Title

Finite element modelling of defects in additively manufactured strut-based lattice structures

Authors

Ifeanyichukwu Echeta[1,2*], Ben Dutton[3], Richard K. Leach[1], Samanta Piano[1]

[1] Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, NG8 1BB, UK

[2] Centre for Additive Manufacturing, Faculty of Engineering, University of Nottingham, Nottingham NG8 1BB, UK

[3] The Manufacturing Technology Centre Ltd, Ansty Park, Coventry, CV7 9JU, UK

Abstract

Strut-based lattice structures produced by powder bed fusion are prone to characteristic manufacturing defects that alter both their form and surface texture. Most studies in the literature focus on a subset of commonly observed defects, typically radius variation and strut waviness; surface defects remain relatively unexplored. Furthermore, there remains a need for the development of a general finite element modelling framework that can implement a range of defects into any strut-based lattice design. This paper presents a modelling framework for implementing a range of both form and surface defects into finite element meshes of strut-based lattices. A signed distance function forms the foundation for this framework, upon which surface meshes can be modified and converted into tetrahedral meshes via open-source software. The paper demonstrates how radius variation, strut waviness, elliptical cross sections and localised surface defects can be modelled in lattice struts, for which intuitive mathematical definitions are provided. A parametric study is performed to assess the sensitivity of the compressive Young's modulus of BCCZ and octet-truss lattices to upskin and downskin surface defects. The results showed higher sensitivity in the octet-truss than in BCCZ; both designs were more sensitive to downskin than to upskin defects.

Keywords

- Additive manufacturing, manufacturing defects, signed distance function, finite element analysis

* Corresponding author: Ifeanyi.Echeta@nottingham.ac.uk

1 Introduction

1.1 Lattice structures and defects

Additive manufacturing (AM) is a popular area of research because it greatly expands the existing design space, allowing for significantly more geometric freedom than in more established manufacturing methods, such as machining, casting and forming. Within AM, a popular area of study is that of lattice structures: a unit cell tessellated in three axes. Lattice structures have many desirable properties, such as high specific strength, high surface area to volume ratio and high impact energy absorption properties [1]. The literature on lattice structures continues to increase, with efforts largely being split between increasing the documentation on the general mechanical properties of lattice designs (e.g. [2–4]) and developing lattice structures for specific applications, such as biomedical implants [5–7], heat exchangers [8,9] and sandwich structures for lightweighting of structural engineering components [10,11]. The utility of AM lattice structures extends to precision engineering applications, where vibration isolating lattice structures have been designed for some low frequency bandwidths [12,13]. Such designs could be incorporated into machine frames in an attempt to reduce the noise in a measurement system.

AM lattice structures are defined by their unit cell, the design of which generally falls into one of two categories: strut-based and surface-based. Strut-based unit cells consist of a network of cylindrical struts connected at nodes. Surface-based unit cells are mathematically defined as the surface connecting the set of points for which a given function has a constant value, that is, an isosurface. The study of strut-based designs dominates the literature, a likely reason being the high level of design control which they provide, as their unit cells are created by explicitly specifying the shape and location of all the features; conversely surface-based designs are governed by a single mathematical equation.

Within AM, metal powder bed fusion (PBF) is commonly used to manufacture lattice structures [14]. There are characteristic defects (i.e. deviations from nominal) that form in lattice structures produced by PBF; these defects being a natural by-product of the layer-wise, powder based process [14]. Defects can be considered as altering both the form and surface texture of lattice structures. For example, form defects include feature misalignments and thickness variations in struts and surfaces. An example of a surface defect is the increased irregularity of the surface of the underside of overhanging features (hereafter described as texture bias). Figure 1.1a shows a common case of feature misalignment called "strut waviness", where the strut's axis deviates from its originally linear design. An example of thickness variations in the radius of struts is shown in Figure 1.1b. Figure 1.1c shows texture bias, where the underside of the struts ("downskin" surface) contains more irregularities than the upper side ("upskin" surface).

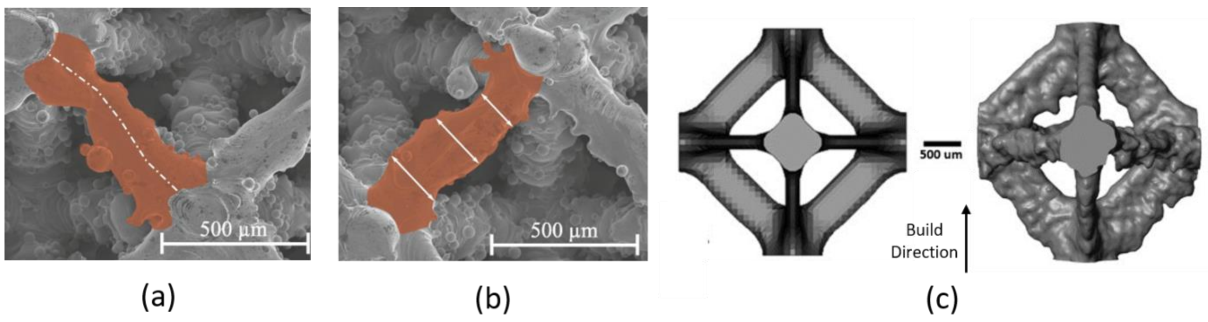


Figure 1.1 – Common defects in lattice structurers. (a) strut waviness [15] *(b) radius variation* [15] *(c) Left: original unit cell design, Right: texture bias in the manufactured part* [16].

1.2 Finite element modelling for lattice structures

Manufacturing defects can have a significant impact on the mechanical properties of lattice structures, with fatigue properties being particularly adversely affected [17–20]. Therefore, it is vital that these defects are included in the finite element (FE) models which are used to simulate lattice mechanical properties. Several FE studies have shown the impact of defects on mechanical properties. For example, Melancon et al. [15] predicted the compressive Young's modulus of the octet-truss lattice and found the error between simulations and experiments to decrease from 57% to 12% upon incorporating defects into the model. Similarly, Karamooz Ravari et al. [21] found the maximum error between simulated and experimental stress-strain response of BCC lattices to reduce from 53% to 27% upon incorporating defects into the FE model. Both [15] and [21] modelled strut waviness and thickness variations.

Defect modelling has only been studied in strut-based lattice structures (to the best of the authors' knowledge at the time of writing this paper) and the methods can be grouped into two categories, based on element type: 3D continuum element modelling and beam element modelling. 3D continuum element methods generally require the development of a CAD model into which defects are applied before conversion to an FE mesh [15,21,22]. Beam elements are also often used to model struts in lattice structures, where defects are modelled by dividing each strut into several beams and varying the location and radius of each element [23–26].

A particularly powerful aspect of FE modelling is the ability to perform parametric studies wherein specific defects can be controlled and analysed in depth. Parametric studies for defects in lattice structures can help quantify the unique impact of a specific defect (or combination of defects) on a given lattice's function. Several parametric studies have been performed for selected defects in lattice structures. Both Liu et al. [23] and Cao et al. [24] have used beam element models to isolate strut waviness and radius variation and investigate their impact on Young's modulus and yield strength under compression. In addition to isolating strut waviness and radius variation, El Elmi et al. [27] investigated the sensitivity of the compressive Young's modulus of octet-truss lattices to mass agglomeration and variations in the local stiffness of the lattice nodes and struts.

1.3 Relevance of this work

The impact of manufacturing defects on lattice structures will vary significantly across the large number of potential designs. Furthermore, whether to consider the impact of a defect as critical or negligible will depend on the constraints specific to a given application. Therefore, developing an FE modelling approach which accommodates parametric studies of a wide range of lattice defects would be a valuable tool. Such a tool would contribute towards the development of "defect-tolerant" lattice structures, where defect sensitivity studies could feed an optimisation process to produce lattice structures that perform reliably in the presence of defects.

This proposed method also has its advantages over image-based approaches, typically the conversion of X-ray computed tomography (XCT) data into an FE mesh (e.g. [28,29]). Although an image-based method arguably provides the most accurate representation of the measured lattice structure, the defects are defined implicitly within the XCT data and cannot be easily modified - this is not suitable for parametric studies. Furthermore, the process of acquiring XCT data – from sample manufacture through to data acquisition and FE mesh conversion – is a highly time-consuming process and not favourable for a high number of simulations.

Although the aforementioned parametric studies (Section 1.2) provide useful insights, they focus mainly on modelling only strut waviness and radius variation – due to the use of beam elements which can model a very limited range of geometries. It would be useful to also study other defects in equal proportion – for example mass agglomeration [27]; also texture bias remains unexplored. Furthermore, in the literature there is little discussion towards developing a generalised modelling approach for implementing the wide range of defects into any strut-based lattice structure.

In this paper we present a defect modelling approach that can apply both form and surface defects into strut-based lattice structures. This modelling approach uses signed distance functions to create triangulated surface meshes of lattice geometries. The surface mesh can then be modified to apply localised surface defects. The surface mesh is converted into an FE mesh of linear tetrahedral elements via open-source MATLAB meshing toolbox, Iso2mesh. This signed distance function approach allows geometries to be easily adapted to any strut-based lattice structure. In this method, all lattice defects are defined using mathematical functions, which provides an intuitive and efficient approach for generating lattice FE meshes with defects – this is also preferable over the use of graphical user interfaces in traditional CAD software.

This work aims to provide a framework for efficient generation of tetrahedral meshes of lattice structures suitable for parametric studies of a wide range of manufacturing defects. Although the authors' main objective is to support the development of machine frames [12,13], this framework can be applied to many FE strut-based lattice studies.

1.4 Structure of paper

The structure of the paper is as follows: Section 2 defines signed distance functions and demonstrates how two-dimensional geometries are extracted using this method; Section 3 extends signed distance functions into three dimensions and explains how lattice struts are modelled without defects; Section 4 gives the mathematical definitions for modelling waviness, radius variation, elliptical cross sections and texture bias in lattice struts; Section 5 describes how the tetrahedral meshes are generated and optimised; Section 6 provides a summary of how this modelling framework is implemented; Section 7 provides a brief FE study to serve as an example of applying the modelling framework; Section 8 then provides discussion and conclusion for the work in the paper.

