ;
                            % Delete elements from local element
                           list so they are not checked again
81
82                      elem_mod = elem2 ; % This will be the
                           next element that needs to be modified
83
84                  end
85
86              end
87
88              checked_elems = checked_elems + 1 ;
89              local_elems(local_elem_del,:) = [] ;
90
91          end
92
93      end
94
95  end
96
97  modded_elems = unique(modded_elems) ;
98
99  if size(modded_elems,1) > 0
100
101     disp([num2str(size(modded_elems,1)) ' elements were
           modified to avoid overlapping'])
102
103 end
104 % Plotting
105 figure
106 hold on
107
```

```
for k = 1 : size(elem_list_orig,1)

    elem = elem_list_orig(k,1) ;
    nodeA = elem_list_orig(k,2) ;
    nodeB = elem_list_orig(k,3) ;

    plot3(node_list_orig([nodeA nodeB],2),node_list_orig([
       nodeA nodeB],3),node_list_orig([nodeA nodeB],4),
       elem_colours{elem})
    scatter3(node_list_orig(nodeA,2),node_list_orig(nodeA,3),
       node_list_orig(nodeA,4),node_colours{nodeA})
    scatter3(node_list_orig(nodeB,2),node_list_orig(nodeB,3),
       node_list_orig(nodeB,4),node_colours{nodeB})
end

figure
hold on

for k = 1 : size(elem_list,1)

    elem = elem_list(k,1) ;
    nodeA = elem_list(k,2) ;
    nodeB = elem_list(k,3) ;

    plot3(node_list([nodeA nodeB],2),node_list([nodeA nodeB
       ],3),node_list([nodeA nodeB],4),elem_mod_colours{elem
       })
    scatter3(node_list(nodeA,2),node_list(nodeA,3),node_list(
       nodeA,4),node_colours{nodeA})
    scatter3(node_list(nodeB,2),node_list(nodeB,3),node_list(
       nodeB,4),node_colours{nodeB})
end
```

Listing 10.10: Function that merges chain members

```
function [elem_list, node_list] = ElementMerger(elem_list,
   node_list)

tolerance = 1e-6 ;

figure
hold on

all_nodes = [elem_list(:,[1 2]) ; elem_list(:,[1 3])] ;

elem_del = [] ;
new_elems = [] ;
for k = 1:size(node_list,1) % 1 : size(node_list,1)

    node = node_list(k,1) ;  % Node of joint

```

```
16      local_elems = all_nodes(all_nodes(:,2) == node,1) ;  %
            Find all elements that connect to joint
17
18      if length(local_elems) == 2 % if only two elements
            connect to node
19
20          vec_e1 = node_list(elem_list(local_elems(1),2),2:4) -
                node_list(elem_list(local_elems(1),3),2:4) ; %
            Vector of first element in pair
21          vec_e2 = node_list(elem_list(local_elems(2),2),2:4) -
                node_list(elem_list(local_elems(2),3),2:4) ; %
            Vector of second element in pair
22          cosine = dot(vec_e1,vec_e2)/(norm(vec_e1)*norm(vec_e2
            )) ; % cosine of two vectors
23
24          if abs(cosine) > 1 - tolerance % If collinear then
            must be a chain
25
26
27              if size(new_elems,1) > 0
28
29                  if sum(k == new_elems(:,2)) + sum(k ==
                        new_elems(:,3)) == 1 % Check to see if
                        this pair is connected to a previously
                        detected pair
30
31                      elem_uni = local_elems([sum(local_elems
                            (1) == elem_del) == 0 ; sum(
                            local_elems(2) == elem_del) == 0]) ;
32                      new_node = elem_list(elem_uni,2:3) ;
33                      new_node = new_node(elem_list(elem_uni
                            ,2:3) ~= k) ;
34
35                      elem_del = [elem_del ; elem_uni] ;
36
37
38                      row_index = logical((k == new_elems(:,2))
                            +(k == new_elems(:,3))) ;
39                      col_index = logical([0 (new_elems(
                            row_index,2:3) == k) 0]) ;
40                      new_elems(row_index, col_index) =
                            new_node ;
41
42                  elseif sum(k == new_elems(:,2)) + sum(k ==
                        new_elems(:,3)) == 2
43
44                      disp('si senor')
45
46                  else
47
```

```
48                      elem_del = [elem_del ; local_elems(1) ;
                           local_elems(2)] ; % Chain elements to
                           be removed
49                      nodes = [elem_list(local_elems(1),2:3)
                           elem_list(local_elems(2),2:3)] ; %
                           Nodes of two elements
50                      nodes = nodes(nodes ~= k) ; % End nodes
                           of chain pair
51                      new_elems = [new_elems ; 0 nodes 0] ;
52
53                  end
54
55              else
56
57                  elem_del = [elem_del ; local_elems(1) ;
                       local_elems(2)] ; % Chain elements to be
                       removed
58                  nodes = [elem_list(local_elems(1),2:3)
                       elem_list(local_elems(2),2:3)] ; % Nodes
                       of two elements
59                  nodes = nodes(nodes ~= k) ; % End nodes of
                       chain pair
60                  new_elems = [new_elems ; 0 nodes 0] ;
61
62              end
63
64
65
66
67          end
68
69      end
70
71  end
72
73  elem_list(elem_del,:) = [] ;
74  elem_list = [elem_list ; new_elems] ;
75  elem_list(:,1) = 1:size(elem_list,1) ;
76
77  if size(new_elems,1) > 0
78
79      disp([num2str(length(elem_del)) ' chain elements merged
           into ' num2str(size(new_elems,1)) ' elements'])
80
81  else
82
83      disp('No elements merged')
84
85  end
86
```

```
87
88  for k = 1 : size(elem_list,1)
89
90      elem = elem_list(k,1) ;
91      nodeA = elem_list(k,2) ;
92      nodeB = elem_list(k,3) ;
93
94      plot3(node_list([nodeA nodeB],2),node_list([nodeA nodeB
            ],3),node_list([nodeA nodeB],4))
95      scatter3(node_list(nodeA,2),node_list(nodeA,3),node_list(
            nodeA,4))
96      scatter3(node_list(nodeB,2),node_list(nodeB,3),node_list(
            nodeB,4))
97  end
```

Listing 10.11: Function that generates an Abaqus input file for analysis of truss structure

