

On the generation and characterisation of internal micro-architectures

David Richard Raymont

College of Engineering, Mathematics and Physical Sciences

University of Exeter

A thesis submitted for the degree of

Doctor of Philosophy in Engineering

May 2011

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signature:

I would like to dedicate this thesis to my loving parents.

Acknowledgements

I would like to thank my supervisors Liang Hao and Philippe Young for their support and guidance over the past three and half years. Their insight has been invaluable throughout this process.

Special thanks to Simpleware and the EPSRC for funding this work as part of a CASE award.

Thanks to Professor Shuguang Li, from the University of Nottingham, for his help with periodic boundary conditions and for supplying a template Abaqus input file.

I would also like to thank Chunze Yan and Ahmed Hussein for fabricating the metal parts using the SLM machine in X-AT, University of Exeter. In particular for generating numerous geometries using the developed software to test the manufacturability of different structures.

Thanks also to Peter Jerrard for allowing me to use the CAD model of the knee implant, which he created based on the original part, in this work.

Abstract

Open cell micro-architectures are used in a large number of applications, ranging from medical, such as bone scaffolds, to industrial, such as heat transfer structures. Traditionally these structures are manufactured using foaming processes, however advances in additive manufacturing (AM) now allow such structures to be designed computationally and fabricated to a high degree of precision.

In this thesis image-based methods are developed for the purpose of generating periodic micro-architectures based on implicit representations. The algorithms developed are shown to be efficient and robust, allowing for the creation of both surface and volume meshes. Methods are presented for the creation of functionally graded structures allowing for arbitrary variations in density between specifiable volume fractions. These algorithms are further extended for domain conforming applications as well as for internal structures in CAD models. By utilising a hybrid approach, imaging techniques can be exploited for the generation of internal structures in CAD models without de-featuring the original external geometry. The structures of interest are also shown to be manufacturable via selective laser melting (SLM).

The issue of characterisation, for linear elastic properties, is addressed through the use of a novel homogenisation technique. Large multi-scale problems in irregular domains are divided into smaller sub-volumes using established tetrahedral volume meshing techniques. By performing a series of virtual tests on these macroelements their effective properties can be computed and subsequently used in macro-simulations. The technique is shown to yield results in excellent agreement with the often used kinematic uniform boundary conditions (KUBC). It is also shown how these properties may be used for visualising the distribution in properties over a domain.

Contents

Contents	iv
List of Figures	x
List of Tables	xv
Nomenclature	xix
1 Introduction	1
2 Background	4
3 Micro-Architecture Generation	10
3.1 Literature Review	12
3.2 Surface Representations	17
3.2.1 Explicit Surfaces	17
3.2.2 Parametric Surfaces	18
3.2.3 Implicit Surfaces	20
3.3 The Marching Cubes Algorithm	24
3.4 Mesh Generation	25

CONTENTS

3.5	Method I: Binary Volume	26
3.5.1	Implementation	30
3.5.2	Performance	32
3.5.3	Remarks	33
3.5.4	Quantifying Reconstruction Accuracy	35
3.5.5	Results	36
3.6	Method II: Binary Volume With Smoothing	38
3.6.1	Rationale	39
3.6.2	Test Case	39
3.6.3	Gaussian Smoothing	40
3.6.4	Mean Smoothing	41
3.6.5	'Pre-Smoothing'	42
3.6.6	Discussion	43
3.7	Method III: Distance Functioned Volume	44
3.7.1	Distance Functions and Implicit Surfaces	47
3.7.2	Choosing <i>MaxDistance</i>	49
3.7.3	Algorithm	51
3.7.4	Summary	51
3.8	Volume Fraction Calibration	53
3.8.1	Calibration by Re-Iso-Surfacing	54
3.8.1.1	Examples	57
3.8.2	Calibration by Erosion/Dilation	58
3.9	Summary	59
4	Functionally Graded Structures	60

CONTENTS

4.1	Generation	61
4.1.1	Apparent Non-Linearity	62
4.2	Specific Density Variations	63
4.2.1	Example	66
4.3	Non-Specific Geometrical Variations	66
4.4	Hybrid Functions & Transition Cells	69
4.4.1	Implicit Transition Cells	70
4.4.2	Discussion	75
4.5	Non-Periodic Functions	76
4.6	Summary	79
5	Domain Conforming Micro-Architectures	80
5.1	Bone Scaffolds	81
5.2	Internal Architectures for CAD Models	85
5.2.1	Further Processing	99
5.3	Implementation	100
5.4	Summary	101
6	Analysis & Characterisation	102
6.1	Computational Modelling and Analysis	102
6.1.1	Background and Literature Review	102
6.1.1.1	Effective and Apparent Properties	104
6.1.2	Select Cell Characteristics	114
6.2	Mechanical Testing	117
6.2.1	Fabrication	117
6.2.1.1	Facet Overhang	118

CONTENTS

6.2.1.2	Use as Supporting Structures	121
6.2.2	Mechanical Testing – Internal Structures	125
6.2.2.1	Results	128
6.2.3	Discussion	128
6.3	Conclusions	131
7	Homogenisation	133
7.1	Concept Overview	133
7.2	2D	135
7.2.1	Constitutive Matrix Recovery from a 3 Noded Triangle Element	135
7.2.1.1	Linear Dependence in Tests	140
7.2.1.2	Least Squares Method	142
7.2.2	Worked Example	142
7.2.2.1	Forward Problem	143
7.2.2.2	Inverse Problem	145
7.2.3	Multi-Scale Triangle Elements	148
7.2.4	Multi-Scale Example	151
7.2.4.1	Homogeneous Case	152
7.2.4.2	Inhomogeneous Cases	154
7.3	3D	155
7.3.1	Constitutive Matrix Recovery from a 4 Noded Tetrahedral Element	155
7.3.2	Worked Example	159
7.3.3	Multi-Scale Tetrahedral Elements	162

CONTENTS

7.3.4	Multi-Scale Example	163
7.3.5	Sub-Volume From Image Data	164
7.3.6	Choice of Virtual Tests	167
7.3.6.1	Criteria For Tests	167
7.3.6.2	Equivalence of Tests	170
7.4	Validation	171
7.4.1	Homogeneous Sub-Volume	172
7.4.2	Comparison to KUBC Results	174
7.5	Approximate Models	174
7.5.1	Model Generation	178
7.5.2	Example	179
7.5.3	Property Visualisation	182
7.5.3.1	Out-of-Plane Properties	184
7.6	Applications	185
7.7	Comparison to Superelements	187
7.8	Conclusions	191
8	Conclusions	194
8.1	Summary	194
8.2	Research Contributions	197
9	Recommendations for Future Work	200
A	Unit Cell Implicit Functions	203
B	Internal Structures Patent	205

CONTENTS

C Internal Structure Wizard	240
D Internal Structures Testing – Experimental Setup	243
References	245

List of Figures

2.1	The granular microstructure in stainless steel	8
3.1	Interpreting intermediate voxel values and homogenised micro-architectures.	14
3.2	Plot of the implicit function $\cos(x) + \cos(y) + 1$	22
3.3	A selection of triply periodic level set surfaces	23
3.4	The 15 base cases for the marching cubes algorithm.	25
3.5	Binary volume representing the implicit function $\cos(x) + \cos(y) + \cos(z) = 0$ for $3 \times 3 \times 3$ cells at $z = 7$ px with a domain of $20 \times 20 \times 20$ px	29
3.6	The misalignment of the ideal surface and sampling points	30
3.7	Comparison of reconstructed surfaces from binary volumes	34
3.8	Accuracy of a binary surface reconstruction using $20 \times 20 \times 20$ px	38
3.9	Comparison of different smoothing algorithms on a binary volume	45
3.10	Voxelising a CAD model	48
3.11	Surface accuracy for the ‘distance-functioned’ volume	49
3.12	Volume fraction calibrated unit cells	57

LIST OF FIGURES

3.13 Unit cells generated to specific volume fractions using different techniques	58
4.1 The relationship between iso-surface value and volume fraction for selected implicit functions	63
4.2 Example usage of a relative density map	67
4.3 Lattice structure generated using a space-warping function	68
4.4 Joining unit cells of different topologies in 2D, without the need for a transition region	70
4.5 Joining implicitly defined cells of different topologies	72
4.6 Space-time blending of the Pinched Primitive and Schoen Gyroid unit cells	73
4.7 Transition cell for Pinched Primitive and Schoen Gyroid unit cells	74
4.8 Example uses of transition cells	75
4.9 Blending unit cells of different sizes over a large region	75
4.10 Periodic lattice structure generated from the union of three cylinders	77
4.11 Distance functioned unit cell and lattice structure	78
5.1 Domain conforming micro-architecture using Boolean intersection	82
5.2 The process of data acquisition and processing for mesh generation	83
5.3 Example scaffold conforming to defect site geometry	85
5.4 CAD model voxelised at 1 mm	87
5.5 Internal structure without a shell	89
5.6 Poor surface reconstruction of a CAD model and internal structure	90
5.7 Internal surface volume and reconstructed surface	91

LIST OF FIGURES

5.8	Cut-away comparison of the entirely image-based approach with the mixed approach	93
5.9	The process of generating an internal structure in a CAD model .	94
5.10	Defining the lattice regions in the knee implant model	95
5.11	Arrangement of implant parts. Top to bottom: Condylar implant, shim, base plate.	96
5.12	Base plate with lattice structure	97
5.13	Comparison between original parts (grey) and prototypes (blue) .	98
5.14	The effects of various kernel sizes on corner bevelling	99
5.15	Cavity closing using morphological opening	100
6.1	Physical interpretation of the Reuss and Voigt bounds. Different colours represent different phases or materials.	105
6.2	Permitted modes of deformation for various boundary conditions .	112
6.3	Convergence of Exx for the Schoen Gyroid with periodic boundary conditions at 60% volume fraction	115
6.4	Apparent Young's modulus of the Schoen Gyroid at various volume fractions	116
6.5	Overhanging facets for the Schoen Gyroid at 15% volume fraction.	119
6.6	Fabricated lattice structures	122
6.7	Fabricated supporting structures.	124
6.8	CAD model with supporting structures. Support is added to any facet with an overhang of less than 40 degrees.	126
6.9	Clipped view of the chosen geometries, both hollow and with an internal structure, for testing	127

LIST OF FIGURES

6.10 Maximum displacement of loaded test parts. A comparison of the finite element and physical model results.	129
7.1 A 3-node constant strain triangle element. Nodes are numbered anti-clockwise	136
7.2 Example 3-node triangle element	143
7.3 Macro triangle (bold) discretised using micro triangles	149
7.4 Example discretised macro element	152
7.5 Comparison of original (grey) mesh and deformed mesh (black) for each test case	153
7.6 4-node tetrahedral element. Nodes are labelled anti-clockwise when viewed from the last node.	156
7.7 Example tetrahedral element	160
7.8 Original macro element and volume cropped from a $25 \times 50 \times 25$ px volume	165
7.9 Cropped volume using distance functions	166
7.10 Example micro-architecture conforming to a tetrahedral macro element	166
7.11 Two macroelements of differing geometry subjected to identical virtual tests	170
7.12 Macro element (solid) within the bounding volume (dashed)	172
7.13 Convergence study for a homogeneous sub-volume	173
7.14 Comparison of KUBC results to the tetrahedral method for the Schoen Gyroid	175
7.15 Length-scales present in a multi-scale irregular domain	176

LIST OF FIGURES

7.16 Macro mesh generation steps	178	
7.17 Displacement of an approximate model of a cantilever lattice beam	180	
7.18 Visualising the variation of Young's modulus over a functionally graded structure	183	
7.19 Plots of different out-of-plane properties for the Schoen Gyroid at 15% volume fraction	186	
C.1 Introduction		240
C.2 Selection of CAD model		241
C.3 Shell parameters		241
C.4 Lattice parameters		242
C.5 Post-generation options		242
D.1 Loading of the box component.		243
D.2 Loading of the sphere component		244
D.3 Loading of the cylinder component		244

List of Tables

3.1	Implicit surface in/out convention	21
3.2	Comparison of runtime for the original and improve binary volume generation algorithms	33
3.3	The effect of increasing resolution on surface accuracy for the Schwarz Primitive	37
3.4	The effect of Gaussian smoothing on a binary volume	41
3.5	The effect of Mean smoothing on a binary volume	42
3.6	The effect of pre-smoothing on a binary volume	43
6.1	Orientations for minimising and maximising the total overhang area for various unit cells	120
6.2	SLM process parameters	121
6.3	Dimensions of the hollow components	125
6.4	Results from mechanical testing	128
7.1	Single DOF displacements and the involved constitutive components.	139
7.2	Coefficient matrix	140
7.3	Test case displacement vectors and corresponding nodal forces . .	144
7.4	Effective macro forces computed from micro node forces	154

LIST OF TABLES

7.5 Forces resulting from virtual tests	161
7.6 Comparison of different homogenised beam models	181

Nomenclature

E Young's modulus

G Shear modulus

K Bulk modulus

$K^{HS\pm}$ Hashin-Shtrikman bounds (bulk modulus)

N_i Element shape function for node i

Q_1, Q_2 Quality metrics

V Volume

V_S Solid volume

V_T Total volume

V_V Void volume

$[a, b)$ The set of real numbers which lie between a and b , excluding b .

$[a, b]$ The set of real numbers which lie between a and b inclusively.

Ω Domain of interest

LIST OF TABLES

Φ	Grading function
Θ	Spacing-warping function
\emptyset	The empty set
$\langle x \rangle$	The spatial average of x over a domain Ω . I.e. $\frac{1}{\Omega} \int_{\Omega} x d\Omega$
\mathbb{R}	The set of real numbers
\mathbb{Z}^+	The set of positive integers
\mathbf{C}	Elastic stiffness matrix
\mathbf{C}^{app}	Apparent elastic stiffness matrix
\mathbf{M}	Matrix (upper-case)
\mathbf{S}	Elastic compliance matrix
\mathbf{S}^{app}	Apparent elastic compliance matrix
\mathbf{g}	Voxel gradient
\mathbf{v}	Vector (lower-case)
\mathbf{v}^\star	Concatenation of a set of vectors. I.e. $[\mathbf{v}_1, \dots, \mathbf{v}_n]^T$
$\mu^{HS\pm}$	Hashin-Shtrikman bounds (shear modulus)
ν	Poisson's ratio
ϕ	Porosity
ρ	Density

LIST OF TABLES

σ	Stress tensor
σ_{SD}	Standard deviation
\tilde{E}	Macro strain tensor
$\tilde{\Sigma}$	Macro stress tensor
ε	Strain tensor

Chapter 1

Introduction

Porous lattice structures have a wide range of potential applications ranging from industry to medicine. Their unique properties, such as a high surface to volume ratio and high energy absorption, coupled with the ability to be manufactured to create custom materials, often make them highly desirable. Recent advances in prototyping and manufacturing techniques now allow these structures to be computationally designed and realised with a precise geometry, whereas traditional manufacturing methods rely on stochastic foaming processes. However, due to their complexity, these structures are often difficult to generate using existing CAD packages. Once an appropriate structure has been generated it is often required that some aspect of its properties be evaluated, such as its bulk stiffness or permeability. One possible method for doing so is to first fabricate the structure and physically test it, however, this approach tends to be expensive, particularly for multiple structures. More commonly, computational simulations are performed using finite element analysis. This in itself, however, introduces further complications. As the structures are often both complex and highly porous

(i.e. low volume fraction) the number of elements, and hence the computational resources required, to produce accurate models often make the simulations unfeasible. To overcome this, various approximations are often used, such as a reduced number of degrees of freedom, the use of beam elements, idealizing the structure, and homogenisation.

The objective of this research is to better address the complications in the generation and characterisation of porous lattice structures and to fully exploit the recent advances in additive manufacturing processes. This thesis will present advances in the generation of domain conforming periodic lattice structures and a novel approach to the homogenisation of linear elastic properties. More specifically this research encompasses the following activities:

1. To develop image-based algorithms for the efficient generation of periodic lattice structures, focusing on the ease of both volumetric and surface meshing.
2. To develop methods and algorithms for the generation of lattice structures with functional and arbitrary grading variations so that custom properties may be achieved.
3. To develop robust algorithms for the purpose of introducing internal lattice structures in CAD models.
4. To characterise the effective properties of periodic lattice structures through analytical models, computational simulations and mechanical testing.
5. To develop a method for generating approximate models of large multi-scale problems using a novel approach to homogenisation.

This thesis is structured as follows.

Chapter 2 presents a more detailed background into the applications of micro-architectures, their creation and evaluation.

Chapter 3 develops a number of new image-based algorithms for the purpose of generating micro-architectures from periodic implicit functions. A number of different algorithms are compared. Methods for generating graded structures with customised mechanical properties are developed in Chapter 4. These methods are then extended in Chapter 5 for micro-architectures conforming to a given domain or within a shell.

Chapter 6 reviews existing techniques for the evaluation of the mechanical properties of micro-architectures. This includes analytical models, computational simulations and physical testing. A number of lattice structures, generated using the methods developed in Chapters 3 to 5 are characterised using finite element analysis and mechanically tested. This chapter also looks at the fabrication of the structures and their possible use as supporting structures in additive manufacturing processes.

In Chapter 7 a novel approach to homogenisation is developed for the purpose of creating approximate models of large multi-scale problems. This new approach is shown to yield results comparable to well-established existing methods.

Chapter 8 presents conclusions.

Chapter 9 presents recommendations for future research.

Chapter 2

Background

The term micro-architectures, sometimes also referred to as micro-structures or meso-structures, can be considered to loosely refer to “small-scale structures”. While the terms themselves imply a length-scale they are often used informally to describe structures on a variety of length-scales, from micro to macro. They may also be used to describe a wide range of different types of structure. Such structures may be further classified as; porous (including both open and closed cell), multi-phase, fibrous, stochastic or regular/uniform, natural or synthetic. Example structures include, but are in no way limited to; honeycombs, cancellous bone, particle-reinforced composites and Voronoï mosaics.

Of most interest to this work are the micro-architectures which fall into the classification of ‘open cell porous’. Open porous structures are now manufactured from a range of materials including polymers, ceramics and metals are used in a number of different applications.

Open porous polymer foams can be found virtually everywhere, including in the home, due to their good energy absorption and relatively low cost. While

naturally occurring polymer foams, such as sponges, have been known for a long time, synthetic polymer foams have only been available since the 1950's (Eaves [2004]). These foams are often used in packaging and seating (both vehicle and home) as their mechanical properties make them very suitable for padding. They may also be found in other applications such as floatation devices, insulation, sound proofing and ear protection. More complex polymer foams, known as 'memory foams', are often used in bedding, particularly in hospitals, where they prevent pressure sores.

Open porous metal foams can be manufactured by replicating an existing open cell polyurethane foam. The polyurethane foam is used as a template in investment casting to create a negative mold (i.e. the porous space around the foam) from which it is then burned out (Queheillalt *et al.* [2004]). This leaves a mold from which metal foams can be cast. Open cell metal foams are used in a variety of applications including compact heat exchangers, where their material properties and high surface to volume ratio make them highly suitable. Their high stiffness, in comparison to other polymeric open cell foams, also makes them suitable for multi-functional applications. For example, when used in sandwich structures they can function both as a supporting structure and aid heat dissipation due to the ability for fluids to pass through the pores (Lu *et al.* [1998], Queheillalt *et al.* [2004]). Open cell metal structures are now also finding uses in medical applications. Some commercially available metal implants utilise a porous metal foam, with properties similar to that of bone, to encourage new bone ingrowth and ultimately aid osseointegration (Zim [2011]). Metal lattice structures may also be used as internal structures within implants to control or reduce their stiffness so that stress shielding effects can be reduced.

Open porous ceramic foams are used in a wide range of applications including fluid filters, catalyst supports, burners, gas diffusers and flame barriers (Jayasinghe & Edirisinghe [2002]). Open cell hydroxyapatite (a calcium phosphate ceramic) structures are also becoming more frequently used as tissue scaffolds, bone substitutes and to aid osseointegration of implants (Heise *et al.* [1990], Starly [2006]).

The structures described thus far are, for the most part, manufactured using various foaming techniques resulting in stochastic structures. However, with advances in additive layer manufacturing (ALM), and the technologies this encompasses, structures with specific geometries can be designed and fabricated. This allows for the computational generation and fabrication of structures with specific cell sizes and rib thicknesses. In particular this allows open cell structures with functionally graded properties to be realised. Another major advantage of this process is that the computational generation can itself include an optimisation process. This has recently been exploited for the purpose of creating optimal scaffold structures. Most notable is the work by Hollister *et al.* (Hollister *et al.* [2002], Hollister [2005], Hollister & Lin [2007]) where unit cells are optimised for both mechanical loading and mass transport. Additive layer manufacturing technologies can be used to fabricate these structures from a variety of different materials including polymers, ceramics and metals, some of which are biocompatible.

The computational generation of micro-architectures is a non-trivial task. Traditional CAD packages have been shown to be unsuitable for the creation of these complex structures, due to the potentially large number of Boolean operations required (Wang *et al.* [2005]). Many methods have been developed recently

to address this issue, some of which are application specific and others more generic. Voxel based methods are often used to model unit cells in optimisation processes due to their simple representation and the ease with which they can be modified (Hollister *et al.* [2002]). More recently implicit representations have been exploited to generate complex micro-architectures with functional gradings (Gabbrielli *et al.* [2008] and Pasko *et al.* [2010]). These implicit methods have been shown to be a highly flexible way of modelling micro-architectures. These methods, and ones similar (such as that in Chen [2007a]), have also been used to incorporate micro-architectures within a shell of an existing CAD model. These so-called *internal structures* are used to create lightweight components, act as heat transfer structures and aid in fabrication.

When dealing with micro-architectures, either those computationally generated or through foaming processes, it is often desirable to model these computationally to determine effective properties. For the case of those manufactured through foaming processes various imaging modalities (MRI, micro-CT, etc...) can be used to acquire the micro-architecture's geometry, which may then be meshed using well established methods. Similarly, image-based algorithms can be applied to generate volume and/or surface meshes for computationally generated lattice structures. This allows computational modelling such as finite element (FE) or computational fluid dynamic (CFD) simulations to be performed on the structures. Such simulations can be used to non-destructively evaluate a variety of properties of a structure without the costs involved with fabrication. These properties may be local effects, such as stress concentrations, or bulk properties, such as stiffness.

However, when dealing with multi-scale structures these simulations can soon

become highly impracticable. This is due to the size of the models required to sufficiently represent the geometry at both the macroscopic and microscopic scale. It is therefore often only feasible to model such structures at the macroscopic scale. For example, when simulating the mechanical behaviour of a stainless steel component the steel itself is modelled as a homogeneous material, despite being composed of a complex microstructure (as shown in Figure 2.1).

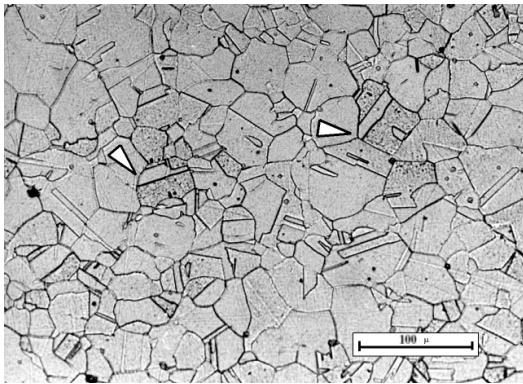


Figure 2.1: The granular microstructure in stainless steel (Source: Ste [2011])

The separation of length-scales present allows the steel to be modelled as a continuum. In the case of steel the effective homogeneous properties are well-established and are widely used in finite element simulations. Where there exists this clear separation of length-scales it is often the case that one or more length-scales can be replaced with an equivalent homogeneous material. The process of determining this equivalent material is known as *homogenisation*. Homogenisation can be performed computationally using a number of different methods. There are, however, a number of issues to address when attempting to determine a micro-architecture's effective properties, such as the boundary conditions to apply and the size (or representativeness) of the sample, both of which are well-studied. Homogenisation is useful not only for multi-scale modelling, but also for

providing feedback as to the bulk properties of a micro-architecture, which may in turn be used to visualise out-of-plane properties (Marmier *et al.* [2010]).

The objective of this research is to better address the complications in the generation and characterisation of porous micro-architectures and to fully exploit the recent advances in additive manufacturing. This research focuses on the development of image-based methods and algorithms for the efficient and robust generation of periodic and functionally graded micro-architectures. A novel approach to the homogenisation of linear elastic properties for large multi-scale problems is also developed.

Chapter 3

Micro-Architecture Generation

This chapter describes the computational generation of micro-architectures. It concentrates exclusively on micro-architectures generated from triply periodic implicit functions, extending the work in Gabbrielli *et al.* [2008]. Image-based algorithms have been developed to efficiently generate periodic lattice structures and represent them with accurate volumetric and surface meshing. A review of existing micro-architecture generation and surface representation techniques is first given to illustrate the advantages and disadvantages of each.

Implicitly defined micro-architectures have been chosen for further study due to the simplicity of their representation in comparison to other more complex methods. Notably these include the explicit boundary representations used in modern CAD (Computer Aided Design) packages. The mathematical equations which define the micro-architectures of interest also provide great flexibility. By introducing functional variations, structures with tailored properties can be generated. This work looks at the generation of functionally graded structures as well as methods for introducing more complex variations. The methods devel-

oped provide a good foundation for use in future optimisation schemes where parametrised models of density variations will be useful.

One of the main prerequisites for performing optimisation using physics based simulations, such as FEA or CFD, is the ability to robustly and efficiently generate volume meshes. The ability to perform various simulations on the micro-architectures also allows the mechanical and fluid properties of the structures to be fully characterised. Previous work (Jung & Torquato [2005]) has explored the fluid properties of triply periodic implicit surfaces. The mechanical properties of the structures are studied in Chapters 6 and 7.

While computational models have many advantages when dealing with the design and testing of micro-architectures, there will ultimately be a need to fabricate a final design to obtain a physical model. Due to their complexity, the models cannot be fabricated using traditional manufacturing methods and so we utilise AM (Additive Manufacturing) to realise the generated models. These manufacturing systems rely on the model's geometry being defined by a surface triangulation, using the standard STL ("Stereolithography") file format. To meet the meshing requirements an image-based approach is taken so that the advantages of image-based meshing can be exploited for both volume and surface meshes.

As in Gabbrielli *et al.* [2008], the structures of interest in this work are based upon the primitive, diamond and gyroid surfaces as well as Neovius' surface. Variations on these surfaces will also be created and used later in this thesis.

3.1 Literature Review

Recent advances in *additive manufacturing* (AM) processes, also known as *additive layer manufacturing* (ALM), *rapid manufacturing* (RM) and *rapid prototyping* (RP), have allowed for the creation of complex geometries to a relatively high level of precision. These manufacturing processes are particularly well-suited to fabricating computationally generated micro-architectures and have allowed for the recent development of methods for doing so.

As with the design of the vast majority of components that are to be manufactured, micro-architectures have previously been created using traditional commercial CAD packages. However these packages have proven to be unsuitable for potentially large complex micro-architectures due to the vast number of Boolean operations required, as shown in Wang *et al.* [2005]. To overcome the difficulties in using generic CAD packages Wang *et al.* [2005] developed a hybrid geometric modelling method for conformal truss structures. Their method was demonstrated to be able to create large triangulated surfaces of repeating unit cells, such as the tetrahedron and Kelvin Foam structures. By using open-ended cylinders for trusses and ‘sealing’ joins with spheres, repeatable unit cells were generated. This, however, relied on the use of a solid modeller (ACIS) to perform the Boolean operations on the trusses and spheres. While this does considerably reduce the number of Boolean operations required, the solid modelling stage was observed to make-up a large percentage of the total processing time. These methods were further developed in Chen [2006] to remove the need for a solid modeller by developing methods for handling the tessellation of truss-sphere intersections. Combined with the elegant algorithms for Boolean operations developed in Chen

[2007b], these methods were further extended for the purpose of creating internal structures within CAD models (Chen [2007a]). This work also allowed for the creation of internal structures with functional variations through the use of warping functions. The functions were used to displace truss' nodes such that specified regions increased in density while others decreased, without influencing the truss thickness. Although this does create regions with different stiffnesses, it has the disadvantage of causing some cells to become stretched, therefore changing their effective properties.

Voxel modelling is an alternative approach to the generation of micro-architectures and is often used in the creation of scaffold architectures. While high resolution images or volumes are normally required to sufficiently represent geometries using voxels, they have the advantage of being particularly straightforward to modify, particularly when using Boolean operations.

A relatively simple image-based approach to the generation of conforming scaffold architectures is presented in Starly [2006]. In this work Starly slices the bounding geometry, as defined using a CAD model, into a number of equally spaced binary images. By using Boolean operations on each slice a number of simple unit cells are then introduced into the geometry. The unit cells themselves are typically solid cubes with spherical or cuboidal voids, thus ensuring they remain stackable. This slice-based approach avoids the need to handle triangulated surfaces for the creation of an STL file. However, this is likely limited to 3D printing where image-based slices may be used. As with any purely voxel-based method, it also results in a poorly defined geometry at the boundaries.

Voxel-based methods are also widely used in topology optimisation. Most notable are the works of Bendsøe and Sigmund. Bendsøe & Sigmund [2003]

presents a review of past works based on the SIMP (*solid isotropic material with penalisation*) model for general topology optimisation. In his approach Bendsøe takes the topology problem as being one of optimal material distribution (Bendsøe [1989]). Rather than considering voxels as either solid or void, intermediate (or grey-scale) values are permitted and a density-stiffness function is used. In the case of the SIMP model:

$$E(\rho) = \rho^p E_0 \quad (3.1)$$

where E_0 the base material stiffness, $E(\rho)$ the stiffness for an element with a density ρ and p is the penalisation exponent, typically between 2 and 3. The penalisation exponent is used to discourage the use of grey-scale voxels in the optimisation process as it is often desirable that the final solution be a binary volume due to their obvious physical interpretation. However, as noted in Rodrigues *et al.* [2002], these intermediate values can be interpreted as being homogenised micro-architectures on a smaller length-scale. An example of this is show in Figure 3.1.

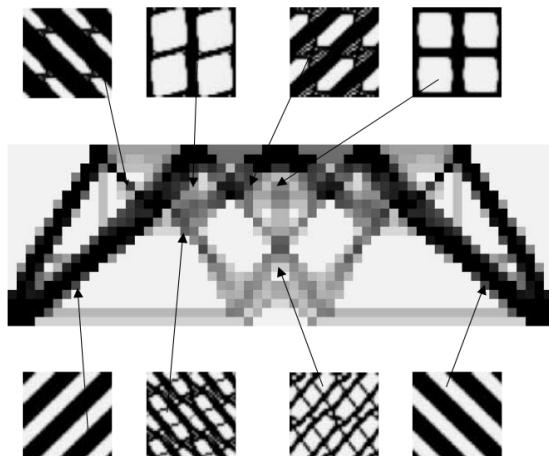


Figure 3.1: Interpreting intermediate voxel values and homogenised micro-architectures. (Source: Rodrigues *et al.* [2002])

These micro-architectures may then be realised by solving the so-called ‘inverse homogenisation’ problem (Sigmund [1994]). This problem has been addressed using the same optimisation methods with the SIMP model in Rodrigues *et al.* [2002]. A more general approach to generating optimal unit cells given a set of macroscopic stresses is developed in Guedes *et al.* [2003], which again uses the SIMP method.

Hollister & Lin [2007] have also used the SIMP method for the purpose of creating optimal unit cells for tissue scaffolds. However, in order to meet the biological requirements for scaffold structures, the unit cells need not only to have similar mechanical properties to bone, but also a similar porosity. To achieve this a two step optimisation process was developed. In the first step a number of iterations were performed using the SIMP model to optimise the elastic properties of the unit cell. The second stage then uses a small number of iterations of the ESO (*evolutionary structural optimisation*, Lin [2005], Steven *et al.* [2000]) algorithm to optimise the fluid phase. This process is repeated iteratively to ultimately yield unit cells with properties similar to that of bone.

Many more similar voxel based approaches have been taken for the optimisation of unit cells (e.g. Lin *et al.* [2004], Sanz-Herrera *et al.* [2009], Hollister [2005] and Adachi *et al.* [2006]). It has been shown that, through the use of these methods, mechanical properties close to the upper elastic limits can be achieved. This includes the creation of unit cells with customised properties, such as a negative Poisson’s ratio (Lin *et al.* [2004]).

Another less frequently used approach to the generation of micro-architectures is through the use of implicit functions. This is the approach taken in the works

by Gabbrielli (Gabbrielli *et al.* [2008] and Gabbrielli [2009]) and more recently by Pasko (Pasko *et al.* [2010] and Pasko *et al.* [2011]).

Gabbrielli uses a set of periodic implicit functions, such as the Schoen Gyroid (Schoen [1970]), to create porous micro-architectures to be used as bone substitutes. By introducing functional variations to the equations Gabbrielli was also able to create functionally graded micro-architectures. However, there were no methods given for precisely controlling the grading, such as the minimum and maximum volume fractions. Additionally, while there were methods for creating conforming micro-architectures they were limited to geometries with implicit representations, such as spheres.

The works by Pasko take a similar approach using periodic implicit functions (Pasko *et al.* [2010]). However the functions differ to those used by Gabbrielli as they are based on the implicit representation of rods. By taking the union of three rods, one along each axis, a simple grid structure was created. A simple blending function was employed to produce a filleting effect at the intersection of the rods. Pasko also demonstrated the creation of functional graded and warped structures. Furthermore, Pasko showed how, using the implicit representation of an object, internal structures can be introduced using Boolean operations on the implicit functions.

The works by both authors demonstrate the flexibility of using implicit representations for the purpose of generating micro-architectures with customised properties. In comparison to other techniques these methods provide a compact representation of the complex structures and, through the use of an appropriate iso-surfacing algorithm, a straightforward way of producing triangulated surfaces.

3.2 Surface Representations

When dealing with solid 3D objects it is often convenient to only model the object's boundaries using a mathematical representation of the surfaces. The choice of surface representation is particularly important for the computational modelling of 3D objects as each has its own advantages and disadvantages. These include the availability and complexity of operations that can be used to manipulate the surface (e.g. smoothing, Boolean operations) as well as the efficiency of the representation. Each representation also has an impact on how models are visualised and ultimately realised (e.g. via rapid manufacturing).

The most common representations can be classified as one of the following forms: explicit, parametric or implicit.

A brief overview of these representations follows – a more in-depth comparison of these forms can be found in Zheng [2008].

3.2.1 Explicit Surfaces

In explicit surface representations points which lie exactly on the surface are explicitly stored. The most common type of explicit surface is a mesh of polygons, typically these are triangles or quadrilaterals although others can be used. These polygons are often stored as an order list of vertex indices – the order being used to define the direction the polygon is facing (i.e. the surface normal).

For example a sphere can be defined by a surface triangulation composed of many triangles and a shared vertex list (Zheng [2008]):

$$G = \langle \mathbf{v}, \mathbf{f} \rangle \tag{3.2}$$

where \mathbf{v} is a set of vertices and \mathbf{f} is a set of triangle faces, each specified as a list of vertex indices: $\mathbf{f}_i = \{< i_1, i_2, i_3 > | i_j = 1, 2, \dots, N\}$.

Clearly the accuracy of this representation is dependent on the number of polygons used to discretise the surface. While increasing this number will produce smoother, more accurate surfaces, the inefficiency of storing points explicitly soon becomes notable. Despite this, explicit surfaces are widely used in computer graphics and game platforms as they are relatively easy to visualise.

Explicit surfaces also become difficult to manage for a variety of operations, most notably Boolean operations, as extra care must be taken to ensure the mesh topology remains ‘valid’. That is, the mesh should, under any operation, remain manifold so as to properly represent the boundary of a solid object. A manifold surface mesh is defined as one enclosing a finite volume, allowing any point to be unambiguously defined as either inside or outside of the surface. Therefore surface meshes with holes or gaps are considered non-manifold.

3.2.2 Parametric Surfaces

Unlike explicit surfaces, parametric surfaces do not store points on a surface. Instead, points on a parametric surface are expressed as a function of the parametric variables (u, v) , which can be generalised to lie on the unit square $[0, 1] \times [0, 1]$ (Zheng [2008]).

Non-Uniform Rational B-Spline (NURBS) surfaces are a form of parametric surface commonly used in CAD packages due to their compact representation, smooth surfaces and easy of manipulation. Other forms of parametric surfaces exist, such as Rational Gaussian (RaG) surfaces (Goshtasby [1993]) and Fourier

Shape Descriptions.

For some applications it is necessary to generate an explicit surface to approximate the parametric surface. This may be for visualisation or increasingly commonly for producing an STL (Stereolithography) file for rapid manufacturing – although methods for processing NURBS surfaces directly have been developed (Starly *et al.* [2005]). The conversion process to an explicit surface is known as polygonisation and is relatively straight-forward.

Parametric forms use control points to allow local finite control of surfaces (Zheng [2008]), providing accurate and intuitive ways to model objects. Complex objects can be modelled using multiple surfaces, or patches, joined together.

The parametric representation of a sphere is given below in Eq. 3.3. In comparison to the explicit form this is an extremely efficient representation as it accurately represents the surface with minimal storage.

$$\mathbf{P} = [x(u, v), y(u, v), z(u, v)] \quad (3.3)$$

where,

$$u \in [-r, r]$$

$$v \in [0, 2\pi]$$

$$x(u, v) = \sqrt{r^2 - u^2} \cos(v)$$

$$y(u, v) = \sqrt{r^2 - u^2} \sin(v)$$

$$z(u, v) = u$$

3.2.3 Implicit Surfaces

Implicit surfaces are define as an isosurface of some function f . In 3D the surface is defined by a set of points $p \in \mathbb{R}^3$ satisfying the equality:

$$f(x, y, z) = 0 \quad (3.4)$$

where $f : \mathbb{R}^3 \mapsto \mathbb{R}$.

As with parametric forms, implicit surfaces provide a compact representation for potentially complex surfaces. They also offer a number of advantages, notably their flexibility (as will be demonstrated later in this work) and well-defined Boolean operations. However, unlike parametric forms they offer little local shape control and manipulating them can be unintuitive.

Implicit forms also make visualisation and polygonisation challenging. Unlike both explicit and parametric forms points on the surface are not stored, nor are they expressed as a function of some other terms. The implicit function f provides a way of verifying if a point (x, y, z) lies on a given isosurface and some measure of distance from the point to the surface. Thus in order to create an approximate explicit surface from an implicit form, the function f must be sampled over some domain. Typically the function is sampled at regular intervals so that a volume image can be constructed. This volume is then isosurfaced using an algorithm such as the marching cubes (Lorensen & Cline [1987], described in §3.3) (or one of the volumetric extensions (Young *et al.* [2008])), marching tetrahedra (Treece *et al.* [1999]) or dual contouring methods (Ju *et al.* [2002]). The sampling of the implicit function can be performed in parallel and the marching cubes algorithm is an efficient method for creating an explicit surface.

The implicit formulation of a unit sphere is given below:

$$\begin{aligned} f(x, y, z) &\equiv x^2 + y^2 + z^2 - 1 \\ &= 0 \end{aligned} \tag{3.5}$$

In this instance the implicit form is not only more compact, but potentially more useful as the sign of the function can be used to designate points as either inside or outside the surface. For this purpose the following convention is adopted:

Condition	Interpretation
$f(x, y, z) = 0$	On surface
$f(x, y, z) < 0$	Inside
$f(x, y, z) > 0$	Outside

Table 3.1: Implicit surface in/out convention

The function can also be used to describe multiple surfaces by iso-surfacing the function at various values. In the case of the sphere, this results in multiple spheres of differing radii.

Implicit surfaces are commonly defined using polynomial expressions, for purposes such as surface fitting. These functions take the form:

$$f(x, y, z) = \sum_{0 \leq i, j, k, i+j+k \leq n} a_{ijk} x^i y^j z^k \tag{3.6}$$

However, the implicit functions of interest to this work are the set of infinitely periodic surfaces, the most notable of which are those discovered by Schoen [1970] and Schwarz [1890]. In addition to being infinitely periodic these surfaces are also approximations of minimal surfaces, that is, the surfaces have a mean curvature

of zero. Using a combination of trigonometric functions in the form given in Eq 3.7 a number of periodic surfaces can be generated (Gabbrielli *et al.* [2008]).

$$\sum_{i=1}^3 \prod_{j=1}^n \cos(x_i) + k = 0 \quad (3.7)$$

The simplest triply periodic (or dual periodic in 2D) function in this form is the Schwarz Primitive, Figure 3.2 shows a plot of this function.

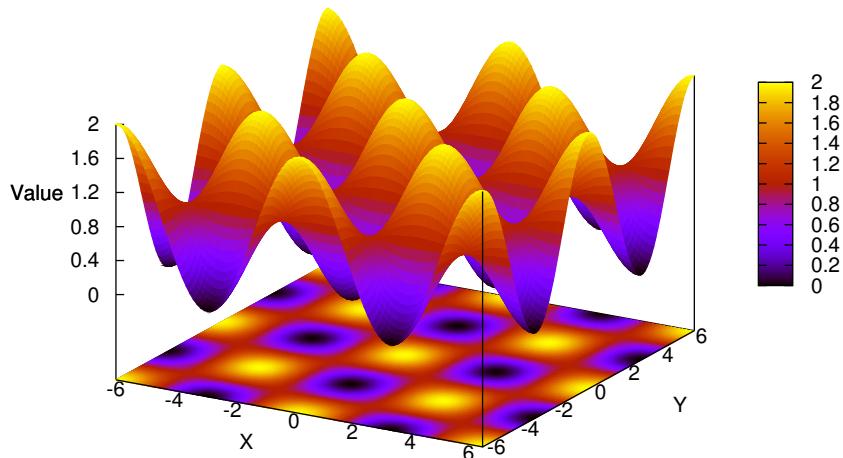


Figure 3.2: Plot of the implicit function $\cos(x) + \cos(y) + 1$

Figure 3.3 shows a selection of triply periodic unit cells, including those from Schoen [1970], Schwarz [1890] and The Scientific Graphics Project ([Sci, 2010, The Scientific Graphics Project]). The implicit functions describing these surfaces may be found in Appendix A.

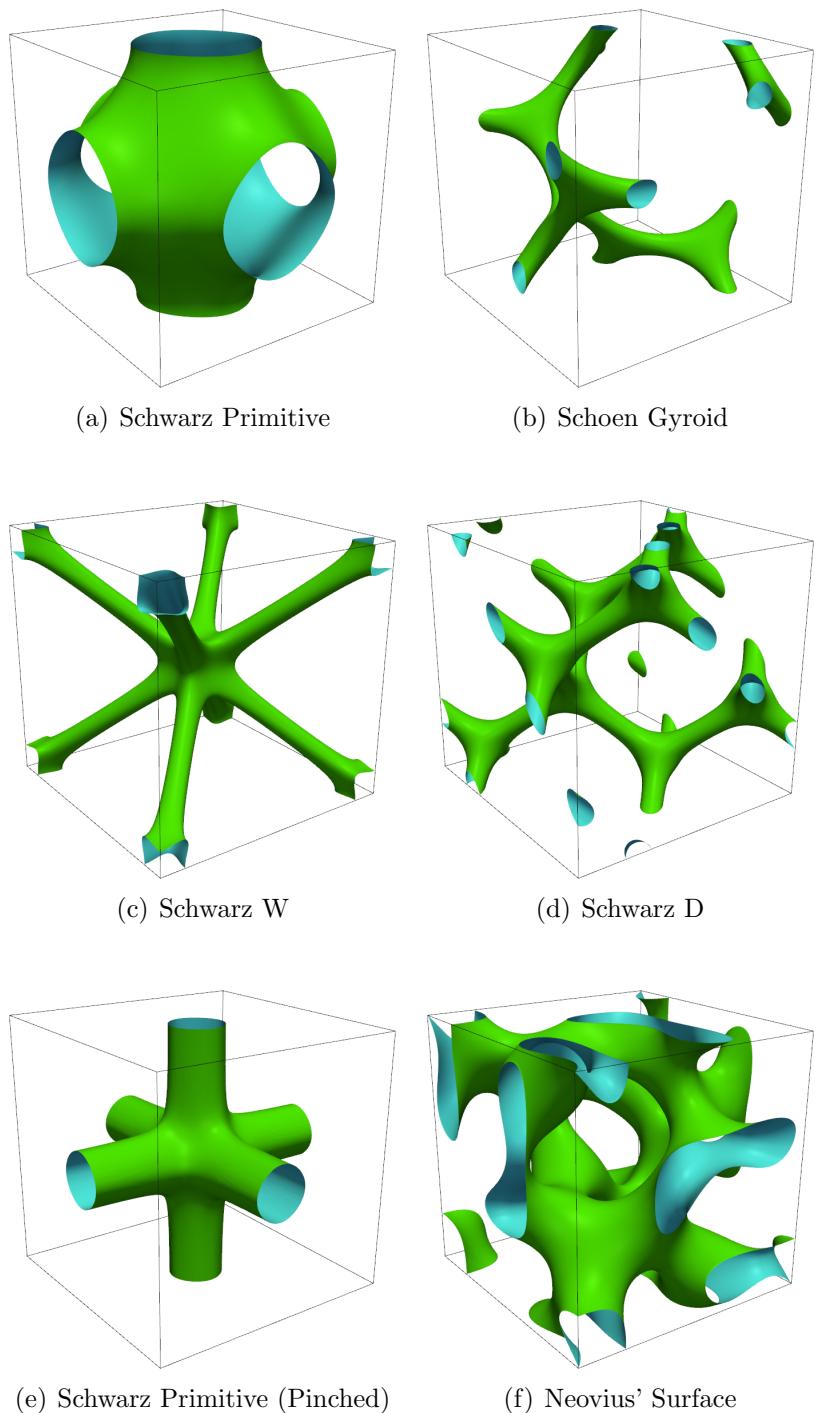


Figure 3.3: A selection of triply periodic level set surfaces

3.3 The Marching Cubes Algorithm

Implicit representations are generally more difficult to visualise, as well as realise, than explicit forms. Due to this fact, it is often desirable to produce an explicit representation from a given implicit representation, such as an implicit function or image data. One popular algorithm for doing so is the *Marching cube algorithm* by Lorensen & Cline [1987]. The marching cubes algorithm is a sequential-traversal method for generating a polygonal mesh of an iso-surface from a regular scalar volumetric data set. The algorithm was originally developed for the efficient visualisation of medical data, such as CT and MRI scan data, but has since been applied to a wide range of fields (Newman & Yi [2006]). Since it was first published in 1987 a number of corrections and variations to the algorithm have been made by other authors. These include modifications to remove the ambiguities in the original algorithm and to create multi-part conforming iso-surfaces. A survey of these algorithms is presented in Newman & Yi [2006]. The following described the algorithm as originally described in Lorensen & Cline [1987].

The marching cubes algorithm operates by sequentially traversing its input data set and considering the “cubes” formed by adjacent voxels (or sample points). The eight voxel centres each become a vertex of a cube. Depending on how the iso-surface intersects a cube one of a set number of pre-defined intersection topologies is chosen to form the polygonal mesh at that point. The intersection of the iso-surface with each cube is determined by the scalar values at each of its vertices. Vertices are marked as either inside or outside of the iso-surface if their value is below or above the iso-value. As there are eight ver-

tices this leads to $2^8 = 256$ possible intersection cases. However, by taking into consideration rotational and reflective symmetries this can be reduced to 15 base cases, as shown in Figure 3.4.

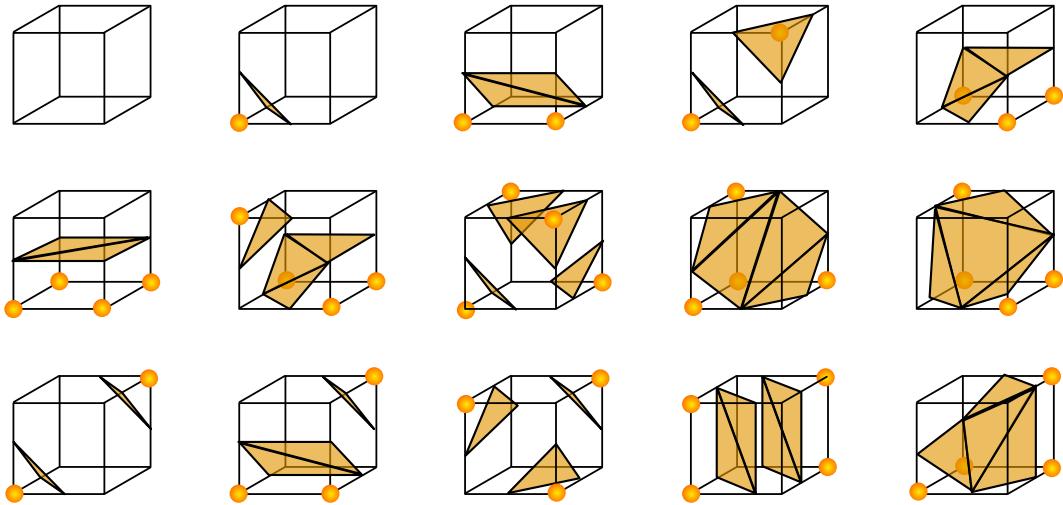


Figure 3.4: The 15 base cases for the marching cubes algorithm. Image source: http://en.wikipedia.org/wiki/Marching_cubes.

To improve the accuracy of the iso-surface reconstruction the iso-surface-edge intersect locations (i.e. the polygon vertices), are approximated using linear interpolation of the edge vertices.

3.4 Mesh Generation

Accurate and robust mesh generation is an important step towards an optimisation process where many lattice structures will be generated and evaluated. The work presented in this section looks at methods for generating image volumes representing the implicit functions so that both volume and surface meshes can be constructed. An entirely image-based approach is taken to exploit the advantages of image-based meshing.

Methods previously used in Gabbrielli *et al.* [2008] relied upon the generation of a floating-point volume which was then iso-surfaced (as implemented in [K3d, 2010, K3dSurf]). While this method is straight-forward, simply requiring that the function be sampled at regular intervals, it becomes difficult to generate a volume mesh and integrate with other image data.

To overcome these difficulties we require that the generated volumes' data-type matches that used by ⁺ScanFE. In the C programming language this is `unsigned char`, an 8 bit integer. By using this data-type the generated volumes can easily be combined with data from other sources, such as medical imaging devices and meshed with ⁺ScanFE.

3.5 Method I: Binary Volume

The most straight-forward translation to image-space that can be made from an implicit function is the generation of a binary volume. By evaluating the function, f , over a range of values voxels can be determined to be either inside or outside and their value set accordingly.

Before evaluating the function it is important that a high enough resolution is chosen for the binary volume in order to sufficiently represent the unit cells.

Assuming the function we wish to evaluate is periodic over 2π (i.e. Eq. 3.8 holds true), the range of values over which the function is to be evaluated is determined by the desired number of unit cells in each direction.

$$f(x', y', z') \equiv f(2\pi n_x + x', 2\pi n_y + y', 2\pi n_z + z'), \quad n_x, n_y, n_z \in \mathbb{Z}^+ \quad (3.8)$$

where n_x, n_y, n_z are the number of unit cells along each axis in the generated volume. Given this, we now have defined a ‘function-space’ coordinate system in which the ideal representation of the lattice structure exists. We also have a voxel-space coordinate system which is used to define the target volume.