2 Signed distance functions

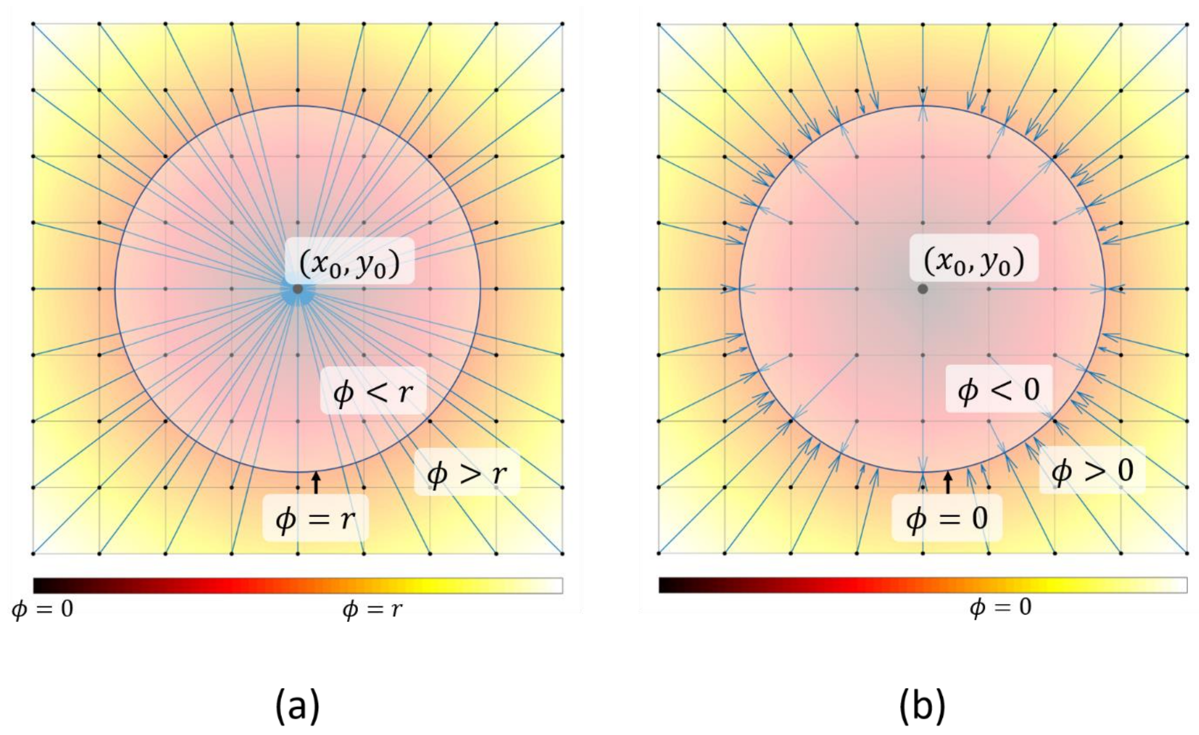


Figure 2.1 – Diagrams explaining the basic operation of signed distance functions. Left: Calculation of Euclidean distance between all points in the domain and (x_0, y_0), the circle of radius r is found at $\phi = r$. Right: Subtraction of constant r, the circle of radius r is found at $\phi = 0$.

Signed distance functions (SDFs) have already been identified as a means of modelling lattice structures [30,31] – this section will explain their operation.

SDFs operate by calculating the Euclidean distance between a given point in space and a predefined boundary. The sign of the distance denotes whether the point lies inside or outside of the predefined boundary, where common convention uses negative and positive signs, respectively. Thus, SDFs produce scalar fields, more specifically distance fields, inside which the predefined boundary is represented by the zero level set. For a simple example, we demonstrate how SDFs are used to model a circle. Consider the SDF $\phi(\vec{x})$ which calculates the Euclidean distance between the point (x_0, y_0) and all (x, y) points in the domain,

$$\phi(\vec{x}) = \sqrt{(x - x_0)^2 + (y - y_0)^2}. \qquad 2.1$$

As shown in Figure 2.1a, the arrows at each (x, y) point in the domain represent the Euclidean distance to the point (x_0, y_0). The resulting distance field has radially increasing values from the origin. Therefore, the circle of radius r is found at $\phi(\vec{x}) = r$, the rth level set. $\phi(\vec{x}) = r$ yields the set of points at an equal distance of r from (x_0, y_0). $\phi(\vec{x}) = r$ is the boundary between two regions: one where $\phi(\vec{x}) < r$ and $\phi(\vec{x}) > r$, also shown in Figure 2.1a. Remembering the aforementioned convention where the zero level set is used to create the geometry of interest, the distance field must be manipulated such that the circle of radius r is the boundary between the regions where $\phi(\vec{x}) < 0$ and $\phi(\vec{x}) > 0$. In this example, the manipulation is straightforward - the SDF is altered as follows:

$$\phi(\vec{x}) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - r = 0. \tag{2.2}$$

The subtraction of the constant r causes each value in the distance field to represent the Euclidean distance between the (x, y) point and the circle of radius r. This is illustrated in Figure 2.1b, where the arrows within the circle represent negative distance values and are thus facing the opposite direction to the arrows outside of the circle. The circle of radius r is now the interface between positive and negative regions, $\phi(\vec{x}) = 0$ as required.

Simplifying eq 2.2 into two terms helps to illustrate the general operation of SDFs,

$$\phi(\vec{x}) = d - r = 0. \tag{2.3}$$

SDFs can be considered as having two stages of operation. The first stage – boundary definition (d) – creates a distance field by calculating the Euclidean distances to a predefined boundary. The second stage – distance field manipulation (r) – is where each value in the distance field is modified, creating positive and negative regions, such that the interface between those regions yields whichever geometry is desired. In the previous example, the boundary was defined as the point (x_0, y_0); the manipulation of the distance field was a subtraction of r from all values such that the desired circle is found at $\phi(\vec{x}) = 0$.

Exploring the stages of boundary definition and distance field manipulation allows the geometric complexity to be increased. For another simple example, the manipulation stage in eq. 2.2 can be modified to

$$\phi(\vec{x}) = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r_i = 0 \tag{2.4}$$

where

$$r_i = f(\theta). \tag{2.5}$$

Figure 2.2 shows an example of a more complex shape, where the SDF for a circle has been modified to vary the radius as a function of θ, where the radius increases linearly with θ. Note the addition of the subscript i, indicating that eq 2.4 is computed iteratively across the domain. Again, the shape is represented by the boundary $\phi(\vec{x}) = 0$.

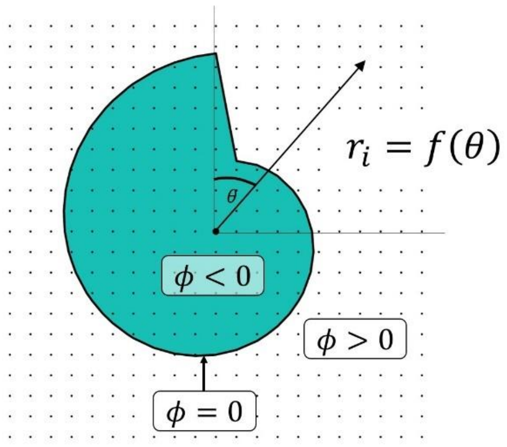


Figure 2.2 – An example of a more complex signed distance function where the radius $r_i = f(\theta)$. The boundary of the shape is still defined as $\phi = 0$. The points inside and outside of the boundary are defined as $\phi < 0$ and $\phi > 0$, respectively.

Additionally, combining multiple distance fields can increase geometric complexity. SDFs lend themselves to Boolean operations (unions, additions, subtractions, etc.) which are used in constructive solid geometry tools. For example, we generate two shapes, ϕ_1 and ϕ_2, based on the SDF defined in eq 2.4 and produce the union of these two shapes by computing $\min(\phi_1, \phi_2)$, as shown in Figure 2.3. Since negative distance values are used to define the points bounded within the geometry, computing the minimum gives precedence to the negative values and so the result combines the geometries from the two input distance fields.

Geometries are extracted from the distance field by calculating the coordinates where the SDF $\phi(\vec{x}) = 0$. This calculation is performed by linear interpolation between adjacent points where a sign change is detected in the distance field. Naturally, a domain of higher resolution is desirable as it reduces the distance over which the interpolation is computed and thus produces a more accurate geometry.

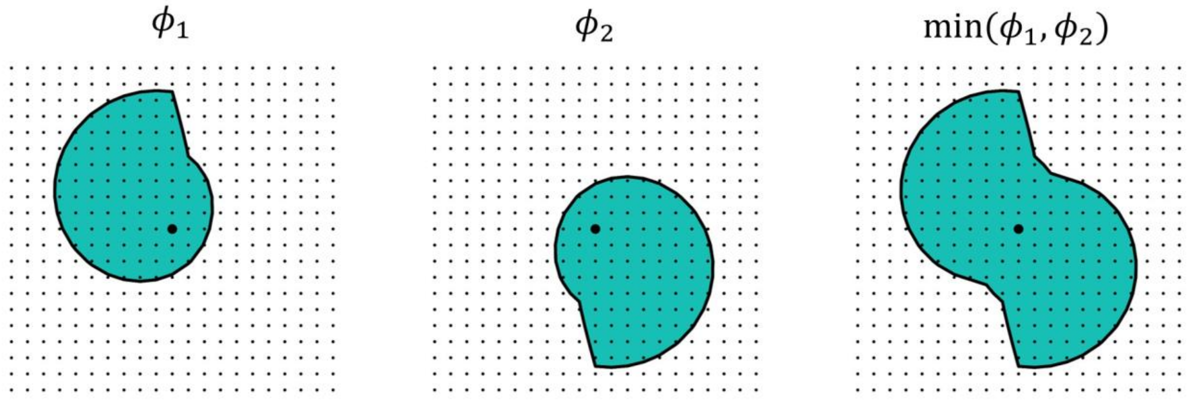


Figure 2.3 – Two distance fields (ϕ_1, ϕ_2) are combined by computing $min\ (\phi_1, \phi_2)$ which performs a Boolean union.

3 SDFs for ideal lattice structures

The surface of a lattice strut is defined by the set of points at a distance r from the strut's medial axis – a line segment, as shown in Figure 3.1a. This surface can be modelled as the zero level set of the SDF that calculates the Euclidean distance between the line segment and each point in a 3D Cartesian grid. The SDF, in this case, is an extension of eq 2.2, where the boundary definition is modified from a single point to a line segment. As shown in Figure 3.1a, the similarity to eq 2.2 can be visualised in the cross-section of the strut, where $\phi < 0$ inside the circle and $\phi > 0$ outside. The SDF $\phi(\vec{x})$ for the strut is given by

$$\phi(\vec{x}) = |\vec{x}_i - \vec{v}_i| - r = 0 \qquad i = 1,2,\ldots,M \tag{3.1}$$

where M is the total number of points in the grid. All arrow notation (e.g. $\vec{v}_i$) denotes coordinates in three axes, for example

$$\vec{v}_i = (x_{\vec{v}_i}, y_{\vec{v}_i}, z_{\vec{v}_i}). \tag{3.2}$$

To further explain the SDF, consider an arbitrary point $\vec{x}_i$ in the Cartesian grid. The Euclidean distance between $\vec{x}_i$ and the line segment $(\vec{l}_2 - \vec{l}_1)$ is given by $|\vec{x}_i - \vec{v}_i|$, where $\vec{v}_i$ is the point on the line segment closest to $\vec{x}_i$. $\vec{v}_i$ is calculated using

$$\vec{v}_i = [\vec{l}_1 + (\vec{l}_2 - \vec{l}_1)t_i] \tag{3.3}$$

and t_i is termed as the intersection ratio. The intersection ratio describes where $\vec{v}_i$ is located within the line segment, as a ratio of the vector $(\vec{l}_2 - \vec{l}_1)$. Thus, $0 \leq t_i \leq 1$ for all points on the line segment. The intersection ratio is calculated using

$$t_i = \frac{(\vec{l}_2 - \vec{x}_i)\cdot(\vec{l}_2 - \vec{l}_1)}{|\vec{l}_2 - \vec{l}_1|^2}. \tag{3.4}$$

In eq 3.4 $\vec{l}_2$ and $\vec{l}_1$ define a line of infinite length, therefore, there are some iterations of $\vec{x}_i$ where $t_i > 1$ or $t_i < 0$, as shown in Figure 3.1b. Such cases identify when the point $\vec{v}_i$ extends beyond the line segment – to prevent this, the following condition is added

$$t_i = \begin{cases} 0, & \text{for} \quad t_i < 0 \\ 1, & \text{for} \quad t_i > 1 \end{cases}. \tag{3.5}$$

Once the distance between a given point and the line segment has been calculated, r is subtracted in order for the desired surface to be represented by $\phi = 0$.