```
1   function AbaqusLineFE(node_list, elem_list, E_mod, poisson,
        analysis_type, NLGEOM, sup_type, applied_force, elem_type,
         global_seed, load_type, sup_nodes, con_nodes, transpose,
        shell_def, SS_data)
2
3   % ------------------ Convert to SI ---------------------
4   node_list(:,5) = [] ; % remove 5th column on node list as not
        used here
5
6   elem_list(:,4) = elem_list(:,4) * 1e-6 ; % Areas mm^2 --> m^2
7   elem_list(:,5) = elem_list(:,5) * 1e-3 ; % Lengths mm --> m
8   node_list(:,2:4) = node_list(:,2:4) * 1e-3 ; % Node locations
        mm --> m
9   E_mod = E_mod * 1e9 ; % GPa --> Pa
10  applied_force(:,1:3) = applied_force(:,1:3) * 1e-3 ; %
        Applied force locations mm --> m
11  applied_force(:,4:6) = applied_force(:,4:6) * 1e3 ; % Applied
        forces kN --> N
12  global_seed = global_seed * 1e-3 ; % Global seed mm --> m
13  geo_tol = 1e-6 ;
14  damping = 0.0 ;
15  elems = {'T3D2' 'B31' 'B32'} ; % T3D2 - Truss element axial
        forces only, B31 2 node Linear beam elements & B32 3 node
        Quadratic Beam elements
16
17  if strcmp('xzy',transpose) == 1
18
19      node_list = [node_list(:,1) node_list(:,2) node_list(:,4)
            node_list(:,3)] ;
20      applied_force = [applied_force(:,1) applied_force(:,3)
            applied_force(:,2) applied_force(:,4) applied_force
            (:,6) applied_force(:,5)] ;
21
22  end
```

```
23
24 % -------------- Element Discretization ----------------
25 % ------------- (FOR BEAM ELEMENTS ONLY) ---------------
26
27 if elem_type > 1 % Discretize each element into multiple beam
       elements
28
29     beam_elem_list = [] ;
30
31     for k = 1 : size(elem_list,1)
32
33         nodeA = elem_list(k,2) ;
34         nodeB = elem_list(k,3) ;
35         vector = node_list(nodeB,2:4) - node_list(nodeA,2:4)
               ;
36         divisions = round(norm(vector)/global_seed) ;
37
38         if divisions >= 2
39
40             nodeA_coords = node_list(nodeA,2:4) ;
41             node_locs = [linspace(nodeA_coords(1),vector(1)+
                   nodeA_coords(1),divisions+1)' linspace(
                   nodeA_coords(2),vector(2)+nodeA_coords(2),
                   divisions+1)' linspace(nodeA_coords(3),vector
                   (3)+nodeA_coords(3),divisions+1)'] ;
42
43             node_locs([1 end],:) = [] ;
44             new_node_nums = size(node_list,1) + 1 : 1 : size(
                   node_list,1) + size(node_locs,1) ;
45             node_list = [node_list ; new_node_nums' node_locs
                   ] ;
46
47             elem_nodes = [nodeA new_node_nums nodeB] ;
48
49             for j = 1:divisions
50
51                 beam_elem_list = [beam_elem_list ; (size(
                       beam_elem_list,1)+1) elem_nodes(j)
                       elem_nodes(j+1) elem_list(k,4) elem_list(k
                       ,5)] ;
52
53             end
54
55         else
56
57             beam_elem_list = [beam_elem_list ; (size(
                   beam_elem_list,1)+1) nodeA nodeB elem_list(k
                   ,4) elem_list(k,5)] ;
58
59         end
```

```
60      end
61
62      elem_list = [] ;
63      elem_list = beam_elem_list ;
64  end
65
66  fid = fopen('FramePPAbaqusInput.inp','w') ;
67
68  fprintf(fid,'%s\n%s\n%s\n%s\n%s\n','*HEADING ','**Generated
        from FramePP Matlab function','*RESTART,WRITE','*PART,
        NAME=LineElems','*END PART') ;
69  fprintf(fid,'%s\n%s\n','*PART, NAME=RigidShell','*END PART')
        ;
70  fprintf(fid,'%s\n%s\n','*ASSEMBLY, NAME=Hinge','*INSTANCE,
        NAME=Frame, PART=LineElems') ;
71  fprintf(fid,'%s\n','*NODE,NSET=NODES') ;
72
73  % --------------- Specifify Node List ------------------
74  for k = 1 : size(node_list,1)
75
76      fprintf(fid,'%i,%f,%f,%f\n',node_list(k,1:4)) ;
77
78  end
79
80  % --------------- Specifify Elem List ------------------
81
82  for k = 1 : size(elem_list,1)
83
84      fprintf(fid,'%s%s%s%s\n','*ELEMENT, TYPE=',elems{
            elem_type},', ELSET=ELEM',num2str(elem_list(k,1))) ;
85      fprintf(fid,'%i,%i,%i\n',elem_list(k,1:3)) ;
86
87  end
88
89  % -------------- Specifify Section Props ---------------
90
91  for k = 1 : size(elem_list,1)
92
93      if elem_type == 1
94
95          fprintf(fid,'%s%i%s\n%f,\n','*SOLID SECTION, ELSET=
                ELEM',elem_list(k,1),', MATERIAL=MAT1',elem_list(k
                ,4)) ;
96
97      else
98
99          radius = sqrt(elem_list(k,4)/pi) ;
100         vector = node_list(elem_list(k,2),2:4) - node_list(
                elem_list(k,3),2:4) ;
101         vector = vector/max(abs(vector)) ;
```

