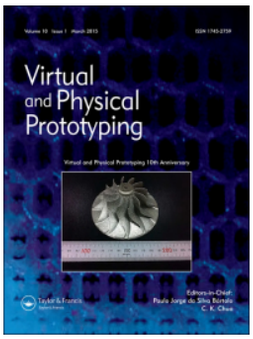

Virtual and Physical Prototyping

ISSN: 1745-2759 (Print) 1745-2767 (Online) Journal homepage: www.tandfonline.com/journals/nvpp20

Voxelization and one-dimensional lattice structures for industrial components using function representation

Antonio Bacciaglia, Alfredo Liverani & Alessandro Ceruti

To cite this article: Antonio Bacciaglia, Alfredo Liverani & Alessandro Ceruti (2025) Voxelization and one-dimensional lattice structures for industrial components using function representation, Virtual and Physical Prototyping, 20:1, e2526032, DOI: 10.1080/17452759.2025.2526032

To link to this article: https://doi.org/10.1080/17452759.2025.2526032

Published online: 10 Jul 2025.

VIRTUAL AND PHYSICAL PROTOTYPING
2025, VOL. 20, NO. 1, e2526032 (25 pages)
https://doi.org/10.1080/17452759.2025.2526032

Taylor & Francis
Taylor & Francis Group

Voxelization and one-dimensional lattice structures for industrial components using function representation

Antonio Bacciaglia [a], Alfredo Liverani [a] and Alessandro Ceruti [a,b]

[a]Department of Industrial Engineering (DIN), University of Bologna, Bologna, Italy; [b]CIRI Aerospace, University of Bologna, Bologna, Italy

ABSTRACT

This paper presents a scalable, open-source method for designing strut-and-node lattice structures for industrial applications, including uniform and graded lattices. Traditional computer-aided design tools struggle with lattice structures with high complexity, prompting the need for alternative approaches. While function representation techniques are commonly applied to triply periodic minimal surface lattices, their use for strut-and-node lattices has been limited. The proposed method defines the unit cell geometry using function representation primitives to model cylindrical struts and spheres, followed by isosurface triangulation and spatial replication within a voxelized design space. To showcase its practical application, two case studies are presented in which industrial components are filled with uniform and graded lattice structures using the newly developed model. The paper includes a comprehensive analysis of the computational cost of the approach. Furthermore, the study evaluates the geometric accuracy and quality of the generated lattice, highlighting their suitability for lightweight design in additive manufacturing. This method eliminates the need for boundary representations of lattice structure models, leading to more efficient data handling. The results of this research have broad implications for developing 3D components optimised for additive manufacturing. The approach targets industrial use, enabling fast, efficient design of complex, lightweight geometries.

ARTICLE HISTORY
Received 14 April 2025
Accepted 21 June 2025

KEYWORDS
Lattice structures; computer-aided design; additive manufacturing; CAD file format; lightweight design

1. Introduction

Weight is a key parameter in the design of industrial components, particularly in the aerospace [1,2] and automotive sectors [3,4], where reducing weight plays a fundamental role in lowering fuel consumption and carbon emissions [5,6]. Lightweight, high-strength components, such as additively manufactured cellular structures, enable the development of more efficient transportation systems [7]. These bio-inspired structures offer exceptional strength-to-mass ratios, multifunctionality, enhanced energy absorption, and efficient material distribution [8–12]. Due to manufacturing constraints, cellular structures have not been as widely accepted as other architectured materials, such as metal foams and honeycomb structures, over the last few decades. Traditional production methods are often complex and costly, diminishing the advantages gained from weight reduction [13].

However, the development of Additive Manufacturing (AM) in the past years [14] has further accelerated the investigation of intricate lightweight geometries. The advent of AM enabled the creation of detailed designs that were previously impossible using traditional manufacturing methods [15–20].

Numerous cellular structure types, including foams (stochastic structures), honeycombs, and lattices (periodic structures), are described in the literature [21–24]. The lattices' regular unit cell arrangement is a characteristic of nonstochastic architectures. Both two- and three-dimensional unit cell arrangements are possible for Voronoi [25], Triply Periodic Minimal Surface (TPMS) structures [26–29], and 3D strut-and-node based lattices [30–32]. This research will focus on strut-and-node-based lattices for the rest of the discussion.

The design and representation of lattice structures present considerable challenges despite the structural advancements provided by additive manufacturing [33–35]. Conventional Computer-Aided Design (CAD) tools often struggle with the geometry complexity typical of lattice geometries, mainly when dealing with many facets and intricate topologies. This complexity can lead to inefficiencies in data handling, difficulties

CONTACT Alessandro Ceruti alessandro.ceruti@unibo.it Department of Industrial Engineering (DIN), University of Bologna, Viale del Risorgimento 2, Bologna, Italy

in mechanical simulations [36], increased computational costs, and limitations in design flexibility [37]. As a result, there is a strong and growing effort within the scientific community to develop alternative modelling methodologies – particularly open-source solutions – that can efficiently represent and manipulate lattice structures while ensuring compatibility with commercial software. These initiatives aim to complement existing AI-assisted tools [38,39] and make advanced lattice design more accessible and customisable.

Wang et al. [40] and [41] introduce a hybrid geometric modelling method that directly produces tessellated models in Standard Triangle Language (STL) format rather than generating complete solid models. This approach combines solid Boundary Representation (B-Rep) modelling [35], which is used to create individual unit trusses, with facet-based (tessellated) modelling. Instead of building an entire solid model, the unit trusses are converted into an STL (triangulated mesh) format [42,43] and assembled to form the final structure, eliminating the need for complex Boolean operations.

Azman et al. [44] propose an effective method for creating and storing lattice structure CAD models using a skeleton 1D model, improving AM efficiency and accuracy. However, this article does not describe how the transition is made from the 3D digital model, produced from the skeleton approach, to a sliced model (i.e. Gcode [45]) ready for additive manufacturing.

Chougrani et al. [46] introduce a lightweight triangulation technique, which significantly reduces the number of triangles needed to represent lattice structures for additive manufacturing, compared to traditional methods like Constrained Delaunay Triangulation (CDT) [47] and Marching Cubes Method (MCM) [48]. This new approach is not sensitive to object orientation and allows for separate control of accuracy and quality, enhancing the efficiency and feasibility of manufacturing complex lattice structures. However, this research does not deepen aspects such as file processing times, a critical aspect in industrial applications related to Time To Market (TTM) [49].

McMillan et al. [50] present the development of a Programmatic Lattice Generator (PLG) method, which reduces processing time by over 60% compared to traditional CAD software. It offers direct control over STL file quality and size. It enables the generation of complex lattice structures independent of designer proficiency, making it a valuable tool for design optimisation and process integration. However, this research does not include a step-by-step description of performing the intersections of all the struts. The obtained STL file suffers from holes and cracks that must be corrected in external software such as Meshlab [51].

Function Representation (FRep) [52,53], also known as implicit function representation, has emerged as a promising method for modelling complex geometries, including lattice structures [54]. FRep defines geometric shapes through continuous real-valued functions. This approach facilitates the representation of complex topologies without requiring explicit boundary definitions, thereby reducing data complexity and enhancing computational efficiency.

Tang et al. integrate lattice frame generation, geometric FRep, and voxelization to overcome the limitations of conventional CAD tools, facilitating the design and simulation of multiscale lattice geometries [55]. However, after the implicit function modelling, Tang et al. convert the 3D FRep lattice into a voxel model specifically to create 2D slice images for direct use in AM processes like Digital Light Projection (DLP) or Liquid Crystal Display (LCD) based Stereolithography (SLA), rather than generating a surface triangulated mesh file. Indeed, the STL file format is more general-purpose and adaptable for any AM process. In particular, it can be used even in the Fused Filament Fabrication (FFF) process to manufacture customised prototypes.

Fryazinov et al. [56] present a FRep modelling approach for multiscale, space-variant cellular structures. This methodology enables the procedural generation of complex, parameterised microstructures, allowing for dynamic spatial variations and direct rendering or fabrication. A procedural function-based modelling approach for volumetric strut-and-node lattices, which can be directly rendered and fabricated without relying on traditional polygonal or voxel-based representations, is presented in [57]. On the one hand, the FRep is widely adopted in the literature for TPMS unit cells, and efficient and open-source tools such as MSLattice are available to the end user [58]. However, even while commercial software, such as nTop [59], suggests the implicit function representation, FRep has not been published in open-source frameworks targeted to industrial applications for strut-and-node lattices. Nowadays, there is a lack of reliable and consolidated methodologies to represent and handle complex lattice structures in CAD systems.

This study proposes a novel framework called VOLFREP (VOxelization of Lattice structures and Function REPresentation) for designing lattice structures using FRep in conjunction with the voxelization of the design domain. This approach can efficiently generate complex lattice configurations by defining the base unit cell geometry as a wireframe model through FRep-based primitives and mathematical operations. Moreover, the voxelization [60,61] of the design domain speeds up the unit cell replication, and only active cells are considered in the lattice design. This approach differs slightly from

the kernel point modelling used by Tang et al. [55], although the objective is similar: to evaluate whether a 1D wireframe unit cell lies within the design volume to be filled. The subsequent triangulation and exportation as STL file of these FRep-defined geometries enables their practical application in AM processes, ensuring compatibility with standard fabrication workflows.

The limitations of traditional CAD tools in handling lattice structures are addressed, and a robust framework for designing and fabricating optimised lattice components is provided. Compared to existing literature contributions, a comprehensive and accurate cost breakdown analysis is included in the manuscript to demonstrate the tool's efficiency in industrial applications. The proposed method streamlines the design process and opens new avenues for developing lightweight, high-performance structures tailored to various industrial applications.

Thus, to summarise, the proposed framework aims to:

- Design uniform and grade lattices with cylindrical struts through wireframe models converted to 3D components through FRep;
- Include the FRep of spheres placed at the lattice joints for safe Boolean intersections and to avoid structure stress concentration;
- Use voxelization to speed up the unit cell replication;
- Provide a usable 3D digital model in STL file format to be compatible with all the AM processes;
- Be computationally efficient to deal with complex topologies typical of industrial applications;
- Provide a precise and comprehensive breakdown of the cost analysis to demonstrate the efficiency of the proposed methodology.

The structure of the paper is as follows: Section 2 briefly presents the VOLFREP framework and the methodology applied to design and tesselate strut-and-node lattice structures. Section 3 presents an industrial case study where the VOLFREP framework is used to obtain the 3D digital model of a lattice structure. Section 4 discusses the results obtained in this research, and Section 5 ends the paper with conclusions and prospects for future work.

2. Materials and methods

This section details the main steps of the proposed VOLFREP framework, which has been coded in MATLAB 2024b [62] environment to test it. Further implementation in other programming languages, such as C++ or Python, is possible, as well as in macros for commercial CAD packages. This tool is used to design lightweight strut-and-node-based lattices, as is shown in the graphic flowchart of VOLFREP (Figure 1).

2.1. Design domain voxelization

The first task of the VOLFREP framework is to compute the design volume, which the lattice structure will fill. The design domain is imported as an STL file and converted to a logical 3D matrix thanks to the voxelization. Voxel-based modelling outperforms standard 3D modelling methods in managing big datasets, keeping fine features inside 3D models, and accelerating geometric operations and manipulations such as rotations and Boolean operations [63]. Compared to more complex discretization techniques, voxelization seldom fails because of its simple unit cell structure.

To determine whether or not the 3D model occupies a single unit cell (voxel), the voxelization technique superimposes a 3D grid onto the 3D model. Each voxel is given a binary value to contribute to the population of a 3D logical matrix: 1 if the unit cell is inside the 3D model and zero otherwise. The user should set the hexahedral cells' voxel resolution for discretization. This value should be consistent with the unit cell dimension of the lattice structure the designer wants (as visible in Figure 1). In particular, at the end of the voxelization, each hexahedral element will be replaced by a strut-and-node lattice unit cell.

After setting the voxel grid, the approach initialises a logical matrix with all null values. The algorithm automatically carries out ray tracing [64] in the three primary directions (x, y, and z). The outcomes from each direction are combined to determine which logical matrix components should be activated. In this research, the *Mesh Voxelization* algorithm by Adam has been implemented [65]; thus, in Appendix A.1, in the STL2Voxel function, the VOXELISE function by Adma is called. For additional information about the voxel method implemented in this research, please refer to [60].