The presented modelling approach for individual struts can be extended to unit cells and full lattice structures. One of the most direct approaches would be to compute the SDF over all the line segments in the geometry. However, if appropriate for an application, the distance field of one strut can be duplicated and recombined to build unit cells and full lattice structures. As shown in Figure 3.2, the BCC unit cell is modelled by computing the minimum of four distance fields, each of which represents an individual strut and is a rotated duplicate of the other. In this paper, lattice structures are then modelled by duplicating the unit cell and concatenating the distance fields. Note that the distance fields must be cropped at the points at which they should overlap, as illustrated in Figure 3.3.

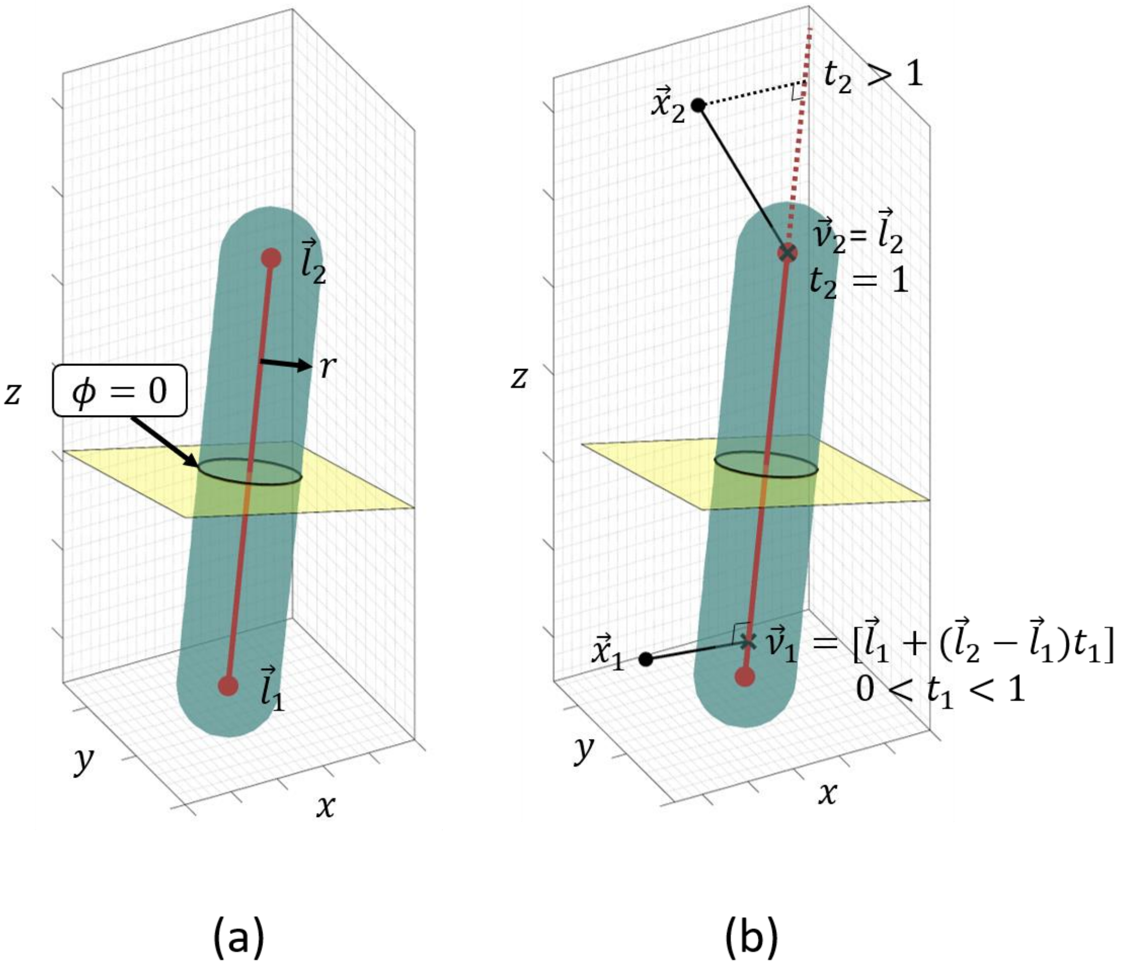


Figure 3.1 – (a) illustration of a lattice strut modelled as the points at distance r from the line segment. (b) illustration of how the intersection ratio is used in the distance calculation. At $\vec{x}_1$, $\vec{v}_1$ is located in between $\vec{l}_2$ and $\vec{l}_1$ and therefore $0 < t_1 < 1$. At $\vec{x}_2$, a case is shown where the Euclidean distance would naturally cause $\vec{v}_2$ to extend beyond the line segment, t_2 is therefore adjusted.

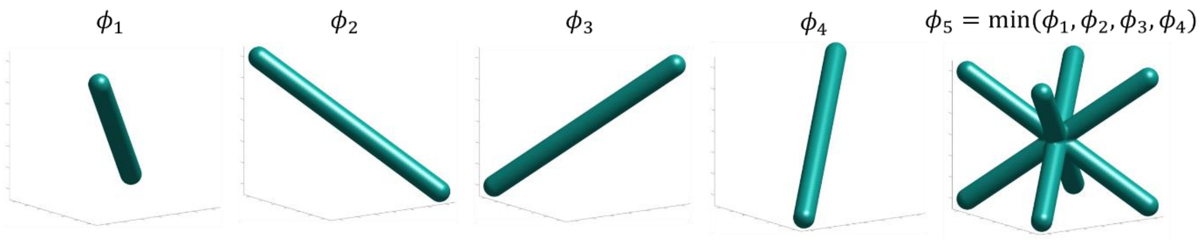


Figure 3.2 – Boolean union performed to create BCC unit cell. The distance for the first strut (ϕ_1) is duplicated and rotated to create three additional struts, ϕ_2, ϕ_3, ϕ_4. The unit cell (ϕ_5) is created by computing $min\ (\phi_1, \phi_2, \phi_3, \phi_4)$.

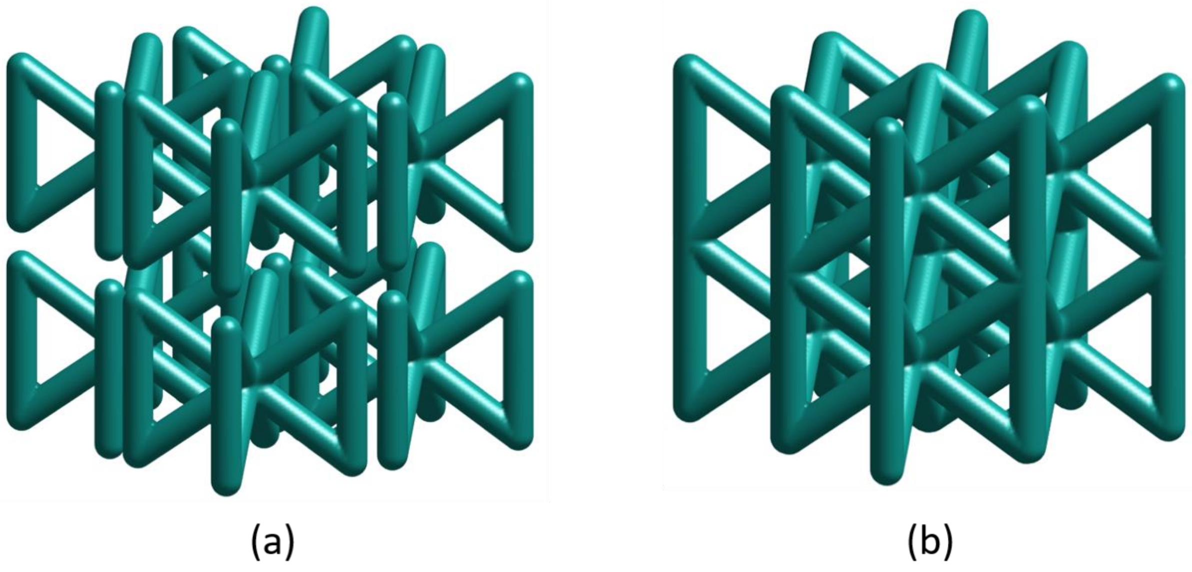


Figure 3.3 – (a) the distance field of the unit cell is duplicated and (b) appropriate gaps are cropped.

4 SDFs for defects

We now consider how to extend SDFs to model defects in lattice structures. Firstly, form defects (waviness, radius variation and elliptical cross sections) are considered, followed by localised surface defects (texture bias).

4.1 *Waviness*

To model waviness, the distance calculation and manipulation remain similar to that required for eq 3.1, however, the boundary definition is modified. As shown in Figure 4.1, the line segment is now partitioned, creating additional vertices used to modify the strut's medial axis. The vertices of the line segments are now defined as

$$\vec{l}_j = \left(x_{\vec{l}_j}, y_{\vec{l}_j}, z_{\vec{l}_j}\right) \qquad j = 1,2,\dots,N+1 \tag{4.1}$$

where N is the number of line segments. The line segments are stored in the matrix $\boldsymbol{L}$, where

$$\boldsymbol{L} = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_N \end{bmatrix} = \begin{bmatrix} \vec{l}_2 - \vec{l}_1 \\ \vec{l}_3 - \vec{l}_2 \\ \vdots \\ \vec{l}_{N+1} - \vec{l}_N \end{bmatrix}. \tag{4.2}$$

The SDF must now compute the Euclidean distance between a given point and the closest line segment in $\boldsymbol{L}$. This SDF is given by

$$\phi(\vec{x}) = \min\left(\boldsymbol{D}_{\vec{x}_i}\right) - r = 0 \tag{4.3}$$

where $\min\left(\boldsymbol{D}_{\vec{x}_i}\right)$ is the minimum of the distances between a given point $\vec{x}_i$ and all the line segments in $\boldsymbol{L}$. To clarify, consider again an arbitrary point $\vec{x}_i$ in the Cartesian grid (Figure 4.1). We calculate the distance between $\vec{x}_i$ and all the line segments, which is stored in the vector

$$\boldsymbol{D}_{\vec{x}_i} = [d_1, d_2, \dots, d_N]. \tag{4.4}$$

Computing the $\min\left(\boldsymbol{D}_{\vec{x}_i}\right)$ gives the Euclidean distance to the closest line segment. Once this distance is found, a subtraction of r is applied, for the same reasons as previously described in Section 2.

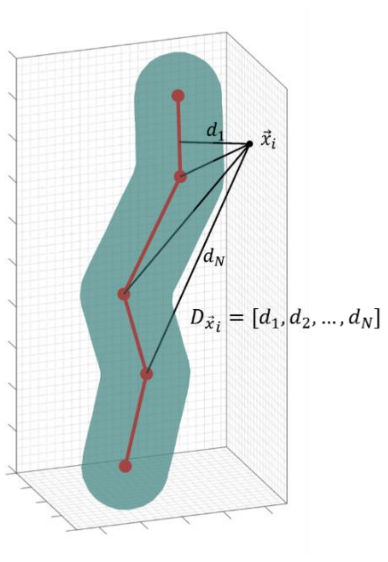


Figure 4.1 – Illustration of the signed distance function used for modelling waviness. The strut's medial axis has been modified.

