Additive Manufacturing 13 (2017) 1–13

Contents lists available at ScienceDirect

Additive Manufacturing

journal homepage: www.elsevier.com/locate/addma

Full Length Article

A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing

A.O. Aremu, J.P.J. Brennan-Craddock, A. Panesar, I.A. Ashcroft*, R.J.M. Hague, R.D. Wildman, C. Tuck

Faculty of Engineering, University of Nottingham, UK

ARTICLE INFO

Article history:
Received 9 October 2015
Received in revised form 11 October 2016
Accepted 27 October 2016
Available online 9 November 2016

Keywords:
Lattice
Voxel
Tesselation
Net-skin
Functional grading

ABSTRACT

Additive Manufacturing (AM) enables the production of geometrically complex parts that are difficult to manufacture by other means. However, conventional CAD systems are limited in the representation of such parts. This issue is exacerbated when lattice structures are combined or embedded within a complex geometry. This paper presents a computationally efficient, voxel-based method of generating lattices comprised of practically any cell type that can conform to an arbitrary external geometry. The method of conforming involves the tessellation and trimming of unit cells that can leave 'hanging' struts at the surface, which is a possible point of weakness in the structure. A method of joining these struts to form an external two dimensional lattice, termed a 'net-skin' is also described. Traditional methods of manufacturing lattice structures generally do not allow variation of cell properties within a structure; however, additive manufacturing enables graded lattices to be generated that are potentially more optimal. A method of functionally grading lattices is, therefore, also described to take advantage of this manufacturing capability.

1. Introduction

The geometric freedom afforded by Additive Manufacturing (AM) allows the fabrication of complex lattice structures without the use of tooling. Rather than forming parts in a subtractive (e.g., machined) or formative (e.g., moulded) manner, AM 'prints' parts in layers. Techniques classified as AM include power bed fusion, vat polymerization, jetting, material extrusion, sheet lamination and direct energy deposition (ASTM 2792-12). Industrial exploitation of these techniques requires research and development in processes, materials and design-systems. While process and materials have advanced considerably, there has been less progress to date on design issues. The development of design strategies for AM enables several advantages over conventional manufacturing. One such advantage is the realization of complex light-weighted parts. By designing a part with material only in the regions that are required, the part's strength to weight ratio is increased. In the aerospace and automotive industries, this translates to economic savings from reduced fuel consumption and a reduced carbon footprint.

Currently, two paradigms are prominently used to realize light-weighted parts for AM. The first, topology optimization (TO), seeks the optimal distribution of material in a part by applying mathematical algorithms to the part's domain [1]. A number of these algorithms have been developed, including homogenization [1], evolutionary structural optimization [ESO] [2,3], solid isotropic material with penalization [4] and level sets [5]. According to Brackett et al. [6], the utilization of these algorithms is currently limited by mesh resolution, manufacturing constraints and post processing. Also, Aremu et al. [7] showed that the complexity and performance of a mechanical part optimized with a bidirectional evolutionary structural optimization [BESO] algorithm could be compromised by values assumed for the optimization parameters.

Recent research has seen the development of AM techniques to manufacture multi-functional components [8]. Optimization of such parts requires multiple objectives, with a resulting series of solutions in a Pareto optimal set [9,10]. The computational effort

☆One of the authors of this article is part of the Editorial Board of the journal. Full responsibility for the editorial and peer-review process for this article lies with the journal's Editor-in-Chief Prof. Ryan Wicker and Deputy Editor Prof. Eric MacDonald. Furthermore, the authors of this article had no and do not currently have access to any confidential information related to its peer-review process.

* Corresponding author.
E-mail address: ian.ashcroft@nottingham.ac.uk (I.A. Ashcroft).

http://dx.doi.org/10.1016/j.addma.2016.10.006

required to generate these sets is high, creating a pressure to use coarse meshes in the analysis. However, a small mesh size is generally preferred for objects to be made with AM, which can beneficially achieve fine details [11].

A second paradigm uses optimized or predesigned lattice structures to introduce multifunctional capabilities in AM parts. Lattices are repetitive arrays of trusses or plates, joined together to achieve unique properties [12]. An optimized lattice is achievable with constrained TO to realize periodic topological solutions as seen in [13,14]. Manufacturing constraints inherent in some AM techniques are yet to be considered in such algorithms; therefore it is difficult to realize structural features of the optimized solutions without the use of supports. Such supports could compromise the quality of the solution since it is impractical to remove supports from lattices. Until a suitable TO algorithm is developed that includes AM constraints for lattice solutions, it is beneficial to consider a more pragmatic approach for designing lattices for AM objects. There is a broad range of predesigned lattices, most of which are inspired by natural systems [15], that could be made with AM techniques. Understanding the multifunctional capabilities of these lattices would support a designer in the exploitation of AM's design space.

A growing number of publications are investigating the structural properties of predesigned lattices with AM techniques; some of these can be seen in [16–24]. Razmirez et al. [16] manufactured open cell lattices with an AM technique called electron beam melting. Yan et al. [17] fabricated gyroid lattices with selective laser melting (SLM) using stainless steel powder and observed an increase in yield strength as cell size was reduced. Contuzzi et al. [18] determined the compressive strength of reinforced pillar textile lattices while Smith et al. [19] investigated the effects of reinforcements on body centred cubic lattices. Yan [20] studied the properties of aluminium lattices at different volume fraction and showed that the micro-hardness and compressive strength decreases with increase in the cell size. Mullen et al. [21] used SLM to produce titanium lattices to support bone growth. McKown [22] investigated the buckling and impact behaviour of body centred cubic lattice and observed contrasting modes of failure for different configurations of the lattice. Vesenjak et al. [23] observed that lattices with circular cells had better energy absorption than quadratic cells, both of which were manufactured with selective laser sintering. Generally, lattices manufactured with AM techniques are self-supporting making them suitable as support structures when building other AM objects. Hussein et al. [24] showed this with gyroid and diamond lattices. In order to enhance the quality of parts, a compromise is needed between the lattice cell sizes and volume fraction of the support. Exploitation of lattices with the design freedom offered by AM relies on the availability of design methods and software tools, which are still limited for AM. Guo and Leu [25] called for conceptual design methods, novel CAD systems and simulation capabilities to address this need. A first step to developing lattice design methods and systems is the selection of appropriate geometry representation schemes, since the capabilities of commercial computer-aided design (CAD) packages in representing lattices are limited.

Conventional (CAD) tools are not efficient at representing lattice structures due to the underlying methods by which they handle and manipulate geometry. The main geometry representation schemes are boundary representation [Brep], constructive solid geometry [CSG], spatial subdivision, medial surface representation and procedural representation [26]. B-rep and CSG, the most common of these, are implemented in CAD systems [26,27]. B-rep stores the features of an object as a series of boundaries defined by vertices, edges and surfaces [28]; while CSG represents the object as Boolean and translational operations between a series of simpler objects [29]. Definition of basic elements in both schemes are cumbersome for highly complex objects. In a lattice structure, the topology of a single truss or plate is relatively simple, however, the sheer number of duplicates required to represent an entire lattice structure could exceeds the capabilities of current CAD systems. Also, B-rep and CSG do not contain information about the interior region enclosed by their primitive constituents. However, such information is becoming essential as AM techniques are developed for objects with multiple materials. Spatial subdivision schemes or volumetric representation could potentially resolve some of these issues; examples include surfrel [30], ray-rep [31], and voxel [32]. Little work has been carried out to date to apply these schemes to solving design issues in AM. Chen and Wang [33] used an adaptive contouring method based on a spatial scheme to repair erroneous B-rep models. Similarly, the voxel representation scheme can be used to embed a vast range of lattice topologies in complex three dimensional objects.

There is a need, therefore, for easy to use, computationally efficient and geometrically flexible methods of generating lattice structures to take advantage of the extension in lattice manufacturing enabled by AM. This paper proposes a novel, voxel-based method of generating uniform or graded lattices that can conform to an arbitrary external geometry via tessellation and trimming. An artefact of such trimmed structures is the existence of weak, unconnected structural features at the boundaries of the domain; which could initiate failure in service. A method of addressing this problem by generating a surface lattice, or net-skin, from the voxel model of the trimmed structure and its domain is also described in the paper.

The next section of the paper discusses some of the general issues concerning the generation of conformal lattice structures. This is followed by descriptions of the proposed methods of generating lattices, trimming to an external geometry, creating a net-skin and functionally grading the lattice. Numerous examples are used to demonstrate these methods.

2. Constructing lattice structures

Lattice structures are composed of repeating elements tessellated across a domain. Consider Fig. 1a, which shows an elementary unit, usually called the 'unit cell'. Tessellating eight units of this cell in the x and four in the y direction gives the lattice shown in Fig. 1b. The lattice structure in this case can be seen to be rectangular. It is often desirable to fit a tessellated lattice to a much more complex domain. The manner in which this is achieved influences the properties of the lattice. Consider the arbitrary domain shown in Fig. 2a. The lattice can either be swept to fit within the domain (Fig. 2b), mesh and mapped into the domain (Fig. 2c), or the domain can be used as a template to trim a tessellated lattice structure as seen in Fig. 2d. Regions of the lattice residing on the interior of the domain are retained whilst others are removed.