In order to evaluate the function at a given voxel we must define a mapping between the discrete voxel-space, measured in pixels, and the continuous function-space (unit-less) coordinate systems. This mapping (Eq. 3.9) rescales the voxel-space coordinates x, y, z to the corresponding position in the function-space, x', y', z' .

$$m_i : \mathbb{Z}^+ \mapsto \mathbb{R}^+ \quad (3.9)$$

A point (x, y, z) in voxel-space can now be mapped to the corresponding point in the function-space, (x', y', z') using a linear rescaling (Eq. 3.10). To simplify the mapping we choose $x'_{\min} = z'_{\min} = y'_{\min} = 0$ since we also have $x_{\min} = z_{\min} = y_{\min} = 0$.

$$\begin{aligned} (x', y', z') &= (m_x(x), m_y(y), m_z(z)) \\ &= \left(\frac{x}{x_{\max}} \times 2\pi n_x, \frac{y}{y_{\max}} \times 2\pi n_y, \frac{z}{z_{\max}} \times 2\pi n_z \right) \end{aligned} \quad (3.10)$$

By iterating over the target volume and mapping each point to the function-space the following rule can be applied to generate the binary representation of the implicit function.

$$V_{x,y,z} = \begin{cases} V^{\min} & f(x', y', z') > 0, \\ V^{\max} & f(x', y', z') \leq 0 \end{cases} \quad (3.11)$$

where $V_{x,y,z}$ is the voxel value at (x, y, z) , and V^{\max} , V^{\min} the maximum and minimum possible voxel values respectfully. In practice the `unsigned char` (8 bit) data-type is often used to store pixel values, leading to the following:

$$\begin{aligned} V^{\max} &= 2^8 - 1 = 255 \\ V^{\min} &= 0 \end{aligned} \tag{3.12}$$

Algorithm 1 describes the process of generating the binary volume. Two additional functions, `GetCurrentPosition()` and `MoveToNextVoxel()`, are used to query the current position in a volume and to move on to an unvisited voxel. The order in which voxels are visited does not affect the algorithm. $x_{\max}, y_{\max}, z_{\max}$ define the size of the target volume in voxels.

Algorithm 1: Binary volume generation

Input: $f, n_x, n_y, n_z, x_{\max}, y_{\max}, z_{\max}$
Output: BinaryVolume
while not all voxels visited **do**
 $(x, y, z) \leftarrow \text{GetCurrentPosition}(\text{BinaryVolume})$
 $(x', y', z') \leftarrow \left(\frac{x}{x_{\max}} \times 2\pi n_x, \frac{y}{y_{\max}} \times 2\pi n_y, \frac{z}{z_{\max}} \times 2\pi n_z \right)$
 if $f(x', y', z') \leq 0$ **then**
 | BinaryVolume[x, y, z] $\leftarrow 255$
 else
 | BinaryVolume[x, y, z] $\leftarrow 0$
 end
 MoveToNextVoxel(BinaryVolume)
end

The efficiency of this algorithm can be greatly improved by exploiting the equality in Equation 3.8. Currently the function f is evaluated for every voxel in the target volume, a computationally expensive process. From Equation 3.8 we know that the function need only be evaluated over the range required to

generate a single unit cell; $x', y', z' \in [0 : 2\pi]$. Any position in the target volume has an effective and equivalent position within this range. By generating a volume for a single unit cell and repeating it throughout the target volume the function will be evaluated a minimal number of times. However, to accommodate this improvement the way in which the target volume is specified must be changed. Currently the algorithm takes as an input the dimensions, in pixels, of the target volume $(x_{\max}, y_{\max}, z_{\max})$ and fits the generated structure within this domain. For cases where the number of cells does not fit wholly into the domain (e.g. $x_{\max} \bmod n_x \neq 0$) there no longer exists a unique unit cell. The effect is most pronounced at low resolutions where single voxel changes can cause a notable change in geometry as shown in Figure 3.5.

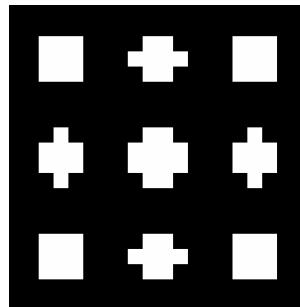


Figure 3.5: Binary volume representing the implicit function $\cos(x) + \cos(y) + \cos(z) = 0$ for $3 \times 3 \times 3$ cells at $z = 7$ px with a domain of $20 \times 20 \times 20$ px

This effect is caused by the discretisation of the space into voxels. Since the ideal number of voxels per cell is non-integer the alignment between the sampling points and the function's iso-surface becomes inconsistent. With voxels being considered either inside or outside a small misalignment can change the representation of some cells as shown in Figure 3.6.

The improved algorithm relies on the existence of a repeatable unit cell in

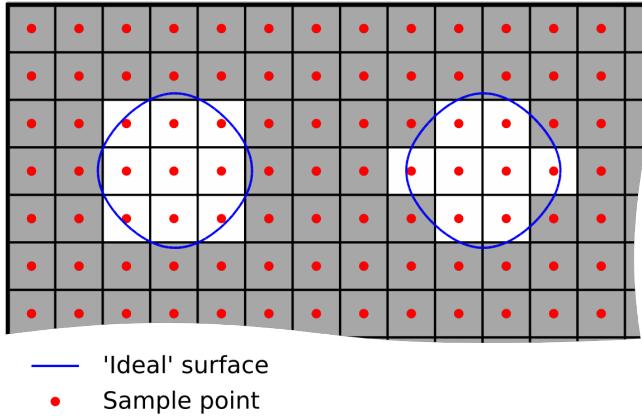


Figure 3.6: The misalignment of the ideal surface and sampling points

order to improve efficiency. As shown, when the size of the target volume is specified this cannot be guaranteed. The solution to this problem is to determine a suitable size for the target volume by specifying the number of cells in each direction as well as the size of a single cell. Given this, the size of the target volume becomes trivial to compute. The steps remaining in the process are simply to generate a single cell and repeat it throughout the target volume. Algorithm 2 details the entire process.

3.5.1 Implementation

Both algorithms were implemented in C++ using the Insight Toolkit (ITK) [2009]. ITK is an open-source, cross-platform image processing framework primarily designed for image segmentation and registration. The framework is heavily templated making it easy to handle images of different dimensionality and data-types. ITK's main strength comes from its pipeline-based architecture. The majority of components within the framework can be classified as either a source, a filter or an output. Pipelines are constructed by joining a number of components

Algorithm 2: Improved binary volume generation

Input: f, n_x, n_y, n_z, α
Output: BinaryVolume

```
BinaryVolume ← GetNewVolume( $\alpha n_x, \alpha n_y, \alpha n_z$ )
CellVolume ← GetNewVolume( $n_x, n_y, n_z$ )
/* Generate the volume for a unit cell */  

while not all voxels visited do
    |  $(x, y, z) \leftarrow \text{GetCurrentPosition}(\text{CellVolume})$ 
    |  $(x', y', z') \leftarrow \left( \frac{x}{n_x} \times 2\pi, \frac{y}{n_y} \times 2\pi, \frac{z}{n_z} \times 2\pi \right)$ 
    | if  $f(x', y', z') \leq 0$  then
    |   | CellVolume[ $x, y, z$ ] ← 255
    | else
    |   | CellVolume[ $x, y, z$ ] ← 0
    | end
    | MoveToNextVoxel(CellVolume)
end  

/* Repeat the cell throughout the volume */  

for  $p=1$  to  $n_x$  do
    | for  $r=1$  to  $n_y$  do
        |   | for  $s=1$  to  $n_z$  do
            |       | while not all voxels visted do
            |           |   |  $(x, y, z) \leftarrow \text{GetCurrentPosition}(\text{BinaryVolume})$ 
            |           |   |  $\beta \leftarrow \text{CellVolume}[x + \alpha n_x p, y + \alpha n_y r, z + \alpha n_z s]$ 
            |           |   | BinaryVolume[ $x, y, z$ ] ←  $\beta$ 
            |           |   | MoveToNextVoxel(CellVolume)
            |       | end
            |   | end
        |   | end
    | end
```

together; usually a source, a number of filters and an output.

As well as providing the framework for multiple filters, ITK provides a large number of image processing filters, file writers and segmentation and registration algorithms.

Algorithm 1 This algorithm was implemented as a single-threaded source, inheriting from `itk::ImageSource`. The class is templated over the pixel data-type and dimensional allowing 2D images to also be generated.

Algorithm 2 Given the nature of this algorithm it was possible to re-use the source filter from the implementation of Algorithm 1 to generate the single unit cell. The remainder of the algorithm was implemented using ITK’s image iterators which provide methods to iterate through an image without needing to take into consideration its dimensionality.

3.5.2 Performance

To measure the performance of both algorithms a test was devised such that the resulting generated volume for both algorithms was the same number of voxels. Each test involved generating the volume 100 times to ensure the times measured represent the actual runtime of each algorithm. Each test was run 3 times to produce an average time. For the purpose of these tests the generated volume was not saved to disk.

The parameters for both algorithms were as follows:

- **Unit cells:** $20 \times 20 \times 20$
- **Target volume:** $200 \times 200 \times 200$ px

The implicit function used is shown in Eq. 3.13 which describes the Schwarz Primitive.

$$f(x, y, z) \equiv \cos(x) + \cos(y) + \cos(z) = 0 \quad (3.13)$$

Table 3.2 shows the performance increase achieved by using the improved algorithm over the original. By computing the representing volume for only a single unit cell the improved algorithm was able to generate the volume in approximately one tenth of the time of the original. The tests were performed on a PC with an Intel Core 2 Duo 2.66 GHz CPU and 8 GB of RAM running Ubuntu Linux 10.04.

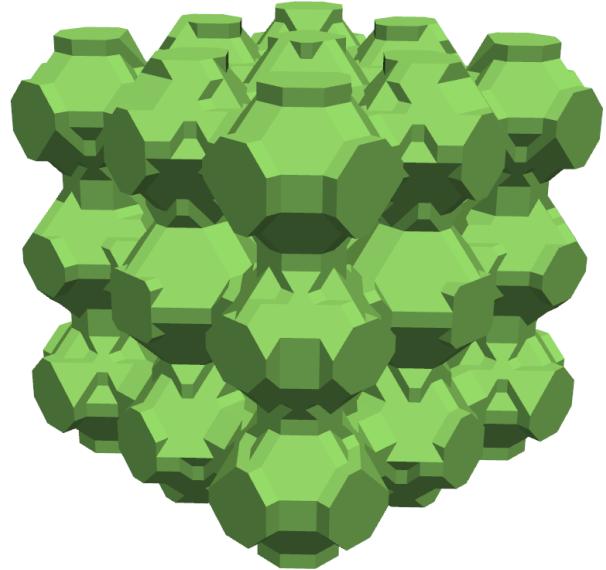
Run	Original (s)	Improved (s)
1	145.02	13.69
2	144.17	14.00
3	145.07	13.85
Avg.	144.75	13.85

Table 3.2: Comparison of runtime for the original and improve binary volume generation algorithms

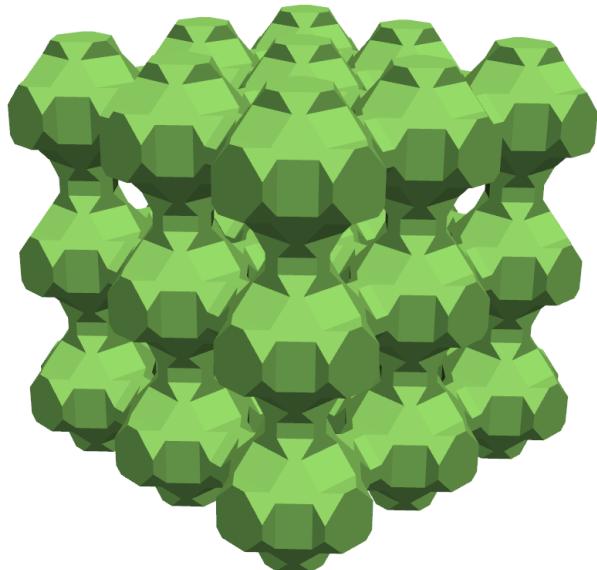
3.5.3 Remarks

The algorithm described in Algorithm 2 provides a computationally efficient method of generating a binary volume representing an implicit function. Notably, it ensures each cell in the generated volume has an identical representation, leading to a consistent surface reconstruction. This is clearly shown in Figure 3.7.

However, despite its efficiency we must consider the accuracy of the reconstructed surface. By this we mean how far the reconstructed surface deviates from the ‘ideal’ surface, as defined by the implicit function. Based purely on



(a) Original algorithm



(b) Improved

Figure 3.7: Comparison of reconstructed surfaces from binary volumes

empirical measurements, it is clear that the binary volumes lead to a very poor surface reconstruction. The surfaces visible in Figure 3.7 lack the curvature seen in Figure 3.3(a), instead being composed of regions of large facets with the same orientation.

3.5.4 Quantifying Reconstruction Accuracy

One of the advantages of using an implicit representation is that it not only defines where the surface of a cell is but also provides us with additional information about any point in the domain. For example, where $f(x, y, z) \neq 0$ we can determine if the point (x, y, z) is inside or outside (or ‘solid or void’) by the sign of the value. More importantly the value itself can be used as a measure of distance from the surface. This can then be exploited as a method for measuring the accuracy of a surface reconstruction.

We know that for an ideal surface reconstruction each point on the triangulated surface should have a value of 0 when evaluated with the implicit function. Points which do not lie on the ideal surface will have a non-zero value, increasing in magnitude with distance. Thus, for a point p on the triangulated surface we can say that:

$$\text{Quality}(p) \equiv |f(p)| \quad (3.14)$$

To produce a single value representing the accuracy of the entire tessellation one of two possible methods could be used. First, the average value of the quality of all points could be taken (Eq. 3.15). This would provide a measure of how far the reconstructed surface deviates from the ideal. However, this may not necessarily represent how close the morphology of the reconstructed surface matches

the ideal. For example, if we were to take an ideal surface representation and erode (or contract) it by a small amount then the average point quality would be poor. Yet we may still wish to consider this contracted surface as being a reasonably accurate representation, since, based on visual inspection, it will have the same morphology as the ideal surface. For this the standard deviation of point quality is likely to provide a more meaningful measure (Eq. 3.16).

$$Q_1 = \frac{\sum_{i=0}^N \text{Quality}(p_i)}{N} \quad (3.15)$$

$$Q_2 = \sqrt{\frac{1}{N} \sum_{i=0}^N (\text{Quality}(p_i) - Q_1)^2} \quad (3.16)$$

As both Q_1 and Q_2 provide different measures of surface accuracy both should be considered independently when measuring the accuracy of a surface reconstruction. Lower values of Q_1 and Q_2 indicate a more accurate reconstruction with $Q_1 = 0$ and $Q_2 = 0$ for a perfect reconstruction.

3.5.5 Results

Using the binary volume algorithm (Algorithm 2) a single cell of the Schwarz Primitive was generated within a $20 \times 20 \times 20$ px volume. To generate the reconstructed surface the marching cubes algorithm was used without any form of capping. This is important to ensure accurate results since points on the caps (which are not part of the ideal surface) will have spuriously poor qualities. To ensure a suitable number of points were evaluated on the surface, and that points lying on facets were also considered, a linear subdivision filter was applied a

number of times to increase the number of points in the tessellation.

For this surface $Q_1 = 0.0569$ and $Q_2 = 0.0323$. Figure 3.8 shows a plot of the point qualities on the surface. The blue lines visible on the surface (which are a result of the colour mapping) indicate where the surface intersects the ideal surface.

To increase the accuracy of the surface reconstruction a typical technique used in imaging is to increase the number of sampling points and hence the resolution of the volume. Table 3.3 shows the effect of increasing the target volume’s resolution on the quality metrics Q_1 and Q_2 .

Resolution	Q_1	Q_2	Elements
10	0.1669	0.0915	2,396
20	0.0569	0.0323	14,276
30	0.0379	0.0221	38,208
40	0.0307	0.0183	77,100
50	0.0225	0.0139	133,984
100	0.0115	0.0072	~ 800,000

Table 3.3: The effect of increasing resolution on surface accuracy for the Schwarz Primitive

As would be expected, increasing the volume’s resolution decreases the values of both Q_1 and Q_2 . However, the increased surface accuracy comes with a significant increase in the number of elements once the volume is meshed. Increasing the resolution beyond $20 \times 20 \times 20$ px results in a small increase in surface accuracy for a disproportionately large increase in the number of elements. To generate even a moderately sized lattice structure with a ‘good’ surface representation using a binary volume would require a large number of elements. The resulting mesh would therefore have a large number of degrees and freedom and any finite element simulations would likely require excessive amounts of computing

resources.

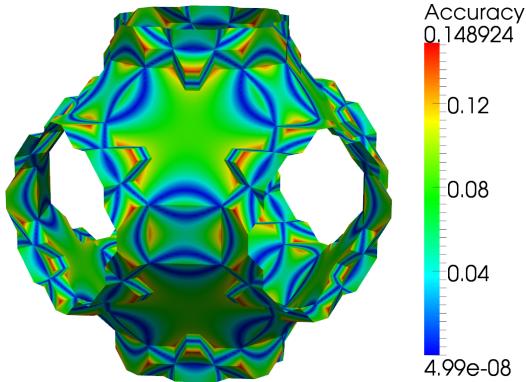


Figure 3.8: Accuracy of a binary surface reconstruction using $20 \times 20 \times 20$ px

3.6 Method II: Binary Volume With Smoothing

In the previous section (§3.5) a basic binary volume algorithm was developed which was able to efficiently generate a volume representing an implicit function. However, as was shown, the surface reconstruction from a binary volume lead to a poor final representation of the function. By using only binary data the interpolation points generated by the marching cubes algorithm will always be exactly half-way between neighbouring voxels, and unlikely to lie on the ideal surface.

A more appropriate method for generating these volumes would be to consider the partial volume effect when greyscale values are used. Partial volume information is recorded in various imaging techniques such as Computed Tomography (CT) where voxels intersect the boundary of the scanned object. In these cases, where a voxel is clearly neither fully in nor out, its value reflects that it is only partially inside the object. This is represented in the volume by a greyscale value.

These greyscale values, when used with the marching cubes algorithm, cause the interpolation points to vary depending on how much of the voxel is determined to be inside the object. By varying the interpolation points in this manner the marching cubes algorithm can produce smoother, more accurate surface reconstructions.

An 8-bit (1 byte) volume allows for 256 interpolation points between voxels, whereas a binary volume only allows for 1. Increasing the voxel's storage to 16-bit (2 bytes) increases this number to 65,536.

3.6.1 Rationale

This section looks briefly at methods for artificially introducing partial volume information to a binary volume by utilising existing smoothing algorithms. The rationale behind this method is that by smoothing a binary volume we know that greyscale values will be introduced which will greatly improve the smoothness of the final surface. By using the algorithm developed in §3.5 and combining it with various image smoothing filters available in ITK we are able to easily test the effectiveness of different algorithms for this purpose.

3.6.2 Test Case

A binary volume of $20 \times 20 \times 20$ pixels, representing a single unit cell of the Schwarz Primitive, was generated to test each smoothing algorithm. Each algorithm was then applied to an individual volume and the surface reconstructed using the marching cubes algorithm. The accuracy of the reconstruction was again measured by the metrics Q_1 and Q_2 . For comparison we know that $Q_1 = 0.0569$ and

$Q_2 = 0.0323$ when no smoothing is applied.

3.6.3 Gaussian Smoothing

Gaussian smoothing is a widely used smoothing algorithm in image processing. The algorithm updates voxel values by taking a weighted average of neighbouring values in order to produce the smoothing effect. The size of the neighbourhood and the voxel weightings are determined by a convolution matrix seeded with a Gaussian distribution. The amount of smoothing performed is specified by σ_{SD} , the standard deviation of the Gaussian distribution. While in theory the Gaussian function is non-zero for each point in the image, values beyond $3\sigma_{SD}$ are small enough to ignore.

An interesting property of the Gaussian smoothing is that it is linearly separable. That is, the process can be applied with a one-dimensional matrix separately in the x, y and z axis to achieve the same effect as applying a three-dimensional matrix once. By doing so the computational efficiency of the algorithm can be improved.

Results

Kernel Size (px)	σ_{SD}	Q_1	Q_2
0.5	0.165347	0.0499	0.02746
0.75	0.248021	0.0378	0.02298
1	0.330694	0.0336	0.02296
1.25	0.413368	0.0330	0.02352
1.5	0.496041	0.0353	0.02276
1.75	0.578715	0.0407	0.02258
2	0.578715	0.0478	0.02245
1.2	0.578715	0.0326	0.02331

Table 3.4: The effect of Gaussian smoothing on a binary volume

3.6.4 Mean Smoothing

As with Gaussian smoothing, mean smoothing is also based on averaging neighbouring voxel values within a given region. Unlike Gaussian smoothing however, neighbouring values are prescribed a weighting equal to $\frac{1}{n}$ for a neighbourhood of n voxels. Hence, as the name suggests, voxel values are replaced with the mean value from the neighbourhood. A notable disadvantage of this filter compared to Gaussian smoothing is that the size of the neighbourhood must be an integer number of pixels. This constraint can mean that in some cases, particularly with low resolution images, one must settle for either too much or too little smoothing. Whereas the value of σ_{SD} in Gaussian smoothing provides fine control over the amount of smoothing.

Results

Size (px)	Q_1	Q_2
1	0.0281	0.0225
2	0.0266	0.0182
3	0.0427	0.0216
4	0.0681	0.0426

Table 3.5: The effect of Mean smoothing on a binary volume

3.6.5 ‘Pre-Smoothing’

Up to this point the smoothing algorithms which have been examined have been relatively basic algorithms. Given the same parameters they perform exactly the same operations regardless of the volume they are being applied to. This lack of adaptivity means that the algorithms cannot ensure that they are topology and volume preserving. To address these issues an anti-aliasing algorithm was developed for smoothing volumes prior to meshing in the commercial software package, ScanIP. The algorithm was originally implemented in such a way that it could be used as a “black box” component in the meshing pipeline. By preserving both topology and volume the user had no need to interact with the filter. The algorithm itself remains proprietary and unpublished.

Results

For each of the tests performed the *part-change* feature of the anti-aliasing filter, which allows voxels to change between solid and void in order to improve smoothing, was disabled.

Iterations	Q_1	Q_2
5	0.0493	0.02658
10	0.0352	0.02276
20	0.0324	0.02051
40	0.0321	0.01762
80	0.0270	0.01650
160	0.0227	0.01591

Table 3.6: The effect of pre-smoothing on a binary volume

3.6.6 Discussion

The three smoothing algorithms tested have clearly shown that the marching cubes algorithm benefits greatly from greyscale image data. Even relatively small amounts of smoothing produce much smoother surfaces in comparison to the binary volume.

Despite not taking into consideration the contents of the volume, the tests showed the two ‘simple’ (Gaussian and Mean) smoothing algorithms were actually able to increase the accuracy of the reconstruction. However, once the size of the convolution matrix (in both cases) reached approximately 2 pixels the de-featuring nature of the algorithms becomes apparent with the values of Q_1 and Q_2 steadily increasing. In contrast to this, we see the pre-smoothing filter continues to decrease these values as the amount of smoothing is increased to the point where it is producing much more accurate representations.

The surfaces in Figure 3.9 show the best results achieved with each smoothing algorithm. We can see that the two non-volume preserving algorithms share the same worst quality point, shown in red on Figures 3.9(a) and 3.9(b), where the surface has contracted due to the smoothing. Many smoothing techniques tend to result in loss of volume, reflected in this case by an increase in Q_1 . Interestingly,

the anti-aliasing filter produced the most accurate results in this same region.

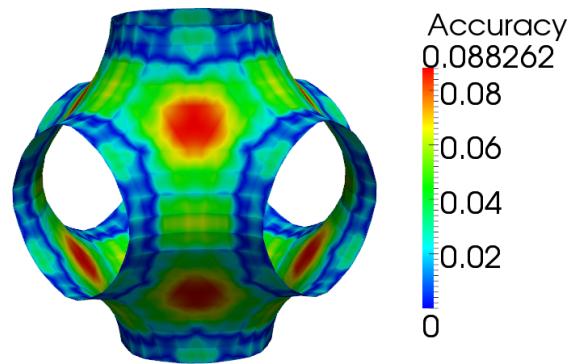
Overall the anti-aliasing filter produced the most accurate representation. By considering features in the image and preserving its volume the algorithm's results were surprisingly accurate given the input image. Not surprisingly, however, is the fact that this accuracy does come at a cost. The complexity of this filter means that it is much more computationally demanding than either the Gaussian or Mean filters. The filter required a large number of iterations (compared to the recommended 20 iterations ([Sca, 2011, ScanIP Manual])) before the values of Q_1 and Q_2 dropped below those for the Mean filter with a neighbourhood size of 2 pixels.

In light of these results, applying the Mean filter (2 pixels) to a binary volume seems a good trade-off of accuracy and computational requirements.

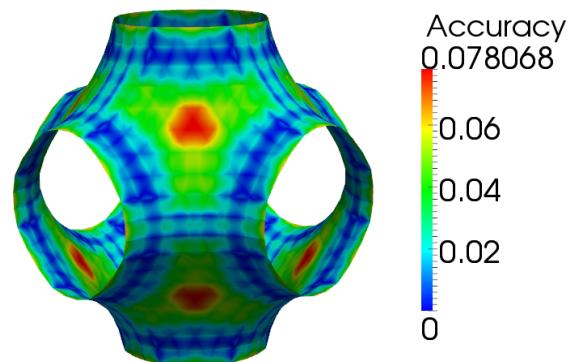
3.7 Method III: Distance Functioned Volume

In the previous section we saw that by introducing greyscale values into a binary volume the resulting surfaces were much smoother and better resembled the ideal surfaces. The greyscale values were introduced in a small region around the object as the result of smoothing algorithms and as such were not aimed specifically at improving the surface reconstruction. By using these values it is possible to control where the interpolation points for the marching cubes algorithm will lie, and hence, where exactly the reconstructed surface will be.

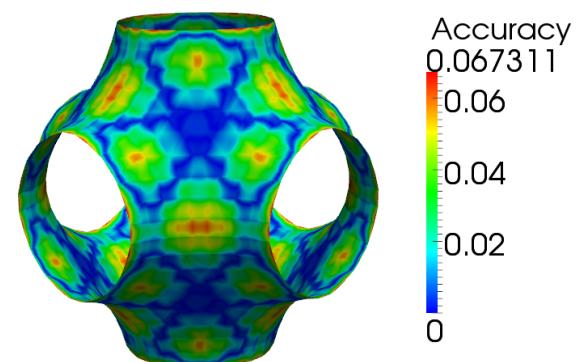
As the name suggests, voxel values in this region should reflect their distance from the surface. Voxels which are closer to the surface should have a value close to the iso-surface value, as this is where the reconstructed surface will be. Voxels



(a) Gaussian smoothing with $\sigma_{SD} = 0.330694$ (1.2 voxels)



(b) Mean smoothing with a radius of 2 voxels



(c) 160 iterations of pre-smoothing

Figure 3.9: Comparison of different smoothing algorithms on a binary volume

further away from the surface should vary in intensity linearly with the distance from the surface, to either solid or void depending on whether or not they are inside or outside the object.

Depending on the original representation of the object, measuring the distance to the surface can be computationally expensive. For example, it may be desirable to convert a CAD model (boundary representation or ‘b-rep’) into an image in order to combine it with an existing image, such as an implant with CT-scans of a patient. The first stage in the process is to define the resolution at which the CAD is to be voxelised, rather than specifying this in pixels, real-world units are used to define the voxel spacing (the distance between adjacent voxels). The volume, which must be large enough to contain the bounding box of the object, can then be divided into four regions of interest as shown in Figure 3.10(a). Each voxel in the volume must then be evaluated to determine which region it belongs to and ultimately its value.

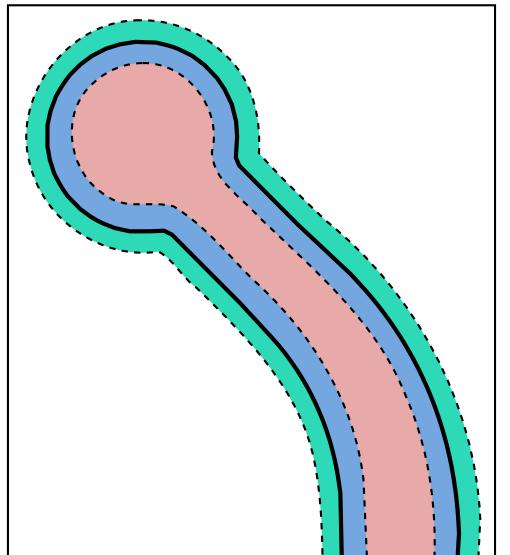
The value of each voxel can then be computed:

$$\begin{aligned}
 IsoSurface &= \frac{V^{\max} + V^{\min}}{2} \\
 ValueOf(Outside) &= V^{\min} \\
 ValueOf(Inside) &= V^{\max} \\
 ValueOf(Inwards) &= \frac{|Dist(p)|}{MaxDistance} \times IsoSurface + IsoSurface \\
 ValueOf(Outwards) &= V^{\max} - ValueOf(Inwards)
 \end{aligned} \tag{3.17}$$

Where V^{\max} and V^{\min} are the maximum and minimum possible voxel values, $Dist(p)$ is the signed distance that point p is from the object's surface (in real-world units) and $MaxDistance$ the maximum distance a voxel can be from the surface before it is considered as either fully in or out. Figure 3.10(b) shows an example image generated using this scheme.

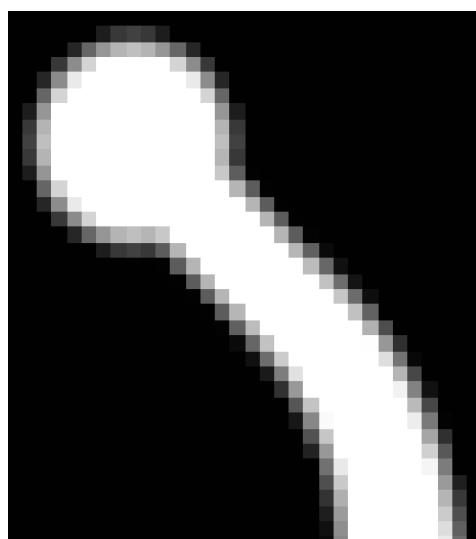
3.7.1 Distance Functions and Implicit Surfaces

In Section 3.5.4 the implicit functions were exploited to provide a measure of distance from the ideal surface, by using this idea again it is possible to substitute $Dist(p)$ in Equation 3.17 with the implicit function $f(p)$. A consequence of this is that the value of $MaxDistance$ must be unit-less, since $f(p)$ is, and an appropriate value needs to be chosen. The size of the greyscale region in the



Void ('outside')	Greyscale (inwards)
Solid ('inside')	Greyscale (outwards)

(a) Regions of interest



(b) Slice from the final volume

Figure 3.10: Voxelising a CAD model

volume should ideally be 2 to 4 pixels; there would be no gain from a larger region and a smaller region would lead to a ‘near binary’ volume exhibiting the same poor surface as in Figure 3.8. For the Schwarz Primitive example used thus far we choose $\textit{MaxDistance} = 0.6$. Repeating the test case, as specified in Section 3.6.2, leads to a reconstructed surface with $Q_1 = 0.003763$ and $Q_2 = 0.003051$.

Figure 3.11 shows the surface accuracy plot using this method.

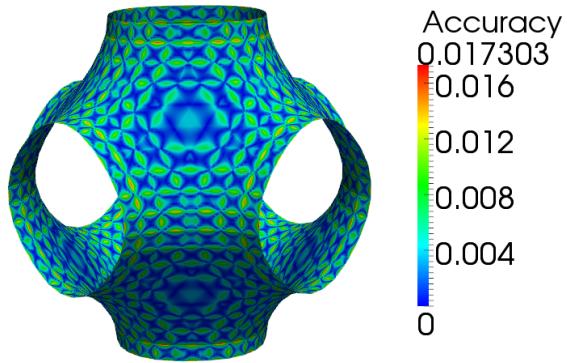


Figure 3.11: Surface accuracy for the ‘distance-functioned’ volume

3.7.2 Choosing $\textit{MaxDistance}$

When producing an image volume from a CAD model the value of $\textit{MaxDistance}$ is straightforward to compute. Since the CAD model is specified using real-world units, usually millimetres, the spacing of the volume is known and an appropriate value for $\textit{MaxDistance}$ can be determined. However, for the case where an implicit function is being treated as a distance function the values obtained are not distances measured in real-world units.

To further complicate the issue, the value for $\textit{MaxDistance}$ is dependant on the implicit function being used. In the previous example we chose $\textit{MaxDistance} = 0.6$ for the Schwarz Primitive, whereas for the Schoen Gyroid a value of 8 is more

appropriate to generate a greyscale region of the desired size. This issue is caused by the difference in the gradient on the ideal surface of the two functions. As the value of $f(p)$ changes more rapidly with the distance from the ideal surface (i.e. where $f(p) = 0$) the effective ‘distance’ of voxels in the greyscale region also increases, thus requiring a larger value of *MaxDistance*.

Solution By measuring the change in value of $f(p)$ over a voxel on, or close, to the ideal surface the gradient magnitude at that point, hence the effective spacing for that region, can be computed. The gradient at a voxel can be calculated as follows:

$$\mathbf{g}_i = \frac{f(p_{i-}^0) - f(p_{i+}^0)}{2} \quad (3.18)$$

where p_i^0 is a voxel on the ideal surface and p_{i-}^0 and p_{i+}^0 are the two neighbouring face-connected voxels along axis i .

From Eq. 3.18 the gradient magnitude can be calculated:

$$|\mathbf{g}| = \sqrt{\sum_{i=0}^d \mathbf{g}_i^2} \quad (3.19)$$

where d is the dimensionality of the image.

With the gradient magnitude at the surface now known, a suitable value for *MaxDistance* can now be computed:

$$MaxDistance = |\mathbf{g}| \cdot GreyScaleSize \quad (3.20)$$

where *GreyScaleSize* is the desired size of the greyscale region in pixels. A value of 2 px is appropriate given the nature of the marching cubes algorithm.

3.7.3 Algorithm

The process of generating the distance functioned volume from an implicit function can be split into two parts. First a volume represented by real values (e.g. `float`) is generated by sampling the function at each voxel, Algorithm 3. Following this the rules in Eq. 3.17 can be applied to the volume to produce the final distance functioned volume. See Algorithm 4.

By dividing the process into these two stages it is possible to find a candidate voxel for p^0 efficiently since the values of the function are only computed once per voxel. While it is likely that p^0 will not lie exactly on the ideal surface, any sufficiently well resolved volume will contain voxels whose close proximity to the surface makes them suitable candidates.

Algorithm 3: Floating-point volume generation

Input: $f, n_x, n_y, n_z, x_{\max}, y_{\max}, z_{\max}$

Output: `FloatVolume`

while *not all voxels visited* **do**

$(x, y, z) \leftarrow \text{GetCurrentPosition}(\text{FloatVolume})$
 $(x', y', z') \leftarrow \left(\frac{x}{x_{\max}} \times 2\pi n_x, \frac{y}{y_{\max}} \times 2\pi n_y, \frac{z}{z_{\max}} \times 2\pi n_z \right)$
 $\text{FloatVolume}[x, y, z] \leftarrow f(x', y', z')$
`MoveToNextVoxel(FloatVolume)`

end

3.7.4 Summary

This section has examined various methods of generating image volumes from implicit functions so that they may later be integrated with other image data, or meshed directly with ⁺ScanFE.

Algorithm 4: Distance functioned volume generation

Input: $\text{FloatVolume}, \text{isoSurface}, f, n_x, n_y, n_z, x_{\max}, y_{\max}, z_{\max}$

Output: DistVolume

```
/* Find the gradient magnitude on the surface */  
p0 ← GetSurfaceVoxel(FloatVolume)  
gm ← 0  
for i=0 to 3 do  
    gm ← gm +  $\left( \frac{f(p_{i-}^0) - f(p_{i+}^0)}{2} \right)^2$   
end  
gm ←  $\sqrt{gm}$   
  
/* Generate the final volume */  
MaxDistance ← 2 · gm  
while not all voxels visted do  
    (x, y, z) ← GetCurrentPosition(DistVolume)  
    v ← FloatVolume[x, y, z]  
    av ← |v|  
    if av > maxDist then  
        if v ≤ isoSurface then DistVolume[x, y, z] ← Vmin  
        else DistVolume[x, y, z] ← Vmax  
    else  
        α ←  $\frac{av}{MaxDistance} \times isoSurface + isoSurface$   
        if v ≤ isoSurface then  
            | DistVolume[x, y, z] ← α  
        else  
            | DistVolume[x, y, z] ← (Vmax - α)  
        end  
    end  
    MoveToNextVoxel(DistVolume)  
end
```

The most straightforward algorithms presented, which generated binary volumes, proved to be the most efficient but at the cost of the accuracy of the surface reconstruction. Surfaces from these algorithms bear little resemblance to the smooth surfaces expected from the trigonometric functions. The introduction of greyscale values through the use of smoothing algorithms improves the surface smoothness, but at the cost of both accuracy and computational resources. The most accurate representations were generated by treating the implicit functions as distance functions and setting voxel values accordingly. Surfaces reconstructed using this algorithm conformed closely to the ‘ideal’ surface. Although more computationally demanding than the binary algorithms, the algorithm presented is an efficient and robust method for generating the desired volumes.

3.8 Volume Fraction Calibration

It is desirable to be able to generate micro-architectures using the methods described in the previous section with a prescribed volume fraction. The volume fraction of a micro-architecture is calculated as the ratio of solid, V_S , to total volume, V_T :

$$\text{Volume fraction} = \frac{V_S}{V_T} \quad (3.21)$$

Porosity is also commonly used to describe the solid to void fraction in micro-structures:

$$\phi = \frac{V_V}{V_T} \quad (3.22)$$

where V_V is the volume of the void.

The volume fraction of a micro-architecture will influence a number of dif-

ferent physical properties related to the bulk behaviour of the structure. For example, one would expect an increase in volume fraction to lead to an increase in Young’s modulus, while at the same time decreasing the pore size and the structure’s permeability. The target application of the micro-architecture will dictate the volume fraction; structures used for bone scaffolds are chosen so that their bulk properties match that of bone (Hollister *et al.* [2002]), whereas internal support structures for aerospace components are chosen to maximise stiffness and minimise weight.

3.8.1 Calibration by Re-Iso-Surfacing

Gabbrielli *et al.* [2008] demonstrated a method of controlling the volume fraction of a structure generated by an implicit function by choosing different iso-surface values. Increasing or decreasing the iso-surface value has the corresponding effect on the volume fraction of the structure. However, the work by Gabbrielli does not allow for precise control of the volume fraction – for example a structure with a 17% volume fraction cannot be generated without guessing an appropriate iso-surface value through trial and error.

For an iso-surface value of k the ideal surface is defined as:

$$f(p) = k \tag{3.23}$$

However, it is more convenient to always use an iso-surface value of 0:

$$\begin{aligned} f'(p) &\equiv f(p) - k \\ &= 0 \end{aligned} \tag{3.24}$$

In order to achieve a certain volume fraction an appropriate value for k needs to be determined. It is known that this value must lie within the range $\text{Min}(f'(p, k)) < i < \text{Max}(f'(p, k))$, as any value outside of this range cannot satisfy the expression $f'(p, k) = 0$ and hence the volume fraction will be 0. Using the bisection method (similar to a binary search algorithm using real numbers) a value for k can be found which minimises the difference between the computed and desired volume fraction, within a given tolerance. The algorithm is as follows:

Algorithm 5: Bisection algorithm

Input: $targetVF, FloatVolume$
Output: $isoSurface$

```

 $min \leftarrow \text{MinOf}(FloatVolume)$ 
 $max \leftarrow \text{MaxOf}(FloatVolume)$ 
 $range \leftarrow (max - min)$ 
 $offset \leftarrow range \times 0.5$ 
 $isoSurface \leftarrow (min + offset)$ 
 $actualVF \leftarrow \text{GetImageVolumeFraction}(FloatVolume, isoSurface)$ 

while  $|targetVF - actualVF| > \epsilon$  do
     $offset \leftarrow offset * 0.5$ 
    if  $actualVF > targetVF$  then
        |  $isoSurface \leftarrow isoSurface + offset$ 
    else
        |  $isoSurface \leftarrow isoSurface - offset$ 
    end
end

```

The algorithm makes use of an additional function, `GetImageVolumeFraction()`, which returns the volume fraction of a floating point volume using a given iso-surface value. An approximation of this value can be computed efficiently by taking the ratio of ‘inside’ voxels to the total number of voxels. A voxel is

considered inside if its value is less than the iso-surface value.

$$\text{Volume fraction approximation} = \frac{\text{Number of inside voxels}}{\text{Total number of voxels}} \quad (3.25)$$

Since this approximation disregards the partial volume effect, sub-voxel changes may not always result in a change in volume fraction. Although it is clearly desirable to be able to measure such changes, the computational cost in doing so outweighs the gain.

When used in combination with Algorithms 3 and 4 a pipeline can be established which allows the generation of lattice structures with specific volume fractions.

The bisection algorithm, as presented, may not be the most efficient algorithm for finding the desired iso-surface value. In some cases the algorithm may require a large number of iterations to meet the criteria to terminate, depending on the value of ϵ . Certain values of ϵ may also cause the algorithm to aim for a degree of accuracy unobtainable when using the inside/outside method of computing the volume fraction – also causing an excessive number of iterations. These issues could be addressed by using a fixed number of iterations in the algorithm.

There also exists a number of algorithms for finding the root of a function (i.e. finding x such that $f(x) = 0$) which may be applicable to this problem. However, as the function used to calculate the volume fraction is discontinuous it may not always be possible to rely on its derivative or gradient, as required in *Newton's method*, the *secant method* and the *false position method*. As with the bisection method, these algorithms are iterative processes which progressively find better approximations to the root of the function. The main difference between these

algorithms and the bisection method is that they use the rate of change of the function to more efficiently approach its root.

Alternatively, the efficiency of the bisection algorithm may be increased by choosing more appropriate initial values for *min* and *max*. One possible method for choosing these values is to first compute a histogram for the volume *FloatVolume* using an appropriate number of uniform bins. Using this histogram it would then be possible to find a bin with a corresponding volume fraction closest to the target volume fraction. The upper and lower values used by the bin can then be used to seed the bisection algorithm. Depending on the accuracy required, a histogram with a large number of bins may also be able to provide a suitable iso-surface value without the use of the bisection algorithm, or any other iterative approximating algorithm.

3.8.1.1 Examples

Figure 3.12 shows a single unit cell of the Schoen Gyroid at 15%, 33% and 50%.

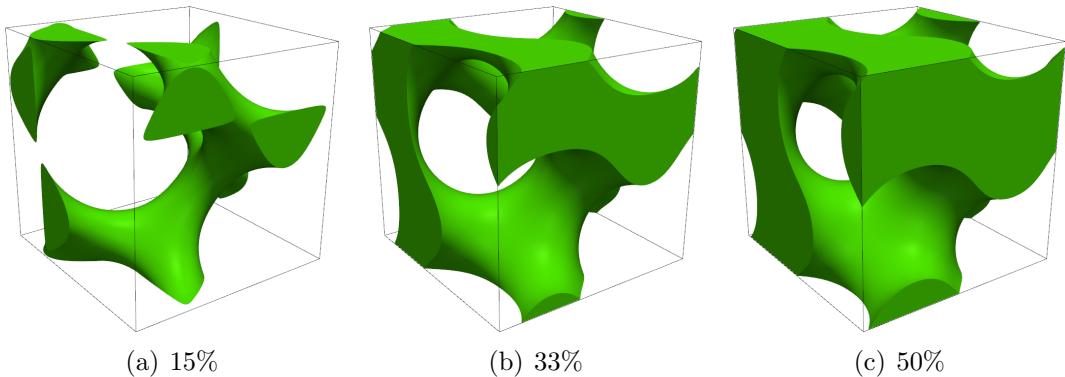


Figure 3.12: Volume fraction calibrated unit cells

3.8.2 Calibration by Erosion/Dilation

Another possible technique for controlling the volume fraction of an implicitly defined structure is though the use of morphological erosion and dilation. Rather than choosing different iso-surface values in order to achieve different volume fractions a single iso-surface is chosen as a starting geometry. Using the algorithms developed in this chapter the corresponding volume image can be generated. Morphological erosion and dilation may then be performed on the volume using an appropriately sized kernel to achieve the desired volume fraction. Figure 3.13 shows how this method can be used to generate different unit cells from the starting cell.

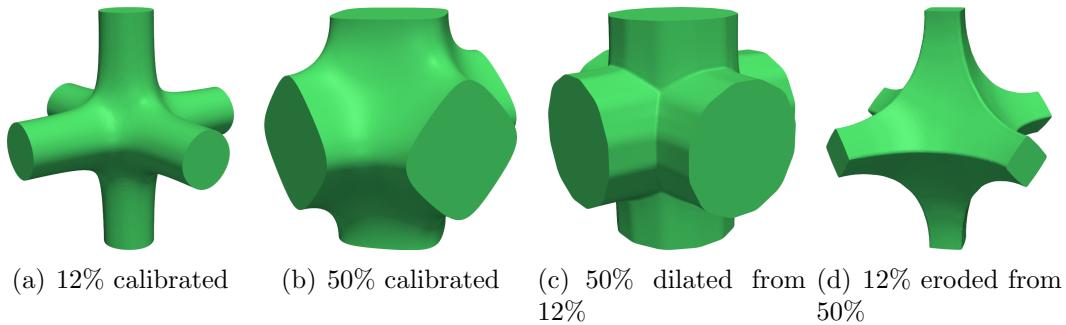


Figure 3.13: Unit cells generated to specific volume fractions using different techniques

The disadvantage of this method is the computational resources required to perform large morphological operations, which are considerably greater than that required to generate a volume from a different iso-surface. It also becomes more difficult to target a specific volume fraction using erosion or dilation and would require additional computation in order to find an appropriate kernel size.

This method may be better suited for smaller changes in volume fraction using sub-voxel morphological operators so that precise volume fractions can be

achieved.

3.9 Summary

This chapter has focused on the development of image-based algorithms for the generation and representation of lattice structures from periodic implicit functions. A number of different algorithms were examined including their effects on the surface reconstruction accuracy. As may have been expected, the binary algorithms proved to be the most efficient but at the cost of surface accuracy. The introduction of greyscale values significantly improves surface accuracy, with the most accurate reconstruction being produced from the distance functioned algorithm developed in §3.7.3. Through the use of a bisection algorithm the micro-architectures may be generated to a specific volume fraction, allowing the creation of micro-architectures with different properties (i.e. stiffness and porosity).

Chapter 4

Functionally Graded Structures

Functionally graded structures (FGSs) are similar in principle to functionally graded materials (FGMs), a relatively recent development whereby a mixture of two or more materials (with differing properties) is varied spatially to create a new material with customised properties. By choosing specific materials and mixing them with a certain gradient, new materials can be created with properties tailored for a specific application. This may include customising the mechanical, thermal and/or electrical properties of the material. The creation of custom materials can ensure that the specific requirements of an application are met, rather than relying on an existing homogeneous material to fulfil the requirements.

Functionally graded structures are typically micro-architectures composed of two or more phases, where one phase may be void. Each phase is typically made of a homogeneous material. As with functionally graded materials there is a spatial variation over the structure which in-turn introduces variations in various properties across the structure. To introduce this variation some aspect of the structure's geometry, such as cell density, cell size, or topology is altered.

Torres-Sanchez & Corney [2009] demonstrated that functionally graded structures could be produced from polyurethane by exposing the substance to various ultrasound signals during the foaming process. In comparison to the computational approaches, when combined with additive manufacturing (AM), the use of ultrasound in the foaming process provides considerably less control over the functional grading in resulting structure. In both Gabbrielli *et al.* [2008] and Chen [2007a] methods for the computational generation of FGSSs are presented which rely on AM to realise designs, allowing for precise control over the final geometry.

4.1 Generation

It has been shown that the value at which the implicit function is iso-surfaced directly controls the final structure's volume fraction. Gabbrielli (Gabbrielli *et al.* [2008]) exploited this fact to produce functionally graded structures by effectively iso-surfacing the implicit functions with a non-constant value. The result of which can be expressed as follows:

$$f(x, y, z) - \Phi = 0 \quad (4.1)$$

where Φ is the *grading function*, dependant on any number of different possible parameters. An example of a linearly graded structure is given in Gabbrielli *et al.* [2008] where the grading function varies linearly between two values, k_1 and k_2 over the z-axis. How these values were determined and the volume fractions there correspond to were not explained.

Using algorithm 5 it becomes possible to choose values for k_1 and k_2 which would result in a functional variation between two specific volume fractions. The following is an example of how Φ may be constructed such that the resulting structure varies in volume fraction linearly from 10% to 50% over the z-axis.

$$\begin{aligned} k_1 &= \text{CalibratedIsoSurface}(f, 0.1) \\ k_2 &= \text{CalibratedIsoSurface}(f, 0.5) \end{aligned} \quad (4.2)$$

$$\Phi(x, y, z) = \frac{z}{z_{\max}} \times (k_2 - k_1) + k_1$$

where the function `CalibratedIsoSurface` represents the process defined by algorithm 5.

4.1.1 Apparent Non-Linearity

In Gabbrielli [2009] it is noted that while the iso-surface value, k , is proportional to the structure's volume fraction, the relationship is non-linear. However, on investigation it appears that, within a certain range, the relationship may be described as quasilinear. Figure 4.1 shows the relationship between the iso-surface value and volume fraction for four implicit functions. From this plot there appears to be some non-linearity at the higher ($> 75\%$) and lower ($< 20\%$) volume fractions. However, this is almost certainly due to the effects of resolution where small features are not sufficiently represented. This is clear from the Neovius' Surface which is the most complex and also exhibits the greatest non-linearity.

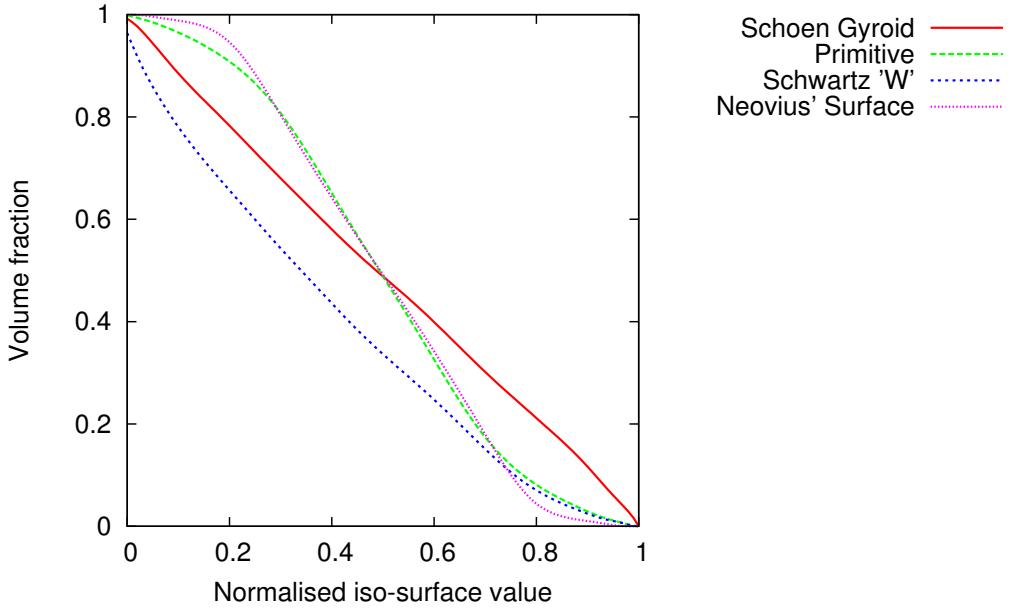


Figure 4.1: The relationship between iso-surface value and volume fraction for selected implicit functions

4.2 Specific Density Variations

Certain desired or perhaps optimal density variations are likely to have highly complex, or non-obvious, mathematical representations leading to difficulties in defining the grading function, Φ . For example, an engineer designing a micro-architecture may wish to manually specify denser regions throughout the structure in order to meet the required mechanical properties. The need for arbitrary density variations becomes more apparent when an optimisation process is considered. Typically size and shape or topology optimisation is used to create optimal micro-architectures or unit cells. At the highest level these optimisation processes can be considered an iterative process wherein an initial guess is continually modified to minimise or maximise a given criterion. For this process to work there must be methods available which allow quantitative modification of some aspect

of the structure’s geometry. In this instance the density of the structure’s cells is the modified parameter.