```
102         normal = [-vector(2) vector(1) 0] + 0.1;
103
104         fprintf(fid,'%s%i%s\n%f,\n','*BEAM SECTION,SECTION=
                CIRC,MATERIAL=MAT1,ELSET=ELEM',elem_list(k,1),',
                MATERIAL=MAT1',radius) ;
105         fprintf(fid,'%f,%f,%f\n',normal) ;
106
107     end
108
109 end
110
111 fprintf(fid,'%s\n','*END INSTANCE') ;
112
113 % --------- Specifying analytical rigid surface --------
114
115 if strcmp('SURFACE',load_type) == 1
116
117     shell_vertices = importdata(['Rigid Surface Definitions\'
            shell_def],',') ;
118
119     shell_vertices = shell_vertices * 1e-3 ; % Convert mm -->
            m
120
121     fprintf(fid,'%s\n%s\n','*INSTANCE, NAME=SHELL, PART=
            RigidShell','*SURFACE, TYPE=CYLINDER, NAME=RSURF') ;
122     fprintf(fid,'%s,    %f, %f\n','START',shell_vertices(1,1)
            ,shell_vertices(1,2)) ;
123
124     for k = 2 : size(shell_vertices,1)
125         fprintf(fid,'%s,    %f, %f\n','LINE',shell_vertices(k
                ,1),shell_vertices(k,2)) ;
126     end
127
128     fprintf(fid,'%s,    %f, %f\n','LINE',shell_vertices(1,1),
            shell_vertices(1,2)) ;
129     fprintf(fid,'%s\n','*END INSTANCE') ;
130
131     fprintf(fid,'%s\n%i, %f, %f, %f\n','*Node',1,
            applied_force(1:3)) ; % Creating reference node
132     fprintf(fid,'%s\n%i,\n','*NSET, nset=ref_node',1) ; %
            Creating node set for reference node
133
134     fprintf(fid,'%s\n','*NSET, nset=con_nodes, instance=Frame
            ') ; % Creating list for nodes connected to shell
135
136     for k = 1:length(con_nodes) % Populating list
137         fprintf(fid,'%i, ', con_nodes(k)) ;
138     end
139
140     if elem_type == 1
```

```
141         fprintf(fid,'\n%s\n','*Rigid Body, ref node=ref_node,
                pin nset=con_nodes, analytical surface=SHELL.
               RSURF') ; % Creating rigid body constraint
142     else
143         fprintf(fid,'\n%s\n','*Rigid Body, ref node=ref_node,
                tie nset=con_nodes, analytical surface=SHELL.
               RSURF') ; % Creating rigid body constraint
144     end
145
146 end
147
148 % ------------- Specifify Supported Nodes --------------
149
150 if size(sup_type,1) ~= size(sup_nodes,1)
151     error('Number of support groups types is not the same as
           the number of supported node groups')
152 end
153
154 for k = 1 : size(sup_nodes,1)
155     fprintf(fid,'%s%i%s\n','*NSET, nset=sup_nodes',k,',
           instance=Frame') ;% Creating list of supported nodes
156     for j = 1:size(sup_nodes,2)
157         fprintf(fid,'%i, ', sup_nodes(k,j)) ;
158     end
159     fprintf(fid,'\n') ;
160 end
161
162 if strcmp('POINT',load_type) == 1
163     force_node_ind = ((node_list(:,2) > (applied_force(1,1)-
           geo_tol)) & (node_list(:,2) <(applied_force(1,1)+
           geo_tol))) + ((node_list(:,3) > (applied_force(1,2)-
           geo_tol)) & (node_list(:,3) <(applied_force(1,2)+
           geo_tol))) + ((node_list(:,4) > (applied_force(1,3)-
           geo_tol)) & (node_list(:,4) <(applied_force(1,3)+
           geo_tol))) ;
164     force_node = node_list(force_node_ind==3,1) ;
165     fprintf(fid,'%s\n','*NSET, nset=PF_nodes, instance=Frame'
           )
166     fprintf(fid,'%i\n',force_node) ;
167 end
168
169 fprintf(fid,'\n%s\n','*END ASSEMBLY') ;
170
171 % ------------- Specifify Material Props ---------------
172
173 fprintf(fid,'%s\n','*MATERIAL, NAME=MAT1') ;
174 fprintf(fid,'%s\n','*ELASTIC') ;
175 fprintf(fid,'%f, %f\n',E_mod,poisson) ;
176
177 % ------------------- Analysis Type --------------------
```

```
178
179 if analysis_type == 1
180     fprintf(fid,'%s\n%s\n','*Step, name=Step-1, perturbation'
            ,'*Static') ;
181 elseif analysis_type > 1
182     stress_strain = importdata(['Material Properties\'
            SS_data],',') ;
183     fprintf(fid,'%s\n','*Plastic') ;
184
185     for k = 1 : size(stress_strain,1)
186         fprintf(fid,'%f,%f\n',stress_strain(k,1),
                stress_strain(k,2)) ;
187     end
188
189     if NLGEOM == 1
190         fprintf(fid,'%s\n','*Step, name=Step-1, nlgeom=YES')
                ;
191     else
192         fprintf(fid,'%s\n','*Step, name=Step-1') ;
193     end
194
195     if analysis_type == 2 % STATIC, GENERAL
196         fprintf(fid,'%s%f%s\n','*Static, stabilize, factor=',
                damping,', allsdtol=0, continue=NO') ;
197     elseif analysis_type == 3 % STATIC, RIKS
198         fprintf(fid,'%s%f%s\n','*Static, RIKS, stabilize,
                factor=',damping,', allsdtol=0, continue=NO') ;
199     end
200     fprintf(fid,'%s\n','1., 1., 1e-05, 1.') ;
201 else
202     error('Incorrect analysis type defined')
203 end
204
205 % ------------ Define support constraints --------------
206
207 fprintf(fid,'%s\n','*BOUNDARY') ;
208 for k = 1 : size(sup_type,1)
209         for j = 1:size(sup_type,2)
210             if sup_type(k,j) > 0
211                 fprintf(fid,'%s%i, %i\n','sup_nodes',k,j) ;
212             end
213         end
214 end
215
216 % ---------- Specify loaded Node(s)/surface ------------
217
218 fprintf(fid,'%s\n','*CLOAD') ;
219
220 if strcmp('POINT',load_type) == 1
221
```