The percentage of active cells can be remarkably low for highly complex geometries, typical of industrial applications, that deviate significantly from a parallelepiped-shaped domain. By deactivating many hexahedral elements, the replication of the unit lattice cell is accelerated, as inactive voxels within the computational domain are disregarded.

Furthermore, thanks to the voxel-to-unit-cell association of the lattice, the design domain can be efficiently and quickly trimmed, as described in the study by Azman et al. in Section 4.4 (Case 1) [44].

2.2. Wireframe unit cell model

The framework described in this research can design and tesselate the strut-and-node lattices employing a limited unit cell portfolio, which will be enlarged in the future. Nowadays, the available unit cells are:

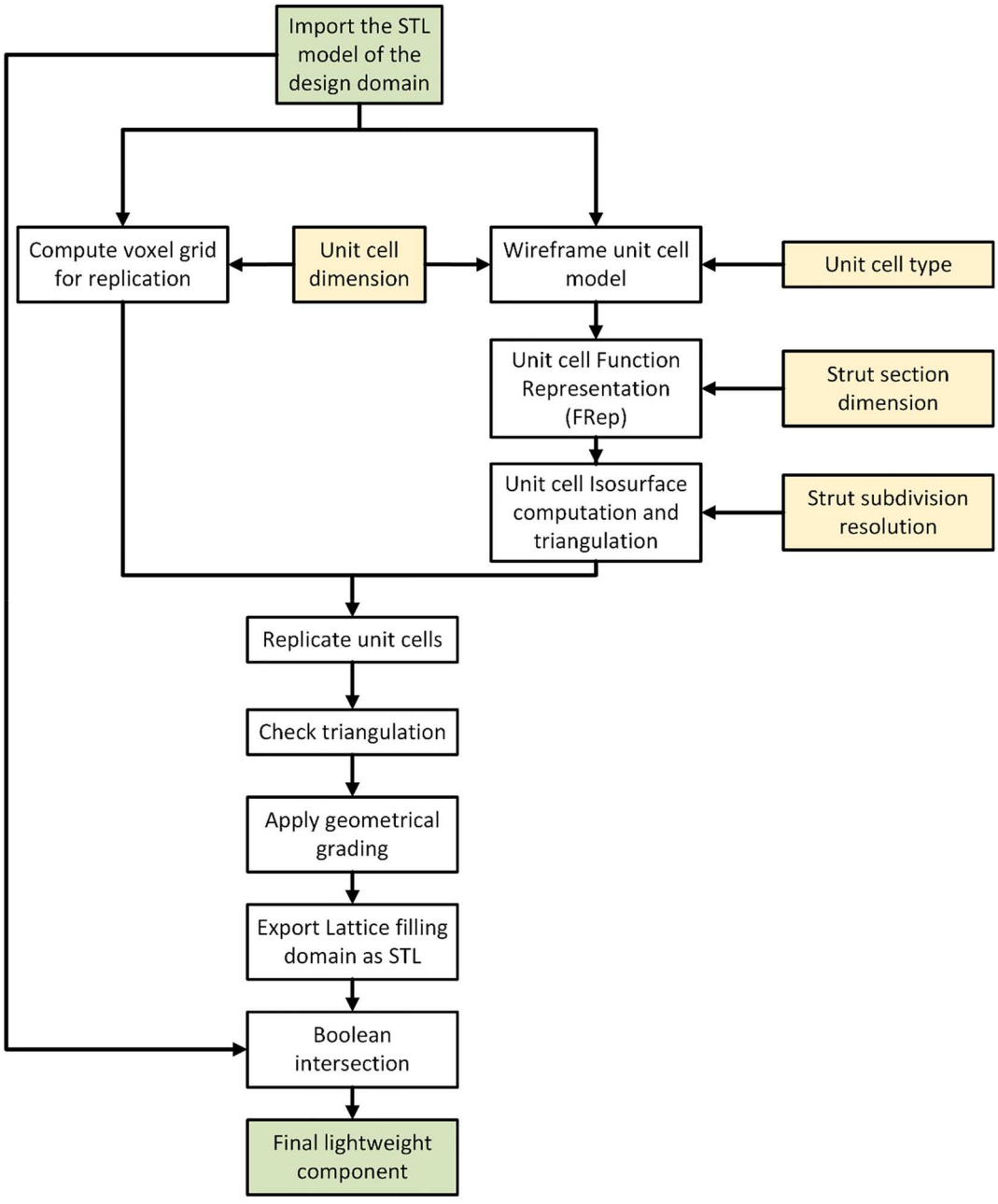

Figure 1. Graphic flowchart of the VOLFREP framework; the user's inputs are in yellow.

- Box: 8 vertices, 12 struts;
- Face Centred Cubic, FCC: 14 vertices, 36 struts;
- Body Centred Cubic, BCC: 9 vertices, 20 struts;
- Octet: 14 vertices, 36 struts;
- Octahedron: 6 vertices, 12 struts;
- Cube Vertex Centroid, CVC: 9 vertices, 8 struts.

Other shapes can be implemented similarly. The Maxwell criterion is a fundamental principle used to determine the mechanical behaviour of lattice structures, particularly in distinguishing between stretch-dominated and bending-dominated configurations [66]. Among the previous bullet list, the VOLFREP framework is capable of handling three stretch-dominated structures (FCC, Octahedron, and Octet) and three bending-dominated structures (Box, BCC, and CVC); the reader can refer to [67] for further information about the deformation mechanism of a cellular solid. The unit cell library currently implemented is visible in Figure 2.

A specific function, which pseudocode is visible in Appendix A.2, is designed to generate the node and connection matrices, respectively ***N*** and ***M***, for the selected unit cell type:

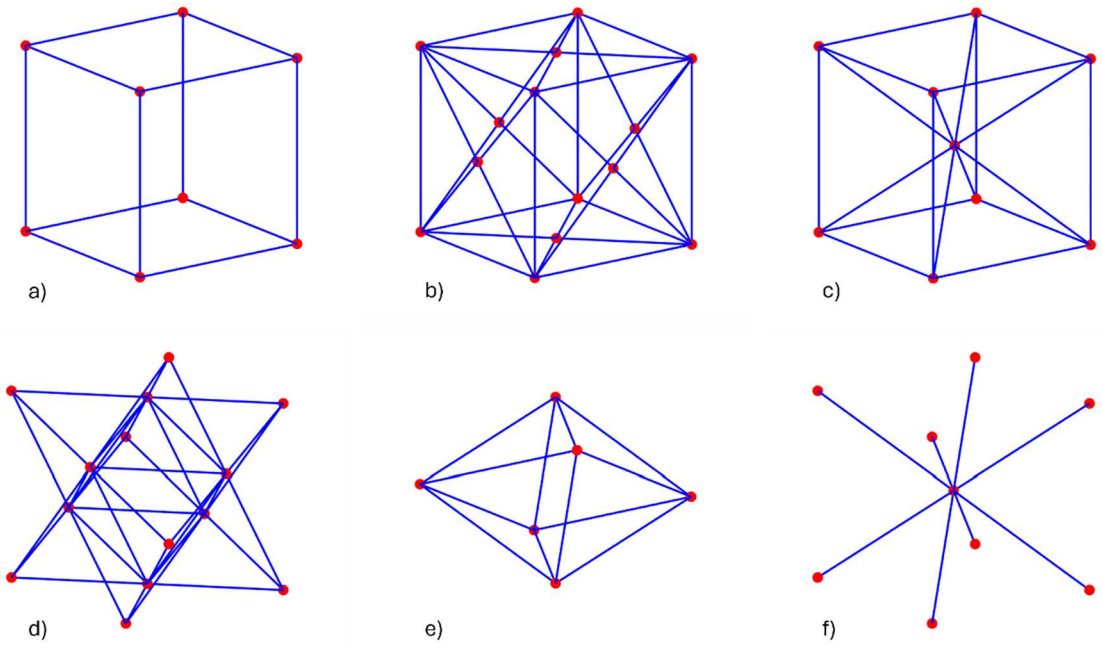

Figure 2. Unit cell library: (a) Box, (b) FCC, (c) BCC, (d) Octet, (e) Octahedron and (f) CVC.

- The nodes matrix $\boldsymbol{N}$ is an $n\times3$, where each row represents the (x, y, z) coordinates of a vertex in the unit cell. Each node is calculated based on the unit strut length L;
- The connectivity matrix $\boldsymbol{M}$ is an $m\times2$ matrix, where each row represents an edge between two nodes. Each row contains two indices referring to rows in $\boldsymbol{N}$.

2.3. Unit cell function representation

In the following step, the VOLFREP framework converts the 1D unit cell model into a 3D implicit model using the FRep approach. In this research, only struts with circular cross sections are modelled; however, the framework can be easily adapted to other section shapes. The implementation of this function is visible as pseudocode in Appendix A.3.

Given the strut section dimension in terms of cylinder radius, a dedicated function uses as inputs the matrices $\boldsymbol{N}$ and $\boldsymbol{M}$. For each connection in $\boldsymbol{M}$, the function returns the FRep to describe a solid cylinder (including the top and bottom caps). This strut, defined by its centerline starting at $\boldsymbol{A}$ $(x_A,\ y_A,\ z_A)$ and ending at $\boldsymbol{B}$ $(x_B,\ y_B,\ z_B)$ with a radius r, according to Figure 3, is composed by:

- the lateral surface, which is defined as a geometric entity where all the points are within a distance from the centerline;
- the top and bottom caps, which are defined as planes perpendicular to the centerline at points $\boldsymbol{A}$ and $\boldsymbol{B}$.

The dedicated MATLAB/OCTAVE function first defines the direction vector of the cylinder axis (Equation 1):

$$\hat{\boldsymbol{d}} = \frac{\boldsymbol{d}}{\boldsymbol{d}},\ \text{where}\ \boldsymbol{d} = (d_x,\ d_y,\ d_z) = (x_B - x_A,\ y_B - y_A,\ z_B - z_A) \tag{1}$$

Then, the code computes a projection function $t(x, y, z)$ to project each point onto the axis, as in Equation 2.

$$t(x, y, z) = \frac{(x - x_A)d_x + (y - y_A)d_y + (z - z_A)d_z}{d_x^2 + d_y^2 + d_z^2} \tag{2}$$

The function fixes t to lay between 0 and 1, thus inside the cylinder: $t_{fixed} = \max(0,\ \min(t,\ 1))$.

Therefore, the closest point on the cylinder axis will be defined as in Equation 3.

$$P(x, y, z) = \boldsymbol{A} + t_{fixed} \cdot \boldsymbol{d} \tag{3}$$

In the following step, the distance function $dist(x, y, z)$ is computed as in Equation 4:

$$dist^2(x, y, z) = (x - P_x)^2 + (y - P_y)^2 + (z - P_z)^2 \tag{4}$$

Thus, the implicit function for the lateral surface of each cylinder can be defined as in Equation 5:

$$dist^2(x, y, z) - r^2 = 0 \tag{5}$$

Moving to the end caps, placed at point A and B, their implicit functions can be defined as in Equation 6:

$$\boldsymbol{d} \cdot (\boldsymbol{r} - \boldsymbol{A}) = 0 \,\text{and}\, \boldsymbol{d} \cdot (\boldsymbol{r} - \boldsymbol{B}) = 0\ \text{where}\ \boldsymbol{r} = (x, y, z) \tag{6}$$

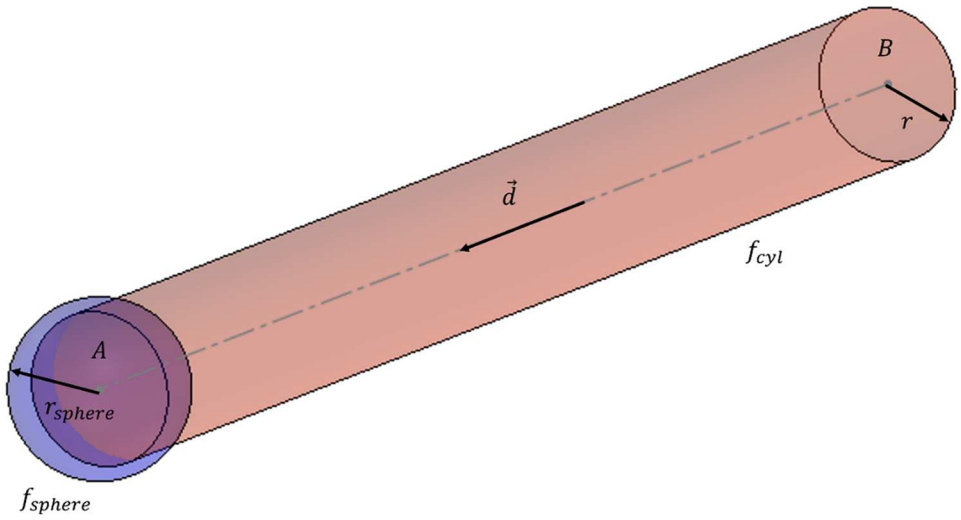

Figure 3. Visual representation of a cylindrical strut (in red) with radius r that connects the points ***A*** and ***B***; an additional sphere (in blue) with radius r_{sphere} is placed at each vertex.