To facilitate the creation of such structures we introduce the concept of a *density map*.

A density map is a representation of the desired *relative* densities of a micro-architecture throughout the domain. The map itself is represented in image-space (either 2D or 3D) at a resolution equal to that of the micro-architecture, ensuring a one-to-one voxel mapping. The voxel values can be encoded using various data types, such as `unsigned char` or `float`, with the maximum and minimum values representing the maximum and minimum desired volume fractions. Intermediate values are interpolated linearly between these two values. In the same manner as with functionally graded structures these values are used to determine the iso-surface value, k .

Using the `CalibratedIsoSurface` function (Algorithm 5) the iso-surface values corresponding to the target minimum and maximum volume fractions, ϕ_{\min} and ϕ_{\max} , can be found.

$$\begin{aligned} k_{\min} &= \text{CalibratedIsoSurface}(f, \phi_{\min}) \\ k_{\max} &= \text{CalibratedIsoSurface}(f, \phi_{\max}) \\ k_{\text{range}} &= |k_{\max} - k_{\min}| \end{aligned} \tag{4.3}$$

However, as the final function will be iso-surfaced using a single value it must be updated to introduce the desired variations. This is achieved by linearly re-scaling the density map to the range $[0 : k_{\text{range}}]$. That is, the range of values in the density map must be equal to the range of target iso-surface values. This re-scaling is required to ‘normalise’ the density map such that when the values

are added to the implicit function the resulting volume can be iso-surfaced using the minimum target iso-surface value, k_{\min} . Voxels corresponding to those in the density map with the minimum value will not be increased as the re-scaled value will be 0 and hence when the final volume is iso-surfaced will produce the desired minimum volume fraction. Similarly, voxels corresponding to those in the density map with the maximum value will be offset by k_{range} and hence will produce the desired maximum volume fraction.

Using the rescaled density map the grading function can be defined:

$$\Phi(x, y, z) \equiv M_{x,y,z} - k_{\min} \quad (4.4)$$

where $M_{x,y,z}$ is value of the voxel at (x, y, z) in the density map and k_{\min} is the iso-surface required to achieve the minimum desired volume fraction.

The implicit function now becomes:

$$f(x', y', z') - \Phi(x, y, z) = 0 \quad (4.5)$$

The values in the density map could be specified in any number of different ways. In 2D (or 3D with a prismatic variation) painting techniques, similar to those used for image segmentation, can be used to manually specify variations. In 3D this becomes more difficult, although it is not impossible to imagine some 3D input device, such as a haptics device, being used to paint in 3D.

While intended for arbitrary variations, the use of a density map does not exclude the use of functional variations. Although it may appear unnecessary and inefficient to generate a density map for a functional variation, rather than

using the function directly, it can be useful for visualisation. Once generated, structures with a large number of cells can easily exceed 100 million polygons, making interactive visualisation computationally expensive. By using various volume rendering techniques or by reducing the density map's dimensionality (i.e. by taking a slice of the volume), it becomes possible to visualise the density variations across the domain.

There are also additional benefits to using a density map: its image-based representation means that image processing techniques can be applied to it. In particular image smoothing filters, such as Gaussian or mean smoothing, can be applied to ensure smooth transitions between areas of different densities. This is particularly desirable to avoid step transitions, where the density changes suddenly between regions, which can lead to stress concentrations in the structure.

4.2.1 Example

Examples of the specific density variations are best shown in 2D where the entire structure can be seen in a static image, as in Figure 4.2.

4.3 Non-Specific Geometrical Variations

In the previous sections techniques for generating lattice structures with density variations were developed for the purpose of controlling the bulk response of the structure. In these structures the cell size (the physical dimensions over which the function is periodic) remains constant. An alternative method for achieving this has been demonstrated in Chen [2007a] whereby the cell size is varied throughout the domain to create regions of different densities. This section will look briefly

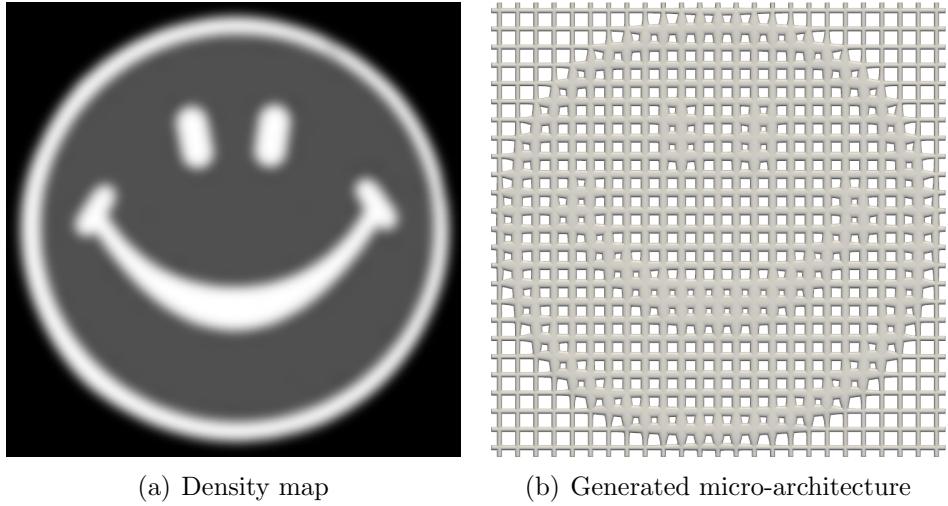


Figure 4.2: Example usage of a relative density map

at how this technique can be applied to the methods used in this chapter.

In comparison to the methods described in this chapter, Chen uses a more simplified strut-based representation for micro-architectures (Chen [2007a]). Unit cells are defined by a set of struts connecting a set of vertices with their topology chosen from a library consisting mainly of polyhedra. The simple representation of the structure allows the use of a space warping function to displace the cell vertices. The space warping function can be defined such that cell size decreases, hence density increases, with either an existing function or discrete values from finite element analysis.

Applying the same technique directly to a surface or volume mesh of a micro-architecture generated using the techniques described, would cause undesirable results. If applied to a surface mesh, the warping function would likely cause the structures ribs to appear stretched, resulting in a loss of accuracy. For the case of a volume mesh the warping would likely have a negative effect on the overall

mesh quality due to the distortion elements would undergo - in-turn causing a loss of accuracy in any finite element or CFD simulations.

Alternatively given a space-warping function Θ it is possible to warp the space the implicit function is evaluated over:

$$f'(x', y', z') \equiv f(\Theta(x, y, z)) \quad (4.6)$$

As an example we choose the space-warping functions developed in Bùi Xuân [2008]. These functions allow the placement of ‘morphing spheres’, each of which has a stable zone and a transition zone. In the stable zone the space is expanded effectively increasing the unit cell size. The transition region is, as the name suggests, the region over which the space is compressed back to the original size. An example is given in Figure 4.3.

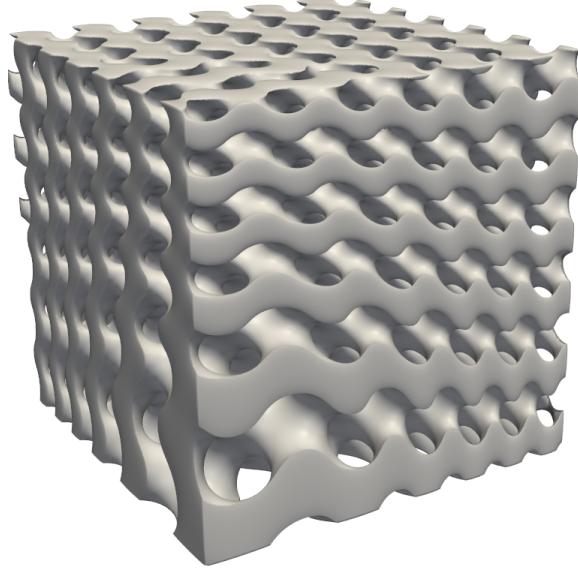


Figure 4.3: Lattice structure generated using a space-warping function

However, as is to be expected with the use of space-warping functions on

this type of lattice structure, the functions cause a ‘squashing’ of ribs. For the functions used in Figure 4.3 this effect is limited to the transition region.

4.4 Hybrid Functions & Transition Cells

In the previous sections methods for generating lattice structures with varied mechanical properties were examined. The principle technique for achieving this was the introduction of density variation throughout a fixed topology structure. Previous studies have achieved custom and varied properties through the use of multiple unit cells of different topologies (Starly [2006], Hollister & Lin [2007]). As the topology of a unit cell can have a considerable influence on its effective properties, such as bulk stiffness or porosity, structures with spatial variations in topology (either discrete or continuous) are able to achieve custom properties.

Introducing topological variations in a structure can prove problematic. As previously stated, it is highly desirable to maintain a continuous structure - any two points chosen at random should be connected. With density variations this is ensured using upper and lower bounds. For the case of topological variations regions composed of different unit cells must be joined or bridged. Assuming the interface between the two regions, R_1 and R_2 , is a plane then there will exist two corresponding sets of nodes N_1 and N_2 . In some instances it may be the case that $N_1 \cap N_2 \neq \emptyset$, that is, the two regions share nodes. For such structures it is possible to take advantage of this relationship and join the two regions by merging the coinciding nodes as in Figure 4.4.

However, for any two arbitrarily chosen unit cells in 3D it is more likely that we will have $N_1 \cap N_2 = \emptyset$ and are not able to directly join the two regions. To

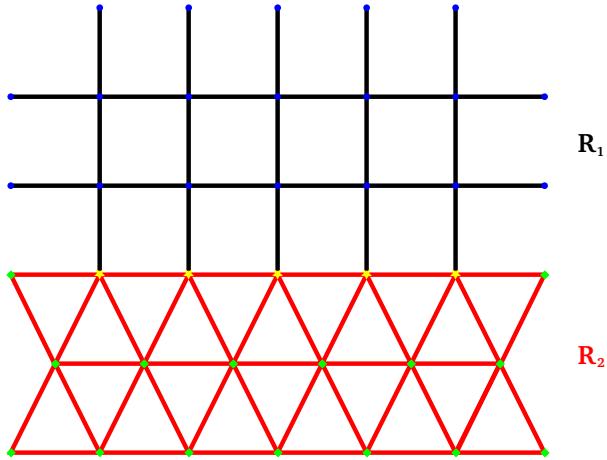


Figure 4.4: Joining unit cells of different topologies in 2D, without the need for a transition region

resolve this we introduce a *transition region* which has the sole purpose of bridging the two interfaces. This method is used in the commercial software NetFabb Net [2009].

When dealing with beam-based micro-architectures the creation of an appropriate ‘sub-structure’ to bridge different unit cells can be achieved simply by introducing additional beams. Usually without the need to introduce additional nodes.

4.4.1 Implicit Transition Cells

The advantages of using a simplified representation of a micro-architecture, such as beams, are clear when we wish to bridge regions of different unit cells. However, the type of structures discussed in detail in this chapter (those with an implicit representation) cannot be treated as a collection of beams and nodes. Instead more complex representations are required to join or bridge cells. In this section we take a novel approach to modelling the region of transition implicitly.

As with beam-based structures there may be cases where different unit cells can be joined directly to form a continuous structure. The criteria for this is dependent on the contact area between the two faces at the interface. For the implicitly defined unit cells the existence of a contact surface can be proved formally. Assuming the unit cells are periodic over 2π and the interface region is a plane perpendicular to the z -axis, then the intersection of the two implicit functions on the plane must yield a value greater than or equal to the iso-surface value. This can be expressed formally by Equation 4.7.

$$\exists x, y (\min[f_1(x, y, z), f_2(x, y, z)] \geq 0), \quad z = 2\pi \quad (4.7)$$

Since the move to image-space requires a discretisation of the implicit function Equation 4.7 should be evaluated over the discretised domain to ensure any topological changes are accounted for.

In cases where there exists a notable contact area it is possible to create a greyscale volume by taking the union of the two volumes generated using Algorithm 4. Figure 4.5 shows how this can be used to join two different unit cells, the Pinched Primitive and the Schoen Gyroid. From the slice of the generated volume (Fig 4.5(a)) it is clear that in this case the contact area between the two cells is minimal, consequently this interface region would likely become a point of failure when loaded.

It is clear that joining different unit cells directly as previously described is not an appropriate or robust solution to the problem. Instead the need for a transition layer for implicitly defined unit cells is apparent.

Using space-time blending methods as described in Pasko *et al.* [2004] it is

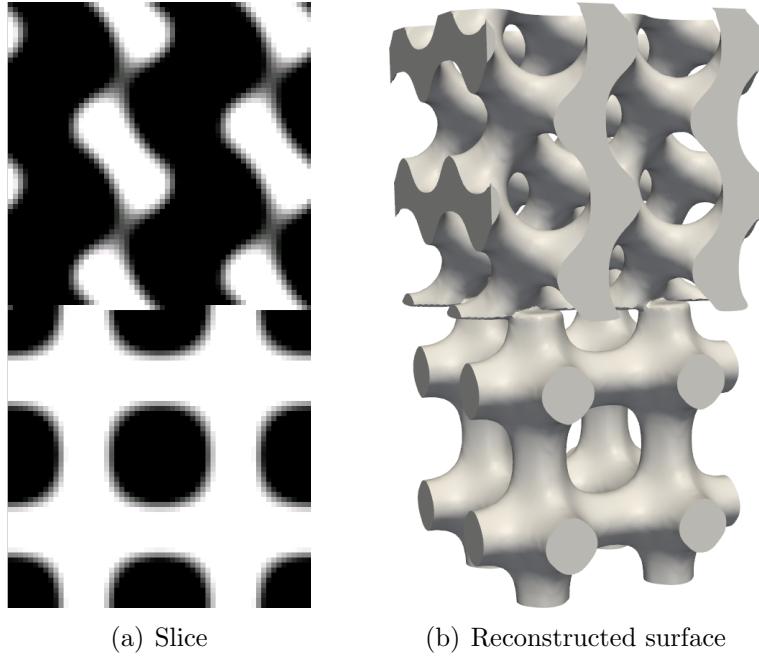


Figure 4.5: Joining implicitly defined cells of different topologies

possible to morph an object with an implicit representation into another over time. This is achieved by taking a weighted average of the two functions and varying the weighting over time. The equation below shows this with a linear variation.

$$f'(x, y, z) \equiv \frac{tf_1(x, y, z) + (1 - t)f_2(x, y, z)}{2}, \quad t \in [0 : 1] \quad (4.8)$$

Figure 4.6 shows the transformation between the Pinched Primitive and Schoen Gyroid unit cells. The transitional states show the new unit cell (f') maintains its continuity – a property required of a transition layer.

Temporal transition between unit cells, while interesting, have no direct practical use for mechanical applications and hence Equation 4.8 should be modified to provide a spatial transition. For the transition layer we desire a transition be-

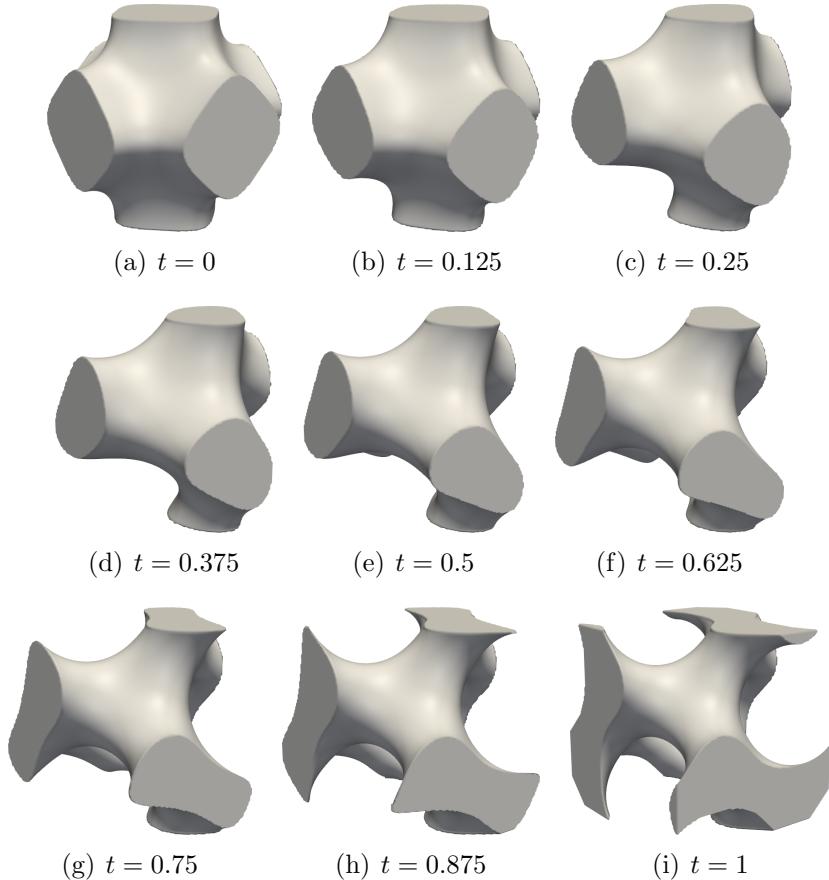


Figure 4.6: Space-time blending of the Pinched Primitive and Schoen Gyroid unit cells

tween two unit cells over some region and that the layer interfaces appropriately with both types of cell. The size of a single transition cell is chosen, arbitrarily, to be equal to one unit cell (2π).

$$f'_{cell}(x, y, z) \equiv \frac{1}{2} \left[\frac{t}{2\pi} f_1^*(x, y, z) + \left(1 - \frac{t}{2\pi}\right) f_2^*(x, y, z) \right], \quad t \in [0 : 2\pi] \quad (4.9)$$

where f_1^* and f_2^* are the implicit functions normalised such that when iso-surfaced at 0.5 they produce a structure with the desired volume fraction. The two functions can represent unit cells of different volume fractions. Figure 4.7 shows an

example transition cell using the Pinched Primitive and Schoen Gyroid unit cells, the value of t is varied over the z -axis.

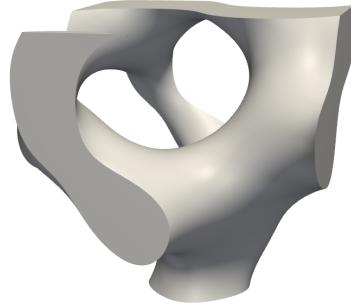


Figure 4.7: Transition cell for Pinched Primitive and Schoen Gyroid unit cells

By further extending Equation 4.9 to use a piecewise linear variation and evaluating f' over a larger range a structure demonstrating the use of the transition cell can be created. Figure 4.8(a) shows an example structure where a transition cell is used to two regions composed of the Pinched Primitive and Schoen Gyroid unit cells. This method may also be used to join regions composed of the same type of unit cell but with differing unit cell size or density, as shown in Figures 4.8(b) and 4.8(c).

The transition region may also be extended further than a single unit cell to produce more interesting structures, as shown in Figure 4.9.

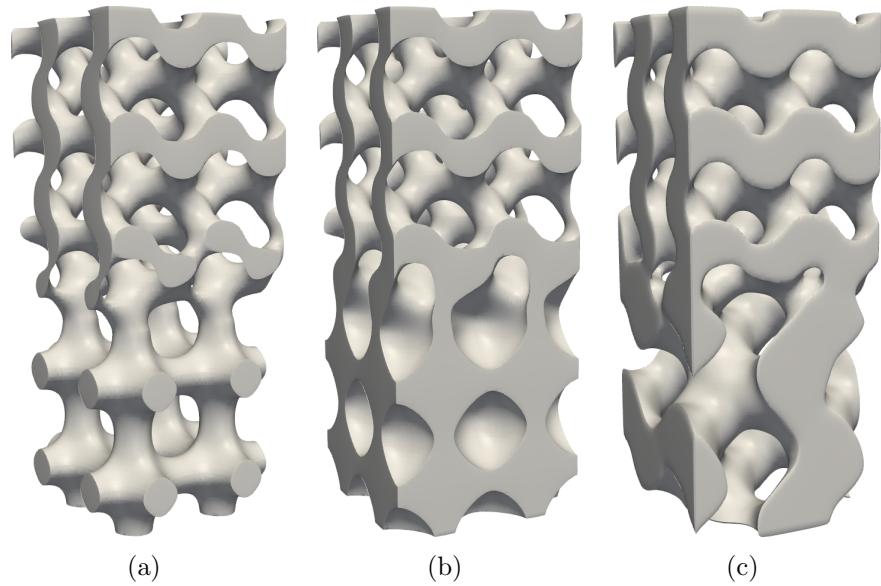


Figure 4.8: Example uses of transition cells

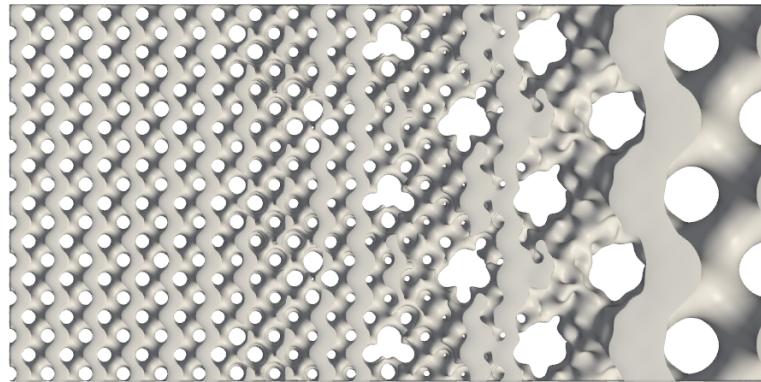


Figure 4.9: Blending unit cells of different sizes over a large region

4.4.2 Discussion

Transition cells, or regions, as blends of two implicit functions appear to be a suitable solution for bridging regions of differing topology or cell size. However, the resulting structures will likely have complex mechanical properties. The transition region will also have properties differing to those of the two topologies it bridges – an unavoidable consequence of bridging different unit cells.

4.5 Non-Periodic Functions

Up to this point we have focused exclusively on implicit modelling with periodic functions that are not true distance functions. The implicit representations of simple geometries, such as cylinders, are often used to construct more complex geometries in a process known as *constructive geometry solid* (CGS). This process is worthy of mention as the same techniques developed can be applied to CGS for lattice structures.

By taking the union of three cylinders, one along each axis, a simple unit cell can be constructed. The implicit equation for a cylinder of infinite length along the z -axis, with a radius r , is as follows:

$$f_3(x, y, z) \equiv x^2 + y^2 - r^2 \quad (4.10)$$

Similarly, cylinders along the x (f_1) and y (f_2) axes can be generated by substituting the variables x and y appropriately. The union of the three cylinders is straightforward to compute:

$$f_{\cup} \equiv \min(f_1, f_2, f_3) \quad (4.11)$$

Unlike the functions studied thus far, f_{\cup} is clearly non-periodic. The resulting iso-surface will be the union of the three cylinders intersecting at the origin. Periodic functions are particularly useful for space-filling applications, as previously demonstrated. It would be possible to extend f_{\cup} to the union of many, offset, cylinders such that they created a lattice structure. Doing so, however, would be inefficient as it would require evaluating Eq 4.10 many more times. Instead it is

possible to introduce periodicity through the use of the mathematical operator modulo:

$$x' = \frac{(x \bmod 2\pi) - \pi}{2\pi} \quad (4.12)$$

with similar expressions for y' and z' . The modified coordinates scale the unit cell to the range $[-0.5 : 0.5]$ so that the intersection of the cylinders occurs in the centre of the cell. With the modified coordinates the function in Eq 4.11 can be used as a periodic function in any of the algorithms developed in this chapter.

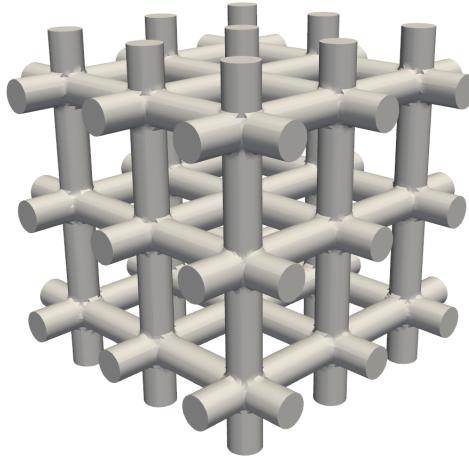


Figure 4.10: Periodic lattice structure generated from the union of three cylinders

Another possible method for generating lattice structures based on an implicit form is to first start with an explicit form. Lattice structures are often represented using beams, or simple polylines. These simple representations can be easily ‘distance-functioned’ to produce a volume which implicitly defines their geometry. The open source framework VTK provides the functionality to do so.

There are a number of advantages to using traditional distance functions, such as that in Eq 4.10 or distance functions from explicit forms. In both cases we assume that the beams have a radius of zero (for the cylinder we simply use

$r = 0$). For explicit forms the generation of a distance function volume is greatly simplified as a signed distance is not required, removing the need to determine which points are inside and which are outside. The resulting volumes also have a useful property not found with the periodic functions previously used; the iso-surface values represent the radius of the beams. For example, to generate a structure with beams 0.2 units in radius the volume is simply iso-surfaced at 0.2 and similarly for any radius greater than the sampling size. The distance functioning also removes the need to perform any special processing where beams meet, such as complex triangulation or introducing fillets (as in Chen [2007a]). The generated volumes can also be modified in the same manner as previously described to introduce variations in density throughout the structure. An example is given in Figure 4.11.

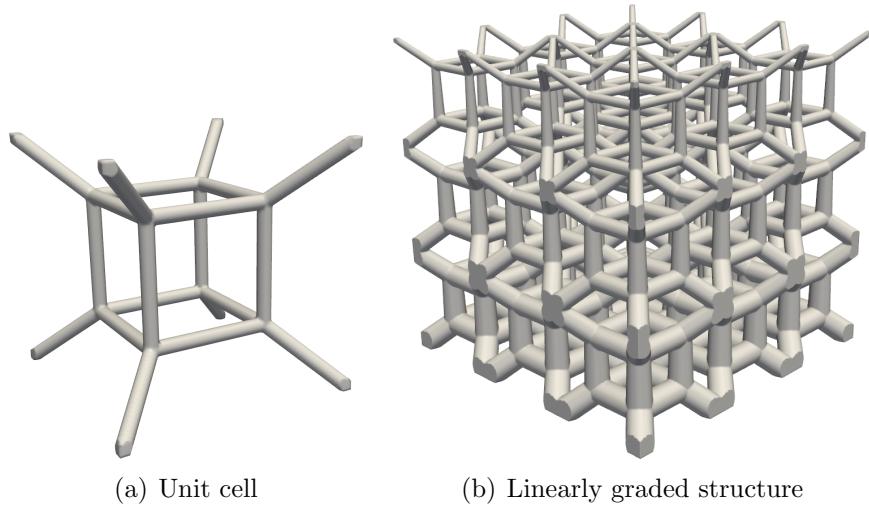


Figure 4.11: Distance functioned unit cell and lattice structure

4.6 Summary

The use of implicit modelling has been demonstrated to be a highly flexible way of modelling functionally graded micro-architectures with customised properties. A new method was developed to allow the creation of micro-architectures with arbitrary density variations. This method enables the creation of micro-architectures with complex, well-defined, variations as will likely be required for optimisation or applications such as bone scaffolds. The variations in density were introduced by effectively iso-surfacing with a non-constant value, referred to as the grading function, Φ . This grading function can be defined using a mathematical expression or through the use of a density map. By using a density map arbitrary variations in density can be easily introduced into the structure.

Novel methods have also been developed to mix unit cells of different topologies to create structures with varied properties. This was achieved by adapting a space-time blending function such that a transition region between two otherwise uniform lattices could be formed.

The methods developed in this chapter were also shown to be applicable to the creation of lattice structures based on the distance functions of beams, allowing the creation of more ‘regular’ unit cells.

Chapter 5

Domain Conforming

Micro-Architectures

The previous chapters have concentrated on the generation of micro-architectures within a cuboidal domain, in particular the domain defined by the Cartesian grid discretisation. For the majority of real-world applications there will exist a pre-defined geometry which will define the domain for the micro-architecture. This geometry may be specified using any number of methods, including CAD models, non-invasive medical imaging or implicit representations. Methods allowing for the generation of micro-architectures conforming to domains of arbitrary complexity are presented in this chapter.

Another application which has proven to be of considerable interest recently is internal architectures. Internal architectures are typically open cell lattice structures generated within a shell. The external geometry of this shell is designed to match that of an existing CAD model. In effect, the creation of the internal structure involves hollowing a CAD model and filling the void with a lattice

structure. Internal architectures can be considered a form of biomimicry design as they closely simulate the naturally occurring structures in many biological structures (e.g. bone, plant stems). That is, a spongy structure (cancellous bone) enclosed in a much denser shell (cortical bone). The purpose of mimicking this arrangement is to reduce the weight of components while maintaining stiffness. Internal structures may also bring further advantages, such as being able to act as a supporting structure in certain manufacturing processes. Novel methods for generating internal architectures in CAD models are presented in this chapter.

Methods for generating domain conforming micro-architectures are required to fully utilise additive manufacturing processes (e.g. SLM, SLS, FDM, etc...). Subtractive methods have previously been used for the same purpose, for example in the Mandible Reconstruction Project (Jamison *et al.* [2010]) a block of hydroxyapatite with a lattice structure was milled to match the geometry of a CAD model. Using subtractive methods to achieve an external geometry for structures which can be produced using additive manufacturing is clearly not an efficient use of material or time.

This chapter looks at methods for generating domain conforming micro-architectures for two applications, bone scaffolds and internal structures for CAD models.

5.1 Bone Scaffolds

Scaffold architectures should be designed such that their external geometry matches that of the defect site they are to occupy. The defect site will be either a region of void or of poor quality bone. Using non-invasive medical imaging techniques, such as MRI or CT scans, it is possible to acquire a voxel-based representation

of the defect site.

It is possible to bound the micro-architecture by the geometry defined by the defect site by taking the intersection of the two solids, as shown in Figure 5.1.

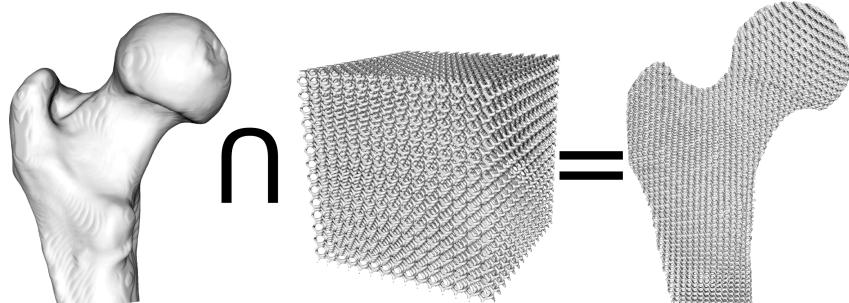


Figure 5.1: Domain conforming micro-architecture using Boolean intersection

This could be achieved using a number of different approaches, separated here into *CAD based* and *image based*.

CAD Based In this process the Boolean operation (intersection) is performed on the two triangulated surfaces (i.e. STL files). Such operations are not well-defined and results can vary depending on the CAD package used. In the worst case the operation may fail to complete. In cases where the operation is successful the result may suffer from poor quality triangles at the interface between the two solids and may require re-meshing in order to produce a volume mesh for FEA or CFD simulations. There are, however, advantages to performing Boolean operations in CAD space. Most notably that the operation is not lossy, that is, that it does not degrade the geometry of the involved CAD models. The two triangulated surfaces may also be generated from different sources, for example one from a CAD package and the other from image-based meshing.

Image Based The image based approach to model construction from medical imaging data is usually based on either the marching cubes (Lorensen & Cline [1987]), volumetric marching cubes (VoMac, Müller & Rüegsegger [1995]) or extended volumetric marching cubes algorithm (EVoMac, Young *et al.* [2008]). Following the segmentation of the image data using techniques such as thresholding, level set methods or manual painting, the segmented volume is processed using the appropriate algorithm – marching cubes for a surface mesh or volumetric marching cubes for a volume mesh. The marching cubes algorithm provides a very efficient and robust solution to surface and volumetric image meshing. Its usage in the mesh generation process is shown in Figure 5.2.

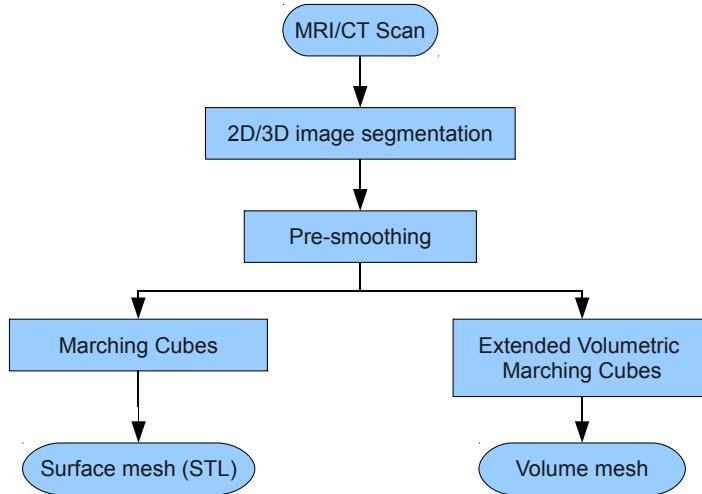


Figure 5.2: The process of data acquisition and processing for mesh generation

As the micro-architectures are generated in image-space it is desirable to also perform the intersection in image-space, where Boolean operations are well-defined, robust and very efficient. The only constraint on this process is that the two volumes must have the same discretisation, that is, the discretisation of both domains must be at the same resolution and extent. The resolution of image

data acquired from medical MRI or CT scans will typically be far too coarse to sufficiently represent a scaffold micro-architecture. To ensure the two volumes have the same resolution the defect volume can be re-sampled, typically using linear interpolation, to a resolution suitable for the micro-architecture. The final resolution will depend on how the data will be processed following the intersection. For rapid prototyping/manufacturing a lower resolution, $10 \times 10 \times 10$ px per unit cell, may be desired in order to maintain a manageable number of triangles. For computational simulations a higher resolution may be required.

With both the defect and micro-architecture volumes at the same resolution the intersection can be computed:

$$V_{\cap} = \min(V_1, V_2) \quad (5.1)$$

where V_1 is a voxel value from the defect volume, V_2 the corresponding voxel in the micro-architecture volume and V_{\cap} the voxel resulting from the intersection of V_1 and V_2 . Unlike a binary Boolean operation, this operation preserves greyscale values ensuring no loss in surface smoothness. Figures 5.3(a) and 5.3(b) show an example domain conforming micro-architecture within the defect site¹ of a mandible.

¹In this instance the defect site has been artificially introduced during segmentation

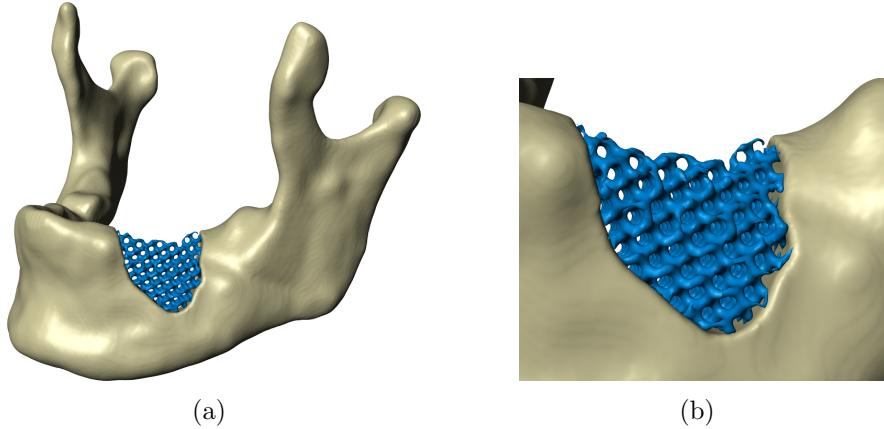


Figure 5.3: Example scaffold conforming to defect site geometry

5.2 Internal Architectures for CAD Models

This section looks at the generation of internal architectures for CAD models, in particular methods for shelling an existing geometry. The architectures used in this section will be the same as in previous sections, although referred to as *internal architectures* as no length-scale is implied for this application.

Parts of the work in this section form a patent application, the details of which are presented in Appendix B.

As with bone scaffolds one of the key objectives of generating internal architectures is the generation of a domain conforming architecture. However, this domain had previously been defined by the external geometry of a given volume in image-space. For internal architectures the domain of interest is defined as a proper subset of the volume enclosed by the CAD model. This is due to the requirement that the internal structure be enclosed by a shell, the external geometry of which should match that of the original CAD model.

The first stage of the process is to explicitly define the domain for the internal architecture by creating the solid shell from the CAD model. This can be achieved

using a traditional CAD package and an offsetting operation. Offsetting the CAD model’s original surface creates a new inner surface which is a given minimum distance from the original; that is, no two points on the surfaces will be closer than the minimum. When the hollow CAD model is exported as an STL file the outer and inner surfaces will be composed of triangles with opposing normals, thus defining a solid shell enclosing a void. With the hollow STL model the appropriate Boolean operations could be performed with an STL file of a lattice structure to produce the final STL file. However, as previously discussed, there are potential issues with performing Boolean operations on triangulated surfaces. Similarly, the offsetting operation for triangulated surfaces involves complex operations to re-mesh the original surface as in Chen [2007b].

The novel solution to these problems is to perform all of the required operations in image-space, as was done in the previous section with bone scaffolds. As in this case the external geometry is defined by a CAD model it will need to be converted into an equivalent representation in image-space. The conversion to image-space can be performed using ⁺ScanCAD from Simpleware which produces a ‘distance functioned’ volume as described in Section 3.7. Figures 5.4(a) and 5.4(b) show a CAD model and slice from the equivalent voxel volume, voxelised with a spacing of 1 mm.

Before the introduction of an internal structure the region of void must be defined. As the external geometry is represented in image-space at this stage the offsetting operation used in CAD packages cannot be used. In image-space morphological erosion can be considered an equivalent operation to offsetting.

Greyscale morphological erosion, as with other morphological operators, uses a kernel, defined in \mathbb{R}^3 in this instance. Unlike binary erosion, greyscale erosion

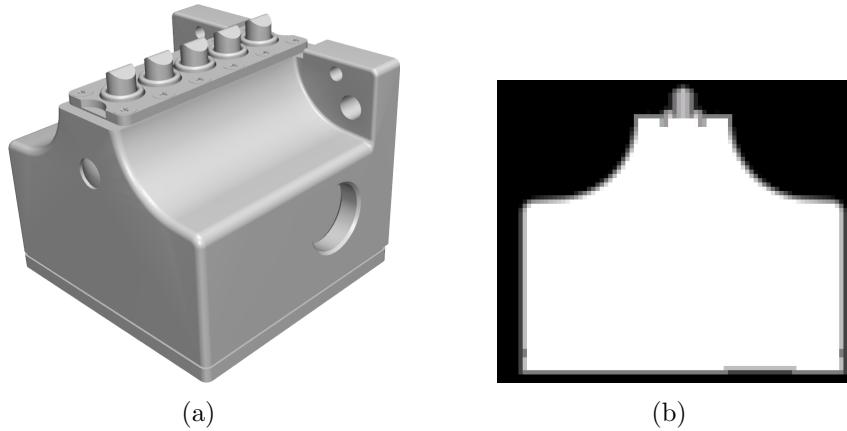


Figure 5.4: CAD model voxelised at 1 mm

preserves the surface values of the volume and therefore the partial volume effect. This is of particular importance as the voxelisation process in ⁺ScanCAD uses the partial volume effect to achieve sub-voxel accuracy, resulting in a better surface representation.

The radius of the kernel dictates the amount of voxels which are effectively removed from the surface of the volume. The dimensions of the kernel can be chosen such that the operation erodes the surface by differing amounts in each direction. To maintain a constant erosion the kernel dimensions must be at the same ratio as the volume's spacing. For convenience this is kept cubic and hence a cubic kernel is always used.

When using standard greyscale erosion the erosion depth must be an integer number of voxels (i.e. a multiple of the volume spacing) due to the constraint that the kernel dimensions must be in whole voxels. For example a volume discretised at 1 mm cannot be eroded by 0.5 mm. It would, however, be possible if the volume had been discretised at a resolution which was a multiple of the desired erosion depth, in this case 0.5 mm. The disadvantage of this solution is that it

requires the domain to be voxelised at a resolution higher than would otherwise be necessary, dramatically increasing the total number of voxels and importantly the time taken to both voxelise and erode the volume.

A more efficient solution can be constructed by extending the greyscale erosion to provide sub-voxel accuracy. As previously noted, ⁺ScanCAD achieves sub-voxel accuracy by using greyscale values to simulate the partial volume effect. A small layer of voxels on the surface of the volume are given greyscale values which reflect their distance to the original surface. These values are interpolated linearly by the marching cubes (and volumetric marching cubes) algorithm in order to place the reconstructed surface with a sub-voxel accuracy. In a similar fashion it is possible to modify the surface greyscale values in a volume such that the reconstructed surface is displaced by an amount less than one unit spacing.

To achieve a sub-voxel erosion with an effective kernel size greater than a single voxel an erosion must be performed using the integer part of the kernel radius. For example, if an erosion of 2.5 voxels is required then the erosion would be 2 voxels using a $5 \times 5 \times 5$ voxel matrix¹. However, rather than replacing the voxel value at the centre of the matrix with the minimum value, the minimum value is scaled linearly by the non-integer part of the kernel radius. This allows for more efficient erosions of arbitrary depth.

Using the sub-voxel greyscale erosion it is now possible to erode the voxelised CAD model to define a region of void within a shell of arbitrary thickness. The resulting volume defines the domain for the internal structure. However, unlike the scaffold structures in §5.1, the internal structure cannot be generated by taking the intersection of the void domain and the lattice volume. Doing so would

¹The radius excludes a central voxel in the matrix, hence the matrix is $5 \times 5 \times 5$ not $4 \times 4 \times 4$.

yield a volume which when iso-surfaced produced a lattice structure bounded by the non-existent shell as shown in Figures 5.5(a) and 5.5(b).

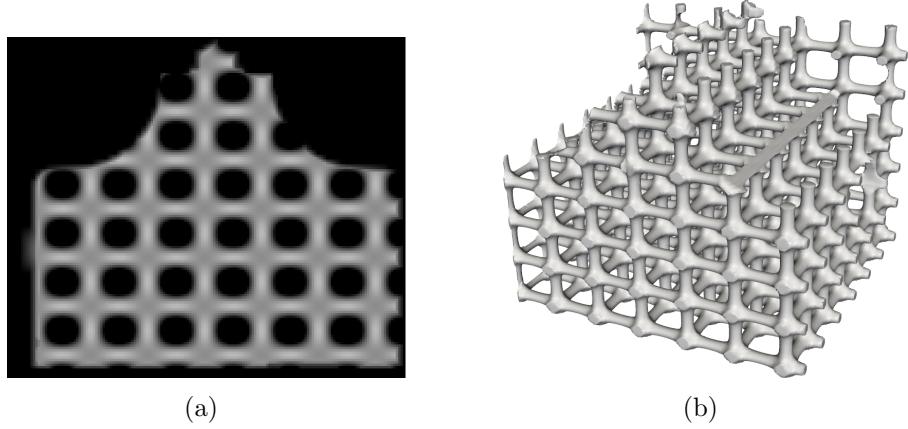


Figure 5.5: Internal structure without a shell

A shell could be added to this volume following a series of Boolean operations. Firstly, the creation of the shell:

$$S = C - E \quad (5.2)$$

where S is the shell volume, C the voxelised CAD model and E the eroded volume. Combining this with the internal structure volume above:

$$I = S \cup (E \cap L) \quad (5.3)$$

where I is the final volume and L the lattice volume. Figure 5.6(a) shows a slice from the volume I using the example CAD model.

It is clear from the reconstructed surface in Figure 5.6(b) that the external geometry of the CAD model is not sufficiently represented at this resolution. While increasing the resolution at which the CAD model is voxelised would better

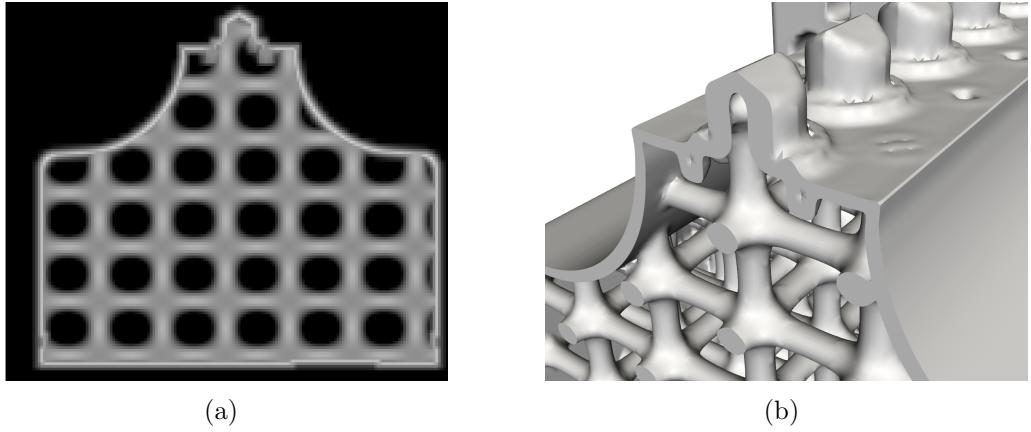


Figure 5.6: Poor surface reconstruction of a CAD model and internal structure

resolve the CAD model’s features, the voxelisation is inherently a lossy process and can never exactly represent the CAD model. For this reason an entirely image-based approach cannot be taken as with the scaffold structures.

While the image-based approach is not sufficient for the CAD’s external geometry it is suitable for the internal surface, which need not exactly reflect the external surface. Features on the external surface should be reproduced exactly, whereas on the internal surface a loss of small features is acceptable. In some instances this may be beneficial as the voxelisation will effectively smooth sharp angles and potentially reduce stress concentrations in the shell. The requirement of a shell ensures that the two surfaces remain disjoint and can therefore be handled independently. By taking advantage of this, the internal surface may be modelled in image-space and the external surface in CAD-space (i.e. a triangulated surface). The two datasets may then be simply appended to produce the final model.

As the external surface is required to remain unaltered only the internal surface need be considered at this stage. The internal surface must be composed of

the inner surface of the shell as well as the internal structure and be constructed such that it may be appended to the external surface to produce the final model. From these requirements it is clear that the volume produced by Eq. 5.3 is not suitable as it explicitly defines the shell, whereas the shell is to be defined implicitly as the volume between the internal and external surfaces. A suitable volume can be constructed using Boolean operations on the same volumes, however. In a similar fashion to the model in Figures 5.5(a) and 5.5(b) the eroded volume is used to bound the lattice volume. However, by first taking the inverse of the eroded volume and combining the result with the lattice volume using a union the inner surface of the shell is produced.

$$I^* = E^{-1} \cup L \quad (5.4)$$

where $^{-1}$ indicates the inverse of a volume. Figure 5.7(a) shows a slice from the volume I^* and Figure 5.7(b) shows the surface reconstructed using the marching cubes algorithm.

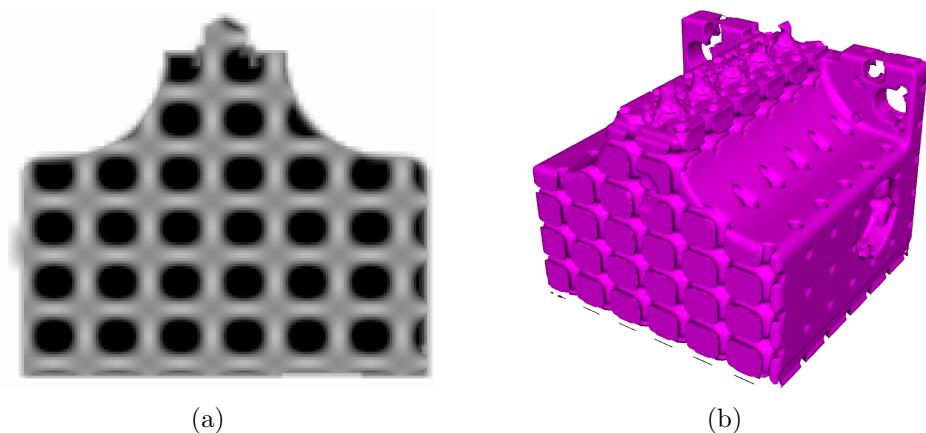


Figure 5.7: Internal surface volume and reconstructed surface

The inversion of the eroded volume causes the marching cubes algorithm to generate triangles with inverted normals as they are generated in the direction of the gradient. These flipped triangles are coloured pink to indicate the backfaces are visible – as in Figure 5.7(b). As the reconstructed surface is actually an internal surfaces in the final model, this effect is desirable to ensure every triangle is correctly orientated. The surface visible in Figure 5.7(b) is the backside of the shell’s inner surface, hence the triangle normals are facing inwards to the centre of the geometry.

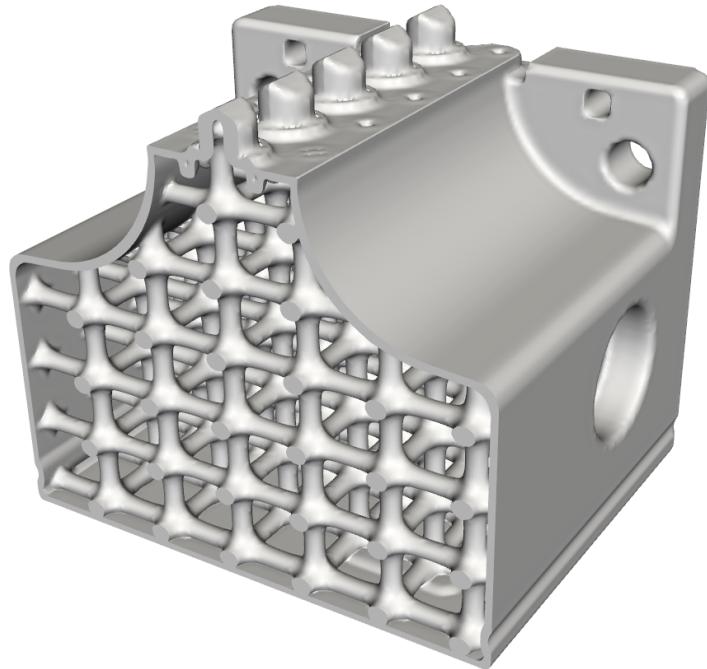
Following the generation of the internal surface, the final stage of the process is to append the external tessellation to produce the final model. As the two surfaces cannot intersect no complex Boolean operations are required and the two datasets are simply appended.

Figure 5.8(b) shows the final model. For comparison Figure 5.8(a) shows model created entirely in image-space.

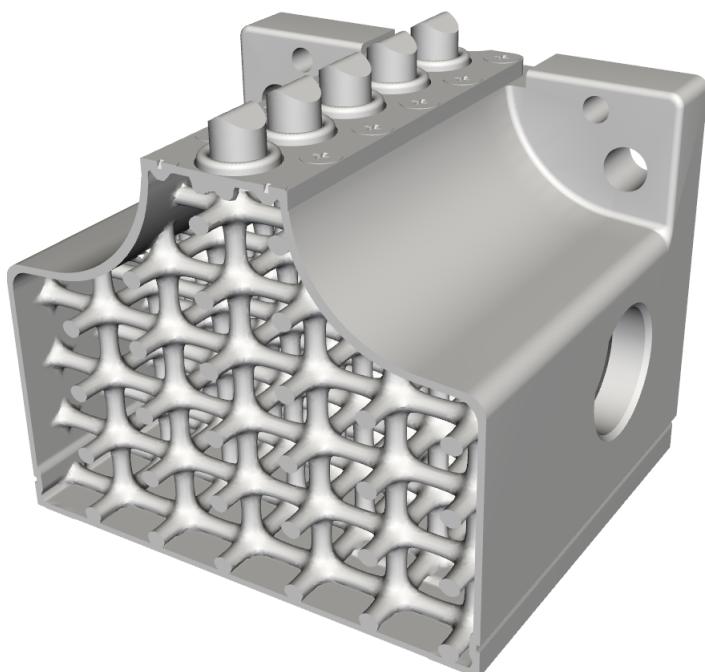
The flowchart in Figure 5.9 shows the entire process of generating an internal structure within a CAD model.