4.2 Radius variation

To apply radius variation, the SDF will be similar to eq 4.3, but the r term must be modified; in other words, the manipulation stage of the SDF must be changed. We require the radius of the strut to vary along the strut's medial axis. Therefore, eq 4.3 is modified to

$$\phi(\vec{x}) = \min\left(\boldsymbol{D}_{\vec{x}_i}\right) - r_i = 0 \tag{4.5}$$

where

$$r_i = f(t_i). \tag{4.6}$$

The value of r_i depends on the intersection ratio of the line segment closest to $\vec{x}_i$. Continuing with the notation from Section 4.1, we create a vector $\boldsymbol{R}$ that describes the radius variation in the strut by assigning a radius value to each vertex $\vec{l}_j$ of the line segments

$$\boldsymbol{R} = [R_1, R_2, \dots, R_{N+1}]. \tag{4.7}$$

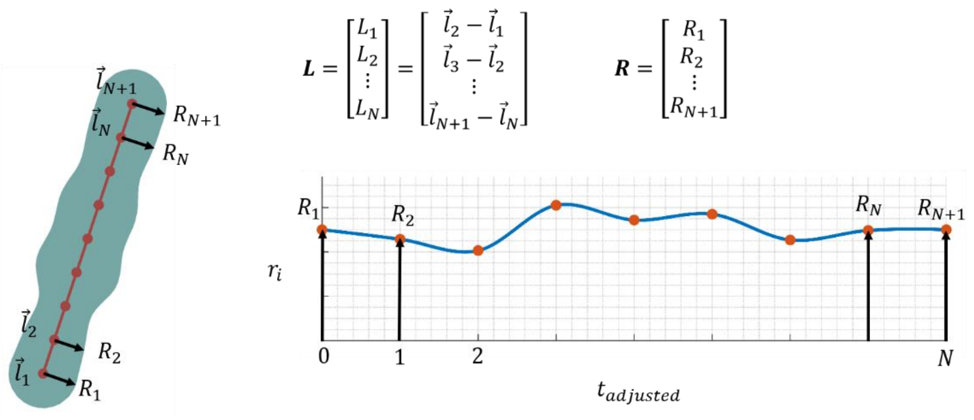
For every point $\vec{x}_i$ in the domain, we find the closest line segment L_j and apply the following condition

$$r_i = \begin{cases} R_j, & if\ t_i = 0 \\ R_{j+1}, & if\ t_i = 1 \\ \text{interpolate within } \boldsymbol{R}, & if\ 0 < t_i < 1 \end{cases}. \tag{4.8}$$

Recalling the definition of $\vec{v}_i$ from eq 3.3, if $\vec{v}_i$ is coincident with any of the vertices $\vec{l}_j$, then either $t_i = 0$ or $t_i = 1$ will be true and thus the r_i value will equal the value in $\boldsymbol{R}$ assigned to $\vec{l}_j$. In all other cases, we interpolate within $\boldsymbol{R}$. An example of eq. 4.8 is shown in Figure 4.2, where cubic interpolation is used. Lastly, because $0 \le t_i \le 1$ for all $\vec{x}_i$, there is currently no way of identifying the line segment to which t_i is associated. Therefore, the following condition is added

$$t_{adjusted} = j - 1 + t_i \tag{4.9}$$

where j is the line segment closest to $\vec{x}_i$. $t_{adjusted}$ is used in eq 4.8 instead of t_i, as shown in Figure 4.2.



$\vec{l}_{N+1}$
$\vec{l}_N$
R_{N+1}
R_N
$\vec{l}_2$
$\vec{l}_1$
R_2
R_1
$\boldsymbol{L} = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_N \end{bmatrix} = \begin{bmatrix} \vec{l}_2 - \vec{l}_1 \\ \vec{l}_3 - \vec{l}_2 \\ \vdots \\ \vec{l}_{N+1} - \vec{l}_N \end{bmatrix}$
$\boldsymbol{R} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{N+1} \end{bmatrix}$
R_1
R_2
R_N
R_{N+1}
r_i
0
1
2
N
$t_{adjusted}$

Figure 4.2 – Illustration of how radius variation is applied to the signed distance function. Left: Radius values are assigned to the vertices of the line segments. Right: An example of $r_i = f(t_i)$.

Although not a defect, the equations in this section could be used to apply a fillet between struts and nodes. Eq 4.6 could be defined such that the radii of the struts increase near lattice nodes. Applying fillets can improve the fatigue strength of lattice structures [32].

4.3 *Elliptical cross sections*

The versatility of SDFs can be further demonstrated by adapting them to model struts using elliptical cross sections (Figure 4.3a); the need for doing so being first identified by Lozanovski et al [22]. An ellipse with major axis a can be defined as the set of points which satisfy the following equation

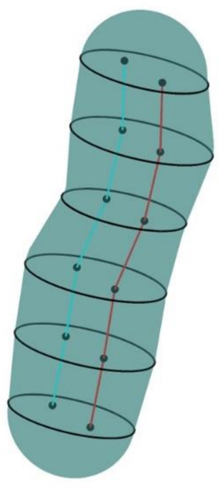
$$r_1 + r_2 = a \tag{4.10}$$

where r_1 and r_2 are the distances between the two fixed points in Figure 4.3b. Eq 4.10 is used to build an SDF which models struts with elliptical cross sections:

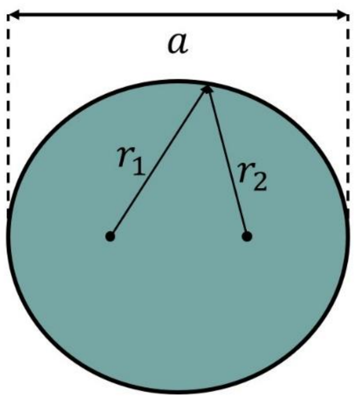
$$\phi(\vec{x}) = \min\left(\boldsymbol{D_1}_{\vec{x}_i}\right) + \min\left(\boldsymbol{D_2}_{\vec{x}_i}\right) - a_i = 0. \tag{4.11}$$

Note the similarities between eq 4.10 and eq 4.11. For every point in the domain $\vec{x}_i$, the distance between all the line segments in the two unconnected paths (blue and red lines) is determined and stored in $\boldsymbol{D_1}_{\vec{x}_i}$ and $\boldsymbol{D_2}_{\vec{x}_i}$. Computing the minimum of $\boldsymbol{D_1}_{\vec{x}_i}$ and $\boldsymbol{D_2}_{\vec{x}_i}$ gives the Euclidean distance between $\vec{x}_i$ and the closest line segment in each path - corresponding to r_1 and r_2 in eq 4.10.

Combinations of defects can also be applied to these struts. For example, the positions of the line segments can be varied to model waviness in the struts (as shown in Figure 4.3a) and the value of a_i can be varied in a similar way to the method described for r_i in Section 4.2, allowing the major axis of the ellipses to vary along the strut.



(a)



(b)

Figure 4.3 – (a) Modelling struts with elliptical cross sections. (b) A diagram of an ellipse with major axis a.

4.4 Texture bias

Surface defects can be applied to the model by modifying the surface mesh produced from the SDF. A triangulated surface of the strut geometry is extracted from the distance field using the MATLAB function *isosurface.m*; this function uses the interpolation method described in Section 2. To model localised surface defects, displacements can be applied to specific points on the surface that meet a given criteria.

Figure 4.4 shows a triangulated surface of a BCCZ unit cell where displacements have been applied to the downskin surface of the inclined struts. To apply these displacements, two characteristics must be defined for each point on the surface – the overhang angle of the strut and the up/downskin nature of the point; these characteristics are defined as θ and α respectively, as illustrated in Figure 4.5. To calculate the overhang angle θ of a strut, we find the angle between the line segment and the positive z-axis (Figure 4.5a). Note that in cases where the strut has waviness, the waviness is ignored for the calculation of θ. To characterise a point as up/downskin, we calculate the angle α between the surface normal and the reference vector shown in Figure 4.5b - this reference vector can be considered as pointing in the upskin direction. Each point on the surface can, therefore, be calculated as up/downskin using the following angular ranges:

$$Upskin: 0° \leq \alpha \leq 90° \qquad 4.12$$

$$Downskin: 90° < \alpha \leq 180°$$

where α is calculated clockwise from the reference vector for surface normals on the right, and vice versa – thus α does not exceed 180°. After characterising each point, localised surface defects can be modelled by building a function that applies displacements δ only to the points on the surface that meet a given criteria defined using θ and α.

Two methods for applying displacements have been considered, as shown in Figure 4.6. Firstly, pseudorandom displacements (Figure 4.6a-b). A displacement δ is applied to each point on the surface, where

$$\delta = \mu \pm \sigma. \qquad 4.13$$

The displacement is randomly selected from a normal distribution with a mean $\mu = 0$ and standard deviation σ, where

$$\sigma = f(\alpha, \theta). \qquad 4.14$$

An example of eq. 4.14 is shown in Figure 4.6b; this function sets $\sigma = 0$ for all vertical struts ($\theta = 0$) and causes σ to increase linearly with α. Note that the example BCCZ unit cell contains struts with two overhang angles: 0° and ~55°. This approach is useful for fast generation of texture bias in strut surfaces, as eq. 4.14 allows for a simple definition of the surface texture, without having to define any specific displacements on the surface – the values are selected randomly. One drawback, however, is that having no control over the exact values of the displacements can result in highly different displacements being applied to adjacent points – this can adversely affect the quality of the surface and present tetrahedral meshing issues. Note that the likelihood of this problem occurring reduces at lower σ values and can become negligible.

The second approach for applying displacements uses a function to predefine all the displacement values (Figure 4.6c-d). Predefining the displacements provides greater control and allows for texture bias which is locally smooth, thus preventing large changes in adjacent points. The predefined approach is a two-step process. Firstly, a simulated surface (ψ) is generated to describe the general distribution of the texture bias in a given strut, where

$$\psi = f(\alpha, t) \tag{4.15}$$

where t is the intersection ratio, as defined in Section 3 (note that t is not required in eq 4.14 because the random selection inherently applies variation along the strut's length). An example of ψ in shown in Figure 4.6c, where

$$\psi = f(\alpha, t) = B \sin(\omega_\alpha \alpha) \sin(\omega_t t) \exp\left(-\frac{(\alpha-\alpha_0)^2}{2\sigma_y^2}\right). \tag{4.16}$$

The frequency coefficients ω_α, ω_t in the sine terms control the number of peaks in the α and t axes, respectively – their amplitudes being given by B. Texture bias can then be modelled using the exponential term (a Gaussian function), which applies damping in the α axis. The Gaussian reduces the function to zero at upskin angles ($0° \leq \alpha \leq 90°$). Lastly, ψ is multiplied by a scaling factor Δ which calculates the displacements δ applied to specific struts, depending on overhang angle:

$$\delta = \psi \times \Delta \tag{4.17}$$

$$\Delta = f(\theta). \tag{4.18}$$

An example of $f(\theta)$ is shown in Figure 4.6d which is a ramp function that sets all displacements to zero for $\theta < 45°$ and linearly increases the scaling factor for increasing $\theta > 45°$, thus increasing the downskin texture bias of the struts with greater overhang angles. Note that the applied displacements are symmetric about the reference vector show in Figure 4.5b, as α does not exceed 180°.

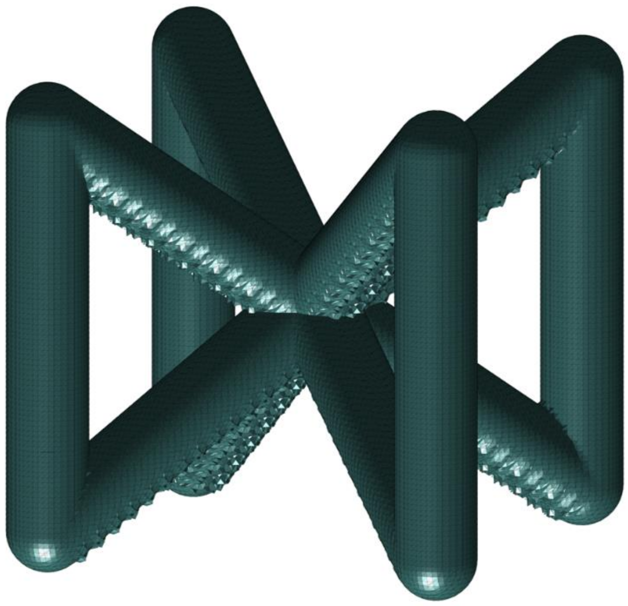


Figure 4.4 – Triangulated surface of BCCZ unit cell. Displacements have been applied to the downskin points.