As the last step, by combining the lateral and the end caps, the function representation for the k-th cylindrical strut f_{cyl} can be described as in Equation 7:

$$f_{cyl_k}(x,y,z) = \max(dist^2 - r^2, \max(-\boldsymbol{d}\cdot(\boldsymbol{r}-\boldsymbol{A}), -\boldsymbol{d}\cdot(\boldsymbol{r}-\boldsymbol{B}))) = 0 \tag{7}$$

Moreover, for each unit cell vertex ***J*** contained in matrix ***N***, the code computes the implicit function of a sphere with radius r_{sphere}. The mathematical definition of the sphere radius is visible in Equation 8 as a function of the slenderness ratio ($slend = r/L$) of the strut.

$$r_{sphere} = r\cdot(1+slend) \tag{8}$$

The j-th sphere is centred on the vertex $\boldsymbol{J}(x_J, y_J, z_J)$ with radius r_{sphere} and its implicit function is described in Equation 9.

$$f_{sphere_j}(x,y,z) = (x-x_J)^2 + (y-y_J)^2 + (z-z_J)^2 - r_{sphere}^2 = 0 \tag{9}$$

The framework adds spheres to the joints to avoid non-manifold entities and provide safe structures to avoid stress concentration, which has not been addressed in [55]. Moreover, the sphere radius should be slightly larger than the cylindrical entities to create a smooth joint and provide safe Boolean operations [40].

Lastly, the overall implicit representation of the unit cell, initialised as a large value, is updated iteratively by taking the pointwise minimum. The overall FRep of the unit cell can be obtained by iterating to the k-th cylinder and the j-th sphere and combining them with the *min* operator, as in Equation 10:

$$f_{cell}(x, y, z) = \min(f_{sphere_j}(x, y, z), f_{cyl_k}(x, y, z)) \tag{10}$$

Through the isosurface MATLAB function, it is possible to define the locus of the points lying on the external surface of the unit cell. The external surfaces are detected by imposing a zero value on the implicit function $f_{cell}(x, y, z) = 0$. The MATLAB function returns a structure where vertices and triangular facets lie on the FRep, which can be directly used to write an STL file.

The isosurface function requires the definition of a grid because this command operates on volumetric scalar data – precisely, a 3D scalar field where each point in space has an associated value. The aforementioned grid is defined along the three main directions in the interval $[-r_{sphere}, L + r_{sphere}]$ to capture the entire unit cell.

It was decided to develop FRep of the single cell rather than the entire structure to maintain the precision of the isosurface command definition. Choosing an appropriate resolution for the lattice structure would have significantly increased computational time, especially for large design domains. On the other hand, modelling the single cell allows for selecting an optimal resolution applied to a much narrower spatial range, thereby speeding up the modelling operations. The mesh resolution can be expressed as proportional to the slenderness ratio of the unit cell: $res = \varepsilon \cdot slend$.

The grid resolution plays a crucial role in determining the level of detail and accuracy for the geometric representation of the lattice structure. Specifically, it affects the density of the beam subdivision, directly impacting how finely the beams and spheres

are represented in the 3D space. Thus, a sensitivity study has been conducted to select the optimal value of ε for an accurate geometry description without penalising the computational effort to create the mesh.

The sensitivity study on the grid resolution considered a lattice with $2 \times 2 \times 2$ BCC unit cells, with $L = 10$ and $r = 1$ to account for the joints in the evaluation, too. The benchmark geometry was generated in SolidWorks 2025 Premium [68], a CAD software that employs a BRep, and the STL file was subsequently exported at the highest possible resolution. The comparison was performed with the same $2 \times 2 \times 2$ lattice modelled using the FRep described above at progressively higher resolutions and then converted to STL. The two 3D models were then imported into Cloud Compare (Version 2.13.2) [69], where the average distance between the two meshes, the Root Mean Square (RMS) value, and the standard deviation were calculated. Figure 4 collects the results of the aforementioned sensitivity study.

According to Figure 4, a mesh resolution of 30 points ($\varepsilon = 3$) in each *x*, *y*, and *z* direction is a reasonable choice because the RMS and standard deviation values stabilise and remain constant as the mesh resolution increases while the computational time is acceptable. Thus, a proportional factor $\varepsilon = 3$ has been used in this research and can be set as a typical value to be refined with further iterations.

2.4. Unit cell replication and triangulation check

The next step of VOLFREP performs the replication of a single unit cell across a structured lattice domain by translating and assembling multiple instances of the base unit cell. First, the translation offset variable is defined to specify the spacing between adjacent unit cells in the three spatial directions. Each cell is spaced with a distance equal to the strut length L in the x, y, and z directions when the framework handles uniform lattice structures.

Then, a triple nested loop iterates over the voxel grid dimensions, determined by the aforementioned voxelization process of the design domain. Before placing a unit cell at a given position (i, j, k), the code checks the corresponding value in the 3D logical matrix coming

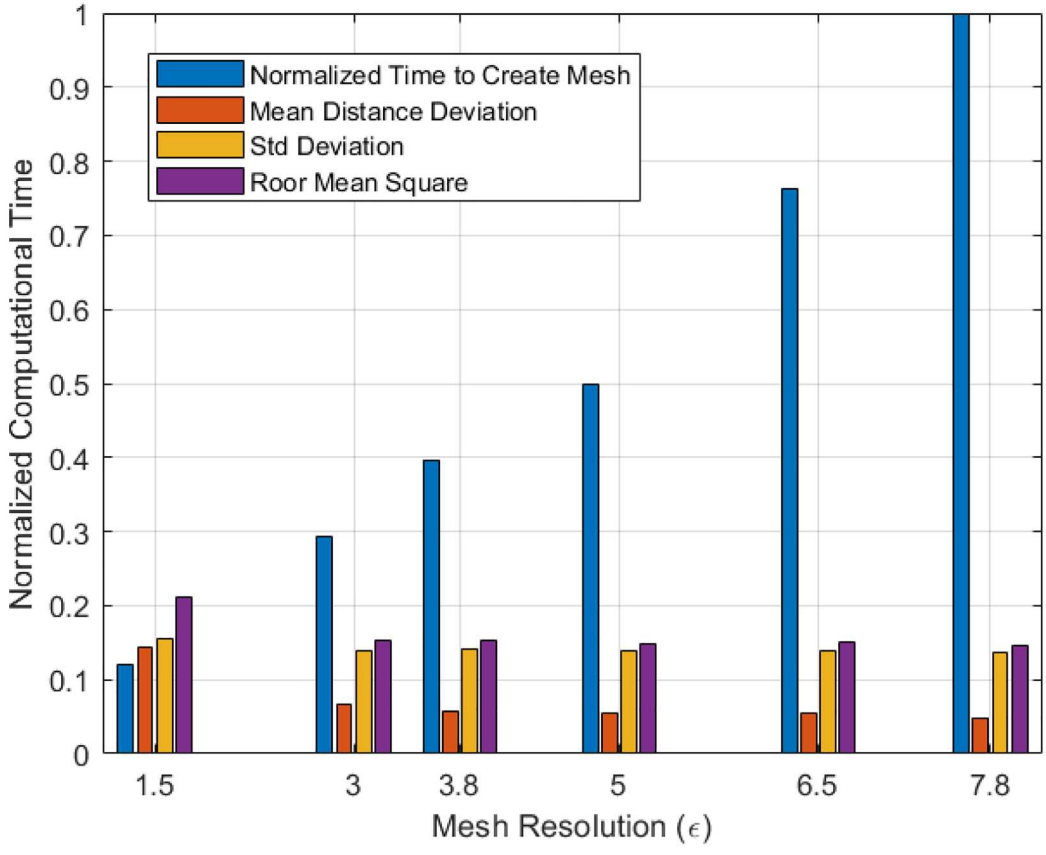

Figure 4. Sensitivity study of the mesh resolution for the FRep sampling; the mesh resolution is plotted against the normalised computational time required to evaluate the isosurface of the $2 \times 2 \times 2$ BCC lattice structure.

from the voxel model. The unit cell is replicated if the voxel at that location is active.

A translation offset is computed for each active voxel to position the unit cell within the lattice. The original vertices of the unit cell are translated by this offset and appended to the list of vertices, which stores the complete lattice structure.

Since each replicated unit cell introduces new vertices, the connectivity matrix of the faces is updated accordingly. A specific variable tracks the cumulative number of vertices added, ensuring correct indexing for each new set of faces. The adjusted faces are then appended to the complete facet list.

This approach enables the efficient construction of a periodic lattice structure by systematically replicating the base unit cell while maintaining correct geometric and topological relationships between the elements. Furthermore, unit cell replication is avoided in the correspondence of deactivated voxels inside the design domain bounding box but outside the design domain itself. On the one hand, replication only occurs when an active voxel is present, indicating that the region under consideration is a part of the design domain. To mention an example, taking the digital model of a 3D finite wing with a 1-meter span, 1-meter chord, and a NACA 2412 airfoil, Figure 5 represent a wing cross-section. Using a BCC unit cell with $L = 20$ and $r = 2$, a 51 x 7 x 50 3D logical matrix is computed.

By filtering just the active voxels, which for the considered example covers 72.3% of the bounding box volume, the design of the overall lattice structure has a halved computational cost.

The reported simple example demonstrates that the voxelization of the design domain is an additional but fundamental step implemented in the VOLFREP framework, making the lattice structure modelling more efficient.

Once the unit cells are replicated, the overall triangulation is checked to avoid non-manifold edges, vertices, facet repetitions, and self-intersection through the use of the GIBBON open-source toolbox for MATLAB [70].

2.5. Graded lattice modelling

The VOLFREP framework is also capable of handling graded lattice infills. In particular, after the cell replication described in Section 2.4, a grading function scales the vertices' positions along a direction $\boldsymbol{g} = \boldsymbol{v}/\boldsymbol{v}$ defined by the user. The overall grading implementation is visible as pseudocode in Appendix A.4. Then, for each vertex $\boldsymbol{p_i}$, a specific MATLAB function evaluates how far the point lies along the grading direction using Equation 11.

$$s_i = \boldsymbol{p_i} \cdot \boldsymbol{g} \tag{11}$$

Let s_{min} and s_{max} be the minimum and maximum values of s_i over all vertices. The normalised position of each vertex along the $\boldsymbol{g}$ direction is $\widehat{s_i}$ (Equation 12).

$$\widehat{s_i} = \frac{s_i - s_{min}}{s_{max} - s_{min}} \tag{12}$$

A scale factor $\lambda_i = f(\hat{s})$ is assigned to each vertex based on its normalised position. Different kinds of grading functions have been included to test the compatibility with VOLFREP:

- scalar grading: $f(\widehat{s_i}) = const$;
- linear grading: $f(\widehat{s_i}) = \lambda_{min} + \widehat{s_i}(\lambda_{max} - \lambda_{min})$;
- polynomial grading: $f(\widehat{s_i}) = \lambda_{min} + (\lambda_{max} - \lambda_{min})\widehat{s_i}^2$;
- exponential grading: $f(\widehat{s_i}) = \lambda_{min} \cdot e^{\ln\left(\frac{\lambda_{max}}{\lambda_{min}}\right)\widehat{s_i}}$.

However, the aforementioned list is not exhaustive; the user can implement whatever geometrical grading with simple modifications on the MATLAB code.

Thus, the new vertices positions of the graded lattice can be assessed using Equation 13.

$$\boldsymbol{p_i^{graded}} = \boldsymbol{p_i} + (\lambda_i - 1)(\boldsymbol{g} \cdot s_i) \tag{13}$$

A visual representation of graded lattices, whose governing equations were proposed in the previous bullet list, is visible in Figure 6.