When constructing a swept lattice (Fig. 2b) an alignment of the lattice structure to the boundaries of the shape is enforced. Such structures have been studied extensively to improve strength to weight ratio, using, as in [34–36]. The fundamental issue with the sweeping method is that the properties of the structure will inevitably become a function of the curvature of the surface, with the unit cell deviating from its original properties. This method will fail entirely on parts with tight surface curvature. Meshed lattices (Fig. 2c) represent those generated with a meshing scheme similar to those used to model domains during a finite element analysis (FEA). The field of FEA is well developed and sophisticated meshing algorithms are used to discretise a domain prior to analysis [37,38]. A possible approach to construct lattice structures, then, is to use the volumetric mesh generated in the meshing stage of FEA. For example, the edges of the FE mesh elements can be used as the

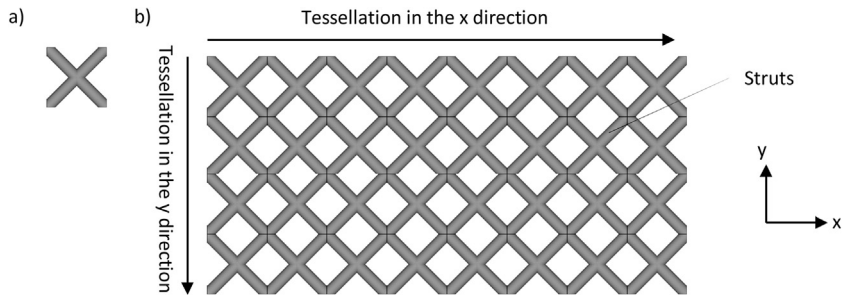

Fig. 1. A two dimensional lattice structure generated by tessellating a unit cell along x and y axes: a) Unit Cell b) Lattice structure.

struts of a lattice structure. The main limitation of this approach is that the emergent structure will be distorted, this being dependent on the external geometry of the object geometry as in the swept lattice, again deviating from the original properties of the lattice. The range of cell types generated by this method is also limited. Trimming tessellated lattices (Fig. 2d) with the domain is a better approach if the properties of the unit cell are to be largely retained. This is akin to a Boolean intersection of the domain and a tessellated lattice structure. The main advantage of the trimming method is that it is robust, since it is equally applicable to both simple and complicated cells and domains. The main disadvantage is the emergence of weak boundaries, where the trimmed unit cells lack support. A possible solution to this problem is to provide a solid skin along the boundaries, as shown in Fig. 3; however, this will limit access to the lattice, which may be detrimental for certain applications, hinder post processing (e.g. unprocessed material removal) and potentially add unnecessary weight. There may also be an aesthetic reason to have the lattice structure visible.

A small number of publications describe methods to embed trimmed lattices in three dimensional domains. Pasko et al. [39] used periodic functions to model regular and irregular lattice structures. Representing lattice structures with functions is an efficient and robust approach to lattice generation, allowing dynamic control of topological features. However, a limited number of lattice topologies can be represented this way. Nguyen et al. [40] developed a method that fits a lattice to a domain by decomposing the domain into easily mapped regions. The method is based on B-rep and was shown to be suitable for lattices composed of truss-like structures but is yet to be proven for others types. Generating lattice structures via voxel models, however, is potentially more versatile than boundary or function representation and forms the basis of the method presented in this paper.

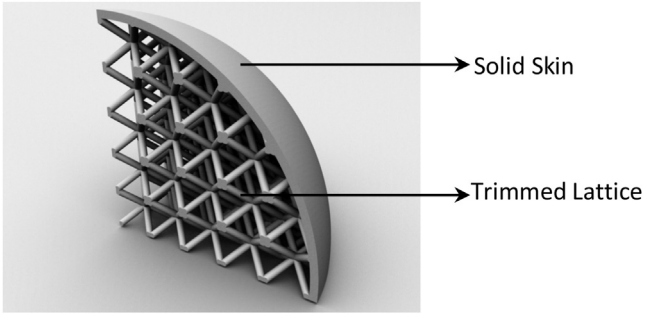

Fig. 3. A lattice structure partially covered by a solid skin.

3. Methodology

In this section the proposed method of creating conformal lattice structure will be described. Firstly, the voxel method of representing a unit cell is described, followed by the method of tessellating and trimming the unit cell to conform to an external geometry. A method of creating a net-skin to join the trimmed struts at the surface is then described to provide better structural integrity to the lattice structure. Finally a method of functionally grading a conformal lattice is described that enables more optimal lattice structures to be generated for specific applications.

3.1. Trimmed lattice structure

The starting point to the method is a geometric description of the lattice unit cell and the external geometry that will form the design domain for the lattice. The topology of the lattice domain and unit cell can conveniently be achieved with conventional CAD packages which represent the domain as a b-rep model, although almost any method of representing geometry can potentially be used. The domain and cell geometries are then voxelized using any

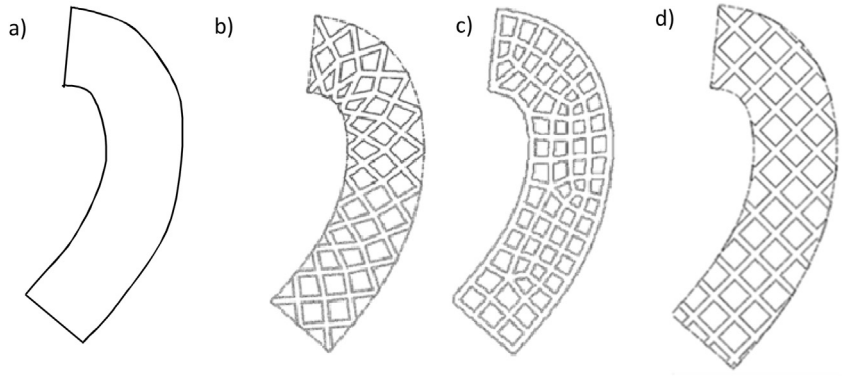

Fig. 2. Lattice generation methods: a) domain b) swept, c) meshed d) trimmed.

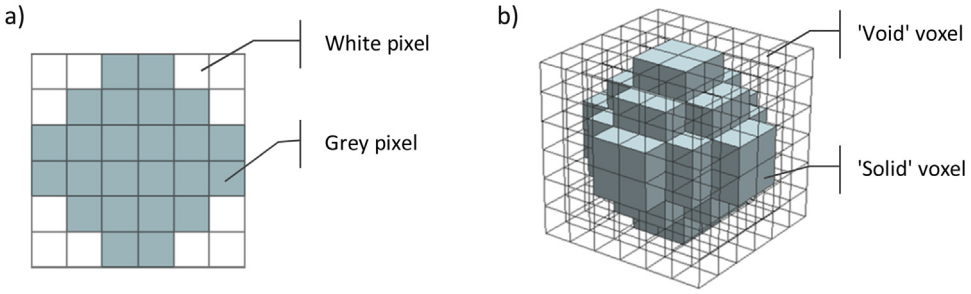

Fig. 4. a) Low resolution pixel image of a circle b) similarly low resolution voxel model of a sphere.

of a number of methods, such as the ones described in [41–43]. Voxel models define the volume of a shape in a bitwise manner, unlike the surface representation in most CAD software. Analogous to the rectangular pixels that represent a 2D image such as a bitmap, voxels are discrete blocks and, as such, voxel models have 'stepped' surfaces [44], as illustrated in Fig. 4. A voxel model is essentially a 3D matrix, with each element of the matrix representing a voxel. The simplest form of these models occurs when the voxels assume a logic value (i.e. one or zero), with one indicating a solid space and zero indicating void space [45].

A wide variety of lattice cells can now be built with additive manufacturing, with some examples shown in Table 1. The intrinsic structural properties of the lattice are built into the unit cell. This might include the dimensions, density, thicknesses of surfaces and diameter of struts, depending on the type of lattice cell. For example a helical diameter defines a spring-like strut. A helical diameter of zero defines a simple straight strut. Voxel models of the domain and unit cell models are considered as a series of 2D slices stacked in a manner that preserves the structural features of the original b-rep model. It should be noted that more voxels are required to represent a highly complex unit cell as compared to cells with simpler topologies. For example, the helical unit cells shown in Table 1 would require more voxels than the cells with straight members.

In order to generate a practically useful lattice structure, the dimensions of the unit cell should be much smaller than that of the domain, i.e. by at least an order of magnitude. To generate the trimmed lattice structure, operations analogous to a Boolean intersection are performed. The voxel model of the unit cell is first tessellated to encompass the domain. A 2D slice of a tessellated unit cell and its corresponding domain is shown in Fig. 5. For descriptive purposes, voxels set to one are referred to as being active and those with a value of zero are inactive. An entirely deactivated array is initiated to contain the trimmed lattice structure; with the same size as that of the domain and the tessellated structure. The domain array is checked, element-wise, and voxels in the trimmed structure are activated if the elements in corresponding positions of the domain and tessellated structure are active, as demonstrated with 'check a' in Fig. 6. Otherwise, voxels in the trimmed structure remain deactivated as illustrated by 'check b' and 'check c'.

The method runs both domain and tessellated structure through an AND gate – finding the maximum value and passing that on to the equivalent element of the trimmed structure array. The process is analogous to a Boolean intersection between the domain and the tessellated structure resulting in material retention in regions that had material in both the domain and the tessellated structure.

For some applications, inclining the lattice at an angle to the domain could enhance performance. Generating a rotated lattice can be achieved by rotating the domain prior to voxelization. Most CAD software is equipped with tools for such an operation. After the domain shown in Fig. 6 is rotated by 45° and voxelized, it is transformed to Fig. 7. The stepped boundaries are a consequence of the voxel representation and are less pronounced when finer voxel meshes are used. Also, the rotated domain would require a larger voxel model since the bounding box would increase. Tessellating and trimming gives the trimmed structure shown in Fig. 7, which could be considered as a rotated lattice with respect to the original domain. Rotating the voxelized domain or tessellated unit cell can achieve similar results but at a greater computational expense than the former strategy; and could introduce precision errors into the process. For a three dimensional domain, rotation of the domain is about the axes of inclination.