An example of a knee implant with integrated lattice structure is given in Figures 5.10 to 5.13. The implant has been reverse engineered from physical parts so that a lattice structure may be introduced. The purpose of doing so is to allow bone ingrowth into the implant to ensure it remains secured. Prototypes have been produced with the use of a high-resolution 3D printer. They are presented purely as proof-of-concept.

The implant CAD models were created by Mr. Peter Jerrard.



(a) Entirely image-based model. Loss of detail is clearly visible on the external geometry.



(b) CAD-image model. The external geometry is maintained precisely.

Figure 5.8: Cut-away comparison of the entirely image-based approach with the mixed approach

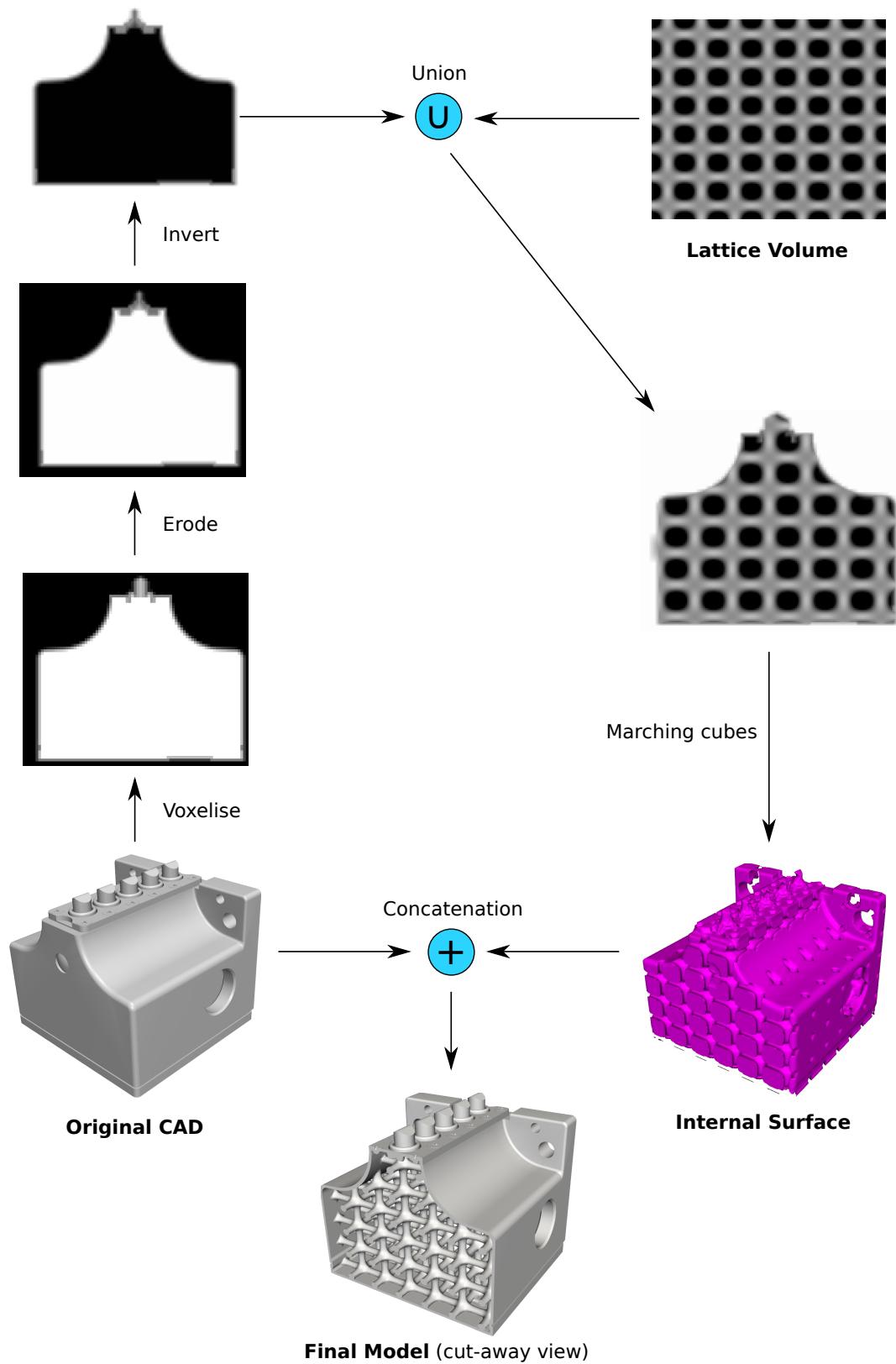
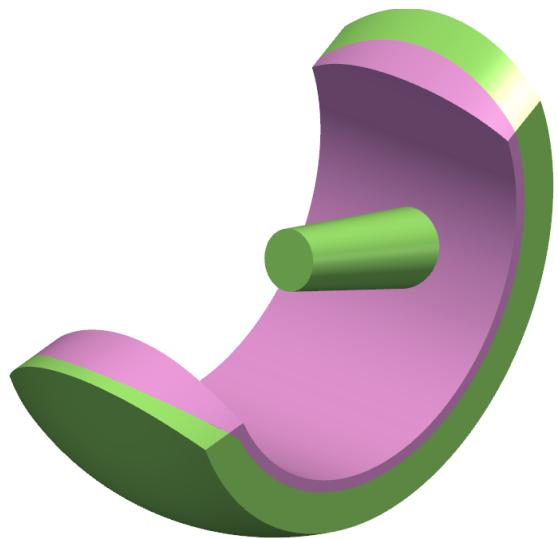
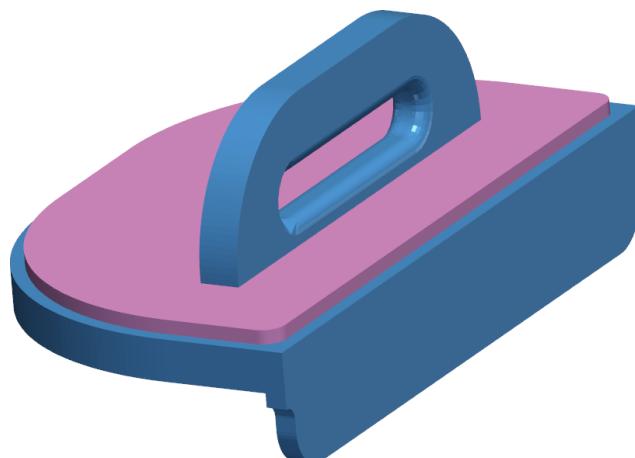


Figure 5.9: The process of generating an internal structure in a CAD model



(a) Condylar implant CAD model. The inner (pink) part is used to define the space the lattice structure will occupy.

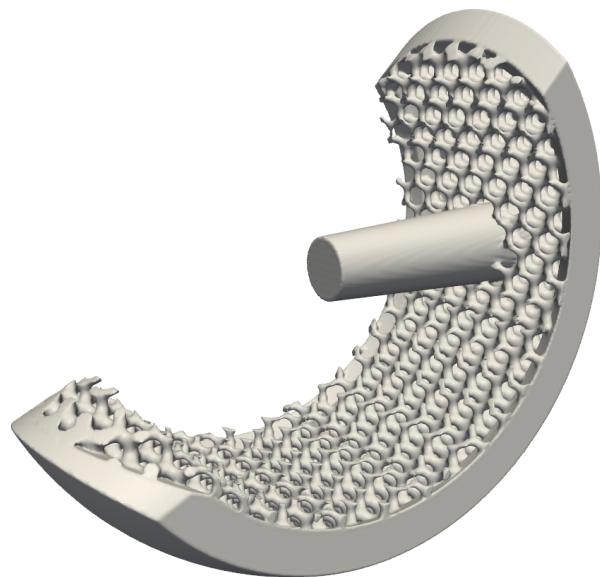


(b) Base plate. The inner (pink) part is used to define the space the lattice structure will occupy.

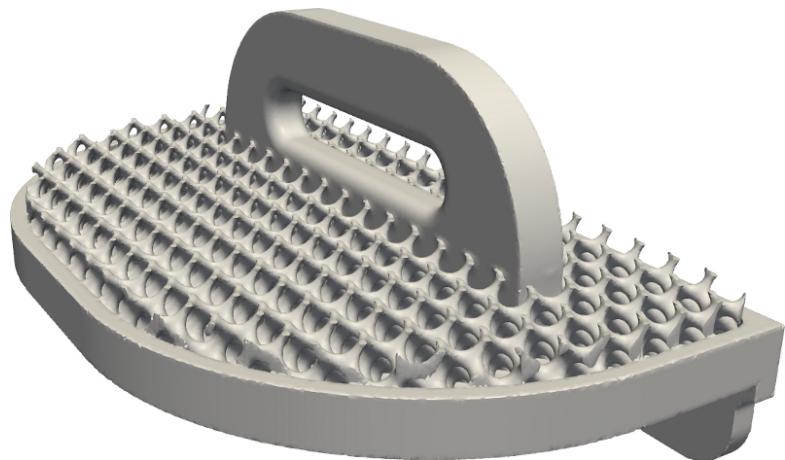
Figure 5.10: Defining the lattice regions in the knee implant model



Figure 5.11: Arrangement of implant parts. Top to bottom: Condylar implant, shim, base plate.



(a) Condylar implant with lattice structure



(b) Base plate with lattice structure

Figure 5.12: Base plate with lattice structure



(a)



(b)

Figure 5.13: Comparison between original parts (grey) and prototypes (blue)

5.2.1 Further Processing

The use of an image-based representation for the internal geometry allows existing image processing filters, such as those available in the ITK framework, to be used to further process the internal geometry. The regular nature of image volumes allows for many well-defined operations to be performed, often in parallel, on the entire domain or a specified region. In comparison equivalent operations on triangulated surfaces, are often more complex.

To reduce the possibility of stress concentrations bevelling is often used to smooth sharp edges in CAD models. Similar results may be achieved by applying a smoothing filter to the eroded volume, consequently de-featuring the internal surface and bevelling edges. The amount of smoothing applied (i.e. kernel size or number of iterations) can be used to control the amount of bevelling. For this reason a kernel based filter appears more suitable as the kernel size can be measured in real-world units. The results of various sized kernels are shown in Figure 5.14.

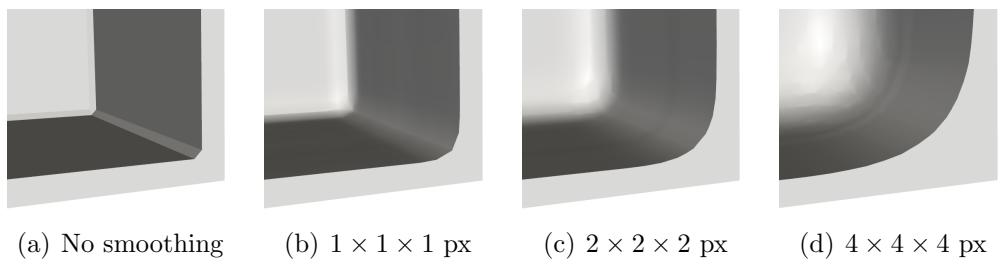


Figure 5.14: The effects of various kernel sizes on corner bevelling

Other common image processing filters, such as morphological closing, may also be applied to the internal volume. The morphological closing filter (dilate followed by a close) can be used to fill-in small voids less than or equal to the

radius of the structuring element. When applied to the internal volume the closing filter can be used to keep small regions solid. Due to the inversion of the eroded volume the closing filter will effectively be operating on the void volume and will hence have an opening effect on the final solid volume. For this reason a morphological opening filter must be applied.

The example in Figure 5.15 demonstrates the effect of applying this filter as well as highlighting the smoothing side-effects.

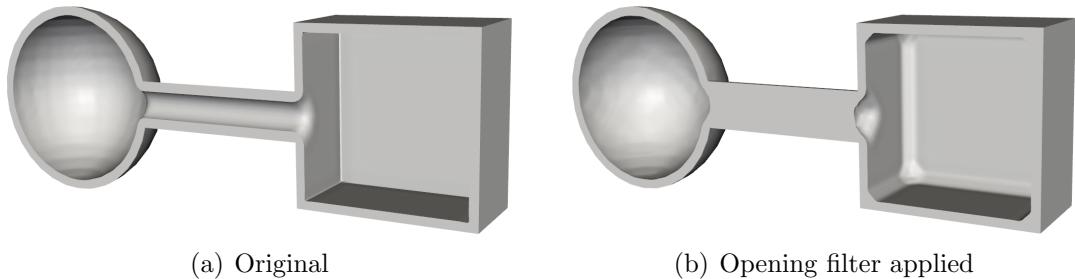


Figure 5.15: Cavity closing using morphological opening

5.3 Implementation

The algorithm for generating internal structures in CAD models, as presented in Figure 5.9, has been implemented in the commercial software ⁺ScanCAD. This particular implementation is based on a ‘wizard’ interface whereby a sequence of dialogues are presented, each allowing a small number of parameter (i.e. unit cell size, shell thickness, etc ...) to be set. Screen-shots are provided in Appendix C.

5.4 Summary

A novel, image-based, approach for the generation of internal structures within CAD based geometries was presented in this chapter. The approach taken relies on sampling the CAD model so that image processing techniques can be applied to create a shell (i.e. hollow geometry). This has the advantage of allowing Boolean operations to be performed in image-space, for the purpose of robustly introducing an internal structure, for example. By exploiting the disjoint nature of hollow triangulated surfaces the CAD model's original external geometry can be precisely preserved.

By providing a set of robust algorithms for the generation of tailored periodic domain-conforming lattice structures the work presented in this chapter represents a strong foundation for any future optimisation work.

Chapter 6

Analysis & Characterisation

6.1 Computational Modelling and Analysis

6.1.1 Background and Literature Review

Computational modelling is an important tool for examining the behaviour of micro-architectures, including those acquired from various imaging modalities as well as computationally generated structures. There are many benefits to this type of modelling, some of which apply generally to computational modelling and some of which are specific to the structures of interest in this work.

Non-destructive evaluation, is, as the name suggests, a general term referring to the analysis of a specimen without causing permanent damage. While this does not refer exclusively to computational modelling, computational modelling is an important tool in NDE. Through the use of such modelling physical samples can be imaged and any number of virtual tests performed. Virtual testing can also be used to test a specimen to failure multiple times, under different conditions.

For certain specimens, applying the desired boundary conditions for testing can be difficult. Computational models have the advantage that virtually any boundary condition, physically permissible or not, can be easily and exactly applied.

When performing physical testing one is seldom able to exactly repeat experimental results. Small changes in the experiment's setup, such as changes in the position of the sample or in the sample itself, cause small variations in results. Computational modelling has the advantage that results are not influenced by changes in the environment and can be exactly repeatable and reproducible.

Clearly, using computationally modelling alone can remove the need for a lot of specialist testing equipment. Modern commercial multi-physics software is capable of performing simulations for a number of physics-based problems beyond simple solid mechanics.

When dealing with structures on the micro scale (of the order of 1 mm) visualising and/or tracking the displacement of individual ribs can be difficult, requiring high resolution video equipment. Computational models, on the other hand, can be visualised using a variety of techniques as well as allowing data to be clipped to reveal 'internal' displacements.

For synthetic or generated micro-architectures computational modelling is particularly beneficial. The ability to evaluate some property of the structure computationally reduces the need to produce physical models and allows many structures to be evaluated iteratively without great cost.

Computational modelling is clearly very suitable for the automation of certain processes. For example, different types of composites and micro-architectures can be computationally generated and evaluated iteratively, such as in an optimisation

process or parametric study.

6.1.1.1 Effective and Apparent Properties

The determination of the effective properties of a multi-phase heterogeneous material, or composite, is an important problem in modern engineering. For this purpose a number of different homogenisation processes have been developed which aim to describe the macroscopic properties of such materials. These methods range from general theoretical bounds, to more specific semi-analytical models, to direct numerical methods. Each of which makes specific assumptions about the morphology, geometry or distribution of microheterogeneities in the material. It is often more accurate to describe the macroscopic properties obtained from homogenisation processes as being the *apparent* properties – i.e. given a set of constraints and assumptions, these are what the macroscopic properties appear to be. These may differ from *effective* properties which is how the sub-volume behaves as part of a larger volume. These effective properties are usually what one strives to obtain from homogenisation.

In this work we limit ourselves to linear elastic properties.

Theoretical Bounds In the most general case upper and lower bounds exist for the macroscopic linear elastic properties of a composite. The most straightforward of which are those given by Voigt [1910] and Reuss [1929], which provide the upper and lower bounds respectively, making no assumptions as to the nature of the structure’s geometry. The Voigt upper bound describes the limit of an effective

elastic modulus, M_V , for a composite of N phases:

$$M_V = \sum_{i=1}^N f_i M_i \quad (6.1)$$

where f_i is the volume fraction of the i th phase and M_i elastic modulus of the i th phase. Here the elastic moduli M_V and M_i may be the Young's modulus, shear modulus or bulk modulus. Similarly, the Reuss bound describes the lower limit of an effective elastic modulus:

$$M_R = \left(\sum_{i=1}^N \frac{f_i}{M_i} \right)^{-1} \quad (6.2)$$

These bounds can be given a physical interpretation as shown in Figure 6.1 (Kassem [2009]). Each figure shows a rod, composed of multiple phases (represented by different colours) stacked either in series or parallel, under tension. In both cases the respective bound will yield the extract solution for the given distribution of phases. It can be shown that the Voigt bound corresponds to the assumption that each phase experiences the same uniform strain (Kassem [2009]). Similarly, the Reuss bound corresponds to assumption that each phase experiences the same uniform stress.

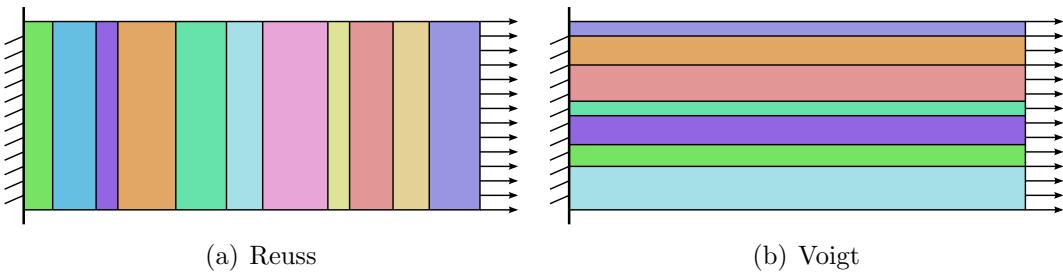


Figure 6.1: Physical interpretation of the Reuss and Voigt bounds. Different colours represent different phases or materials.

The Voigt-Reuss bounds provide a wide range of permissible values for an effective elastic modulus. Another set of bounds which give the narrowest range for an isotropic distribution of phases are the *Hashin-Shtrikman bounds* (Hashin & Shtrikman [1963]).

$$K^{HS\pm} = K_1 + \frac{f_2}{(K_2 - K_1)^{-1} + f_1(K_1 + \frac{4}{3}\mu_1)^{-1}} \quad (6.3)$$

$$\mu^{HS\pm} = \mu_1 + \frac{f_2}{(\mu_2 - \mu_1)^{-1} + 2f_1(K_1 + 2\mu_1)/[5\mu_1(K_1 + \frac{4}{3}\mu_1)]}$$

where K_1 , K_2 are the bulk moduli of each phase, μ_1 , μ_2 the shear moduli of each phase and f_1 and f_2 the volume fraction of each phase. For the case where $K_1 > K_2$ the values of $K^{HS\pm}$ and $\mu^{HS\pm}$ will correspond to the upper bounds. Conversely, for the case where $K_1 < K_2$ the bounds correspond to the lower bounds.

More complex, third order bounds, for random media are described by Beran [1968] and for two-phase composites by Miller [1969], Milton [1982] and Torquato [1991].

Semi-Analytical Methods The bounds obtained for the range of effective elastic properties may have limited uses. Depending on the chosen bounds and relative moduli of the phases the bounds may cover a considerable range with lower bounds potentially reaching zero. In practise one usually desires a single value estimate of effective elastic properties.

More direct approaches have been taken for the estimation of effective elastic

properties of composites with some assumed geometry. For the case of sphere-reinforced composites two independent elastic constants can be estimated using the methods described in Mori & Tanaka [1973] (also Benveniste [1987]), Dvorkin *et al.* [1994], via the self-consistent scheme (Hill [1965]) and via the differential method (McLaughlin [1977]). Such methods have potential practical advantages over finite element based methods, where high mesh densities are required for spherical inclusions.

Direct Numerical Methods Finally, a different method for determining effective properties is through the use of numerical simulations on a sample of the structure. Typically these simulations are performed using finite element analysis and directly model the morphology of the structure. For these simulations there are two important factors which can affect the effective properties; the size of the sample tested and boundary conditions applied.

To ensure the computed properties reflect the actual properties of the composite a sufficiently large sample of the structure is required. This sample is referred to as a *representative volume element*, or RVE. A generally accepted definition for the RVE is a volume V heterogeneous material that is sufficiently large to be statistically representative of the composite (Kanit *et al.* [2003]). Typically this requires that the volume contains a statistically representative sample of all micro-heterogeneities. While there currently does not exist a single strict definition for an RVE, a number are given in the literature.

Drugan & Willis [1996] give the following definition:

“It is the smallest material volume element of the composite for which the usual spatially constant (overall modulus) macroscopic con-

stitutive representation is a sufficiently accurate model to represent mean constitutive response.”

Hill [1963] states:

“...a sample that (a) is structurally entirely typical of the whole mixture on average, and (b) contains a sufficient number of inclusions for the apparent overall moduli to be effectively independent of the surface values of traction and displacement, so long as these values are ‘macroscopically uniform’.”

Evesque [2000] includes an important reference to macroscopic heterogeneity in his definition. Here the term *representative elementary volume* (REV) is used:

“The size of this REV should be large enough with respect to the individual grain size in order to define overall quantities such as stresses and strains, but it should be small enough in order not to hide macroscopic heterogeneity.”

Further definitions may be found in Gitman *et al.* [2007].

As shown in Kanit *et al.* [2003] the size of the RVE is dependent upon the investigated morphology and physical property (i.e. elasticity, thermal, etc . . .). In some instances the size of the RVE may result in a model greater than may be handled with the available computational resources. In these cases it is possible to compute effective properties by averaging over multiple realisations of smaller samples, providing no bias is introduced due to edge effects.

Ostoja-Starzewski [2002] notes that the RVE is ‘well-defined’ in only two cases:

1. A unit cell from a periodic microstructure

-
2. Statistically representative volume containing a very large (mathematically infinite) set of micro-heterogeneities

The required size of the RVE can also be influenced by the boundary conditions applied.

There are three main types of boundary conditions: *kinematic uniform boundary conditions*, *static uniform boundary conditions* and *periodic boundary conditions*. Here we consider a volume V , having a boundary ∂V , consisting of a heterogeneous linear elastic material.

Kinematic Uniform Boundary Conditions (KUBC) The displacement of the entire boundary ∂V is defined as homogeneous, allowing no ‘fluctuations’ (Fritzen & Böhlke [2010]). For each point $x \in \partial V$ a displacement is imposed using a symmetric tensor $\underline{\underline{E}}$:

$$u(x) = \underline{\underline{E}} \cdot \underline{x}, \quad \forall x \in \partial V, \quad u(x)_i = \underline{\underline{E}}_{ij} x_j \quad (6.4)$$

where $u(x)$ is the displacement vector for the point x and the value of $\underline{\underline{E}}$ does not depend on x . This implies that (Kanit *et al.* [2003]):

$$\langle \underline{\underline{\varepsilon}} \rangle \equiv \frac{1}{V} \int_V \underline{\underline{\varepsilon}} dV = \underline{\underline{E}} \quad (6.5)$$

The macroscopic stress tensor may then be calculated using the spatial average of the local stresses:

$$\underline{\underline{\Sigma}} \equiv \langle \sigma \rangle = \frac{1}{V} \int_V \sigma dV \quad (6.6)$$

Static Uniform Boundary Conditions (SUBC) A traction vector $t(x)$ is defined at the boundary:

$$t(x) = \sigma \cdot n = \Sigma \cdot n, \quad \forall x \in \partial V, \quad t(x)_i = \Sigma_{ij} n_j \quad (6.7)$$

where n is a vector normal to ∂V at a point x . This implies:

$$\langle \sigma \rangle \equiv \frac{1}{V} \int_V \sigma dV = \Sigma \quad (6.8)$$

The macroscopic strain tensor is then calculated using the spatial average:

$$\underline{\underline{E}} \equiv \langle \varepsilon \rangle = \frac{1}{V} \int_V \varepsilon dV \quad (6.9)$$

Mixed Uniform Boundary Conditions (MIX) A combination of KUBC and SUBC may be used to more closely represent experimental setups (Jiang *et al.* [2001]). This is the sometimes referred to as a *displacement-traction* boundary condition.

$$(u(x) - \underline{\underline{E}} \cdot x) \cdot (t(x) - \Sigma \cdot n) = 0, \quad \forall x \in \partial V \quad (6.10)$$

Two other possible mixed boundary conditions are *displacement-periodic* and *traction-periodic*. Mixed boundary conditions have the disadvantage that they may only be realised in materials having at least orthotropic elastic symmetry properties (Hazanov & Amieur [1995]).

Periodic Boundary Conditions (PERIODIC) For structures assumed to be infinitely periodic appropriate boundary conditions can be applied such that the RVE can be reduced to a single unit cell. The boundary conditions must be

applied on parallel pairs of surfaces, denoted $k+$ and $k-$ (Pahr & Zysset [2008]).

$${}^{k+}u(x) - {}^{k-}u(x) = \underline{E}\Delta^k x, \quad \forall x \in {}^k\partial V \quad (6.11)$$

where $\Delta^k x$ is a constant distance vector between the two surfaces. To use this condition in a displacement based FE solver it is convenient to move all terms to the left hand side, as this form is often required by commercial finite element packages:

$${}^{k+}u(x) - {}^{k-}u(x) - \underline{E}\Delta^k x = 0 \quad (6.12)$$

However, there are some ‘special cases’ which are more constrained than the majority of nodes on the cuboid’s surface. These are the nodes designated as vertex and edges nodes, both of which are subject to different constraints, the full details of which may be found in Li & Wongsto [2004].

Each of these boundary conditions limits the macroscopic permitted modes of deformation of the volume. Figure 6.2 shows some, exaggerated, examples of the permitted modes of deformation for each set of boundary conditions.

Computing Apparent Properties Given an appropriately sized RVE and chosen boundary conditions, six required simulations can be performed (three compressive, three shear) for the purpose of determining the apparent stiffness matrix, \mathbf{C}^{app} , and/or compliance matrix, \mathbf{S}^{app} . A number of different methods exist for computing these values from the results of the simulations, the most straightforward of which involves computing a set of effective macroscopic stresses and strains (i.e. Σ and \underline{E} for each load case). These averages can be computed

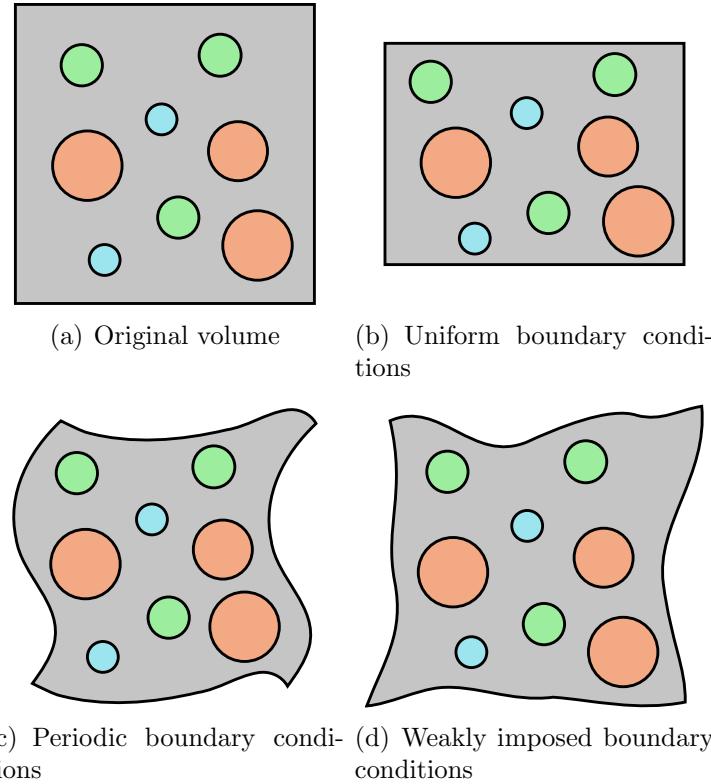


Figure 6.2: Permitted modes of deformation for various boundary conditions

(Pahr & Zysset [2008]) as follows:

$$\underline{\Sigma} = \frac{1}{V} \sum_{n_{IP}} \sigma_{ij} V_{IP}, \quad \underline{E} = \frac{1}{V} \sum_{n_{IP}} \varepsilon_{ij} V_{IP} \quad (6.13)$$

where n_{IP} is the number of integration points and V_{IP} is the integration point volume. Using the sets of effective stresses and strains (where one has been imposed on the RVE and the other computed) a system of linear equations can be constructed to compute either \mathbf{C}^{app} or \mathbf{S}^{app} , depending on the chosen boundary

conditions.

$$\underline{\Sigma}^{\star} = \begin{bmatrix} \Sigma_1 \\ \vdots \\ \Sigma_6 \end{bmatrix}, \quad \underline{E}^{\star} = \begin{bmatrix} E_1 \\ \vdots \\ E_6 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} m_{1,1} & \cdots & m_{1,36} \\ \vdots & \ddots & \vdots \\ m_{36,1} & \cdots & m_{36,36} \end{bmatrix} \quad (6.14)$$

where the matrix \mathbf{M} is the coefficient matrix and $\underline{\Sigma}^{\star}$ and \underline{E}^{\star} represent the concatenation of the averaged stress and strain vectors respectively. For the case of KUBC we then have:

$$x_C = \mathbf{M}^{-1} \cdot \underline{E}^{\star} \quad (6.15)$$

where \mathbf{M} is dependent on Σ^{\star} and x_C is the vector of the 36 values from \mathbf{C}^{app} . Similarly, for the case of SUBC we have:

$$x_S = \mathbf{M}^{-1} \cdot \underline{\Sigma}^{\star} \quad (6.16)$$

where \mathbf{M} is dependent on E^{\star} and x_S is the vector of the 36 values from \mathbf{S}^{app} .

From these equations it becomes apparent that KUBC can be used to directly compute the apparent stiffness matrix while SUBC yields the apparent compliance matrix. While this method is straightforward, it soon becomes computationally expensive for a large number of elements, as is often required for an RVE. More efficient alternatives exist such as the strain energy approach (Ostoja-Starzewski [2006]) and the master node concept (Pahr & Rammerstorfer [2004], Pahr & Rammerstorfer [2006]).

While it is true that, in the most general case, $\mathbf{C}^{-1} = \mathbf{S}$, it is noted in Pahr & Zysset [2008] that it may not be the case that $(\mathbf{C}_{KUBC}^{app})^{-1} = \mathbf{S}_{SUBC}^{app}$. However,

for a sufficiently large RVE the apparent moduli do not depend on the type of boundary condition applied and instead converges to the effective moduli of the material (Sab [1992]). This leads to the following condition:

$$\mathbf{C}_{\text{KUBC}}^{\text{app}} = \mathbf{S}_{\text{SUBC}}^{\text{app}-1} = \mathbf{C}^{\text{eff}} = \mathbf{S}^{\text{eff}-1}, \quad V \rightarrow \infty \quad (6.17)$$

For volumes which are not large enough to reach this criteria the apparent moduli obtained from the two boundary conditions has been shown to bound the effective moduli (Huet [1990]):

$$\mathbf{S}_{\text{SUBC}}^{\text{app}-1} \leq \mathbf{C}^{\text{eff}} \leq \mathbf{C}_{\text{KUBC}}^{\text{app}} \quad (6.18)$$

Kanit *et al.* [2003] notes that PERIODIC solutions also lie within this range. However, no relation between \mathbf{C}^{eff} and $\mathbf{C}_{\text{PERIODIC}}^{\text{app}}$ is given.

6.1.2 Select Cell Characteristics

Using the methods described in the previous section the Schoen Gyroid has been homogenised at a number of different volume fractions. In each case a sample of $8 \times 8 \times 8$ unit cells was considered. It is acknowledged that this sample size may not be sufficiently large to ensure complete convergence to effective properties. However, the purpose of these simulations was to compare apparent properties of the structure under similar conditions. This also includes the apparent properties determined using periodic boundary conditions as they are often regarded as the exact solution for periodic structures.

To ensure that the results achieved sufficient mesh independence a convergence

study was first performed. However, due to the number of elements present in an $8 \times 8 \times 8$ unit cell structure, the KUBC simulations were limited to the highest resolution permitted with 48 GB of memory. Thus, a convergence study was performed only for the PERIODIC solutions, where only a single unit cell is required. A single apparent property, the Young's modulus along the x-axis was chosen for comparison at each resolution. Results are presented in Figure 6.3.

The solid phase was given the isotropic material properties $E = 1$ and $\nu = 0.3$.

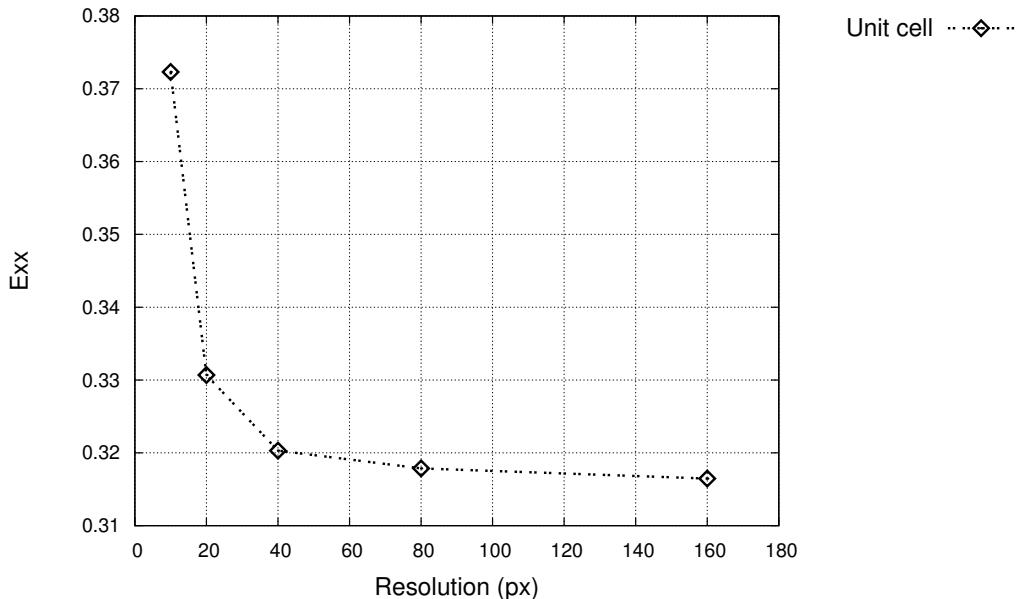


Figure 6.3: Convergence of E_{xx} for the Schoen Gyroid with periodic boundary conditions at 60% volume fraction

From these results we see the rate of convergence is very rapid up to 40 px and much slower for the higher resolution volumes, as may be expected. We therefore choose a resolution of 40 px for the PERIODIC simulations.

The apparent properties of the Schoen Gyroid, as computed using KUBC and PERIODIC, are given in Figure 6.4.

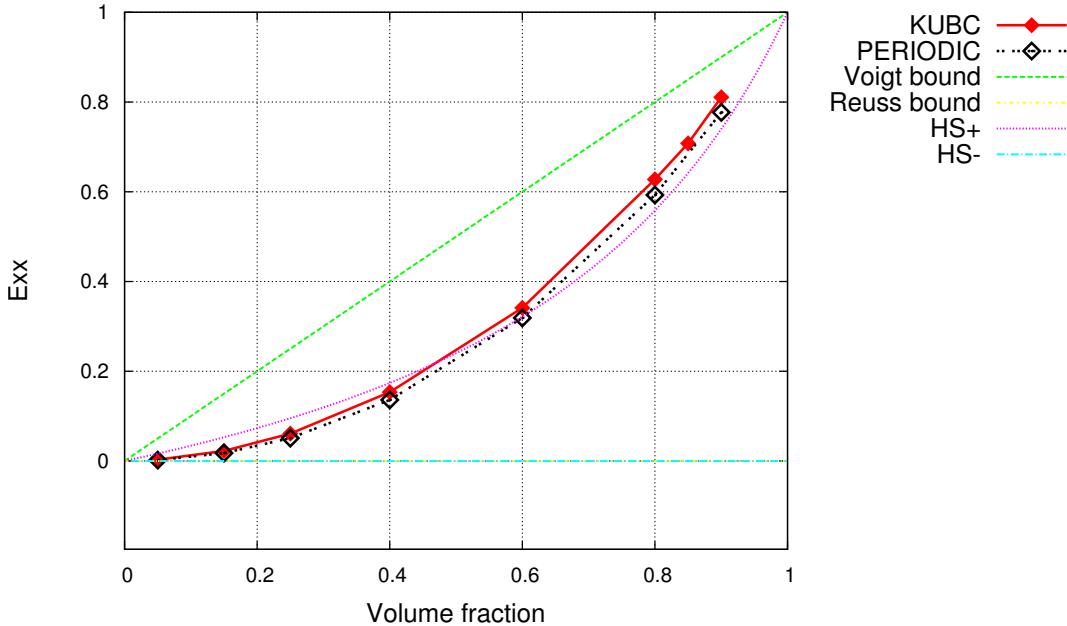


Figure 6.4: Apparent Young's modulus of the Schoen Gyroid at various volume fractions. Computed from a sample equivalent to $8 \times 8 \times 8$ unit cells. As one phase of the structure is void both the Reuss and Hashin-Shtrikman lower (HS-) bounds are equal to 0.

As expected we observe that the PERIODIC predictions are less stiff than those from KUBC, although the predictions are much closer than those in Kanit *et al.* [2003]. This suggests that the sample size used is approaching an RVE. It was also observed that in both cases the predicted apparent stiffness exceeded the Hashin-Shtrikman upper bound for volume fractions greater than 60%. This is likely due to morphological changes in the unit cells as the volume fraction changes, resulting in structures which may no longer be considered an isotropic distribution of material.

6.2 Mechanical Testing

The advantages and uses of computational modelling have been presented in the previous section. These types of models are often based on a number of assumptions, such as the distribution of microheterogeneities or boundary conditions. It is therefore important to perform physical testing for the purpose of validating any computational models.

In this section a number of lattice structures have been fabricated and mechanically tested to determine their effective bulk moduli. A number of internal structures with simple geometries were also tested. The results are compared to the computational models presented previously.

6.2.1 Fabrication

The fabrication of computationally generated micro-architectures is important not only so that we may validate our computational modelling but also so that we may use the structures in real-world applications. However, the potential complexity of the structures makes them difficult, if not impossible, to manufacture using traditional processes. To fabricate these structures we rely on additive manufacturing (AM).

As is suggested by their name, additive manufacturing processes are able to create solid parts by continuously adding material one layer at a time. This is in contrast to the more traditional subtractive manufacturing processes, like CNC (computer numerical control) milling, where material is continuously removed from a solid block until the desired geometry remains.

By constructing geometries layer-by-layer AM processes are able to overcome

many of the limitations of traditional manufacturing processes and fabricate complex geometries. However, many AM processes introduce their own constraints, particularly with respect to overhanging surfaces and supporting structures. For example, metal powders are unable to support large amounts of overhanging surfaces. To overcome this additional supporting structures are introduced. While this does allow more complex geometries to be built, the supporting structures themselves consume additional material and require removal once the build has completed. The amount of overhang permitted is dependent on many factors including the material being processed, the overhang angle and the object's geometry (Thomas [2009]). In contrast, when processing a polymer like nylon in an SLS (selective laser sintering) system structures with any amount of overhang can be fabricated. The unsintered powder on the bed is able to support the solid geometry as it is formed.

In order to fabricate a part using an AM process the desired geometry must be specified in a format which can be translated into a series of instructions required to perform the build. The vast majority of AM system use the STL file format for this purpose. STL files are a basic file format, allowing the definition of triangulated surfaces with little constraint. For example, surfaces need not be manifold. These triangulated surfaces are then contoured at regular intervals, a process known as *slicing*, so that each layer of the build may be defined.

6.2.1.1 Facet Overhang

Before attempting to fabricate a structure from metal (via SLM, Selective Laser Melting) it is often necessary to first ensure that there are no overhanging surfaces, or to add supporting structures to such facets. In this instance the parts are to be

fabricated from stainless steel (316L) using an MCP Realizer; for this combination facets are considered to be overhanging if the angle between the facets and build-plane is below 40 degrees.

For the case of the internal structures it is highly desirable that the lattice structure be manufacturable without the need to introduce supporting structures, as they will be impossible to remove. Thus we now examine a select number of unit cells in order to try and evaluate their suitability for the chosen process. To do so we consider the total surface area considered to be overhanging.

The overhang of a facet can be calculated from the angle between the facet's normal and the plane defining the base of the build:

$$\theta_f = 180 - \left(\mathbf{p} \cdot \mathbf{n}_f \times \frac{180}{\pi} \right) \quad (6.19)$$

where \mathbf{p} is the vector defining the build plane, \mathbf{n}_f the facet normal and θ_f the overhang angle in degrees. The facet normal is defined by the order of the facet's vertices, according to the right hand rule. Using Eq 6.19 we may mark facets on a model as overhanging or not. Figure 6.5 shows an example of this.

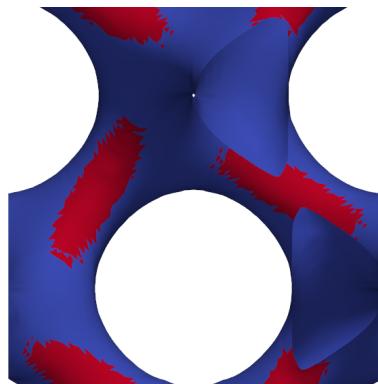


Figure 6.5: Overhanging facets (red) for the Schoen Gyroid at 15% volume fraction. Viewed in the direction of the build (i.e. towards $z = \infty$).

Using this method it is possible to perform an exhaustive search to determine the orientation at which the total overhang surface area is minimised. Doing so will likely yield the most suitable orientation for fabrication of the structure. The search is performed by rotating the plane \mathbf{p} rather than each vertex at each orientation. The surfaces used are non-manifold to ensure the closing ‘caps’ do not influence the results.

Unit Cell	Min		Max	
	Rotation	Overhang	Rotation	Overhang
Gyroid	(356.4, 180)	1630	(0, 313.2)	3167
Diamond	(349.2, 187.2)	2981	(225, 180)	3643
Neovius’	(207, 10.8)	3582	(181.8, 136.8)	4256
W	(300.6, 144)	1698	(226.8, 100.8)	2171
Pinched Primitive	(289.8, 316.8)	1540	(248.4, 194.4)	2371

Table 6.1: Orientations for minimising and maximising total overhang area for various unit cells. Rotations are given as a roll-pitch pair, in degrees. Overhang is the total overhang area measured in units².

From the results presented in Table 6.1 it is clear that none of the selected unit cells have an orientation at which none of their facets are considered overhanging. This, however, does not preclude them from being fabricated. While overhanging surfaces are generally avoided small amounts of overhang are acceptable. Furthermore, facet overhang itself does not completely dictate whether or not a part will build successfully without additional support or not. To be certain the structures are suitable for this process we must attempt to fabricate samples.

It is highly desirable that the structures be manufacturable at all orientations, without support. Parts are often rotated for optimal positioning in the build, to minimise the amount of supporting structures required or to improve the surface finish (Xu *et al.* [1999]). Thus, it is useful if the internal structure in a part does

not in anyway limit or hinder the build in this respect. To test this we attempt to fabricate a number of unit cells at the orientation at which the total overhang area is maximised. The purpose of doing so is that if they build successfully at this orientation, they will likely build at all other orientations. The results are presented in Figure 6.6. The samples were fabricated in stainless steel (316L) by Dr. Chunze Yan using an MTT SLM Realiser with the following process parameters:

Parameter	Value
Laser power	95 W
Scan time per point	250 μ s
Point distance	40 μ m
Scan spacing	75 μ m
Layer thickness	75 μ m

Table 6.2: SLM process parameters

From these results it is clear that each of the chosen unit cells has successfully built using this process. While not definitive evidence that the structures will build at all orientations (for the chosen cell size), the results are very encouraging. Further samples of each unit cell at various orientations, volume fractions and cell sizes would need to be fabricated to ensure more definitive results. The vast number of possible combinations precludes a more exhaustive set of experiments from being performed.

6.2.1.2 Use as Supporting Structures

As the lattice structures have been shown to be manufacturable on their own we must now verify that they can support a surface. This is important as when the structures are used as internal structures it is highly desirable to avoid supporting

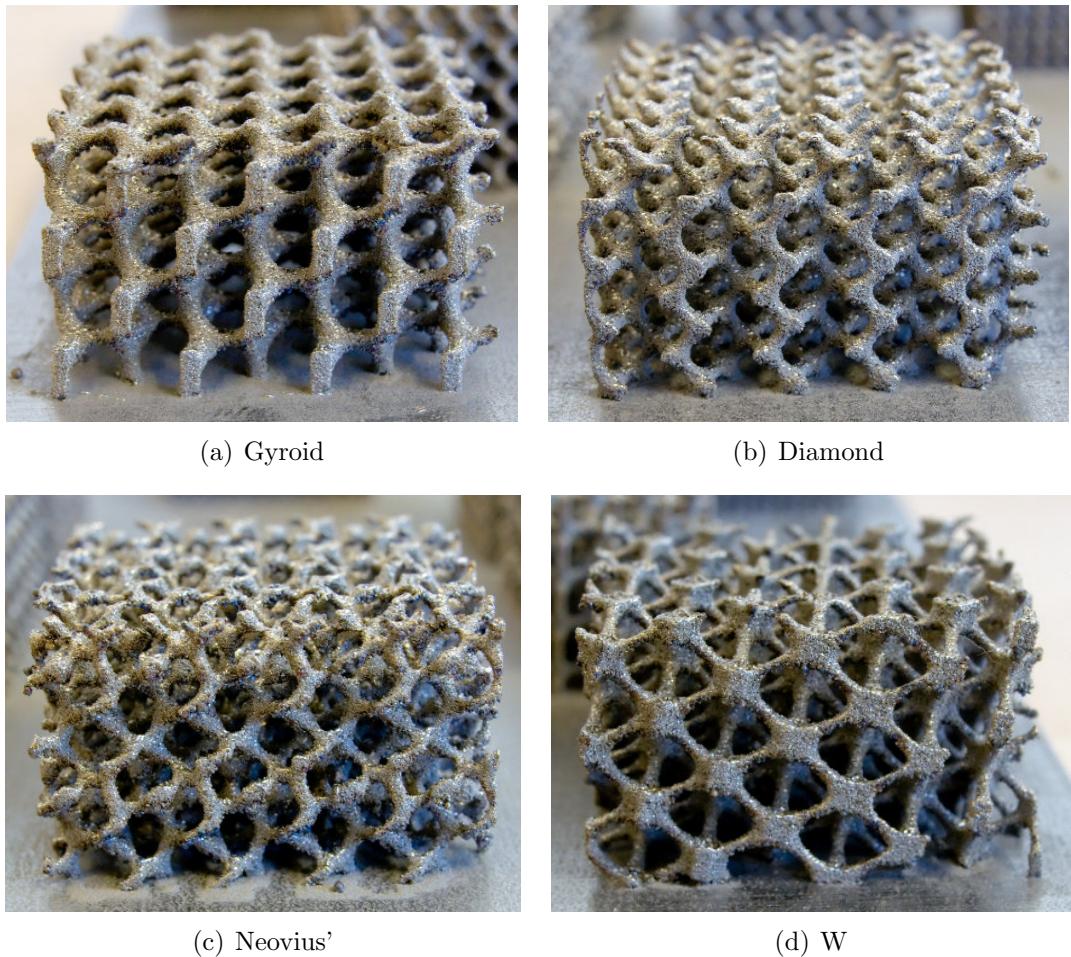


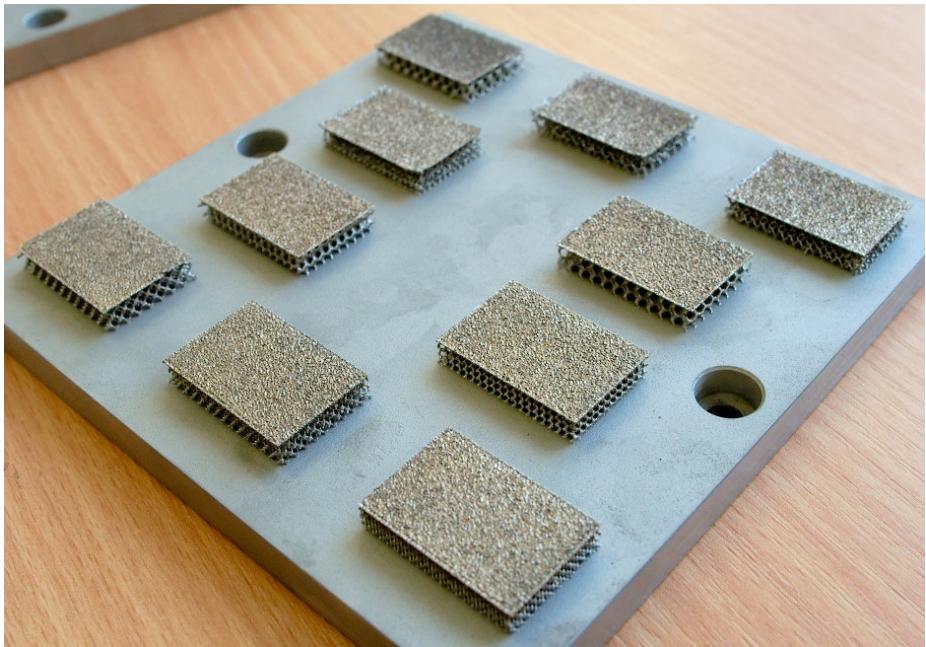
Figure 6.6: Fabricated lattice structures. Each sample measures $25 \times 15 \times 25$ mm ($W \times H \times D$), with 5 mm unit cells at 15% volume fraction. Each unit cell has been orientated to maximise the total overhang surface area.

structures not only on the lattice but also the inner surface of the shell. In this situation the lattice structure would itself be acting as a supporting structure for the shell.