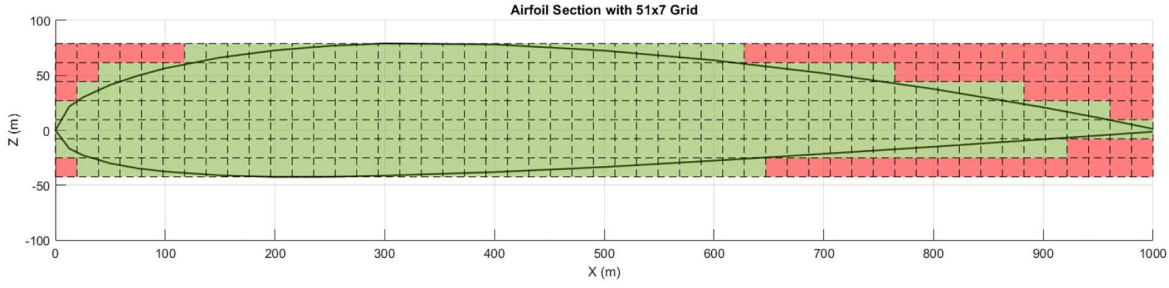

Figure 5. Graphical representation of active (green) and deactivated (red) voxels; only the active hexahedral cells will be considered for the unit cell replication to construct the lattice structure.

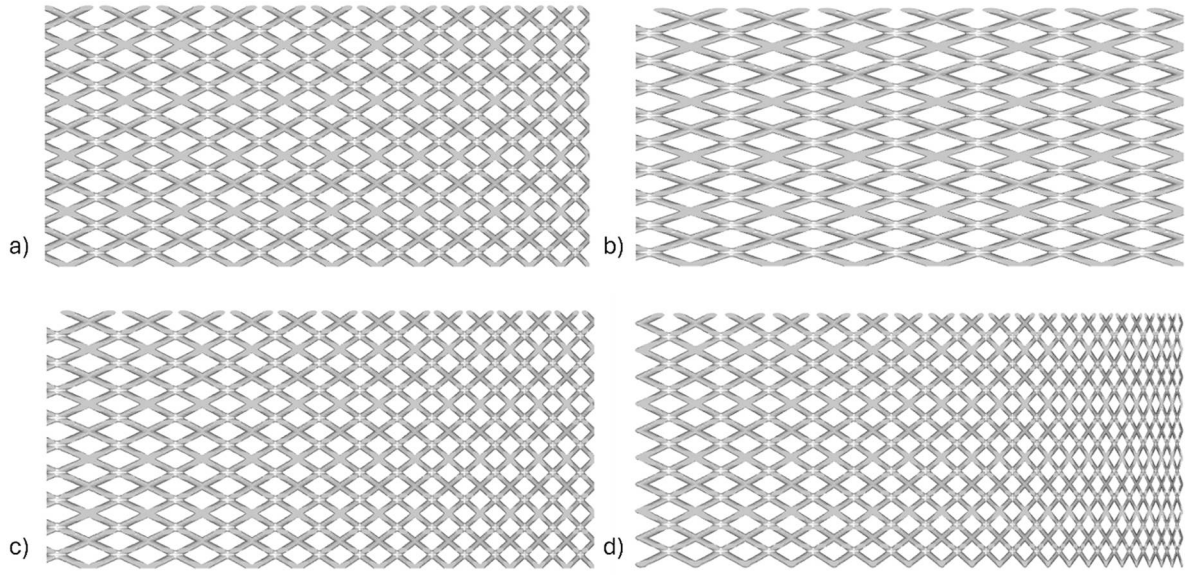

Figure 6. Graphical comparison of a parallelepiped filled with CVC unit cell using a grading function along the x direction: (a) linear grading, (b) scalar grading, (c) polynomial grading and (d) exponential grading.

2.6. Boolean intersection with the triangulated design domain

In the final step of the proposed framework, visible in the flowchart of Figure 1, the watertight lattice structure generated via the VOLFREP framework is exported as a binary STL. Compared to Tang et al. [55], where the voxel-based model is generated to create 2D sliced images for DLP, STL files ensure high compatibility with all AM technologies.

Certain struts of the filling lattice will cross the borders of the component the user wishes to fill with cellular structures through the voxel-based unit cell replication.

Thus, a Boolean intersect operation is necessary to trim the final lattice to match the final component's shape. The MATLAB code of VOLFREP has been linked with a Python environment to use the *trimesh* module (Version 4.6.9) [71], which is widely used by companies such as NVIDIA [72] and Cura Ultimaker [73]. This module efficiently implements mesh Boolean operations and is necessary to compute the last step of the proposed framework.

The Boolean operation may be obtained by a mathematical operation if the FRep of the design domain is available. However, the design domain of real-wise industrial components often shows very complex geometries, causing difficulties in converting into a FRep rather than simple shapes that analytical functions can describe.

Thus, the authors decided to complete the domain and lattice intersection task with an open-source toolbox that manages triangular Boolean mesh operations well.

The *trimesh* Python module supports robust mesh operations on triangular surface representations. Specifically, the binary STL file representing the design domain (e.g. a solid volume) intersects with the binary STL file of the VOLFREP-derived lattice structure. The *trimesh.boolean.intersection()* function is used to compute the geometric intersection between the two meshes, effectively trimming the lattice structure to fit precisely within the design domain. This operation returns a new watertight mesh that retains only the portions of the lattice structure enclosed by the design volume.

The reader can refer to [71] for additional implementation details and documentation for further insight into algorithmic handling, robustness, and numerical precision of Boolean operations.

The Boolean operation implementation and the main MATLAB code are visible in Appendix A.5 as pseudocode.

Figure 7 shows an example of a simple 3D model filled with a uniform CVC lattice. The final geometry is obtained through a mesh intersection between a shell model of the component with a uniform thickness and the CVC lattice structure.

2.7. VOLFREP computational analysis

The VOLFREP framework has been assessed, considering usability and computational efficiency, and compared to McMillan MATLAB code [50], the BRep modelling

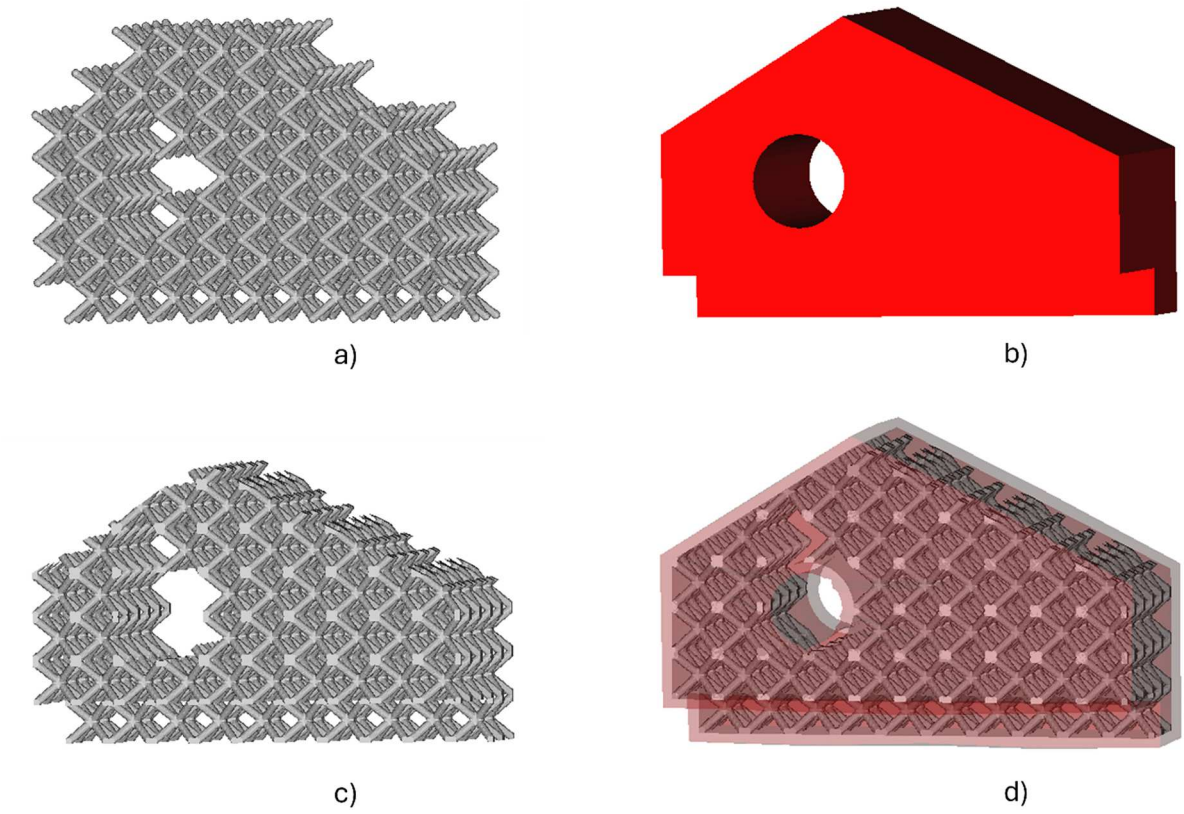

Figure 7. (a) The voxel-based lattice structure model intersects the digital model of the design domain (b); in (c) the trimmed lattice structure and in (d) the final model with the shell and the filling lattice structure.

through Solidworks 2025 Premium, a commercial CAD software package, a Constructive Solid Geometry (CSG) tool called LSWM [74], and the Blender addon *Volume Lattice* by SineWave based on parametric design [75]. The Solidworks modelling approach followed the Bastos report [76] that exploits the welding workbench to quickly and easily model the struts of the lattices and the STL binary file saved using 'fine' mesh settings.

The BRep approach has no unit cell library limitation and can be adapted to any unit cell. On the one hand, the Volume Lattice addon has a wide strut-and-node and TMPS unit cell portfolio; on the other hand, the McMillan codes have a restricted cell portfolio. Thus, for the scope of this comparison, only the Box, CVC (called BCC in McMillan tool) and FCC unit cells are considered. Lastly, the LSWM tool has just the Box unit cell implemented, but it has been taken under consideration in this comparison because it can be seen as an automatic tool that efficiently performs the Boolean operations typical of the CSG tools.

A lattice structure with $10 \times 10 \times 10$ unit cell repetitions in the three principal directions is considered. The unit cell is made up of cylindrical struts with 10 mm in length and 1 mm in radius. The simulations were run on a workstation with 96 GB of RAM and an Intel i9-14900F at 2.00 GHz CPU. The results are collected in Table 1. In the McMillan tool, a 'Cross-section order' of 30 is chosen to obtain a section as close as possible to a circle. In the Volume Lattice approach, mid-quality settings are used to obtain the STL binary file of the computed lattices. The Volume Lattice computational time is obtained by summing the preview and the surface meshing timings.

It is necessary to report that the FCC unit cell modelling has failed using the McMillan codes for unknown reasons. From the collected data, it is possible to say

Table 1. A comparison of three different approaches used to create a $10 \times 10 \times 10$ uniform lattice structure with three different unit cells; the data include the computational time required to obtain a binary STL file of the lattice structure and the file dimension.

	Box		FCC$^{\Delta}$		CVC*	
Modelling approach	Time [s]	File [MB]	Time [s]	File [MB]	Time [s]	File [MB]
McMillan [50]	56,33	28,63	Failed	Failed	204	43,44
VOLFREP	22,08	55,17	41,43	146,74	12,35	101,41
BRep	712	33,02	657	121,32	414	61,75
CSG [74]	52,81	141,237	N.A.	N.A.	N.A.	N.A.
Volume Lattice [75]	93,12	67,26	440,36	191,21	303,31	142,42

(*): in the McMillan tool, the CVC unit cell is called BCC; ($^{\Delta}$): in Volume Lattice, the FCC is not available, but 'Beam' is quite similar with the same n. of struts; N.A.: Not Available.

that the VOLFREP framework can efficiently model the lattice structures with a mean 97% time reduction compared to manual BRep modelling in commercial CAD software packages. Moreover, it has a convenient computational cost with respect to the McMillan approach for the simplest Box cell (60% reduction), but the computational fork increases by far when using struts in oblique directions, as in CVC. Here, the time saving reaches a value close to 90%. Then, if VOLFREP is compared with the CSG approach, it can be seen from Table 1 that the former is 67% faster than the CSG method, with a resulting STL file 60% lighter in terms of occupied memory. Due to the low computational performance of the CSG tool and minimal unit cell library, no other test will be involved in the following.