While voxelizing and tessellating the cell, it is important to predetermine the number of voxels required for both cell and domain. The ratio between the size of the unit cell and the domain determine the number of cells that can be tessellated across the domain and the cell size. For example, a cubic domain with 10 mm sides and modelled with 150 voxels will realize a trimmed lattice with 2 mm cell size if the unit cells are modelled with 30 voxels; reducing the number of voxels employed for the domain while fixing the cell voxels at 30 would increase the unit cell size. However, such a reduction could lead to a loss of geometric precision. This can also happen if the numbers of voxels are reduced in the cell. It is preferable to utilize the least number of voxels that capture the most essential cellular features and then determine the number of

Table 1
A selection of structure types integrated into the structure trimming process.

Cell	Kelvin/truncated octahedron				Cubic	Alternated cubic
Strut type	Triangular cross-section	Cylindrical	Wave	Helical	Helical	Cylindrical
Model View						
Example slice						

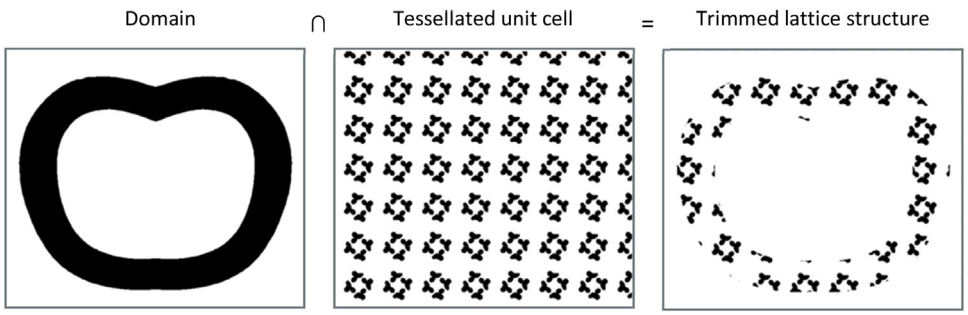

Fig. 5. Generation of a trimmed structure slice – analogous to a Boolean intersection.

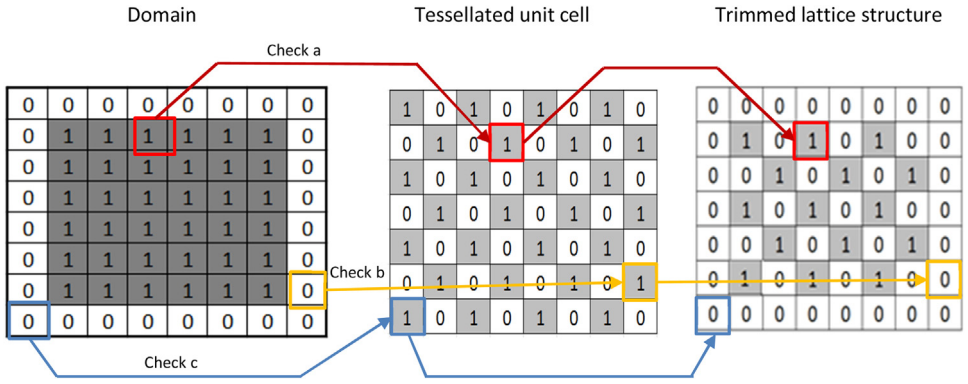

Fig. 6. Trimming structure to a domain – analogous to a Boolean intersection.

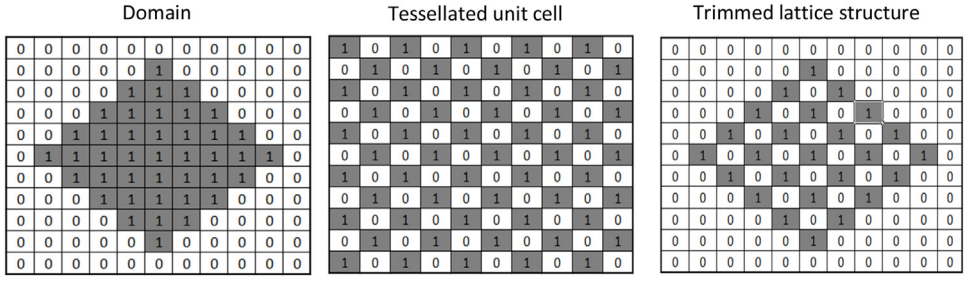

Fig. 7. Rotation of a Domain with respect to the tessellated lattice.

voxels required for the domain to achieve a specific cell size. This is illustrated for a mechanical part in the next section.

3.2. An example of a complex mechanical part embedded with a lattice structure

The capability of the voxel lattice generation method to conform to an external geometry is illustrated in Fig. 8, where regions of a mechanical part are latticed while the others remain solid. Regions of the part to be latticed are classified as the lattice domain and are coloured green in Fig. 8a. The other regions are therefore the non-design domain since they interact with neighbouring parts (Non-design is grey). The cell shown in Fig. 8b belongs to the family of minimal surfaces governed by mathematical functions, that has potentially useful properties for AM systems. The part experiences thermal cycling in-service. Embedding a lattice in the part offers a way to dissipate heat transferred (since lattices usually have a high surface area) while reducing the weight of the part. The lattice domain is reshown in its bounding box in Fig. 8c. To achieve a cell size of 4 mm with a cellular voxel array of 30; the size of the domain array was set at 489 by 459 by 88. This size is an approximate multiple of the bounding box. By changing the resolution of these arrays, the cell size is controllable, as mentioned previously. Using the voxel method, the part was latticed and manufactured via selective laser melting using an aluminium alloy. Fig. 8d shows the manufactured latticed plate after post processing.

It is important to keep the ratio of number of voxels in x, y and z axes to the ratio of the dimensions of the bounding box to avoid erroneous scaling of the part. The boundary of the trimmed lattice was left open to facilitate the removal of unprocessed powder, as illustrated in the enlarged views in Fig. 8d. The open-regions of the trimmed lattice could aid access to the internal regions of the lattice, which has a relatively high surface area compared to the solid regions of the part. Such an access is potentially useful, e.g. in extracting heat from the lattice via a fluid. A porous or net-skin is preferable in open-regions to support fluid transport while connecting hanging features. For weight-bearing applications, the use of net over solid skin minimizes the contribution of the skin to the weight of the part. No net-skin was included in Fig. 8 as the topography of the double gyroid, ensures features at the trimmed boundaries remain connected to the interior regions. Hanging features are therefore less evident for this type of lattice. However, skins are highly beneficial for strut–based lattices and also for other reasons, such as providing smooth fluid surfaces, connecting or

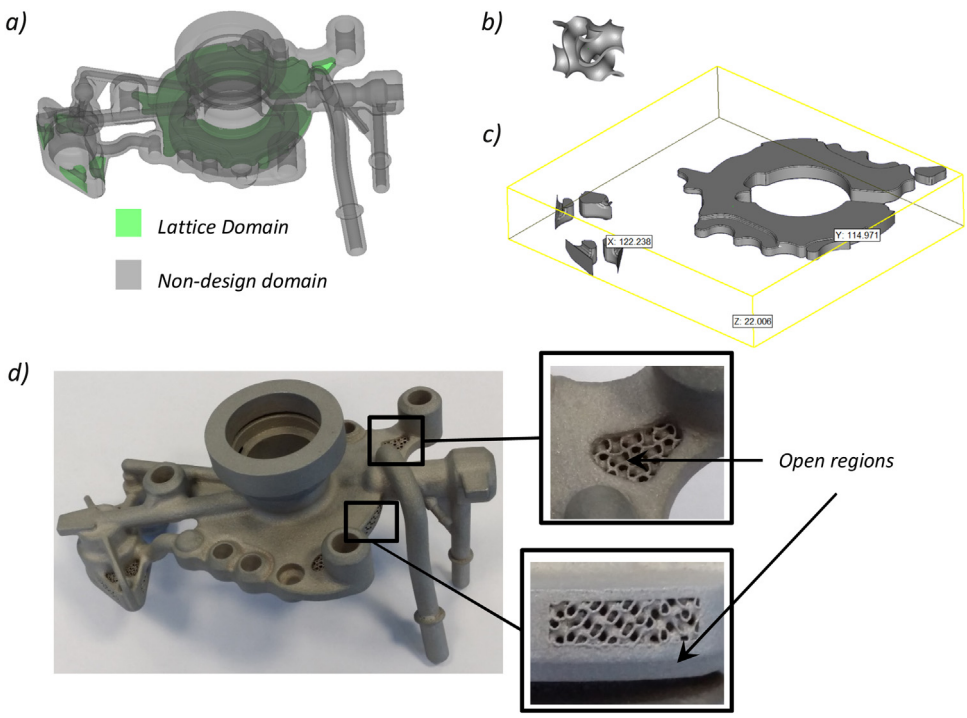

Fig. 8. A structural part embedded with a double gyroid lattice: (a) part geometry with lattice domain defined, (b) double gyroid unit cell, (c) lattice domain in bounding box, (d) latticed part manufactured with selective laser melting.

handling surfaces or for aesthetic reasons. A method of generating net-skins from the voxel model of the trimmed lattice is described in the next section.

3.3. Generating a net-skin via orthographic voxel projection

It is generally preferable to connect cut members at the boundaries of trimmed strut-based lattices as they provide a weak point in the lattice structure and are also aesthetically displeasing. Consider the trimmed lattice shown in Fig. 9a, with cut members at its boundary. Adding a net-skin to the trimmed lattice, results in the structure shown in Fig. 9b. The skin is a porous form of the solid skin, and can be considered as a 2D surface lattice conformal to the outer geometry, as shown in Fig. 9c. This type of skin improves the integrity of the structure whilst aiding post processing operations, such as the removal of powder. This section describes an approach to generate net-skins to connect the hanging features at the boundaries of a trimmed lattice.