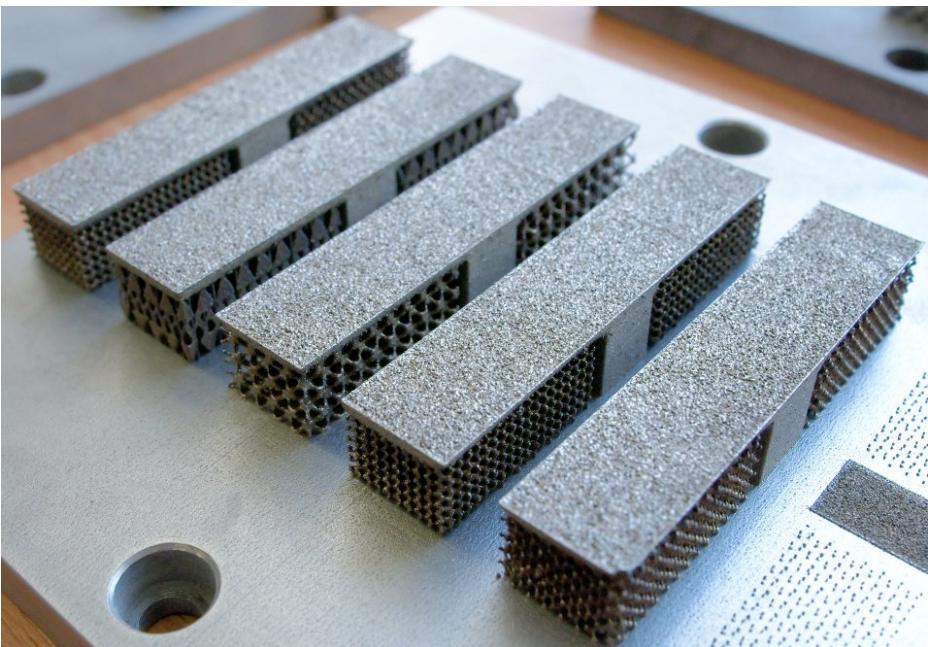
A number of small samples have been fabricated to verify the supporting capabilities of the lattice structures. A 3 mm thick plate is added to the top of each structure to act as the shell of a component. As a small amount of overhang, 0.8 to 1 mm, is permitted unit cells of different sizes have also been tested. These samples are shown in Figure 6.7. The samples were designed and fabricated in stainless steel (316L) by Ahmed Hussein using an MTT SLM Realiser with the process parameters given in Table 6.2.

The chosen lattice structures, at the specified cell sizes and volume fractions, are clearly capable of acting as basic supporting structures. This is particularly advantageous as it means the internal structures can become truly multi-functional. The structures can therefore serve different purposes, such as both to increase a component's stiffness and to act as internal support to aid in fabrication.

As these lattice structures are able to function as internal supporting structures another obvious application worth consideration is their use as *external* supporting structures. The generation of external supporting structures is a non-trivial task due to the many different cases encountered, and is not an objective of this thesis. However, simple external support structures are essentially domain conforming lattice structures. Thus, the methods developed in this thesis may be applied once an suitable domain is established. An example is presented in Figure 6.8. For this example the domain for the lattice structure has been constructed by ‘casting down’ from overhanging surfaces to either another point on



(a) Samples of the Gyroid, W, Diamond and Neovius' structures with 2 mm cell sizes supporting a thin plate. Unit cell volume fraction is 15%.



(b) Left to right: Diamond, Magics support structure, W, Gyroid, Neovius'. Unit cell size is 3 mm, with a volume fraction of 15%. Each sample is 70 mm long, with two 30 mm cantilevers.

Figure 6.7: Fabricated supporting structures.

the surface or the base of the build (i.e. $z = 0$).

6.2.2 Mechanical Testing – Internal Structures

The main purpose of introducing an internal lattice structure within a geometry is to try improve the general stiffness of the part. Clearly for a specific set of loading conditions there will exist a more appropriate internal structure where material is distributed along the direction of loading paths (as would result from topology optimisation), compared to the generally isotropic distribution seen in the lattice structures of interest. There may, however, be additional applications for internal lattice structures, as previously noted.

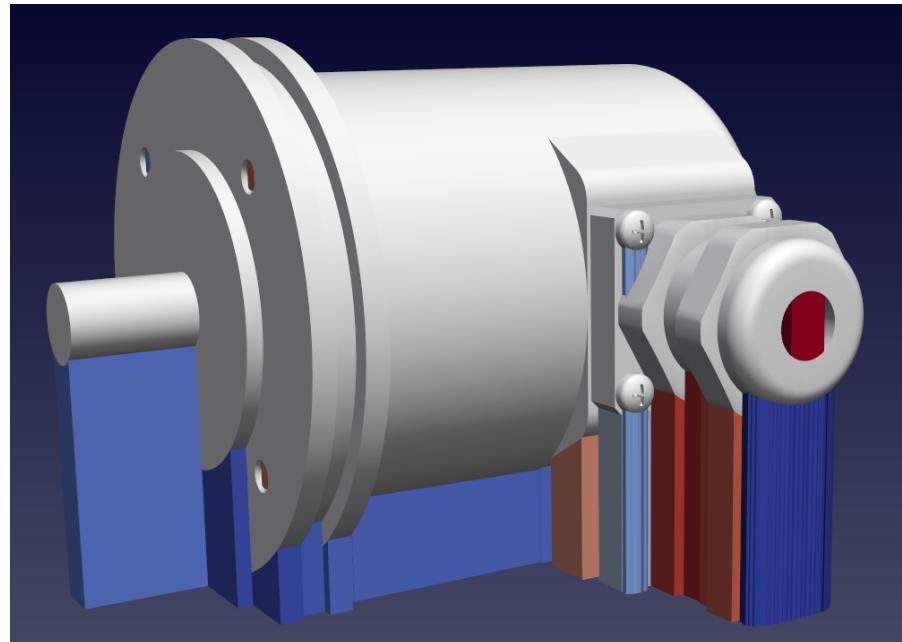
To test the effect of introducing an internal lattice structure three geometries are considered; a sphere, a box and a cylinder. For each geometry an internal structure version and equal volume hollow part is generated. Figure 6.9 shows these parts. Dimensions of the parts are given in Table 6.3.

Component	Dimensions (mm)	Shell thickness (mm)
Box	$75 \times 75 \times 37.5$	3.1
Sphere	$r = 25$	2.9
Cylinder	$r = 30, l = 65$	4.6

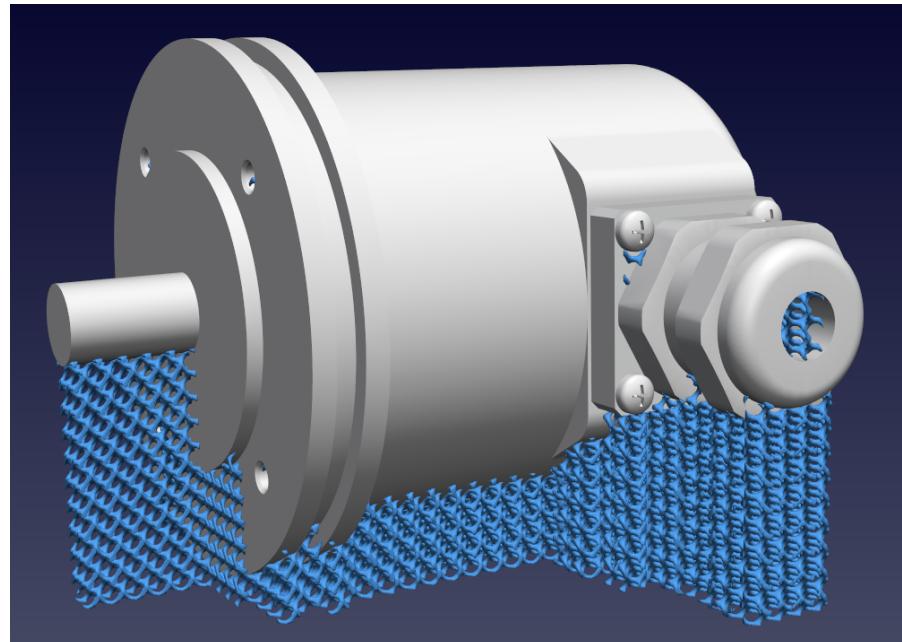
Table 6.3: Dimensions of the hollow components

Each internal structure version of the components has a 2 mm shell. The lattice structure chosen is the Pinched Primitive at 15% volume fraction with a cell size of 1 cm. As is clear in Figure 6.9, the cylinder is open-ended so that when loaded along its length the end faces would not become the supporting element in the structure.

The six components were fabricated using SLS and made from nylon-12.



(a) CAD model with solid support structure.



(b) CAD model with lattice supporting structure.

Figure 6.8: CAD model with supporting structures. Support is added to any facet with an overhang of less than 40 degrees.

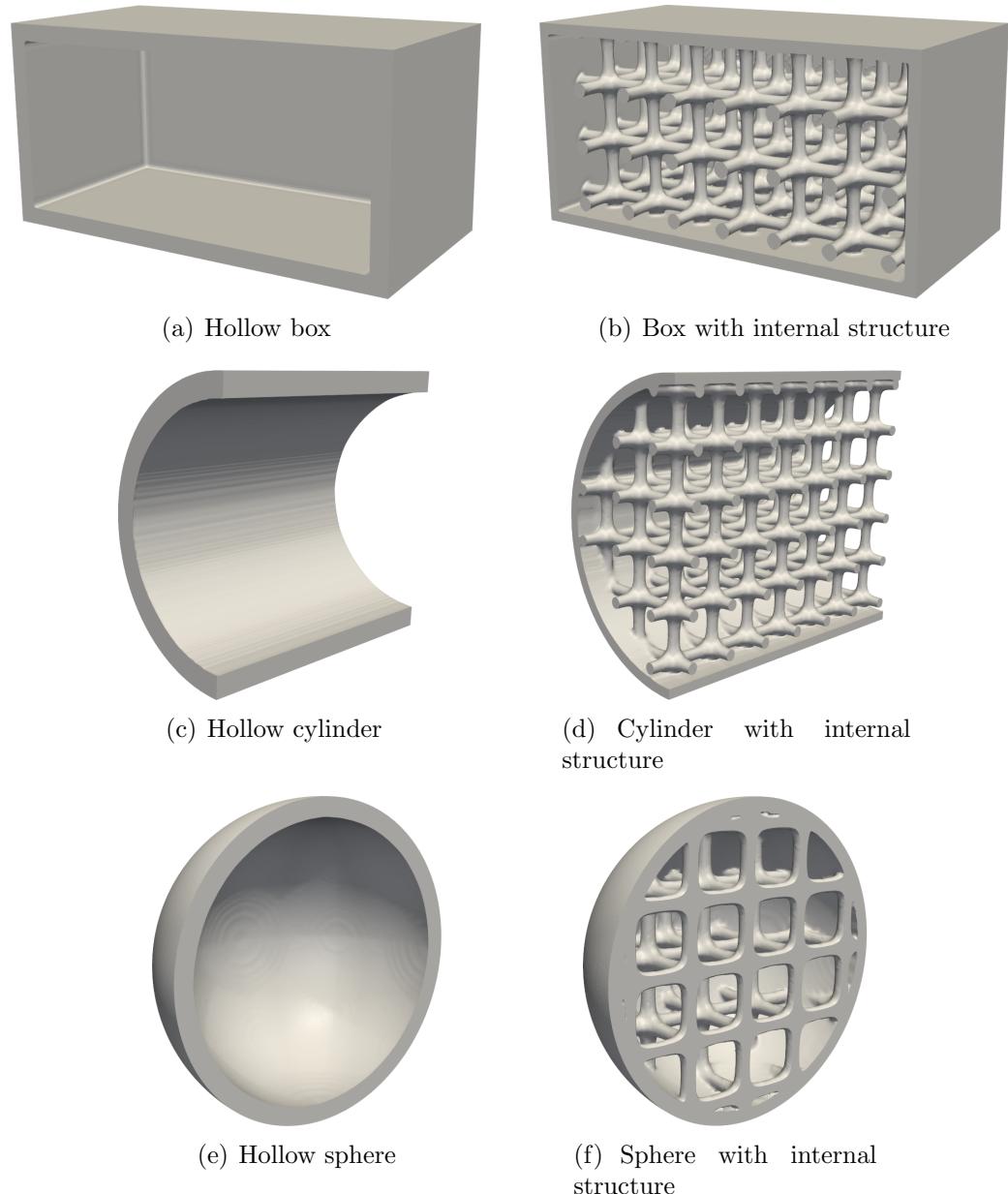


Figure 6.9: Clipped view of the chosen geometries, both hollow and with an internal structure, for testing

Each component was subjected to uniaxial compression using a Lloyd Instruments 20kN EZ20. Images of the experimental setup may be found in Appendix D.

6.2.2.1 Results

To ensure the displacement measured for each component was not spurious each loading was repeated ten times and an average was taken. The results of the physical testing are given in Table 6.4. A comparison of these results to those obtained via finite element analysis is given in Figure 6.10. The mechanical testing was performed by Dr. Chunze Yan.

Component	Applied force (N)	Displacement (mm)
Box (hollow)	20	0.148
Box (internal structure)	20	0.073
Sphere (hollow)	40	0.101
Sphere (internal structure)	40	0.121
Cylinder (hollow)	40	0.188
Cylinder (internal structure)	41	0.130

Table 6.4: Results from mechanical testing

6.2.3 Discussion

From the results presented in the previous section is it clear that, in the majority of cases, the components with an internal structure were stiffer than their hollow equivalent. However, it is also clear that there is large discrepancy between the finite element analysis and physical models. This discrepancy includes the absolute displacements as well as the relative displacements (i.e. the ratio of the displacement of the hollow component to the internal structure version). In the worst case it is observed that, during the physical testing, the hollow sphere

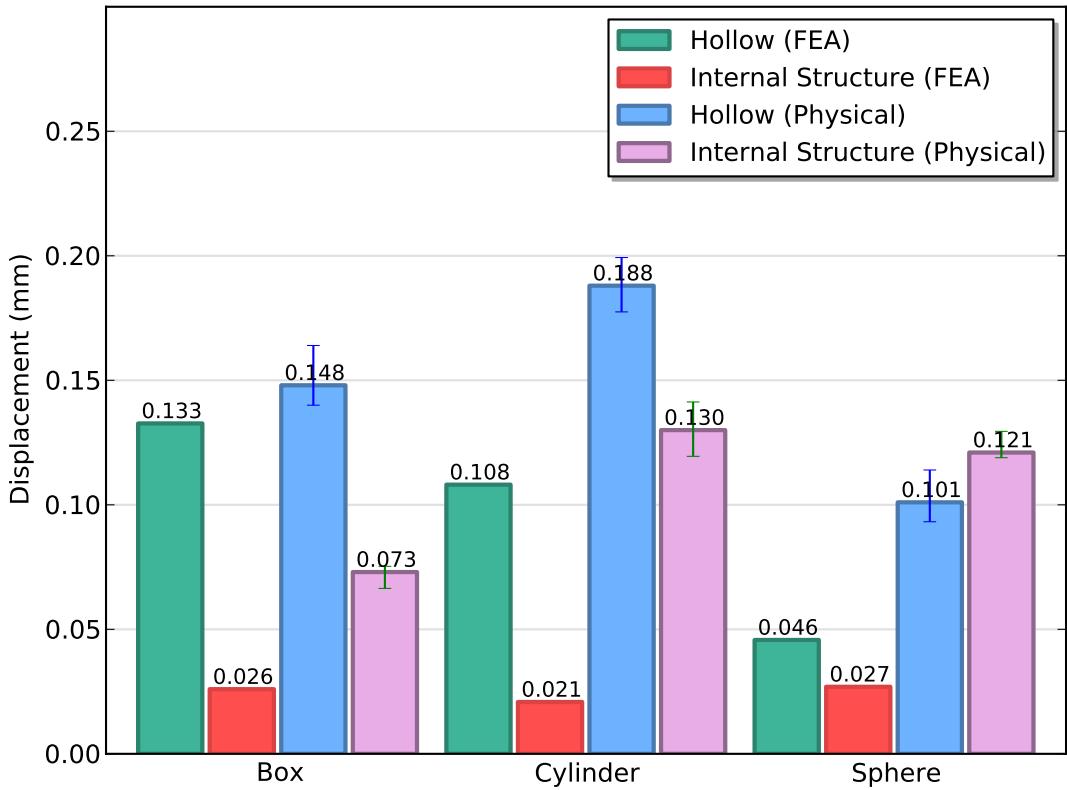


Figure 6.10: Maximum displacement of loaded test parts. A comparison of the finite element and physical model results.

proved to be consistently stiffer than the equivalent internal structure version. This is in direct contrast to the results predicted in the FE simulations. The discrepancies observed in the results may be due to a number of possible sources such as variability in the fabrication process or incorrect assumptions made in the FE models. It is well known that there exists a degree of variability in ALM processes which may cause parts to be built with varying material properties or geometry, depending on many factors such as build position. While such variations may affect the outcome of the experiments it is not expected that their influence would be sufficient to cause the observed discrepancies.

In order to validate the finite element modelling it is possible to use an analyt-

ical model to determine the maximum displacement of the component. For this, the model developed in Young [2003], originally for modelling blunt head impacts, is chosen. The model combines Hertzian contact stiffness and shell stiffness to determine an effective stiffness for a thick shell.

Reissner [1947] gives the stiffness for a shell, k_{sh} , as follows:

$$k_{sh} = 2.3 \frac{E_{sh} h^2}{R_{sh} (1 - \nu_{sh}^2)^{1/2}} \quad (6.20)$$

where E_{sh} and ν_{sh} are the Young's modulus and Poisson's Ratio of the shell and R_{sh} and h are its radius and thickness.

The Hertzian contact stiffness (Hertz [1882]), k_H , is calculated as follows:

$$k_H = \frac{4}{3} R^{*1/2} E^* \quad (6.21)$$

where R^* is the radius of the shell and E^* its Young's modulus. In this instance $R^* = R_{sh}$ and $E^* = E_{sh}$.

From Young [2003] we can now determine the effective stiffness, $k_{\text{effective}}$:

$$k_{\text{effective}} = \left(\frac{1}{k_{sh}} + \frac{1}{k_H^{2/3} F^{1/3}} \right)^{-1} \quad (6.22)$$

where F is the applied force. Substituting the appropriate values into Eq 6.22 we find $k_{\text{effective}} = 8.799 \times 10^5$ N/m. Using Hooke's law the maximum displacement of shell can be calculated:

$$\Delta x = \frac{F}{k_{\text{effective}}} \quad (6.23)$$

where $F = 40$ N. This gives $\Delta x = 0.045459$ mm. In comparison the FE sim-

ulations predicted $\Delta x_{\text{FEA}} = 0.045731$ mm. The good agreement between the analytical and FE solutions suggests that some of the assumptions made in the FE modelling, such as boundary conditions, are suitable and that the simulations are valid at least for the linear range.

One possible explanation is that for these very small displacements (where Δx is of the same order of magnitude as the nylon particle size) we are only seeing the deformation of the surface layer of particles. As such, it may no longer be appropriate to consider the material as homogeneous, as in the FE simulations.

It is, however, more likely that the discrepancies observed are due to plastic deformation. In the case of the two curved surfaces (the cylinder and sphere) the contact area between the load cell and object will have initially been small and grown as the load was applied. With a small load area there is the possibility that the resulting stresses were greater than the yield limit of the material. This theory is supported by the good agreement between the analytical solution for the hollow sphere and FE results, both of which modelled the nylon as purely elastic.

6.3 Conclusions

In this chapter we have reviewed the techniques currently used to determine the apparent properties of sub-volumes or, as in this instance, large samples or lattice structures. We have observed the influence of different boundary conditions on the apparent properties of the Schoen Gyroid and obtained results in agreement with observations found in literature. That is, that the kinematic uniform boundary conditions not only required large samples to overcome edge effects, but

also over-predicted the stiffness of the structures in comparison to the periodic boundary conditions. Clearly, for the structures studied in this thesis periodic boundary conditions are by far the most appropriate method of determining apparent properties.

The suitability of additive manufacturing processes for fabricating these lattice structures in a variety of different materials has also been demonstrated. We have shown how these periodic lattice structures are manufacturable at any orientation, allowing them to not only be used as an internal supporting structure (for hollow parts), but potentially also as external supporting structures. The work presented in this thesis for this purpose represents only the initial findings and more research into this application is certainly required. In particular, how variations in the external supporting structure's density can ensure certain areas are stiffer than others to better support a given part while ensuring the structure is removable.

A number of simple geometries with internal structures were fabricated and mechanically tested. The results were compared to finite element simulations of the same geometries and found to be in poor agreement. By comparing the finite element predictions of one geometry, the hollow sphere, we were able to validate the consistency of the models under the assumptions made. However, further work is required to ensure the finite element predictions better match those measured experimentally.

Chapter 7

Homogenisation

7.1 Concept Overview

When dealing with micro-architectures one issue likely to eventually arise is the analysis of the mechanical properties of macroscopically inhomogeneous multi-scale structures. Structures of this class may be naturally occurring, such as bone, or computationally generated using methods such as those described in Chapter 3. The generated micro-architectures are an obvious target for numerical optimisation due to their flexibility. The optimisation of these structures will inevitably introduce macroscopic inhomogeneities at some stage in the process. The bulk response of these structures can be determined by performing ‘full’ finite element analysis, that is with the entire geometry discretised at a resolution high enough to ensure mesh independence. However, these full models may easily exceed hundreds of millions, potentially billions, of degrees of freedom. A problem only exacerbated by the fact that the number of degrees of freedom grows with the cube of the resolution. Solving problems of this magnitude is possible with

the use of super computing facilities, although they will likely require hundreds of hours of CPU time. For a very limited number of simulations this may be an acceptable solution, with the added advantage of also capturing localised effects. However, in an iterative optimisation process where the ‘performance’ of the structure may be evaluated thousands of times the use of full FEA simulations becomes highly impractical. When the performance of a structure is evaluated in an optimisation process typically only some aspect of the bulk response, such as deflection, is considered. For such properties full FEA simulations model the problem may be excessive. Thus, there is a need for a method to approximate the models in order to reduce the time required to evaluate the structure.

For many applications where the micro-architecture is macroscopically homogeneous there will exist a representative volume element (RVE) which may be tested using appropriate boundary conditions to determine the effective properties, as demonstrated in Chapter 6. These effective properties may then be assigned to the elements in an appropriately coarse discretisation of the domain. For example, when modelling the deformation of polyurethane seating it would be highly impractical, if not impossible, to attempt to include all local deformations. Instead an approximate model consisting of a few hundred thousand elements can be used.

This chapter presents a novel method for the generation of such a model. The approach taken is based on creating a coarse tetrahedral discretisation of the domain using traditional volume meshing techniques and assigning appropriate material properties based on the homogenisation of the tetrahedrons’ contents.

A similar approach is taken in medical imaging with some natural structures, such as bone, where the Young’s modulus is expressed as a function of

the Hounsfield value (a measure of radiodensity) from the image data. The relation between the two values is commonly determined through mechanical testing. However, this approach is limited to isotropic material properties and even for the well-studied femur there are considerable variations in the relation in the literature (e.g. Heymans *et al.* [1977] and Rho *et al.* [1995]). This method also requires some prior knowledge of the behaviour of the structure.

The approach taken in this work is based on treating sub-volumes of the structure as actual elements and, through a series of tests, inferring appropriate effective material properties.

7.2 2D

To demonstrate the proposed homogenisation technique we first concentrate on developing it for 2D problems. While there may not be demand for producing 2D approximate models, as very large 2D linear problems can be solved with modest hardware requirements, the simplicity of 2D problems provides a good starting point for explaining the proposed technique. There are, however, other possible uses for this technique which may be useful in 2D, as will be shown later in this chapter.

7.2.1 Constitutive Matrix Recovery from a 3 Noded Triangle Element

To highlight the basic principle of the proposed homogenisation method this section demonstrates how the material properties of a simple 3-node element can be recovered by way of virtual testing. This in itself has little to no direct practical

use, but the principle is fundamental to the proposed homogenisation method.

The Constant Strain Triangle (CST) element for plane stress, as shown in Figure 7.1, is chosen due to its simplicity.

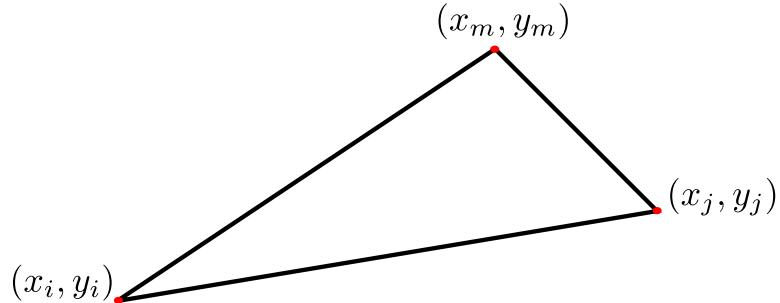


Figure 7.1: A 3-node constant strain triangle element. Nodes are numbered anti-clockwise

We know that the material matrix and element geometry determine the behaviour of the element. This is clear from how the element's stiffness matrix is formed:

$$\mathbf{K} = t \mathbf{AB}^T \mathbf{D} \mathbf{B} \quad (7.1)$$

where A is the area of the triangle, \mathbf{D} the *constitutive* (or *stress/strain*) matrix and t the thickness of the element (assumed to be equal to 1 hereon in). The \mathbf{B} matrix is constructed from the element's shape functions, a set of linear displacement functions.

Area of a triangle Twice the area of the triangle can be computed using the following determinant:

$$2A = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix} \quad (7.2)$$

Constitutive matrix The constitutive matrix describes the relationship between stress and strain in the element:

$$\sigma = \mathbf{D} \cdot \varepsilon \quad (7.3)$$

For the case of plane stress, an isotropic material in 2D is defined as:

$$\mathbf{D} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (7.4)$$

where E is the Young's modulus and ν the Poisson's ratio. Due to symmetry in the matrix there are only six independent components and hence six unknown values to find for the case of general anisotropy.

B Matrix The matrix \mathbf{B} is constructed as follows:

$$\mathbf{B} = \frac{1}{2A} \begin{bmatrix} \beta_i & 0 & \beta_j & 0 & \beta_m & 0 \\ 0 & \gamma_i & 0 & \gamma_j & 0 & \gamma_m \\ \gamma_i & \beta_i & \gamma_j & \beta_j & \gamma_m & \beta_m \end{bmatrix} \quad (7.5)$$

where

$$\begin{aligned} \beta_i &= y_j - y_m, & \beta_j &= y_m - y_i, & \beta_m &= y_i - y_j \\ \gamma_i &= x_m - x_j, & \gamma_j &= x_i - x_m, & \gamma_m &= x_j - x_i \end{aligned} \quad (7.6)$$

In this problem we have a triangle element of known geometry consisting of an unknown (assumed) homogeneous material. As previously stated, the aim of this exercise is to demonstrate that the material properties of the element

can be recovered using a series of virtual tests. The only tests which can be performed using the finite element method in this instance involve either applying displacements or forces to nodes.

We know that applying a displacement to one of the element's nodes will result in a force, as described by Hooke's Law:

$$\mathbf{f} = \mathbf{K} \cdot \mathbf{u} \quad (7.7)$$

where \mathbf{u} is the displacement vector. As this is more straightforward than applying forces to nodes (\mathbf{K}^{-1} need not be computed) the virtual tests will be described in terms of nodal displacement. With a known displacement vector the only values which remain unknown are those in the constitutive matrix \mathbf{D} , as expected. Thus, it is possible to express the nodal forces in terms of \mathbf{D} . For this we use the notation in Eq 7.8 to indicate the forces \mathbf{f} are a function of the constitutive matrix \mathbf{D} and the displacement vector \mathbf{u} . While the forces are clearly also a function of \mathbf{B} and A (i.e. the element's geometry), we omit these expressions in favour of compactness as they remain invariant throughout this section.

$$\mathbf{f}_n(\mathbf{D}, \mathbf{u}_n) \quad (7.8)$$

Here subscript n indicates the test number. Using the symbolic computing features in the commercial CAS (computer algebra system) package Maple (Map [2011]) it is possible to determine the full expression for the term in Eq 7.8. However, even with this simple 2D element the full expression is far too long to be given here.

We know that the value of \mathbf{f}_n must be equal to the force vector \mathcal{F}_n measured from the virtual tests. Thus we have:

$$\mathbf{f}_n(\mathbf{D}, \mathbf{u}_n) = \mathcal{F}_n(\mathbf{u}_n) \quad (7.9)$$

where $\mathcal{F}_n(\mathbf{u}_n)$ is the vector of measured forces for the displacements \mathbf{u}_n . The expression in Eq 7.9 will provide a system of linear equations with a set of unknowns. However, by evaluating the expression in Eq 7.8 symbolically with a simple displacement vector it becomes clear that the forces are not a function of all the values in the constitutive matrix \mathbf{D} . The table below shows a selection of displacement vectors and the corresponding constitutive components which effect the resulting forces.

Test Number	Displacement Vector	Constitutive Components
1	$[1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$	$\mathbf{D}_{11}, \mathbf{D}_{21}, \mathbf{D}_{31}, \mathbf{D}_{32}, \mathbf{D}_{33}$
2	$[0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$	$\mathbf{D}_{21}, \mathbf{D}_{22}, \mathbf{D}_{31}, \mathbf{D}_{32}, \mathbf{D}_{33}$
3	$[0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$	$\mathbf{D}_{11}, \mathbf{D}_{21}, \mathbf{D}_{31}, \mathbf{D}_{32}, \mathbf{D}_{33}$

Table 7.1: Single DOF displacements and the involved constitutive components. Displacement vector are ordered as follows: $[x_i \ x_j \ x_m \ \dots \ z_m]^T$.

From this table it is clear that a single virtual test is insufficient to recover the full set of constitutive components. It may therefore be tempting to assume that the addition of test number 2 is sufficient as all constitutive components would then be involved. However, it can be shown that resulting system of linear equations contains linearly dependant equations and therefore remains an under-determined system.

7.2.1.1 Linear Dependence in Tests

The Rouché-Capelli theorem states that a system of linear equations with n variables has a unique solution if the rank of the coefficient matrix is equal to n . The rank of the matrix is equal to the number of independent equations in the system.

By constructing a matrix of the coefficients of the constitutive components we can test for linear dependence by computing the rank of the matrix. If the rank of the matrix is less than the number of variables then the system is likely under-determined. As the full expressions for these terms are far too long to be given here the example below is based on a scalene triangle.

	\mathbf{D}_{11}	\mathbf{D}_{12}	\mathbf{D}_{13}	\mathbf{D}_{22}	\mathbf{D}_{23}	\mathbf{D}_{33}
$\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1)_1$	0.20	0	0.60	0	0	0.45
$\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1)_2$	0	0.30	0.20	0	0.45	0.30
$\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1)_3$	-0.40	0	-0.70	0	0	-0.15
$\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1)_4$	0	-0.10	-0.40	0	-0.15	-0.60
$\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1)_5$	0.20	0	0.10	0	0	-0.30
$\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1)_6$	0	-0.20	0.20	0	-0.30	0.30
$\mathbf{f}_2(\mathbf{D}, \mathbf{u}_2)_1$	0	0.30	0.20	0	0.45	0.30
$\mathbf{f}_2(\mathbf{D}, \mathbf{u}_2)_2$	0	0	0	0.45	0.60	0.20
$\mathbf{f}_2(\mathbf{D}, \mathbf{u}_2)_3$	0	-0.60	-0.40	0	-0.15	-0.10
$\mathbf{f}_2(\mathbf{D}, \mathbf{u}_2)_4$	0	0	0	-0.15	-0.70	-0.40
$\mathbf{f}_2(\mathbf{D}, \mathbf{u}_2)_5$	0	0.30	0.20	0	-0.30	-0.20
$\mathbf{f}_2(\mathbf{D}, \mathbf{u}_2)_6$	0	0	0	-0.30	0.10	0.20

Table 7.2: Coefficient matrix

The table above show the values from the coefficient matrix for the first two tests in Table 7.1. Each value in the matrix is the coefficient of the constitutive component at the top of the column for the force component given at the beginning of the row. For example, if the full expression for the third component of the force vector, given the displacement vector \mathbf{u}_1 , (i.e $\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1)_3$) were given the

coefficient of $\mathbf{D}_{1,3}$ would be -0.70 .

By computing the rank of this matrix it becomes apparent that there are only 5 linearly independent equations in the system. Since there are 6 unknowns, the system is under-determined and hence has no exact solution. In order to find an exact solution additional, linearly independent, equations must be added to the system. The only way to introduce more equations in this instance is to perform an additional test – Test 3 in Table 7.1. With the addition of this test the coefficient matrix becomes 18×6 with a rank of 6. Thus there exists a solution for the system.

It may at this stage be tempting to displace additional degrees of freedom in one or more tests in order to try and reduce the total number of tests required. However, it can be shown that more complex displacement vectors do not introduce any additional independent equations. For this reason we can determine that three virtual tests are required to recover the full constitutive matrix from a 3-node triangle element.

To find the actual values of the constitutive matrix the following system must be solved:

$$\mathbf{M} \cdot \mathbf{d} = \mathcal{F} \quad (7.10)$$

where \mathbf{M} is the coefficient matrix, \mathbf{d} the vector of unknowns (i.e. $\mathbf{D}_{1,1}, \dots, \mathbf{D}_{3,3}$) and \mathcal{F} the vector of measured forces. As the system is overdetermined (i.e. contains more equations than unknowns) the coefficient matrix is non-square and hence the value of \mathbf{d} cannot be computed by $\mathbf{d} = \mathbf{M}^{-1} \cdot \mathcal{F}$. To solve this system it is possible to use the method of least squares.

7.2.1.2 Least Squares Method

For an overdetermined system of linear equations $\mathbf{Ax} = \mathbf{b}$ the least squares method can be used to find an approximate solution which satisfies:

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\| \quad (7.11)$$

That is, the set of values which minimise the error in the system $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$.

Conveniently this solution can be expressed as (Anton & Rorres [2005]):

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (7.12)$$

The matrix $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ is known as the *pseudo inverse* of A . Using this equation it is possible to find a solution for the system in Eq 7.10:

$$\mathbf{d} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathcal{F} \quad (7.13)$$

7.2.2 Worked Example

This section presents a brief worked example demonstrating how the constitutive matrix for a triangle element can be recovered using multiple virtual tests. The element shown in Figure 7.2 is used in this example. The element was given isotropic material properties with $E = 1$ and $\nu = 0.25$ – the values which are recovered.

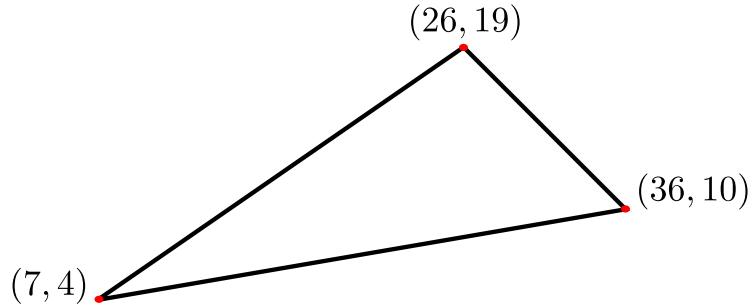


Figure 7.2: Example 3-node triangle element

7.2.2.1 Forward Problem

The forward problem involves computing the nodal forces for each test case. This is achieved by constructing the element's stiffness matrix and using Hooke's Law.

Twice the area of the triangle is computed using Eq 7.2:

$$\begin{aligned}
 2A &= \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix} \\
 &= \begin{vmatrix} 1 & 7 & 4 \\ 1 & 26 & 19 \\ 1 & 36 & 10 \end{vmatrix} \\
 A &= 160.5 \text{ units}^2
 \end{aligned} \tag{7.14}$$

The **B** matrix can be constructed using Eq 7.5 and 7.6:

$$\begin{aligned}
 \beta_i &= -9, & \beta_j &= 15, & \beta_m &= -6 \\
 \gamma_i &= -10, & \gamma_j &= -19, & \gamma_m &= 29
 \end{aligned} \tag{7.15}$$

$$\mathbf{B} = \begin{bmatrix} -0.028 & 0.0 & 0.047 & 0.0 & -0.019 & 0.0 \\ 0.0 & -0.031 & 0.0 & -0.059 & 0.0 & 0.090 \\ -0.031 & -0.028 & -0.059 & 0.047 & 0.090 & -0.019 \end{bmatrix} \quad (7.16)$$

The element's stiffness matrix, \mathbf{K} , may then be constructed by Eq 7.1:

$$\mathbf{K} = \begin{bmatrix} 0.1969 & 0.0935 & -0.1059 & -0.0224 & -0.0910 & -0.0710 \\ 0.0935 & 0.2166 & 0.0442 & 0.2316 & -0.1377 & -0.4482 \\ -0.1059 & 0.0442 & 0.5987 & -0.2959 & -0.4928 & 0.2517 \\ -0.0224 & 0.2316 & -0.2959 & 0.7400 & 0.3184 & -0.9715 \\ -0.0910 & -0.1377 & -0.4928 & 0.3184 & 0.5838 & -0.1807 \\ -0.0710 & -0.4482 & 0.2517 & -0.9715 & -0.1807 & 1.4197 \end{bmatrix} \quad (7.17)$$

For each test case the nodal forces can be computed using Hooke's Law:

$$\mathbf{f} = \mathbf{K} \cdot \mathbf{u} \quad (7.18)$$

The table below shows the nodal force vector for each of the three test cases.

Displacement Vector	Force Vector
$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0.20 & 0.09 & -0.11 & -0.02 & -0.09 & -0.07 \end{bmatrix}^T$
$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0.09 & 0.22 & 0.04 & 0.23 & -0.14 & -0.45 \end{bmatrix}^T$
$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} -0.11 & 0.04 & 0.60 & -0.30 & -0.49 & 0.25 \end{bmatrix}^T$

Table 7.3: Test case displacement vectors and corresponding nodal forces

7.2.2.2 Inverse Problem

With the nodal forces measured from the virtual tests the inverse problem is the process of recovering the element's constitutive matrix using the methods established in §7.2.1.

The element's constitutive matrix can be recovered using Eq 7.13, for this we must first compute the coefficient matrix \mathbf{M} . The process for doing so is described in §7.2.1.1. As the full expressions for the nodal forces in terms of \mathbf{D} (i.e. $\mathbf{f}_n(\mathbf{D}, \mathbf{u}_n)$) are too long to be given here the matrix \mathbf{M} is computed using the commercial CAS package Maple and given below.

$$\mathbf{M} = \begin{bmatrix} 0.1262 & 0 & 0.2804 & 0 & 0 & 0.1558 \\ 0 & 0.1402 & 0.1262 & 0 & 0.1558 & 0.1402 \\ -0.2103 & 0 & 0.0327 & 0 & 0 & 0.2959 \\ 0 & 0.2663 & -0.2103 & 0 & 0.2959 & -0.2336 \\ 0.0841 & 0 & -0.3131 & 0 & 0 & -0.4517 \\ 0 & -0.4065 & 0.0841 & 0 & -0.4517 & 0.0935 \\ 0 & 0.1402 & 0.1262 & 0 & 0.1558 & 0.1402 \\ 0 & 0 & 0 & 0.1558 & 0.2804 & 0.1262 \\ 0 & -0.2336 & -0.2103 & 0 & 0.2959 & 0.2663 \\ 0 & 0 & 0 & 0.2959 & 0.0327 & -0.2103 \\ 0 & 0.0935 & 0.0841 & 0 & -0.4517 & -0.4065 \\ 0 & 0 & 0 & -0.4517 & -0.3131 & 0.0841 \\ -0.2103 & 0 & 0.0327 & 0 & 0 & 0.2959 \\ 0 & -0.2336 & -0.2103 & 0 & 0.2959 & 0.2663 \\ 0.3505 & 0 & -0.8878 & 0 & 0 & 0.5623 \\ 0 & -0.4439 & 0.3505 & 0 & 0.5623 & -0.4439 \\ -0.1402 & 0 & 0.8551 & 0 & 0 & -0.8582 \\ 0 & 0.6776 & -0.1402 & 0 & -0.8582 & 0.1776 \end{bmatrix} \quad (7.19)$$

From this it is now possible to compute the six independent constitutive com-

ponents using Eq 7.13:

$$\begin{aligned} \mathbf{d} &= (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathcal{F} \\ &= \begin{bmatrix} 1.0667 \\ 0.2667 \\ 0 \\ 1.0667 \\ 0 \\ 0.4 \end{bmatrix} \end{aligned} \quad (7.20)$$

This vector corresponds to the following constitutive matrix:

$$\mathbf{D}_{\text{recovered}} = \begin{bmatrix} 1.0667 & 0.2667 & 0 \\ & 1.0667 & 0 \\ & Sym & 0.4 \end{bmatrix} \quad (7.21)$$

As the original material properties the element are known it is possible to verify the accuracy of the recovery by comparing the two constitutive matrices. Substituting $E = 1$ and $\nu = 0.25$ in Eq 7.4 we get:

$$\mathbf{D} = \begin{bmatrix} 1.0667 & 0.2667 & 0 \\ & 1.0667 & 0 \\ & Sym & 0.4 \end{bmatrix} \quad (7.22)$$

It is clear that $\mathbf{D}_{\text{recovered}} = \mathbf{D}$ and hence that the constitutive matrix has been accurately recovered by way of virtual testing.

7.2.3 Multi-Scale Triangle Elements

Section 7.2.2 demonstrated how the constitutive matrix of a 3-node triangle element can be recovered by way of virtual testing. In each of these virtual tests a single node is displaced along one axis. The resulting forces are then equated to the nodal forces in terms of the effective constitutive matrix in order to find the matrix values. As has been noted, this in itself has little direct practical use since the constitutive matrix to be recovered must first be known.

For practical applications we aim to recover an effective constitutive matrix from a sub-volume bounded by a triangle (now referred to as the *macro element*). This macro element will contain smaller triangles used to discretise the sub-volume, which we shall refer to as *micro elements*. To compute an effective constitutive matrix the basic principle presented in the previous section shall be used, with a number of changes to accommodate the micro elements.

In the previous scheme the virtual tests could be described using a simple displacement vector as they were performed on a single element. However, in this case the virtual tests are to be performed on the *micro mesh*. That is, the mesh of smaller triangles shown in Figure 7.3. To achieve this we constrain the displacement of the so-called ‘external micro nodes’ to the surface of the macro element. This constraint is used so that the area discretised by the micro elements is limited to behave as if it were the macro element. To perform the virtual tests we must first compute the displacement of each of the external micro nodes.

Given a displacement vector $\mathbf{u} = \{u_i, v_i, u_j, v_j, u_m, v_m\}^T$ for the macro element it is possible to compute the displacement of any point inside the element by using

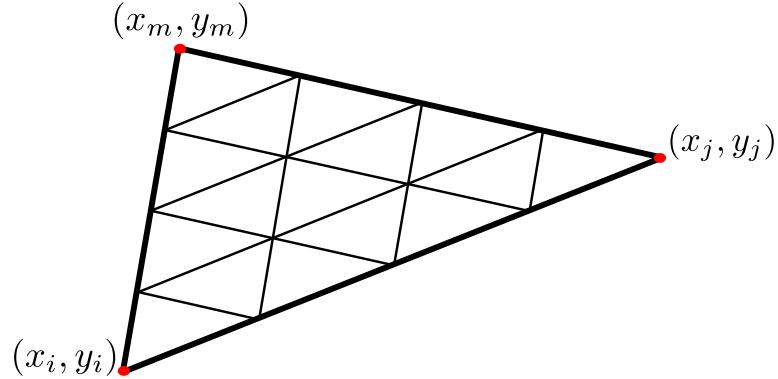


Figure 7.3: Macro triangle (bold) discretised using micro triangles

the element's shape functions¹. For the 3-node triangle:

$$\alpha_i = x_j y_m - y_j x_m, \quad \alpha_j = y_i x_m - x_i y_m, \quad \alpha_m = x_i y_j - y_i x_j \quad (7.23)$$

$$\begin{aligned} N_i &= \frac{1}{2A}(\alpha_i + \beta_i x + \gamma_i y) \\ N_j &= \frac{1}{2A}(\alpha_j + \beta_j x + \gamma_j y) \\ N_m &= \frac{1}{2A}(\alpha_m + \beta_m x + \gamma_m y) \end{aligned} \quad (7.24)$$

Using these expressions the displacement of the point can be described:

$$\begin{aligned} \Delta x &= N_i u_i + N_j u_j + N_m u_m \\ \Delta y &= N_i v_i + N_j v_j + N_m v_m \end{aligned} \quad (7.25)$$

where (x, y) is the point's original location and $(\Delta x, \Delta y)$ the displacement vector.

¹Not to be confused with the displacement of *micro nodes*, which for the large part remain unknown.

β_i , β_j , β_m , γ_i , γ_j and γ_m are given in Eq 7.6.

Previously three displacement vectors were used to recover the material properties of a homogeneous triangle element. By using the displacement functions in Eq 7.25 it is possible to map each of these vectors to a set of displacement vectors for each external micro node. Thus, the virtual tests to be performed on the micro elements are defined. The tests are performed by displacing the external micro nodes and computing the resulting forces. The internal micro nodes are unconstrained and therefore have an unknown displacement. In order to solve this kind of problem a finite element package such as the commercial software Abaqus or open-source CalculiX ([Cal, 2010, CalculiX]) is used.

To follow the same methodology as with the single homogeneous triangle element we must calculate the forces on the macro triangle, for each test. This highlights a potential difficulty. In the previous situation each virtual test was performed directly on the macro triangle and the reaction forces trivially computed. For the case where the macro triangle is discretised reaction forces are computed for each external micro node. Thus, the effective forces on the macro nodes must be calculated. The effective forces represent the forces required to displace the macro nodes of an identical macro element with homogeneous material properties representing the behaviour of the micro mesh. These forces are computed by taking a weighted sum of the micro forces using the element's shape functions.

For a set of micro forces $\mathbf{f}_1^X \dots \mathbf{f}_n^X$ and $\mathbf{f}_1^Y \dots \mathbf{f}_n^Y$ the effective macro forces are

computed as follows:

$$\mathcal{F}_t = \begin{bmatrix} \sum_{k=1}^n N_i \cdot {}^t \mathbf{f}_k^X \\ \sum_{k=1}^n N_i \cdot {}^t \mathbf{f}_k^Y \\ \sum_{k=1}^n N_j \cdot {}^t \mathbf{f}_k^X \\ \sum_{k=1}^n N_j \cdot {}^t \mathbf{f}_k^Y \\ \sum_{k=1}^n N_m \cdot {}^t \mathbf{f}_k^X \\ \sum_{k=1}^n N_m \cdot {}^t \mathbf{f}_k^Y \end{bmatrix} \quad (7.26)$$

where n is the number of external micro nodes and t is the test number. With these three vectors the full effective force vector can be compiled:

$$\mathcal{F}_{\text{effective}} = [\mathcal{F}_1 \ \mathcal{F}_2 \ \mathcal{F}_3]^T \quad (7.27)$$

Substituting $\mathcal{F} = \mathcal{F}_{\text{effective}}$ in Eq 7.13 it is possible to find the effective constitutive matrix:

$$\mathbf{d}_{\text{effective}} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathcal{F}_{\text{effective}} \quad (7.28)$$

7.2.4 Multi-Scale Example

Using the methods developed in the previous section this example shows how the effective properties of a discretised macro triangle can be recovered. The macro triangle shown in Figure 7.4 and its 16 micro triangles are used for this example.

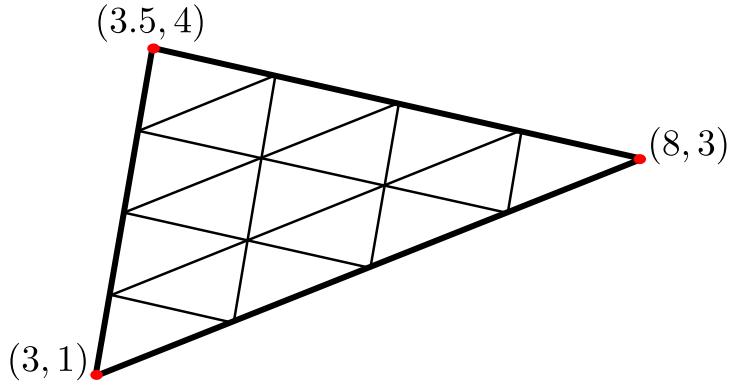


Figure 7.4: Example discretised macro element

7.2.4.1 Homogeneous Case

This is the case where all micro elements have the same values $E = 1$ and $\nu = 0.25$.

By using the three displacement vectors given in Table 7.1 the displacement vectors for the micro nodes are computed using the macro element's shape functions. Figure 7.5 shows the displacement of the external micro nodes for each of the three test cases. The internal micro nodes are given an approximate position for the purpose of the figure.

Micro Element Type There exists a selection of different 3 node elements which could be used for the finite element simulations on the micro meshes. In the commercial package Abaqus these elements are either linear strain (i.e. element type CPE3) or linear stress (i.e. CPS3). There also exists a constant strain triangle (CST), on which the macroelements in this section are based, which is not available in Abaqus. The constant strain element is commonly avoided due to its poor approximation of the strain field. However, it has been observed that in this application the CST element must be used in order to accurately recover the effective properties of the micro mesh. For the case where the micro mesh is

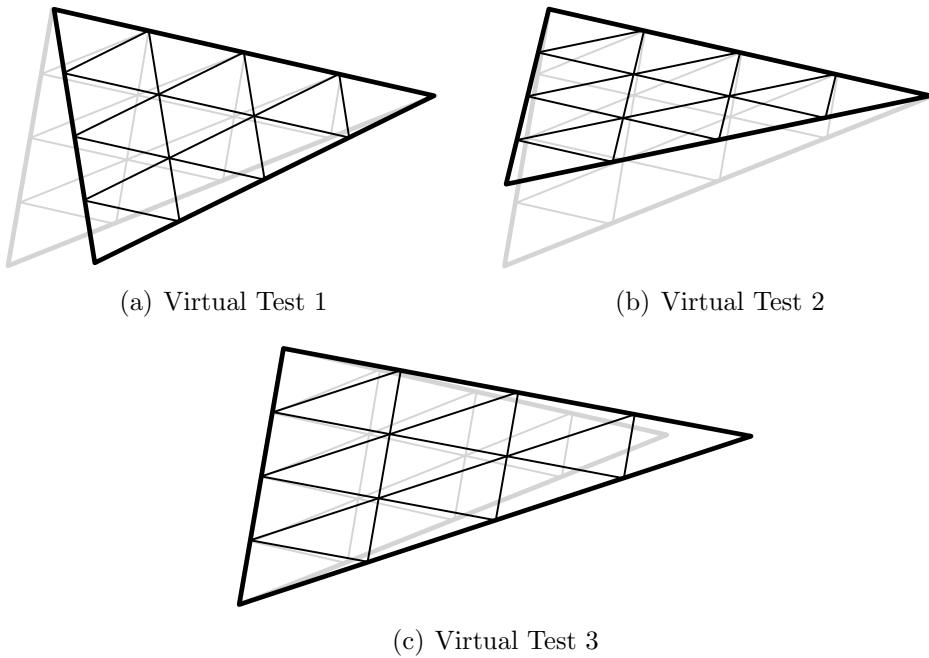


Figure 7.5: Comparison of original (grey) mesh and deformed mesh (black) for each test case

homogeneous the use of Abaqus and the linear strain elements results in effective properties not equal to the micro element properties. It is suspected that the macro and micro elements should be of the same type to ensure accurate property recovery. Later examples will demonstrate the accuracy of this method using CST elements therefore eliminating the need to investigate the more complex element types.

The finite element simulations, as described in Figure 7.5, are performed using constant strain triangle elements. As this element is not available in Abaqus, the simulations are performed using a Maple worksheet created by Artur Portela ([Map, 2010, Finite Elements Package]). This worksheet is suitable for the small problems in this section, providing the displacement of internal nodes and reaction forces for the external nodes.

Rather than giving the force for each micro node, the results below show the effective macro forces calculated using Eq 7.26.

Test Number	Effective Macro Force Vector
1	$[0.327 \ 0.107 \ -0.082 \ -0.188 \ -0.245 \ 0.081]^T$
2	$[0.107 \ 0.786 \ -0.121 \ -0.043 \ 0.014 \ -0.829]^T$
3	$[-0.829 \ -0.121 \ 0.346 \ -0.036 \ -0.264 \ 0.157]^T$

Table 7.4: Effective macro forces computed from micro node forces

After constructing the vector $\mathcal{F}_{\text{effective}}$ the effective constitutive matrix can be computed using Eq 7.28. The results are given below in matrix form.

$$\mathbf{D}_{\text{effective}} = \begin{bmatrix} 1.0667 & 0.2667 & 9.4353 \times 10^{-11} \\ & 1.0667 & -9.0836 \times 10^{-11} \\ Sym & & 0.4 \end{bmatrix} \quad (7.29)$$

This constitutive matrix corresponds to a homogeneous material with $E = 1$ and $\nu = 0.25$ showing the method has accurately recovered the initial material properties.