Lastly, the comparison with the Volume Lattice addon for Blender reveals that it is outperformed by VOLFREP for all the tested cells (87% mean reduction in computational time and 23% mean file size reduction). However, it represents a valid alternative for the lattice design if a user prefers a Graphical User Interface (GUI), an extensive unit cell library, and the possibility of having a rapid preview of the lattice structure before its actual modelling. On the other hand, Volume Lattice has lower computational efficiency and is less intuitive in the unit cell dimension settings.

The resulting CVC $10 \times 10 \times 10$ lattice, saved as a binary STL file, is imported in Meshlab v2023.12 and analysed in terms of mesh quality, an important characteristic not only for Finite Element Analyses (FEA) but also for AM. The mesh quality comparison is reported in Table 2, and the metrics used are [77]:

(1) Area vs Maximum side ratio: the average ratio between the area of each triangle and the length of its longest edge. The ratio indicates how 'fat' or 'thin' the triangles are. A higher ratio may indicate more evenly shaped triangles, while a smaller ratio indicates longer, thinner triangles;
(2) Area standard deviation: the size of each triangle; a higher standard deviation suggests that the mesh has triangles of significantly different sizes, indicating non-uniform resolution. Thus, for a consistent mesh, the area standard deviation should be low;
(3) Inradius vs circumradius ratio: the relationship between the inradius (the radius of the largest circle that can fit inside the triangle) and the circumradius (the radius of the circle passing through all three vertices of the triangle). This ratio is used to evaluate the shape quality of the triangles:
    - A value close to one indicates that the triangle is nearly equilateral, ideal for numerical simulations and additive manufacturing;
    - A value significantly lower than one suggests that the triangle is more elongated or skewed.

Table 2. A comparison of the mesh quality of the three approaches used to create a $10 \times 10 \times 10$ uniform CVC lattice structure; McMillan's mesh has been repaired in Meshlab to close all the holes before evaluating the mesh quality.

Mesh quality metric (μ: average value; σ standard deviation)	VOLFREP	BRep	McMillan	Volume Lattice
Area/Max Side μ	0,513	0,114	0,037	0,350
Area/Max Side σ	0,127	0,187	0,031	0,159
Area σ	0,026	0,174	0,122	0,159
Inradius/Cirmuradius μ	0,800	0,188	0,060	0,563
Inradius/Cirmuradius σ	0,141	0,304	0,049	0,231

Additional analysis can be conducted by modelling a parallelepiped lattice structure with an increasing number of cell replications for the complete unit cell library implemented in VOLFREP. The results are collected in Figure 8. As expected, the computational cost for replicating the entire unit cell library increases exponentially when the number of unit cells grows. Thus, the prearranged voxelization step to deactivate some voxels is fundamental to lower the overall computations of VOLFREP.

Once the VOLFREP framework has been widely described, the following section shows its application in an industrial case study.

3. Results

3.1. Uniform lattice modelling: GE bracket case study

The VOLFREP framework is validated through a case study that uses the 3D digital model of the GE bracket [78], a well-known benchmark in the aerospace industry.

The topology, converted into a binary STL file, has a bounding box of $178{,}54 \times 62{,}50 \times 108{,}31$ mm and is visible in Figure 9.

The original geometry is converted into a shell body with a constant thickness of 3 mm with Solidworks software. The VOLFREP framework is used to fill the inner volume with a uniform lattice structure, being available the binary STL file of the external shell.

The case study involves the creation of a uniform strut-and-node unit cell, where each cylindrical strut has a length of 10 mm and a radius of 1 mm.

The design domain voxelization results in an $18 \times 7 \times 11$ logical matrix where just 44,6% of the voxels are active. The final lattice structure counts 618 unit cells

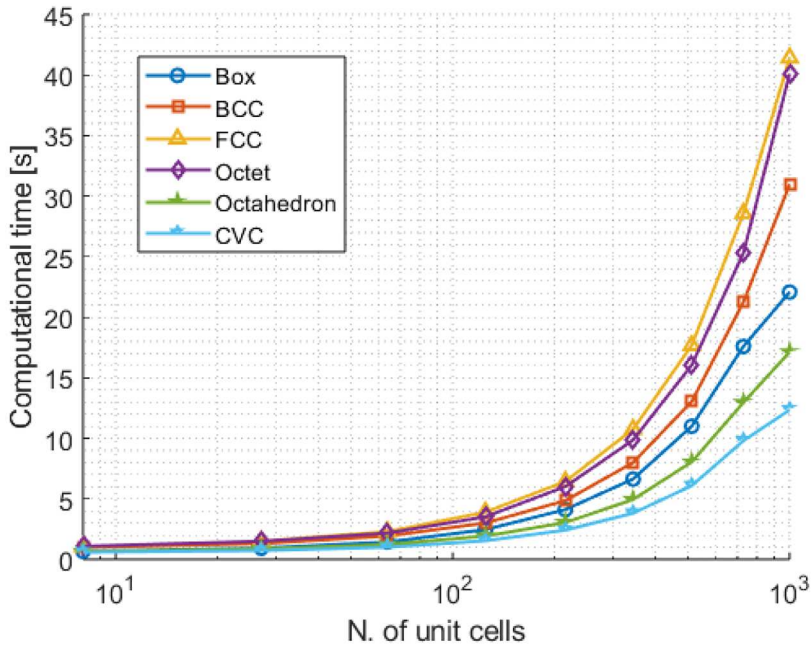

Figure 8. Computational time required to model and export an STL binary file representing a parallelepiped lattice structure; the analysis is performed for each unit cell at the disposal of VOLFREP.

before the trim phase occurs to fit the corresponding inner volume through the Boolean intersection perfectly.

The GE bracket has been filled with all six available unit cell topologies to demonstrate the overall capabilities of the VOLFREP framework. The cylindrical struts have a length of 10 mm and a radius of 1 mm. As mentioned in Section 2, a resolution of 30 points along the three principal directions is used for the isosurface evaluation. In the following, the unit cell is replicated as described in Section 2.4, and the close vertices are merged. The GIBBON toolbox checks the final triangulated mesh to avoid non-manifolds and self-intersections. Figure 10 shows the uniform lattice structure overlapped with the bracket design domain for the CVC case.

Once the lattice binary STL file is calculated, the Boolean intersection with the design domain model is processed to trim the resulting lattice structure. As a last step, it is possible to compute a unique binary STL file that merges the shell digital model and the trimmed lattice structure through a union Boolean operation.

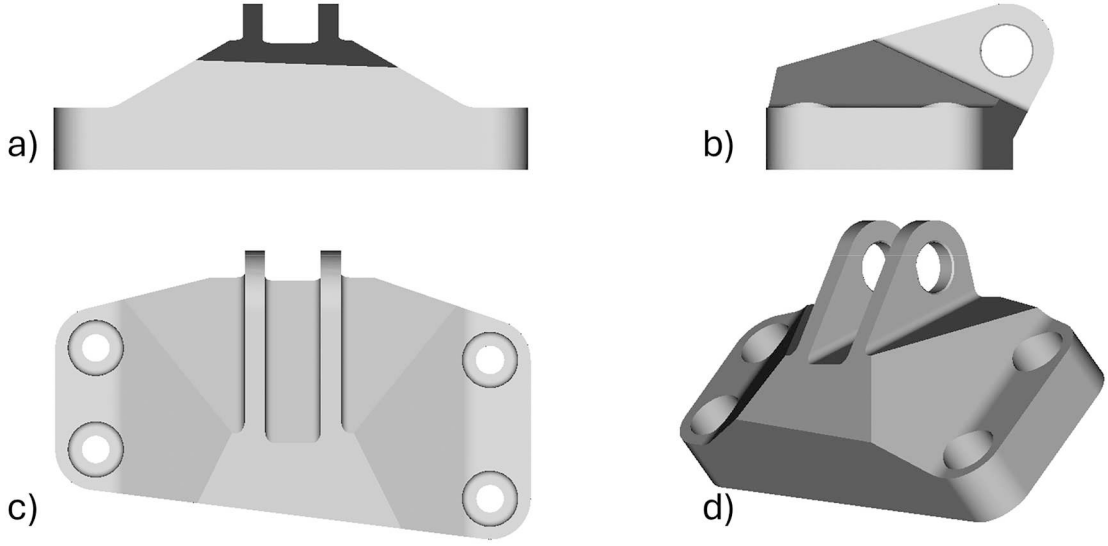

Figure 9. GE bracket case study: (a) frontal, (b) lateral, (c) top and perspective view.

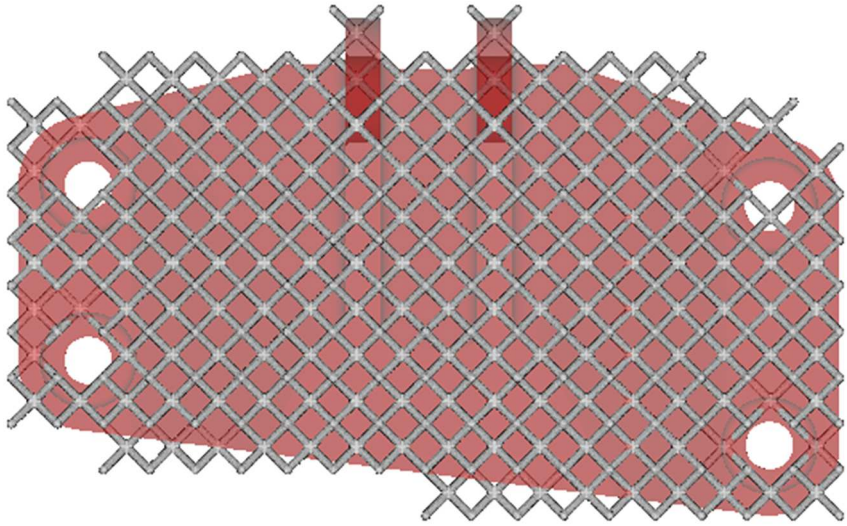

Figure 10. Top view of the GE bracket case study overlapped with the uniform lattice structure of CVC unit cells.

On the one hand, Table 3 collects the computational time required to obtain the final triangular mesh file of the external shell and the trimmed infill lattice structure for the entire unit cell library of VOLFREP. On the other hand, Figure 11 shows a breakdown of computational costs for the specific case study for each operation in the VOLFREP framework.

Figure 11(a) shows the computational cost spread to all the main steps of the VOLFREP framework, while Figure 11(b) shows how the costs would change if the voxelization step is not applied. Indeed, the cell replication would be highly demanding in covering the entire design domain bounding box compared to a smaller subdomain of active voxels.

However, when the number of unit cells in the modelled lattice increases, the cell replication execution time becomes paramount. On the other hand, STL2Voxel and Boolean intersection operations have a minor impact on the total computational time. Indeed, Figure 12(a) shows the computational time required for the three more demanding operations implemented in the framework using the GE bracket as a benchmark model, filled with CVC unit cells with decreasing dimensions, thus increasing number of unit cells.

Once the more demanding operation has been detected for an increasing number of lattice unit cells, an additional computational study focused on finding the maximum viable model size. In particular, a cubic domain filled with a CVC lattice structure with an increasing number of cell replications along the three axes has been selected. The computational time required for the cell replication has been recorded for each test, as visible in Figure 12(b). The trend follows a cubic polynomial function with $R^2 = 0.999$. It is possible to access the maximum number of cells the VOLFREP can handle, assuming that, as a temporal threshold in industrial applications, cell replication has a maximum computational time of 1 h. By extrapolating the data in Figure 12(b), 17,576 unit cells can be safely modelled in 1 h. Assuming a unit cell length of 10 mm, the VOLFREP framework could cover a design domain of 17,576,000 $mm^{3,}$ which can be considered a satisfactory result.

At the end of the process implemented in VOLFREP, the obtained STL file can be imported into any pre-processing software, such as the slicers for the FFF technology. The 3D model is converted to obtain the machine-readable file (i.e. Gcode [45]) that contains the manufacturing settings chosen by the designer. As a proof of concept, the GE bracket filled with uniform CVC lattice is manufactured with FFF technology using PLA material. Figure 13 shows the prototype with a shell cutaway to observe the inner lattice infill modelled through the VOLFREP framework.

3.2. Fabrication, mechanical simulation and geometric accuracy

The proposed VOLFREP approach is used to model a uniform $3 \times 2 \times 1$ lattice structure using the CVC unit cell ($L = 20$, $r = 2$). Watertight models produced by the proposed geometric modelling method can be

Table 3. Computational time required to obtain the final STL binary file for the proposed case study.