It is possible to generate a net-skin from the voxel image of the trimmed lattice structure, A. This approach consists of the four stages shown in Fig. 10. The main idea of the method is to construct a voxel model of the net-skin by classifying voxels on a solid skin as either active or inactive. The first step is therefore to determine the voxel image of a solid skin. Once a solid skin has been created; projection of active voxels in the trimmed structure can be used to determine which voxels in the solid skin will be active. The selection of voxels to project is important in terms of computational efficiency. The set of voxels projected is therefore limited to those closest to the solid skin.

Firstly, the solid skin is defined by a layer of active voxels residing on the boundary of the voxel domain, as illustrated in Fig. 11 for a two dimensional case. Interior voxels (Fig. 11a) are deactivated by changing their value to zero; giving the hollow image of the solid skin shown in Fig. 11b. The boundary voxels can be determined by finding the number of active neighbouring voxels. All eight surrounding voxels of an interior voxel are active, in contrast to those for boundary voxels.

Assuming the voxelized domain is denoted by an array, B and the solid skin is C, this can be represented by

$$C_{ij} = \begin{cases} 1, \textit{if } \left(B_{ij} = 1\right) \wedge \left(f\left(B_{ij}\right) < 8\right) \\ 0, \textit{otherwise} \end{cases}$$

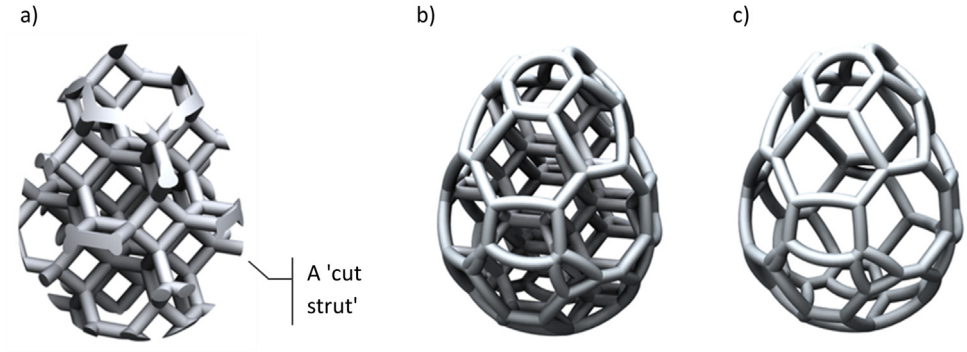

Fig. 9. a) Trimmed lattice structure, b) Lattice structure with net-skin, c) Net-skin.

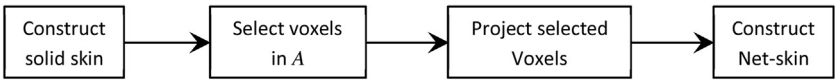

Fig. 10. The four stages in net-skin construction from the trimmed lattice structure.

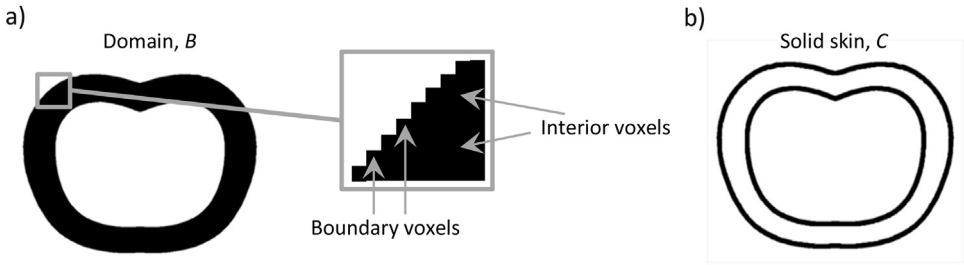

Fig. 11. A two dimensional solid of Net-skin for an arbitrary domain a) Voxelized Domain b) Solid skin.

where i and j are row and column indices, $f\left(B_{ij}\right)$ is an operation on B_{ij} to determine the number of active voxels connected to B_{ij}. MATLAB's "*bwperim*" function can be used to determine the solid skin. For a three dimensional case, 26 voxels are in the neighbourhood of a single voxel. Therefore, equation 1 becomes,

$$C_{ijk} = \begin{cases} 1, if\ \left(B_{ijK} = 1\right) \wedge \left(f\left(B_{ijk}\right) < 26\right) \\ 0, otherwise \end{cases}$$

where k is the index in the third dimension. Secondly, active voxels residing in A whose indices are close to those in C are selected for projection. This stage improves the computational efficiency of the method by limiting the number of voxels projected to this set, which is sufficient to form the net-skin. Active voxels in the grey region in Fig. 12 belong to this set and are used to form the net-skin. The size of the selected region would differ for different trimmed lattices, therefore, for a complete skin, the size of the region must be controlled adaptively until a suitable skin is formed. The selection of voxels in the concerned region consists of two operations. The first is the erosion of the voxelized domain B which is then subtracted from the trimmed lattice structure A. The eroded domain is used to remove active voxels at the center of the domain from a projection set. For a three dimensional lattice an orthogonal view of the trimmed lattice from the top, bottom, right or left would show that active voxels at the center of the domain are shielded by those closest to the solid skin. Therefore, these center voxels do not influence the net-skin.

If the domain shown in Fig. 11a is eroded, the image shown in Fig. 13a is created. The broken line illustrates the boundary of the un-eroded image. Subtracting the eroded domain from the trimmed lattice gives Fig. 13b. In order to increase the number of voxels projected, the domain is eroded a number of times to reduce the region subtracted from the trimmed structures. It is possible to construct an array D containing voxels to project from A and B using

$$D_{ijk} = \begin{cases} 1, if A - (B \ominus T) = 1 \\ 0, otherwise \end{cases}$$

where $\ominus$ is a morphological erosion of B with a structuring element T [46]. MATLAB's "*imerode*" function can be used to erode B. Active voxels in D are projected along orthogonal axes in the third stage. The projection is achieved by adding integral multiples of unit vectors along these axes to the indices of active voxels in D. The three dimensional Cartesian coordinate axis shown in Fig. 14a can be represented as the array in Fig. 14b. The first row in I is a unit vector along the positive x-axis while other rows represent those for the other axes. Multiplying any row in I with an integer, n and adding

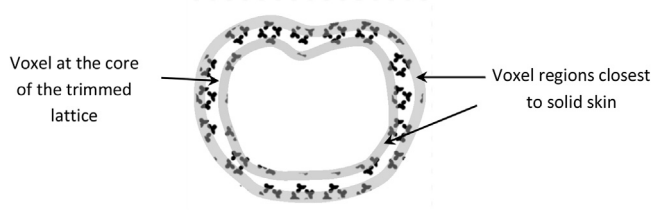

Fig. 12. An illustration of selected voxels for projection.

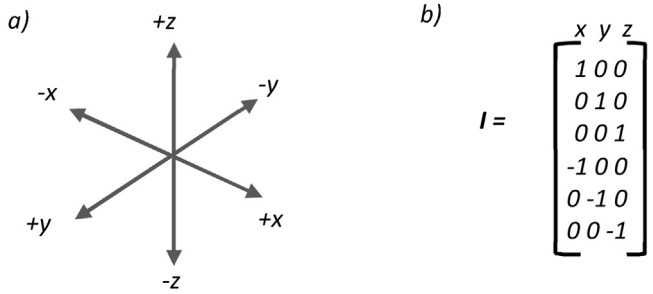

Fig. 14. Cartesian coordinates system a) graphical axes b) array **I** containing vectors of the coordinate axis.

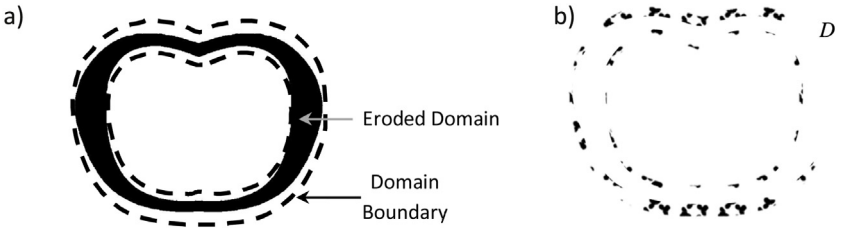

Fig. 13. Morphological operations to select active voxels in trimmed lattice: a) Eroding lattice domain b) Subtracting eroded domain from trimmed lattice.