7.2.4.2 Inhomogeneous Cases

It has been clearly demonstrated that the homogenisation method can accurately recover the constitutive matrix of a homogeneous material. However, this has little practical uses beyond validating the technique used as the homogeneous material properties must be known from the start. Clearly the elements in the micro mesh need not all have the same properties nor completely discretise the macro element (i.e. there may be regions of void). In this case, where the sub-volume is inhomogeneous, the same method can be applied to recover the

effective constitutive matrix. However, as there are no trivial cases in 2D with known effective properties this will be left to be demonstrated later in 3D where more complex structures with known properties can be homogenised.

7.3 3D

The previous sections presented the principle of the homogenisation technique and demonstrated the accuracy of the method at recovering homogeneous material properties in 2D. This section extends the method to 3D and the recovery of effective material properties of inhomogeneous sub-volumes. In the previous section the multi-scale example given had been manually constructed for a fictitious domain conforming exactly to the macro element. However, in real-world applications where the geometry to be homogenised has been acquired (or generated) using an imaging technique it is unlikely to conform exactly to the chosen macro element. To address this a method for ‘cutting’ image volumes to accurately fit macroelements is developed in this section.

As with the 2D case a simple element was chosen to use as the macro element for the homogenisation, the 4 node tetrahedron.

7.3.1 Constitutive Matrix Recovery from a 4 Noded Tetrahedral Element

Following the same format as in §7.2.1 we first aim to show the constitutive matrix used by the 4 noded linear tetrahedron (as shown in Figure 7.6) can be recovered by way of virtual testing.

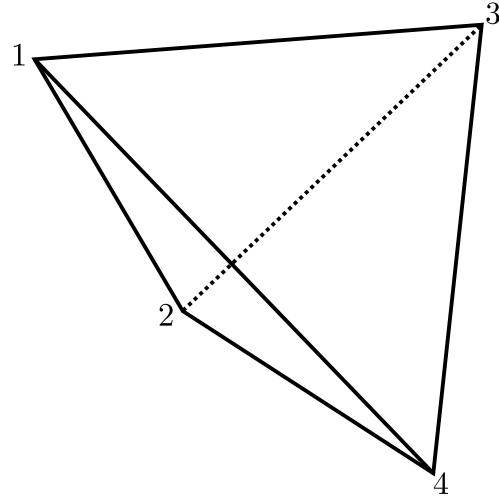


Figure 7.6: 4-node tetrahedral element. Nodes are labelled anti-clockwise when viewed from the last node.

The stiffness matrix of the tetrahedral element is computed as follows:

$$\mathbf{K} = \mathbf{B}^T \mathbf{D} \mathbf{B} V \quad (7.30)$$

Volume of a tetrahedron Six times the volume of a tetrahedron can be computed using the following determinant:

$$6V = \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} \quad (7.31)$$

Constitutive matrix The constitutive matrix for an isotropic material is defined as:

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (7.32)$$

where E is the Young's modulus and ν the Poisson's ratio. Notably this matrix is of the order 6×6 , compared to the 3×3 matrix in 2D. Thus, for the case of general anisotropy, there are 21 unknown values to recover.

B Matrix The matrix \mathbf{B} is constructed as follows:

$$\mathbf{B} = [\mathbf{B}_1 \ \mathbf{B}_2 \ \mathbf{B}_3 \ \mathbf{B}_4] \quad (7.33)$$

where

$$\mathbf{B}_1 = \frac{1}{6V} \begin{bmatrix} \beta_1 & 0 & 0 \\ 0 & \gamma_1 & 0 \\ 0 & 0 & \delta_1 \\ \gamma_1 & \beta_1 & 0 \\ 0 & \delta_1 & \gamma_1 \\ \delta_1 & 0 & \beta_1 \end{bmatrix} \quad (7.34)$$

with similar expressions for \mathbf{B}_2 , \mathbf{B}_3 and \mathbf{B}_4 , where:

$$\beta_1 = - \begin{vmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix}, \quad \gamma_1 = \begin{vmatrix} 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{vmatrix}, \quad \delta_1 = - \begin{vmatrix} 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{vmatrix} \quad (7.35)$$

and

$$\beta_2 = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix}, \quad \gamma_2 = - \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{vmatrix}, \quad \delta_2 = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{vmatrix} \quad (7.36)$$

and

$$\beta_3 = - \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_4 & z_4 \end{vmatrix}, \quad \gamma_3 = \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_4 & z_4 \end{vmatrix}, \quad \delta_3 = - \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_4 & y_4 \end{vmatrix} \quad (7.37)$$

and

$$\beta_4 = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix}, \quad \gamma_4 = - \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \end{vmatrix}, \quad \delta_4 = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \quad (7.38)$$

Following the same process established in §7.2.1 we displace nodes on the macro element in order to measure a force which can then be equated to the force in terms of the constitutive matrix. This is expressed by the following

equality:

$$\mathbf{f}_n(\mathbf{D}, \mathbf{u}_n) = \mathcal{F}_n(\mathbf{u}_n) \quad (7.39)$$

The following displacement vectors are used for performing the virtual tests. An explanation of the choice of these vectors is given later in §7.3.6.

$$\begin{aligned} \mathbf{u}_1 &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_2 &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_3 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_4 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T \\ \mathbf{u}_5 &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_6 &= [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \end{aligned} \quad (7.40)$$

For each virtual test on the macro element the resulting forces are calculated using Hooke's law and the force vector \mathcal{F} constructed. The matrix \mathbf{M} is then calculated and the constitutive matrix recovered using Eq 7.13.

7.3.2 Worked Example

Following the format presented in the 2D section a worked example is given demonstrating the process of recovering the constitutive matrix from a tetrahedral element. The element used in this section is shown in Figure 7.7.

In 2D the macro element was given simple isotropic material properties. For this example a fictitious anisotropic constitutive matrix, given in Eq 7.41, is used

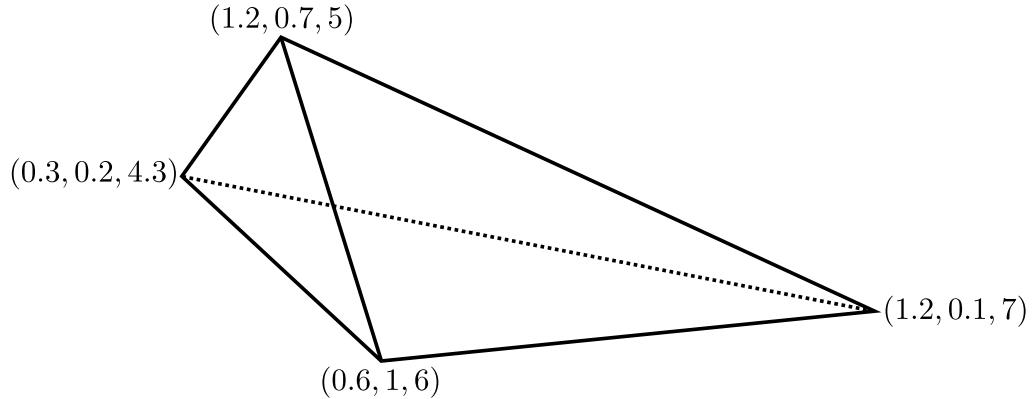


Figure 7.7: Example tetrahedral element

to prove that the matrix can be recovered in the most general case.

$$\mathbf{D} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & \\ 12 & 13 & 14 & 15 & & \\ 16 & 17 & 18 & & & \\ 19 & 20 & & & & \\ Sym & & & & & 21 \end{bmatrix} \quad (7.41)$$

The displacement vectors in Eq 7.40 are used for the six virtual tests performed on the element. The resulting forces are given in Table 7.5.

From these values the coefficient matrix \mathbf{M} (of the order 72×12) can be constructed and the vector \mathbf{d} can be recovered using Eq 7.13. This corresponds to

Component	f₁	f₂	f₃	f₄	f₅	f₆
1	-0.12	-1.18	0.27	-0.04	1.08	2.47
2	1.08	-5.02	-1.64	0.59	5.28	7.38
3	2.47	-7.80	-2.11	1.04	7.38	11.23
4	-1.18	5.13	1.48	-0.5	-5.02	-7.8
5	-1.33	5.97	1.82	-0.94	-6.14	-8.92
6	-2.24	8.01	2.27	-1.06	-7.77	-11.66
7	-0.27	1.48	0.57	-0.04	-1.64	-2.11
8	0.29	-0.46	-0.14	-0.32	0.27	0.5
9	-0.48	1.71	0.46	-0.17	-1.58	-2.59
10	1.57	-5.43	-1.78	0.57	5.58	7.45
11	-0.04	-0.5	-0.04	0.68	0.59	1.04
12	0.25	-1.91	-0.62	0.18	1.98	3.03

Table 7.5: Forces resulting from virtual tests

the following matrix:

$$\mathbf{D} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ & 7 & 8 & 9 & 10 & 11 \\ & & 12 & 13 & 14 & 15 \\ & & & 16 & 17 & 18 \\ & & & & 19 & 20 \\ & & & & & 21 \\ Sym & & & & & \end{bmatrix} \quad (7.42)$$

The actual values recovered are only ‘accurate’ when given to six decimal places, or fewer. This is likely due to floating point precision. Despite this we can still consider this an accurate recovery of the constitutive matrix.

7.3.3 Multi-Scale Tetrahedral Elements

Given a macro displacement vector $\{u_1, v_1, w_1, \dots, u_4, v_4, w_4\}^T$ the displacement of a point inside the tetrahedron is determined by the element's shape functions:

$$\begin{Bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \\ u_4 \\ v_4 \\ w_4 \end{Bmatrix} \quad (7.43)$$

where

$$N_1 = \frac{(\alpha_1 + \beta_1 x + \gamma_1 y + \delta_1 z)}{6V}, \quad N_2 = \frac{(\alpha_2 + \beta_2 x + \gamma_2 y + \delta_2 z)}{6V} \quad (7.44)$$

$$N_3 = \frac{(\alpha_3 + \beta_3 x + \gamma_3 y + \delta_3 z)}{6V}, \quad N_4 = \frac{(\alpha_4 + \beta_4 x + \gamma_4 y + \delta_4 z)}{6V}$$

Using these expressions the boundary conditions for the micro mesh can be established and the resulting micro forces solved for using finite element analysis. Once the micro forces have been computed the effective macro forces can

calculated, again weighted by the element's shape functions.

$$\mathcal{F}_t = [\mathcal{F}_t^A \quad \mathcal{F}_t^B \quad \mathcal{F}_t^C \quad \mathcal{F}_t^D]^T \quad (7.45)$$

where

$$\begin{aligned} \mathcal{F}_t^A &= \left[\begin{array}{c} \sum_{k=1}^n N_1 \cdot {}^t \mathbf{f}_k^X \\ \sum_{k=1}^n N_1 \cdot {}^t \mathbf{f}_k^Y \\ \sum_{k=1}^n N_1 \cdot {}^t \mathbf{f}_k^Z \end{array} \right], \quad \mathcal{F}_t^B = \left[\begin{array}{c} \sum_{k=1}^n N_2 \cdot {}^t \mathbf{f}_k^X \\ \sum_{k=1}^n N_2 \cdot {}^t \mathbf{f}_k^Y \\ \sum_{k=1}^n N_2 \cdot {}^t \mathbf{f}_k^Z \end{array} \right] \\ \mathcal{F}_t^C &= \left[\begin{array}{c} \sum_{k=1}^n N_3 \cdot {}^t \mathbf{f}_k^X \\ \sum_{k=1}^n N_3 \cdot {}^t \mathbf{f}_k^Y \\ \sum_{k=1}^n N_3 \cdot {}^t \mathbf{f}_k^Z \end{array} \right], \quad \mathcal{F}_t^D = \left[\begin{array}{c} \sum_{k=1}^n N_4 \cdot {}^t \mathbf{f}_k^X \\ \sum_{k=1}^n N_4 \cdot {}^t \mathbf{f}_k^Y \\ \sum_{k=1}^n N_4 \cdot {}^t \mathbf{f}_k^Z \end{array} \right] \end{aligned} \quad (7.46)$$

With these force vectors the full effective force vector can be constructed:

$$\mathcal{F}_{\text{effective}} = [\mathcal{F}_1 \quad \mathcal{F}_2 \quad \mathcal{F}_3 \quad \mathcal{F}_4 \quad \mathcal{F}_5 \quad \mathcal{F}_6]^T \quad (7.47)$$

Substituting $\mathcal{F} = \mathcal{F}_{\text{effective}}$ in Eq 7.13 it is possible to find the effective constitutive matrix:

$$\mathbf{d}_{\text{effective}} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathcal{F}_{\text{effective}} \quad (7.48)$$

7.3.4 Multi-Scale Example

Given the similarities between the constant strain triangle and linear tetrahedron it is expected that a macro tetrahedron discretised exactly with micro elements

would be able to accurately recover a constitutive matrix. Also, for practical 3D problems where the structure to homogenise is defined using image data the sub-volume is unlikely to exactly conform to the macro element (see the following section). For these reasons an ‘ideal’ multi-scale example is not given here. Instead it is left for later in this work where a more realistic example can be given.

7.3.5 Sub-Volume From Image Data

For practical applications of the homogenisation the sub-volume within the macro element will likely be a multi-phase structure, such as a solid/void micro-architecture. The geometry of which can be specified using image data, either generated, such as the micro-architectures in Chapter 3, or acquired using one of the various imaging modalities (e.g. MRI, or CT). Traditionally, for any characterisation or homogenisation, the sub-volume of interest is cropped to fit either a cubic or cuboidal domain. For image data, where the domain is aligned with the primary axes, the cropping process is trivial - voxels inside the domain are simply extracted. However, when the domain is not aligned with the image data, or as in this case is non-cuboidal, simply extracting the voxels considered inside leads to a surface which conforms very poorly to the macro element. Figure 7.8 shows the result of using this method to extract a tetrahedral sub-volume from a ‘solid’ (i.e. all voxel values are greater than the iso-surface value) image volume.

In this instance many of the surface nodes on the mesh shown in Figure 7.8(b) do no lie on the surface of the macro element – a requirement for the homogenisation. To ensure the nodes lie exactly on the surface of the macro

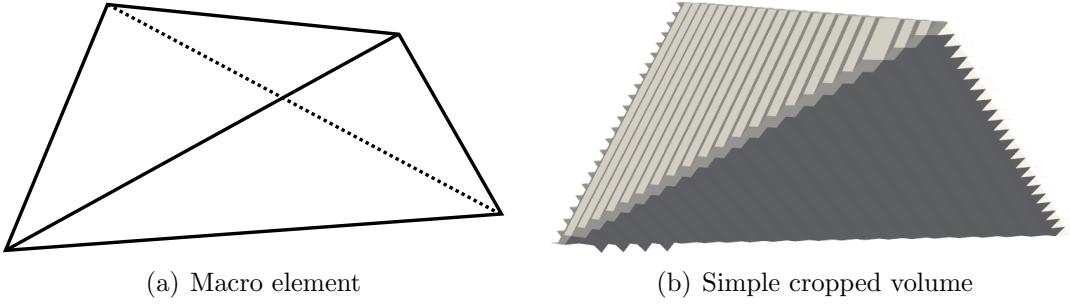


Figure 7.8: Original macro element and volume cropped from a $25 \times 50 \times 25$ px volume

element we can modify the values of voxels near the surface in order to create a flat iso-surface. These voxel values should reflect their distance from the closest face of the macro element in the same way that a CAD model is voxelised.

Using the following rules a volume can be better ‘cropped’ to fit a given macro element:

$$V_{\text{new}} = \begin{cases} V_{\text{old}} & \text{Inside, } d > d_{\max}, \\ 0 & \text{Outside, } d > d_{\max}, \\ \left(\frac{d}{d_{\max}} \times i\right) + i & \text{Inside, } d \leq d_{\max}, \\ \left(1 - \frac{d}{d_{\max}}\right) \times i & \text{Outside, } d \leq d_{\max}, V_{\text{old}} \geq i \end{cases} \quad (7.49)$$

where V_{new} is the new voxel value, V_{old} the current voxel value, d the distance from the current voxel to the nearest face of the macro element, d_{\max} the maximum distance from the macro element to adjust voxel values and i the iso-surface value. Figure 7.9 shows a tetrahedral sub-volume extracted from an image volume using these rules.

As sharp edges cannot be represented in image data the resulting sub-volume cannot exactly conform to the macro element, as highlighted by the jagged edges in the sub-volume. Figure 7.10 shows an example micro-architecture cropped to

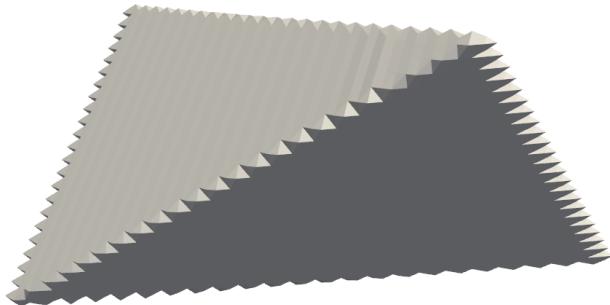


Figure 7.9: Cropped volume using distance functions

fit a tetrahedral macro element.

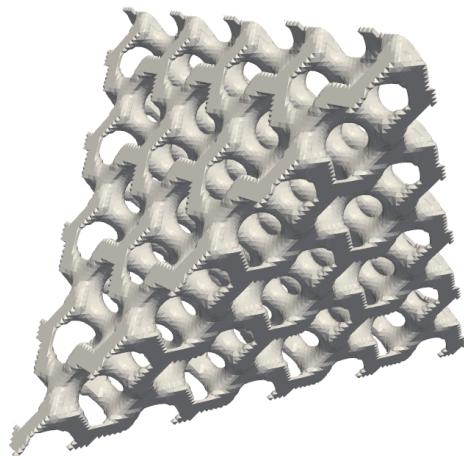


Figure 7.10: Example micro-architecture conforming to a tetrahedral macro element

After the sub-volume has been extracted a volume mesh must be generated to act as the micro mesh in the homogenisation. The sub-volume is meshed using ⁺ScanFE. To ensure the mesh consists of good quality elements the “off surface” option is used which allows nodes to deviate from the iso-surface for the purpose of improving element quality. Consequently, when locating nodes which lie on the macro element’s surface a small tolerance must be used. A tolerance of appropriately 1 unit spacing appears suitable.

7.3.6 Choice of Virtual Tests

Up to this point the virtual tests performed have been arbitrarily chosen as the sequential displacement of single degrees of freedom. Using the Maple worksheets developed for this chapter it is possible to easily examine the effects of various boundary conditions.

7.3.6.1 Criteria For Tests

Some observations can be made about possible criteria for the virtual tests used in this homogenisation method. By taking advantage of the symbolic computing functions in Maple we can examine the effect of various displacement vectors in the most general case, where the macro element's geometry is defined symbolically.

Up to this point the virtual tests performed have been arbitrarily chosen as the sequential displacement of single degrees of freedom. In 2D this has proven to be sufficient in recovering the macro element's constitutive matrix. However, as these were arbitrarily chosen it is worth investigating other displacement vectors in order to try and establish a set of rules for single DOF displacement tests.

Previously the following displacement vectors had been used:

$$\begin{aligned}\mathbf{u}_1 &= [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_2 &= [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_3 &= [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T\end{aligned}\tag{7.50}$$

With these displacement vectors the macro force vectors (i.e. $\mathbf{f}(\mathbf{D}, \mathbf{u})$) take the

form:

$$\begin{aligned}\mathbf{f}_1(\mathbf{D}, \mathbf{u}_1) &= [A \ B \ A \ B \ A \ B]^T \\ \mathbf{f}_2(\mathbf{D}, \mathbf{u}_2) &= [B \ C \ B \ C \ B \ C]^T \\ \mathbf{f}_3(\mathbf{D}, \mathbf{u}_3) &= [A \ B \ A \ B \ A \ B]^T\end{aligned}\tag{7.51}$$

where

$$\begin{aligned}A &= \xi(\mathbf{D}_{1,1}, \mathbf{D}_{3,1}, \mathbf{D}_{3,3}) \\ B &= \xi(\mathbf{D}_{2,1}, \mathbf{D}_{3,2}, \mathbf{D}_{3,1}, \mathbf{D}_{3,3}) \\ C &= \xi(\mathbf{D}_{2,2}, \mathbf{D}_{3,2}, \mathbf{D}_{3,3})\end{aligned}\tag{7.52}$$

Here $\xi(\dots)$ is used to indicate that the force component is a function of the given constitutive components. For these displacement vectors each value in the constitutive matrix is involved in the macro forces. However, if we change the second test so that $\mathbf{u}_2 = [0 \ 0 \ 0 \ 0 \ 1 \ 0]^T$ we then get the following:

$$\mathbf{f}_2(\mathbf{D}, \mathbf{u}_2) = [A \ B \ A \ B \ A \ B]^T\tag{7.53}$$

By changing the displacement vector such that all displacements are along the x -axis we now lose dependence on the term $\mathbf{D}_{2,2}$ and therefore have an underdetermined system. This is to be expected as the term $\mathbf{D}_{2,2}$ is heavily dependant on the value of E_y for an orthotropic material. Conversely, if we consider the case where there are only displacements along the y -axis then we lose the term $\mathbf{D}_{1,1}$ which is heavily dependant on E_x . Based on these observations we may conclude that the tests must contain a displacement along each axis. However, with the 3 node triangle in 2D there are a limited number of unique displacement vectors on which to base this conclusion. Using the same methods described it is possible to explore the effects of different displacement vectors when using the 4 node

tetrahedral element in 3D.

To test the theory that the tests must contain at-least one displacement along each axis we choose the six tests for the tetrahedron such that there are four displacements along the x -axis and one along each of the y and z axes.

$$\begin{aligned}\mathbf{u}_1 &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_2 &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_3 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_4 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]^T \\ \mathbf{u}_5 &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{u}_6 &= [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T\end{aligned}\tag{7.54}$$

However, this results in an underdetermined system due to linear dependence rather than missing terms. Changing one of the displacement vectors \mathbf{u}_1 , \mathbf{u}_2 , \mathbf{u}_3 or \mathbf{u}_4 such that the displacement is along either the y or z axis results in an overdetermined system which may then be solved using the least squares method. In this case there are a different number of displacements along each axis. By trying different combinations of displacement vectors matching this criteria we can come the conclusion that for single DOF displacements there must be a different number of displacements along each axis. Based on these observations we now have two rules for choosing displacement vectors:

1. There must not be duplicated displacement vectors
2. There must be a different number of displacements along each axis

7.3.6.2 Equivalence of Tests

It can be shown that displacing single degrees of freedom in the virtual tests can result in ‘duplicate tests’ depending on the geometry of the macro element. To demonstrate this we consider the macroelements in Figure 7.11.

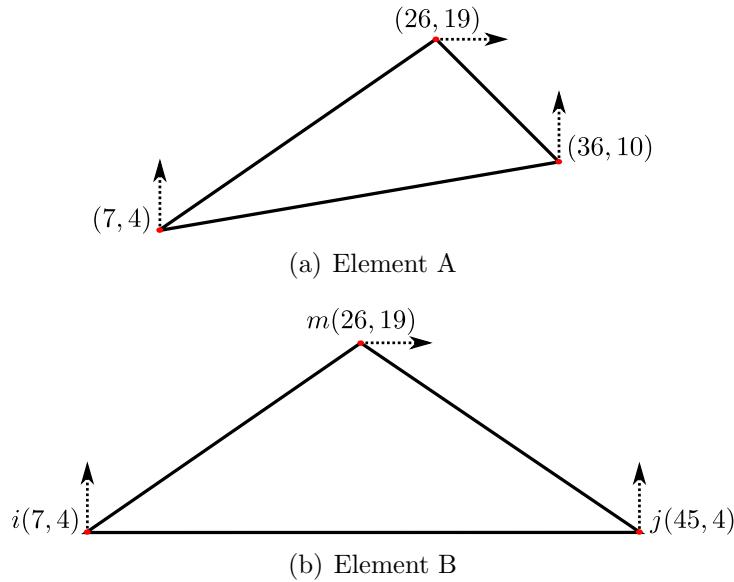


Figure 7.11: Two macroelements of differing geometry subjected to identical virtual tests

In each of the six tests described the magnitude of the displacement is equal to one unit.

For the first case the constitutive matrix of Element A can be accurately recovered using the methods described in this chapter. The set of tests chosen differ from those previously used for the 2D elements, but still meet the requirements described in §7.3.6.1. However, when the same set of tests are applied to Element B the resulting system of linear equations becomes underdetermined and hence the constitutive matrix cannot be recovered. By examining the tests in Figure 7.11(b) it is clear that the two tests displacing the i and j nodes (i.e. the two

displacements along the y -axis) result in the same deformed element.

One possible solution to this problem is to choose the displacement vectors specifically for each different geometry such that they do not result in duplicate tests. In this instance the displacement vectors given in Table 7.3 would be suitable for Element B, however they would likely result in the same issues for other symmetrical macroelements. Alternatively, more complex displacement vectors can be used to avoid issues with symmetry. By using a set of randomly orientated displacement vectors we can ensure that macroelements generated by automated meshing processes can handled without the need to choose custom tests. While, in theory, it is possible that a set of randomly chosen vectors may still result in duplicated tests, in practise this appears highly unlikely and so is an acceptable solution.

7.4 Validation

It has already been shown that a single tetrahedron, exactly equal to the macro element, is sufficient to recover a given constitutive matrix. It has also been shown that a triangular macro element discretised exactly using triangle elements can also accurately recover a constitutive matrix. Given the similarities between the two types of elements used (linear triangle and tetrahedron) it is expected that a tetrahedral macro element discretised by tetrahedrons would accurately recover a given constitutive matrix. However, we are interested in the case where the volume of interest is defined using image data. In this case the sub-volume may not exactly conform to the macro element, as shown in §7.3.5.

Before attempting to compute the effective properties of a unknown struc-

ture, we must first test the homogenisation method in cases where we know or can compute the correct constitutive matrix. Thus, a number of tests will be performed in this section for the purpose of attempting to validate the developed homogenisation method. In each of these tests the domain and macro element shown in Figure 7.12 are used. As the macro element must be a tetrahedron the chosen element is the largest tetrahedron which will fit inside a cube, occupying one third of the cube's volume. Thus only one third of the original volume will be used in the virtual tests. Due to this fact it is expected that the size of the original volume will have to be larger than would normally be required to account for the ‘lost’ two thirds of the cube’s volume.

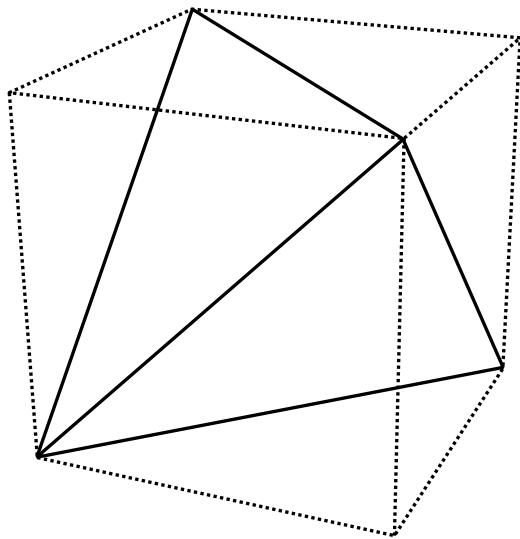


Figure 7.12: Macro element (solid) within the bounding volume (dashed)

7.4.1 Homogeneous Sub-Volume

The first, and perhaps most obvious, validation test to perform is the case where the sub-volume of interest is a homogeneous material with a known value. This

test will differ from that performed in §7.2.4 as the micro mesh will be generated from image data.

As with any finite element simulations a convergence study is performed to ensure mesh independence is achieved. For this study the volume bounding the macro element is increased in resolution from $10 \times 10 \times 10$ px to $160 \times 160 \times 160$ px. So that a single value may be plotted we use $E_{xx} = \frac{1}{S_{11}}$. The results are given in Figure 7.13.

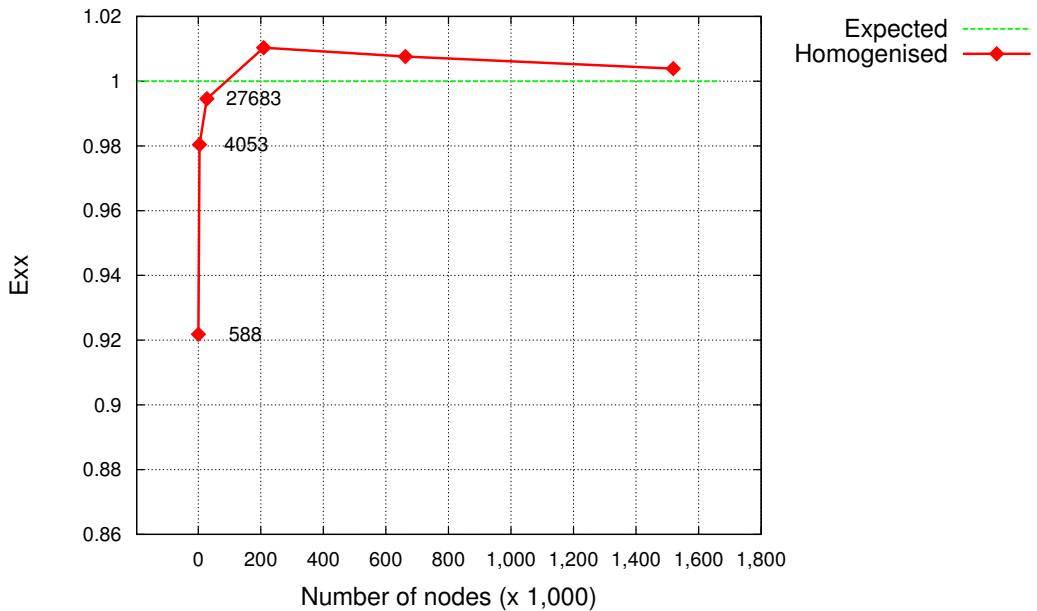


Figure 7.13: Convergence study for a homogeneous sub-volume

In these results we see unusual convergence from below followed by the expected convergence from above. This is likely due to two competing effects; at the lower resolutions the tetrahedron will be poorly represented causing lower than expected values for E_{xx} . As the tetrahedron is discretised with greater accuracy this effect reduces as convergence to the geometry is effectively achieved. Beyond this point the lack of degrees of freedom becomes the influencing effect causing

an over-estimated value for E_{xx} and the more usual convergence from above. In this range the value of E_{xx} is recovered with at most a 1% error.

7.4.2 Comparison to KUBC Results

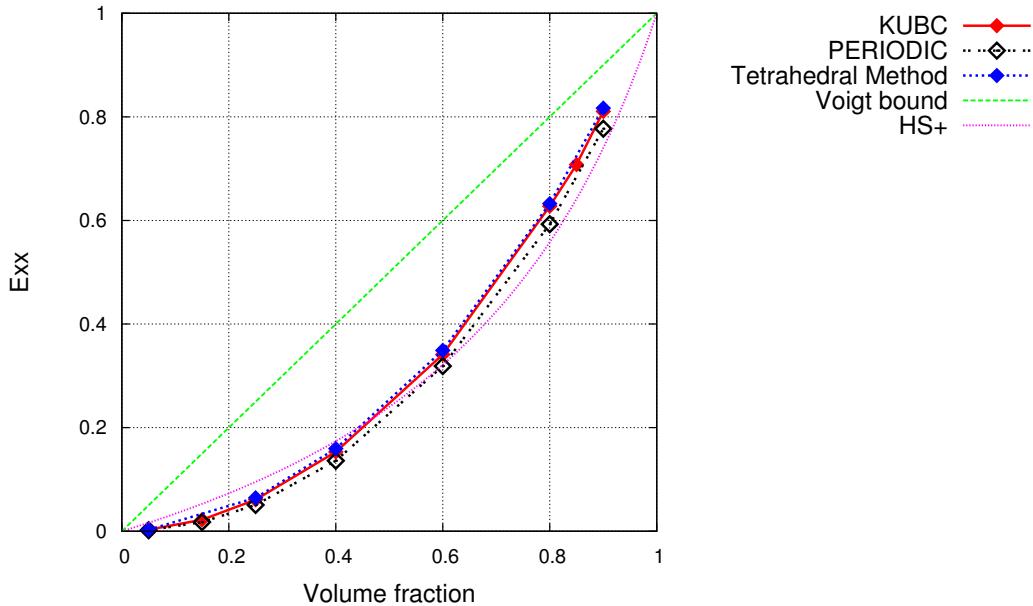
To validate the developed method with more complex structures we compare the results obtained in §6.1.2 to those obtained using the tetrahedral approach. As this method requires a tetrahedral domain, compared to the cuboidal domain used in the KUBC homogenisation, we must ensure the comparisons performed are fair. To do so we use the same number of unit cells within both domains. The results presented in Figure 7.14 show excellent agreement between the two homogenisation methods.

7.5 Approximate Models

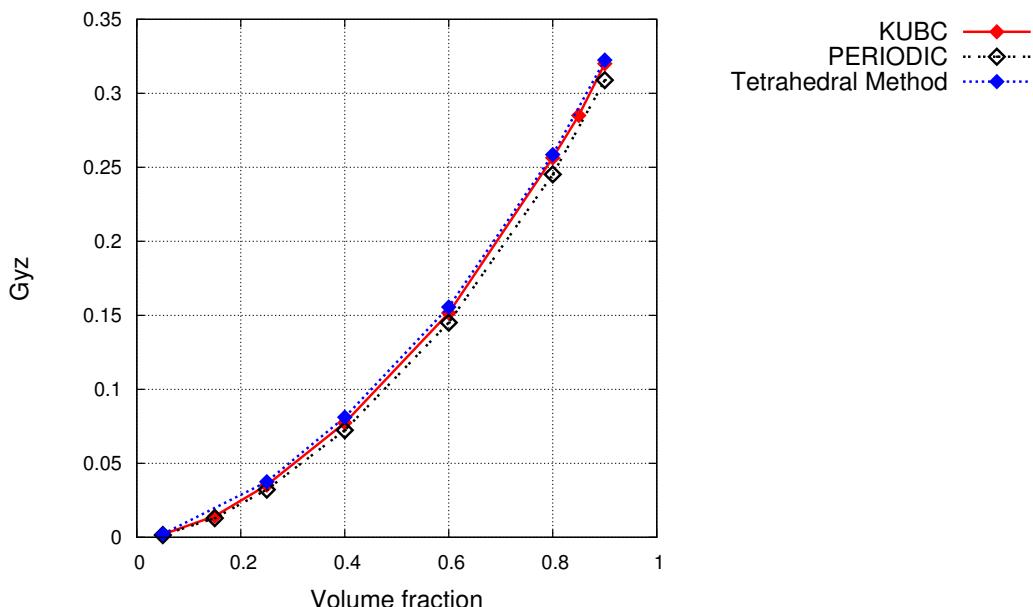
With the ability to determine an effective constitutive matrix for an arbitrary tetrahedral sub-volume we can now address the issue of multi-scale problems. Of particular interest are the set of problems having an irregular (i.e. non-cuboidal) domain. While problems of a more regular nature may be addressed with more conventional methods of determining effective constitutive matrices, they are never the less addressable using the methods developed in this chapter.

Multi-scale problems featuring two or more distinguishable length-scales are particularly computationally challenging to model exactly¹. If we consider an arbitrary irregular domain Ω then it is possible to identify a number of length-scale or feature-lengths, as shown in Figure 7.15.

¹Of course, even a ‘full’ finite element simulation is an approximation



(a) Apparent Young's Modulus (Exx) for $8 \times 8 \times 8$ unit cells of the Schoen Gyroid



(b) Apparent Shear Modulus (Gyz) for $8 \times 8 \times 8$ unit cells of the Schoen Gyroid

Figure 7.14: Comparison of KUBC results to the tetrahedral method for the Schoen Gyroid

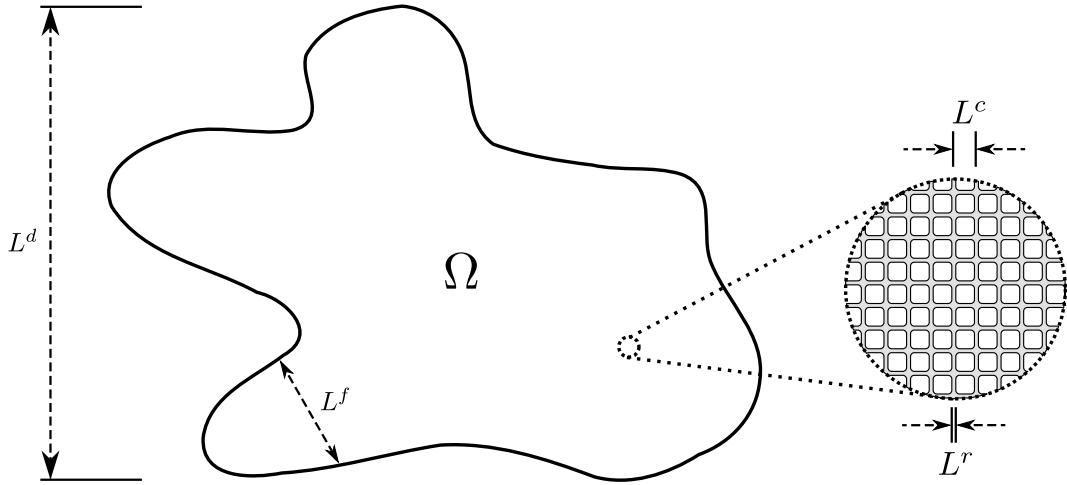


Figure 7.15: Length-scales present in a multi-scale irregular domain

These feature-lengths are as follows:

- \mathbf{L}^r – Rib or strut length scale
- \mathbf{L}^c – Unit cell length scale (may be less identifiable for natural structures)
- \mathbf{L}^f – Macroscopic feature size
- \mathbf{L}^d – Domain length scale

It may also be possible to identify an additional length-scale, L^g , over which some aspect of the micro-architecture's geometry varies. Although it is perhaps not so clear as to how this scale is identified.

Assuming a clear separation of length-scales, it may be possible to construct a relationship between the identified feature-lengths, as expressed by Eq 7.55.

$$L^r < L^c < L^g < L^f < L^d \quad (7.55)$$

However, it is possible there may be cases where $L^f < L^g$, that is, a small rate of change in micro-architecture across the domain.

For many problems it is highly impractical to attempt to include all length-scales in a finite element model, consequently it is often desirable to only capture the coarser details. Rather than simply excluding the smaller length-scales we aim to produce a coarse mesh with appropriate homogeneous material properties. Thus there is a need to be able to compute the apparent properties for each element in this coarse mesh (previously referred to as *macroelements*), as has been demonstrated in §7.3. While clearly not a direct replacement for a full resolution model, this approximate, or macro, model should prove satisfactory for modelling the bulk response of the domain.

As with any approximation, this macro model with has some disadvantages. Most notably the inability to capture local effects, such as displacement or stresses in ribs. It may, however, be possible to recover these local effects by applying the macroscopic displacements (resulting from a macroscopic simulation) to the microscopic model. To overcome boundary effects a number of macroelements may be required for this purpose.

Simulations performed using the macro model will also have to be limited to small strain to ensure they stay within the linear response of the structure. This limitation is the result of performing linear simulations in order to determine the effective constitutive matrix of a sub-volume; the apparent properties will only be valid for the linear response of the sub-volume.

7.5.1 Model Generation

To generate an approximate model from an original multi-scale dataset an appropriate macro-scale mesh must first be generated. As the aim is to eventually model the domain as homogeneous the macro-scale mesh must reflect this. The homogeneous domain should fit the bounds of the original domain as closely as possible, representing the result of a ‘shrink wrap’ operation. Ideally such an operation would be automated, however for simple geometries it is possible to manually define the homogeneous domain.

For example, when using the commercial software ScanIP the original resolution dataset would be imported and very roughly segmented in order to establish the bounds of the geometry as in Fig 7.16(b). The resulting dataset is referred to as a *mask*. Using the tools available, such as flood-fill, paint and morphological close, the mask can be filled-in as in Fig 7.16(c).

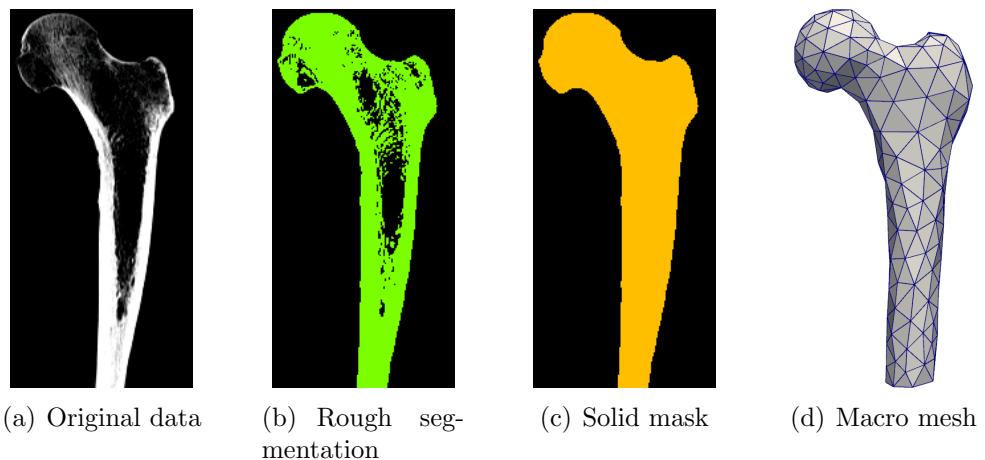


Figure 7.16: Macro mesh generation steps

The solid mask generated from this process will still be at the same resolution as the original volume. If a mesh were to be generated from this mask

using a marching cubes based mesher then the resulting mesh could potentially consist of hundreds of millions of elements. Ideally some form of mesh-based decimation would be utilised to severely reduce the number of elements, however, meshing and decimating such a high-resolution volume would itself be extremely computationally demanding. An alternative solution is to re-sample the volume to a considerably lower resolution so that the resulting mesh consists of an appropriate number of elements. Typically, for the same number of elements, the down-sampling approach will de-feature the volume more than the decimation of a high resolution mesh. For cases where $L^f \approx L^d$ this may be acceptable, in other cases a compromise can be taken by using both down-sampling and mesh-based decimation. Figure 7.16(d) shows an example macro mesh generated using this process.

With an appropriate macro-scale mesh generated the effective properties of each of its elements can be computed using the methods developed in §7.3. To do so requires an ‘accurate’ segmentation of the volume at the original resolution which may be generated using traditional segmentation techniques. Once these values have been established the mesh, with new material properties, can be saved as an input file for the finite element package of choice.

7.5.2 Example

The example chosen is a simple cantilever beam composed of a lattice structure. The length of the beam is 4,000 units with an aspect ratio of 1 : 10. The lattice structure is the Schoen Gyroid at 15% volume fraction with a normalised Young’s modulus ($E = 1$) and $\nu = 0.3$ for material properties. There are $10 \times 10 \times 100$

unit cells within the beam. It is estimated that the full resolution problem would require 615 million tetrahedral elements to sufficiently discretise the structure. To generate the approximate model the domain is discretised using 580 4-noded tetrahedral elements and processed using the methods described. The results from this process are then converted to an input file for the open source FE solver CalculiX. Appropriate boundary conditions are then applied, in this instance nodes at $x = 0$ are fixed and those at $x = 4000$ are loaded with a force of 0.1 N. The resulting deformed mesh is shown in Figure 7.17.

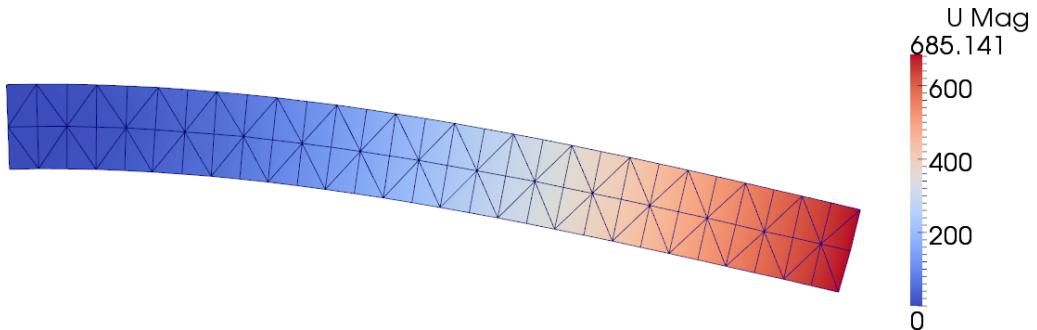


Figure 7.17: Displacement of an approximate model of a cantilever lattice beam

Clearly the full resolution model is too large to repeat the simulation for comparison without super-computing facilities. However, from the results in §7.4.2 it is clear that macroelements should contain at least $8 \times 8 \times 8$ unit cells. In this instance each unit cell will contain considerably fewer than 512 unit cells, therefore causing the apparent stiffness to be over-predicted. While increasing the size of the macro elements could potentially alleviate this effect, it will, however, cause over-stiffening due to the reduced number of DOF in the model. We have therefore identified two, seemingly competing, sources of over-stiffening in the modelling of this structure. It is possible to perform additional simulations so that the influence of each of these effects can be studied independently. The

over-stiffening due to the coarse mesh used can be measured by creating a finer mesh such that mesh independence is achieved. This finer mesh can be created by sub-dividing the original, with the new elements inheriting their parent's material properties. The influence of the second effect, the over-stiffening due to less than representative macroelements, can be measured by using the apparent properties established in §7.4.2 with the fine mesh. The results of performing the simulations with these models are given in Table 7.6.

Model	Num. Elements	Max Displacement
Original	580	658.141
Sub-divided	83,520	728.813
RVE material	83,520	746.863

Table 7.6: Comparison of different homogenised beam models

As may have been expected, the two additional models produced progressively less stiff responses. The introduction of additional degrees of freedom, as in the sub-divided model, has been shown to have the greatest influence on the stiffness of the structure. Despite this, it is clear that the over-predicted material properties in the original model also contribute significantly to the models over-stiffness, as seen by the difference in displacement for the ‘sub-divided’ and ‘RVE material’ models. It should be noted that relative contribution of these two effects, as shown here, is only valid for this specific model.

Based on these observations we may conclude that, while this size problem does appear large, it is not sufficiently large to ensure macroscopic mesh independence and that each macroelement is an RVE. However, from the results presented here it is clear that the two causes of over-stiffening can be alleviated to a degree through the sub-division of the macro mesh.

7.5.3 Property Visualisation

Another possible use for the models produced using the approach described in this chapter is in the visualisation of effective properties. As each element in the macro mesh is associated with an effective constitutive matrix it is possible to visualise the model with each element's colour determined by this matrix. In doing so we can visualise both relative and absolute variations of some chosen property across the domain. To achieve this a single value from the constitutive matrix must be chosen to be mapped to a colour. For example:

$$E_{xx} = \frac{1}{\mathbf{S}_{11}}, \quad G_{yz} = \frac{1}{\mathbf{S}_{44}} \quad (7.56)$$

with similar expressions for E_{yy} , E_{zz} , G_{zx} and G_{xy} , where \mathbf{S} is the compliance matrix. Alternatively, an averaging scheme may be used to find ‘equivalent’ isotropic properties (Marmier *et al.* [2010]). Three possible schemes are: Voigt [1910], Reuss & Angew [1929] and Hill [1952].

The Voigt scheme uses weighted averages from the stiffness matrix, \mathbf{C} , to compute the bulk modulus K and shear modulus G :

$$K_V = \frac{A + 2B}{3}, \quad G_V = \frac{A - B + 3D}{5} \quad (7.57)$$

where

$$A = \frac{\mathbf{C}_{11} + \mathbf{C}_{22} + \mathbf{C}_{33}}{3}, \quad B = \frac{\mathbf{C}_{23} + \mathbf{C}_{13} + \mathbf{C}_{12}}{3}, \quad D = \frac{\mathbf{C}_{44} + \mathbf{C}_{55} + \mathbf{C}_{66}}{3} \quad (7.58)$$

Similarly, the Reuss scheme is based on weighted averages from the compliance

matrix \mathbf{S} :

$$K_R = \frac{1}{3a + 6b}, \quad G_R = \frac{5}{4a - 4b + 3d} \quad (7.59)$$

where

$$a = \frac{\mathbf{S}_{11} + \mathbf{S}_{22} + \mathbf{S}_{33}}{3}, \quad b = \frac{\mathbf{S}_{23} + \mathbf{S}_{13} + \mathbf{S}_{12}}{3}, \quad d = \frac{\mathbf{S}_{44} + \mathbf{S}_{55} + \mathbf{S}_{66}}{3} \quad (7.60)$$

The Hill scheme is the arithmetic average of the Voigt and Reuss values (Marmier *et al.* [2010]).

For each scheme the equivalent Young's modulus and Poisson's ratio can be calculated using the standard relationship between E , ν , K and G for an isotropic material.

Figure 7.18 shows an example macro-scale model visualised using different techniques.

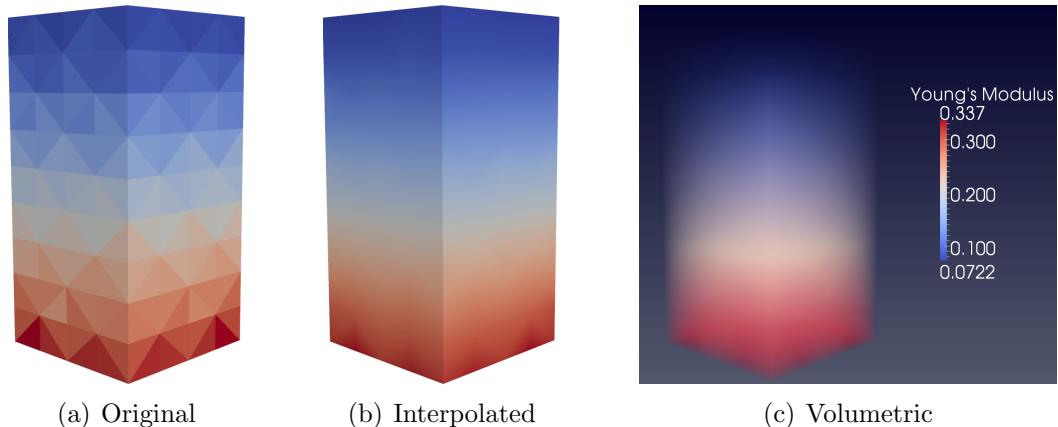


Figure 7.18: Visualising the variation of Young's modulus over a functionally graded structure

7.5.3.1 Out-of-Plane Properties

In addition to visualising the distribution of properties across a domain, it can also be useful to visualise the properties of a sub-volume (or in this instance a macro element). As shown in the previous section, the effective constitutive matrix of a macro element can be used to extract a number of properties. These properties correspond to the sub-volume's current coordinate system. However, it is possible to extract more information from the matrix by performing various rotations in order to change the coordinate system. By doing so it is possible to examine the effective out-of-plane properties of a sub-volume.

By visualising these properties it is possible to identify non-obvious properties of a structure, such as orientations at which a negative Poisson's ratio can be observed. They may also aid in locating maxima and minima of properties, such as directions in which the Young's modulus is maximised/minimised. The following gives a brief overview of how this is achieved, Marmier *et al.* [2010] provides more complete details.

Using summation notation, the rotation of a compliance tensor can be expressed by:

$$S_{pqrs} = r_{pi}r_{qj}r_{rk}r_{sl}S_{ijkl} \quad (7.61)$$

where r is the orthogonal rotation matrix (following the X-Z'-X'' convention):

$$r = \begin{bmatrix} c_2 & -c_3s_2 & s_3s_2 \\ c_1s_1 & c_1c_2c_3 - s_1s_3 & -c_2c_1s_3 - c_3s_1 \\ -c_1s_2 & c_1s_3 + c_3c_2s_1 & c_1c_3 - c_2s_1s_3 \end{bmatrix} \quad (7.62)$$

Here the trigometric terms have been abbreviated. For example c_2 stands for

$\cos(\beta)$ and s_3 is in-place of $\sin(\gamma)$. The three angles referred to in this matrix are the Euler angles $\alpha \in (-\pi, \pi]$, $\beta \in [0, \pi]$ and $\gamma \in (-\pi, \pi]$. By iterating over these ranges with an appropriate step size the compliance matrix for all possible orientations can be computed. From this the out-of-plane properties can be extracted.

