Pattern transformation of three-dimensional periodic structures

T.W.J. de Geus

June 2011
MT 11.22

Internship at
Harvard University,
School of Engineering and Applied Sciences,
Cambridge Massachusetts, United States

Supervisors:
Dr. K. Bertoldi
Prof. dr. ir. M.G.D. Geers
**Notation**

\(a\)  
scalar

\(\vec{a}\)  
vector

\(|a|\)  
absolute value

\(\vec{a}\)  
a \(N \times 1\) column, in this case of scalars

\(\mathbf{A}\)  
a \(N \times N\) matrix, in this case of scalars

\(\mathbf{A}^T\)  
matrix transpose

\(\vec{a} \vec{b}\)  
dyadic product

\(\vec{a} \cdot \vec{b}\)  
inner- or dot-product

\(\vec{a} \times \vec{b}\)  
outer- or cross-product

\(\mathbf{A} = \vec{a} \vec{b}\)  
second or tensor

\(||\vec{a}|| = \sqrt{\vec{a} \cdot \vec{a}}\)  
Euclidean 2-norm, or length of vector \(\vec{a}\).

\(\dot{a}\)  
first order time-derivative

\(\ddot{a}\)  
second order time-derivative
# Contents

Chapter 1  **Introduction**  

Chapter 2  **Problem statement**  
2.1. Two-dimensional periodic array of holes  
2.2. Unit cell with periodic boundaries  
2.3. Three-dimensional periodic structures  

Chapter 3  **Modeling**  
3.1. General introduction to a buckling analysis  
3.2. Material behavior  
3.3. Macroscopic instability  
3.4. Stability analysis of infinite periodic structures by means of the buckling analysis  
3.5. Stability analysis of infinite periodic structures by means of the Bloch analysis  
3.6. Post-transformation analysis  

Chapter 4  **Obtaining a periodic mesh**  
4.1. Selection of a mesh algorithm  
4.2. DISTMESH: a simple mesh generator in MATLAB  
4.3. Method of obtaining a periodic mesh  
4.4. Influence of level-set function on the behavior of the algorithm  

Chapter 5  **Results for different geometries**  
5.1. Cylindrical voids  
5.2. Simple cubic lattice  
5.3. Face Centered Cubic lattice  
5.4. Body Centered Cubic lattice  
5.5. Other geometries  

Chapter 6  **Conclusion and recommendations**  

Chapter A  **Level-set functions of polygons**  
A.1. Vector calculus  
A.2. Distance to a rectangle  
A.3. Distance to a triangle  
A.4. Distance to a cube  
A.5. Distance to a tetrahedron  

Bibliography
Chapter 1

Introduction

Throughout history, the honeycomb has served as an inspiration of many field of research. More specifically, research directed to these cellular solids shows the favorable properties of these structures; motivating their use in many applications including mechanical, acoustic and thermal. Obviously, the benefits are centered around the intrinsic low density. On the other hand, it appears that these structures are capable of evidencing relatively high strength and stiffness. The mechanical behavior of such structures is reasonable well understood as it is summarized in the book of Gibson and Ashby [14].

When loaded beyond the initial elastic limit, these structures collapse benefiting from high energy absorption. In compression, it has been shown that this collapse commonly is the result of the buckling of member or wall of the structure. This then leads to localization of deformation in bands. Subsequent collapse is concentrated in the vicinity of these bands, which can then progress through the structure at relatively constant stress. Studies to these phenomena include [22, 23, 24, 34] for hexagonal shaped honeycombs and [25, 26, 27, 30, 31] for their circular counterparts.

Contrary to the localization observed in cellular structures, the mechanical instability can also induce homogeneous pattern transformation that occurs throughout the structure. The latter is observed in periodic elastomeric structures. These instabilities, that are the result of buckling of the ligaments under compressive loading, occur at relatively low strains and, more importantly, result from an elastic instability. The pattern transformation is therefore fully reversible and repeatable [3, 4], making these periodic structure attractive for a number of practical applications. Examples of these applications are pronounced for (sub)micron structures with a periodicity comparable to the acoustic wavelength or even the wavelength of visible light. The former category comprises the phononic periodic structures, having applications as sound filters and acoustic mirrors, by the creation of band gaps. The latter category of photonic structures are attractive for controlling and manipulating light.

Recent advances in production techniques such as microfabrication processes [6, 17], interference lithography [7, 12, 20, 32, 35], and thermodynamically driven self-assembly [29, 37] have let to the ability to experimentally investigate the pattern transformation ranging between the millimeter length scale to the (sub)micron length scale [4, 4, 21]. Many of these study combine a numerical approach with an experimental approach to convincingly demonstrate these pattern transformations.

Combining these production techniques with the pattern transformation properties may lead to the technological advance of imprinting complex patterns during the fabrication process using a minimum number of steps. For example, this could be done by fabricating a relatively simple periodic structure of a stiff substrate, and subsequently heating the substrate above the glass temperature, applying a load to trigger the transformation, and then freezing the pattern in by cooling. This technique could be taken a step further whereby a reheating step would recover
the initial pattern [21].

An equally exciting prospect is that the pattern can be switched on and off by the application of an external force. This opens up the prospect of controllable sound and light filters at the (sub-)micron length scale.

The mechanical behavior of these structures is observed super-elastic. This response is characterized by an initial linear elastic behavior with a sudden departure from linearity to a plateau stress, coinciding with the transformation of pattern. Afterwards, the structure can deform at nearly constant stress. Cyclic loading confirms the reversibility of this behavior, whereby the initial cycle often shows a slight deviation [4, 21].

The current work continues on the investigation of two-dimensional periodic structures by [4]. In the latter, a periodic pattern of equally sized holes in a rubber matrix was investigated rigorously. Comparison between the experimental and the numerical analyses showed a good correspondence. Throughout this report we consider three-dimensional periodic structures and analyze them numerically using the techniques outlined in [4]. For the definition of these structures, the popular interference lithography fabrication technique will serve as an inspiration. Unfortunately, this choice forces us to rigorously consider the discretization of such structures.

This report is structured as follows. The problem definition is elaborated in chapter 2. Next, in chapter 3 the numerical modeling is discussed. In chapter 4 a detailed description is presented on the mesh algorithm. A number of different three-dimensional structures are then investigated in chapter 5. The report ends with summarizing the conclusions.
Chapter 2

Problem statement

Before the techniques that are applied to investigate pattern transformations in three-dimensional periodic structures are discussed, the problem statement is elaborated on. The current work continues on the work of Bertoldi et. al. [4] who performed a rigorous analysis on the pattern transformation of a two-dimensional periodic structure comprising a periodic pattern of equally sized holes in a rubber matrix. In this work, this configuration was analyzed both experimentally and numerically. It was found that the pattern transformation is related to the buckling of the laminas separating the holes and that it's happens completely within the elastic regime. Two numerical techniques were introduced to retrieve the pattern transformation in structures that were assumed infinitely extendable in all directions. Also, the material response both prior to as well as after the pattern transformation was analyzed. An excellent agreement was found between the simulations and the experiments.

In this report the pattern transformation in three-dimensional structures is investigated, whereby it is assumed that the structure can be extended infinitely in all directions. The background of this assumption is discussed in section 2.1 illustrated by the configuration used in [4]. Then, we discuss the unit cell and the general boundary conditions that accompany this assumption, in section 2.2. Finally, the general description of the three-dimensional periodic structures that are used throughout the remainder of this report is introduced in section 2.3. Besides important restrictions, a number of important definitions are introduced in this chapter.

2.1 Two-dimensional periodic array of holes

Bertoldi et al. [4] concluded that the pattern transformation in periodic structures is the result of mechanical instability. This is closely related to instabilities often observed in cellular solids. Opposite to the latter category, the pattern transformation is not localized in bands, rather it occurs homogeneously throughout the structure. Following these observations, a two-dimensional periodic array of equally holes in a neo-Hookean elastic rubber matrix is modeled, schematically represented in Figure 2.1(a). Then, a uniaxial compression is applied in vertical direction.

To retrieve the pattern transformation, the buckling of the laminas separating the holes in analyzed. Therefore, the matrix material is discretized in triangular elements; subsequently the *BUCKLE option of the commercial Finite Element implementation ABAQUS is utilized. This type of analysis corresponds to an eigenvalue problem, whereby the eigenvalue corresponds to (a scaled parameters of) the critical load at which the buckling of occurs. The eigenmode then represents the scaled post-buckle deflection. For the periodic array of holes this deflection is shown in Figure 2.1 wherein we clearly see a change of pattern sufficiently remote to the boundaries. Specifically an periodic ellipsoidal pattern appears, whereby the primary axis of one ellipse is perpendicular to that of its nearest neighbors.
In this study, we are not particularly interested in the boundary effects present in the previous analysis. Furthermore, structures that are used in practice are characterized by a large repetition of the primitive structure. The assumption is thus made that the primitive structure repeats infinitely in all directions, thus eliminating boundary effects. Subsequently, all analyses are performed on a representative portion of the infinite, or macroscopic, structure. Some general characteristics of this microscopic analyses our outlined in the next section.

2.2 Unit cell with periodic boundaries

The analyses of an infinite periodic structure are performed by means of microscopic analyses. In this section the general microscopic analyses are outlined for the Finite Element method, whereby special paid to let the response of these analyses best mimic the macroscopic response.

As is frequently demonstrated in multi-scale analyses the microscopic analysis of an infinite structure is performed on a so-called Representative Volume Element (RVE). As was demonstrated above, the instabilities that are encountered induce a pattern transformation. Consequently the RVEs of the undeformed and post-buckle may be different. For the example we considered above in Figure 2.1 we observed that in the undeformed structure the RVE comprises a rectangle containing one hole. In the post-buckle configuration on the other hand the RVE contains four holes, two in each direction. The smallest RVE in the undeformed geometry shall be denoted primitive cell from this point forward. The RVE of the post-buckle configuration is then characterized by a periodic stacking of this primitive cell. This is characterized by the periodicity vector \( \vec{p} \) which is simply denoted periodicity.

In the following, we assume all primitive cells to have mutually perpendicular edges of equal length \( a \). This cell can then be characterized by a Cartesian coordinate system (see Fig. 2.2(a)). The latter can be used to define the periodicity \( \vec{p} \) using its components in each direction:

\[
\vec{p} = p_1 \vec{e}_1 + p_2 \vec{e}_2 + p_3 \vec{e}_3
\]  

(2.1)

whereby we restrict ourselves to integer values of \( p_i \), with \( i = 1, 2, 3 \). By convention, we usually omit the vector notation and simply denote the periodicity \( p_1 - p_2 - p_3 \). For the unit cell of Fig. 2.2(a) an example of the (undeformed) RVE of the post-buckle structure characterized by the periodicity \( 3 - 2 - 1 \) is included in Fig. 2.2(b).

In example of the periodic structure of holes in a matrix the post-buckle structure is thus characterized by a periodicity \( 2 - 2 \). For this example both the primitive cell and the RVE are...
2.2. UNIT CELL WITH PERIODIC BOUNDARIES

(a) primitive cell

(b) periodicity $3 - 2 - 1$

Figure 2.2: Sketch of the periodicity $3 - 2 - 1$ (b) of the primitive cell in (a).

(a) primitive cell

(b) RVE

Figure 2.3: Primitive cell and RVE of the periodic hole pattern of Fig. 2.1

included in Figure 2.3, where the primitive cell is a rectangle of equal length $a$ with a hole of diameter $9a/10$. In this diagram we can observe that the RVE of the post-buckle structure is in fact a stacking of the primitive cell characterized by the periodicity. Defining the primitive cell and periodicity thus suffices to define the RVE of the post-buckle structure. We remark that the RVE of the post-buckle structure is defined in the undeformed state, while it is characterized by the buckling deflection.

Now that we have defined both the primitive cell and the RVE, we can elaborate on the general approach to let the microscopic response mimic the macroscopic response. It was remarked above that the microscopic analysis is performed on the RVE. Following popular multi-scale approaches, periodic boundary conditions are applied to the RVE, in which the deformation of the boundary is coupled to the macroscopic deformation. More specifically, pair of node is selected for which the deformation is prescribed as:

$$u \left( X + R \right) - u \left( X \right) = (F - I) \cdot R = H \cdot R \quad (2.2)$$

In this equation $X$ denotes the position in the undeformed configuration, $R$ is the vector connecting the nodes of the pair. The macroscopic deformation gradient tensor $F$ coincides with the volume average of the corresponding local quantities. Which nodes constitute the nodes pairs is discussed below, however we remark here that the nodes on the boundaries are required to be periodic in the sense that a boundary node must have a counterpart on the boundary it opposes. Applied to the cubic unit cell, when a boundary node is located on the boundary plane at $0\vec{e}_1$ another node must be located at $a\vec{e}_1$.

The periodic boundary conditions (2.2) can be implemented in a number of ways. We choose to couple each boundary-node pair to a set of virtual nodes. This has the benefit that each degree of freedom of these virtual nodes corresponds to a component of the macroscopic deformation $H$. Consequently, the macroscopic (homogenized by volume average) can directly be retrieved.
from the corresponding reaction forces $\mathcal{F}$, 

$$ u^i \left( \vec{X} + \vec{R} \right) - u^i \left( \vec{X} \right) = \sum_{j=1}^{3} e_j^i \vec{R}^i \quad (2.3) $$

In the previous we have thus introduced that we apply a deformation to the RVE. Generally, some components of the macroscopic deformation $\mathcal{H}$ are prescribed, and others are derived from the equilibrium we are solving. For example for uniaxial loading in 11-direction, we apply the strain $\varepsilon$ to $H_{11}$, the components $H_{22}$ and $H_{13}$ are then determined from stress equilibrium; all other components of $\mathcal{H}$ are set zero.