```
222     fprintf(fid,'%s,%i,%f\n','PF_nodes',1,applied_force(1,4))
            ;
223     fprintf(fid,'%s,%i,%f\n','PF_nodes',2,applied_force(1,5))
            ;
224     fprintf(fid,'%s,%i,%f\n','PF_nodes',3,applied_force(1,6))
            ;
225
226 elseif strcmp('SURFACE',load_type) == 1
227
228     fprintf(fid,'%s,%i,%f\n','ref_node',1,applied_force(1,4))
            ;
229     fprintf(fid,'%s,%i,%f\n','ref_node',2,applied_force(1,5))
            ;
230     fprintf(fid,'%s,%i,%f\n','ref_node',3,applied_force(1,6))
            ;
231
232 else
233
234     error('Incorrect force type specified')
235
236 end
237
238
239 fprintf(fid,'%s','*END STEP') ;
240
241 fclose(fid) ;
```

References

Sung-Hoon Ahn, Michael Montero, Dan Odell, Shad Roundy, and Paul K. Wright. Anisotropic material properties of fused deposition modeling ABS. *Rapid Prototyping Journal*, 8(4):248–257, 2002. ISSN 1355-2546. doi: 10.1108/13552540210441166. URL http://www.emeraldinsight.com/10.1108/13552540210441166.

S. S. Al-Bermani, M. L. Blackmore, W. Zhang, and I. Todd. The Origin of Microstructural Diversity, Texture, and Mechanical Properties in Electron Beam Melted Ti-6Al-4V. *Metallurgical and Materials Transactions A*, 41(13):3422–3434, August 2010. ISSN 1073-5623. doi: 10.1007/s11661-010-0397-x. URL http://www.springerlink.com/index/10.1007/s11661-010-0397-x.

Gregoire Allaire, Frederic Gournay, Francois Jouve, and Anca-Maria Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control Cybern*, 34(October 2004):59–80, 2004.

Alphons A Antonysamy, Phil B Prangnell, and Jonathan Meyer. Effect of Wall Thickness Transitions on Texture and Grain Structure in Additive Layer Manufacture (ALM) of Ti-6Al-4V. *Materials Science Forum*, 709:205–210, 2012. doi: 10.4028/www.scientific.net/MSF.706-709.205.

Alphons Anandaraj Antonysamy. *Microstructure, Texture and Mechanical Property Evolution during Additive Manufacturing of Ti6Al4V Alloy for Aerospace Applications*. Traditional, The University of Manchester, 2012.

Arcam AB. Arcam A2 Technical Data Sheet, 2013. URL http://www.arcam.com/wp-content/uploads/Arcam-A2.pdf.

A Aremu, I Ashcroft, R Hague, R Wildman, and C Tuck. Suitability of SIMP and BESO Topology Optimization Algorithms for Additive Manufacture. In *Twenty First Annual International Solid Freeform Fabrication Symposium*, pages 679–692, Austin, 2010.

H. Attar, M. Calin, L.C. Zhang, S. Scudino, and J. Eckert. Manufacture by selective laser melting and mechanical behavior of commercially pure titanium. *Materials Science and Engineering: A*, 593:170–177, 2014. ISSN 09215093. doi: 10.1016/j.msea.2013.11.038. URL http://www.sciencedirect.com/science/article/pii/S0921509313012665.

A. Bauereiß, T. Scharowsky, and C. Körner. Defect generation and propagation mechanism during additive manufacturing by selective beam melting. *Journal of Materials Processing Technology*, 214(11):2522–2528, November 2014. ISSN 09240136. doi: 10.1016/j.jmatprotec.2014.05.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0924013614001691.

Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, November 1988. ISSN 00457825. doi: 10.1016/0045-7825(88)90086-2. URL http://linkinghub.elsevier.com/retrieve/pii/0045782588900862.

M.P. Bendsøe. Optimal shape design as a material distribution problem. *Structural and Multidisciplinary Optimization*, 202:193–202, 1989. URL http://www.springerlink.com/index/r14qv5r038736663.pdf.

M.P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69:635–654, 1999. URL http://www.springerlink.com/index/A0X3MX3W4RKALQDV.pdf.

M.P. Bendsøe and O Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, 2003. ISBN 3540429921, 9783540429920.

S Bontha, N Klingbeil, P Kobryn, and H Fraser. Thermal process maps for predicting solidification microstructure in laser fabrication of thin-wall structures. *Journal of Materials Processing Technology*, 178(1-3):135–142, September 2006. ISSN 09240136. doi: 10.1016/j.jmatprotec.2006.03.155. URL http://linkinghub.elsevier.com/retrieve/pii/S0924013606002949.

R Boyer. An overview on the use of titanium in the aerospace industry. *Materials Science and Engineering: A*, 213(1-2):103–114, August 1996. ISSN 09215093. doi: 10.1016/0921-5093(96)10233-1. URL http://linkinghub.elsevier.com/retrieve/pii/0921509396102331.

D Brackett, I Ashcroft, and R Hague. Topology optimisation for additive manufacture. In *Twenty Second Annual International Solid Freeform Fabrication Symposium*, pages 348–362, Austin, 2011.

CJ Brampton, Hyunsun A Kim, and James L Cunningham. Applications of 3D Level Set Topology Optimization. In *ASME 2012 ...*, pages 1–8, Chicago, 2012. URL http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1736315.

E. Brandl, B. Baufeld, C. Leyens, and R. Gault. Additive manufactured Ti-6Al-4V using welding wire: comparison of laser and arc beam deposition and evaluation with respect to aerospace material specifications. *Physics Procedia*, 5:595–606, January 2010. ISSN 18753892. doi: 10.1016/j.phpro.2010.08.087. URL http://linkinghub.elsevier.com/retrieve/pii/S1875389210005134.

Erhard Brandl, Christoph Leyens, Frank Palm, and Perrine Onteniente. Wire instead of powder ? Properties of additive manufactured Ti-6Al-4V for aerospace applications. In *Euro-uRapid*, 2008.

Erhard Brandl, Christoph Leyens, and Frank Palm. Mechanical Properties of Additive Manufactured Ti-6Al-4V Using Wire and Powder Based Processes. *IOP Conference Series: Materials Science and Engineering*, 26:012004, December 2011. ISSN 1757-899X. doi: 10.1088/1757-899X/26/1/012004. URL http://stacks.iop.org/1757-899X/26/i=1/a=012004?key=crossref.c362f0236ee048bdb0a9dfeda93401d5.