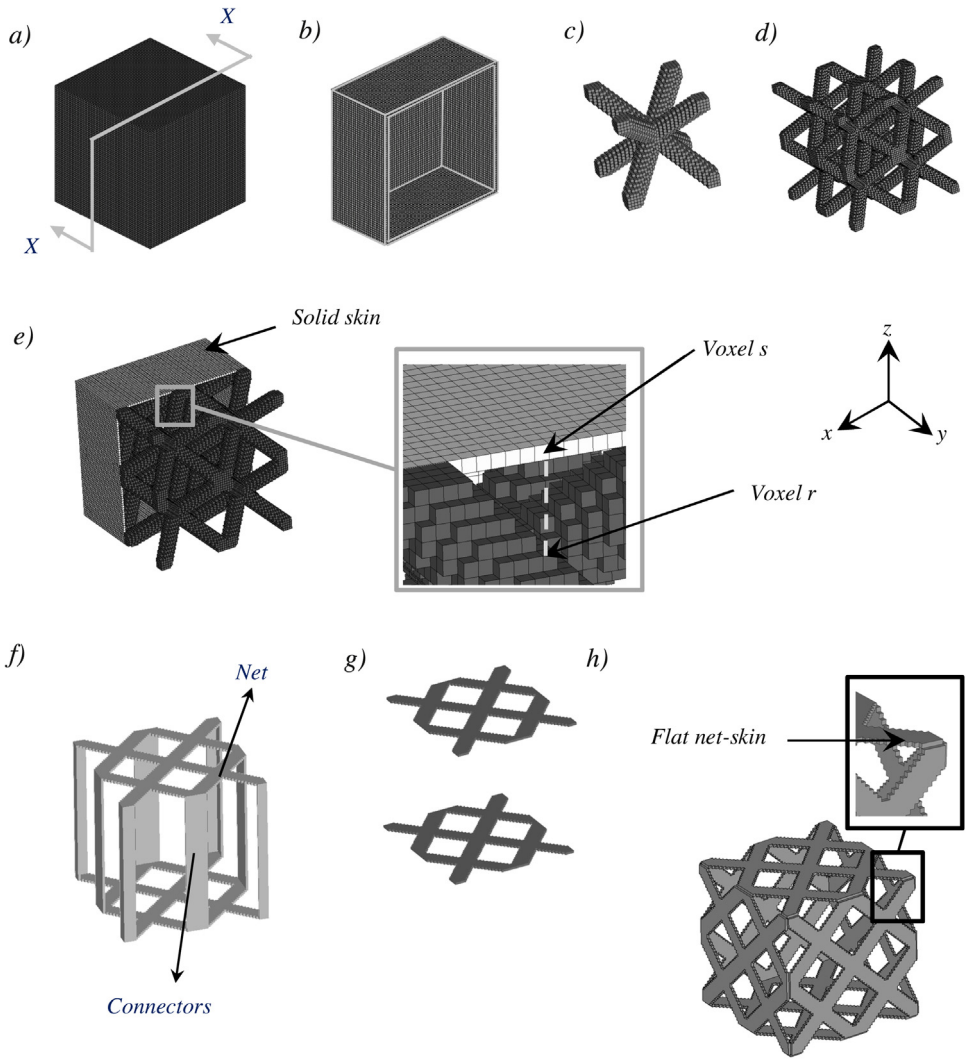

Fig. 15. First stage in extracting boundary voxels illustrated with a cube a) cube model b) Sectional view X–X of solid skin c) BCC Unit cell d) trimmed lattice e) lattice and solid skin f) projection along z-axis with edge voxels g) projection along z-axis without edge voxels h) Net-skin.

to a voxel in D translates that voxel in D. The value assumed by n determines the extent to which a voxel is projected. Mathematically, the projection of this single voxel along the x-axis is given by

$$V = [ijk] + n[100]$$

where V is a vector containing the indices of the new voxel. Similarly, active voxels in D are projected in the other five Cartesian directions, with a series of n values.

The value of n is selected to ensure all projections reach the boundary of the solid skin. Assuming $\boldsymbol{V}$ is the set of vectors derived from the projection, in the third stage, active voxels in C with indices absent in $\boldsymbol{V}$ are deactivated to achieve the net skin. Hence, the net skin array, E is determined by,

$$E_{ijk} = \begin{cases} 1, \textit{if } \left(B_{ijk} = 1\right) \wedge ([i, j, k] \in V) \\ 0, \textit{otherwise} \end{cases}$$

The selected active voxels may be projected in all 6 Cartesian directions depending on the topology of the domain. However, for some applications, it may be sufficient to project along a single axis. For example if the boundaries of a lattice domain are in contact with a non-design solid wall. In such a case, projection of active voxels in D along an axis that is not in contact with the other regions of an object is sufficient. Such an operation requires only two rows in I, i.e. the rows that reflect the positive and negative directions of the chosen axis. For a three dimensional case, consider the voxelized image of a cube shown in Fig. 15a; this has the solid skin in Fig. 15b. Embedding a 2 by 2 by 2 body centred cubic (BCC) unit cell (Fig. 15c) in the cube gives the geometry in Fig. 15d. The section view of the solid skin is superimposed on the lattice and shown in Fig. 15e. Projecting voxel 'r', as denoted by the broken line, along the z-axis, marks 's' for activation, as illustrated in the enlarged view. Projecting selected voxels along the z-axis axis in both positive and negative directions gives the image shown in Fig. 15f. The skin has two regions, the net and connectors. The connectors appear in the skin due to the inclusion of voxels at the edge of the lattice. Excluding these voxels gives the image in Fig. 15 g while combining projections in all three Cartesian coordinates gives the geometry in Fig. 15 h without the connectors. For objects with features that are generally orthogonal to each other, combining the portion of the skins in the three axes is relatively straight forward.

For highly curved or arbitrary shaped objects, it is difficult to derive an aesthetically acceptable skin with the connectors in the x, y and z axes. Also, combining the net-skin for three axes without unconnected regions can be challenging. It is therefore best to project along a single axis with the connectors included for such topologies. The two patterns formed on the net-skin still achieve

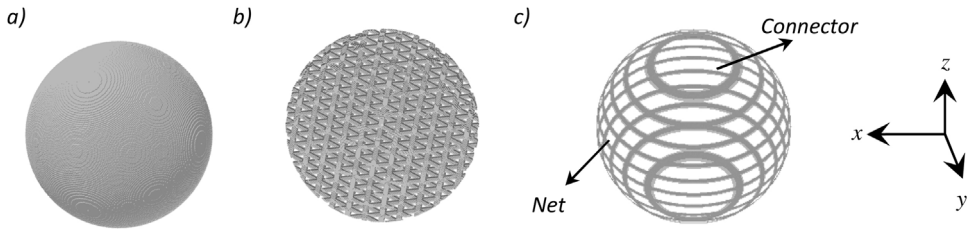

Fig. 16. Net skin generation from a unit cell and a sphere a) Sphere b) unit cell c) trimmed structure d) Net skin.

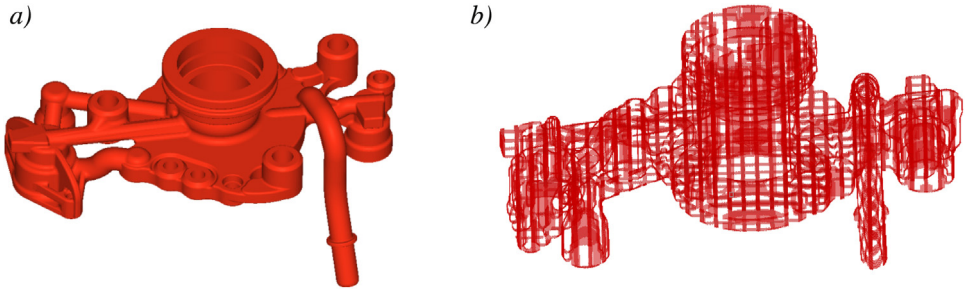

Fig. 17. A highly complex domain to illustrate net skin construction via voxel projection a) Complex geometry b) Net skin.

the aim of strengthening the boundary of the trimmed lattice by connecting hanging features in this region. This is illustrated for a spherical domain in Fig. 16a. A 4 by 4 by 4 BCC lattice is embedded in the domain to give Fig. 16b. Projecting selected voxels along the x-axis gives the net-skin shown in Fig. 16c. It is preferable to project in the direction of maximum surface area to maximize the appearance of the net and minimize connectors. The whole domain of the part shown in Fig. 8 is illustrated in Fig. 17a and the net-skin created via voxel projection for this domain is shown in Fig. 17b.

The method described above is suitable for a broad range of lattices, however, it might be less beneficial for surface, as opposed to truss, type lattices. Generally, the net-skin imposes an orthographic projection of the trimmed lattice on the solid skin. Considering the double gyroid and BCC cells as representative of surface and truss type lattices respectively; as viewed from the side, Fig. 18a and 18b shows the orthogonal views that would be seen. These represent the views transferred to the net-skin during the projection method. The regions that are identified as 'holes' are the porous parts of the net-skin. For the double gyroid, this region is relatively small compared to that of the BCC cell and can be seen to be comprised of a circular array. At higher densities, the circular regions become smaller and generating a net-skin with the voxel projection method simply achieves a near-solid skin. Theoretically, the net-skin method is applicable to the double gyroid and similar lattices, however, the benefits of such a skin could be minimal. Fig. 8c shows the side view of a double gyroid skin trimmed to a cubic domain. The net-skin generated by fitting the double gyroid to the complex domain shown in Fig. 17a is illustrated in Fig. 18d, with

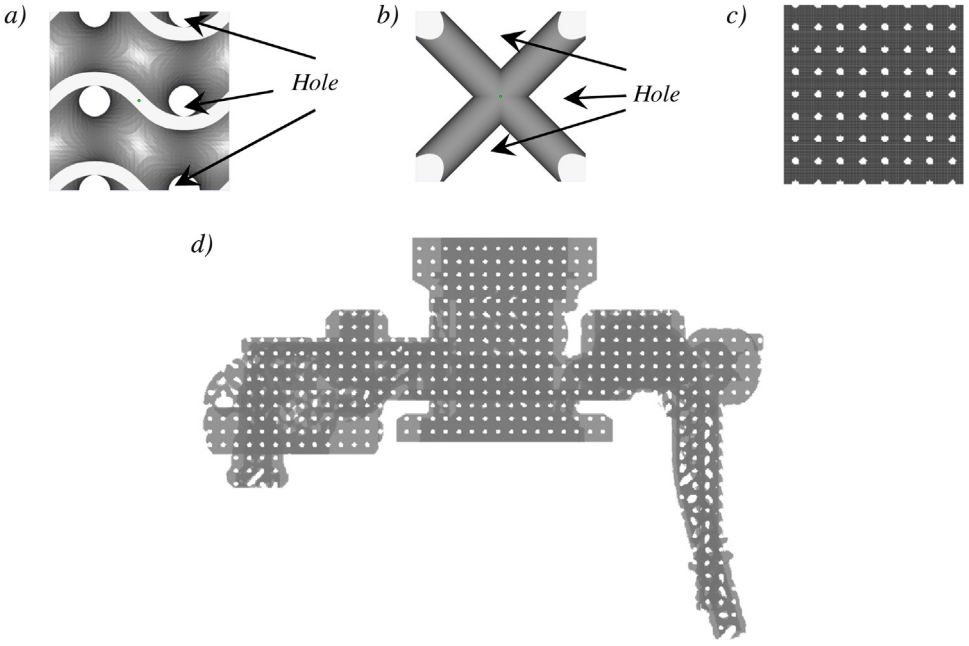

Fig. 18. Side views of double gyroid and BCC cells; double gyroid net-skins: a) Double gyroid cell b) BCC cell c) Front view of double gyroid net-skin derived from a 4 by 4 by 4 cubic domain d) Front view of double gyroid net-skin derived from a complicated domain.