We have not yet considered which boundary-nodes constitute the node pairs. For (cubic) unit cells we distinguish three groups: faces, edges, and corners. For consistency we introduce a unit cell coordinate system originating from one of the corners of the unit cell. We set the nodes with position $\vec{X}$ as the face, edge, or corner that coincides with this coordinate system. In Figure 2.4 the links are illustrated; in this figure the coordinate system is highlighted. We observe that a face is linked to the face it opposes. On the other hand, three edges are linked to the edges that coincides with the relevant coordinate axis. Next, seven corners are linked to the corners coinciding with the origin of the coordinate system.

In this section we have introduced the primitive cell and Representative Volume Element. In the next section, the primitive cells of a large family of three-dimensional structures is introduced. The description of this family is a direct consequence of a fabrication technique that can be used to produce these structures.

### 2.3 Three-dimensional periodic structures

Among others, interference lithography is commonly used to fabricate structures with a typical length scale varying from millimeters to micrometers or even smaller [7, 8, 17, 32, 35]. In the book by Maldovan and Thomas [20] interference lithography is introduced as an efficient technique for the fabrication of (three-dimensional) periodic structures. The technique encompasses the non-homogeneous distribution of energy, generated by the superposition of electromagnetic planar waves, which is registered in a photoresist material; which allows for the fabrication of large-area periodic structures which are defect-free. Since this technique operates using electromagnetic waves, a Fourier series provides an efficient input. By definition, Fourier series describe structures that repeat periodically in space. By manipulation of its coefficients a large number of different periodic structures are defined by this series. In [20] a significan portion of them is presented. It is found here that the primitive cell is defined as a cubic cell of unit length $a$. This constant corresponds to the lattice constant of the periodic structure.
Let us consider one example taken from [20]. As a consequence of using the Fourier series, the structure is defined by an implicit level-set function

$$f(x, y, z) = \cos\left(\frac{2\pi x}{a}\right) \cos\left(\frac{2\pi y}{a}\right) + \cos\left(\frac{2\pi y}{a}\right) \cos\left(\frac{2\pi z}{a}\right) + \cos\left(\frac{2\pi z}{a}\right) \cos\left(\frac{2\pi x}{a}\right) + t$$

(2.4)

with $a$ the lattice constant and $x, y, z$ the coordinates in the direction of the local Cartesian coordinate system defined in the center of the unit cell, thus ranging from $-a/2$ to $a/2$ in each direction. This local coordinate system coincides with the global coordinate system (defined above by $\vec{e}_i$, with $i = 1, 2, 3$). Furthermore, the coordinate system aligns with the edges of the unit cell. This definition is illustrated by Fig. 2.5(a). The constant $t$, in the equation above, determines the porosity, or volume fraction of material. When the following definition is applied:

$$f \leq 0 \rightarrow \text{material}$$
$$f > 0 \rightarrow \text{air}$$

(2.5) (2.6)

and we take $t = 0.90$, a Finite Element discretization of this structure is drawn in Figure 2.5(b).

This example illustrates two key features of the periodic structures that are considered throughout this report. Firstly, the structure is defined using a function for which the boundary of the structure is located at the zero-level set; and the material is located at negative level-set values. Secondly, periodic structures are defined by the implicit variants of these functions. Consequently, an equation has to be solved to locate the boundary of this structure. In contrast, the two-dimensional example of a circular holes in rubber matrix, that was presented above, is defined by the explicit function

$$f(x, y) = x^2 + y^2 - t^2$$

(2.7)

where $t$ defines the porosity, specifically it defines the radius of the circle. These two features play a critical role in meshing, or discretization, of the structure which is further investigated in chapter 4.

In this chapter we have introduced the problem considered in this report. We have seen that the pattern-transformation of three-dimensional structures is considered in structures that can be extended infinitely in all directions. This gave rise to the definition of primitive cells and Representative Volume Elements which are used to use microscopic analyses to calculate the response of the entire structure. The discretization of the primitive cell has to defined such that it meets the periodicity requirements posed by the boundary conditions, which is defined in chapter 4. First, we discuss the analyses we perform in the next chapter.
To find the critical load, as well as the post-transformation pattern of the infinite structure; two approaches are possible. The two approaches comprise a simple buckle analysis performed on a Representative Volume Element (RVE) defined for the post-transformation structure, or a Bloch analysis on the primitive cell. It will be shown that the first technique is simple but suffers from severe computational limitations particularly in three-dimensions. The second technique is more favorable and is thus used for the analysis of three-dimensional periodic structures.

In the remainder of this chapter the derivation of the buckling analysis for a one-dimensional example is performed in section 3.1. Thereafter, we return to the problem as it was formulated in the previous chapter. In section 3.2 the material behavior of the rubber matrix is discussed. Next, the conditions that must hold to perform a microscopic stability analysis are discussed. In sections 3.4 and 3.5 the buckling and Blochwave technique are discussed that are used to perform the stability analysis. Finally, in section 3.6 the method to obtain the complete stress–strain response both prior and post to buckling is discussed.

### 3.1 General introduction to a buckling analysis

Buckling is the phenomenon where a body can suffer severe deformation originating from a small perturbation. Consequently, the body behaves unstable. Stability is best illustrated by considering the equilibrium of a ball, such as in Figure 3.1. From this diagram, we can observe that three different states of equilibrium are distinguished: stable, neutral, and unstable. To determine the state of equilibrium of a body, not by considering the equilibrium state itself, but by considering the effect of small perturbation. Indeed, although the configurations of Figure 3.1 all represent the same equilibrium state, the response to a perturbation in horizontal position of the ball is different. In the case of a stable equilibrium configuration, in Fig. 3.1(a) the ball returns to its original equilibrium position immediately after the perturbation. For the neutral equilibrium in Fig. 3.1(b) the ball remains at a new equilibrium position. The most dramatic effect is observed for the unstable equilibrium in Fig. 3.1(c) where the ball will not reach equilibrium. The latter corresponds to the buckling phenomenon. Buckling is therefore investigated by considering the effect of a small perturbation.

For academic problems, both the buckling load as well as the shape of the deformed geometry can be calculated analytically. In other cases, the Finite Element method is popular to perform such an analysis. In this section, we introduce the buckling analysis for the Finite Element method. The buckling load as well as the deflection of the geometry are retrieved by solving an eigenvalue problem. This problem is completely derived for a one-dimensional problem below, generalization to two- or three-dimensions is easily accomplished.

The reference problem is that of the Euler-Bernoulli beam. This well know example comprises
a thin beam subjected to an uni-axial load. The problem is illustrated in Figure 3.2 where we observe the beam of length $L$ is subject to a load $P$ on one-side; on the other side the beam is fixed. Both the undeformed geometry and the buckled deflection, characterized by $\psi(x)$ (with $x$ the position), are included in this diagram. We assume that no body forces and, rather than a static problem, assume a dynamic problem (since buckling does not correspond to a static equilibrium):

$$EA \frac{d^2 u}{dx^2} - \rho AL \ddot{u} = 0$$

Herein we have assumed the beam of uniform cross-section $A$, and to satisfy isotropic linear elastic constitutive behavior characterized by the Young’s modulus $E$. $u(x)$ is the displacement of the beam calculated using linear elasticity, as opposed to the deflection $\psi(x)$ that constitutes the buckled shape. Consequently, $\ddot{u}$ is the acceleration or second order time derivative $u(x)$. Finally, the mass is accounted for through the density $\rho$ of the material.

Returning to the problem definition in Figure 3.2 we observe the following boundary conditions to characterize this problem:

$$u(x = 0) = u(x = L) = 0$$
3.1. GENERAL INTRODUCTION TO A BUCKLING ANALYSIS

Which is generally not sufficient to uniquely solve (3.1). However, for the buckling analysis the latter choice proves useful.

Next (3.1) is multiplied by a test function and integrated over the domain \( x \in (0, L) \). Subsequently, the equation is integrated by parts and an approximation is introduced, specifically we divide the domain into \( m = n - 1 \) linear elements – drawn in Fig. 3.3 – with \( n \) the number of nodes. Finally, we arrive at the following system of equations:

\[
\alpha K \ddot{u} + M \dot{u} = \beta q \quad (3.3)
\]

where the column \( q \) contains the external forces on each node, \( u \) comprises the displacement of each node. The constants govern

\[
\alpha = \frac{E}{\rho L} \quad \beta = \frac{1}{\rho AL} \quad (3.4)
\]

It is easily verified that the so-called stiffness matrix \( K \) and mass matrix \( M \) are calculated as follows:

\[
K = \int_0^L \frac{dN}{dx} \frac{dN^T}{dx} \, dx \quad (3.5)
\]

\[
M = \int_0^L N(x)N^T(x) \, dx \quad (3.6)
\]

wherein \( N(x) \) the – linear – interpolation functions.

The boundary conditions of (3.2) require the system (3.3) to be partitioned. The buckling analysis in only applied on that part of the system for which the displacements are not prescribed. In particular, the buckling load \( k \) and deflection \( \psi_k \) satisfy the following eigenvalue problem [16]:

\[
\begin{pmatrix}
K_{ff} + \frac{\lambda_k}{\alpha} M_{ff}
\end{pmatrix} \psi_k = 0 \quad (3.7)
\]

with \( f \) corresponding to the interior nodes. By definition, the absolute magnitude of the eigenmode \( \psi_k \) is not determined; the boundary conditions of (3.2) determine both ends of the domain.

The Euler-Bernoulli beam is a classical textbook example, for instance in [11, 16]. The references provide an analytical solution for the critical buckling deflection

\[
\psi_k(x) = A \sin \frac{\pi k x}{L} \quad k = 1, 2, ..., n \quad (3.8)
\]

where \( A \) is an arbitrary scaling constant. In the diagram of Figure 3.4 we observe that the analytical and numerical solution coincide, for the first, second and third eigenmode (i.e. \( k = 1, 2, 3 \) respectively). Notice that the analytical and numerical deflections are scaled such that they obtain the same maximum.

Above a dynamic, rather than a static, problem was assumed to arrive at the eigenvalue problem in (3.7). For large displacement theory no such assumption is required. In this case, the incremental stiffness matrix \( K_t \), generalized to three-dimensions, can be directly applied to find the buckling load \( k \) and deflection \( \psi_k \) from the following eigenvalue problem:

\[
\begin{pmatrix}
K_0 + \lambda_k K_t
\end{pmatrix} \psi_k = 0 \quad (3.9)
\]

herein, \( K_0 \) is the initial stiffness matrix. The latter analysis is performed in the commercial ABAQUS software in the function *BUCKLE.
Figure 3.4: Buckling deflection of an Euler-Bernoulli beam, for $\alpha = 1 \text{ m/s}^2$ and 20 linear elements. The eigenmodes are displayed as crosses, whereas the analytical solution is formed by a solid line. Both are scaled such that the maximum deflection coincides.

A final remark is made with respect to the scaling of the buckling deflection (or eigenmode). The physical explanation traces back to the discussing of stability, where it was concluded that the phenomenon of buckling corresponds an unstable equilibrium. In this case (Fig.3.1(c)) the system never arrive at a new equilibrium, and no precise values can be retrieved.

### 3.2 Material behavior

In the previous section we have made a side step to general concept of the buckling analysis for the Finite Element method. In this section we return to the main problem, specifically the material model of the rubber matrix is introduced.

For the rubber, we choose a material that is rate-independent and incrementally linear material, and assume that all deformation is elastic. We thus model the material as a compressible neo-Hookean solid. The material model is characterized by the strain energy density

$$W = \frac{G}{2} (I_1 - 3) + \frac{K}{2} (J - 1)^2 + G \log J$$

whereby $I_1$ the first invariant, defined as the trace of the right Cauchy-Green tensor $C = F^T F$. On the other hand, the third invariant is defined as the determinant of this tensor. Consequently, the compressibility $J$ is the square root of the latter invariant. The materials constant comprise the shear modulus $G$ and the bulk modulus $K$. In our three-dimensional analyses we set the Young’s modulus $E = 1 \text{ Pa}$, and the material is taken nearly compressible by setting the Poisson’s ratio $\nu = 0.4997$. We can relate these parameters to the shear and bulk modulus through:

$$G = \frac{E}{2(1 + \nu)} \quad K = \frac{2G(1 + \nu)}{3(1 - 2\nu)}$$

In the remainder of this chapter two methods are discussed to investigate the instabilities.
3.3 Macroscopic instability

In the previous chapter we have outlined that we assume a structure that can be extended infinitely in all directions. We have introduced that this assumption opens the perspective to perform a microscopic analysis on a Representative Volume Element (RVE) to retrieve the macroscopic response. For the stability analysis, which is performed to find the post-transformation structure, the microscopic analysis can only represent the macroscopic behavior if the instability is in fact microscopic, i.e. the post-transformation structure is characterized by a low periodicity. In this section the loss of ellipticity analysis is introduced to check that the (possible) microscopic instability is favorable over the macroscopic instability.