S Bureerat and J Limtragool. Structural topology optimisation using simulated annealing with multiresolution design variables. *Finite Elements in Analysis and Design*, 44(12-13):738–747, August 2008. ISSN 0168874X. doi: 10.1016/j.finel.2008.04.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0168874X08000681.

WG Burgers. On the process of transition of the cubic-body-centered modification into the hexagonal-close-packed modification of zirconium. *Physica*, 1(7-12):561–586, 1934. URL http://www.sciencedirect.com/science/article/pii/S0031891434802443.

Omer Cansizoglu, Ola L.A Harrysson, Harvey A West II, Denis R Cormier, and Tushar Mahale. Applications of structural optimization in direct metal fabrication. *Rapid Prototyping Journal*, 14(2):114–122, 2008. ISSN 1355-2546. doi: 10.1108/13552540810862082. URL http://www.emeraldinsight.com/10.1108/13552540810862082.

B. Caulfield, P.E. McHugh, and S. Lohfeld. Dependence of mechanical properties of polyamide components on build parameters in the SLS process. *Journal of Materials Processing Technology*, 182(1-3):477–488, February 2007. ISSN 09240136. doi: 10.1016/j.jmatprotec.2006.09.007. URL http://linkinghub.elsevier.com/retrieve/pii/S0924013606007886.

Kwai S. Chan, Marie Koike, Robert L. Mason, and Toru Okabe. Fatigue Life of Titanium Alloys Fabricated by Additive Layer Manufacturing Techniques for Dental Implants. *Metallurgical and Materials Transactions A*, 44(2):1010–1022, October 2012. ISSN 1073-5623. doi: 10.1007/s11661-012-1470-4. URL http://link.springer.com/10.1007/s11661-012-1470-4.

T Y Chen and Y H Chiou. Structural Topology Optimization Using Genetic Algorithms. In *World Congress on Engineering*, volume III, pages 3–7, London, 2013. ISBN 9789881925299.

A. L Cooke and J. A Soons. Variability in the Geometric Accuracy of Additively Manufactured Test Parts National Institute of Standards and Technology * Gaithersburg, MD, USA. In *SFF symposium*, pages 1–12, Austin, 2010.

Wael Darwich, Matthew Gilbert, and Andy Tyas. Optimum structure to carry a uniform load between pinned supports. *Structural and Multidisciplinary Optimization*, 42(1):33–42, January 2010a. ISSN 1615-147X. doi: 10.1007/s00158-009-0467-0. URL http://link.springer.com/10.1007/s00158-009-0467-0.

Wael Darwich, Matthew Gilbert, and Andy Tyas. Optimum structure to carry a uniform load between pinned supports. *Structural and Multidisciplinary Optimization*, 42(1):33–42, 2010b.

Joshua D Deaton and Ramana V Grandhi. A survey of structural and multidisciplinary continuum topology optimization : post 2000. *Structural and Multidisciplinary Optimization*, 2013. doi: 10.1007/s00158-013-0956-z.

Philippe Decaudin, Andre Gagalowicz, Inria Rocquencourt, and Le Chesnay Cedex. Fusion of 3D shapes. In *Fifth Eurographics Workshop on Animation and Simulation*, Oslo, 1994.

Peter Dewhurst, David G Taggart, Murat Demircubuk, and Arun Nair. The design of minimum-weight product structures : a preliminary case study combining theoretical optimum layout analysis with FEM studies . In *ABAQUS Users Conference*, pages 1–13, 2003.

N. P. Dijk, K. Maute, M. Langelaar, and F. Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48 (3):437–472, March 2013. ISSN 1615-147X. doi: 10.1007/s00158-013-0912-y. URL http://link.springer.com/10.1007/s00158-013-0912-y.

William S. Dorn, Ralph E. Gomory, and Herbert J. Greenberg. Automatic design of optimal structures. *Journal de Mechanique*, 3:25–52, 1964.

P. Edwards, a. O'Conner, and M. Ramulu. Electron Beam Additive Manufacturing of Titanium Components: Properties and Performance. *Journal of Manufacturing Science and Engineering*, 135(6):061016, November 2013. ISSN 1087-1357. doi: 10.1115/1.4025773. URL http://manufacturingscience.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4025773.

I Gibson and DW Rosen. *Additive manufacturing technologies: rapid prototyping to direct digital manufacturing*. Springer, 2009. ISBN 9781441911193. URL http://books.google.com/books?hl=en&lr=&id=jcFs0VVi9OAC&oi=fnd&pg=PR4&dq=Additive+Manufacturing+Technologies&ots=W8_in1GGS_&sig=Ho-FKFhVo7Mnn9Lb0btnBgTBgso.

Matthew Gilbert and Andrew Tyas. Layout optimization of large-scale pin-jointed frames. *Engineering Computations*, 20(8):1044–1064, 2003. doi: 10.1108/02644400310503017. URL http://www.ingentaconnect.com/content/mcb/182/2003/00000020/00000008/art00006.

Matthew Gilbert, Wael Darwich, Andy Tyas, and Paul Shepherd. Application of Large-scale Layout Optimization Techniques in Structural Engineering Practice. In *6th World Congresses of Structural and Multidisciplinary Optimization*, number June, Rio de Janeiro, 2005.

Cezary Graczykowski and Tomasz Lewiński. Michell cantilevers constructed within a half strip. Tabulation of selected benchmark results. *Structural and Multidisciplinary Optimization*, 42(6):869–877, July 2010. ISSN 1615-147X. doi: 10.1007/s00158-010-0525-7. URL http://link.springer.com/10.1007/s00158-010-0525-7.

R.T. Haftka and Ramana V Grandhi. STRUCTURAL SHAPE OPTIMIZATION-A Raphael T. HAFTKA SURVEY*. *Computer Methods Applied Mechanics and Engineering*, (57):91–106, 1986.

John W. Halloran, Vladislava Tomeckova, Susan Gentry, Suman Das, Paul Cilino, Dajun Yuan, Rui Guo, Andirudh Rudraraju, Peng Shao, Tao Wu, Taiwo R. Alabi, Wil Baker, Daira Legdzina, Dennis Wolski, Walter R. Zimbeck, and David Long. Photopolymerization of powder suspensions for shaping ceramics. *Journal of the European Ceramic Society*, 31(14):2613–2619, November 2011. ISSN 09552219. doi: 10.1016/j.jeurceramsoc.2010.12.003. URL http://linkinghub.elsevier.com/retrieve/pii/S0955221910005613.