Unit cell topology	Box	BCC	FCC	Octet	Octahedron	CVC
Time [s]	231	335	363	422	277	248
File [MB]	44,16	104,71	117,50	150,57	97,02	78,52

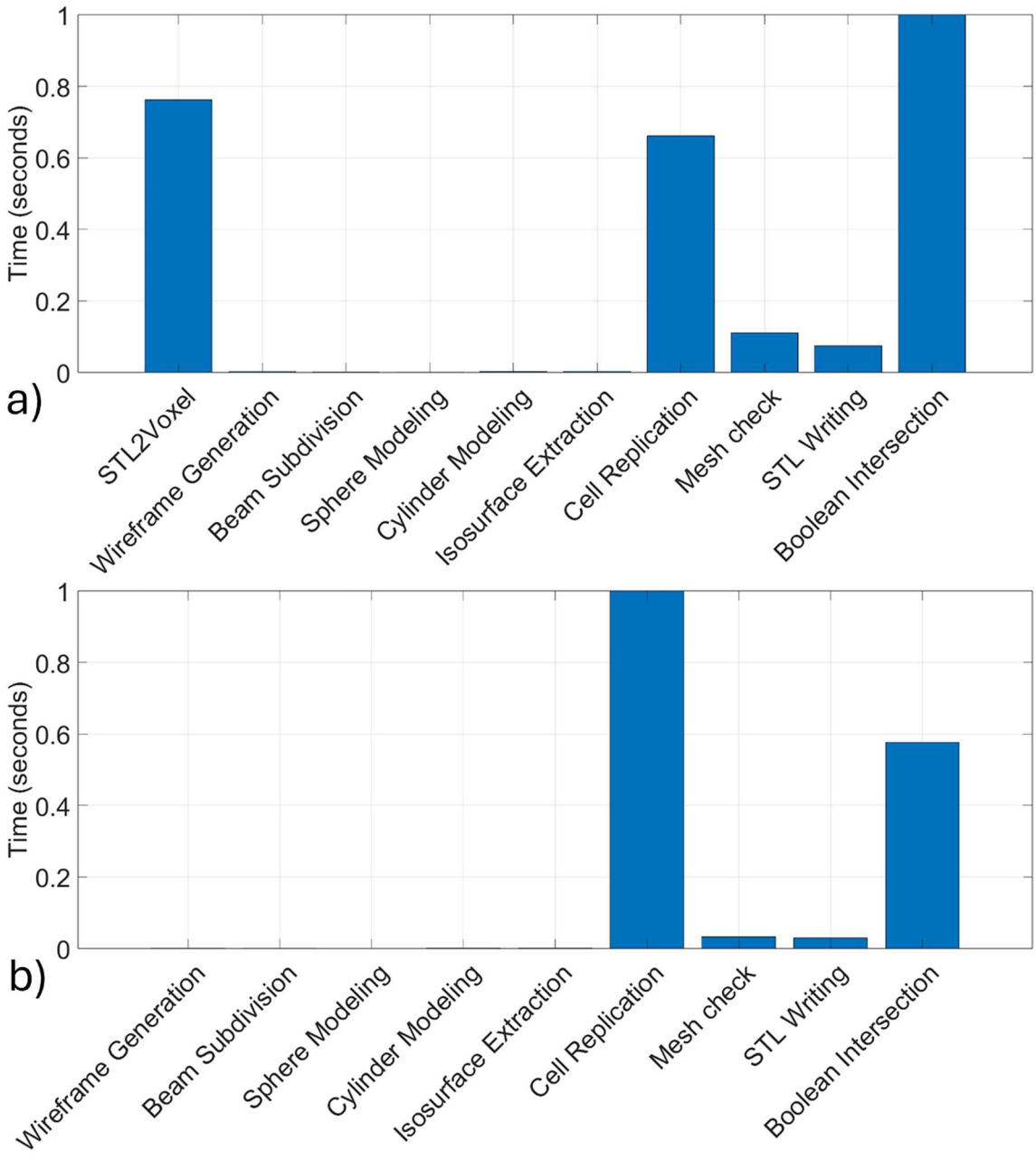

Figure 11. Computation of the relative cost breakdown for the GE bracket case study; each execution time is normalised by the maximum execution time. (a) The complete VOLFREP framework is analysed; (b) the computational cost changes if the voxelization process is deleted and the entire design domain is used for cell replication.

easily transformed into STEP files that are helpful for Finite Element Analysis.

A numerical tensile test with a fixed external load has been performed on the sample depicted in Figure 14 to verify the structural efficiency of a PETG-fabricated lattice sample.

The results demonstrate a limited displacement magnitude under a 200N tensile load, proving that VOLFREP can model structurally sound components. In the future, tensile tests on manufactured samples will be performed to validate the accuracy of the FEA model for the fabricated lattice sample.

The same sample has been manufactured and 3D scanned using a Creality Cr-Scan 01 [79]. This scanner uses structured light, and its level of accuracy has been evaluated in previous studies [80]. To assess the geometric accuracy of the manufactured PETG component, the sample has been scanned and compared to the 3D digital model coming from the VOLFREP framework using CloudCompare. The superimposition of both STL meshes is visible in Figure 15(a), while Figure 15(b) shows a histogram that represents a statistical distribution of the distance errors between corresponding points of both geometries.

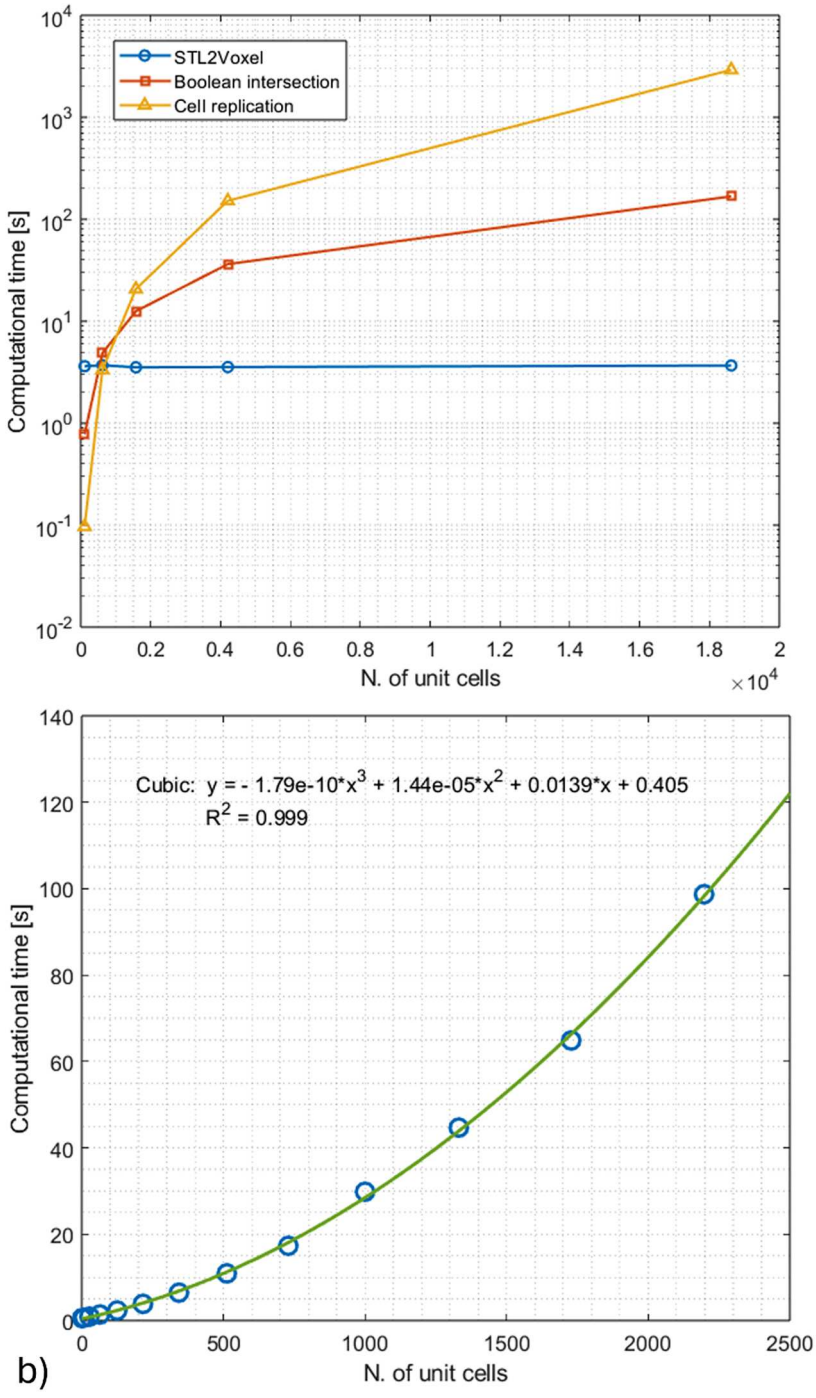

Figure 12. (a) Computational time analysis of STL2Voxel, cell replication and Boolean intersection operations as the number of CVC cells increases; (b) cell replication time saturation study as the number of CVC cells increases.

The results demonstrate that the manufactured component is slightly upscaled compared to the 3D digital STL of the VOLFREP approach, as visible by the statistical peak of Figure 15(b) shifted towards the right (positive distances, points lying above/outside the reference surface).

However, this discrepancy is limited and around the accuracy of the 3D scanner, as previous studies on the Reverse Engineering (RE) equipment have demonstrated. Thus, it is possible to state that the VOLFREP framework can model 3D lattice structures with high manufacturing fidelity.

3.3. Graded lattice modelling: piston rod case study

The capability of modelling graded lattices is shown through a second case study that uses the 3D digital model of a piston rod.

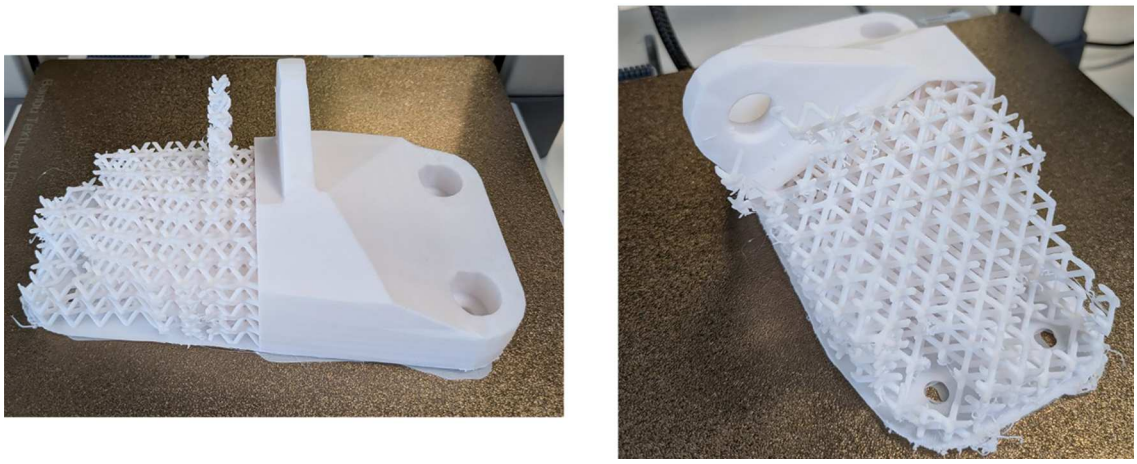

Figure 13. GE bracket manufactured with the FFF technology in PLA material; a shell cutaway brings to light the lattice structure infill using the CVC unit cell.

The topology, converted into a binary STL file, has a bounding box of $121 \times 28 \times 11.6$ mm and is visible in Figure 16.

The original geometry is converted into a shell body with a constant thickness of 1 mm with Solidworks software. The VOLFREP framework is used to fill the inner volume with a graded lattice structure, being available the binary STL file of the external shell.

The case study involves the creation of a linearly graded strut-and-node unit cell, where the base unit cell has a cylindrical strut with a length of 2 mm and a radius of 0.2 mm. In this specific case study, the following options are selected for the grading function:

- $\lambda_{min} = 1$;
- $\lambda_{max} = 2$;
- Grading direction: [1,0,0] (along the main axis of the piston rod).

Taking as reference the top view shown in Figure 15(a), the graded lattice described by the aforementioned parameters will result in a doubled strut length along the x direction on the right extreme of the rod.

The design domain voxelization results in a $26 \times 14 \times 6$ logical matrix where just 39.7% of the voxels are active. The final lattice structure counts 867 unit cells before the trim phase occurs to fit the corresponding inner volume through the Boolean intersection.