In [4, 33] it is shown that the loss of ellipticity analysis is performed on the primitive cell (i.e. the RVE of the undeformed geometry). To mimic the macroscopic behavior periodic boundary conditions, such as in (2.2), are applied. The loss of ellipticity conditions states that if an instability occurs it is microscopic only if

\[ L_{ijkl}^H N_j N_l m_k > 0 \; \forall \vec{N}, \vec{m} \tag{3.12} \]

In this equation \( L^H \) is the homogenized fourth order tangent moduli tensor. \( \vec{N} \) and \( \vec{m} \) are unit vectors. When we have determined \( L^H \) we thus check all combinations of two vectors on the locus of unit sphere (or unit circle in 2-D), whereby any vector \( \vec{n} \) on this locus satisfies:

\[ \vec{n} = \sin \theta \cos \varphi \vec{e}_x + \sin \theta \sin \varphi \vec{e}_y + \cos \theta \vec{e}_z \; \; 0 \leq \varphi \leq 2\pi \; \text{and} \; 0 \leq \theta \leq \pi \tag{3.13} \]

In practice, we incrementally load the primitive cell to find at which load (e.g. strain or temperature difference) loss of ellipticity occurs. In other words, we determine up to which load the instability can be found by performing a microscopic analysis. Therefore, we have to calculate the homogenized tangent moduli \( L^H \) at the end of each loading increment. Due to our choice of rate-independent and incrementally linear material behavior, this tensor is related to the homogenized rate of first Piola-Kirchhoff stress \( \dot{\vec{S}} \) and the homogenized rate of deformation gradient \( \dot{\vec{F}} \):

\[ \dot{\vec{S}} = L^H : \dot{\vec{F}} \tag{3.14} \]

the homogenized quantities are retrieved by taking the volume average of the local quantities. When the periodic boundary conditions are linked to virtual nodes as in (2.2), \( \dot{\vec{F}} \) is directly applied to these nodes. The reaction force on these nodes directly correspond to the components of \( \dot{\vec{S}} \). \( L^H \) is fully determined by applying nine independent perturbation tensors \( \dot{\vec{F}} \) and relating it to the stress response through \( L^H \) at the end of each increment.

We again consider the 2-D example of a periodic structure of holes in a rubber matrix. The loss of ellipticity analysis for this example is performed on the primitive cell of Fig. 2.3(a), which is subject to constrained swelling. For this case \( \dot{\vec{H}} = 0 \) and a temperature difference \( \Delta T \) is applied to the entire volume. At each incremental temperature we calculate the left-hand side of (3.12) to obtain a typical curve as is included in Figure 3.5. From this curve we can conclude that for a temperature difference of \( \alpha \Delta T = 0.04 \) at maximum a microscopic instability analysis may be performed.

3.4 Stability analysis of infinite periodic structures by means of the buckling analysis

In the first section the buckling analysis using the Finite Element method was briefly discussed. In this section, we apply this analysis to investigate the pattern change of infinite periodic structures, whereby we assume that (3.12) is satisfied at all times. The buckling analysis is performed using the ABAQUS software by means of the *BUCKLE option, which yields the eigenvalue \( \lambda \) representing a scale factor of the buckling load as well as the the deflection mode \( \psi(\vec{x}) \) defined in...
3.5 Stability analysis of infinite periodic structures by means of the Bloch analysis

Contrary to the buckling analysis, the Bloch analysis may be performed on the primitive cell even though it does not coincide with the RVE of the post-transformation configuration \[13\]. In this analysis we determine at which load an instability may occur for a certain periodicity.
3.5. STABILITY ANALYSIS OF INFINITE PERIODIC STRUCTURES BY MEANS OF THE BLOCH ANALYSIS

Figure 3.6: Eigenmodes, or buckled deformation modes, for RVEs comprising different periodicities of the unit cell of Fig. 2.3(a).

Figure 3.7: The first eigenvalue $\lambda_1$, corresponding to the scaled buckling load, for RVEs comprising different periodicities.
and repeat this for a number of different periodicities to obtain the most favorable (i.e. the configuration for which the instability occurs at the lowest loading increment). We perform a microscopic analysis, and thus investigate those loads for which a microscopic instability is also macroscopically favorable, i.e. for those loads at which (3.12) is satisfied.

In the Finite Element method the instability corresponds to a singular stiffness matrix $K$, i.e. a zero eigenvalue. To find the load at which the instability occurs, for each periodicity $\vec{p}$ we incrementally load the primitive cell and at the end of each increment calculate the eigenvalue of the stiffness matrix. The increment at which a zero eigenvalue is found thus corresponds to the load at which the instability occurs; the eigenmode at this increment corresponds to the buckling deflection mode.

To be able to check an arbitrary periodicity $\vec{p}$ using the primitive cell, we replace the periodic boundary conditions with the classical Bloch condition (for example found in [4, 18]) in the eigenvalue problem. Since we operate under the assumption that the primitive cell is cubic, of unit length $a$ and aligns with the Cartesian coordinate system; the classical condition simplifies to:

$$\vec{u}(\vec{X} + \vec{R}) = \vec{u}(\vec{X}) \exp\left(\sqrt{-1} \frac{2\pi \vec{p}}{\bar{a}}\right) \quad (3.15)$$

which replaces (2.2) in the eigenvalue analysis at the end of each load increment.

As most commercial implementations, the ABAQUS software does not provide for imaginary values. Following [1] we model a real and an imaginary mesh. Both meshes are subjected to the periodic boundary conditions in (2.2), and are thus uncoupled, during the loading increment. During the calculation of the eigenvalue at the the end of each increment using (3.15) we apply the real part of this equation to the real mesh and the imaginary part to the imaginary mesh:

$$\vec{u}^{\text{Re}}(\vec{X} + \vec{R}) = \vec{u}^{\text{Re}}(\vec{X}) \cos\left(\frac{2\pi \vec{p}}{\bar{a}}\right) - \vec{u}^{\text{Im}}(\vec{X}) \sin\left(\frac{2\pi \vec{p}}{\bar{a}}\right) \quad (3.16)$$

$$\vec{u}^{\text{Im}}(\vec{X} + \vec{R}) = \vec{u}^{\text{Re}}(\vec{X}) \sin\left(\frac{2\pi \vec{p}}{\bar{a}}\right) + \vec{u}^{\text{Im}}(\vec{X}) \cos\left(\frac{2\pi \vec{p}}{\bar{a}}\right) \quad (3.17)$$

For the 2-D example of a periodic stacking of holes, a typical result is plotted in Figure 3.8. As for the loss of ellipticity example, the primitive cell of Fig. 2.3(a) is subjected to uniaxial swelling due to a temperature difference $\Delta T$, whereby $\vec{F} = 0$. The range of temperature difference is constrained due to this analysis, since we are performing a microscopic analysis and thus must be convinced that a microscopic instability is favorable. From the result we conclude that, as expected, the 2−2 configuration is most favorable as it yield a zero eigenvalue first.
3.6. POST-TRANSFORMATION ANALYSIS

At the loading increment at which the eigenvalue is zero for a given periodicity \( \vec{p} \), the eigenmode corresponds to the post-buckle deflection. This eigenmode is defined only on the primitive cell, but can be used to reconstruct the mode of the complete post-transformation RVE. The eigenmode \( \vec{\Psi} \) is defined on each node of the mesh and has a real and imaginary part. Due to the previous split, we start by combining the eigenmode of the real and imaginary mesh:

\[
\vec{\Psi}(\vec{X}) = \vec{\Psi}_{\text{Re}}(\vec{X}) + i\vec{\Psi}_{\text{Im}}(\vec{X})
\]

where the imaginary unit \( i \). We then apply the Bloch condition to reconstruct the RVE comprising \( \vec{p} \) unit cells. Herein:

\[
\vec{\Psi}(\vec{X} + \vec{R}) = \vec{\Psi}(\vec{X}) \exp \left( i2\pi\vec{R} \cdot \vec{R} \right)
\]

Where \( \vec{R} \) is again defined as the vector containing two nodes. Contrary to (3.15), (3.19) is no longer confined to the boundary nodes, but is applied to all nodes.

It is best to consider an example. For the periodic stacking of holes the primitive cell and RVE are sketched in Figure 2.3. We start by adding the (complex) eigenmode \( \vec{\Psi}(\vec{X}) \), resulting from the Bloch analysis, to the nodal positions of the primitive cell. The real part of the result is represented in Fig. 3.9(a). Subsequently, the mode for the entire RVE is reconstructed through (3.19). This procedure is illustrated by Fig. 3.9(b). For example, all nodes in the light gray primitive cells denoted by blue crosses are calculated from the corresponding node in the primitive cell (marker dark gray). For the lower right corner we then find \( \vec{R} = \vec{e}_1 \), for the upper left corner \( \vec{R} = \vec{e}_2 \) and finally for the upper right corner \( \vec{R} = \vec{e}_1 + \vec{e}_2 \).

3.6 Post-transformation analysis

In the previous section we have defined all analyses needed to construct both the shape and periodicity of the transformation of pattern. What remains is define the method that is used to retrieve the stress strain curve of both prior to and after the transformation of pattern. In [4] it is found that the complete stress–strain curve is calculated by performing a microscopic analysis on the RVE (i.e. a stacking of the primitive cell according to the periodicity). Needless to say, the periodic boundary conditions of (2.2) are applied to the RVE.

To retrieve the stress–strain curve also after the transformation of pattern, the (reconstructed) eigenmode \( \vec{\Psi}(\vec{X}) \) is added to the mesh a small perturbation. This is exaggeratedly
Figure 3.10: RVE used to calculate the stress–strain behavior to prior to and after the transformation of pattern of a structure characterized by the unit cell in Fig. 2.3(a).

Figure 3.11: Stress–strain response taken from [4], whereby the porosity is approximately 60% as resulting from a diameter of 9\(a/10\) and the plane strain condition is applied for the third direction.

Illustrated if Fig. 3.10 for the periodic stacking of holes considered throughout this chapter. For this example a typical stress–strain response is shown in Fig. 3.11 which is characteristic for the pattern transformation of periodic structures. In this result we observe so-called super elastic behavior, which is calculated by a linear elastic response prior to the transformation of pattern. At the instance of the instability the stress reaches a stress-plateau whereby the structure continues to deform at nearly constant stress.
Chapter 4

Obtaining a periodic mesh

This chapter covers the method that we use to obtain periodic meshes of complicated three-dimensional geometries that may be concave and are defined by level-set functions that may be implicit. A mesh is a discretization of an arbitrary geometry by means of a finite number of elements (for example Figure 4.1). Each element is composed of a number of nodes, the number and relative position within the element is dependent on its type. For instance for simple linear elements all nodes are positioned at the intersection of element-edges. This information, including the node numbering, is included in the connectivity. For example, three nodes form a linear triangular element in two-dimensions (2-D). The connectivity then comprises nodes one, two, and three.

Previously, we have assumed an infinite structure. Therefore, the analysis is performed on RVEs. In this study we limit ourselves to cubes (or rectangles in 2-D) of unit edge length. The analysis requires the application of (quasi-)periodic boundary conditions. The discretization, or mesh, is therefore required to be periodic in the sense that each node on a boundary (or edge of the unit cell) should have a counterpart on the opposing boundary. This implies that only a unit variation in the position vector is allowed in the relevant direction, the other components should coincide for each node pair.

Obtaining a periodic mesh is often not standard in commercial algorithms. Therefore, we investigate the strategy to obtain such a mesh in detail. The mesh algorithm, capable to mesh geometries defined by implicit functions, is discussed in sections 4.1 and 4.2. Thereafter, we discuss the approach to obtain a periodic mesh in section 4.3. For the most part, the concepts are not specific to the algorithm selected. Finally, we discuss how the functions used to define the geometry relate to stability of the algorithm. We moreover discuss the definition of such functions for specific cubes and tetrahedrals (and their two-dimensional counterparts) used to obtain primitive cells, or unit-cells, in section 4.4.

Figure 4.1: Example of a mesh, or discretization in a finite number of elements, of an arbitrary geometry.
4.1 Selection of a mesh algorithm

Review of the literature concerning mesh algorithms reveals a number of different approaches. Bossen and Heckbert [5] divide these approaches in four categories. The first two categories use a mathematical smoothing technique to, iteratively, smoothen the nodes such that the geometry is discretized optimally. A common mathematical smoothing technique is Laplacian smoothening; where, in each iteration, each node is repositioned to the average position of all its nearest neighbors [12]. The nearest neighbors are determined by means of the connectivity, which marks the difference between the two categories. One of these categories iteratively calculates the connectivity, allowing insertion and deletion of nodes; whereas the other category calculates the connectivity only once, after the initiation of the nodes, thus prohibiting insertion or deletion of nodes and allowing only minor displacements relative to the initial positions. The last two categories are both termed pliant mesh generation algorithms. These kind of algorithms often simulate a physically based attraction and repulsion behavior between nodes. Again, the difference between these two categories is in the calculation of the connectivity. One of these categories incrementally repositions, inserts, or deletes nodes and only afterwards calculates the connectivity. The other category incorporates the calculation of the connectivity in the iterative process.

The first two categories of mesh algorithms, generally improve the element shape. However, on some concave domains – which are common in periodic structures [20] – nodes are pulled outside the boundary. This reference suggests using a pliant method in which calculation of the connectivity is incorporated in the iterative process. The implementation by Persson and Strang [28] is an example of such a method. This method is not restricted to convex geometries. Furthermore, (implicit) level-set functions – also common to periodic structures – can serve as an input for this algorithm. Conveniently, the so-called Distmesh implementation in MATLAB is freely available for an arbitrary number of dimensions.