L. He and M. Gilbert. Rationalization of trusses generated via layout optimization. *Structural and Multidisciplinary Optimization*, pages 1–18, 2015. ISSN 1615-147X. doi: 10.1007/s00158-015-1260-x. URL http://dx.doi.org/10.1007/s00158-015-1260-x.

W. S. Hemp. *Optimum Structures*. Clarendon Press, Oxford, 1973.

Peter Hougardy. Topology optimization of engine and gearbox mount castings. In *3rd European Hyperworks Technology Conference*, number November, 2009.

Nikolas Hrabe and Timothy Quinn. Effects of processing on microstructure and mechanical properties of a titanium alloy (Ti6Al4V) fabricated using electron beam melting (EBM), Part 2: Energy input, orientation, and location. *Materials Science and Engineering: A*, 573:271–277, June 2013. ISSN 09215093. doi: 10.1016/j.msea.2013.02.065. URL http://linkinghub.elsevier.com/retrieve/pii/S0921509313002554.

X. Huang and Y.M. Xie. Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, 43(14):1039–1049, October 2007. ISSN 0168874X. doi: 10.1016/j.finel.2007.06.006. URL http://linkinghub.elsevier.com/retrieve/pii/S0168874X07000777.

Xiaodong Huang and Yi-Min Xie. A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization*, 41(5):671–683, March 2010. ISSN 1615-147X. doi: 10.1007/s00158-010-0487-9. URL http://link.springer.com/10.1007/s00158-010-0487-9.

Charles. W. Hull. Apparatus for production of three-dimensional objects by stereolithography, 1986. URL http://www.google.com/patents/US4575330.

M Humbert, L Germain, N Gey, P Bocher, and M Jahazi. Study of the variant selection in sharp textured regions of bimodal IMI 834 billet. *Materials Science and Engineering: A*, 430(1-2):157–164, August 2006. ISSN 09215093. doi: 10.1016/j.msea.2006.05.047. URL http://linkinghub.elsevier.com/retrieve/pii/S092150930600832X.

Zhan Kang, Xiaoming Wang, and Rui Wang. Topology optimization of space vehicle structures considering attitude control effort. *Finite Elements in Analysis and Design*, 45(6-7):431–438, May 2009. ISSN 0168874X. doi: 10.1016/j.finel.2008.12.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0168874X08001881.

K Kempen, L Thijs, B Vrancken, S Buls, J Van Humbeeck, and J.P Kruth. Lowering thermal gradients in selective laser melting by pre-heating the baseplate. In *SFF symposium*, Austin, 2013.

H Khalid Rafi, N.V Karthik, Thomas L Starr, and Brent E Stucker. Mechanical property evaluation of Ti-6Al-4V parts made using Electron Beam Melting. In *Twenty Third Annual International Solid Freeform Fabrication Symposium*, pages 526–535, Austin, 2012.

P Kobryn. Microstructure and texture evolution during solidification processing of Ti6Al4V. *Journal of Materials Processing Technology*, 135(2-3):330–339, April 2003. ISSN 09240136. doi: 10.1016/S0924-0136(02)00865-8. URL http://linkinghub.elsevier.com/retrieve/pii/S0924013602008658.

P A Kobryn and S L Semiatin. The Laser Additive Manufacture of Ti-6Al-4V. *JOM*, (September):40–42, 2001.

Lars Krog, Alastair Tucker, M. Kemp, and Richard Boyd. Topology optimization of aircraft wing box ribs. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 1–16, 2004. URL http://www.altairhyperworks.co.uk/html/en-GB/Campaign/pages/papers/Aero/Paper6Airbus.pdf.

M Król, L A Dobrzaski, Reimann, and I Czaja. Surface quality in selective laser melting of metal powders. *Archives of Material Science and Engineering*, 60(2):87–92, 2013.

J.P. Kruth, M. Bartscher, S. Carmignato, R. Schmitt, L. De Chiffre, and a. Weckenmann. Computed tomography for dimensional metrology. *CIRP Annals - Manufacturing Technology*, 60(2):821–842, January 2011. ISSN 00078506. doi: 10.1016/j.cirp.2011.05.006. URL http://linkinghub.elsevier.com/retrieve/pii/S0007850611002083.

T Lewiński, M Zhou, and G.I.N. Rozvany. Extended exact least-weight truss layouts - Part II: Unsymmetric Cantilevers. *International Journal of Mechanical Sciences*, 36(5): 399–419, 1994.

Hao Li, Peigen Li, Liang Gao, Li Zhang, and Tao Wu. A level set method for topological shape optimization of 3D structures with extrusion constraints. *Computer Methods in Applied Mechanics and Engineering*, 283:615–635, January 2015. ISSN 00457825. doi: 10.1016/j.cma.2014.10.006. URL http://linkinghub.elsevier.com/retrieve/pii/S0045782514003764.

Guan-Chun Luh and Chun-Yi Lin. Structural topology optimization using ant colony optimization algorithm. *Applied Soft Computing*, 9(4):1343–1353, September 2009. ISSN 15684946. doi: 10.1016/j.asoc.2009.06.001. URL http://linkinghub.elsevier.com/retrieve/pii/S1568494609000672.

Guan-Chun Luh, Chun-Yi Lin, and Yu-Shu Lin. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, 11(2):2833–2844, March 2011. ISSN 15684946. doi: 10.1016/j.asoc.2010.11.013. URL http://linkinghub.elsevier.com/retrieve/pii/S1568494610002905.

G. Lutjering, J . C Williams, and A Gysler. *MICROSTRUCTURE AND MECHANICAL PROPERTIES OF TITANIUM ALLOYS*. 2000.

F. Martina, J. Mehnen, S.W. Williams, P. Colegrove, and F. Wang. Investigation of the benefits of plasma deposition for the additive layer manufacture of Ti6Al4V. *Journal of Materials Processing Technology*, 212(6):1377–1386, June 2012. ISSN 09240136. doi: 10.1016/j.jmatprotec.2012.02.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0924013612000532.

F Martina, S W. Williams, and P Colegrove. Improved microstructure and increased mechanical properties of additive manufacture produced Ti-6Al-4V by interpass cold rolling. In *SFF symposium*, pages 490–496, Austin, 2013.

Peter Mercelis and Jean-Pierre Kruth. Residual stresses in selective laser sintering and selective laser melting. *Rapid Prototyping Journal*, 12(5):254–265, 2006. ISSN 1355-2546. doi: 10.1108/13552540610707013. URL http://www.emeraldinsight.com/10.1108/13552540610707013.

A. G. M. Michell. The Limits of Economy of Material in Frame-structures. 1904.