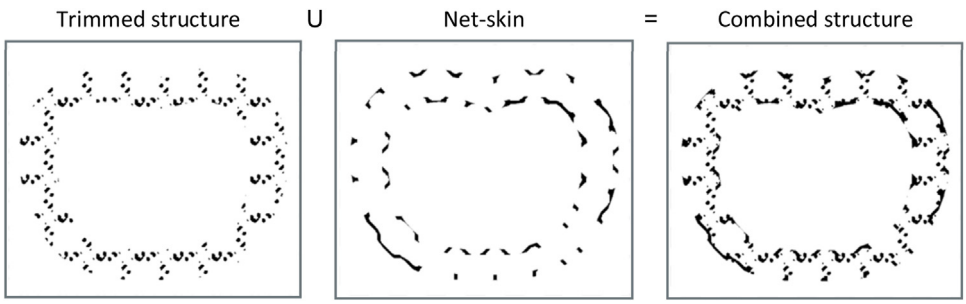

Fig. 19. Combining the trimmed structure and the net-skin with a union operation.

the skin showing lower porosity than that in Fig. 17b. This could be useful since it will offer more support to the trimmed lattice, but would potentially add unnecessary weight and restrict powder removal. An alternative method that can be used to generate the skin for surface type lattices is described in [47].

3.4. Combining structure elements

The generation of the trimmed structure and net-skin have been described in the previous sections. These exist as a pair of equally sized voxel arrays. To combine the two, a union operation is performed. A deactivated array is initialized with the same size as the trimmed structure and net-skin. Elements in the array are activated if the corresponding voxels in the trimmed structure or net skin are active. This overlays the net-skin on top of the trimmed structure, essentially uniting the two, as illustrated in Fig. 19.

3.5. Conversion to slice file

A possible route to manufacturing the combined structure created by the preceding steps is to convert the voxel model to a common geometric form for AM, such as STL, slice the resulting geometry and generate the machine file. However, the number of facets required to define the lattice as an STL file makes this computationally expensive and it is difficult to achieve trimmed lattices with highly refined details via this route. It is preferable, therefore, to construct the machine files directly from the voxel models. For some AM processes that utilize raster slices (e.g. jetting), each layer of the voxel model is well suited for writing the bitmap files required by the process. Other AM processes, such as selective laser sintering, require vector-based slice files. To convert the combined structure to a vector slice file, the outlines of the shapes represented by the array are traced. This is depicted on a small section of an individual strut in Fig. 20. In this case, a Matlab edge detection algorithm called '*bwboundaries*' was used, which is part of the image processing toolbox [48]. This algorithm searches for a boundary pixel (i.e. a solid pixel with at least one void pixel as a neighbour, and then searches its neighbours in a clockwise or anticlockwise fashion for the next solid pixel. This identifies an ordered series of boundary pixels and a separate series is generated for each individual shape on the bitmap. Each boundary pixel is defined by its row and column co-ordinates; these co-ordinates are multiplied by a scaling factor to convert to millimetre measurements. The scaling factor is dependent on the resolution of the bitmap slices input into the methodology: by default, one pixel equals 0.14 mm. Each boundary string can be written as a 'polyline', for example to a Common Layer Interface (CLI) file used by laser sintering machines [49].

3.6. Functional grading

The voxel based approach to lattice generation described above also allows the realization of functionally graded structures. This can be achieved by varying strut (or surface) geometry across a part. As mentioned previously, the voxel representation of the voxel structure considers matrices of ones and zeros to represent solid and void respectively. This is analogous to the black and white bitmaps used to describe the various methods in this paper: where black and white are used to denote solid and void. However, the method also has the capability to generate functionally graded structures by overlaying a greyscale image on to the domain. A simple example would be to overlay a gradient, as demonstrated in Fig. 21. Rather than the binary reasoning of 'if black, construct structure', the process becomes 'if a voxel is a particular shade of grey, construct a particular increment of structure'. A gradient is depicted with an 8-bit greyscale image that is overlaid over each domain slice with a Boolean intersection (to allow the assignment of different structure types to a range of values); the resultant domain slice is also a greyscale, 8-bit bitmap. To be able to generate a functionally graded structure, a library of cells is required that increment from one end of the graded range to another. Fig. 21 grades from a

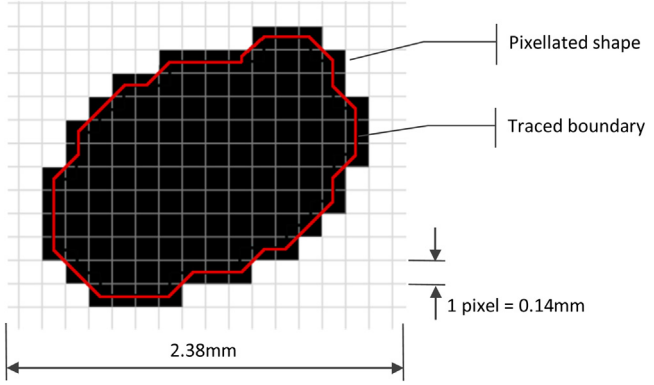

Fig. 20. Tracing the boundary of a strut in a voxelized domain.

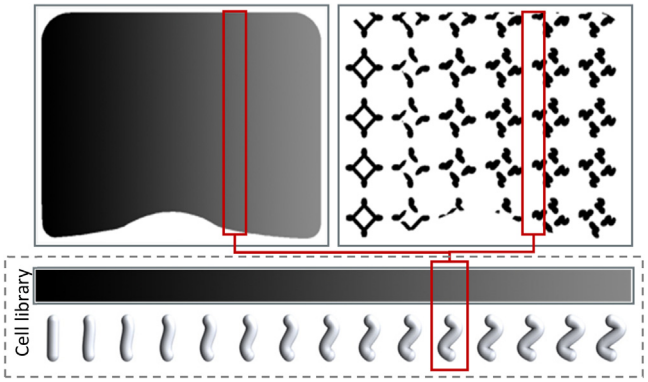

Fig. 21. Functionally grading a domain.

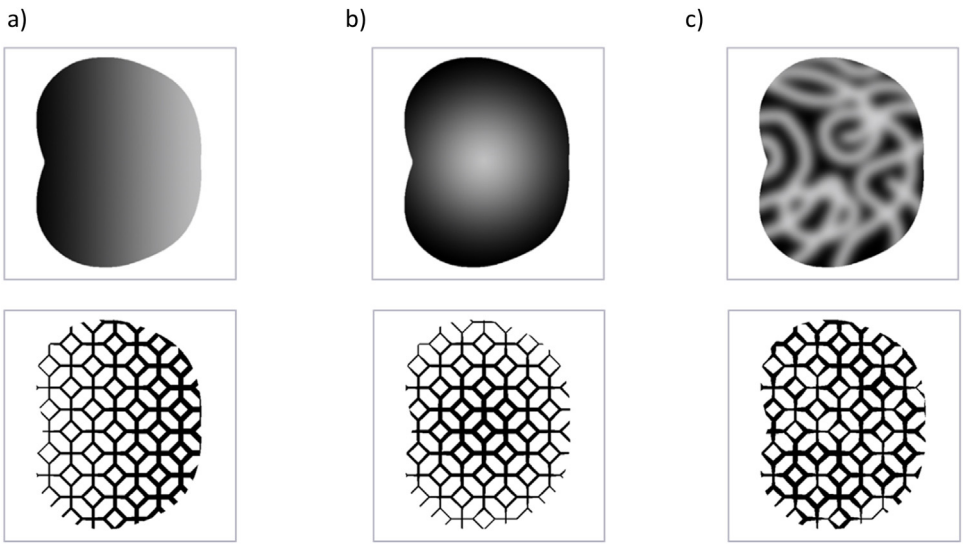

Fig. 22. Functionally graded structures generated from input gradients.

straight strut to a helix strut. Between the two extremes are helical struts that gradually reduce in amplitude to zero.

Another simple example of a suitable range would be to grade from a thin strut of a particular design to a thicker variant, as shown in Fig. 22. There are sixteen increments of the strut design; the difference in amplitude between increments is less than can be resolved by the bitmap resolution (i.e. less than the pixel width: 0.14 mm). In practice, any number of design increments could be implemented for functional grading. The overlaid image used to determine strut increment is an 8-bit bitmap. This is a 256 state greyscale range (values of 0–255), so when divided into the sixteen increments, sixteen shades of grey correspond to each strut increment. 8-bit greyscale bitmaps are a standard bitmap format and thus many programmes exist that can generate them. Many image-manipulation programmes can generate gradients that can be written as 8-bit bitmaps, making the generation of gradients a straightforward task. Any 8-bit bitmap can be used as input to generate functionally graded structures.