A very popular way to calculate the connectivity for tetrahedral elements from an arbitrary number of nodes, is the Delaunay algorithm. This algorithm is a triangulation method that maximizes the smallest angle between the element edges, equivalent to the property that the circumspheres of the elements contain no other triangulation points (i.e. nodes) in their interior. The popularity of the algorithm is due to its long existence and computational efficiency [10, 12, 19].

4.2 Distmesh: a simple mesh generator in Matlab

In the previous section, review of the literature lead to the choice of a so-called pliant mesh algorithm in which, besides the nodal positions, the connectivity is updated in each iteration. It was concluded that the Distmesh implementation by Persson and Strang suits our purpose well. In this section we summarize the background of this algorithm, which is useful in applying this algorithm to obtain periodic meshes of periodic structures (in section 4.3). In the implementation, we confine ourselves to linear tetrahedral elements. This type of elements corresponds to a simple three-node triangular element in two-dimensions, and a four-node tetrahedral element in three-dimensions. This assumption substantially simplifies calculating the connectivity, enabling the usage of the stable and efficient Delaunay triangulation algorithm. Once more, combining the mesh with a commercial Finite Element implementation becomes straightforward since the numbering (i.e. order in which the nodes are accounted for in the connectivity) is to some degree arbitrary.

To optimize the discretization, Persson and Strang model the mesh as a truss structure, thus introducing physically based interaction between the nodes. Each element-edge is modeled as a truss and each node as a joint. For simplicity, we refer to the joints of the truss structure as nodes. Besides an appropriate force-displacement relation for the trusses, boundary conditions are applied. These boundary conditions penalize nodes that fall outside the geometry that is discretized. Optimizing the discretization (i.e. meshing the geometry) thus corresponds to,
iteratively, solving force equilibrium for the truss structure.

We arrive at this equilibrium of forces by assigning a force–displacement relationship to each truss. The target geometry then enters through the boundary conditions, for which, at every boundary node, a reaction force is modeled normal to the face of the geometry. This force is taken just large enough to keep the node from moving outside the geometry. Generally, the total force $\vec{F}_i$ acting in a node $i$ then consists of two components: the sum of the internal forces $\vec{F}_{i\text{ int}}$ acting on that node and the external force $\vec{F}_{i\text{ ext}}$ applied only at the boundaries. By definition the latter equals zero for all interior nodes. Force equilibrium implies that the total force $\vec{F}_i$ has to equal zero for all nodes, which yields the following system of equations:

$$\vec{F}_i = \vec{F}_{i\text{ int}} + \vec{F}_{i\text{ ext}} = \vec{0}$$

for a set of equilibrium positions $\vec{p}$. The latter corresponds to a $N \times 1$ column containing the equilibrium position of each node, $N$ therefore denotes the number of nodes. Following this definition, the system of equations also consists of $N$ equations.

A simple approach to solve (4.1) is to introduce an artificial dependence on an artificial time $\tau$:

$$\frac{d\vec{p}}{d\tau} = \vec{F}_i$$

(4.2)

This allows for the application of the Euler explicit algorithm to numerically solve the system of equations (4.1):

$$\vec{p}(\tau + \Delta \tau) = \vec{p}(\tau) + \Delta \tau \vec{F} \left( \vec{p}(\tau) \right)$$

(4.3)

Operationally, the iterative procedure is applied in two steps. To this end, we introduce a pseudo time-step $\zeta = \Delta \tau$. In the iteration the first step comprises a ‘real’ iteration in which all nodes are moved whereby the internal forces act as driving force. In the second step all nodes exterior to the geometry are moved to the closest point on the surface of the geometry. Hence:

$$\vec{p}(\tau + \zeta) = \vec{p}(\tau) + \zeta \vec{F}_{\text{int}} \left( \vec{p}(\tau) \right)$$

(4.4)

$$\vec{p}(\tau + \Delta \tau) = \vec{p}(\tau + \zeta) + f \left( \vec{p}(\tau + \zeta) \right) \vec{F} \left( \vec{p}(\tau + \zeta) \right)$$

(4.5)

Before focusing on the second step in (4.5), we concentrate on the first step in (4.4). For this step external forces are set to zero. Therefore, only the internal forces constitute to the change in position. To calculate the internal forces, the trusses are modeled as a special kind of spring that only exert a repulse force when compressed. The force acting on node $i$ by extension of the truss between that node and node $j$ follows from the length $l_{ij}$ of that truss and its directionality $\vec{n}_{ij}$, whereby we define

$$l_{ij} = || \vec{p}^i - \vec{p}^j || \quad \text{and} \quad \vec{n}_{ij} = \frac{\vec{p}^i - \vec{p}^j}{l_{ij}}$$

Hence

$$F_{i\text{\rightarrow i}}^{\text{int}} = \begin{cases} k(l_{ij} - l_0) \vec{n}_{ij} & \text{if } l_{ij} < l_0 \\ \vec{0} & \text{if } l_{ij} \geq l_0 \end{cases}$$

(4.7)

where $k$ is the spring constant, for which we choose $k = 1$. The initial length of the truss $l_0$ corresponds to the desired length of the truss (i.e. element-edge), which the method allows to be chosen as a non-uniform distribution. The total internal force on node $i$ is the summation of the individual truss contributions (which are found using the connectivity):

$$\vec{F}_{\text{int}} = \sum_j \vec{F}_{j\rightarrow i}^{\text{int}}$$

(4.8)
The second step in (4.5), bringing nodes exterior to the geometry back to the closest boundary point of the geometry, is accomplished using the description of the target geometry. As the name Distmesh suggest, the algorithm uses a distance to define the target geometry. In section 2.3 the geometry was already defined as such, using a level-set function \( f(\vec{p}) \). The value of such a function corresponds to the distance of an arbitrary point in space to the boundary of the geometry. This implies that the surface of the geometry corresponds to the zero level-set. The inner and outer domain of the geometry are distinguished by assigning a negative or positive sign respectively. To illustrate this step, we let \( \vec{p}_i \) be the position of a node exterior to the geometry. The corresponding closest point on the surface of the geometry \( \vec{p}_i^* \), by definition, satisfies zero level-set:

\[
f(\vec{p}_i^*) = 0 \tag{4.9}
\]

Since \( \vec{p}_i^* \) is defined as the closest point on the surface of the geometry, we require \( \vec{p}_i - \vec{p}_i^* \) to be parallel to the gradient \( \nabla f(\vec{p}_i^*) \). For a node \( i \) exterior to the geometry, we thus need to solve

\[
L(\vec{p}_i) = \begin{bmatrix}
    f(\vec{p}_i^*) \\
    \nabla f(\vec{p}_i^*) \times (\vec{p}_i - \vec{p}_i^*)
\end{bmatrix} = \begin{bmatrix}
    0 \\
    \vec{0}
\end{bmatrix} \tag{4.10}
\]

Unlike the first step, a numerical approximation has to be made. This is already outlined in (4.5), whereby the distance is found by evaluating \( f(\vec{p}) \), defining the geometry, in each relevant node. The direction of the gradient in each relevant node \( \vec{t}_i \), is estimated numerically. To this end, a perturbation \( \psi \) is applied in each direction of the Cartesian coordinate system. For instance, the gradient in \( x \)-direction

\[
\Delta x^i = \frac{f(x^i + \psi, y^i, z^i) - f(x^i, y^i, z^i)}{\psi} \tag{4.11}
\]

The numerical estimation of the direction of the gradient

\[
\vec{t}_i = \left[ (\Delta x^i)^2 + (\Delta y^i)^2 + (\Delta z^i)^2 \right]^{-\frac{1}{2}} \left( \Delta x^i \vec{e}_x + \Delta y^i \vec{e}_y + \Delta z^i \vec{e}_z \right) \tag{4.12}
\]

Besides for the application of the boundary conditions, the geometry is used for the calculation of the connectivity. In principle the connectivity can be calculated using the Delaunay algorithm, which is readily available in standard MATLAB. For convex geometries, the standard algorithm yields satisfactory results, illustrated in Fig. 4.2(a) for a 2D circle. For concave geometries, on the other hand, the Delaunay algorithm fails to describe the geometry properly (Fig. 4.2(b)). As a solution the function \( f \) describing the geometry is consulted. To this end, fictitious nodes \( \vec{p}_{\text{mid}} \) are placed at the average position of all nodes spanning the element. Only those elements for which \( \vec{p}_{\text{mid}} \) is inside the geometry are included in the final connectivity. These elements thus satisfy

\[
f(\vec{p}_{\text{mid}}) < -\varepsilon \tag{4.13}
\]

where \( \varepsilon \) represents the accuracy. For convex geometries, this is satisfied for all elements calculated by the Delaunay algorithm. The results of this adapted algorithm is outlined in Figure 4.2(c).

In this section we have discussed the background of the Distmesh algorithm by Persson and Strang. We have seen that a physical analogy is assumed between a finite element discretization and a truss structure. The mesh is optimized using force equilibrium, applying boundary conditions to account for the geometry. The geometry is implemented using a level-set function, of which the zero level-set corresponds to the boundary. The influence of this function on the algorithm is investigated in section 4.3 but we mention here that either a continuous or a discrete (numerical) function can be used, or even a discrete set of values within a suitable domain inside and surrounding the geometry. In the next section a strategy to obtain a periodic mesh is discussed. The Distmesh algorithm serves as a starting point for the implementation thereof, but the concepts apply to any meshing technique.
4.3 Method of obtaining a periodic mesh

In this section we concentrate on calculating a periodic mesh, whereby we relate the strategy applied to the type of geometry. This way, computational effort is minimized while at the same time maximizing periodicity within the primitive cell. The DISTMESH algorithm serves as a starting point for the implementation, but the concepts are not restricted to the choice of algorithm. We begin by concentrating on two-dimensions, before generalizing to three-dimensions.

Let us begin by recapturing on the primitive cell such as it was defined in chapter 2. Generalized to three-dimensions we define the primitive cell as a cube with all edges of unit length $a$, which corresponds to the lattice constant for the specific structure. The cube is centered on the origin of a local Cartesian coordinate system, hence $x, y, z \in (-a/2, a/2)$. The direction of the edges coincide with the mutually perpendicular directions of the Cartesian coordinate system. Exemplifying for two-dimensions corresponds to focusing on the intersection plane at $z = 0$. The unit cell then reduces to a rectangle for which $x, y \in (-a/2, a/2)$.

Periodic structures commonly possess a number symmetries. The strategy proposed here uses these symmetries such that only a part of the unit cell is modeled; the periodic mesh is obtained through mirroring over the respective symmetry axes. We thus benefit from computational efficiency and enhanced periodicity within the unit cell.

The simplest situation occurs when a geometry posses symmetry about the axes of the Cartesian coordinate system on which the unit cell is centered, i.e.

$$f(x', y') = f(x, y) \quad \text{for} \quad (x', y') \rightarrow (-x, y), (x, -y)$$

(4.14)

From Figure 4.3(a) we can conclude that it suffices to mesh a quarter (hatched in the drawing) and mirror the result about each axis to obtain a mesh of the complete geometry. Using this approach, the edges of the unit cell mesh generated are automatically periodic. This type of geometry is so simple because a periodic mesh can be obtained using any mesh algorithm in its standard form. Notice that, when standard periodic boundary conditions are applied these conditions can be adapted such that the Finite Element analysis can be performed on this quarter, making mirroring obsolete.

A slightly more elaborate approach is required when the geometry posses symmetry about one or both diagonals of the unit cell (these diagonals by definition coincide with the diagonals of the Cartesian coordinate system), or none of the symmetries mentioned. In the former case, we reside in the approach outlined above in which the mesh is calculated for only part of the geometry, albeit adding constraints for the edges that coincide with an edge of the unit cell. In the latter case, where the geometry posses no symmetry, a boundary has to be enslaved to the boundary it opposes (Fig. 4.3(b)). From the implementation perspective both these cases are the same. However, using symmetry requires calculating a mesh on a significantly smaller geometry, making it computationally favorable. Especially in three-dimensions this has
CHAPTER 4. OBTAINING A PERIODIC MESH

Figure 4.3: Approach to obtain a periodic of a geometry that is symmetric about the axes of the Cartesian coordinate system [a] and of a geometry that posses no symmetry [b]. The letters denote arbitrary nodes on the edge, whereby in the latter figure, the underlined letters indicate that the position of that node is related to the corresponding node on the opposing boundary (e.g. the $x$-coordinate of $A$ is copied from $A'$).

a pronounced effect.

Below we concentrate on the implementation of additional constraints, focused on the Distmesh algorithm. Thereafter the details of these constraints are discussed for different geometry classes. This is then repeated for three-dimensions.

Imposing additional constraints on the geometry’s boundaries corresponds to imposing additional constraints to the balance of forces in (4.1). Particularly, a penalty method is applied in which additional equations are appended to (4.1). These equations can impose a boundary to become symmetric about a certain axis using an averaging procedure; or, ensure that the positions of nodes on one boundary are enslaved to corresponding nodes on another boundary. Rather than specifying equations, we choose to append the iteration with an additional step (following the application of external forces in (4.4–4.5)). For these particular constraints the benefit of using a mesh algorithm that calculates the connectivity during the iterative procedure is pronounced, since these constraints can encompass the addition or removal of nodes.