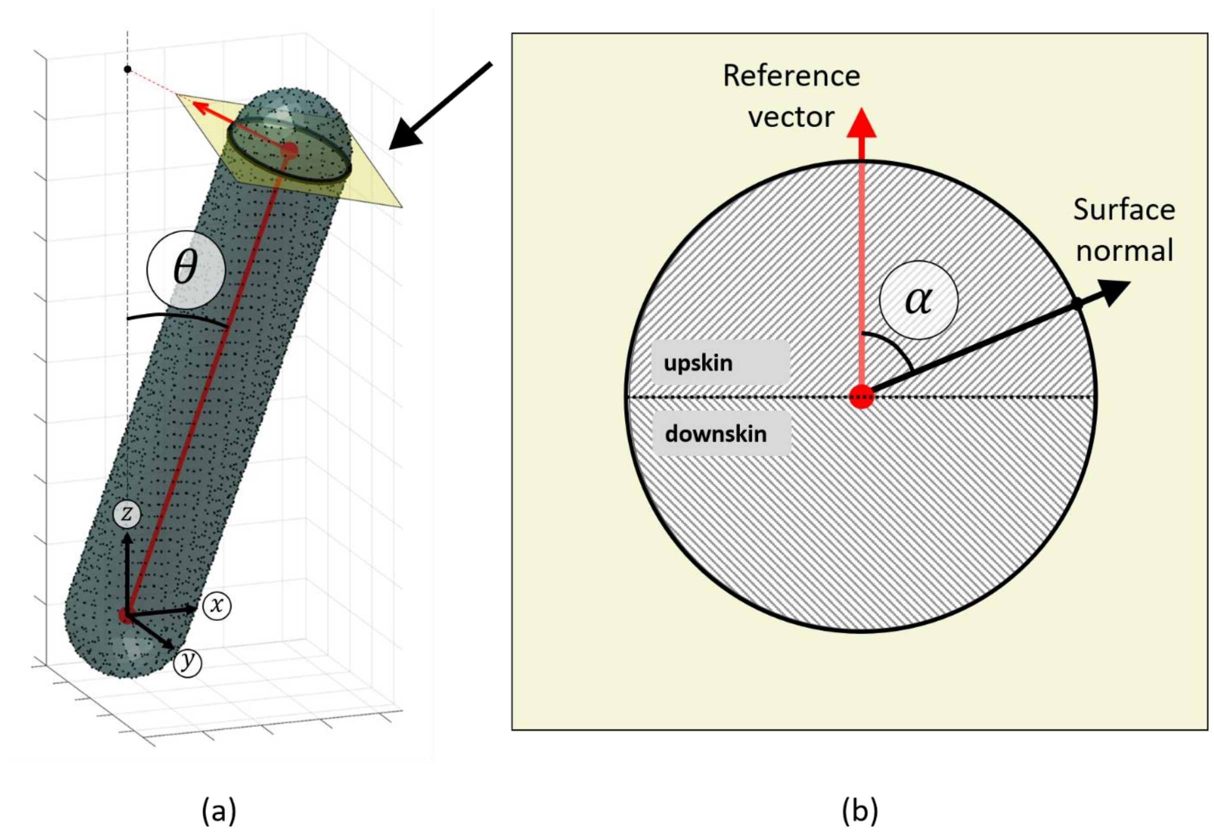


Figure 4.5 – Illustrations explaining the calculation of (a) overhang angle θ and (b) up/downskin angle α.

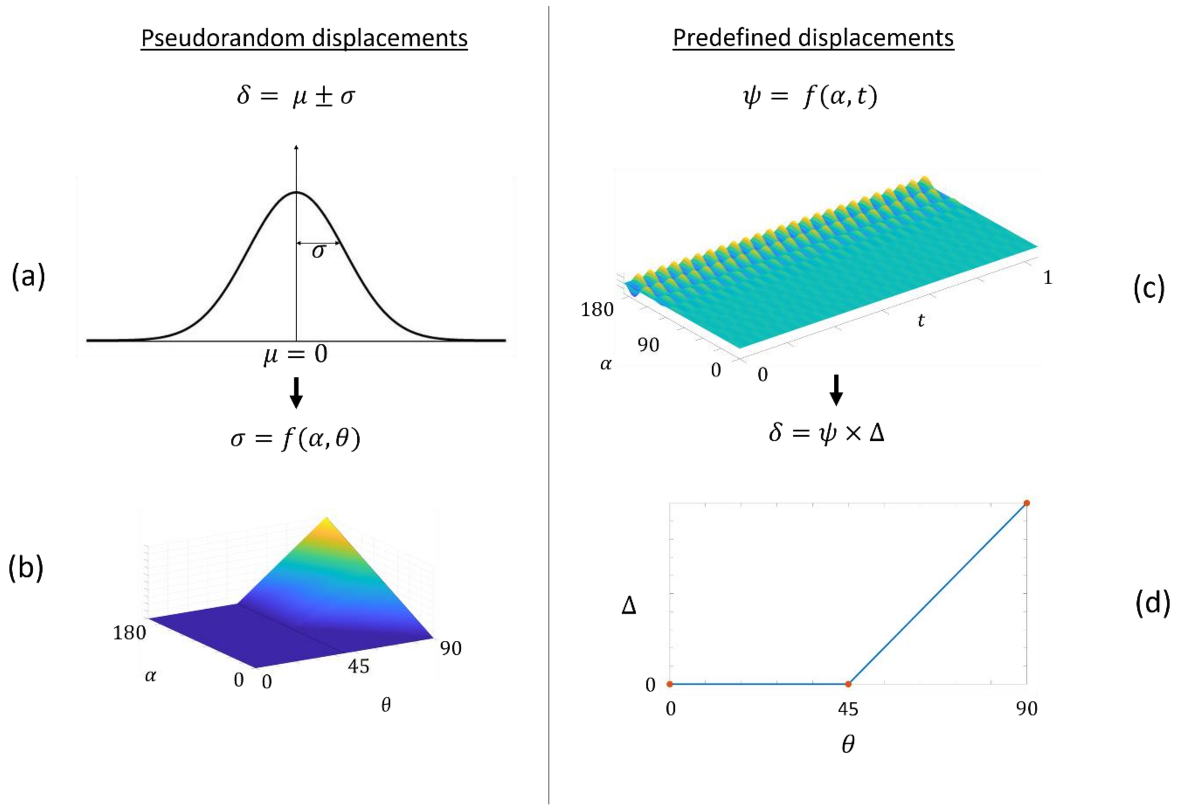


Figure 4.6 – Methods for applying displacements to the strut surface. Left: pseudorandom displacements are selected from normal distribution (a) with standard deviation σ where $\sigma = f(\alpha, \theta)$, as shown in (b). Right: simulated surface ψ (c) is generated and multiplied by a scaling factor Δ (d).

5 FE mesh generation

The surface mesh must be converted into a tetrahedral mesh in order to be used for FE modelling. This conversion is performed using open-source MATLAB meshing toolbox, Iso2mesh [33]. Iso2mesh communicates directly with MATLAB's isosurface.m function. Isosurface.m outputs a three-column matrix of (x, y, z) coordinates and a three-column triangulation matrix which indexes the triangular faces of the surface mesh – these are the inputs for Iso2mesh. We use the Iso2mesh function 'cgals2m' that operates using embedded CGAL meshing algorithms to create the output mesh [34]. The output mesh is created using a restricted Delaunay tetrahedralisation (RDT).

To create the tetrahedral mesh, the RDT first computes a set of sample points on the input surface and creates a 3D triangulation of these sample points. Additional points are then iteratively created and refined until a given criteria on the size and shape of the mesh elements is satisfied [34]. The output mesh is defined by a new three-column matrix that includes the new (x, y, z) points in the tetrahedral mesh, and a four-column matrix that indexes the individual tetrahedra. An example of the conversion from surface mesh to tetrahedral mesh is shown in Figure 5.1.

To constrain the output mesh, we modify two parameters. The first parameter controls the size of the triangular faces on the surface, therefore, controlling how well the input surface is preserved in the tetrahedralisation. This surface constraint is termed $radbound$ (short for radius boundary) and it is the upper bound on the radius of the circumcircle for each of the faces on the surface of the output mesh. The second parameter $maxvol$ (short for maximum volume) controls the size of all the elements in the mesh and is the upper bound on the volume of the circumsphere for each of the elements. Using $radbound$ and $maxvol$ allow for a graded mesh, where smaller elements are on the surface to better approximate the geometry, and large elements are within the body.

To demonstrate how appropriate meshing parameters are selected, we will use an example of the BCCZ unit cell. To select a suitable $radbound$ value, a comparison is performed between the volumes enclosed by the input surface mesh and the output mesh. The percentage error ϵ between these two volumes is given by

$$\epsilon = \left(1 - \frac{V_m}{V_s}\right) \times 100 \qquad 5.1$$

where V_m and V_s are the volumes of the output tetrahedral mesh and the input surface mesh, respectively. Table 1 illustrates the effect of selected $radbound$ values on the error ϵ, where $radbound$ has been normalised with the strut's radius ($radbound/r$). Ten meshes were created for each $radbound$ value, and the mean and standard deviations of the errors are calculated. Averaging is required because the RDT does not provide a single unique solution, therefore, some variance exists between output meshes with the same input parameters. As shown in Table 1, the mean error reduces to below 1% at a normalised $radbound$ of 0.2, showing good agreement between the input and output meshes. The low standard deviation values confirm relatively high repeatability in the output meshes.

Consideration must also be given to the shape of the mesh elements. Reducing $radbound$ may result in a steep gradient between the sizes of the surface and body elements. A steep gradient may have a detrimental effect on the shape of some tetrahedra, creating low quality elements. To prevent low quality elements, the $maxvol$ parameter is used to reduce the overall size of the elements, thus reducing the gradient between the size of the surface and body elements. The shape quality of each element is quantified by comparing each element to the equilateral tetrahedron derived from the element's circumsphere. The quality value is then calculated as the quotient of these two volumes, as shown in Figure 5.2. This quality value will be between 0 and 1, where 1 indicates the highest quality i.e. the tetrahedral element is an equilateral tetrahedron.

To investigate mesh quality, the $radbound/r$ parameter is fixed at 0.2, and $maxvol$ is varied. The $maxvol$ values have been normalised using the volume of the sphere with radius equal to the strut's radius. To aid in visualising the effect of $maxvol$ on mesh quality, we calculate the distance between each element's centroid and the closest line segment. This distance value can be used to indicate whether the element is at the surface of the mesh (high distance value) or deep within the body (low distance value). The histograms in Figure 5.3 show the effect of $maxvol$ on the distribution of mesh quality. At higher $maxvol$ values, low quality surface elements dominate the distribution (Figure 5.3a-b). As $maxvol$ is decreased, more elements of lower volume are used and the distribution improves (Figure 5.3c-d).