Once it can be seen that a graded lattice can be generated from a grey scale, the question of how to generate the grey-scale arises. In [50] it is shown how a stress analysis or a density based topology optimization can be used to generate a grey-scale that represents the optimal distribution of material density within a structure, based on a specific set of loads and constraints. In [50] a dithering method was used to generate a point cloud of varying density that was then used to generate a lattice that was functionally graded by varying the cell size through the structure. The method of generating graded lattice structures in this paper provides an interesting comparison by which a similar starting point, i.e. a grey scale representing the optimal distribution of material, can be used to generate a lattice with constant cell size but variations in individual cell definition through the structure to provide the functional grading. In both cases, however, a critical aspect of the success of the method is in mapping the performance of the cells in the lattice structure to material density (or stiffness) in the continuum structure used in the stress analysis (or topology optimization).

4. Discussion

It has been shown that voxel-based models are very versatile and efficient building blocks for constructing lattice structures. As these models represent the entire volume rather than a series of boundaries, as seen with B-rep models, there is no associated risk of generating manifold errors or invalid geometries. The speed of the trimming method is less dependent on the topology of the domain, but could be reliant on the complexity of the unit cell, which will now be considered. The Mobius knot domain shown in Fig. 23a is embedded with the unit cells shown in Fig. 23b and 23c. The simple cubic cell (Fig. 23b) has three cylindrical struts arrayed orthogonally to each other and connected at the center of the cell. The F_2BCC unit cell (Fig. 23c) is composed of 12 struts, four of which connect the

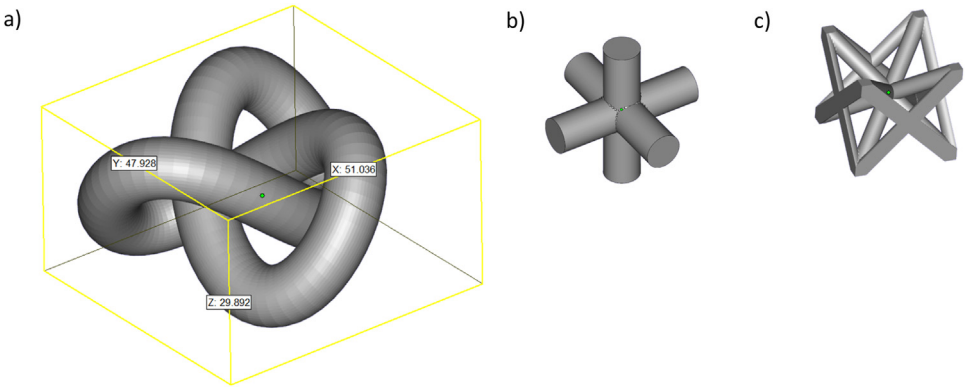

Fig. 23. Mobius knot Domain, simple cubic cell and F_2BCC: a) Mobius Knot b) Simple Cubic c) F_2BCC.

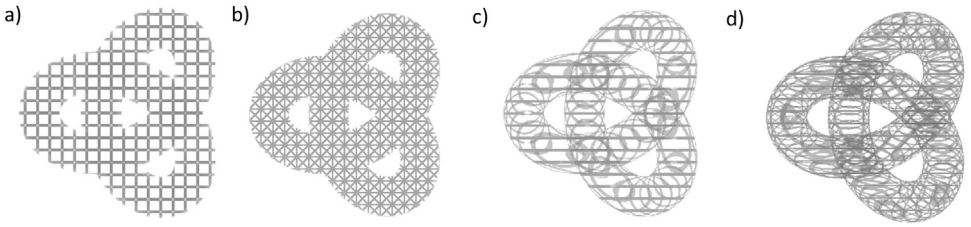

Fig. 24. Trimmed lattice and Net skins for simple cubic and F_2BCC in Mobius knot domain: a) Simple cubic b) F_2BCC c) Net-Skin Simple cubic d) Net-Skin F_2BCC.

Table 2
Breakdown of time spent at each stage of trimming two lattice types namely the simple cubic and F_2BCC.

Triangles	Simple Cubic		F2BCC	
	49442		171994	
	Mean Time (s)	Standard Deviation (s)	Mean Time (s)	Standard Deviation (s)
Domain Voxelization	69.34	0.22	69.34	0.22
Cell Voxelization	1.62	0.01	5.45	0.03
Trimming Structure	0.31	0.01	0.31	0.01
Net Skin	99.46	0.40	124.82	0.40
Structure and Net Skin	0.08	0.00	0.08	0.00
CLI file	9.84	0.07	24.53	0.43
Total	180.65	0.71	224.53	1.09

corners of different faces of the cell while another eight connect the corners sharing a face.

Half of the struts lying on the faces reside in adjacent cells when tessellated, therefore only half of these face struts exist in a single cell. The relatively higher number of struts and connectivity in the F_2BCC to that in the simple cubic cell suggest that F_2BCC has greater complexity. B-rep STL files of both cells and the domain were voxelized with a MATLAB voxelization script [51], and then subjected to the lattice generation steps described in the previous sections. The cells were voxelized with 30 by 30 by 30 voxels while the domain was voxelized with 511 by 480 by 299 voxels to bring the cell size to 3 mm. The trimmed lattices and their net-skins are shown in Fig. 24 with the time taken at each step shown in Table 2. Each of the steps was repeated 100 times with the mean time and standard deviation determined. A desktop computer with Intel(R) Core(TM) CPU i7-3820 (3.60 GHz) and 64GB of RAM on a 64-Bit windows operating system was used for all the steps to ensure a fair comparison.

It can be seen in Table 2 that the time for domain voxelization was equal for both cell type, this is logical as the cells were fitted to the same domain. However, the higher complexity associated with the F_2BCC requires it to be modelled with a greater number of triangles, a consequence of which is a higher voxelization time (three times) than that of the simple cubic cell. Much of the difference in voxelization time can be attributed to the time required to read and process the greater number of facets in an F_2BCC STL.

It can be seen that the same amount of time is spent in trimming the tessellated lattice and combining the trimmed lattice with the domain. The low standard deviation suggests the trimming step has high repeatability and is less dependent on the complexity of the cell. However, more time is spent constructing the net-skin for the F_2BCC lattice than for the simple cubic lattice, as the projection method is also dependent on the complexity of the underlying cell. Combining the trimmed structure with the net-skin required the same time for both cell types but CLI file generation differed significantly. It can be inferred from the table that the domain voxelization and net-skin generation steps determine the rate at which the lattice is generated, with only the latter being significantly affected by the complexity of the unit cell.

An additional advantage of the proposed method is that it is scalable; resolution can be easily changed to generate large parts at a low resolution (for 'draft quality' structures or for coarser AM processes) or small parts at a high resolution. Generating a large part (i.e. one requiring a large number of voxels to define the design domain) with high resolution could drastically increase the computational cost, hence, a 'drafting' capability is of significant practical benefit. High resolution should allow better representation of the lattice detail; however there is a minimum level of detail that can be achieved with an AM system (or any other manufacturing process), also the build volume of most AM machines is limited, hence, there are calculable limits to the maximum resolution needed and size of files. The voxel method is therefore well suited for these machines since the computational cost of the method is reasonable and can be matched to machine capablity.

In summary, this paper has attempted to address the need for a method to generate lattice geometries that is more compatible than current CAD methods with the structures realizable via AM. Characterizing the actual performance of such lattice structures is still an active area of research, and as the aim of the current paper was to present a novel method of lattice generation, the functional requirements of the trimmed lattices were not explicitly considered. The effects solid and net-skins on the functional performance of the lattices requires further research, however, [50] may be referred to for initial work in this field.

5. Summary and conclusions

This paper has described a novel, voxel-based method for the generation of trimmed lattice structures that is highly flexible in terms of both lattice cell type and external geometry. The method is suitable for highly complex lattices which can be realized via additive manufacture. A potential weakness of the method of creating conformal lattices by tessellation and trimming unit cells is the creation of poorly supported cell features at the surface, which is particularly problematic for certain strut-based lattices. This problem was addressed in the paper by proposing a method of constructing 'net-skins' from the trimmed voxel images. These skins connect hanging features at the boundaries of the trimmed lattices to minimize structural weakness in this region, while promoting the removing of unprocessed material after manufacture and reduced weight compared to a full skin. It was also noted that AM is capable of generating functionally graded lattices that

may behave more optimally than a uniform lattice under complex loading. Therefore, it was also demonstrated how the method could be extended to efficiently generate functionally graded lattice structures. Whilst the paper demonstrates the capability of the voxel-based process in generating the types of lattice enabled by AM, the most significant weakness of the method, namely that any part constructed has a stepped surface due to the discrete nature of the voxels, should also be recognised. This problem can be addressed by increasing the voxel resolution. This comes with a computational cost; however, when designing for AM, the surface finish limitations of the process provides a calculable limit beyond which further resolution is unnecessary.

Acknowledgements

This work was funded by the engineering and physical science research council (EPSRC) under grant EP/I033335/1 and Innovate UK under grant TS/J002518/1.