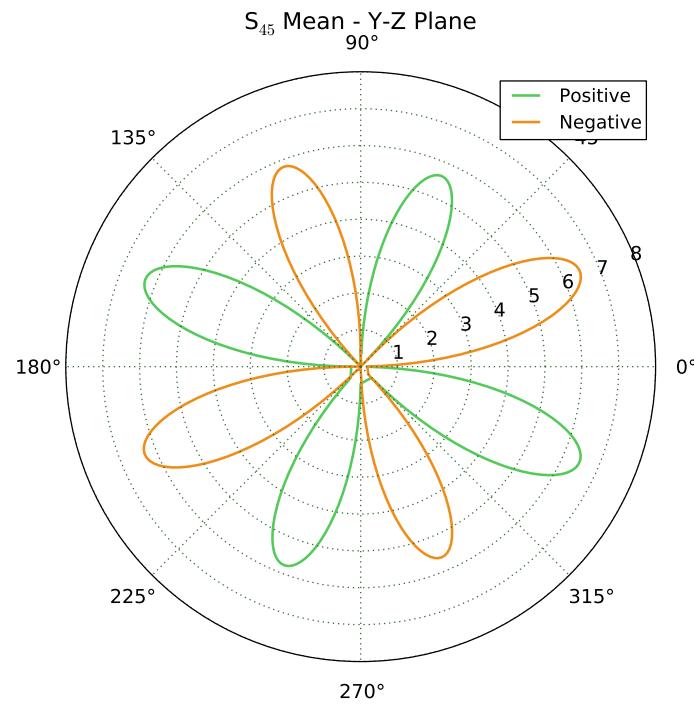
Simple properties such as the Young's modulus will remain invariant through rotations about X'' (i.e. with γ) and so may be plotted straightforwardly using polar or spherical plots. More complex properties, such as Poisson's ratio, are dependant on perpendicular stresses and therefore vary with γ . In order to plot these properties only the minimum, maximum and mean values are considered (Marmier *et al.* [2010]).

Fig 7.19 shows some example plots using the compliance matrix computed for the Schoen Gyroid structure at 15% volume fraction, with isotropic material properties $E = 1$ and $\nu = 0.3$.

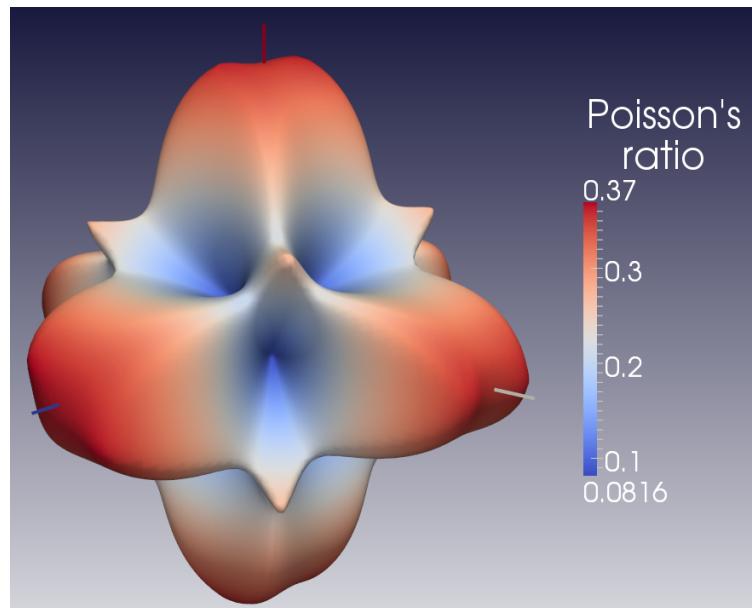
7.6 Applications

The homogenisation technique described in this chapter, while applicable to many multi-scale applications, may be best utilised with a certain set of applications. These are the applications in which the vast majority of macroelements contain a spatially varying or irregular structure. In addition to this the macroelements should each contain different representative samples of the structure.

In cases where these two conditions are not met, such as when the domain is composed of a periodic structure, the described technique will likely require an excessive number of virtual tests to be performed. This is due to the fact that, in



(a) Polar plot of the mean value of S_{45} in the Y-Z plane



(b) Spherical plot of the minimum Poisson's ratio

Figure 7.19: Plots of different out-of-plane properties for the Schoen Gyroid at 15% volume fraction

3D, the six required tests are performed on each macro element to determine their apparent properties. For the cases where there does not exist sufficient macroscopic variation in the structure many of the macroelements will have similar apparent properties. Consequentially, many of the virtual tests performed would be redundant and a more traditional RVE-based approach (as described in §6.1) would be more appropriate.

7.7 Comparison to Superelements

The methods described in this Chapter have been developed to address large multi-scale problems by diving them into smaller, more manageable sub-problems. This type of strategy can be generally described as being a “divide and conquer” approach to the problem. A similar method was developed in the 1960s (Felippa [1986]) by aerospace engineers based on the hierarchical decomposition of a complex structure into substructures. The substructures could effectively be modelled independently of one another to produce reduced models, or *superelements*, which could then be used to construct a reduced model of the original structure. For example, an aircraft wing may be decomposed into smaller substructures, but itself be a superelement when performing simulations on the entire aircraft.

In general superelements are constructed from a collection of elements or other superelements. The choice of these elements is driven by two different, yet potentially overlapping, views; *bottom up* or *top down*. When taking a bottom up approach superelements are built up from a collection of primitive elements to the desired size. Superelements constructed in this fashion are often classified as *macroelements*. Alternatively, a top down approach can be taken whereby

a structure is decomposed into *identifiable* smaller structures. For example, an aircraft being split into fuselage and wings. These superelements are classified as *substructures*.

The distinction between macroelements and substructures may not always be a clear one. For certain applications, particularly “non-CAD” geometries such as natural structures, there may not exist such well-defined substructures and hence the classification of a superelement as either a macroelement or substructure becomes difficult. As there are no set rules for such a classification, the term ‘superelement’ is generally used in both cases.

Despite the distinction made between macroelements and substructures the process of creating the superelements in both cases is the same, regardless of the number of elements involved. This process involves effectively eliminating all of the displacement degrees of freedom (DOF) associated with internal DOFs using a technique called *static condensation*. Internal DOFs are those which are not connected to DOF outside of the superelement. Degrees of freedom which are not designated internal are boundary DOFs and may be connected to additional DOF outside of the superelement.

Felippa [1986] demonstrates how the process of static condensation eliminates internal degrees of freedom using explicit matrix operations. The assembled stiffness equations for a superelement are considered:

$$\begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b \\ \mathbf{f}_i \end{bmatrix} \quad (7.63)$$

where the sub-vectors \mathbf{u}_b and \mathbf{u}_i represent the boundary and internal degrees of

freedom, respectively. Given the second equation:

$$\mathbf{K}_{ib}\mathbf{u}_b + \mathbf{K}_{ii}\mathbf{u}_i = \mathbf{f}_i \quad (7.64)$$

it is possible to solve for the internal displacements if \mathbf{K}_{ii} is non-singular:

$$\mathbf{u}_i = \mathbf{K}_{ii}^{-1}(\mathbf{f}_i - \mathbf{K}_{ib}\mathbf{u}_b) \quad (7.65)$$

Substituting Eq 7.65 into the first equation from Eq 7.63 yields the condensed stiffness equations:

$$\tilde{\mathbf{K}}_{bb}\mathbf{u}_b = \tilde{\mathbf{f}}_b \quad (7.66)$$

In this equation (Eq 7.66),

$$\tilde{\mathbf{K}}_{bb} = \mathbf{K}_{bb} - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib}, \quad \tilde{\mathbf{f}}_b = \mathbf{f}_b - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{f}_i \quad (7.67)$$

are the condensed stiffness matrix and force vector, respectively, of the substructure. In order to treat the superelement as a ‘regular’ element in the model, these values are taken to be the element’s stiffness matrix and nodal force vector, respectively. Following the static condensation of the internal DOF, the processing of the model can continue without any special consideration for the superelement.

The methods developed in this chapter used multiple finite element simulations to compute the effective stiffness of a macroelement. From this stiffness matrix the effective constitutive matrix, \mathbf{D} , is calculated. Calculating this matrix, rather than the stiffness matrix alone, allows the process to be independent of a specific solver as existing solvers will accept anisotropic material properties

for the macroelements. In comparison, the superelement approach requires a certain degree of integration with the solver. However, it does also offer a number of advantages in some applications. For instance, in the case where substructures are being used it may be possible to exploit repetition in the structure to re-use superelements and hence reduce the number of static condensation operations required. This is aided by the fact that, with the superelement approach, the division of the structure into substructures is a manual process whereas the proposed method utilises a semi-automated meshing process. The substructure approach can also be used to facilitate the division of labour for large models as substructures can be developed independently, providing their interface to other superelements remains consistent (Felippa [1986]).

Both the proposed method and the superelement approach have the advantage of overcoming the computational limitations present when processing large models. Both can be said to take a “divide and conquer” approach, however, the way in which they achieve this differs considerably. An important difference between the two methods lies in how the superelements are constructed. In the case of the proposed method, the contents of each macroelement is homogenised resulting in an approximation of its behaviour. While the superelement approach does eliminate internal degrees of freedom, it is possible to recover them following the completion of the simulation therefore potentially providing an exact solution.

7.8 Conclusions

Large multi-scale problems are computationally challenging even for the simple case of linear static simulations.

The methods developed in this chapter have shown how these problems can be addressed through the generation of an approximate homogeneous model.

For very large linear static problems, where there exists a clear separation of length scales, the division of the problem into many smaller disjoint problems can result in considerably reduced computational requirements in comparison to the construction and solving of a full model.

Such a division of the problem also brings with it a number of advantages. Most notably is that the division creates independent sub-volumes (unlike mesh decomposition methods used in the parallel computation of full resolution models which require some degree of intercommunication) allowing the model generation process to be very efficiently run in parallel either on a single computer or cluster with MPI. Alternatively, this also allows the entire problem to be processed in series using only a single CPU. The main limiting factor which will determine whether or not a problem is too large to approximate is the amount of memory required to solve the micro-scale problems for the largest sub-volume. Providing there is sufficient main memory available to achieve this then problems of any size may be processed either in parallel or series with very modest hardware requirements.

From the results given in §7.5.2 it appears that the homogenisation technique over-predicts the stiffness of a model when the macro elements contain less than representative volumes. As the size of the macro element was increased the effec-

tive stiffness decreased, suggesting that the macroelements should ideally contain representative volumes. Given the need to have statistically representative volume elements with other forms of homogenisation and characterisation this is not surprising. As a direct consequence of this the minimum size problem which can be approximated (for the purpose of performing macro-scale simulations) dramatically increases. To achieve RVE macroelements it would not be sufficient to simply increase the size of the macroelements as this would have two undesirable side effects: firstly, this would further de-feature the domain and more importantly it would also be removing degrees of freedom from the macro model, therefore increasing the model's stiffness. Due to this minimum size increase the model approximation will likely now only be suitable for very large problems where convergence can be achieved on both lengths, that is, convergence to and RVE at the micro scale and convergence at the macro scale (i.e. mesh independence). Consequently this may leave a set of problems which, without super computing facilities, are too large to solve at the original resolution, but that are also too small to achieve convergence on both length scales.

It has also been shown that the tetrahedral-based homogenisation technique developed has potential uses for material characterisation. However, the advantages of using tetrahedral sub-volumes in this case can become disadvantages. As many samples are usually cubic or cuboidal (or easily cropped to such a shape with minimal loss) we loose two-thirds of the volume due to the largest tetrahedron which can fit inside the sample. For synthetic or generated structures, where a larger volume can be generated, this may not be an issue. For physical samples which have been imaged there is the possibility that one-third of the volume is no longer statistically representative. However, it may be possible to

extend the methods developed in this Chapter to element types better suited to cuboidal samples, such as hexahedral elements – a topic for further research.

One point of interest is that despite the different approach taken to developing this homogenisation technique the resulting boundary conditions bear notable resemblance to those in kinematic uniform boundary conditions (KUBC). In both cases the boundary conditions restrict the movement of the surface nodes to the movement of the domain. The choice of macro element in this instance also means the displacements are imposed as a linear function of the geometry. The results presented in §7.4.2 show good agreement between the two homogenisation methods thus validating the developed technique.

Chapter 8

Conclusions

8.1 Summary

Due to their high surface to volume ratio and porosity there are a wide range of potential applications, from industrial to medical, for open cell internal micro-architectures. The inherent topological complexity of these structures also make them difficult to both generate and model computationally. In this thesis methods have been presented and explored to generate domain conforming lattice structures and to model the macroscopic behaviour of such structures.

The use of implicit modelling, particularly with triply periodic functions, has been shown to be highly flexible for the generation of open cell micro-architectures. Several image-based algorithms were developed and evaluated and it was clearly shown that, to achieve accurate and smooth models, the additional computation required to generated greyscale volumes was an acceptable overhead. By taking an image-based approach it is possible to ensure that the generation and manipulation of the micro-architectures remains robust, despite any geomet-

ric complexities. The image-based approach also allows the micro-architectures to be straightforwardly integrated with existing image data, as may be required in medical applications, such as for the generation of conforming scaffolds. However, this method does inherit some disadvantages as a result of the use of imaging. Notably this includes the inability to represent sharp features and the requirement of high resolution volumes for low volume fraction structures, resulting in an excessive number of elements (for finite element modelling) and/or triangles (for ALM). An issue addressable to a certain degree by the use of decimation. The advantages, however, considerably outweigh the disadvantages.

Algorithms were developed allowing for the creation of micro-architectures with a specified volume fraction by choosing an appropriate value at which to iso-surface. By using a variable iso-surface value variations in density could be introduced to a structure, either defined by a mathematical expression or through the use of a density map. These variations in density in-turn introduce variations in properties, such as stiffness, allowing for the creation of functionally graded structures.

The algorithms presented were then extended for the purpose of generating domain conforming internal structures where the domain is enclosed in, or bounded by, a shell. By sampling the external geometry it was possible to work entirely in image-space to generate the new internal surface. While this clearly also has disadvantages, such as de-featuring due to sampling, the vast majority of applications will be unaffected by a de-featured internal surface. An exception to this may be an external geometry with very small holes or insets. Although the methods used in Chen [2007a] preserve feature edges on the internal surface, this may not always be desirable. The methods developed (as in the patent applica-

tion in Appendix B) allow the majority of processing to be performed in parallel so that internal structures can be quickly introduced into CAD models.

In Chapter 6 a number of simple geometries with internal structures were evaluated to determine the influence of introducing an internal structure. Results obtained via mechanical testing and finite element analysis showed very poor agreement. They did, however, agree that in the vast majority of cases the models with internal structures were stiffer than their hollow equivalents. Further tests also demonstrated the use of the internal structures as supporting structures in the fabrication of metal components via ALM. This shows that the structures have excellent multi-functional potential where their mechanical (or thermal, etc ...) properties are not only useful for the function of a component, but also during its manufacturing. With increasing interest and recent advances in additive layer manufacturing technologies multi-functional structures, and the tools for their generation and evaluation, are likely to become increasingly popular.

As has already been stated, following the generation of a micro-architecture it is often desirable to evaluate some property of the structure, such as its effective properties. The methods developed in this work allow the efficient and reliable creation of volumetric meshes using well-established meshing techniques and software (⁺ScanFE in this instance). These meshes may be used to characterise the structure using existing homogenisation techniques. However, the work in this thesis attempts to address the issue of large multi-scale structures. In order to do so a novel approach to homogenisation was developed utilising the concept of macroelements, where the macroelements in this instance are tetrahedra. The use of tetrahedra allows arbitrary domains to be divided into independent sub-volumes, each then characterised so that a homogeneous macroscopic model can

be constructed. The homogenisation itself was shown to yield apparent properties very close to those obtained using the frequently used kinematic uniform boundary conditions. However, it was also shown to have similar drawbacks, in particular that the restrictive boundary conditions mean that the RVE must be very large to overcome ‘edge effects’. For the case where the RVE is not large enough for this we saw over-predicted properties, as is commonly seen when using KUBC. The consequence of this is that the size of the problem required to achieve convergence at both length scales (i.e. to effective properties and macroscopic convergence) is considerably larger than expected. However, the method itself remains a valid approach to large multi-scale problems.

8.2 Research Contributions

The research contributions may be briefly summarised as follows:

1. The development of image-based algorithms for the efficient generation of implicitly defined periodic micro-architectures. The developed algorithms allow for functionally and arbitrarily graded structured micro-structures with pre-specified volume fractions. The image-based approach allows for the fast and robust generation of volumetric and surface meshes.
2. The development of algorithms for the generation of transition cells for the purpose of bridging two implicitly defined unit cells of differing topology. By combining unit cells of different topologies custom properties can be achieved without the need to vary the volume fraction.
3. The development of image-based algorithms for the generation of domain

conforming internal structures (with or without grading variations), including a novel approach to the creation of hollow components from CAD models. This approach also allows for the robust introduction of internal structures using the developed algorithms.

4. The investigation into the mechanical properties of the Schoen Gyroid at various volume fractions. This includes the discovery of an appropriately sized sample to simulate when using uniform boundary conditions.
5. The discovery that, at certain cell sizes, certain lattice structures, based on triply periodic implicit functions, can be manufactured at all orientations via SLM. This suggests that the lattice structures can act not only as internal supporting structures, but have the potential to also be used as external supporting structures.
6. The development of a novel approach to linear elastic homogenisation based on non-cuboidal domains, triangles and tetrahedra (in 2D and 3D, respectively) in this instance. The developed method has been shown to yield apparent properties comparable to those obtained via the often used kinematic uniform boundary conditions.
7. The development of a technique for creating approximate homogeneous models from large multi-scale linear elastic problems. By utilising the developed tetrahedral based homogenisation large problems can be processed efficiently either in parallel or in series. Thus the method is capable of reducing the computational requirements for evaluating the bulk response of such structures. It may also be used as a tool for visualising the macroscopic

distribution of properties over a domain.

Chapter 9

Recommendations for Future Work

As there is a limited amount of time in which a thesis such as this may be completed there will invariably be areas which can be developed further or explored in greater depth. The following comprises of a number of possible research tasks for future work. They are presented in no particular order.

Micro-Architectures

1. With the ability to generate periodic lattice structures with arbitrary density variations, either domain conforming or within a shell, an obvious question arises: *What is the optimal density variation for a given application?* As is suggested in the question, the answer will likely require an iterative optimisation process wherein many possible solutions are evaluated. However, as we now know, evaluating a single micro-architecture is computationally expensive, let alone many thousands iteratively. The optimisation problem

itself is one of optimal material distribution, similar to that in topology optimisation. Further research into how topology optimisation methods, particularly those in Bendsøe & Kikuchi [1988] and Young *et al.* [1999], may be adapted for this purpose.

2. This thesis has highlighted the multi-functional nature of implicitly defined periodic micro-architectures, particularly when combined with additive layer manufacturing technologies. However, some aspects have only been touched upon. Further research into the fabrication of these structures, particularly out of different metals and bio-compatible materials such as hydroxyapatite could potentially extend their uses. Addressing practical issues, such as the removal of the lattice when used as a supporting structure and the slicing of very large STL files, could also be beneficial.
3. The development of an algorithm to determine the feature size of a CAD model, in order to provide an appropriate scale to voxelise it at, would improve the efficiency of the internal structure algorithm. It would also reduce the user interaction with the algorithm and potentially remove the need to occasionally re-run the process at different resolutions.
4. Further exploration of implicit and constructive solid geometry modelling for the purpose of generating lattice structures using the methods developed in this thesis. These methods would allow unit cells to be designed using more ‘CAD-like’ techniques.

Homogenisation

1. Currently the only types of element which have been used as macro elements

are the simple linear triangle and tetrahedron. These elements are often regarded as poor due to their over-stiffness and constant strain field. Future work could examine the use the more complex high-order elements and their effect on the apparent properties of a sub-volume. Points of interest may include; *Do the more complex displacement functions relax the boundary conditions?* and *How do the additional macro nodes influence the choice of virtual tests?* There may also be benefits to using hexahedral elements.

2. The homogenisation approach taken was developed for linear elastic properties. Further research into the possible use of this approach for other properties, such as permeability, thermal, electromagnetic or even non-linear elasticity may further extend the applications of this multi-scale approach.

Appendix A

Unit Cell Implicit Functions

Schwarz Primitive:

$$\cos(x) + \cos(y) + \cos(z) - 1 = 0$$

Schoen Gyroid:

$$10(\cos(x) \times \sin(y) + \cos(y) \times \sin(z) + \cos(z) \times \sin(x)) - \\ 0.5(\cos(2x) \times \cos(2y) + \cos(2y) \times \cos(2z) + \cos(2z) \times \cos(2x)) = 0$$

Schwarz W:

$$10(\cos(x) \times \cos(y) + \cos(y) \times \cos(z) + \cos(z) \times \cos(x)) - \\ 5(\cos(2x) + \cos(2y) + \cos(2z)) - 14 = 0$$

Schwarz W:

$$\begin{aligned} & 10(\sin(x - \frac{\pi}{4}) \times \sin(y - \frac{\pi}{4}) \times \sin(z - \frac{\pi}{4}) + \sin(x - \frac{\pi}{4}) \times \\ & \cos(y - \frac{\pi}{4}) \times \cos(z - \frac{\pi}{4}) + \cos(x - \frac{\pi}{4}) \times \sin(y - \frac{\pi}{4}) \times \\ & \cos(z - \frac{\pi}{4}) + \cos(x - \frac{\pi}{4}) \times \cos(y - \frac{\pi}{4}) \times \sin(z - \frac{\pi}{4})) - \\ & 0.7(\cos(4x) + \cos(4y) + \cos(4z)) - 11 = 0 \end{aligned}$$

Schwarz Primitive (Pinched):

$$\begin{aligned} & -(2.25(\cos(x) + \cos(y) + \cos(z)) + \\ & 1.25(\cos(x) \times \cos(y) + \cos(y) \times \cos(z) + \cos(z) \times \cos(x))) = 0 \end{aligned}$$

Neovius' Surface:

$$\begin{aligned} & -(-\sin(x) \times \sin(y) \times \sin(z) + \sin(2x) \times \sin(y) + \sin(2y) \times \\ & \sin(z) + \sin(x) \times \sin(2z) - \cos(x) \times \cos(y) * \cos(z) + \\ & \sin(2x) \times \cos(z) + \cos(x) \times \sin(2y) + \cos(y) \times \sin(2z)) = 0 \end{aligned}$$

Appendix B

Internal Structures Patent

Patent pending. Filed in the UK and USA.

Application Data Sheet

Application Information

Application Type: Nonprovisional
Subject Matter: Utility
Title: IMAGE PROCESSING METHOD AND
METHOD OF THREE-DIMENSIONAL
PRINTING INCORPORATING THE SAME
Attorney Docket Number: 0410/2437-US0
Total Drawing Sheets: 5
Small Entity?: Yes
Petition included?: No
Request for Early Publication ?: No
Request for Non-Publication?: No
Secrecy Order in Parent Appl.?: No

Applicant Information

Applicant Authority Type: Inventor
Primary Citizenship Country: Great Britain
Status: Full Capacity
Given Name: Philippes
Middle Name:: Georges
Family Name: Young
City of Residence: Lympstone
State or Province of Residence:: Devon
Country of Residence: United Kingdom
Street of mailing address: Longmeadow Road
City of mailing address: Lympstone
State of mailing address: Devon
Country of mailing address: United Kingdom
Postal or Zip Code of mailing address: EX8 5LW

Application Data Sheet

Applicant Authority Type: Inventor
Primary Citizenship Country: Great Britain
Status: Full Capacity
Given Name: David
Family Name: Raymont
City of Residence: Okehampton
State or Province of Residence:: Devon
Country of Residence: United Kingdom
Street of mailing address: 49 Balmoral Crescent
City of mailing address: Okehampton
State of mailing address: Devon
Country of mailing address: United Kingdom
Postal or Zip Code of mailing address: EX20 1GN

Correspondence Information

Correspondence Customer Number: 76808

Representative Information

Representative Customer Number: 76808

Domestic Priority Information

Foreign Priority Information

Application Number:	Filing Date:	Country:
1003065.8	February 23, 2010	Great Britain

Application Data Sheet

Assignee Information

Assignee name: Simpleware Limited
Street of mailing address: Bradninch Hall, Castle Street
City of mailing address: Exeter
State or Province of mailing address: Devon
Country of mailing address: United Kingdom
Postal or Zip Code of mailing address EX4 3PL

Signature

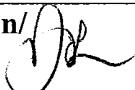
Signature	/david leason/ 			Date (YYYY-MM-DD)	2011-02-18
First Name	David	Last Name	Leason	Reg. No.	36,195

IMAGE PROCESSING METHOD AND METHOD OF THREE-DIMENSIONAL
PRINTING INCORPORATING THE SAME

This patent application claims the benefit of priority
5 under 35 U.S.C. Section 119(e) from Great Britain Patent
Application No. 1003065.8, filed on February 23, 2010, now
pending and entitled "Image Processing Method and Method of
Three-Dimensional Printing Incorporating the Same", which is
hereby incorporated by reference in its entirety.

10

FIELD OF THE INVENTION

The invention relates to the preparation of image data
for three-dimensional printing, e.g. for use in the
manufacture of three-dimensional models in rapid prototyping
15 apparatus or the like, where a three-dimensional model is
built by successive deposition of a plurality of layers, e.g.
using selective laser sintering and selective laser melting.

BACKGROUND TO THE INVENTION

It is known to build physical three-dimensional models
20 using additive manufacturing technology. Typically, a virtual
design of the three-dimensional model, e.g. represented
through computer-aided design (CAD) software or the like, is
transformed into a plurality of thin (quasi-two-dimensional)
25 cross-sectional layers which are built on one another in
succession.

In some circumstances, it may be desirable to replace the
interior of a virtual solid three-dimensional model (e.g. a
30 CAD model, in any of a number of possible formats) with a void
or an internal framework. A model with an internal void may
resemble a shell of the original model. An internal framework
may comprise an array of support struts or a microstructure
such as an open-celled foam or the like. There are a number
35 of reasons for replacing the solid interior of an original
model. One reason may be to use less material in the creation
of a physical model (e.g. by a three-dimensional printer).
Another reason may be to create a lighter model whilst
maintaining appropriate stiffness or rigidity. A further
40 reason may be to enhance the model's physical properties in
some way, e.g. to prevent warping or the like. For example,

an internal microstructure may give an object different electromagnetic properties, which may be of interest in the field of radar technology.

The conventional approach to hollowing out a solid CAD object starts with a CAD representation of the object, which is a collection of vector-based representations of graphical entities corresponding to the different outer surfaces of the object. To achieve hollowing out, the conventional approach moves the outer surfaces inwards by a fixed amount (typically specified by the user). The set of surfaces spawned by this movement are designated as internal surfaces, whereby the original (external) and spawned (internal) surfaces are used to define the exterior and interior of the hollowed out model. This process is sometimes called shelling or offsetting.

Difficulties can arise with this conventional approach because displacing outer surfaces inwards can cause the formation of new intersections or result in surfaces not meeting at edges. It is possible to handle these difficulties for relatively simple objects, although inconvenient if it must be done manually. However, for more complex objects, such as those requiring a microstructure to be formed in the interior of the object, these difficulties present significant problems.

One known attempt to introduce a microstructure into a three-dimensional CAD object involved merging a CAD description of a microstructure with the internal surface(s) of the hollowed out model spawned by the shelling process described above. This technique is difficult and unreliable, e.g. because the intersections between the microstructure and the internal surface(s) of the hollowed out model need to be recomputed in order to create a new closed internal surface. As the complexity in the geometry of the microstructure increases, this process becomes non-trivial and far from robust and often results in undesirable sharp edges where the microstructure and the CAD model interface.

SUMMARY OF THE INVENTION

At its most general, the present invention proposes eroding a digitised (e.g. voxelized) representation of a virtual three-dimensional object (e.g. CAD model) to create the internal volume of the object. After the internal volume

is created in this way, a vector-based surface representation (e.g. CAD model, in STL format or the like) thereof can be generated and simply combined with a corresponding vector-based surface representation of the original virtual three-dimensional object to yield a hollowed out model in a format suitable for three-dimensional printing. A microstructure may be introduced into the interior by extracting a volume corresponding to the inverse of that microstructure from the eroded digitised representation and then generating a vector-based surface representation of the interior space from that extracted volume. In practice, this may be achieved by subtracting a digitised representation of a microstructure from the eroded digitised three-dimensional representation of the object, generating a vector-based surface representation of what in effect is the internal spaces of the microstructure, reversing the surface normals of the generated vector-based surface representation so that the outward facing surface(s) thereof becomes inward facing surface(s) and concatenating the result with the original vector-based surface representation of the object.

Obtaining a voxelized three-dimensional representation of a three-dimensional CAD model is a lossy transformation, and as such is a counter-intuitive step for those wishing to obtain an accurate physical model by three-dimensional printing, for whom remaining in CAD form is seen as essential. However, the present invention is built on the unexpected realisation that a switch to a digitised representation to create an interior surface (which can subsequently be converted to a vector-based surface representation) both improves the robustness of the shelling process and provides an efficient mechanism for introducing an internal microstructure. Moreover, because the new surfaces generated from the eroded model and the original CAD surfaces are nested, and therefore do not come into contact, appropriate computer descriptions of the internal surfaces generated can simply be concatenated with the original CAD model to form a new hollowed CAD model.

According a first aspect of the invention, there may be provided a method of preparing a virtual three-dimensional object for three-dimensional printing, the method comprising: obtaining a vector-based surface representation of a virtual

three-dimensional object; sampling the volume enclosed by the three-dimensional object to generate a digitised three-dimensional representation of the object; eroding the digitised three-dimensional representation of the object to generate an internal volume; generating a vector-based surface representation of the internal volume, wherein the surfaces of the internal volume represent internal surfaces; and combining the vector-based surface representation of the virtual three-dimensional object with the vector-based surface representation of the internal volume. The method may thus convert the vector-based representation to a digitised, i.e. granular, representation only for the purpose of erosion.

The virtual three-dimensional object may be a computer-aided design (CAD) model (e.g. used in engineering software packages such as AutoCAD and CATIA) or an output from any suitable computer graphics programme. The vector-based surface representation may be in any suitable format, e.g. STL (stereolithography), SAT, IGES, X3D (XML-based) or the like.

The digitised three-dimensional representation of the object may comprise a voxelized model of the virtual three-dimensional object, i.e. effectively a three-dimensional bitmapped image of a volume enclosing the virtual three-dimensional object. Defining the enclosing volume and performing the sampling steps may be achieved using known techniques. For example, binary sampling on a regular grid may provide a binary bitmapped representation. Alternatively, distance function sampling may be applied to provide a greyscale bitmapped representation in which the distance to the surface of each sample point is encoded using a greyscale value.

Eroding the digitised three-dimensional representation of the object may comprise removing voxels at the surface of the voxelized model in a conventional manner. For example, the erosion may utilise a morphological operator arranged to uniformly remove one or more layers of voxels from the outside surface of a bitmapped representation. Alternatively, the erosion may be refined to a sub-voxel resolution by modifying appropriately the greyscale values of surface voxels.

The removed layer of voxels may have a variable depth around the voxelized model. For example, it may be desirable for the shell to have different thickness. Accordingly the

number of voxels removed from the surface of the voxelized model may vary by location. Furthermore, if it is desirable for the shape of the surface of the internal volume to map closely the shape of the outer surface, the sampling rate may 5 be variable within the enclosed volume. For example, the sampling rate may be higher in a boundary region containing the outer surface of the virtual three-dimensional object and lower in the centre of the object. The variable sampling rate may be achieved by morphing the shape of a uniform sampling 10 mesh, as disclosed in GB 2 460 411.

A particular advantage of eroding a digitised representation to obtain an internal volume is the ease at which problems arising from the intersection of inwardly moved surfaces are avoided, because by definition the erosion acts 15 to shrink the object rather than merely displacing its surfaces.

In one embodiment, to ensure the surfaces of the internal volume represent internal surfaces the method may include 20 inverting the digitised representation of the internal volume before generating the vector-based surface representation of the internal volume, whereby the surfaces of the internal volume automatically represent internal surface upon generating the vector-based surface representation of the internal volume.

25 In another embodiment, instead of inverting the eroded image, the method may include designating an exterior surface defined by the vector-based surface representation of the internal volume as an internal surface before combining it with the vector-based surface representation of the virtual 30 three-dimensional object. Designating an exterior surface defined by the vector-based surface representation of the internal volume as an internal surface may include reversing the direction of a plurality of surface normals associated with the exterior surface.

35 The method may include applying a microstructure to the internal volume. For example, the method may include performing a union operation to combine a digitised representation of a microstructure with the inverted digitised internal volume, wherein the generated internal volume 40 consists of the microstructure.

Alternatively, the method may include, after eroding the digitised three-dimensional representation of the object, subtracting a digitised representation of a microstructure from the eroded digitised three-dimensional representation of the object, wherein the generated internal volume consists of the microstructure. Herein, subtracting means performing a voxel-by-voxel calculation in which the sampling value (e.g. binary or greyscale value) for each voxel of the digitised representation of a microstructure is subtracted from a corresponding voxel of the eroded digitised three-dimensional representation of the object. The result may be the creation of an inverted version of the microstructure in the generated internal volume.

Subtracting digitised representations in this manner carries less of a processing burden than intersecting vector-based volumes.

The digitised representation of a microstructure may be generated from tri-periodic mathematical functions or by sampling a virtual three-dimensional model of a microstructure, e.g. defined by a vector-based surface representation, or by scanning a real physical object. For example, the primitive (P) surface, the diamond (D) surface and the gyroid (G) surface may be used to represent microstructures. These minimal surfaces may be defined in mathematical functions using x, y and z coordinates. This method may include modifying the tri-periodic mathematical functions to introduce density variations inside the internal volume [1]. Computer-based analysis techniques, e.g. finite element analysis or the like, may be used to optimise the density variations. Auxetic microstructures or microstructures exhibiting a negative Poisson's ratio may be used.

Applying the internal volume (e.g. having a microstructure) to the (solid) interior of the original virtual three-dimensional object may be achieved in a number of ways, depending on the format of that object. In this aspect of the invention, the original object is defined by a vector-based surface representation. A vector-based surface representation of the internal volume may be generated using any of a number of meshing techniques, e.g. the marching cubes approach used in an image processing technique disclosed in US

4,710,876 which allows for reconstruction of smooth surfaces from image data. The vector-based surface representation of the internal volume may thus comprise a triangulated surface representation which effectively defines as its outer surface
5 the "new" inside surface of the object. By designating this outer surface as an internal surface, e.g. by reversing the direction of the surface normals associated with the outer surface, the internal volume can be combined with the original vector-representation to yield a hollow object.

10 Combining the vector-based surface representations of the virtual three-dimensional object and the internal volume may comprise performing a concatenation operation to unify sets of data corresponding to the original CAD representations of the virtual three-dimensional object and the newly generated
15 internal boundary respectively. The sets of data may be in a format suitable for such an operation, e.g. in STL format. The eroding step ensures that there is no intersecting of surfaces on the original vector-based representation of the object and the vector-based representation of the internal
20 volume, which permits straightforward use of concatenation operation.

25 In another aspect, the present invention is used in a process for creating a physical model using three-dimensional printing. In this aspect, the steps of the method described above may be followed by: outputting the combined vector-based surface representation to a three-dimensional printer; and operating the three-dimensional printer to create a physical model corresponding to the combined vector-based surface representation. In this context, three-dimensional printer
30 may mean a device for manufacturing a three-dimensional model by depositing a plurality of layers of material one of top of another. Conventional three-dimensional printers, e.g. rapid prototyping machines or the like, may be used.

35 In this aspect, the method may be executed by a suitably programmed computer in communication with the three-dimensional printer. Accordingly, the invention may include a computer program product comprising a computer-readable storage medium having software instruction stored thereon, the software instructions being executable by a computer to
40 perform the steps of a method as set out above.

In another aspect, the invention may be used for performing computer-based analysis of a hollowed virtual three-dimensional object. According to this aspect, there may be provided a method of preparing a virtual three-dimensional object for computer-based analysis, the method comprising:
5 obtaining a vector-based surface representation of a virtual three-dimensional object; sampling the volume enclosed by the three-dimensional object to generate a digitised three-dimensional representation of the object; eroding the
10 digitised three-dimensional representation of the object to generate an internal volume; combining the digitised three-dimensional representation of the object with digitised three-dimensional representation of the internal volume; and
15 performing computer-based analysis on the combined digitised three-dimensional representation.

This method may provide as its output a digitised representation of a hollowed-out virtual three-dimensional object. The interior of the object may have a microstructure, as disclosed above. However, in this case a vector-based
20 surface representation of the internal volume is not generated to create the final hollowed-out object. Instead, the digitised internal volume is combined with the digitised version of the original virtual three-dimensional object.
This combining step may be achieved by subtracting the
25 digitised internal volume from the digitised version of the original virtual three-dimensional object. The output is thus a voxelized representation of the hollowed out object, which can be segmented and analysed using conventional techniques.
For example, a suitably segmented representation may be
30 subjected to finite element or finite volume analysis.

In another aspect, the invention may be applied to assist with the analysis of how applying a microstructured interior to virtual three-dimensional object derived from a real (physical) three-dimensional object affects that object's
35 properties. According to this aspect, there may be provided a method of preparing a virtual three-dimensional object for three-dimensional printing, the method comprising: scanning a physical three-dimensional object to obtain a digitised three-dimensional representation of the object; eroding the
40 digitised three-dimensional representation of the object to generate an internal volume; subtracting a digitised

representation of a microstructure from the eroded digitised three-dimensional representation of the object, wherein the generated internal volume consists of the microstructure; and subtracting the digitised three-dimensional representation of the internal volume from the digitised three-dimensional representation of the object. The representation resulting from this image may be passed on to computer-based analysis tools as mentioned above, or may be passed to a three-dimensional printer for printing.

Alternatively or additionally, a vector-based surface representation of the digitised three-dimensional representation obtained by scanning the object may be generated and combined with a vector-based surface representation of the internal volume, e.g. for use in displaying and/or printing the object.

In another aspect, the invention may provide apparatus for preparing a virtual three-dimensional object for three-dimensional printing, the apparatus comprising: an input device for receiving a vector-based surface representation of a virtual three-dimensional object; a memory arranged to store the vector-based surface representation of a virtual three-dimensional object; a processor programmed to: sample the volume enclosed by the three-dimensional object to generate a digitised three-dimensional representation of the object, erode the digitised three-dimensional representation of the object to generate an internal volume, generate a vector-based surface representation of the internal volume, designate the vector-based surface representation of the internal volume as an internal surface, and combine the vector-based surface representation of the virtual three-dimensional object with the vector-based surface representation of the internal volume; and an output device for outputting the combined vector-based surface representation of the object and internal volume.

The input device may be a user-operated device, such as a mouse, keyboard or touch screen, or a device for receiving other types of data, e.g. electrical signals or binary code, such as an antenna, disk drive or the like. The output device may be any type of suitable communication apparatus for transmitting data representative of the combined vector-based

surface representation of the object and internal volume, e.g. a wireless transmitter, wired link or the like.

The output device may be connected to a display and/or a three-dimensional printer equipped to receive and process the data transmitted by the output device.

The memory may store thereon a library comprising a plurality of digitised representations of microstructures, wherein the processor is further programmed to: permit selection via the input device of a microstructure in the library; and intersect the eroded digitised three-dimensional representation of the object with a digitised representation of the selected microstructure, wherein the generated internal volume consists of the microstructure.

15 BRIEF DESCRIPTION OF THE DRAWINGS

An example of the invention is described below with reference to the accompanying drawings, in which:

Fig. 1 is a graphical illustration of the steps of a method that is an embodiment of the invention;

Fig. 2 is a flow diagram of the steps of a method that is an embodiment of the invention;

Fig. 3 is a graphical illustration of the steps of a method that is another embodiment of the invention;

Fig. 4 is a flow diagram of the steps of a method that is another embodiment of the invention;

Fig. 5 is a schematic diagram of a processing environment suitable for executing one or more steps of a method according to the invention; and

Fig. 6 is schematic diagram of an apparatus that is an embodiment of the invention.

30 DETAILED DESCRIPTION OF CERTAIN EMBODIMENTS, OPTIONS AND PREFERENCES

Fig. 1 illustrates a method that is an embodiment of the invention. The aim of this embodiment is to introduce into a solid (or filled) model an internal space, which internal space may have a microstructure, e.g. foam or the like.

The method starts from a vector-based representation of a three-dimensional model, which in this example is a perspective view of a cuboidal CAD model 10. A first step 50 in the method comprises converting the CAD model 10 into a

three-dimensional image or digital representation 12. This can be carried out by sampling the volume enclosed by the CAD model 10. In effect the model is converted into a three-dimensional bitmapped image, e.g. each bit representing a voxel of the digital representation. For convenience in this example, the voxels in the digital representation 12 are shown as cuboidal, but other suitable shapes may be used.

The conversion of the CAD model 10 into the digital representation 12 is a lossy transformation. Detailed features of the CAD model (e.g. at corners and edges) can be lost in the conversion. Conversion may be performed using a signed distance function rather than a straight binarisation to give a better representation. However, the accuracy of the outer surface of the digital representation 12 is not in fact critical, because it needs to be eroded anyway to fulfil its purpose of constructing a new internal surface for the CAD model 10.

A second step 52 of the method comprises eroding the three-dimensional digital representation 12 by a prescribed amount to create an eroded digital representation 14 (e.g. voxelized image) which is effectively shrunk in size from the original. This eroding step may be carried out using standard image processing techniques in which surface voxels to a depth of one or more are removed from all surfaces. The amount of shrinkage may correspond to one or more times the voxel side length. In a preferred embodiment, the surface voxels of the three-dimensional digital representation 12 possess greyscale values derived during the conversion step 50. In reconstructing the surface the greyscale values of voxels on the surface can be used to determine interpolation points on which interpolated surface will lie. In one embodiment, the erosion step may comprise removing one or more voxel layers and assigning to the outer voxel layer of the shrunken digitised representation the same greyscale value as the original outer layer. Alternatively, the erosion step may comprise removing zero or more voxels from the surface and decreasing the greyscale value of one or more of the voxels in the outer voxel layer of the shrunken digitised representation by a certain (e.g. predetermined or user-selected) percentage which will move the surface by a corresponding proportion of a voxel side length. This may be particularly advantageous

where a specific shell thickness is desired which is not an integer multiple of the sampling rate employed in the voxelization. For example, eroding to sub-voxel resolution may enable a 2.5 mm thickness shell to be achieved for a model sampled at 1 mm. which could only otherwise be achieved by resampling the model, e.g. at 0.5 mm.

The sampling rate selected for creating the digitised representation 12 of the original CAD model 10 affects the relative size of voxels in the final image, and can therefore affect the detail with which features on the ultimately created internal surface can be represented. Where a microstructure is to be introduced, it is desirable for the eroded digital representation 14 to have the same resolution as a digitised representation of that microstructure. This may be achieved by selecting a sampling rate for the conversion step 50 that is equal to the sampling rate used to create the digitised representation of the microstructure. However, in practice since it is likely that the microstructure will require a resolution much higher than that which is suitable (or sufficient) for digitising (voxelizing) the original CAD model 10, an alternative approach is to perform the conversion step 50 at a lower sampling rate (resolution) and then resample the digital representation 12 to match the resolution of the digitised representation of the microstructure. This alternative approach may reduce the computational burden of computing the distance function required as part of the conversion step 50.

In this embodiment, the resolution at which the CAD is digitised is determined by multiplying the average side length of the original CAD model's bounding box by a scale factor, which may be preset or selectable by a user, e.g. from a range bounded by preset upper and lower limits.

Where sub-voxel resolution erosion techniques are employed, the shell thickness need not necessarily be a multiple of the sampling rate. For example, a volume with a sampling rate of 1 mm could be effectively eroded by 0.3 mm.

The result of the second step is thus an eroded image 14 that can be used as a three-dimensional representation (e.g. voxelized image) of an internal volume (i.e. region to be hollowed out) of the original CAD model 10.

A third step 54 of the method may comprise modifying the eroded image 14 by merging it with a digitised representation of an appropriate microstructure 16. In this embodiment, the third step 54 comprises subtracting the microstructure 16 from the eroded image 14 to yield a representation that is effectively the digital negative of the microstructure 18 occupying the internal volume, i.e. effectively an inverted microstructure in the internal volume.

A fourth step 56 generates a vector-based surface representation 20 of the inverted microstructure. This step may use any of a number of techniques such as the marching cubes approach. The vector-based surface representation 20 may comprise a triangulated surface, which effectively describes the new surfaces/boundaries of the interior volume to be introduced to the original CAD model 10. The vector-based surface representation 20 (shown graphically in Fig. 1) may be in the STL format.

In a fifth step 57, the outward facing surface of the vector-based surface representation 20 is designated as an inward facing surface. This may be achieved by flipping the normals of the vector-based surface representation 20, which also has the effect of reversing the void/solid volumes, and hence introducing the microstructure. Where the marching cubes approach is used, the surface normal direction is defined by the order in which the triangle's vertices are specified; the direction is flipped by reversing this order for each triangle.

In a sixth step 58, the vector-based surface representation 20 of the interior surface is merged/integrated with the original exterior surface representation of the CAD model 10 to form robustly a hollowed out (shelled) model 22 of prescribed thickness having a microstructured interior. Fig. 1 shows a cut-away view of the hollowed out model 22, where the internal microstructure obtained by concatenating the inverted vector-based surface representation 20 with the original CAD model 10.

Fig. 2 is a flow chart summarising the steps of another method that is an embodiment of the invention. The method is similar to that described above, for creating a hollowed out version of an original CAD model. In this embodiment, the

hollowed out CAD model is displayed and/or used by a three-dimensional printer to make a physical model.

The method comprises a first step 60 of importing an original three-dimensional CAD model. The step of importing may act to convert the CAD model into STL format. In a second step 62, the CAD model is voxelized to create a digitised (e.g. bitmapped) representation thereof, which may be referred to as a voxel model. This step may correspond to the first step 50 described above.

In a third step 64, the voxel model is eroded by removing n voxels (or parts of voxels with greyscale erosion) from each point on its surface. This step may correspond to the second step 52 described above.

In a fourth step 66, a digitised representation of a microstructure is subtracted from the eroded voxel model to create of the volume to be removed from the interior of the original CAD model. This step may correspond to the third step 54 described above.

In a fifth step 68, a vector-based surface representation (in STL format) of the intersected eroded voxel model is generated, e.g. using a conventional marching cubes technique. This step may correspond to the fourth step 56 described above.

In a sixth step 70, the surface normals of the STL version of the intersected eroded voxel model are flipped such that its external surfaces are defined as internal surfaces. This step may correspond to the fifth step 57 described above.

In a seventh step 72, the intersected eroded voxel model with flipped surface normal is combined with the STL version of the original CAD model to create the hollowed out model. This step may correspond to the sixth step 58 described above. The combining may comprise a Boolean operation to unify the STL data sets. Since the internal volume was created from an eroded version of the original CAD model, there will be no intersecting surfaces, which simplifies processing.

The combined STL version of the hollowed out model may be output for external use. For example, in step 74 the combined STL version of the hollowed out model may be processed by a conventional graphics program and displayed, e.g. on a monitor or the like. Alternatively or additionally, the combined STL version of the hollowed out model may be outputted to a three-

dimensional printer, which may be capable of deriving quasi-two-dimensional slices thereof to create a physical model corresponding to the hollowed out model by sequentially depositing layers of material corresponding to the derived slices.

Fig. 3 illustrates a method that is another embodiment of the invention. This embodiment is similar to the method described in Fig. 1, but avoids the step of flipping the normals of the vector-based surface representation of the eroded voxel model by inverting the digitised eroded voxel model.

Similarly to the method discussed with reference to Fig. 1, the method of this embodiment starts from a vector-based representation of a three-dimensional model, which in this example is a perspective view of a CAD model 210. A first step 250 in the method comprises converting the CAD model 210 into a three-dimensional image or digital representation 212. This can be carried out by sampling the volume enclosed by the CAD model 210, as discussed above.

A second step 252 of the method comprises eroding the three-dimensional digital representation 212 by a prescribed amount to create an eroded digital representation 214 (e.g. voxelized image) which is effectively shrunk in size from the original. This eroding step may be carried out using standard image processing techniques discussed above.

A third step 253 of the method comprises inverting the eroded image 214 to generate an inverted eroded model 215, e.g. by flipping the bit values or performing a suitable greyscale inversion. The inverted eroded model 215 constitutes a void in a solid universe.

A fourth step 255 of the method may comprises modifying the inverted eroded model 215 by merging it with a digitised representation of an appropriate microstructure 216 (represented in Fig. 3 as a lattice volume). In this embodiment, the fourth step 255 comprises performing a union function to add the microstructure 216 to the inverted eroded model 215 to yield a representation 217 that is effectively a small volume of microstructure within a solid space.

A fifth step 256 generates a vector-based surface representation 219 of the inverted eroded model with

microstructure 217, e.g. using any of the techniques contemplated above.

In a sixth step 258, the vector-based surface representation 219 of the inverted eroded model with microstructure is merged/re-integrated with the original exterior surface representation of the CAD model 210 to form robustly a hollowed out (shelled) model 222 of prescribed thickness having a microstructured interior. Similarly to Fig. 1, Fig. 3 shows a cut-away view of the hollowed out model 222, which shows the internal microstructure obtained by concatenating the vector-based surface representation 219 of the inverted eroded model with microstructure with the original CAD model 210.

Fig. 4 is a flow chart summarising the steps of a method 15 that is another embodiment of the invention. The method is similar to the method described above with reference to Fig. 3, for creating a hollowed out version of an original CAD model. In this embodiment, the hollowed out CAD model is displayed and/or used by a three-dimensional printer to make a 20 physical model.

The method comprises a first step 260 of importing an original three-dimensional CAD model, a second step 262 of voxelizing the CAD model to create a digitised (e.g. bitmapped) representation thereof, which may be referred to as 25 a voxel model, and a third step 264 of eroding the voxel model by removing n voxels (or parts of voxels with greyscale erosion) from each point on its surface. These steps may correspond to the first step 60, second step 62 and third step 64 of Fig. 2.

In a fourth step 265, the eroded voxel model is inverted, i.e. all the voxel bits values are inverted. This step can be envisaged as swapping a solid representation of the eroded model in space for a solid space in which eroded model appears as a isolated hollow. The purpose of this inversion is that 35 the inverted eroded model is perceived as an inner surface of the solid space rather than an outer surface of a solid volume.

In a fifth step 267, a digitised representation of a microstructure is combined with the inverted eroded voxel model (e.g. using a union function). This step causes the 40

microstructure to appear within the isolated hollow that is the inverted eroded model.

In a sixth step 269, a vector-based surface representation (in STL format) of the inverted eroded voxel model with microstructure is generated, e.g. using a conventional marching cubes technique. This step effectively defines the surfaces of the inverted eroded model as internal surfaces of the solid space.

In a seventh step 272, the STL format surface representation generated in the sixth step is simply combined (e.g. concatenated) with the STL version of the original CAD model to create the hollowed out model, which may be displayed (step 274) or output to a three-dimensional printer (step 276). These steps may correspond to steps 72, 74 and 76 of Fig. 2.

Fig. 5 shows a processing environment 100, e.g. a personal computer or the like, that can be used to implement steps of the invention.

The processing environment 100 includes a processing unit (CPU) 102 in communication with a memory 108 via a data communication bus 104. The processing environment 100 includes a power supply 106, which may be mains power or a portable source of energy, e.g. a rechargeable or non-rechargeable battery. The processing environment 100 also includes one or more network interfaces 110, a display 112, and a keyboard 114 and input/output interface 116 which may function as a user input unit.