MOSEK APS. The MOSEK optimization software, 2014. URL http://www.mosek.com.

K Mumtaz, P Vora, and N Hopkinson. A method to eliminate anchors/supports from directly laser melted metal powder bed processes. In *SFF symposium*, pages 55–64, Austin, 2011.

Stanley Osher and James a Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, November 1988. ISSN 00219991. doi: 10.1016/0021-9991(88)90002-2. URL http://linkinghub.elsevier.com/retrieve/pii/0021999188900022.

N Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. ISSN 0018-9472. doi: 10.1109/TSMC.1979.4310076.

E. W. Parkes. Joints in Optimum Frameworks. *International Journal for Solids and Structures*, I1:1017–1022, 1975.

MN Pavlović. A statically determinate truss model for thin shells: Onesurface analysis (membrane hypothesis). *International journal for numerical methods in Engineering*, 20(September 1983):1841–1861, 1984a. URL http://onlinelibrary.wiley.com/doi/10.1002/nme.1620201007/abstract.

MN Pavlović. A statically determinate truss model for thin shells: Twosurface analysis (bending theory). *International journal for numerical methods in*, 20(September 1983):1863–1884, 1984b. URL http://onlinelibrary.wiley.com/doi/10.1002/nme.1620201008/abstract.

M. Peters, C.H. Kumpfert, and C. Leyens. Titanium Alloys for Aerospace Applications. *Advanced Engineering Materials*, 5(6):419–427, 2003.

Pat Picariello. ASTM International Technical Committee F42 Fact Sheet, 2009.

D.A Porter and K.E Easterling. *Phase transformations in metals and alloys*. Cheltenham : Nelson Thornes, 2001.

W. Prager. Optimal layout of cantilever trusses. *Journal of Optimization Theory and Applications*, 23(1):111–117, September 1977. ISSN 0022-3239. doi: 10.1007/BF00932301. URL http://link.springer.com/10.1007/BF00932301.

T J Pritchard, M Gilbert, and A Tyas. Plastic Layout Optimization of Large-Scale Frameworks Subject to Multiple Load Cases, Member Self-Weight and with Joint Length Penalties. In *6th World Congresses of Structural and Multidisciplinary Optimization*, number June, Rio de Janeiro, 2005.

H K Rafi, N V Karthik, Haijun Gong, Thomas L Starr, and Brent E Stucker. Microstructures and Mechanical Properties of Ti6Al4V Parts Fabricated by Selective Laser Melting and Electron Beam Melting. *Materials Engineering and Performance*, 22(December): 3872–3883, 2013. doi: 10.1007/s11665-013-0658-0.

Amer Farhan Rafique, He LinShu, Ali Kamran, and Qasim Zeeshan. Multidisciplinary design of air launched satellite launch vehicle: Performance comparison of heuristic optimization methods. *Acta Astronautica*, 67(7-8):826–844, October 2010. ISSN 00945765. doi: 10.1016/j.actaastro.2010.05.016. URL http://linkinghub.elsevier.com/retrieve/pii/S0094576510001670.

R. Rezaie, M. Badrossamay, a. Ghaie, and H. Moosavi. Topology Optimization for Fused Deposition Modeling Process. *Procedia CIRP*, 6:522–527, January 2013. ISSN 22128271. doi: 10.1016/j.procir.2013.03.098. URL http://linkinghub.elsevier.com/retrieve/pii/S2212827113001728.

G I N Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, pages 217–237, 2009. doi: 10.1007/s00158-007-0217-0.

M. Schevenels, B.S. Lazarov, and O. Sigmund. Robust topology optimization accounting for spatially varying manufacturing errors. *Computer Methods in Applied Mechanics and Engineering*, 200(49-52):3613–3627, December 2011. ISSN 00457825. doi: 10.1016/j.cma.2011.08.006. URL http://linkinghub.elsevier.com/retrieve/pii/S0045782511002611.

Yu-Deok Seo, Hyun-Jung Kim, and Sung-Kie Youn. Shape optimization and its extension to topological design based on isogeometric analysis. *International Journal of Solids and Structures*, 47(11-12):1618–1640, June 2010. ISSN 00207683. doi: 10.1016/j.ijsolstr.2010.03.004. URL http://linkinghub.elsevier.com/retrieve/pii/S002076831000079X.

O. Sigmund. Topology optimization: a tool for the tailoring of structures and materials. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358(1765):211, 2000. URL http://rsta.royalsocietypublishing.org/content/358/1765/211.short.

O Sigmund. A 99 line topology optimization code written in Matlab. (21):120–127, 2001.

O Sigmund and J Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization*, (2):68–75, 1998. URL http://link.springer.com/article/10.1007/BF01214002.

Ole Sigmund. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, 43(5):589–596, March 2011. ISSN 1615-147X. doi: 10.1007/s00158-011-0638-7. URL http://www.springerlink.com/index/10.1007/s00158-011-0638-7.

Ole Sigmund and Kurt Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, pages 1031–1055, August 2013. ISSN 1615-147X. doi: 10.1007/s00158-013-0978-6. URL http://link.springer.com/10.1007/s00158-013-0978-6.

Tomasz Sokó and Tomasz Lewiński. On the solution of the three forces problem and its application in optimal designing of a class of symmetric plane frameworks of least weight. *Structural and Multidisciplinary Optimization*, 42(6):835–853, August 2010. ISSN 1615-147X. doi: 10.1007/s00158-010-0556-0. URL http://link.springer.com/10.1007/s00158-010-0556-0.

Anoop Kumar Sood, R.K. Ohdar, and S.S. Mahapatra. Improving dimensional accuracy of Fused Deposition Modelling processed part using grey Taguchi method. *Materials & Design*, 30(10):4243–4252, December 2009. ISSN 02613069. doi: 10.1016/j.matdes.2009.04.030. URL http://linkinghub.elsevier.com/retrieve/pii/S0261306909001794.

Anoop Kumar Sood, R.K. Ohdar, and S.S. Mahapatra. Parametric appraisal of mechanical property of fused deposition modelling processed parts. *Materials & Design*, 31(1):287–295, January 2010. ISSN 02613069. doi: 10.1016/j.matdes.2009.06.016. URL http://linkinghub.elsevier.com/retrieve/pii/S0261306909002945.