References

[1] M.P. Bendsoe, sigmund, O, in: Topological Optimization: Theory, Methods and Applications, Springer-Verlag, Berlin, 2004, p. 370.
[2] M. Abdi, R. Wildman, I. Ashcroft, Evolutionary topology optimization using the extended finite element method and isolines, Eng. Optim. (2013), http://dx.doi.org/10.1080/0305215x.2013.791815.
[3] X. Huang, Y. Xie, Convergent and Mesh-independent Solutions for the Bi-directional Evolutionary Structural Optimization Method Finite Elements in Analysis and Design, 43, 2007, pp. 1039–1049.
[4] G.I.N. Rozvany, M. Zhou, T. Birker, Generalized shape optimization without homogenization Structural Optimization, 4, 1992, pp. 250–252.
[5] M.Y. Wang, X. Wang, D. Guo, A level set for method for structural topology optimization Computer Methods in Applied Mechanics and Engineering, 192, 2003, pp. 227–246.
[6] D. Brackett, I. Ashcroft, R. Hague, Topology Optimization for Additive Manufacture, 21st Solid Freeform Fabrication Symposium, 2011, p. 12.
[7] A. Aremu, I. Ashcroft, R. Wildman, R. Hague, C. Tuck, D. Brackett, The effects of bidirectional evolutionary structural optimisation parameters on an industrial designed component for additive manufacture, Proc. Inst. Mech. Eng., Part B: J. Eng. Manuf. 227 (6) (2013) 794–807, http://dx.doi.org/10.1177/0954405412463857.
[8] B.M. Wood, Introduction to additive manufacturing Design and Manufacture of Plastic Components for Multifunctionality, 00005-3, 2016, pp. 171–204, http://dx.doi.org/10.1016/B978-0-323-34061-8.
[9] J.A. Madeira, H.C. Rodrigues, H. Pina, Multiobjective topology optimization of structures using genetic algorithms with chromosome repairing, Struct. Multidiscipl. Optim. (2006), http://dx.doi.org/10.1007/s00158-006-0007-0.
[10] J.A. Madeira, H.C. Rodrigues, H. Pina, Multi-objective optimization of structures topology by genetic algorithms, Adv. Eng. Softw. 36 (2005) 21–28.
[11] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C.B. Williams, C.C.L. Wang, Y.C. Shin, S. Zhang, The status, challenges, and future of additive manufacturing in engineering, Comput.-Aided Des. 69 (2015) 65–89.
[12] M.F. Ashby, The properties of foams and lattices, Phil. Trans. R. Soc. A 364 (2006) 15–30, http://dx.doi.org/10.1098/rsta.2005.1678.
[13] X. Huang, S. Zhou, Y.M. Xie, Topology Optimization of microstructures of cellular materials and composites for macrostructures, Comput. Mater. Sci. 67 (2013) 397–407.
[14] X. Huang, A. Radman, Y. Xie, Topological design of microstructures of cellular materials for maximum bulk or shear modulus, Comput. Mater. Sci. 50 (2011) 1861–1870.
[15] L.J. Gibson, M.F. Ashby, Cellular Solids: Structure and Properties, Cambridge University Press, 1997.
[16] D.A. Ramirez, L.E. Murr, S.J. Li, Y.X. Tian, E. Martinez, J.L. Martinez, B.I. Machado, S.M. Gaytan, F. Medina, R.B. Wicker, Open-cellular copper structures fabricated by additive manufacturing using electron beam melting, Mater. Sci. Eng. A 528 (2011) 5379–5386.
[17] C. Yan, L. Hao, A. Hussein, D. Raymont, Evaluations of cellular lattice structures, manufactured using selective laser melting, Int. J. Mach. Tools Manuf. (2012) 32–38.
[18] N. Contuzzi, S.L. Campanelli, C. Casavola, Lamberti, Manufacturing and characterization of 18Ni Marage 300 lattice component by selective laser melting, Materials (2013) 3451–3468.
[19] M. Smith, Z. Guan, W.J. Cantwell, Finite element modelling of the compressive response of lattice structures manufactured using the selective laser melting technique, Int. J. Mech. Sci. 28–41 (2013) 67.
[20] C. Yan, L. Hao, A. Hussein, P. Young, J. Huang, W. Zhu, Microstructure and mechanical properties of aluminium alloy cellular lattice structures manufactured by direct metal laser sintering, Mater. Sci. Eng. A 628 (2015) 238–246.
[21] L. Mullen, R.C. Stamp, W.K. Brooks, E. Jones, C.J. Sutcliffe, Selective laser melting: a regular unit cell approach for the manufacture of porous, titanium, bone in-growth constructs, suitable for orthopaedic applications, J. Biomed. Mater. Res. B Appl. Biomater. 89B (2009) 325–334, http://dx.doi.org/10.1002/jbm.b.31219.
[22] S. Mckwon, Y. Shen, W.K. Brookes, C.J. Sutcliffe, W.J. Cantwell, G.S. Langdon, G.N. Nurick, M.D. Theobald, The quasi-static and blast loading response of lattice structures, Int. J. Impact Eng. 35 (2008) 795–810.
[23] M. Vesenjak, L. Krstulovic-Opara, Z. Ren, Z. Domazet, Cell shape effect evaluation of polyamide cellular structures, Polym. Test. 29 (2010) 991–994.
[24] A. Hussein, L. Hao, C. Yan, R. Everson, P. Young, Advanced lattice support structures for metal additive manufacturing, J. Mater. Process. Technol. 213 (2013) 1019–1026.
[25] N. Guo, C. Leu, Additive manufacturing: technology, applications and research needs</it>, Front, Mech. Eng. 8 (3) (2013) 215–243.
[26] C.M. Hoffmann, Solid modelling, in: J.E. Goodman, O'Rourke (Eds.), CRC Handbook on Discrete and Computational Geometry, CRC Press, Boca Raton, FL, 2004, p.1560.
[27] D.E. LaCourse, Handbook of Solid Modelling, McGraw-Hill, 1995, 2016.
[28] M.M.M. Sarcar, K.M. Rao, L. Narayan, Computer Aided Design and Manufacturing, PHI Learning Pvt, 2008, p.728.
[29] I. Stroud, H. Nagy, Solid Modelling and CAD Systems: How to Survive a CAD System, Springer Science & Business Media, 2011, p.689.
[30] M.W. Jones, J.A. Baerentzen, M. Sramek, 3D distance fields: a survey of techniques and applications, IEEE Trans. Vis. Comput. Graphics 12 (4) (2006) 581–599.
[31] M. Pauly, R. Keiser, L.P. Kobbelt, M. Gross, Shape modelling with point-sampled geometry, Proceedings of ACM SIGGRAPH 2003, Computer Graphics Proceedings Annual Conference Series (2003) 641–650.
[32] A. Kaufman, An algorithm for 3D scan-conversion of polygons, in: Eurographics'87, Elsevier Science Publishers, 1987.
[33] Chen Y. Wang, Regulating complex geometries using layered depth-normal images for rapid prototyping and manufacturing, Rapid Prototyp. J. 17 (4) (2013) 253–268.
[34] H. Wang, Y. Chen, D.W. Rosen, A Hybrid Geometric modelling method for large scale conformal cellular structures, 25th Computers and Information in Engineering Conference, Parts A and B vol. 3 (2005) 421–427.
[35] C. Chu, G. Graf, D.W. Rosen, Design for additive manufacturing of cellular structures, Comput.-Aided Des. Appl. 5 (5) (2008) 686–696.
[36] D.W. Rosen, Computer-aided design for additive manufacturing of cellular structures, Comput.-Aided Des. Appl. 4 (5) (2007) 585–594.
[37] G.E. Farin, Curves and Surfaces for CAGD: a Practical Guide, 5th ed., Academic Press, 2002.
[38] J. Fish, T. Belytschko, A First Course in Finite Elements, John Wiley & Sons, 2007.
[39] A. Pasko, O. Fryazinov, T. Vilbrandt, P. Fayolle, V. Adzhiev, Procedural function-based modelling of volumetric microstructures, Graphical Models 73 (5) (2011) 165–181.
[40] J. Nguyen, S. Park, D.W. Rosen, L. Folgar, J. Williams, Conformal lattice structure design and fabrication, 22nd Solid Freeform Fabrication Symposium (2012) (p. 24).
[41] A. Karabassi, G. Papaioannou, T. Theoharis, A depth duffer based voxelization algorithm, J. Graphics Tools 4 (4) (1999) 5–10.
[42] S. Fang, H. Chen, Hardware accelerated voxelization, in: Volume Graphics, 2000, pp. 301–315.
[43] S. Thon, G. Gesquiere, R. Raffin, A Low Cost Anatialiased Space Filled Voxelization of Polygonal Objects, International Conference Graphion, 2004.
[44] V. Chandru, S. Manohar, Voxel-based modelling for layered manufacturing, IEEE Comp. Graphics 15 (no. 6) (1995) 42–47.
[45] A.E. Kaufman, Volume visualization of the ascending thoracic aorta using isotropic MDCT data: protocol optimization, ACM Comput. Surv. (CSUR) 28 (no. 1) (1996) 165–167.
[46] I.M. Pitas, Digital Image Processing Algorithms and Applications, John Wiley & Sons, New York, 2000 (p. 419).
[47] A.O. Aremu, I. Maskery, C.T. Tuck, I.A. Ashcroft, R. Wildman, R. Hague, Effects of net and solid skins on self-Supporting lattice structures, Challenges in Mechanics of Time Dependent Materials, Volume 2: Proceedings of the 2015 Annual Conference on Experimental and Applied Mechanics (2015) (p. 104).
[48] R.C. Gonzalez, R.E. Woods, S.L. Eddins, Digital Image Processing Using MATLAB, Pearson Prentice Hall, New Jersey, 2004.
[49] Common Layer Interface (CLI): Version 2.0 Specification.
[50] D.J. Brackett, I.A. Ashcroft, R.D. Wildman, R.J.M. Hague, An error diffusion based method to generate functionally graded cellular structures, Comput. Struct. 138 (2014) 102–111.
[51] J. Aitkenhead, Mesh voxelization, in: MATLAB Central File Exchange, 2013, Retrieved May, 2016.