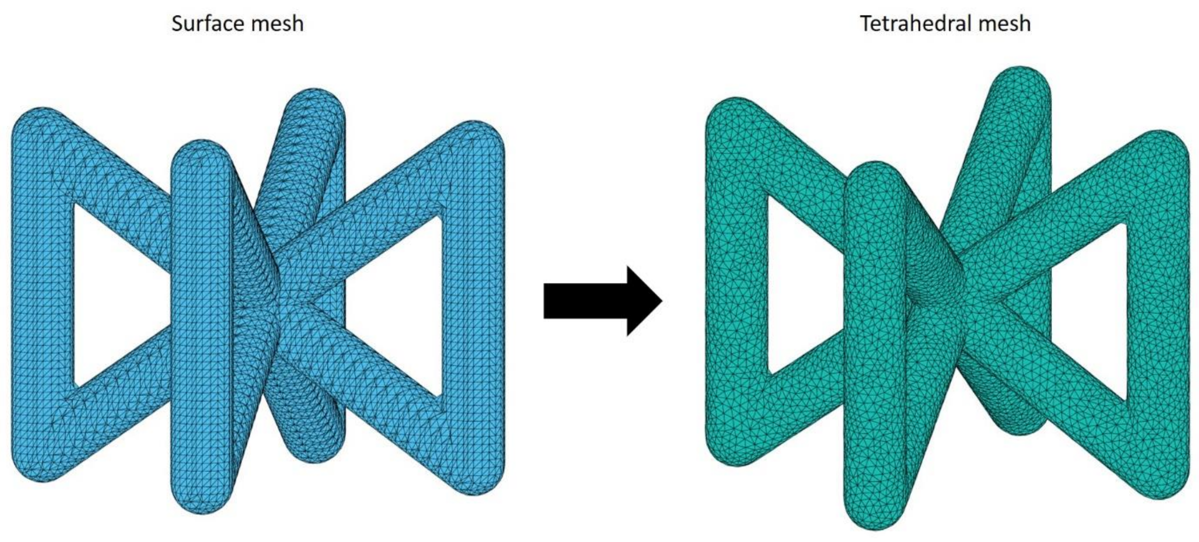
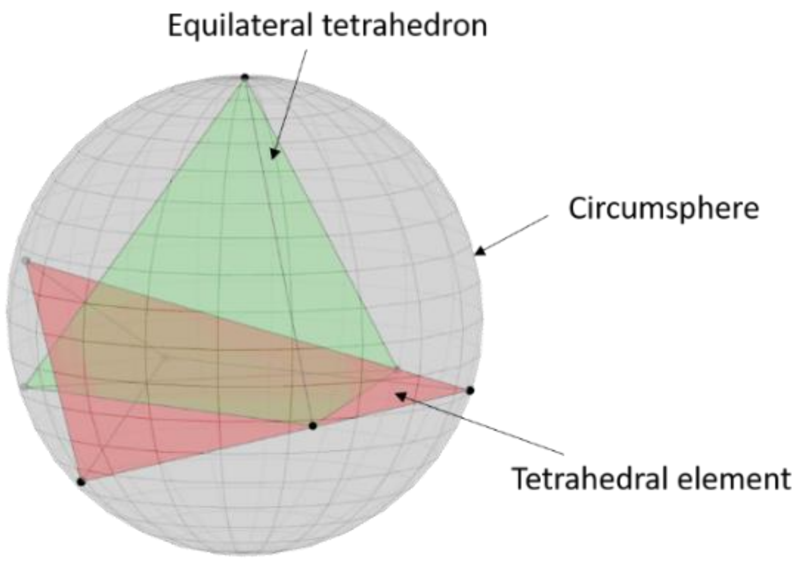


Figure 5.1 – Left: triangulated surface mesh; the output of isosurface.m. Right: tetrahedral mesh; the output of Iso2mesh.

Radbound/*r*	**1**	**0.5**	**0.2**	**0.1**
Mean error (%)	15.8	4.18	0.68	0.17
Std dev (%)	1.2	0.7	0.52	0.21

Table 1 – Mean percentage error between the volumes of the input surface mesh and output tetrahedral mesh. Standard deviation taken from 10 repeats for each radbound/r value (percentage of the mean).



$$quality = \frac{vol_{tetrahedron}}{vol_{equilateral}}$$

Figure 5.2 – Mesh quality calculation.

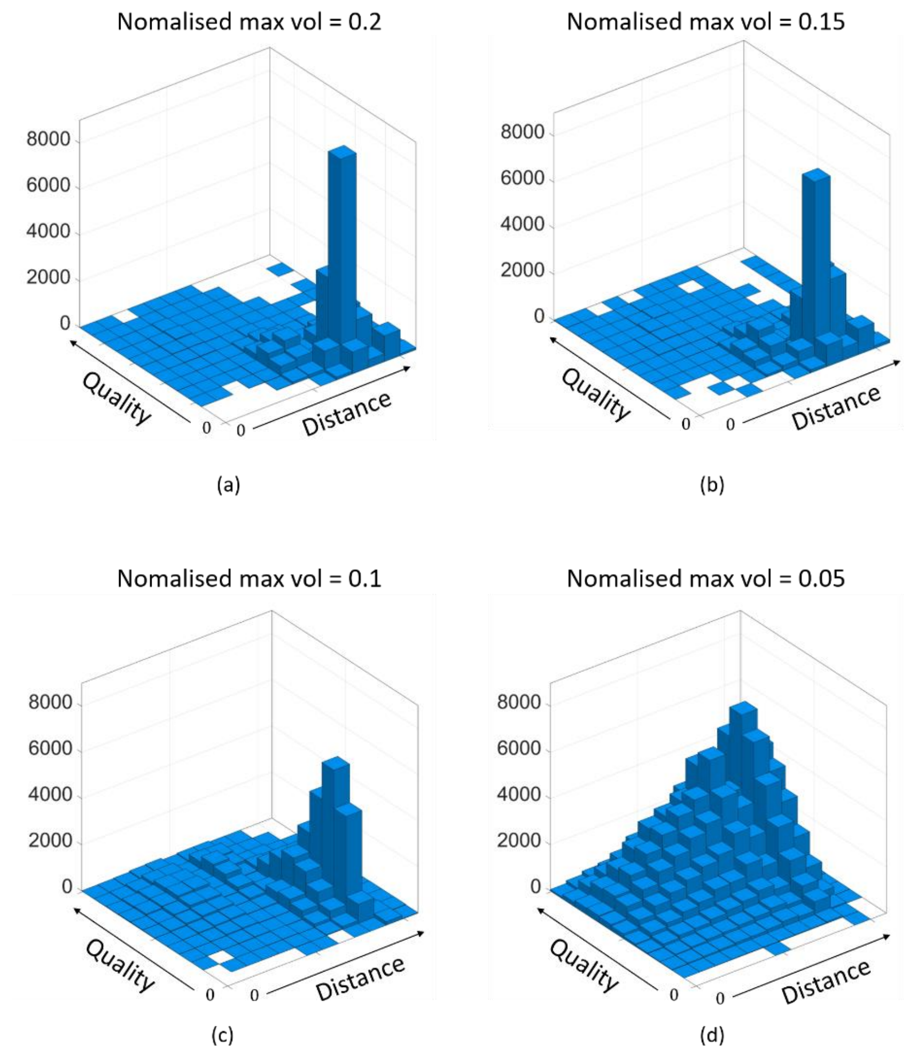


Figure 5.3 – Effect of $maxvol$ on the quality of mesh elements.

6 Implementation

This section provides a summary of the method presented in this paper for modelling both form and surface defects in strut-based lattice structures using tetrahedral meshes. The following points follow the information shown in Figure 6.1.

- The first step is the signed distance function, which outputs a 3D distance field based on a calculation of the Euclidean distance between all points in the domain and the line segments used to define the strut's medial axis.
- The distance field is input into isosurface.m, which determines the coordinates at which the SDF = 0. Isosurface.m outputs a surface mesh and indexing matrix.
- If surface defects are to be applied, the coordinates of the surface mesh are input into the surface defects function, which modifies the position of points in a given area.
- The coordinates of the surface mesh and the indexing matrix are input into Iso2mesh, using appropriate $radbound$ and $maxvol$ values. Iso2mesh outputs the coordinates of the new tetrahedral mesh and an element indexing matrix.
- Lastly, the outputs from Iso2mesh are used to check the quality of the individual elements.

These modelling functions provide full control over the definition of defects within the lattice geometry and are therefore suitable for conducting parametric studies of lattice behaviour in the presence of defects. Additionally, these functions are compatible with any measurement process (e.g. XCT, focus variation) which can quantify the geometry of a lattice structure – the selection of the modelling parameters can be informed by measurement data, enabling a more realistic representation of the defects in the lattice struts.

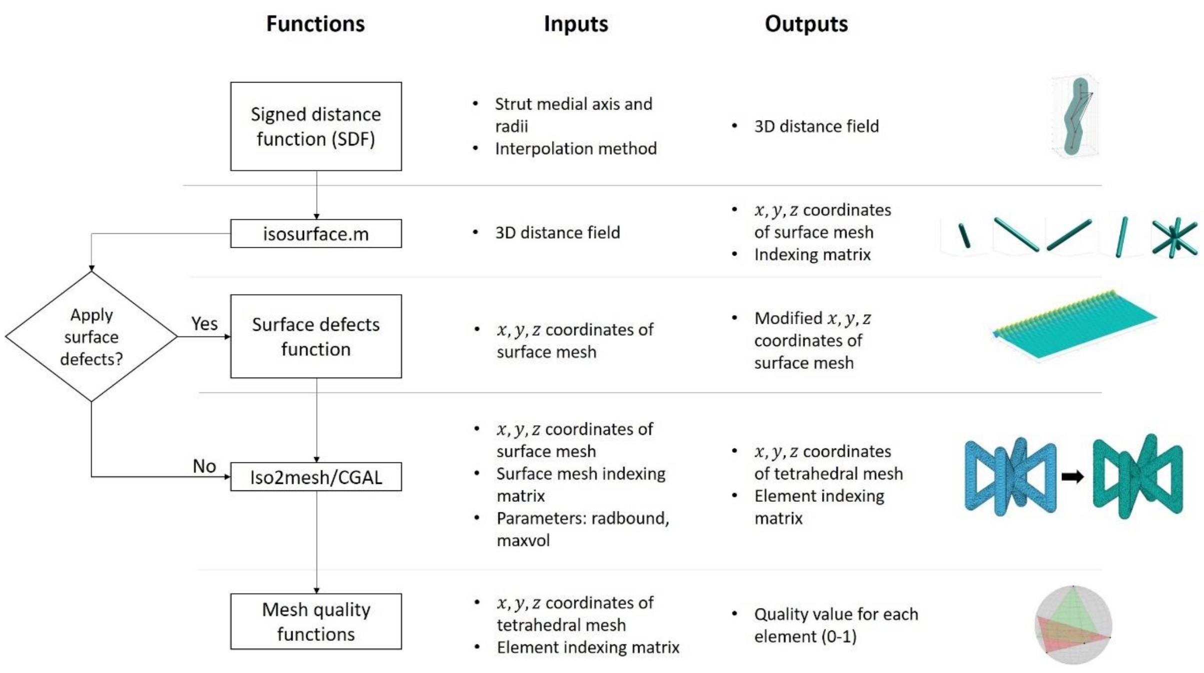

Functions
Inputs
Outputs
Signed distance function (SDF)
• Strut medial axis and radii
• Interpolation method
• 3D distance field
isosurface.m
• 3D distance field
• x, y, z coordinates of surface mesh
• Indexing matrix
Apply surface defects?
Yes
Surface defects function
• x, y, z coordinates of surface mesh
• Modified x, y, z coordinates of surface mesh
No
Iso2mesh/CGAL
• x, y, z coordinates of surface mesh
• Surface mesh indexing matrix
• Parameters: radbound, maxvol
• x, y, z coordinates of tetrahedral mesh
• Element indexing matrix
Mesh quality functions
• x, y, z coordinates of tetrahedral mesh
• Element indexing matrix
• Quality value for each element (0-1)

Figure 6.1 – Flow chart of the modelling process.