Computational efficiency is enhanced in two ways. The first employs the observation that the constraints are only imposed locally on the mesh: only on the boundaries that coincide with an edge of the unit cell. As a result, the overall distortion of these constraints is small. These constraints therefore need not be incorporated in each iteration. It is however essential that they are incorporated as the final step of the last iteration. The second enhancement in computational efficiency is accomplished by setting a distribution of nodes of the relevant boundary on beforehand and substituting this distribution instead of evaluating the constraints. The final distribution of nodes, for the boundary, can be approximated well by realizing that there is little room for variation considering the constraints and the desired spacing of nodes.

The specifics of the additional constraints are discussed next. As discussed, for a geometry that posses no symmetry simple constraints can be applied which involve enslaving each node on one boundary to the corresponding node on the boundary it opposes (Fig. 4.3[b]). For the latter two geometry classes, we distinguish geometries that posses symmetry about the positive and negative diagonal of the unit cell, and geometries that posses symmetry about only one of these diagonals. For the former we find

$$f(x', y') = f(x, y) \quad \text{for} \quad (x', y') \rightarrow (y, x), (-y, -x)$$

(4.15)

A diagram is included in Figure 4.4(a)–(b) from which the conclusion is drawn that the mesh is calculated for a quarter of the geometry (hatched in the diagram) and mirrored twice to obtain the full geometry. We observe that mirroring has to be preceded by ensuring the edge to be symmetric about the relevant axis. Periodic structures generally posses this extra symmetry. If not, the unit cell could not be repeated to obtain the infinite structure.
4.3. METHOD OF OBTAINING A PERIODIC MESH

A simple mirroring procedure is applied, which is repeated with extra constraints. In the latter case, the latter is illustrated for a geometry which possesses symmetry about only one diagonal.

For the latter class, the geometry possesses symmetry over only one of the diagonals, i.e.

\[ f(x', y') = f(x, y) \quad \text{for} \ (x', y') \rightarrow (y, x) \ \text{or} \ (-y, -x) \]  

besides the extra symmetry, one boundary of the meshed half is enslaved to the other boundary. In the diagram of Figure 4.4(c) we see that these boundaries are those which coincide with the edges of the unit cell.

In three dimensions, all concepts are the same. The main difference is that instead of edges that are lines the geometry has edges that are planes. Below the different types of geometries are repeated that were discussed for two-dimensions, with the exception of geometries that possess no symmetry since extension to three-dimensions is trivial.

When the geometry is possesses axes symmetry, i.e.

\[ f(x', y', z') = f(x, y, z) \quad \text{for} \ (x', y', z') \rightarrow (-x, y, z), (x, -y, z), (x, y, -z) \]  

calculating the mesh for one-eighth of the unit cell results in a periodic mesh as the result of mirroring (Fig. 4.5).

When the geometry only possesses symmetry about the (positive and negative) diagonals:

\[ f(x', y', z') = f(x, y, z) \quad \text{for} \ (x', y', z') \rightarrow (y, x, z), (z, y, x), (x, z, y), \]  
\[ (-y, -x, z), (-z, y, -x), (x, -z, -y) \]  

modeling one-twelfth suffices (Fig. 4.6(a)). To ensure that the resulting mesh is periodic, the boundary plane that coincides with the unit cell edge has to be symmetric about the relevant
CHAPTER 4. OBTAINING A PERIODIC MESH

Figure 4.6: Application of symmetry for a geometry that possesses symmetry about both positive and negative diagonals of the unit cell. (a) shows the highlighted part of the geometry for which the mesh is calculated. (b) shows the extra constraint.

Figure 4.7: Application of symmetry for a geometry that possesses symmetry about either the positive or negative diagonals of the unit cell. (a) shows the highlighted part of the geometry for which the mesh is calculated. (b) shows the extra constraint.

The geometry can also possess symmetry about either the positive or the negative diagonals:

\[ f(x', y', z') = f(x, y, z) \quad \text{for} \quad (x', y', z') \rightarrow (y, x, z), (z, y, x), (x, z, y), \]
\[ \text{or} \quad (-y, -x, z), (-z, y, -x), (x, -z, -y) \]  \hspace{1cm} (4.19)

Modeling one-sixth of the geometry is required (Fig. 4.7(a)). To obtain a periodic mesh, additionally one boundary plane that coincides with the unit cell edge should be made symmetric about the diagonal that is not incorporated in the global symmetry: also the edges of that plane, that coincide with the unit cell edges, should be made symmetric about the relevant axes (Fig. 4.7(b)). Finally, of the two boundary planes that coincide with unit cell edges, one should be enslaved to the one to which the additional constraints are imposed. Generally, this is allowed for periodic geometries.

The method to enhance computation efficiency are particularized by setting a complete plane of nodes on beforehand. This plane is then substituted for the computed plane at the end of the iteration. Notice how the mesh algorithm can be used to calculate this distribution, by restriction to a particular 2D plane. The relevant symmetries are then applied using an averaging procedure.

A final remark is made with respect to the element type. In the previous discussion linear elements have been assumed. In many types of analyses quadratic elements are preferable. The linear elements are transformed to quadratic ones by adding a node in the middle of each element edge. Since we are using tetrahedral elements (triangles on a plane) by definition the number of
4.4 Influence of level-set function on the behavior of the algorithm

In section 4.2, a level-set function $f$ was applied to calculate the external forces contributing to the overall force balance. Thus far, the $f$ was assumed known and it was assumed that its represents the distance from an arbitrary point to the surface of the geometry, positioned at $f = 0$. In this section we start of with a simple example. Then the latter assumption is elaborated on, as well as the influence of $f$ on the different parts of the algorithm where it is used. We finish with the definition of such a function for specific non-smooth geometries used to generate unit cells. As for the previous section, we restrict ourselves to two-dimensions in the examples. The concepts are however not submitted to such a restriction. In the last part of this section, specific three-dimensional geometries are considered.

Previously, a circle was used to illustrate the difference between convex and concave geometries. A simple circle (Fig. 4.2(a)) is defined by the level-set function

$$f_{\text{circle}}(\vec{p}) = x^2 + y^2 - r^2$$  \hspace{1cm} (4.20)

where $x$ and $y$ are the projections of the position vector $\vec{p}$ on the $x$- and $y$-axis of the Cartesian coordinate system on which the circle is centered respectively, $r$ is the radius of the circle. The concave geometry (Fig. 4.2(b)) is obtained by inverting the circle and calculating the difference between it and the rectangle. We then find:

$$f_{\text{concave}}(\vec{p}) = \max \left[ f_{\text{cube}}(\vec{p}), -f_{\text{circle}}(\vec{p}) \right]$$ \hspace{1cm} (4.21)

in which $f_{\text{cube}}$ defines the unit cell, generalized for three-dimensions. This example is confined to two-dimensions, hence

$$f_{\text{cube}} = f_{\text{cube}}(x, y, z = 0) = f_{\text{rectangle}}$$ \hspace{1cm} (4.22)

The details of these function are presented in appendix A.

More generally, we restrict ourselves to unit cells with a periodic geometry in the interior. All such geometries satisfy the general level-set function

$$f_{\text{unit}}(\vec{p}) = \max \left[ f_{\text{cube}}(\vec{p}), \alpha f_{\text{inner}}(\vec{p}) \right]$$ \hspace{1cm} (4.23)

wherein the constant $\alpha$ is introduced below.

For the example of the circle, it is trivial to show that $f_{\text{circle}}$ coincides with the distance, from an arbitrary point $\vec{p}$ to the surface of the circle. In reality, the level-set function $f$ does not necessarily coincides with this distance, rather it represents a scaled value. Therefore, the distance

$$d(\vec{p}) = \alpha_e f(\vec{p})$$ \hspace{1cm} (4.24)

with $\alpha_e$, a constant, termed the distance scaling factor in our framework. Since the surface of the geometry corresponds to $f = 0$, by definition the scaling has no effect on its location. According to this definition $\alpha_e = 1$ for the circle.

In practice, the exact distance scaling factor $\alpha_e$ is often approximated. For this approximation, denoted by $\alpha$,

$$|d(\vec{p}) - \alpha f(\vec{p})| = \delta$$ \hspace{1cm} (4.25)
CHAP fER 4. Obtaining a Periodic Mesh

Figure 4.8: Unit, or primitive cell defined by (4.23). The dimensions of the unit cell correspond to the lattice constant $a$ corresponding to the periodic structure constituting to the inner geometry. This particular geometry is defined by (4.28).

whereby $\delta \to 0$ for $\alpha \to \alpha_e$. Returning to the general definition in (4.23) we assumed $f_{\text{cube}}$ to exactly represent the distance. We thus find

$$d_{\text{unit}}(\vec{p}) \to \max [f_{\text{cube}}(\vec{p}), \alpha f_{\text{inner}}(\vec{p})] \quad \text{for} \quad \alpha \to \alpha_e \quad (4.26)$$

In general we approximate $\alpha_e$ by scaling the maximum value of $f_{\text{inner}}$ by maximum distance to an edge of the unit cell, equaling $a/2$. Hence

$$\alpha_e \approx \alpha = \frac{a}{2 \max(|f_{\text{inner}}|)} \quad (4.27)$$

A better approximation can be obtained numerically by calculating a fine grid of points on the surface of the geometry and then calculating the (minimum) distance to this surface starting from an arbitrary position(s) and comparing this to the value of $f_{\text{inner}}$.

The influence of the accuracy of the approximation of the distance scaling factor $\alpha$ is investigated for a periodic structure defined by

$$f_{\text{inner}}(\vec{p}) = \cos\left(\frac{2\pi}{a}[-x+y]\right) + \cos\left(\frac{2\pi}{a}[x-y]\right) + \cos\left(\frac{2\pi}{a}[x+y]\right) - t \quad (4.28)$$

which is adapted from reference [20]. Herein the lattice constant $a = 1$. The porosity (or volume fraction of material) is determined by the parameter $t$, for which we choose $t = 0$. The dimensions of the rectangle – defined by $f_{\text{cube}}$ – in which this geometry is confined, corresponds to the lattice constant, i.e. $x,y \in (-a/2,a/2)$. The optimized mesh is presented in Figure 4.8.

Below, the effect of different approximations $\alpha$ to the exact distance scaling factor $\alpha_e$ is exemplified. Therefore, we apply $\alpha >> \alpha_e$, $\alpha \approx \alpha_e$, and $\alpha << \alpha_e$, thus respective overestimating, approximating, and underestimating the exact distance. Concentrating on the geometry by (4.28) we respectively set $\alpha$: 1, $a/6$, and $a/30$. Judging from the optimized mesh displayed in Figure 4.8 we observe that the distance is $a/4$ at most. For the given values of $\alpha$, the estimated distance at different points along the x-axis (i.e. $y = 0$) is plotted in Figure 4.9. In this diagram the interior of the predicted geometry is highlighted gray. The predicted distance interior to the geometry is not relevant since the mesh calculation is dominated by the internal force in these regions. Exterior to the geometry we observe a (significant) overestimation of the distance for the upper curve, a fair approximation for the middle curve, and an underestimation for the lower curve.

When $\alpha \approx \alpha_e$ (i.e. using (4.27) we choose $\alpha = a/6$ for this geometry) the mesh algorithm yields satisfactory results. In Figure 4.10 the results of different steps of optimization of the nodal positions, undertaken by the mesh algorithm, are visualized. We observe that the algorithm starts with an initial grid of nodes in Fig. 4.10(a) from which the nodes that exterior to the geometry are rejected using (4.23) in Fig. 4.10(b). Then the positions of the remaining nodes are optimized, the result is found in Figure 4.10(c).
4.4. INFLUENCE OF LEVEL-SET FUNCTION ON THE BEHAVIOR OF THE ALGORITHM

Figure 4.9: The distance predicted by scaling $f_{\text{inner}}$ in (4.28) by different distance scaling factors $\alpha$. The predicted geometry is highlighted gray.

Figure 4.10: Nodal positions after different steps of the mesh algorithm, for $\alpha \approx \alpha_e$. 
When \( \alpha \) is chosen differently, such that \( \alpha f_{\text{per}} \neq d_{\text{per}} \), the first two steps (Fig. 4.10(a)-(b)) remain unchanged, since the location of the surface is not altered. However, the algorithm breaks down during the optimization step. If the distance is overestimated or underestimated, the external forces are either too large or too low respectively. As a result nodes outside the boundary are moved across the boundary, or left outside the boundary. This affects the internal forces, breaking down also this part of the iteration. When we analyze the result of the algorithm after a given number of iterations we observe for the former case that indeed nodes are concentrated in the neighborhood of the boundary in Fig. 4.11(a), whereas for the latter case little nodes are placed on the boundary (Fig. 4.11(b)). In the former case the external forces dominate the internal forces, which is reversed for the latter case.

So far we have observed the influence of the approximation of the scaling distance factor \( \alpha \) on the optimization of nodal positions. This factor also proofs of influence for the calculation of the connectivity, albeit less pronounced. Recall from section 4.2 that the geometry, for concave geometries, is used to correct the connectivity calculated by the Delaunay algorithm. Therefore (4.13) is evaluated for fictitious averaged nodes within each element. Following (4.23), we expect that too little elements are rejected if \( \varepsilon / \alpha \) is small (or \( \alpha >> \alpha_c \)). If the ratio is large (or \( \alpha << \alpha_c \)), on the other hand, too many elements are rejected. For the concave geometry of 4.8 we observe in Figure 4.12 that, as expected, the Delaunay algorithm fails. For large distance scale factor \( \alpha \) the same result is obtained as for \( \alpha \approx \alpha_c \), whereby the desired geometry is retrieved. On the other hand, for \( \alpha << \alpha_c \) too many elements are rejected.