N. Stanford and P.S. Bate. Crystallographic variant selection in Ti6Al4V. *Acta Materialia*, 52(17):5215–5224, October 2004. ISSN 13596454. doi: 10.1016/j.actamat.2004.07.034. URL http://linkinghub.elsevier.com/retrieve/pii/S1359645404004434.

G Strang and R Kohn. Hencky-prandtl nets and constrained michell trusses. *Computer Methods in Applied Mechanics and Engineering*, 36(2):207–222, 1983.

Ruiyi Su, Liangjin Gui, and Zijie Fan. Topology and Sizing Optimization of Truss Structures Using Adaptive Genetic Algorithm with Node Matrix Encoding. *2009 Fifth International Conference on Natural Computation*, pages 485–491, 2009. doi: 10.1109/ICNC.2009.300. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5362845.

Krister Svanberg. The Method of Moving Aymptotes - A New Method For Structural Optimization. *International journal for numerical methods in Engineering*, 24:359–373, 1987.

Krister Svanberg. A CLASS OF GLOBALLY CONVERGENT OPTIMIZATION METHODS BASED ON CONSERVATIVE CONVEX SEPARABLE APPROXIMATIONS. *Society for Industrial and Applied Mathematics*, 12(2):555–573, 2002.

M. Szilvśi-Nagy and Gy. Mátyási. Analysis of STL files. *Mathematical and Computer Modelling*, 38(7-9):945–960, October 2003. ISSN 08957177. doi: 10.1016/S0895-7177(03)90079-3. URL http://linkinghub.elsevier.com/retrieve/pii/S0895717703900793.

S. Tammas-Williams, H. Zhao, F. Léonard, F. Derguti, I. Todd, and P.B. Prangnell. XCT Analysis of the Influence of Melt Strategies on Defect Population in Ti-6Al-4V Components Manufactured by Selective Electron Beam Melting. *Materials Characterization*, 102:47–61, 2015. ISSN 10445803. doi: 10.1016/j.matchar.2015.02.008. URL http://linkinghub.elsevier.com/retrieve/pii/S104458031500039X.

Lore Thijs, Frederik Verhaeghe, Tom Craeghs, Jan Van Humbeeck, and Jean-Pierre Kruth. A study of the microstructural evolution during selective laser melting of Ti6Al4V. *Acta Materialia*, 58(9):3303–3312, May 2010. ISSN 13596454. doi: 10.1016/j.actamat.2010.02.004. URL http://linkinghub.elsevier.com/retrieve/pii/S135964541000090X.

Matthew Tomlin and Jonathan Meyer. Topology Optimization of an Additive Layer Manufactured (ALM) Aerospace Part. In *The 7th Altair CAE Technology Conference 2011*, pages 1–9, Bristol, 2011.

Andy Tyas, Matthew Gilbert, Thomas Pritchard, and Mappin Street. Practical Plastic Layout Optimization of Trusses Incorporating Stability Considerations. In *6th World Congresses of Structural and Multidisciplinary Optimization*, number June, Rio de Janeiro, 2005.

Laurent van Belle, Guillaume Vansteenkiste, and Jean Claude Boyer. Investigation of Residual Stresses Induced during the Selective Laser Melting Process. *Key Engineering Materials*, 554-557:1828–1834, June 2013. ISSN 1662-9795. doi: 10.4028/www.scientific.net/KEM.554-557.1828. URL http://www.scientific.net/KEM.554-557.1828.

Ben Vandenbroucke and Jean-Pierre Kruth. Selective laser melting of biocompatible metals for rapid manufacturing of medical parts. *Rapid Prototyping Journal*, 13(4):196–203, 2007. ISSN 1355-2546. doi: 10.1108/13552540710776142. URL http://www.emeraldinsight.com/10.1108/13552540710776142.

M. Vasquez, B. Haworth, and N. Hopkinson. Optimum sintering region for laser sintered nylon-12. In *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, volume 225, pages 2240–2248, October 2011. ISBN 0954405411414. doi: 10.1177/0954405411414994. URL http://pib.sagepub.com/lookup/doi/10.1177/0954405411414994.

Carlos H. Villanueva and Kurt Maute. Density and level set-XFEM schemes for topology optimization of 3-D structures. *Computational Mechanics*, pages 133–150, April 2014. ISSN 0178-7675. doi: 10.1007/s00466-014-1027-z. URL http://link.springer.com/10.1007/s00466-014-1027-z.

B Vrancken, R Wauthle, J.P. Kruth, and J Van Humbeeck. Study of the influence of material properties on residual stress in selective laser melting. In *Twenty Forth Annual International Solid Freeform Fabrication Symposium*, Austin, 2013. URL http://utwired.engr.utexas.edu/lff/symposium/proceedingsArchive/pubs/Manuscripts/2013/2013-31-Vrancken.pdf.

David Wieloch and Ismail Taslim. Driving Million Dollar Savings at Ford using Structural Optimisation. In *The Altair Technology Conference*, number Figure 1, pages 5.1 – 5.7, 2004.

Chun-Yin Wu and Ko-Ying Tseng. Topology optimization of structures using modified binary differential evolution. *Structural and Multidisciplinary Optimization*, 42(6):939–953, July 2010. ISSN 1615-147X. doi: 10.1007/s00158-010-0523-9. URL http://www.springerlink.com/index/10.1007/s00158-010-0523-9.

Y M Xie, V Young, and G.P. Steven. 3D and multiple load case bi-directional evolutionary structural optimization (BESO). *Structural Optimization*, 18:183–192, 1999.

YM Xie. A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49(5):885–896, 1993. URL http://www.sciencedirect.com/science/article/pii/004579499390035C.

M Yulin. A level set method for structural topology optimization and its applications. *Advances in Engineering Software*, 35(7):415–441, July 2004. ISSN 09659978. doi: 10.1016/j.advengsoft.2004.06.004. URL http://linkinghub.elsevier.com/retrieve/pii/S0965997804000705.

H. Zarringhalam, N. Hopkinson, N.F. Kamperman, and J.J. de Vlieger. Effects of processing on microstructure and properties of SLS Nylon 12. *Materials Science and Engineering: A*, 435-436:172–180, November 2006. ISSN 09215093. doi: 10.1016/j.msea.2006.07.084. URL http://linkinghub.elsevier.com/retrieve/pii/S092150930601478X.

M Zhou and G.I.N Rozvany. The coc algorithm, part ii: topological, geometrical and generalized shape optimizaion. *Comput. Method Appl. Mech. Eng.*, 89(1-3):309–336, 1991.