The piston rod topology has been filled with all six available unit cell topologies to demonstrate the overall capabilities of the VOLFREP framework. For example, Figure 17 shows the graded lattice structure overlapped with the bracket design domain for the CVC case.

Table 4 collects the computational time required to obtain the final triangular mesh file of the external shell and the trimmed infill lattice structure for the entire unit cell library of VOLFREP.

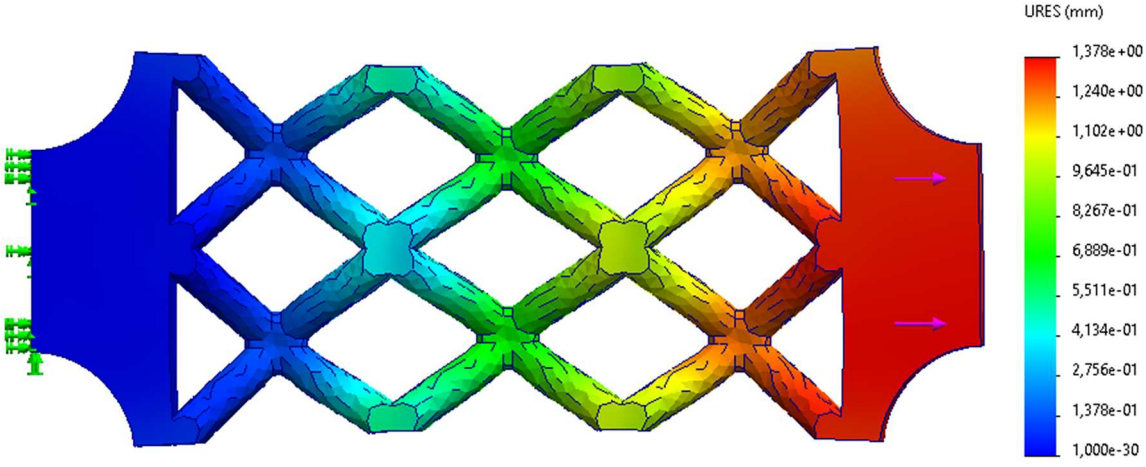

Figure 14. The simulation result (displacement magnitude) of the FEA model for tensile test samples under 200 N load along the x-axis.

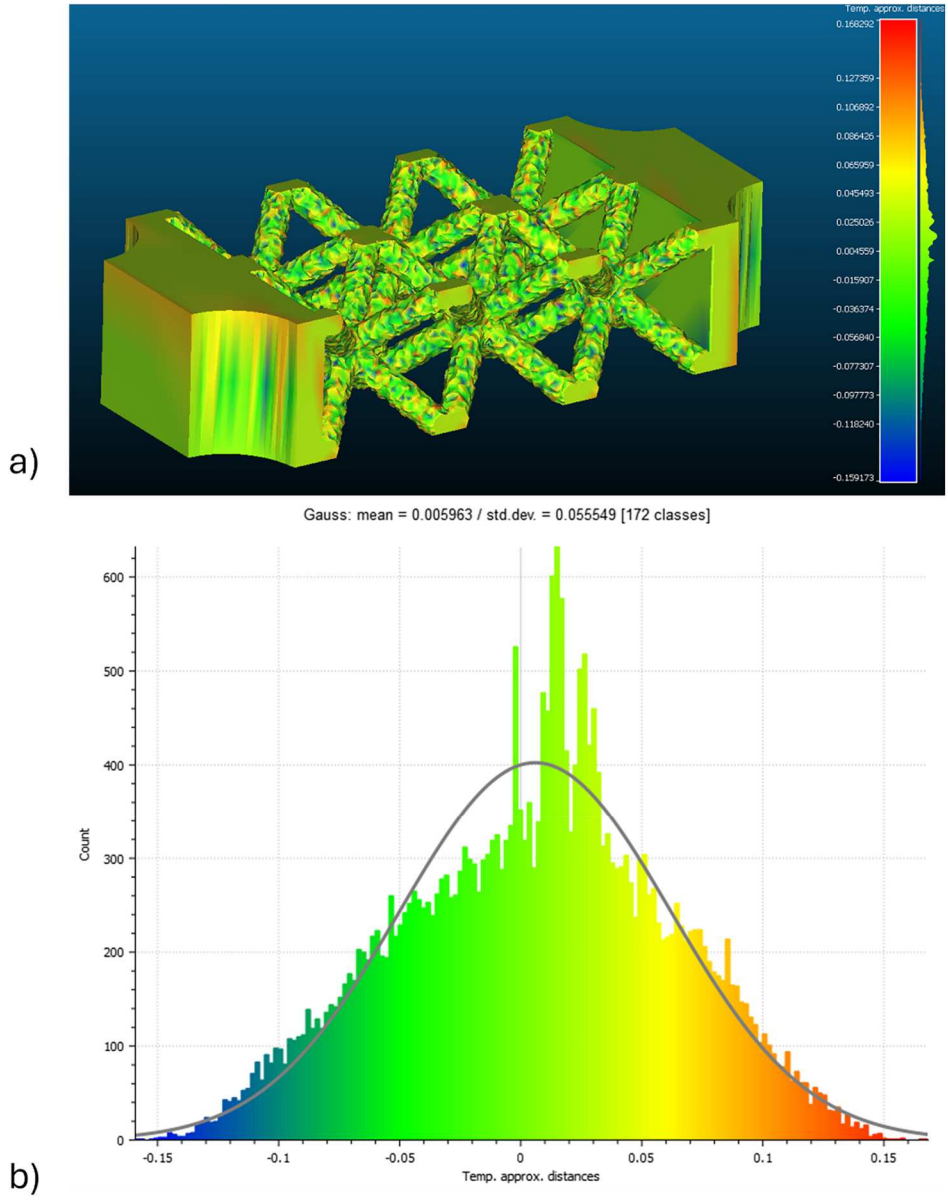

Figure 15. STL mesh comparison between the digital model obtained in VOLFREP and the manufactured part in PETG material; (a) a visual comparison in CloudCompare with a colour map showing the relative distances and (b) the statistical distribution of the distance errors between corresponding points of both geometries.

4. Discussion

The proposed VOLFREP approach demonstrates significant improvements in the design of lattice structures compared to the traditional BRep method. A reduction in computational time of nearly 90% is observed while maintaining high geometrical accuracy, as indicated by the low RSM error of 0.15 mm. This level of accuracy aligns with the precision of many AM technologies, ensuring the method's applicability in practical manufacturing workflows.

As demonstrated by its high inradius/circumradius ratio and low standard deviations for area and area/max side, VOLPREP produces the most consistent and uniform triangles with a reasonably high quality throughout the STL mesh (Table 2). It will likely produce the best 3D printing outcomes, with smoother surfaces and higher accuracy.

With a low inradius/circumradius ratio (indicating distorted triangles) and higher standard deviations in area and area/max side, the BRep approach exhibits the highest variability throughout the mesh. This result suggests that the mesh may contain irregular and poorly shaped triangles, potentially resulting in printing errors and poor print quality. After the post-processing correction to close holes, McMillan's tool still has some issues with distorted triangles, as evidenced by the

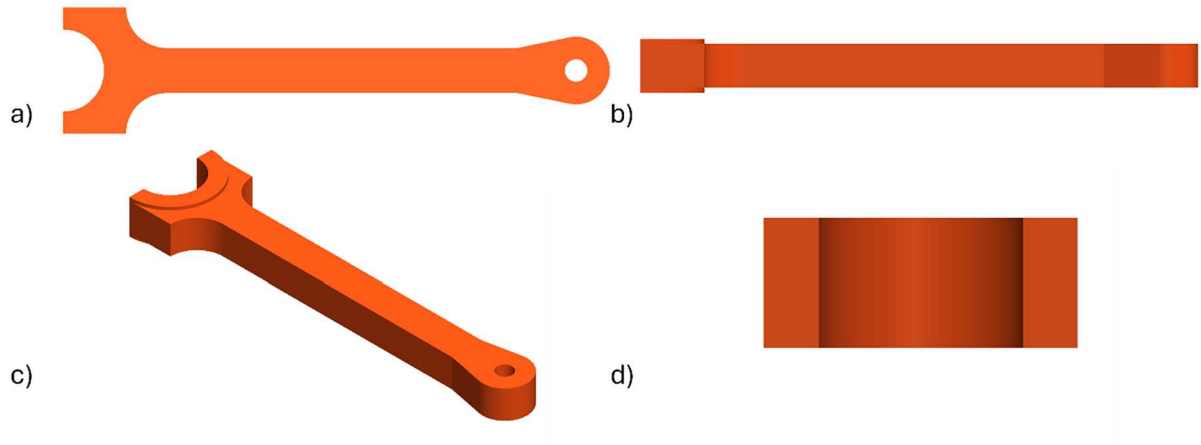

Figure 16. Piston rod case study: (a) top, (b) frontal, (c) perspective and lateral view.

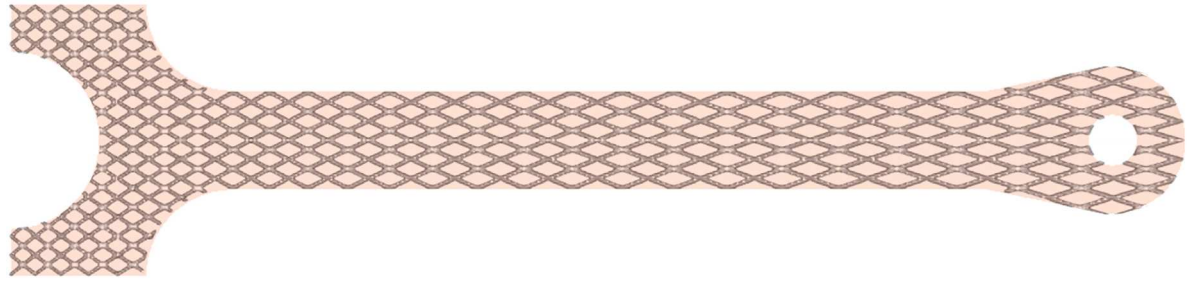

Figure 17. Top view of the piston rod case study overlapped with the linearly graded lattice structure of CVC unit cells after the boolean intersection phase.

poor inradius/circumradius ratios. However, the low standard deviation indicates that the distortion is consistent throughout the mesh. It may be an excellent balance of quality and consistency, though not as high-grade as VOLPREP.

Comparing Volume Lattice addon with VOLFREP, the results show that the triangles are moderately well-shaped, closer to equilateral than in BRep or McMillan, and more consistent through all the mesh, though still slightly elongated compared to VOLPREP. Regarding the triangle's area, Volume Lattice suggests a moderate variation in triangle size, worse than VOLFREP. Lastly, Volume Lattice shows a strong result for the inradius/circumradius metric, second only to VOLPREP. This result indicates that the mesh triangles are relatively regular and round, suitable for printability, and have moderate consistency in triangle regularity, such as VOLFREP.

A key advantage of VOLFREP is its voxelization-based approach, which has two fundamental advantages. The former is the link between the unit cells and the voxels, accelerating lattice design. The latter is the speed-up of the unit cell replication: only active voxels are considered, thus limiting the computational costs (see Figure 11(a)), similar to what is done in [55] with kernel points. To illustrate the method's effectiveness, an inner lattice structure with the CVC unit cell was constructed in 248 s (Table 3) using the voxel-based modelling in the GE bracket case study instead of 284 s without the voxelization procedure. Indeed, even though a further step in the VOLFREP is added, a significant decrease in time spent in the cell replication stage (3,04 s vs 15,40 s) justifies this design choice. This gain can be magnified in the case of more complex geometries.

Additionally, VOLFREP outperforms McMillan's approach by producing lattices that do not need post-processing in external software (i.e. hole filling) while supporting a broader range of unit cell topologies.

Table 4. Computational time required to obtain the final STL binary file for the piston rod case study.

Unit cell topology	Box	BCC	FCC	Octet	Octahedron	CVC
Time [s]	164	220	252	281	162	149
File [MB]	17,11	41,16	46,19	48,24	37,30	29,68

As the last comparison, VOLFREP shows high computational efficiency compared to the Volume Lattice addon for Blender, even if it lacks a GUI and the possibility of having a rapid 3D preview of the lattice before the actual 3D modelling. In particular, an 87% reduction in computational time for a set of unit cells and a 23% reduction in file size have been demonstrated. For example, considering the GE bracket case study as a valuable example in the industrial field, the Volume Lattice has computed the CVC uniform lattice with mid-geometric accuracy and $r = 1$, $L = 10$ in 27.6 min with an increased computational time of 91% compared to VOLFREP.