We formulated the expectation that too little elements would be rejected for \( \alpha >> \alpha_c \), whereas Fig. 4.12(b) shows no such behavior. The expected behavior only becomes pronounced when little perturbations are applied to the mesh. By this we mean that nodes are positioned close to each other near the boundary. Choosing \( \alpha >> \alpha_c \) in this case results in very thin elements near the boundary (Fig. 4.13(a)) which can be problematic in a Finite Elements analysis. For \( \alpha \) approximately equal to the exact value, these elements are not included (Fig. 4.13(b)).
4.4. INFLUENCE OF LEVEL-SET FUNCTION ON THE BEHAVIOR OF THE ALGORITHM

The described problem is expected to occur often in 3D, especially for geometries for which the surface is characterized by high gradients.

Above, we have seen the influence of the distance scaling parameter $\alpha$ on the optimization of nodal positions as well as on the calculation of the connectivity. We have concluded that both for the optimization of nodes and for the calculation of the connectivity it is best to closely approximate $\alpha_c$. In practice, this observation is confirmed, whereby we remark that when the approximation is not exact we can best choose $\alpha$ slightly lower for the optimization of nodes than for the calculation of connectivity. In our discussion, we have assumed the level-set function of cubes and tetrahedrals to be known and exactly corresponds to the distance. Below we briefly comment on this assumption.

Cubes and tetrahedrals are non-smooth geometries that have straight edges. To calculate the distance we therefore distinguish a number of cases in which the distance is determined as the distance to a point, line, or plane. For rectangles (2D) and cubes (3D) Persson and Strang defined a simple approximation, which takes the minimum distance to all of the planes (lines in 2D). With this approximation also the tetrahedron (triangle in 2D) can be defined, see appendix A. In this appendix also the exact distance definition is included. When we compare the approximation with the exact distance we observe in Figure 4.14 that interior to the geometry the approximation coincides with the exact distance definition. However, exterior to the geometry the approximation of the distance can be up to 90% lower than the exact value. For cubes (and rectangles in 2D) this error is relatively low and confined to the corners (Fig 4.14(a)), but for triangles in 2D and tetrahedrals in 3D the error is significant spanning most of the exterior of the geometry (Fig 4.14(b)–(c)). Persson and Strang recognize this limitation for cubes, introducing fixed nodes at the corners. Following the observations on the distance scaling factor we emphasize the benefit of using the exact distance, especially of tetrahedrals.

In this chapter, the mesh algorithm of Persson and Strang was discussed. We have presented the benefits of this algorithm for the application with periodic structure defined by (implicit) level-set function, as well as for boundaries that have a periodic distribution of nodes. An approach to obtain a primitive cell with such a periodic distribution of nodes on the boundaries has been introduced. Furthermore, the influence of certain approximations in the description of the geometry on the mesh algorithm was discussed, whereby it was concluded that, especially in 3D, these approximations can prove influential. In practice, these approximations can prove critical especially for concave geometries whereby the geometry’s surface is characterized by large gradients.

In the previous everything was discussed that is needed to analyze the pattern transformation of three-dimensional periodic structure. In last chapter of this report a number of different geometries a meshed and then analyzed.
Figure 4.14: Relative difference between the approximate and exact distance.
Chapter 5

Results for different geometries

In chapter 2 we have introduced the problem of pattern transformation of periodic structures. We have defined the family of three-dimensional structures which are analyzed in the is section. To this end we apply the Bloch analysis that is introduced in chapter 3 to the primitive cell of the structure. In all instances we first calculate the relevant load at which loss of ellipticity occurs. The Bloch analysis is then performed whereby this load is taken as a maximum. The primitive cell is meshed using the technique outlined in chapter 4. We check all combinations where the periodicity ranges from 1 to 4 in each direction.

Throughout this chapter, first a reference 3-D structure is defined which is comparable to the 2-D structure used as an example in the previous chapters. Then in sections 5.2–5.3 three different 3-D structures are investigated that result from a Simple, Body Centered, and Face Centered Cubic lattice. In the last section a structure is investigated for which it will be shown that the instability encountered does not result in a transformation of pattern.

The primitive cell for each structure is constructed by confining the structure in a unit cell. This unit cell is sketched in Figure 2.5(a) wherein we can see that a local Cartesian coordinate system (denoted by \(x, y, z\)) is defined in the center of the unit cell which aligns with the global coordinate system (denoted by \(1, 2, 3\)). Also, the unit cell is assumed cubic, characterized by length \(a\). Therefore, the cell ranges from \(x, y, z \in (-a/2, a/2)\).

5.1 Cylindrical voids

Through the problem definition and modeling description, in chapter 2.8 a two-dimensional periodic structure of holes in a matrix was taken as example. The primitive cell of this structure is found in Figure 2.3(a). In this section, a three-dimensional equivalent is considered, which is characterized by cylindrical cuts in all directions. The diameter of the cylinders is taken the same as that of the 2-D holes, i.e. \(9a/10\). The primitive cell of this structure is plotted in Fig. 5.1. Because of the three-dimensional nature, the porosity of this structure is 20% higher than the 60% porosity found for the 2-D structure.

For this structure we apply four different loading conditions: uni- and biaxial loading, by respectively prescribing one and two components of the deformation \(H\). Triaxial loading is applied in two ways: (i) constrained swelling, in which all components of \(H\) are set zero and a temperature difference is applied; and (ii) pressure loading of the inner surface, in which no components of \(H\) are prescribed. The latter load case is assumed to model the situation in which etching of the structure was not perfect, and material remains connecting the inner surfaces.

The analysis reveals that all loading conditions result in the same periodicity and the same critical mode, however the instability does not occur at the same equivalent strain. What is more is that the equivalence of periodicities if different for the different loading conditions. E.g.
for triaxial loading the $2 - 2 - 1$, $2 - 1 - 2$ and $1 - 2 - 2$ periodicities constitute to the same response. For uniaxial loading in 11-direction only the $2 - 1 - 1$ and $2 - 1 - 2$ periodicities are equivalent.

For each load case, the critical configuration is characterized by a periodicity of $2 - 2 - 1$ (or its equivalent). At all times, this microscopic instability occurs well before the loss of ellipticity. The critical eigenmode is depicted in Figure 5.2 next to the corresponding undeformed RVE. From this figure we can conclude that the instability corresponds to a transformation of pattern, in particular a two-dimensional transformation equivalent to that of the 2-D structure.

In the following we zoom in on the triaxial and uniaxial load cases. In particular, we analyze the stress–strain behavior both prior to and after the transformation of pattern. To this end, the RVE of Fig. 5.2(a) is perturbed with the eigenmode (Fig. 5.2(b)). Loading this RVE then reveals the complete stress–strain response. The results in Fig. 5.3–5.4 show that relevant component of the homogenized first Piola-Kirchoff stress $\bar{S}_{i j}$ normalized with the Young’s modulus $E$ as a function of the strain. For uniaxial loading, the structure is loaded in the 11-direction, and therefore the response in reported in this direction. For triaxial loading, the 22-component of the response is fully equivalent to the 11-component, we therefore only report the latter. Both load-cases reveal a superelastic response in the direction of the instability, in this case the 11-
Figure 5.3: Stress–strain response as a result of uniaxial loading in the 11-direction.

Figure 5.4: Stress–strain response as a result of triaxial loading, in particular constrained swelling.

direction. For triaxial loading, we moreover observe that in the direction perpendicular to the instability the response is linear as no instability occurs in this direction.

Compared to the equivalent 2-D structure, for which the result for uniaxial loading is presented in Fig. 3.11, the energy absorption (characterized by the area under the stress–strain curve) is larger for the 3-D structure. The difference is particularly present when the energy absorption is normalized by the weight of the structure. We can thus conclude that, although the instability for this structure is two-dimensional, the three-dimensional structure benefits in energy absorption to weight ratio.

## 5.2 Simple cubic lattice

The previous section motivates us exploit three-dimensional structures, because of its benefit over two-dimensional structures. In the following sections we exploit a number of different 3-D structures based on the Fourier-series. Each of these structures can be found in [20]. In the upcoming three sections cubic lattices are taken as an inspiration, thereafter more random 3-D structure are considered. If not reported otherwise we consider constrained swelling as load-case.

This section considers the structure that is inspired by the Simple Cubic lattice, such as schematically represented in Figure 5.5(a). This structure is characterized by the following function

\[
f(x, y, z) = \cos \left( \frac{2\pi x}{a} \right) + \cos \left( \frac{2\pi y}{a} \right) + \cos \left( \frac{2\pi z}{a} \right) - \frac{1}{2} \cos \left( \frac{2\pi x}{a} \right) \cos \left( \frac{2\pi y}{a} \right) \cos \left( \frac{2\pi z}{a} \right)
- \frac{1}{2} \cos \left( \frac{2\pi y}{a} \right) \cos \left( \frac{2\pi z}{a} \right) - \frac{1}{2} \cos \left( \frac{2\pi z}{a} \right) \cos \left( \frac{2\pi x}{a} \right) + t
\] (5.1)
CHAPTER 5. RESULTS FOR DIFFERENT GEOMETRIES

Figure 5.5: Schematical representation of the Simple Cubic (SC), Face Centered Cubic (FCC) and Body Centered Cubic (BCC) lattice.

Figure 5.6: Primitive cell of the structure defined by (5.1), which is inspired on the Simple Cubic lattice. The porosity of this structure is 90%.

The structure is then defined by

\[
\begin{align*}
f & \geq 0 \rightarrow \text{material} \\
f & < 0 \rightarrow \text{air}
\end{align*}
\]

Notice that this structure is symmetric about the axes of the Cartesian coordinate system. Numerically one finds that we obtain a porosity of 90% for \( t = -1.34 \).

The primitive cell for this structure is shown in Figure 5.6 wherein we observe that this structure mimics the inverse of the structure that was considered in the previous section. In the right figure, the cut is made according to the sketch of Figure 5.7(a) the coordinates of this cut are defined according to the local coordinate system located in the center of the primitive cell.

Not surprisingly, the critical mode of this structure is equivalent to that previously considered, i.e. a \( 2 - 2 - 1 \) periodicity is obtained shown if Figure 5.8. Again, we retrieve a two-dimensional transformation of pattern.
Figure 5.7: Definition of cut views. Notice how the coordinates are defined with the local coordinate system.

Figure 5.8: Undeformed RVE and reconstructed critical mode for the structure inspired by the SC lattice. The post-buckle structure is characterized by a periodicity of $2 - 2 - 1$. 
5.3 Face Centered Cubic lattice

Next, we consider a structure that is based on a Face Centered Cubic (FCC) lattice, which is illustrated in Fig. 5.5(b). Therefore we replace (5.1) by:

\[ f(x, y, z) = 4 \cos \left( \frac{2\pi x}{a} \right) \cos \left( \frac{2\pi y}{a} \right) \cos \left( \frac{2\pi z}{a} \right) + \cos \left( \frac{4\pi x}{a} \right) \cos \left( \frac{4\pi y}{a} \right) \cos \left( \frac{4\pi z}{a} \right) + \cos \left( \frac{4\pi y}{a} \right) \cos \left( \frac{4\pi z}{a} \right) \cos \left( \frac{4\pi x}{a} \right) + t \]  

(5.4)

Also this function is symmetric over all local coordinate axes. We set \( t = -0.67 \) to obtain a porosity of 80%. For this structure, it is impossible to define a connected structure with a higher porosity. The primitive cell of this structure is found in Fig. 5.9.

We retrieve a three-dimensional pattern transformation for this structure, which is characterized by the critical periodicity of \( 2 \times 2 \times 2 \). The three-dimensional nature of the pattern transformation becomes clear when we consider two different cuts in Fig. 5.10. In this figure, the cuts are made according to Fig. 5.7(b)–(c). The cuts of the undeformed RVE coincide because of the symmetry that characterizes the primitive cell. After the transformation of pattern, we observe a clear difference between the two directions, thus confirming the three-dimensional nature of the transformation of pattern.

5.4 Body Centered Cubic lattice

For the structure that is based on the Body Centered Cubic lattice in Figure 5.11 we replace (5.1) by:

\[ -f(x, y, z) = \cos \left( \frac{2\pi x}{a} \right) \cos \left( \frac{2\pi y}{a} \right) + \cos \left( \frac{2\pi y}{a} \right) \cos \left( \frac{2\pi z}{a} \right) \cos \left( \frac{2\pi x}{a} \right) + \cos \left( \frac{2\pi z}{a} \right) \cos \left( \frac{2\pi x}{a} \right) + t \]  

(5.5)

For \( t = 0.90 \) we find a porosity of 90%. The primitive cell of this geometry can be found in Fig. 5.11.

For this structure we find that the macroscopically instability occurs before a microscopic instability is encountered. This observed for uni-, bi- and triaxial loading.
5.4. BODY CENTERED CUBIC LATTICE

Figure 5.10: Cut of the undeformed RVE and reconstructed critical mode cut along $x, y = \pm a/10$ for the structure inspired by the FCC lattice. The post-buckle structure is characterized by a periodicity of $2-2-2$. Notice that the cuts of the undeformed RVE coincide for both directions because of the symmetry of the primitive cell.

Figure 5.11: Primitive cell of the structure defined by (5.5), which is inspired on the Body Centered Cubic lattice. The porosity of this structure is 90%.
CHAPTER 5. RESULTS FOR DIFFERENT GEOMETRIES

(a) primitive cell

(b) cut along $x = 0$

Figure 5.12: Primitive cell of the structure defined by (5.6). The porosity of this structure is 60%.