7 FE study: texture bias

As a demonstration, we perform an FE parametric study to investigate the impact of texture bias on the compressive Young's modulus of two popular lattice structures: BCCZ and octet-truss. Figure 7.1 shows the two lattice structures and their dimensions. Texture bias is modelled using the method from Section 4.4, where a simulated surface is used to apply displacements to the strut surfaces. We simulate the texture bias by defining a simulated surface ψ which applies peaks and troughs to one side of the lattice strut, thus replicating the general discrepancy between upskin and downskin surfaces. The simulated surface ψ is defined using eq 4.16 where $\omega_\alpha = \omega_t = 8$, $\sigma_y = \pi/5$ and $\alpha_0 = 0$ for upskin bias, and $\alpha_0 = \pi$ for downskin bias. ψ has a form similar to that in Figure 4.6c. In this parametric study, the amplitude B of ψ is the parameter being investigated, where B is increased through seven states for this study, $B = 0.1, 0.2, \dots, 0.7$. The displacements δ applied to the struts are then calculated using

$$\delta = \psi \times \gamma \times \Delta, \tag{7.1}$$

where

$$\gamma = \begin{cases} 0.1, & \text{if } \psi > 0 \\ 1, & \text{if } \psi \leq 0 \end{cases}; \tag{7.2}$$

$$\Delta = \begin{cases} 1, & \text{if } \theta > 0 \\ 0, & \text{if } \theta = 0 \end{cases}. \tag{7.3}$$

γ reduces the values of the positive displacements so that the function predominantly applies negative displacements that reduce the radius of the strut surfaces. Δ excludes the vertical struts ($\theta = 0$) from the texture bias. For each value of B, five meshes are created, to be used for averaging the results. Iso2mesh parameters $radbound$ and $maxvol$ were both set to 0.2.

Commercial FE software, Abaqus, was used to perform the simulations. A script was developed in MATLAB to create the FE meshes, and to create and run the Abaqus input files. As shown in Figure 7.1, the upper and lower faces are flattened in order to apply boundary conditions. The nodes on the bottom face (red) are constrained in all degrees of freedom. Compression is applied by displacing the nodes on the top face (yellow) a tenth of the lattice height (~2 mm) in the negative z-direction. The material used is Ti6Al4V with Young's modulus 126 GPa and Poisson ratio 0.32. We extract the reaction force F from the top face (yellow) and calculate the Young's modulus E with the following equation

$$E = FL/AU \tag{7.4}$$

where L is the original length (height) of the lattice structure, A is the cross-sectional area of the lattice and U is the displacement applied to the nodes on the top face.

Figure 7.2 shows the relationship between the predicted Young's modulus of the lattices and the amplitude of the texture bias. Both plots show lower Young's modulus for downskin bias, with some divergence between upskin and downskin upon increasing amplitude. The results show the octet-truss structure to be more sensitive than BCCZ to the texture bias, as shown in Table 2, which lists the percentage change in Young's modulus between the lowest and highest amplitudes applied to the surfaces.

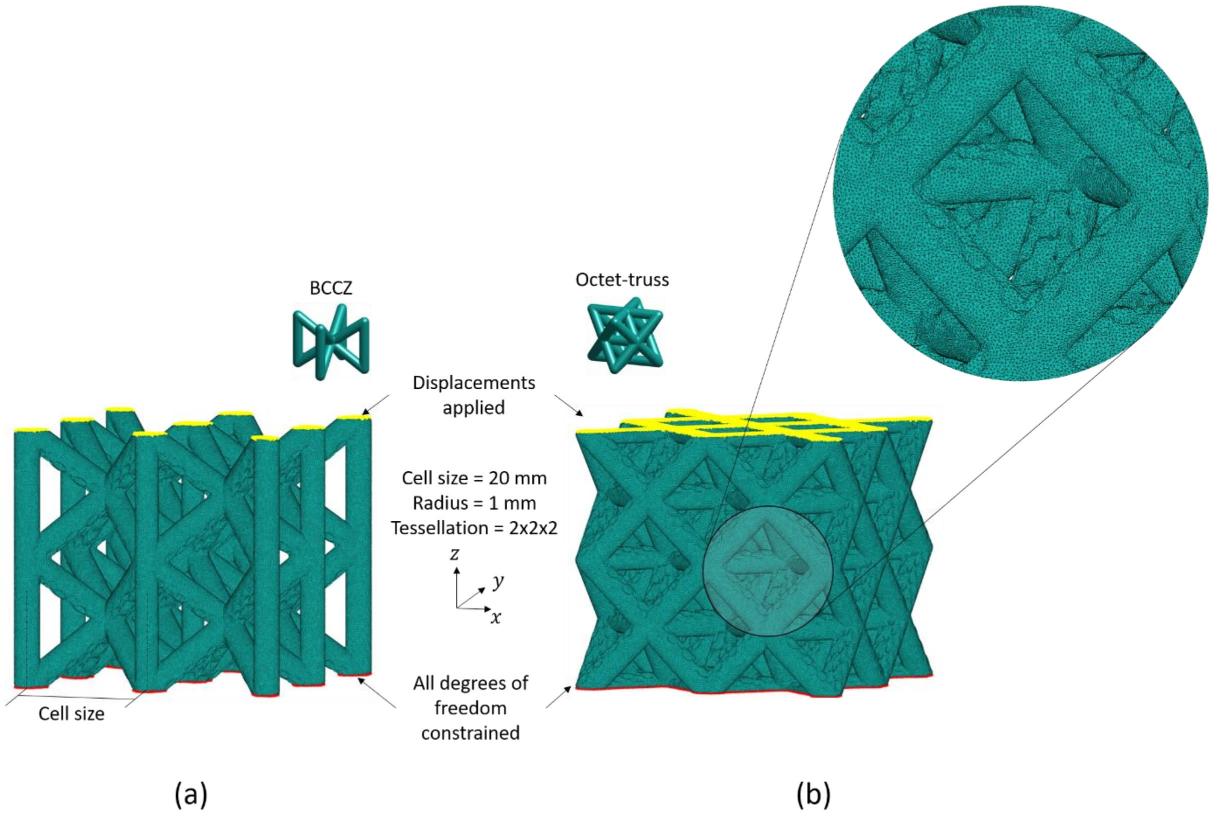
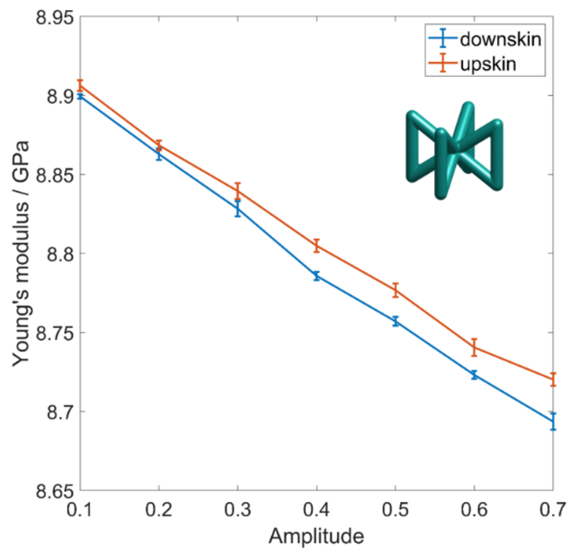
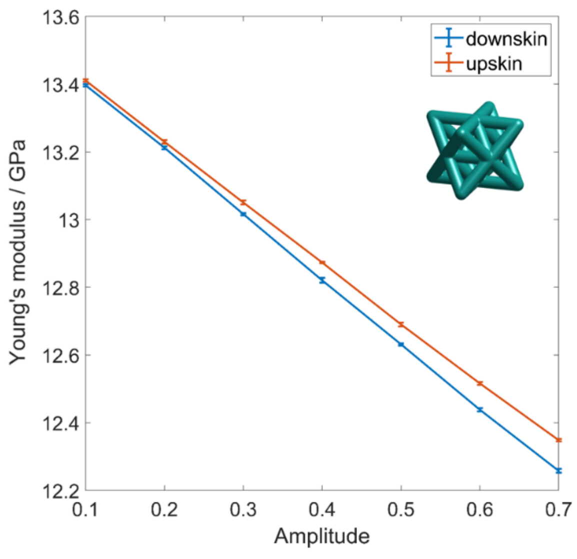


Figure 7.1 – Tetrahedral meshes of BCCZ and octet-truss lattices. (a) Downskin texture bias has been applied to the BCCZ lattice; (b) upskin texture bias has been applied to the octet-truss.



(a)



(b)

Figure 7.2 – Simulation results showing the effect of texture bias on the Young's modulus of (a) BCCZ and (b) octet-truss lattices. Error bars are the standard deviations from five repeats.

	BCCZ	Octet
Downskin	2.31	8.50
Upskin	2.09	7.92

Table 2 - Percentage reduction in compressive Young's modulus.

8 Discussion and conclusion

This paper has presented an FE framework for modelling form and surface defects in strut-based lattice structures. Intuitive mathematical definitions have been provided for modelling strut waviness, radius variation, elliptical cross-sections and texture bias. Conversion from surface mesh to tetrahedral mesh was performed via open-source MATLAB toolbox, Iso2mesh. Simple mesh quality tools were developed to aid in optimising Iso2mesh parameters. A parametric FE study was performed to simulate the impact of texture bias on the compressive Young’s modulus of the BCCZ lattice and octet-truss lattice.

8.1 Defect modelling

The modelling framework presented in this paper provides a high level of control over the form and surface of lattice struts. The mathematical definitions for the defects are well suited for FE parametric studies – individual variables from the underlying functions can be isolated and studied, as demonstrated in this paper. This modelling framework could be easily adapted to other types of defects not considered in this paper, such as notched or completely broken struts. The high geometric control of this approach could also be used to improve lattice designs, for example by applying fillets, which can improve fatigue strength.

8.2 FE study

The texture bias study showed the octet-truss to demonstrate a higher sensitivity to texture bias, with a greater reduction in compressive Young’s modulus than the BCCZ lattice. These results are expected, because all the struts in the octet-truss are inclined and experience a bending load, whereas the BCCZ is reinforced with axially loaded vertical struts. The divergence between the upskin and downskin plots in Figure 7.2 suggests that downskin defects impact the Young’s modulus more strongly in both lattice structures. However, the maximum difference between the upskin and downskin plots is less than 1%, which may be negligible, depending on the application. Nevertheless, this texture bias study is useful for quantifying how much more sensitive the octet-truss is to texture bias than the BCCZ lattice. Visual inspection of Figure 7.2 suggests a linear Young’s modulus response of the octet-truss, in contrast to the BCCZ response which is notably less linear – the source of this behaviour is unclear and may be due to the number of repeat measurements.

8.3 Limitations

Regarding the SDF, increasing the domain resolution and/or the number of line segments in the strut can cause the SDF to become computationally expensive. Replicating the real AM process, where multiple discontinuities are applied in each sub-millimetre layer, may be unfeasible. Therefore, as the computational load increases, it becomes increasingly important to utilise the design’s symmetry and duplicate/transform the underlying distance fields where possible – defects may also need duplicating, which undermines the model’s representation of the as-built lattice.

Considering the FE modelling stage, the use of tetrahedral elements allows for greater control over the geometry, however, they are significantly more computationally expensive than beam elements. Increasing the tessellation further may require homogenisation techniques (as demonstrated elsewhere, [15] for example). Increasing the tessellation is an important step in determining whether the results from the study in Section 7 have converged – additionally, the seemingly non-linear BCCZ response may not manifest at higher tessellations.