The network interface 110 is arranged to connect the processing environment 100 to one or more networks, and may be constructed for use with any suitable communication protocol. The display 112 may be a liquid crystal display (LCD) or any other type of display that can be used with a computing device. The display 112 may also form part of the user input unit by comprising a touch sensitive screen.

The input/output interface 116 can be arranged to communicate with other external devices using any suitable communication technology.

The memory 108 in the processing environment 100 may comprise a RAM 118 and a ROM 120. The memory 108 is arranged to store information such as computer executable instructions, data structures and the like. The ROM 120 may store a basic

input/output (BIOS) 122 for controlling low-level operation of the processing environment 100. The RAM 118 may store an operating system 124 for controlling the general operation of the processing environment 100. Conventional computer
5 operating systems may be used.

The RAM 118 may also store applications 128 which run on the operating system. One such application may be the image processing method discussed above. The RAM 118 may also include a general data storage region 126 which is arranged to
10 store information for use with the applications 128.

In one embodiment, the invention may provide a computer program product comprising a computer-readable storage medium having software instructions stored thereon, the software instructions being executable in the processing environment
15 100 to carry out one or more steps of the invention set out above. The software instructions may be stored in the memory 108, e.g. as an application in the RAM 118. The original CAD model may be imported from a remote network via the network interface 110 or may be input directly by a user using the
20 keyboard 114 and/or other input devices (e.g. a mouse or the like). The result of the method may be displayed on the display 112 or output to a three-dimensional printer either via the network (through network interface 110) or through a suitably configured output port (e.g. part of the input/output
25 interface 116).

Fig. 6 shows a schematic representation of an apparatus suitable for implementing the invention. The apparatus comprises a three-dimensional printer 130 in communication via a link 132 to a computer 134. The link 132 may be wired or
30 wireless, and may be direct or via a network and/or the internet. The computer is suitable programmed to perform the method steps described above and forward data representative of the hollowed out model to the three-dimensional printer for printing.

35 The embodiments above relate to examples where the original model has a vector-based surface representation. In other examples, the original model may already be in a digitised format. For example, the original model may be obtained by scanning a real (physical) model using any of a wide range of 3d imaging modalities such as MRI (Magnetic Resonance Imaging), CT (Computed Tomography) and Ultrasound.
40

In this case, the hollowed out model may be obtained by removing the digitised internal volume from the original digitised model. If a vector-based surface representation (e.g. in STL format) is required for subsequent processing, 5 the marching cubes technique may be used to generate it from the digitised hollowed out model.

Alternatively or additionally, the hollowed out digitised model may be used to perform computer-based analysis. For example, it may be segmented using conventional techniques and 10 exported to a finite element analysis package, such as +ScanFE manufactured by Simpleware Limited.

The microstructure in the internal may be adapted to act as a support structure, e.g. a internal structure having a plurality of support elements extending across the internal 15 volume to provide structural rigidity in one or more directions. For example, the microstructure may be biased in one direction (e.g. the direction of principal loading) to provide support.

20 REFERENCE

[1] Gabbrielli, R., Turner, I. G., and Bowen, C. R.:
Development of Modelling Methods for Materials to be Used as
Bone Substitutes, Key Engineering Materials 361-363, volume
25 361-363, 903-906, 2008

CLAIMS

1. A method of preparing a virtual three-dimensional object for three-dimensional printing, the method comprising:
 - 5 obtaining a vector-based surface representation of a virtual three-dimensional object;
 - sampling the volume enclosed by the three-dimensional object to generate a digitised three-dimensional representation of the object;
 - 10 eroding the digitised three-dimensional representation of the object to generate an internal volume;
 - generating a vector-based surface representation of the internal volume, wherein the surfaces of the internal volume represent internal surfaces; and
 - 15 combining the vector-based surface representation of the virtual three-dimensional object with the vector-based surface representation of the internal volume.
2. A method according to claim 1 including inverting the digitised representation of the internal volume before generating the vector-based surface representation of the internal volume, whereby the surfaces of the internal volume automatically represent internal surface upon generating the vector-based surface representation of the internal volume.
- 25 3. A method according to claim 1 including designating an exterior surface defined by the vector-based surface representation of the internal volume as an internal surface before combining it with the vector-based surface representation of the virtual three-dimensional object.
- 35 4. A method according to claim 3, wherein designating an exterior surface defined by the vector-based surface representation of the internal volume as an internal surface includes reversing the direction of a plurality of surface normals associated with the exterior surface.
5. A method according to claim 1 including applying a microstructure to the internal volume.

6. A method according to claim 2 including performing a union operation to combine a digitised representation of a microstructure with the inverted digitised internal volume, wherein the generated internal volume consists of the
5 microstructure.

7. A method according to claim 3 including, after
eroding the digitised three-dimensional representation of the
object, subtracting a digitised representation of a
10 microstructure from the eroded digitised three-dimensional
representation of the object, wherein the generated internal
volume consists of the microstructure.

8. A method according to claim 1, wherein the digitised
15 three-dimensional representation of the object comprises a
voxelized model of the virtual three-dimensional object, and
wherein eroding the digitised three-dimensional representation
of the object comprises removing voxels at the surface of the
voxelized model.
20

9. A method according to claim 8, wherein eroding the
digitised three-dimensional representation of the object
comprises removing a layer of voxels from the whole surface of
the voxelized model.
25

10. A method according to claim 9, wherein the removed
layer of voxels has a variable depth.

11. A method according to claim 10, including
30 determining the variable depth by way of single or multiple
(iterative) computer-based analysis.

12. A method according to claim 1, wherein combining the
vector-based surface representations of the virtual three-
35 dimensional object and the internal volume comprising
performing a concatenation operation to unify sets of data
corresponding to the vector-based surface representations of
the original virtual three-dimensional object and the internal
volume respectively.
40

13. A method according to claim 12, wherein the sets of data are in STL format.

5 14. A method according to claim 5, wherein a digitised representation of the microstructure is generated from tri-periodic mathematical functions.

10 15. A method according to claim 14 including modifying the tri-periodic mathematical functions to introduce density variations inside the internal volume.

15 16. A method according to claim 5, wherein a digitised representation of the microstructure is generated by sampling a virtual three-dimensional vector-based model of a microstructure or by scanning a real physical object.

20 17. A computer program product comprising a computer-readable storage medium having software instructions stored thereon, the software instructions being executable by a computer to perform the steps of a method according to claim 1.

25 18. A method of generating a physical model using three-dimensional printing, the method comprising:

 inputting a vector-based surface representation of a virtual three-dimensional object;

 sampling the volume enclosed by the three-dimensional object to generate a digitised three-dimensional representation of the object;

30 30 eroding the digitised three-dimensional representation of the object to generate an internal volume;

 generating a vector-based surface representation of the internal volume, wherein the surfaces of the internal volume represent internal surfaces; and

35 35 combining the vector-based surface representation of the virtual three-dimensional object with the vector-based surface representation of the internal volume.

 outputting the combined vector-based surface representation to a three-dimensional printer; and

operating the three-dimensional printer to create a physical model corresponding to the combined vector-based surface representation.

5 19. A method of preparing a virtual three-dimensional object for computer-based analysis, the method comprising:
obtaining a vector-based surface representation of a virtual three-dimensional object;
sampling the volume enclosed by the three-dimensional
10 object to generate a digitised three-dimensional representation of the object;
eroding the digitised three-dimensional representation of the object to generate an internal volume;
15 combining the digitised three-dimensional representation of the object with digitised three-dimensional representation of the internal volume; and
performing computer-based analysis on the combined digitised three-dimensional representation.

20 20. A method according to claim 19 including, after eroding the digitised three-dimensional representation of the object, subtracting a digitised representation of a microstructure from the eroded digitised three-dimensional representation of the object, wherein the generated internal volume consists of the microstructure.
25

30 21. A method according to claim 19 including, after eroding the digitised three-dimensional representation of the object, inverting the digitised representation of the internal volume and performing a union operation to combine a digitised representation of a microstructure with the inverted digitised internal volume, wherein the generated internal volume consists of the microstructure.

35 22. A method according to claim 19, wherein the computer-based analysis includes finite element and/or finite volume analysis.

40 23. A method of preparing a virtual three-dimensional object for three-dimensional printing, the method comprising:

scanning a physical three-dimensional object to obtain a digitised three-dimensional representation of the object;

eroding the digitised three-dimensional representation of the object to generate an internal volume;

5 subtracting a digitised representation of a microstructure from the eroded digitised three-dimensional representation of the object, wherein the generated internal volume consists of an inverted version of the microstructure; and

10 subtracting the generated internal volume from the digitised three-dimensional representation of the object to create a hollowed out representation of the object.

24. A method of preparing a virtual three-dimensional object for three-dimensional printing, the method comprising:

15 scanning a physical three-dimensional object to obtain a digitised three-dimensional representation of the object;

eroding the digitised three-dimensional representation of the object to generate an internal volume;

20 inverting the digitised representation of the internal volume;

performing a union operation to combine a digitised representation of a microstructure with the inverted digitised internal volume, wherein the generated internal volume

25 consists of the microstructure; and

combining the digitised three-dimensional representation of the object with digitised three-dimensional representation of the internal volume to create a hollowed out representation of the object.

30 25. A method according to claim 23 including generating a vector-based surface representation of the hollowed-out representation of the object.

35 26. Apparatus for preparing a virtual three-dimensional object for three-dimensional printing, the apparatus comprising:

an input device for receiving a vector-based surface representation of a virtual three-dimensional object;

40 a memory arranged to store the vector-based surface representation of a virtual three-dimensional object;

a processor programmed to:

sample the volume enclosed by the three-dimensional object to generate a digitised three-dimensional representation of the object,

5 erode the digitised three-dimensional representation of the object to generate an internal volume,

generate a vector-based surface representation of the internal volume, wherein the surfaces of the internal volume represent internal surfaces, and

10 combine the vector-based surface representation of the virtual three-dimensional object with the vector-based surface representation of the internal volume; and

an output device for outputting the combined vector-based surface representation of the object and internal volume.

15

27. Apparatus according to claim 26 including a display and/or a three-dimensional printer connected to the output device.

20

28. Apparatus according to claim 26, wherein the memory has a library comprising a plurality of digitised representations of microstructures stored thereon, and the processor is further programmed to:

25 permit selection via the input device of a microstructure in the library; and

apply the digitised representation of the selected microstructure to the eroded digitised three-dimensional representation of the object,

30 wherein the generated internal volume consists of the microstructure.

ABSTRACT

A method for preparing image data for three-dimensional printing in which a digitised (e.g. voxelized) representation of a virtual three-dimensional object (e.g. CAD model) is
5 eroded to create an internal volume for the object.

Subsequently, a vector-based surface representation of this internal volume is generated and simply combined with a corresponding vector-based surface representation of the original virtual three-dimensional object to yield a hollowed
10 out model in a format suitable for three-dimensional printing.

A microstructure may be introduced into the interior of the hollowed out model, e.g. by extracting a volume corresponding to the inverse of that microstructure from the eroded digitised representation.

1/5

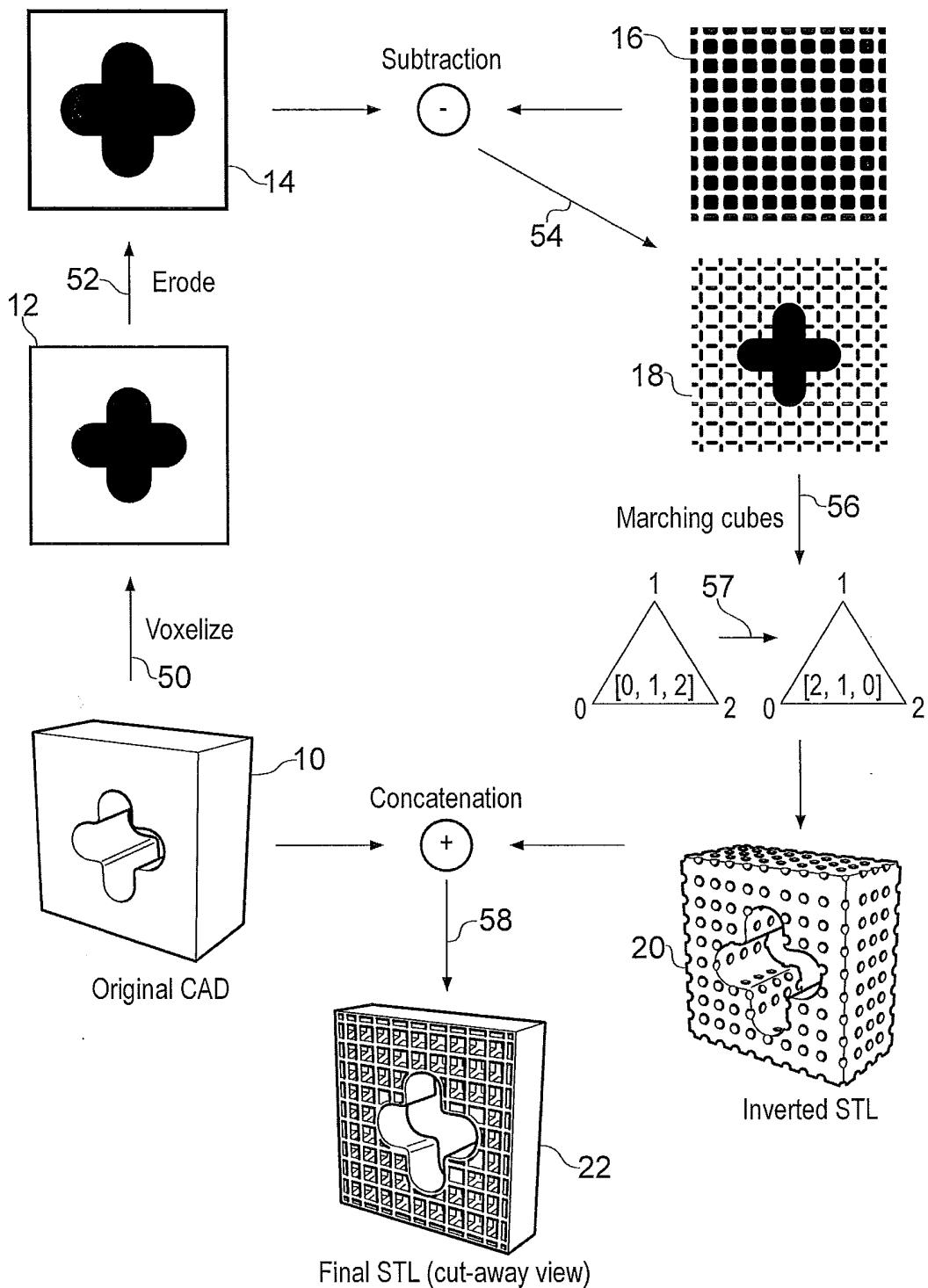


FIG. 1

2/5

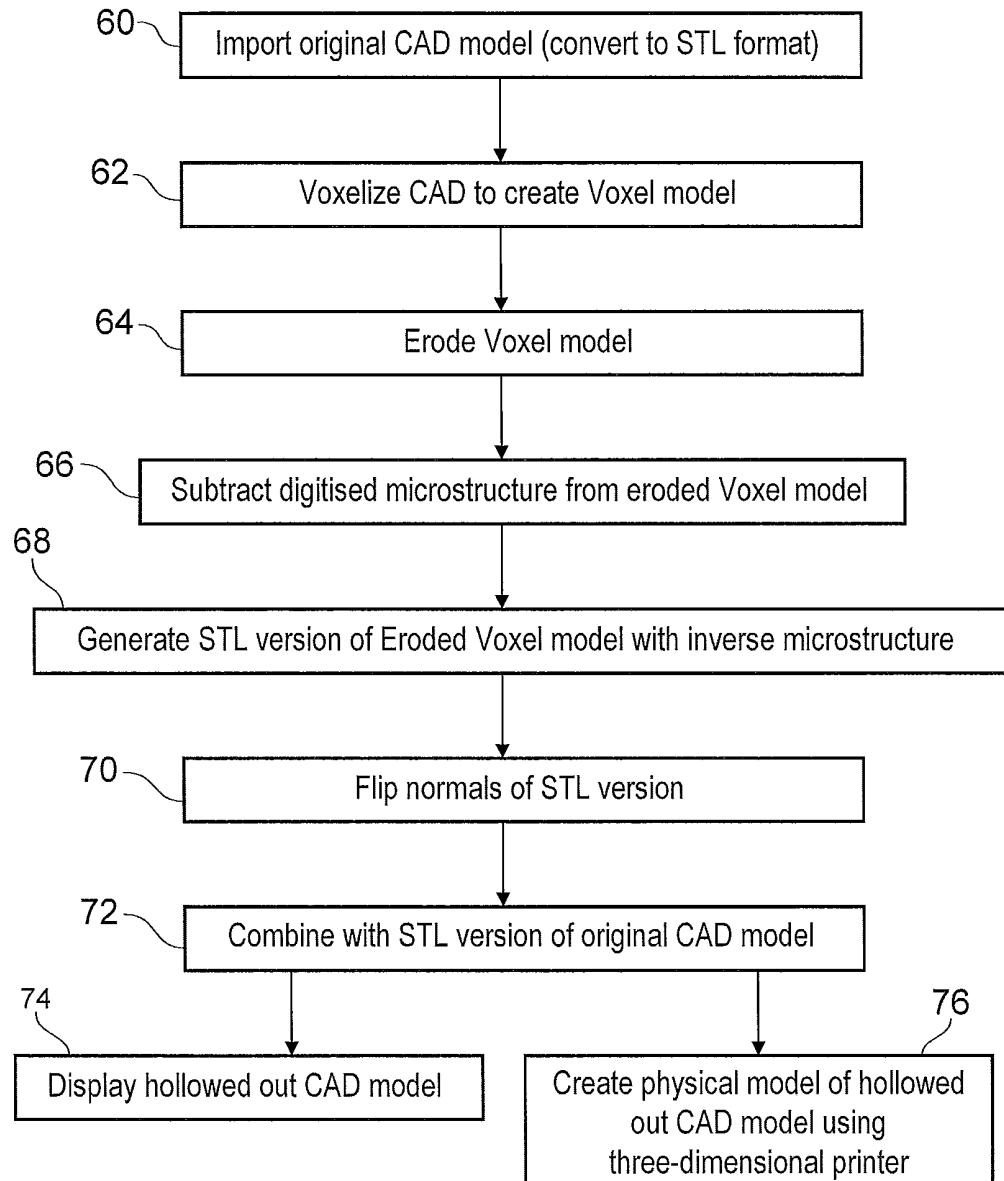


FIG. 2

3/5

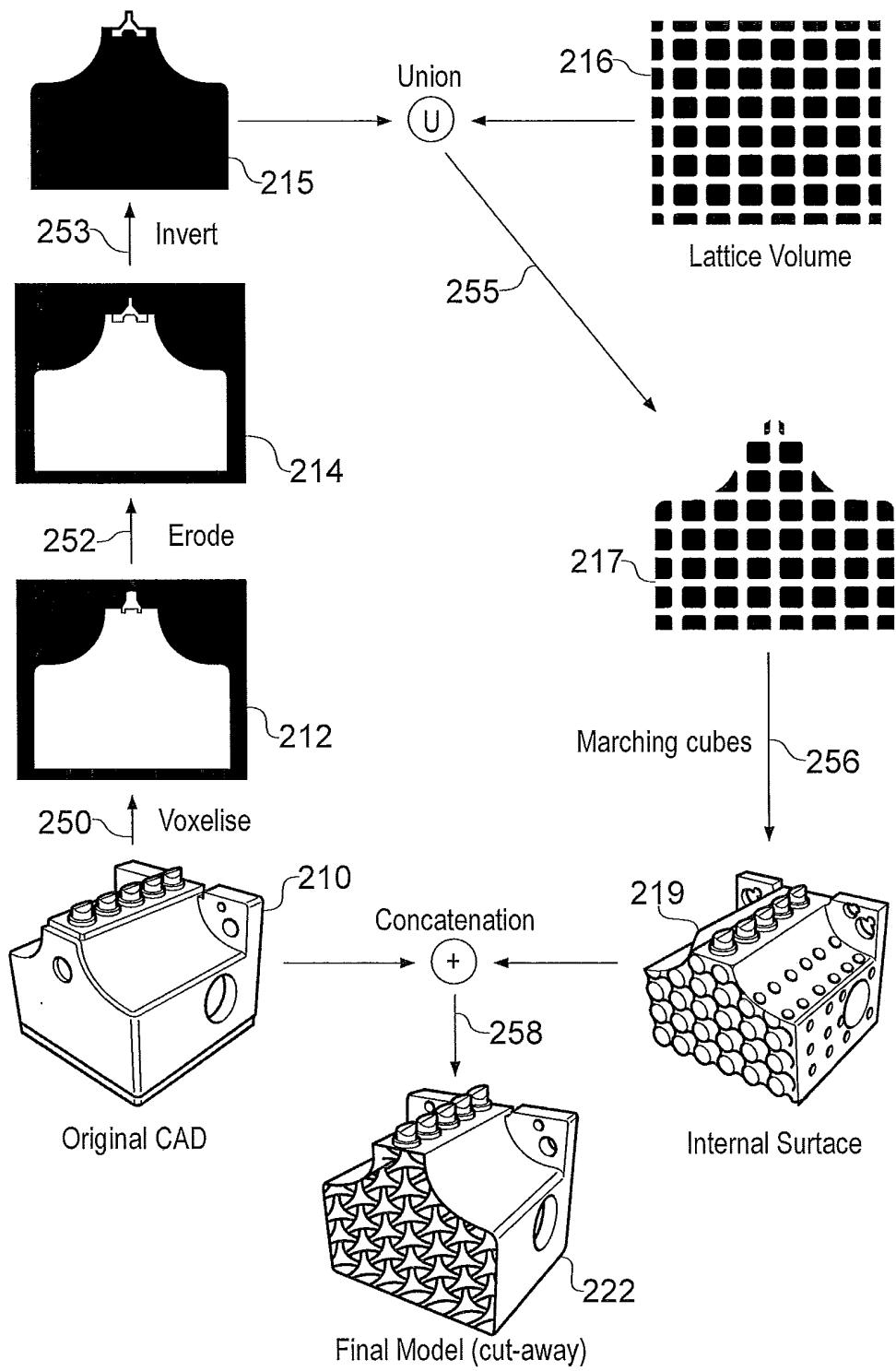


FIG. 3

4/5

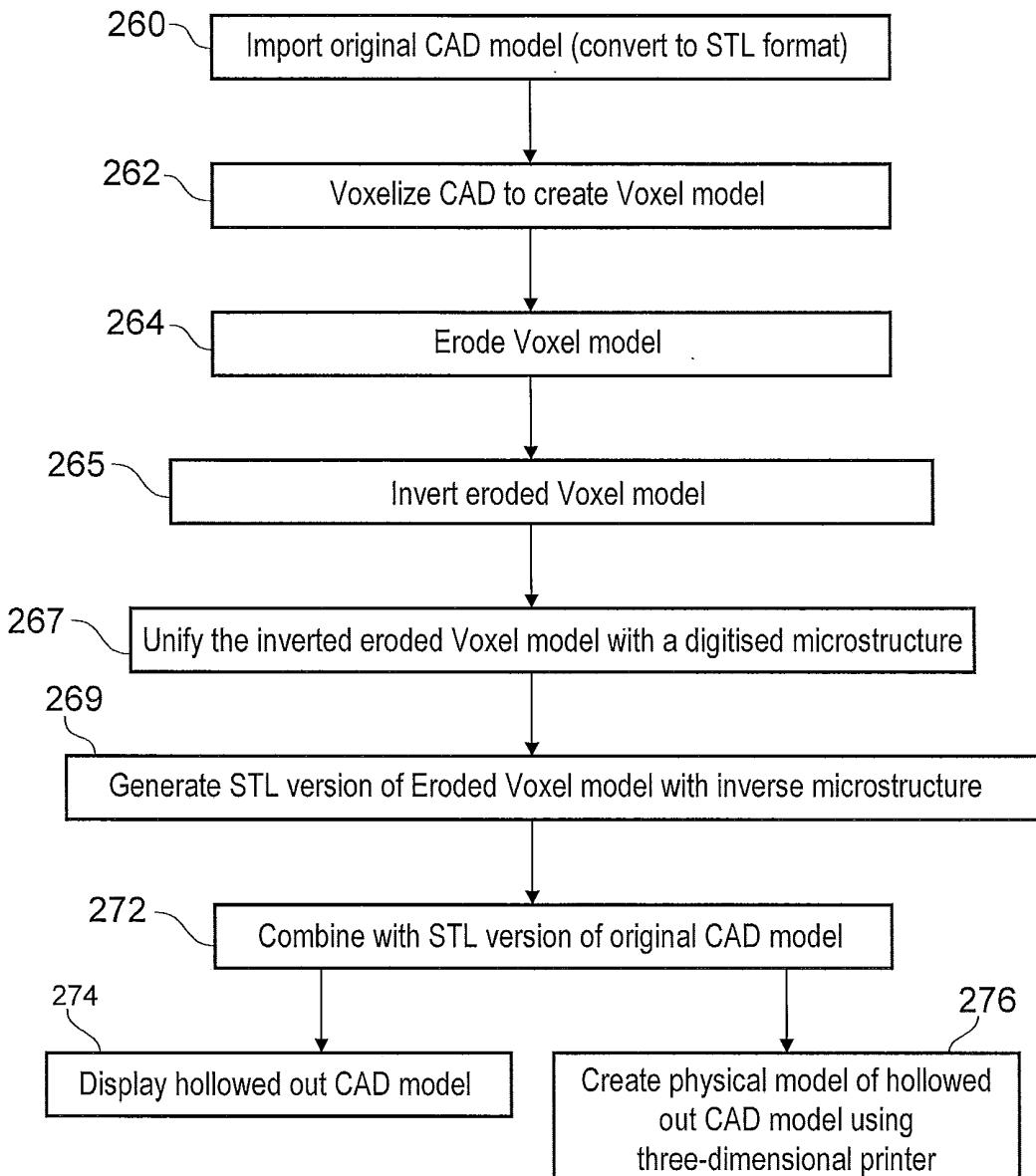
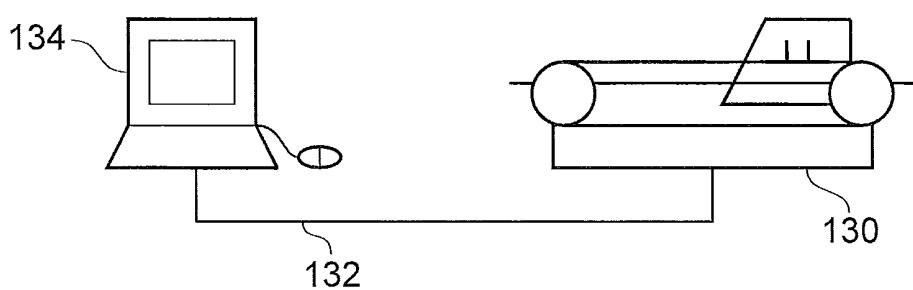
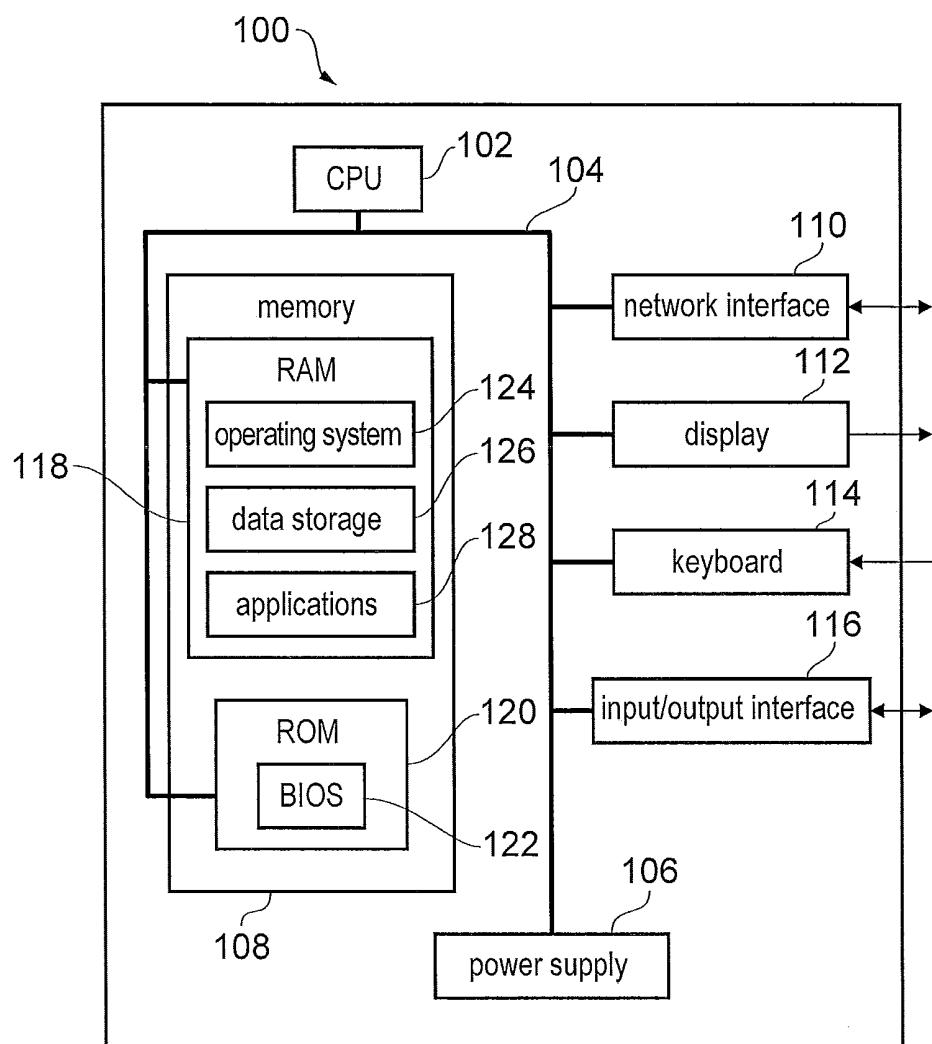


FIG. 4

5/5



Appendix C

Internal Structure Wizard

As implemented in ⁺ScanCAD.



Figure C.1: Introduction

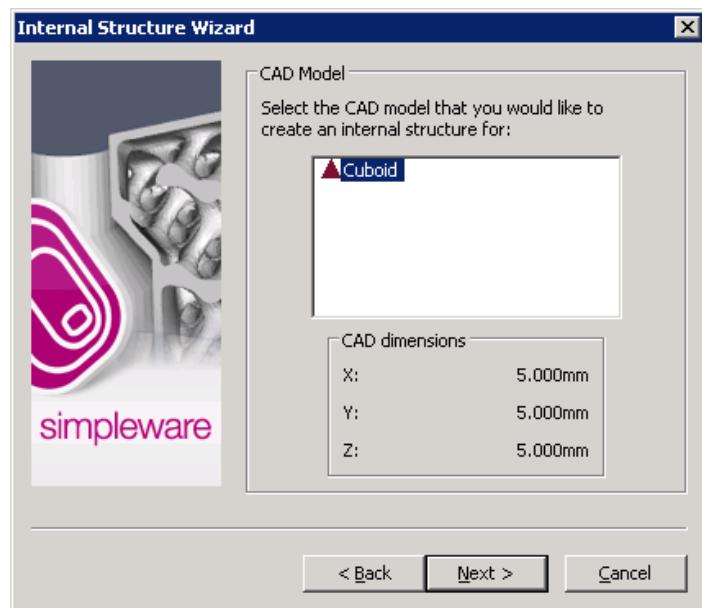


Figure C.2: Selection of CAD model



Figure C.3: Shell parameters

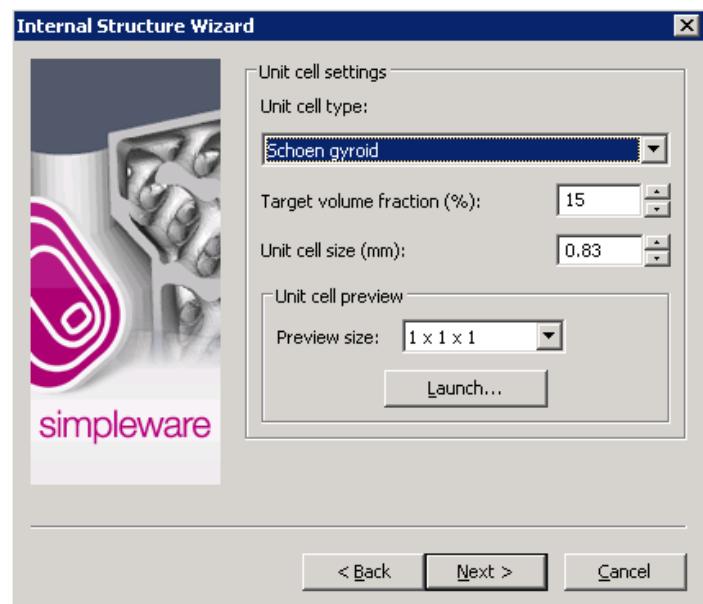


Figure C.4: Lattice parameters

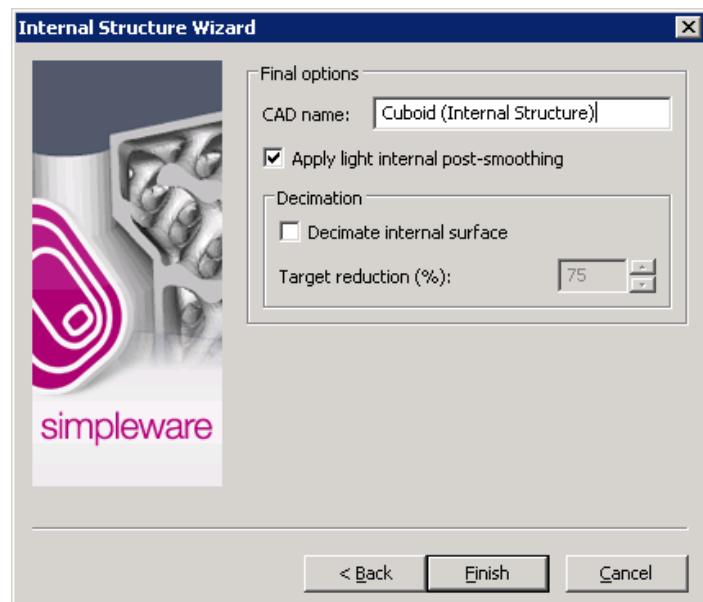


Figure C.5: Post-generation options

Appendix D

Internal Structures Testing – Experimental Setup

Both hollow and internal structure components are loaded in the same fashion.



Figure D.1: Loading of the box component. Not visible is a small metal disk 15 mm in diameter on top of the component.



Figure D.2: Loading of the sphere component



Figure D.3: Loading of the cylinder component

References

- (2009). Insight toolkit (ITK). <http://itk.org/>. 30
- (2009). Netfabb. <http://www.netfabb.com>. 70
- (2010). CalculiX. <http://www.dhondt.de/>. 150
- (2010). Finite element package. <http://www.maplesoft.com/applications/view.aspx?SID=4671>. 153
- (2010). K3dSurf. <http://k3dsurf.sourceforge.net/>. 26
- (2010). The scientific graphics project. <http://www.msri.org/about/sgp/SGP/index.html>. 22
- (2011). Maple. <http://www.maplesoft.com>. 138
- (2011). Metals and alloys. <http://www.msm.cam.ac.uk/phase-trans/abstracts/annealing.twin.html>. 8
- (2011). *ScanIP User's Manual*. Simpleware. 44
- (2011). Zimmer. <http://www.zimmer.com/ctl?template=CP&op=global&action=1&id=33>. 5

REFERENCES

- ADACHI, T., OSAKO, Y., TANAKA, M., HOJO, M. & HOLLISTER, S.J. (2006). Framework for optimal design of porous scaffold microstructure by computational simulation of bone regeneration. *Biomaterials*, **27**, 3964–3972. 15
- ANTON, H. & RORRES, C. (2005). *Elementary Linear Algebra with Applications*. Wiley. 142
- BENDSØE, M. (1989). Optimal shape design as a material distribution problem. *Structural and multidisciplinary optimization*, **1**, 193–202. 14
- BENDSØE, M. & KIKUCHI, N. (1988). Generating optimal topologies in optimal design using a homogenization method. *Comp. Meth. Appl. Mech. Engrg*, **71**, 197–224. 201
- BENDSØE, M. & SIGMUND, O. (2003). *Topology optimization: theory, methods, and applications*. Springer Verlag. 13
- BENVENISTE, Y. (1987). A new approach to the application of mori-tanaka's theory in composite materials. *Mechanics of Materials*, **6**, 147–157. 107
- BERAN, M. (1968). *Statistical Continuum Theories*. John Wiley, New York. 106
- BÙI XUÂN, V. (2008). *Development of a pre-processing environment for 3D image based finite element mesh generation*. Ph.D. thesis, School of Engineering, Computer Science and Mathematics, University of Exeter. 68
- CHEN, Y. (2006). A mesh-based geometric modeling method for general structures. *ASME Conference Proceedings*, **2006**, 269–281. 12
- CHEN, Y. (2007a). 3d texture mapping for rapid manufacturing. *Computer-Aided Design & Applications*, **4**, 761–771. 7, 13, 61, 66, 67, 78, 195

REFERENCES

- CHEN, Y. (2007b). An accurate sampling-based method for approximating geometry. *Computer-Aided Design*, 975–986. 12, 86
- DRUGAN, W.J. & WILLIS, J.R. (1996). A micromechanics-based nonlocal constitutive equation and estimates of representative volume element size for elastic composites. *Journal of the Mechanics and Physics of Solids*, 44, 497–524. 107
- DVORKIN, J., NUR, A. & YIN, H. (1994). Effective properties of cemented granular materials. *Mechanics of Materials*, 18, 351–366. 107
- EAVES, D. (2004). *Handbook of Plastic Foams*. Smithers Rapra Press. 5
- EVESQUE, P. (2000). Fluctuations, correlation and representative elementary volume (rev) in granular materials. *Poudres & Grains*, 11, 6–17. 108
- FELIPPA, C.A. (1986). *Introduction To Finite Element Methods*. Department of Aerospace Engineering Sciences and Center for Aerospace Structures, University of Colorado, Boulder, Colorado. 187, 188, 190
- FRITZEN, F. & BÖHLKE, T. (2010). Influence of the type of boundary conditions on the numerical properties of unit cell problems. *Technische Mechanik*, 30, 354–363. 109
- GABBRIELLI, R. (2009). *Foam geometry and structural design of porous material*. Ph.D. thesis, Department of Mechanical Engineering, University of Bath. 16, 62
- GABBRIELLI, R., TURNER, I.G. & BOWEN, C.R. (2008). Development of mod-

REFERENCES

- elling methods for materials to be used as bone substitutes. *Key Engineering Materials*, **361-363**, 903–906. 7, 10, 11, 16, 22, 26, 54, 61
- GITMAN, I., ASKES, H. & SLUYS, L. (2007). Representative volume: Existence and size determination. *Engineering Fracture Mechanics*, **74**, 2518–2534. 108
- GOSHTASBY, A. (1993). Design and recovery of 2-d and 3-d shapes using rational gaussian curves and surfaces. *International Journal of Computer Vision (IJCV)*, **10**, 233–256. 18
- GUEDES, J., RODRIGUES, H. & BENDSØE, M. (2003). A material optimization model to approximate energy bounds for cellular materials under multiload conditions. *Structural and multidisciplinary optimization*, **25**, 446–452. 15
- HASHIN, Z. & SHTRIKMAN, S. (1963). A variational approach to the theory of the elastic behaviour of multiphase materials. *Journal of the Mechanics and Physics of Solids*, **11**, 127–140. 106
- HAZANOV, S. & AMIEUR, M. (1995). On overall properties of elastic heterogeneous bodies smaller than the representative volume. *International Journal of Engineering Science*, **33**, 1289–1301. 110
- HEISE, U., OSBORN, J.F. & DUWE, F. (1990). Hydroxyapatite ceramic as a bone substitute. *International Orthopaedics*, **14**, 329–338, 10.1007/BF00178768. 6
- HERTZ, H. (1882). Uber die beruhrung fester elastischer korper. *Zeitschrift fuer die Reine und Angewandte Mathematik*, **92**, 156–171. 130

REFERENCES

- HEYMANS, P., MARTENS, L., AERNOUDT, M., DE MEESTER, E., MULIER, P. & DUCHEYNE, J. (1977). The mechanical behaviour of intracondylar cancellous bone of the femur at different loading rates. *Journal of Biomechanics*, **10**, 747–762. 135
- HILL, R. (1952). The elastic behaviour of a crystalline aggregate. *Proceedings of the Physical Society. Section A*, **65**, 349. 182
- HILL, R. (1963). Elastic properties of reinforced solids: Some theoretical principles. *Journal of the Mechanics and Physics of Solids*, **11**, 357–372. 108
- HILL, R. (1965). A self-consistent mechanics of composite materials. *Journal of the Mechanics and Physics of Solids*, **13**, 213–222. 107
- HOLLISTER, S., MADDOX, R. & TABOAS, J. (2002). Optimal design and fabrication of scaffolds to mimic tissue properties and satisfy biological constraints. *Biomaterials*, **23**, 4095–4103. 6, 7, 54
- HOLLISTER, S.J. (2005). Porous scaffold design for tissue engineering. *Nature Materials*, **4**, 518–524. 6, 15
- HOLLISTER, S.J. & LIN, C.Y. (2007). Computational design of tissue engineering scaffolds. *Comput. Methods Appl. Mech. Engrg.*, **196**, 2991–2998. 6, 15, 69
- HUET, C. (1990). Application of variational concepts to size effects in elastic heterogeneous bodies. *Journal of Mechanics Physics of Solids*, **38**, 813–841. 114

REFERENCES

- JAMISON, R., GOLDWASSER, M., GROSSER, B., SINN-HANLON, J. & CESARANO, J. (2010). Mandible reconstruction project. <http://www.itg.uiuc.edu/technology/reconstruction/>. 81
- JAYASINGHE, S. & EDIRISINGHE, M. (2002). A novel method of forming open cell ceramic foam. *Journal of Porous Materials*, **9**, 265–273, 10.1023/A:1021648812377. 6
- JIANG, M., ALZEBDEH, K., JASIUK, I. & OSTOJA-STARZEWSKI, M. (2001). Scale and boundary conditions effects in elastic properties of random composites. *Acta Mechanica*, **148**, 63–78, 10.1007/BF01183669. 110
- JU, T., LOSASSO, F., SCHAEFER, S. & WARREN, J. (2002). Dual contouring of hermite data. *ACM Transactions on Graphics (TOG)*, **21**, 339–346. 20
- JUNG, Y. & TORQUATO, S. (2005). Fluid permeabilities of triply periodic minimal surfaces. *Phys. Rev. E*, **72**, 056319. 11
- KANIT, T., FOREST, S., GALLIET, I., MOUNOURY, V. & JEULIN, D. (2003). Determination of the size of the representative volume element for random composites: statistical and numerical approach. *International Journal of Solids and Structures*, **40**, 3647–3679. 107, 108, 109, 114, 116
- KASSEM, G.A. (2009). *Micromechanical Material Models for Polymer Composites Through Advanced Numerical Simulation Techniques*. Ph.D. thesis, Rheinisch-Westfälische Technische Hochschule Aachen. 105
- LI, S. & WONGSTO, A. (2004). Unit cells for micromechanical analyses of particle-reinforced composites. *Mechanics of materials*, **36**, 543–572. 111

REFERENCES

- LIN, C. (2005). Ph.D. thesis, The University of Michigan. 15
- LIN, C.Y., KIKUCHI, N. & HOLLISTER, S.J. (2004). A novel method for biomaterial scaffold internal architecture design to match bone elastic properties with desired porosity. *Journal of Biomechanics*, **37**, 623–636. 15
- LORENSEN, W.E. & CLINE, H.E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 163–169, ACM, New York, NY, USA. 20, 24, 83
- LU, T., STONE, H. & ASHBY, M. (1998). Heat transfer in open-cell foams. *Acta Mater*, **46**, 3619–3635. 5
- MARMIER, A., LETHBRIDGE, Z.A., WALTON, R.I., SMITH, C.W., PARKER, S.C. & EVANS, K.E. (2010). Elam: A computer program for the analysis and representation of anisotropic elastic properties. *Computer Physics Communications*, **181**, 2102–2115. 9, 182, 183, 184, 185
- MC LAUGHLIN, R. (1977). A study of the differential scheme for composite materials. *International Journal of Engineering Science*, **15**, 237–244. 107
- MILLER, M.N. (1969). Bounds for effective electrical, thermal, and magnetic properties of heterogeneous materials. *Journal of Mathematical Physics*, **10**, 1988–2004. 106
- MILTON, G. (1982). Bounds on the elastic and transport properties of two-component composites. *Journal of the Mechanics and Physics of Solids*, **30**, 177–191. 106

REFERENCES

- MORI, T. & TANAKA, K. (1973). Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta Metallurgica*, **21**, 571–574.
- 107
- MÜLLER, R. & RÜEGSEGGER, P. (1995). Three-dimensional finite element modelling of non-invasively assessed trabecular bone structures. *Medical engineering & physics*, **17**, 126–133. 83
- NEWMAN, T.S. & YI, H. (2006). A survey of the marching cubes algorithm. *Computers & Graphics*, **30**, 854–879. 24
- OSTOJA-STARZEWSKI, M. (2002). Microstructural randomness versus representative volume element in thermomechanics. *Journal of Applied Mechanics*, **69**, 25–35. 108
- OSTOJA-STARZEWSKI, M. (2006). Material spatial randomness: From statistical to representative volume element. *Probabilistic Engineering Mechanics*, **21**, 112–132. 113
- PAHR, D. & ZYSSET, P. (2008). Influence of boundary conditions on computed apparent elastic properties of cancellous bone. *Biomechanics and Modeling in Mechanobiology*, **7**, 463–476, 10.1007/s10237-007-0109-7. 111, 112, 113
- PAHR, D.H. & RAMMERSTORFER, F.G. (2004). Experimental and numerical investigations of perforated cfr woven fabric laminates. *Composites Science and Technology*, **64**, 1403–1410. 113
- PAHR, D.H. & RAMMERSTORFER, F.G. (2006). Buckling of honeycomb sand-

REFERENCES

- wiches: Periodic finite element considerations. *Computer Modeling in Engineering*, **12**, 229–242. 113
- PASKO, A., VILBRANDT, T., FRYAZINOV, O. & ADZHIEV, V. (2010). Procedural function-based spatial microstructures. In *2010 Shape Modeling International Conference*, 47–56, IEEE. 7, 16
- PASKO, A., FRYAZINOV, O., VILBRANDT, T., FAYOLLE, P. & ADZHIEV, V. (2011). Procedural Function-based Modelling of Volumetric Microstructures. *Graphical Models*. 16
- PASKO, G., PASKO, A. & KUNII, T. (2004). Space-time blending. *Computer Animation and Virtual Worlds*, **15**, 109–121. 71
- QUEHEILLALT, D.T., KATSUMURA, Y. & WADLEY, H.N. (2004). Synthesis of stochastic open cell ni-based foams. *Scripta Materialia*, **50**, 313–317. 5
- REISSNER, E. (1947). Stresses and small displacements of shallow spherical shells. *Journal of Mathematics and Physics*, **25**, 279–300. 130
- REUSS, A. (1929). Berechnung der fließgrenze von mischkristallen auf grund der plastizitätsbedingung für einkristalle. *Zeitschrift für Angewandte Mathematik und Mechanik*, **9**, 49–58. 104
- REUSS, A. & ANGEW, Z. (1929). Calculation of the flow limits of mixed crystals on the basis of the plasticity of mono-crystals. *Math. Mech*, **9**, 49. 182
- RHO, J.Y., HOBATHO, M.C. & ASHMAN, R.B. (1995). Relations of mechanical properties to density and ct numbers in human bone. *Medical Engineering & Physics*, **17**, 347–355. 135

REFERENCES

- RODRIGUES, H., GUEDES, J. & BENDSØE, M. (2002). Hierarchical optimization of material and structure. *Struct. Multidisc. Optim.*, **24**, 1–10. 14, 15
- SAB, K. (1992). On the homogenization and the simulation of random materials. *European journal of mechanics. A. Solids*, **11**, 585–607. 114
- SANZ-HERRERA, J., GARCÍA-AZNAR, J. & DOBLARÉ, M. (2009). On scaffold designing for bone regeneration: A computational multiscale approach. *Acta Biomaterialia*, **5**, 219–229. 15
- SCHOEN, A.H. (1970). Infinite periodic minimal surfaces without self-intersection. Tech. rep., National Aeronautics and Space Administration, Electronics Research Center, Cambridge, Mass. 02139. 16, 21, 22
- SCHWARZ, H. (1890). *Gesammelte mathematische abhandlungen*. Springer-Verlag, Berlin. 21, 22
- SIGMUND, O. (1994). Materials with prescribed constitutive parameters: an inverse homogenization problem. *International Journal of Solids and Structures*, **31**, 2313–2329. 15
- STARLY, B. (2006). *Biomimetic Design And Fabrication Of Tissue Engineered Scaffolds Using Computer Aided Tissue Engineering*. Ph.D. thesis, Department of Mechanical Engineering and Mechanics, Drexel University. 6, 13, 69
- STARLY, B., LAU, A., SUN, W., LAU, W. & BRADBURY, T. (2005). Direct slicing of step based nurbs models for layered manufacturing. *Computer-Aided Design*, **37**, 387–397. 19

REFERENCES

- STEVEN, G., LI, Q. & XIE, Y. (2000). Evolutionary topology and shape design for general physical field problems. *Computational mechanics*, **26**, 129–139. 15
- THOMAS, D. (2009). *The Development of Design Rules for Selective Laser Melting*. Ph.D. thesis, University of Wales Institute, Cardiff. 118
- TORQUATO, S. (1991). Random heterogeneous media: Microstructure and improved bounds on effective properties. *Applied Mechanics Reviews*, **44**, 37–76. 106
- TORRES-SANCHEZ, C. & CORNEY, J.R. (2009). Toward functionally graded cellular microstructures. *Journal of Mechanical Design*, **131**, 091011. 60
- TREECE, G., PRAGER, R. & GEE, A. (1999). Regularised marching tetrahedra: improved iso-surface extraction. *Computers & Graphics*, **23**, 583–598. 20
- VOIGT, W. (1910). *Lehrbuch der kristallphysik*. B. G. Teubner, Leipzig. 104, 182
- WANG, H., CHEN, Y. & ROSEN, D.W. (2005). A hybrid geometric modeling method for large scale conformal cellular structures. *ASME Conference Proceedings*, **2005**, 421–427. 6, 12
- XU, F., LOH, H. & WONG, Y. (1999). Considerations and selection of optimal orientation for different rapid prototyping systems. *Rapid Prototyping Journal*, **5**, 54–60. 120
- YOUNG, P., BERESFORD-WEST, T., COWARD, S., NOTARBERARDINO, B., WALKER, B. & ABDUL-AZIZ, A. (2008). An efficient approach to converting three-dimensional image data into highly accurate computational models.

REFERENCES

- Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **366**, 3155–3173. 20, 83
- YOUNG, P.G. (2003). An analytical model to predict the response of fluid-filled shells to impact—a model for blunt head impacts. *Journal of Sound and Vibration*, **267**, 1107–1126. 130
- YOUNG, V., QUERIN, O., STEVEN, G. & XIE, Y. (1999). 3d and multiple load case bi-directional evolutionary structural optimization (beso). *Structural Optimization*, **18**, 183–192. 201
- ZHENG, B. (2008). *2D Curve And 3D Surface Representation Using Implicit Polynomial And Its Applications*. Ph.D. thesis, The Graduate School Of The University Of Tokyo. 17, 18, 19