5.5 Other geometries

Except for the first, the different three-dimensional structures we have discussed so far are inspired on a lattice model. In the following we deviate from this assumption to investigate also different structures. To obtained the structure in Figure 5.12 we replace (5.1) by:

$$-f(x, y, z) = \cos \left( \frac{2\pi}{a} [-x + y + z] \right) + \cos \left( \frac{2\pi}{a} [x - y + z] \right)$$

$$+ \cos \left( \frac{2\pi}{a} [x + y - z] \right) + \cos \left( \frac{2\pi}{a} [x + y + z] \right) - t$$

whereby we find a porosity of 60% for $t = 0.25$. From the primitive cell we observe why the porosity is chosen low for this structure. Increasing the porosity would sharpen the edges separating the voids, thus making the structure impossible to mesh. This limitation occurs in fabrication of this structure.

When we perform a Bloch analysis for this structure we observe that for a periodicity of $1 - 1 - 2$ a zero eigenvalue is found before a microscopic analysis in encountered. However the curves of the eigenvalues against the loading – i.e. the temperature difference $\alpha \Delta T$ – are non-smooth. When we analyze the critical mode, plotted in Figure 5.13, we can conclude that the instability does not induce the buckling of the laminas separating the voids. Rather the deformation localizes in those areas. The type of analysis performed here is not suited for this type of configuration. We can thus conclude that no transformation of pattern is encountered for this structure, although it may have that appearance.

In this chapter we have considered a number of three-dimensional structures. Analysis of the stress–strain response revealed that a super-elastic response characterizes the pattern transformation of these structures. Furthermore, we have observed that a slender three-dimensional periodic structure does not necessarily reveal a microscopic instability corresponding to a transformation of pattern. Finally we observed that rather than the buckling of laminas, localization of deformation may occur. For the latter, the deformed structure may appear that have undergone a transformation of pattern. However, this phenomenon is beyond the scope of this report.
Figure 5.13: Undeformed RVE and reconstructed critical mode for the structure characterized by (5.6). Localization of deformation occurs, rather than the buckling of laminas.
Chapter 6

Conclusion and recommendations

In this report we have focused on pattern transformations in three-dimensional periodic structures. A popular fabrication technique for these structures at the micro- and nanometer length scale is interference lithography. Because of the nature of this technique, any structure is expressed as a Fourier series. Rather that defining arbitrary structures that would have to undergo such a conversion for fabrication, we have chosen the Fourier series as the definition of the structures. Many of these structures are considered in the book of Maldovan and Thomas [20], from which we have selected only a few.

In previous work (among others [3, 4]), the transformation of two-dimensional infinite periodic structures was investigated using the Finite Element method. In particular a microscopic analysis in the form of a Bloch analysis was performed on a primitive cell. These analyses were combined with the loss of ellipticity analysis to ensure that a macroscopic instability is not covered up by performing a microscopic analysis.

These analysis techniques require Finite Element meshes of the primitive cells, containing the periodic structures. Due to the nature of these analyses, the boundaries of these meshes are required to be periodic. In itself this requirement can proof challenging, however for the class of three-dimensional periodic structures we chose a new approach had to be consulted. Due to the nature of the definition of these structure – using implicit level-set function – existing commercial mesh algorithms were disqualified. The freely available mesh algorithm by Persson and Strang [28] was conveniently based on this definition of geometries. The stability of this algorithm was improved for the specific family of structures we consider. Furthermore it was extended with a capability to create periodic meshes for geometries that vary in the level of symmetry.

Comparison of an equivalent two- and three-dimensional structure revealed that the energy absorption to weight ratio can be increased by application of the latter category. Furthermore it was revealed that the periodicity of both coincide, therefore it is expected that the photonic and phononic capabilities remain unchanged.

The pattern transformation of a number of different three-dimensional periodic structure was investigated. This analysis revealed that depending on the structure both two- and three-dimensional pattern transformations can be encountered. We moreover observed that slenderness of the structure is no guarantee for the occurrence of pattern transformation. Next we also observed that localization of deformation may be favorable over the buckling of lamina, thus hindering pattern transformations.

Future work should be directed to the characterization of the stress–strain response of the different periodic structure for which pattern transformations are observed. Studies in this direction are expected to reveal an increase in energy absorption to weight ratio of three-dimensional structures compared to two-dimensional structures. It would be interesting to characterize the photonic and phononic behavior of these structures.
Appendix A

Level-set functions of polygons

For the mesh algorithm level-set functions are used. Where possible the level-sets should be equal to the distance to the geometry. This appendix devoted to defining such functions for non-smooth geometries, specifically the polygons that play a role in obtaining periodic meshes of unit cells. The geometries considered here have sharp corners. This has the effect that the distance from an arbitrary point to the geometry cannot be calculated using a single function. In contrast, a combination of cases are distinguished in which the distance is calculated as the distance to a point, or to a line. In three-dimensions, also the distance from to a plane appears relevant.

Below, the relevant part of vector calculus is briefly reviewed in section A.1. Then, subsequently specific two-dimensional rectangle and triangle are discussed (section A.2–A.3). Afterwards, the three-dimensional cube and tetrahedron are discussed in section A.4–A.5.

A.1 Vector calculus

The distance from a point to a another point, a line, or a plane are needed to calculate the distance to the relevant geometries. To this end we discuss these calculations originating from vector calculus. For the most part this section in generalized for three-dimensions, whereby examples from references [2,36] have served as a guideline. We notice on beforehand that the distance from a point to a plane in principle is only defined in three-dimensions, but can also be viewed upon as the distance between a point and a line in two-dimensions. For simplicity we will start by reviewing on the distance to a point, before considering the distance to a plane and a line. Throughout this section we will calculate the shortest distance from an arbitrary point \( P \).

The distance between two points the most elementary. Let us calculate the distance between \( P \) and \( P_1 \) with respective coordinates \( \vec{x} \) and \( \vec{x}_1 \). We can find the distance using the vector connecting both points (see Fig. A.1(a)):

\[
\vec{v} = \vec{x}_1 - \vec{x} \tag{A.1}
\]

The distance \( d \) is simply the length of this vector, hence

\[
d = || \vec{v} || = \sqrt{\vec{v} \cdot \vec{v}} \tag{A.2}
\]

The distance from a point to a plane will be considered next. We will limit ourselves to the three-dimensional case (Fig. A.1(c)), before projecting the method exemplified to two-dimensions. Let us describe the plane by the level-set function

\[
f(x, y, z) = Ax + By + Cz + D = 0 \tag{A.3}
\]
APPENDIX A. LEVEL-SET FUNCTIONS OF POLYGONS

By definition the plane normal
\[ \vec{n} = A\vec{e}_x + B\vec{e}_y + C\vec{e}_z \]  \hspace{1cm} (A.4)

We define \( \vec{x}_0 \) as the position vector of the arbitrary point \( P_0 = (x_0, y_0, z_0) \) on the given plane, \( \vec{x} \) is again the position vector of \( P = (x, y, z) \). The projection of the difference vector \( \vec{x} - \vec{x}_0 \) on the plane normal \( \vec{n} \) provides us with the distance
\[ d = \frac{|(\vec{x} - \vec{x}_0) \cdot \vec{n}|}{||\vec{n}||} \]  \hspace{1cm} (A.5)

This expression can be rewritten in terms of (A.3) by making use of the fact that \( P_0 \) lies on the plane:
\[ \vec{x}_0 \cdot \vec{n} = -D \]  \hspace{1cm} (A.6)

Substitution of (A.6), (A.4), and (A.3) into (A.5) yields
\[ d = \frac{|Ax + By + Cz + D|}{\sqrt{A^2 + B^2 + C^2}} \]  \hspace{1cm} (A.7)

Finally we focus on the distance between a point and a line. If limited to two-dimensions we distinguish two approaches, whereas in three-dimensions we only find one. The first approach, that is limited to two-dimensions, considers a line as the projection of a plane parallel to the third dimension. This plane is therefore characterized by the function
\[ f(x, y) = Ax + By + D = 0 \]  \hspace{1cm} (A.8)

which in three-dimensions indeed constitutes to the described plane. Since this function coincides with (A.3) for \( C = 0 \) we find the distance
\[ d = \frac{|Ax + By + D|}{\sqrt{A^2 + B^2}} \]  \hspace{1cm} (A.9)

The second approach is to view the line a vector between two points. This approach – generalized for three-dimensions – is illustrated by Figure A.1(b). We define the positions of

Figure A.1: The distance from an arbitrary point \( P \) to: a point (a), a line (b), or a plane (c).
the points $P_1$ and $P_2$ that span the line as $\vec{x}_1$ and $\vec{x}_2$ respectively. As can be concluded from the sketch, the distance

$$d = ||\vec{x}_1 - \vec{x}_2|| \sin \theta$$  \hfill (A.10)

We can make use of the general property of the cross-product:

$$||\vec{a} \times \vec{b}|| = ||\vec{a}|| ||\vec{b}|| \sin \theta$$  \hfill (A.11)

to redefine the distance in terms of the vector $\vec{v}$ between the points $P_1$ and $P_2$. The result reads:

$$d = \frac{||(\vec{x}_1 - \vec{x}) \times \vec{v}||}{||\vec{v}||}$$  \hfill (A.12)

The remainder of this appendix is devoted to the definition of the level-set functions for the specific non-smooth geometries we have defined.

### A.2 Distance to a rectangle

![Figure A.2: Rectangle with edges of equal length $a$. The functions of (A.13)–(A.16) are also indicated.](image)

This section, and the next, are restricted to two-dimensions. We consider a rectangle before considering a triangle (in the next section). Furthermore, a restriction is made on the dimensions of the geometries for the sake of simplicity. This restriction fits that of the construction of the unit-cells.

A rectangle is fully defined by the length and the width. We restrict ourselves to rectangles with edges of equal length $a$. We thus set the corners to $P_1 = (-a/2, -a/2)$, $P_2 = (a/2, -a/2)$, $P_3 = (a/2, a/2)$, $P_4 = (-a/2, a/2)$. We introduce four functions describing the boundaries:

\begin{align*}
  f_1(x, y) &= -x - a/2 = 0 \quad \text{(A.13)} \\
  f_2(x, y) &= x - a/2 = 0 \quad \text{(A.14)} \\
  f_3(x, y) &= -y - a/2 = 0 \quad \text{(A.15)} \\
  f_4(x, y) &= y - a/2 = 0 \quad \text{(A.16)}
\end{align*}

for which the signs coincides with that in Figure A.2. At this point we have a choice whether we want to calculate the distance exactly or approximate the distance as is suggested in [28]. We will first focus on the exact calculation, the approximation is a natural extension.
For the exact calculation of the distance we have to define a number of cases. For a rectangle this is exemplified in Figure A.2. From this illustration it becomes apparent that these cases form two groups: the distance to a point, and the distance to a line; both were discussed in the previous section. Each of these groups comprises four different cases that are discussed in detail below.

The distance from an arbitrary point \( P \) with coordinates \( \vec{x} \) is dependent on the region wherein \( P \) is positioned. This is determined by evaluating the functions (A.13)–(A.16). The details are presented below in pseudo-code. The notation is used is:

\[
\begin{align*}
d &= d(P \rightarrow f_i) \text{distance from } P \text{ to the line } f_i \\
d &= d(P \rightarrow P_i) \text{distance from } P \text{ to the point } P_i
\end{align*}
\]  

(A.17)  
(A.18)

Furthermore we use that \( f_i \) is the value of that function in \( P \). The pseudo-code then reads:

\[
\text{if } f_i \leq 0 \ \forall \ i = 1, 2, 3, 4 \quad d = -\min d(P \rightarrow f_i), \ i = 1, 2, 3, 4
\]

else if \( f_3 \leq 0 \ & \ f_4 \leq 0 \ & \ f_1 \geq 0 \quad d = d(P \rightarrow f_1) \)

else if \( f_3 \leq 0 \ & \ f_4 \leq 0 \ & \ f_2 \geq 0 \quad d = d(P \rightarrow f_2) \)

else if \( f_1 \leq 0 \ & \ f_2 \leq 0 \ & \ f_3 \geq 0 \quad d = d(P \rightarrow f_3) \)

else if \( f_1 \leq 0 \ & \ f_2 \leq 0 \ & \ f_4 \geq 0 \quad d = d(P \rightarrow f_4) \)

else if \( f_1 \geq 0 \ & \ f_3 \geq 0 \quad d = d(P \rightarrow P_1) \)

else if \( f_2 \geq 0 \ & \ f_3 \geq 0 \quad d = d(P \rightarrow P_2) \)

else if \( f_2 \geq 0 \ & \ f_4 \geq 0 \quad d = d(P \rightarrow P_3) \)

else if \( f_1 \geq 0 \ & \ f_4 \geq 0 \quad d = d(P \rightarrow P_4) \)

Notice that we have assigned a negative distance to one case. The point satisfying this case are position in the interior of the geometry. This definition coincide that in chapter 4.

The distance to the same rectangle can be approximated. To this end only the inside or outside region are distinguished, whereby the distance is calculated as the minimal distance to one of the four functions. In pseudo-code:

\[
\text{if } f_i \leq 0 \ \forall \ i = 1, 2, 3, 4 \quad d = -1 \cdot \min d(P \rightarrow f_i), \ i = 1, 2, 3, 4
\]

else \( d = \min d(P \rightarrow f_i), \ i = 1, 2, 3, 4 \)
A.3 Distance to a triangle

Figure A.3: Triangle resulting from cutting the rectangle of Fig. A.2 along the diagonals. The different functions of (A.13)–(A.16) and (A.19)–(A.21) are also indicated.