8.4 Future work

This work can be continued in several ways. Further parametric studies can be conducted to investigate the impact of different defects on specific lattice behaviour. The modelling of defects could be improved by using measurement data to inform the selection of parameters in the underlying functions. Note that the simulated surface used in the FE study (eq 7.1) was built using an arbitrary function sufficient for the generation of graded peaks and valleys on the surface. Future modelling of texture bias could use a simulated surface possessing statistical properties obtained from focus variation measurement data of lattice struts, for example. Mechanical testing of BCCZ and octet-truss lattices will be crucial for validating the accuracy of this model. Higher tessellation lattice structure models will be tested, to investigate the convergence of mechanical properties - homogenisation techniques may also be investigated.

This modelling framework can also be extended beyond the study of defects on compressive properties, for example, the impact of texture bias on heat transfer or fluid flow could be investigated using this approach. The presented modelling framework is versatile and will hopefully be used for improving the understanding of AM lattice structures in a range of applications.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/M008983/1, EP/L01534X/1]

9 References

[1] L.J. Gibson, M.F. Ashby, Cellular Solids: Structure and properties, 2nd ed., Cambridge University Press, Cambridge, 1997.

[2] P. Köhnen, C. Haase, J. Bültmann, S. Ziegler, J.H. Schleifenbaum, W. Bleck, Mechanical properties and deformation behavior of additively manufactured lattice structures of stainless steel, Mater. Des. 145 (2018) 205–217. doi:10.1016/j.matdes.2018.02.062.

[3] T. Tancogne-Dejean, D. Mohr, Stiffness and specific energy absorption of additively-manufactured metallic BCC metamaterials composed of tapered beams, Int. J. Mech. Sci. 141 (2018) 101–116. doi:10.1016/j.ijmecsci.2018.03.027.

[4] I. Maskery, A. Hussey, A. Panesar, A. Aremu, C. Tuck, I. Ashcroft, R. Hague, An investigation into reinforced and functionally graded lattice structures, J. Cell. Plast. 53 (2017) 151–165. doi:10.1177/0021955X16639035.

[5] S. Ma, Q. Tang, X. Han, Q. Feng, J. Song, R. Setchi, Y. Liu, Y. Liu, A. Goulas, D.S. Engstrøm, Y.Y. Tse, N. Zhen, Manufacturability, Mechanical Properties, Mass-Transport Properties and Biocompatibility of Triply Periodic Minimal Surface (TPMS) Porous Scaffolds Fabricated by Selective Laser Melting, Mater. Des. 195 (2020) 1–15. doi:10.1016/j.matdes.2020.109034.

[6] N. Taniguchi, S. Fujibayashi, M. Takemoto, K. Sasaki, B. Otsuki, T. Nakamura, T. Matsushita, T. Kokubo, S. Matsuda, Effect of pore size on bone ingrowth into porous titanium implants fabricated by additive manufacturing: An in vivo experiment, Mater. Sci. Eng. C. 59 (2016) 690–701. doi:10.1016/j.msec.2015.10.069.

[7] D. Carluccio, C. Xu, J. Venezuela, Y. Cao, D. Kent, M. Bermingham, A.G. Demir, B. Previtali, Q. Ye, M. Dargusch, Additively manufactured iron-manganese for biodegradable porous load-bearing bone scaffold applications, Acta Biomater. 103 (2020) 346–360. doi:10.1016/j.actbio.2019.12.018.

[8] S. Catchpole-Smith, R.R.J. Sélo, A.W. Davis, I.A. Ashcroft, C.J. Tuck, A. Clare, Thermal conductivity of TPMS lattice structures manufactured via laser powder bed fusion, Addit. Manuf. 30 (2019) 100846. doi:10.1016/j.addma.2019.100846.

[9] A. Chaudhari, P. Ekade, S. Krishnan, Experimental investigation of heat transfer and fluid flow in octet-truss lattice geometry, Int. J. Therm. Sci. 143 (2019) 64–75. doi:10.1016/j.ijthermalsci.2019.05.003.

[10] A. Beharic, R. Rodriguez Egui, L. Yang, Drop-weight impact characteristics of additively manufactured sandwich structures with different cellular designs, Mater. Des. 145 (2018) 122–134. doi:10.1016/j.matdes.2018.02.066.

[11] R.A.W. Mines, S. Tsopanos, Y. Shen, R. Hasan, S.T. McKown, Drop weight impact behaviour of sandwich panels with metallic micro lattice cores, Int. J. Impact Eng. 60 (2013) 120–132. doi:10.1016/j.ijimpeng.2013.04.007.

[12] W. Elmadih, W.P. Syam, I. Maskery, R.K. Leach, Designing low frequency bandgaps in additively manufactured parts using internal resonators, Proc. ASPE, Las Vegas, USA. (2018).

[13] W. Elmadih, D. Chronopoulos, W.P. Syam, I. Maskery, H. Meng, R.K. Leach, Three-dimensional resonating metamaterials for low-frequency vibration attenuation, Sci. Rep. 9 (2019) 1–8. doi:10.1038/s41598-019-47644-0.

[14] I. Echeta, X. Feng, B. Dutton, R.K. Leach, S. Piano, Review of defects in lattice structures manufactured by powder bed fusion, Int. J. Adv. Manuf. Technol. 106 (2020) 2649–2668. doi:10.1007/s00170-019-04753-4.

[15] D. Melancon, Z.S. Bagheri, R.B. Johnston, L. Liu, M. Tanzer, D. Pasini, Mechanical characterization of structurally porous biomaterials built via additive manufacturing: experiments, predictive models, and design maps for load-bearing bone replacement implants, Acta Biomater. 63 (2017) 350–368. doi:10.1016/j.actbio.2017.09.013.

[16] T.B. Sercombe, X. Xu, V.J. Challis, R. Green, S. Yue, Z. Zhang, P.D. Lee, Failure modes in high strength and stiffness to weight scaffolds produced by Selective Laser Melting, Mater. Des. 67 (2015) 501–508. doi:10.1016/j.matdes.2014.10.063.

[17] G. Dong, Y. Tang, Y.F. Zhao, A Survey of Modeling of Lattice Structures Fabricated by Additive Manufacturing, J. Mech. Des. 139 (2017) 100906. doi:10.1115/1.4037305.

[18] L. Boniotti, S. Beretta, L. Patriarca, L. Rigoni, S. Foletti, Experimental and numerical investigation on compressive fatigue strength of lattice structures of AlSi7Mg manufactured by SLM, Int. J. Fatigue. 128 (2019) 105181. doi:10.1016/j.ijfatigue.2019.06.041.

[19] M. Dallago, B. Winiarski, F. Zanini, S. Carmignato, M. Benedetti, On the effect of geometrical imperfections and defects on the fatigue strength of cellular lattice structures additively manufactured via Selective Laser Melting, Int. J. Fatigue. 124 (2019) 348–360. doi:10.1016/j.ijfatigue.2019.03.019.

[20] M. Benedetti, A. du Plessis, R.O. Ritchie, M. Dallago, S.M.J. Razavi, F. Berto, Architected cellular materials: A review on their mechanical properties towards fatigue-tolerant design and fabrication, Mater. Sci. Eng. R Reports. 144 (2021) 100606. doi:10.1016/j.mser.2021.100606.

[21] M.R. Karamooz Ravari, S. Nasr Esfahani, M. Taheri Andani, M. Kadkhodaei, A. Ghaei, H. Karaca, M. Elahinia, On the effects of geometry, defects, and material asymmetry on the mechanical response of shape memory alloy cellular lattice structures, Smart Mater. Struct. 25 (2016). doi:10.1088/0964-1726/25/2/025008.

[22] B. Lozanovski, M. Leary, P. Tran, D. Shidid, M. Qian, P. Choong, M. Brandt, Computational modelling of strut defects in SLM manufactured lattice structures, Mater. Des. 171 (2019). doi:10.1016/j.matdes.2019.107671.

[23] L. Liu, P. Kamm, F. García-Moreno, J. Banhart, D. Pasini, Elastic and failure response of imperfect three-dimensional metallic lattices: the role of geometric defects induced by Selective Laser Melting, J. Mech. Phys. Solids. 107 (2017) 160–184. doi:10.1016/j.jmps.2017.07.003.

[24] X. Cao, Y. Jiang, T. Zhao, P. Wang, Y. Wang, Z. Chen, Y. Li, D. Xiao, D. Fang, Compression experiment and numerical evaluation on mechanical responses of the lattice structures with stochastic geometric defects originated from additive-manufacturing, Compos. Part B Eng. 194 (2020) 108030. doi:10.1016/j.compositesb.2020.108030.

[25] H. Lei, C. Li, J. Meng, H. Zhou, Y. Liu, X. Zhang, P. Wang, D. Fang, Evaluation of compressive properties of SLM-fabricated multi-layer lattice structures by experimental test and μ-CT-based finite element analysis, Mater. Des. 169 (2019) 107685. doi:10.1016/j.matdes.2019.107685.

[26] B. Lozanovski, D. Downing, P. Tran, D. Shidid, M. Qian, P. Choong, M. Brandt, M. Leary, A Monte Carlo simulation-based approach to realistic modelling of additively manufactured lattice structures, Addit. Manuf. 32 (2020) 101092. doi:10.1016/j.addma.2020.101092.

[27] A. El Elmi, D. Melancon, M. Asgari, L. Liu, D. Pasini, Experimental and numerical

investigation of selective laser melting-induced defects in Ti-6Al-4V octet truss lattice material: The role of material microstructure and morphological variations, J. Mater. Res. (2020) 1–13. doi:10.1557/jmr.2020.75.

[28] A. du Plessis, I. Yadroitsava, I. Yadroitsev, Ti6Al4V lightweight lattice structures manufactured by laser powder bed fusion for load-bearing applications, Opt. Laser Technol. 108 (2018) 521–528. doi:10.1016/j.optlastec.2018.07.050.

[29] N. Korshunova, G. Alaimo, S.B. Hosseini, M. Carraturo, A. Reali, J. Niiranen, F. Auricchio, E. Rank, S. Kollmannsberger, Image-based numerical characterization and experimental validation of tensile behavior of octet-truss lattice structures, Addit. Manuf. 41 (2021) 101949. doi:10.1016/j.addma.2021.101949.

[30] A. Panesar, M. Abdi, D. Hickman, I. Ashcroft, Strategies for functionally graded lattice structures derived using topology optimisation for Additive Manufacturing, Addit. Manuf. 19 (2018) 81–94. doi:10.1016/J.ADDMA.2017.11.008.

[31] D.J. Yoo, Heterogeneous porous scaffold design for tissue engineering using triply periodic minimal surfaces, Int. J. Precis. Eng. Manuf. 13 (2012) 527–537. doi:10.1007/s12541-012-0068-5.

[32] M. Dallago, S. Raghavendra, V. Luchin, G. Zappini, D. Pasini, M. Benedetti, The role of node fillet, unit-cell size and strut orientation on the fatigue strength of Ti-6Al-4V lattice materials additively manufactured via laser powder bed fusion, Int. J. Fatigue. 142 (2021) 105946. doi:10.1016/j.ijfatigue.2020.105946.

[33] Q. Fang, D.A. Boas, Tetrahedral mesh generation from volumetric binary and grayscale images, 2009 IEEE Int. Symp. Biomed. Imaging. (2009) 1142–1145. doi:10.1109/ISBI.2009.5193259.

[34] The CGAL Project, CGAL, Computational Geometry Algorithms Library, (2020). https://www.cgal.org (accessed March 16, 2020).