By argument that we use the triangle for the formation of unit-cell geometries, we require the triangle to have an angle that is perpendicular in addition we require the edges that coincide with the unit-cell to be of length $a$. This type if triangle – sketched in Figure A.3 – is one half of the rectangle discussed in the previous section cut along the diagonal.

Since we still restrict ourselves to two-dimensions, the type of cases is the same as for the rectangle. Since the triangle is formed by cutting the rectangle some definitions coincide.

We define the corners as $P_1 = (-a/2, -a/2)$, $P_2 = (a/2, -a/2)$, $P_3 = (a/2, a/2)$. The functions describing the boundaries and the cut-off point to distinct the cases are partly taken the same as for the rectangle (A.13)–(A.16), and partly formulated as follows:

$$f_5(x, y) = -x + y = 0 \quad \text{(A.19)}$$
$$f_6(x, y) = -x - y - a = 0 \quad \text{(A.20)}$$
$$f_7(x, y) = x + y - a = 0 \quad \text{(A.21)}$$

The exact distance in pseudo-code:

```plaintext
if $f_i \leq 0 \ \forall \ i = 3, 4, 5$  then $d = -\min d(P \rightarrow f_i)$, $i = 3, 4, 5$
else if $f_3 \leq 0 \ \& \ f_4 \leq 0 \ \& \ f_2 \geq 0$  then $d = d(P \rightarrow f_2)$
else if $f_3 \leq 0 \ \& \ f_2 \leq 0 \ \& \ f_3 \geq 0$  then $d = d(P \rightarrow f_3)$
else if $f_6 \leq 0 \ \& \ f_7 \leq 0 \ \& \ f_5 \geq 0$  then $d = d(P \rightarrow f_5)$
else if $f_1 \geq 0 \ \& \ f_6 \geq 0$  then $d = d(P \rightarrow P_1)$
else if $f_2 \geq 0 \ \& \ f_3 \geq 0$  then $d = d(P \rightarrow P_2)$
else if $f_2 \geq 0 \ \& \ f_7 \geq 0$  then $d = d(P \rightarrow P_3)$
```

We can also approximate this distance:

```plaintext
if $f_i \leq 0 \ \forall \ i = 3, 4, 5$  then $d = -1 \cdot \min d(P \rightarrow f_i)$, $i = 3, 4, 5$
else  then $d = \min d(P \rightarrow f_i)$, $i = 3, 4, 5$
```
A.4 Distance to a cube

In the previous section the restriction was set to two-dimensions. In the upcoming two sections we remove this restriction and calculate the distance to a cube and a tetrahedron. Obviously visualization of the different planes is much more limited than for two-dimensions.

In this section we consider a cube (see Fig. A.4) of equal edge length $a$. The corners can thus be defined as $P_1 = (-a/2, -a/2, -a/2)$, $P_2 = (-a/2, a/2, -a/2)$, $P_3 = (a/2, -a/2, -a/2)$, $P_4 = (-a/2, -a/2, a/2)$, $P_5 = (a/2, a/2, -a/2)$, $P_6 = (-a/2, a/2, a/2)$, $P_7 = (a/2, a/2, a/2)$, $P_8 = (a/2, -a/2, a/2)$. Furthermore, we define the following functions corresponding to each of the boundary planes:

\[
\begin{align*}
    f_1(x, y) &= -x - a/2 = 0 \\  
    f_2(x, y) &= -y - a/2 = 0 \\  
    f_3(x, y) &= -z - a/2 = 0 \\  
    f_4(x, y) &= x - a/2 = 0 \\  
    f_5(x, y) &= y - a/2 = 0 \\  
    f_6(x, y) &= z - a/2 = 0
\end{align*}
\]

Using these function we can write the exact distance as pseudo-code. In addition to the notation of (A.17) and (A.18) we use that

\[
d = d(P \to l_{ij}) \text{distance from } P \text{ to the line between the points } P_i \text{ and } P_j
\]
The pseudo-code:

if \( f_i \leq 0 \forall i = 1, 2, ..., 6 \) \( d = \min d(P \rightarrow f_i) \), \( i = 1, 2, ..., 6 \)
else if \( f_i \leq 0 \forall i = 2, 3, 5, 6 \) \& \( f_1 \geq 0 \) \( d = d(P \rightarrow f_1) \)
else if \( f_i \leq 0 \forall i = 1, 3, 4, 6 \) \& \( f_2 \geq 0 \) \( d = d(P \rightarrow f_2) \)
else if \( f_i \leq 0 \forall i = 1, 2, 4, 5 \) \& \( f_3 \geq 0 \) \( d = d(P \rightarrow f_3) \)
else if \( f_i \leq 0 \forall i = 2, 3, 5, 6 \) \& \( f_4 \geq 0 \) \( d = d(P \rightarrow f_4) \)
else if \( f_i \leq 0 \forall i = 1, 3, 4, 6 \) \& \( f_5 \geq 0 \) \( d = d(P \rightarrow f_5) \)
else if \( f_i \leq 0 \forall i = 1, 2, 4, 5 \) \& \( f_6 \geq 0 \) \( d = d(P \rightarrow f_6) \)
else if \( f_i \leq 0 \forall i = 2, 5 \) \& \( f_i \geq 0 \forall i = 1, 3 \) \( d = d(P \rightarrow l_{12}) \)
else if \( f_i \leq 0 \forall i = 1, 4 \) \& \( f_i \geq 0 \forall i = 3, 5 \) \( d = d(P \rightarrow l_{23}) \)
else if \( f_i \leq 0 \forall i = 2, 5 \) \& \( f_i \geq 0 \forall i = 3, 4 \) \( d = d(P \rightarrow l_{34}) \)
else if \( f_i \leq 0 \forall i = 1, 4 \) \& \( f_i \geq 0 \forall i = 2, 3 \) \( d = d(P \rightarrow l_{41}) \)
else if \( f_i \leq 0 \forall i = 2, 5 \) \& \( f_i \geq 0 \forall i = 1, 6 \) \( d = d(P \rightarrow l_{56}) \)
else if \( f_i \leq 0 \forall i = 1, 4 \) \& \( f_i \geq 0 \forall i = 6, 5 \) \( d = d(P \rightarrow l_{67}) \)
else if \( f_i \leq 0 \forall i = 2, 5 \) \& \( f_i \geq 0 \forall i = 6, 4 \) \( d = d(P \rightarrow l_{78}) \)
else if \( f_i \leq 0 \forall i = 1, 4 \) \& \( f_i \geq 0 \forall i = 2, 6 \) \( d = d(P \rightarrow l_{85}) \)
else if \( f_i \leq 0 \forall i = 3, 6 \) \& \( f_i \geq 0 \forall i = 1, 2 \) \( d = d(P \rightarrow l_{43}) \)
else if \( f_i \leq 0 \forall i = 3, 6 \) \& \( f_i \geq 0 \forall i = 1, 5 \) \( d = d(P \rightarrow l_{26}) \)
else if \( f_i \leq 0 \forall i = 3, 6 \) \& \( f_i \geq 0 \forall i = 4, 5 \) \( d = d(P \rightarrow l_{37}) \)
else if \( f_i \leq 0 \forall i = 3, 6 \) \& \( f_i \geq 0 \forall i = 2, 4 \) \( d = d(P \rightarrow l_{48}) \)
else if \( f_i \geq 0 \forall i = 1, 2, 3 \) \( d = d(P \rightarrow P_1) \)
else if \( f_i \geq 0 \forall i = 1, 3, 5 \) \( d = d(P \rightarrow P_2) \)
else if \( f_i \geq 0 \forall i = 3, 4, 5 \) \( d = d(P \rightarrow P_3) \)
else if \( f_i \geq 0 \forall i = 2, 3, 4 \) \( d = d(P \rightarrow P_4) \)
else if \( f_i \geq 0 \forall i = 1, 2, 6 \) \( d = d(P \rightarrow P_5) \)
else if \( f_i \geq 0 \forall i = 1, 5, 6 \) \( d = d(P \rightarrow P_6) \)
else if \( f_i \geq 0 \forall i = 4, 5, 6 \) \( d = d(P \rightarrow P_7) \)
else if \( f_i \geq 0 \forall i = 2, 4, 6 \) \( d = d(P \rightarrow P_8) \)

An approximation can be made \[28\]. In our framework the approximation is written:

if \( f_i \leq 0 \forall i = 1, 2, ..., 6 \) \( d = -1 \cdot \min d(P \rightarrow f_i) \), \( i = 1, 2, ..., 6 \)
else \( d = \min d(P \rightarrow f_i) \), \( i = 1, 2, ..., 6 \)
A.5 Distance to a tetrahedron

![Tetrahedron diagram](image)

Figure A.5: Tetrahedron that results from cutting the cube from Fig. A.4 along the diagonal planes parallel to each of the axes.

Finally we calculate the distance to a tetrahedron. The tetrahedron is limited in dimension such that it corresponds to the cube of the previous section cut along the diagonal planes parallel to each of the axes. As is illustrated in Figure A.5, the the corners are defined as $P_1 = (-a/2, -a/2, -a/2)$, $P_2 = (-a/2, a/2, -a/2)$, $P_3 = (a/2, a/2, -a/2)$, $P_4 = (a/2, a/2, a/2)$.

The different planes – perpendicular to each of the planes of the tetrahedron – needed to distinguish between all possible cases are:

\[
\begin{align*}
  f_1(x, y) &= -z - a/2 = 0 \
  f_2(x, y) &= x - y = 0 \
  f_3(x, y) &= y - a/2 = 0 \
  f_4(x, y) &= -x + z = 0 \
  f_5(x, y) &= -x - a/2 = 0 \
  f_6(x, y) &= x - a/2 = 0 \
  f_7(x, y) &= x + y - a = 0 \
  f_8(x, y) &= -x - y + 2z = 0 \
  f_9(x, y) &= -x - z - a = 0 \
  f_{10}(x, y) &= x - 2y + z = 0 \
  f_{11}(x, y) &= x + z - a = 0 \
  f_{12}(x, y) &= -y - a/2 = 0 \
  f_{13}(x, y) &= z - a/2 = 0 \
  f_{14}(x, y) &= x + y + 2z - 2a = 0 \
  f_{15}(x, y) &= -x + y - 2z - 2a = 0 \
  f_{16}(x, y) &= -x - y - a = 0
\end{align*}
\]
This results in the following exact distance calculation, in pseudo-code:

\[
\begin{align*}
\text{if } & f_1 \leq 0 \forall i = 1, 2, 3, 4 \quad d = -\min d(P \rightarrow f_i), \ i = 1, 2, 3, 4 \\
\text{else if } & f_1 \leq 0 \forall i = 2, 3, 5 \quad & d = d(P \rightarrow f_1) \\
\text{else if } & f_1 \leq 0 \forall i = 1, 7, 8 \quad & d = d(P \rightarrow f_2) \\
\text{else if } & f_1 \leq 0 \forall i = 1, 4, 6 \quad & d = d(P \rightarrow f_3) \\
\text{else if } & f_1 \leq 0 \forall i = 3, 9, 10 \quad & d = d(P \rightarrow f_4) \\
\text{else if } & f_1 \leq 0 \forall i = 3, 12 \quad & d = d(P \rightarrow l_{12}) \\
\text{else if } & f_1 \leq 0 \forall i = 5, 9 \quad & d = d(P \rightarrow l_{23}) \\
\text{else if } & f_1 \leq 0 \forall i = 1, 3 \quad & d = d(P \rightarrow l_{14}) \\
\text{else if } & f_1 \leq 0 \forall i = 2, 9, 11 \quad & d = d(P \rightarrow l_{14}) \\
\text{else if } & f_1 \leq 0 \forall i = 14, 15 \quad & d = d(P \rightarrow P_1) \\
\text{else if } & f_1 \leq 0 \forall i = 9, 11 \quad & d = d(P \rightarrow l_{24}) \\
\text{else if } & f_1 \leq 0 \forall i = 1, 13 \quad & d = d(P \rightarrow l_{34}) \\
\text{else if } & f_1 \leq 0 \forall i = 1, 13 \quad & d = d(P \rightarrow P_1) \\
\text{else if } & f_1 \geq 0 \forall i = 2, 5, 16 \quad & d = d(P \rightarrow P_1) \\
\text{else if } & f_1 \geq 0 \forall i = 3, 5, 9 \quad & d = d(P \rightarrow P_2) \\
\text{else if } & f_1 \geq 0 \forall i = 1, 6, 7 \quad & d = d(P \rightarrow P_3) \\
\text{else if } & f_1 \geq 0 \forall i = 11, 13 \quad & d = d(P \rightarrow P_4) \\
\text{else if } & f_1 \geq 0 \forall i = 2, 13, 14 \quad & d = d(P \rightarrow P_4)
\end{align*}
\]

In contrast, the approximation is simply

\[
\begin{align*}
\text{if } & f_1 \leq 0 \forall i = 1, 2, 3, 4 \quad d = -1 \cdot \min d(P \rightarrow f_i), \ i = 1, 2, 3, 4 \\
\text{else } & d = \min d(P \rightarrow f_i), \ i = 1, 2, 3, 4
\end{align*}
\]
Bibliography