The FCC and Octet structures exhibited the highest computational demands among the tested unit cells. This result was expected due to their complex topology, characterised by more struts per unit cell. Despite this, the method remains effective across different lattice configurations. An increase in the computational cost that follows a polynomial cubic law as the number of lattice cells grows has been observed.

An operational cost breakdown of the VOLFREP framework highlights Boolean intersection, cell replication and voxelization as the most computationally expensive processes. While the importance of the voxelization process has been widely discussed earlier, the Boolean intersection of two meshes is required to trim the filling lattice structure. As mentioned, the FRep of the design domain is available only for trivial shapes. Conversely, industrial components are far from trivial geometries, thus limiting the applicability of this approach. For this reason, the more robust mesh Boolean intersection using the *trimesh* module has been adopted. However, optimising these steps could enhance the framework's performance, particularly for complex lattice structures. Additional simulations reveal that the cell replication phase becomes predominant in the case of a lattice structure with a large number of unit cells. In particular, a cubic polynomial behaviour fits the collected data well; by extrapolating them, it is possible to estimate the maximum viable model size, which can be modelled in one hour (a realistic time threshold in industrial applications).

The VOLFREP generated a small uniform sample suitable for FEA and fabrication. Numerical tensile testing confirmed its structural robustness under a significant tensile load. Moreover, a comparison between a 3D-scanned PETG-printed sample and the digital model showed minimal geometric deviation within the scanner's accuracy range. These results confirm that VOLFREP ensures both mechanical soundness and high manufacturing fidelity.

As a last result, Section 3.3 provides an industrial case study where grade lattices are implemented. The results confirm that the CVC, Octahedron and Box unit cells are the lightest from the computational point of view, while the Octet has the highest impact on the required computational resources.

The results confirm that the VOLFREP approach significantly accelerates lattice generation while maintaining manufacturability and accuracy. These improvements make it a viable alternative to traditional BRep techniques in designing industrial components. Moreover, the VOLFREP methodology can be easily integrated with existing CAD exploiting libraries already available in the literature.

4.1. VOLFREP pros and cons analysis

The above-discussed results can be summarised in a bullet list containing the main advantages of the VOLFREP approach:

- The hybrid Voxel-based FRep modelling is adaptable to any unit cell topology;
- High geometry fidelity is achieved when compared to the traditional BRep approach;
- Low computation cost compared to other approaches available in the literature to obtain a ready-to-be-manufactured component;
- High mesh quality of the resulting STL binary file yields the best results in 3D printing, ensuring smoother surfaces and better accuracy. Compared to voxel-based 2D images typical of DLP or LCD-based SLA technologies, the STL file guarantees high compatibility with all AM processes.
- The isosurface evaluation is at the unit cell level and not to the entire design domain to avoid using a very fine mesh to capture all the structure geometry. An additional sphere is modelled on the lattice structure joints to guarantee a safe Boolean operation and to avoid stress concentrations in the structure;
- The association of active voxels and the unit cell repetition is fruitful in avoiding losing computational power in the unit cell repetition outside the design domain;
- Graded lattice structures can be easily implemented through MATLAB handle functions that reveal high design flexibility;
- The mesh Boolean intersection between the design domain and the inner lattice is flexible to any domain shape;

However, the VOLFREP framework is limited to only six unit cell topologies for lattice structures. In the future, the possibility of orienting the lattice cell

pattern will be investigated with further research since, up to now, the cells have been replicated only along the three principal directions.

In future studies, the lattice structure modelled through the VOLFREP framework will be used for achieving the structural performance (i.e. stiffness, strength and modal analysis) using mechanical tests to validate that the VOLFREP models are not only computationally efficient but also structurally sound.

Nonetheless, this research provides a benchmark to the scientific community of the computational costs of designing ready-to-be-manufactured complex industrial components filled with strut-an-node lattice structures. This aspect was previously missing in the literature, where FRep had already been widely discussed without a detailed computational cost analysis.

5. Conclusions

This paper presents VOLFREP, a new hybrid voxel-based and Function Representation technique for modelling uniform and graded strut-and-node lattice systems. VOLFREP outperforms standard Boundary Representation approaches regarding computational efficiency, lowering modelling time by roughly 90% while maintaining good geometric accuracy (RMS error of 0.15 mm).

The framework's voxelization method is critical for improving lattice generation, minimising superfluous calculations, and facilitating fast unit cell replication. Indeed, this operation could be highly demanding for a massive number of lattice cells. The voxel-based implementation results in a 13% reduction in design time for the proposed GE bracket case study, compared to an approach without voxelization, demonstrating VOLFREP's benefits in industrial applications.

The study also examines the computational resource demands of the available unit cells (Box, BCC, FCC, Octet, Octahedron, and CVC), with FCC and Octet ones requiring the most processing time due to their complicated topology. Compared to previous approaches in the literature, VOLFREP provides fully manufacturable lattice structures without the need for further post-processing while offering a broad unit cell portfolio.

In contrast, BRep produces distorted triangles with high variability, compromising print quality and requiring extensive corrections. McMillan's method offers better consistency but still suffers from distortion and limited accuracy. The Volume Lattice addon performs moderately in mesh quality but incurs a 91% longer computation time and generates files 23% larger than VOLFREP. VOLFREP stands out by combining excellent STL mesh quality (low standard deviations and near-equilateral triangle shape), minimal computational cost, and high manufacturability.

This study fills a scientific gap in the literature by establishing a baseline for the computational cost of the design and conversion to STL binary format file for ready-to-manufactured lattice structures. Previously, FRep techniques were frequently discussed but lacked specific performance evaluations. The VOLFREP framework provides a scalable, adaptable, and computationally efficient method for creating lightweight, high-performance industrial components, notably for additive manufacturing.

A computational cost breakdown analysis shows that the Boolean intersection is one of the most computationally expensive phases. Future enhancements will improve the Boolean operations to increase efficiency, focusing on grading unit cells and lattice orientation control.

Credit authorship contribution statement

Antonio Bacciaglia: Writing – review & editing, Writing – original draft, Visualisation, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualisation. **Alfredo Liverani**: Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Alessandro Ceruti**: Writing – review & editing, Writing – original draft, Validation, Methodology, Conceptualisation, Supervision.

Author contributions

CRediT: **Antonio Bacciaglia:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; **Alfredo Liverani:** Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing; **Alessandro Ceruti:** Conceptualization, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing.

Disclosure statement

For the author Antonio Bacciaglia, University of Bologna has been funded through the MOST – Sustainable Mobility National Research Center, sponsored by the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023). This manuscript reflects only the authors' views and opinions; neither the European Union nor the European Commission can be considered responsible for them.

Data availability statement

The authors declare that the OCTAVE / MATLAB functions developed for this study are available as supplementary data.

Data can be downloaded from https://doi.org/10.6092/unibo/amsacta/8368. The methodology and other data necessary to reproduce the research are included in the manuscript.

ORCID

Antonio Bacciaglia http://orcid.org/0000-0002-4384-6300
Alfredo Liverani http://orcid.org/0000-0002-3255-9381
Alessandro Ceruti http://orcid.org/0000-0001-7947-3932

References

[1] Moon SK, Tan YE, Hwang J, et al. Application of 3D printing technology for designing light-weight unmanned aerial vehicle wing structures. Int J Precis Eng Manuf-Green Tech. 2014;1:223–228. doi:10.1007/s40684-014-0028-x
[2] Pecho P, Ažaltovič V, Kandera B, et al. Introduction study of design and layout of UAVs 3D printed wings in relation to optimal lightweight and load distribution. Trans Res Proc. 2019;40:861–868. doi:10.1016/j.trpro.2019.07.121
[3] Mantovani S, Campo G, Giacalone M. Steering column support topology optimization including lattice structure for metal additive manufacturing. In: Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science; 2020. doi:10.1177/0954406220947121
[4] Chinthavali M. 3D printing technology for automotive applications. In: 2016 International Symposium on 3D Power Electronics Integration and Manufacturing (3D-PEIM), IEEE, Raleigh, NC, USA; 2016.p. 1–13. doi:10.1109/3DPEIM.2016.7570535
[5] Beyer C. Strategic implications of current trends in additive manufacturing. J Manuf Sci Eng. 2014;136:064701. doi:10.1115/1.4028599
[6] Plocher J, Panesar A. Review on design and structural optimisation in additive manufacturing: towards next-generation lightweight structures. Mater Des. 2019;183:108164. doi:10.1016/j.matdes.2019.108164
[7] Huang R, Riddle M, Graziano D, et al. Energy and emissions saving potential of additive manufacturing: the case of lightweight aircraft components. J Cleaner Prod. 2016;135:1559–1570. doi:10.1016/j.jclepro.2015.04.109
[8] Vigliotti A, Pasini D. Stiffness and strength of tridimensional periodic lattices. Comput Methods Appl Mech Eng. 2012;229–232:27–43. doi:10.1016/j.cma.2012.03.018
[9] Ashby MF. The properties of foams and lattices. Phil Trans R Soc A. 2006;364:15–30. doi:10.1098/rsta.2005.1678
[10] Dumas M, Terriault P, Brailovski V. Modelling and characterization of a porosity graded lattice structure for additively manufactured biomaterials. Mater Des. 2017;121:383–392. doi:10.1016/j.matdes.2017.02.021
[11] Wang Y, Zhang L, Daynes S, et al. Design of graded lattice structure with optimized mesostructures for additive manufacturing. Mater Des. 2018;142:114–123. doi:10.1016/j.matdes.2018.01.011
[12] Campo GA, Vettorello A, Giacalone M. Optimization methodology for continuous heterogeneous structures: A preliminary design of an engine mounting bracket. KEM. 2019;827:116–121. doi:10.4028/www.scientific.net/KEM.827.116
[13] Attaran M. The rise of 3-D printing: the advantages of additive manufacturing over traditional manufacturing. Bus Horiz. 2017;60:677–688. doi:10.1016/j.bushor.2017.05.011
[14] Hull CW. Apparatus for production of three-dimensional objects by stereolithography, US4575330A; 1986. https://patents.google.com/patent/US4575330A/en (accessed August 29, 2024).
[15] Guo N, Leu MC. Additive manufacturing: technology, applications and research needs. Front Mech Eng. 2013;8:215–243. doi:10.1007/s11465-013-0248-8
[16] Ngo TD, Kashani A, Imbalzano G, et al. Additive manufacturing (3D printing): a review of materials, methods, applications and challenges. Compos B Eng. 2018;143:172–196. doi:10.1016/j.compositesb.2018.02.012
[17] Zhang X, Cui W, Li W, et al. A hybrid process integrating reverse engineering, Pre-repair processing, additive manufacturing, and material testing for component remanufacturing. Materials (Basel). 2019;12:1961. doi:10.3390/ma12121961
[18] Pereira T, Kennedy JV, Potgieter J. A comparison of traditional manufacturing vs additive manufacturing, the best method for the job. Proc Manuf. 2019;30:11–18. doi:10.1016/j.promfg.2019.02.003
[19] Frizziero L, Donnici G, Venditti G, et al. Design of an innovative sanitation system for bike-sharing service. Heliyon. 2024;10:e26595. doi:10.1016/j.heliyon.2024.e26595
[20] Bassoli E, Mantovani S, Giacalone M, et al. On the technological feasibility of additively manufactured self-supporting AlSi10Mg lattice structures. Adv Eng Mater. 2023;25:2201074. doi:10.1002/adem.202201074
[21] Tao W, Leu MC. Design of lattice structure for additive manufacturing. In: 2016 International Symposium on Flexible Automation (ISFA), IEEE, Cleveland, OH, USA; 2016. p. 325–332. doi:10.1109/ISFA.2016.7790182
[22] Donnici G, Freddi M, Liverani A. RSM applied to lattice patterns for stiffness optimization. RPJ. 2024;30:345–356. doi:10.1108/RPJ-03-2024-0134
[23] Dragoni E, Ciace VA. Mechanical design and modelling of lightweight additively manufactured lattice structures evolved from regular three-dimensional tessellations. Proc Inst Mech Eng Part C: J Mech Eng Sci. 2021;235:1759–1773. doi:10.1177/0954406219885959
[24] Savio G, Meneghello R, Concheri G. Geometric modeling of lattice structures for additive manufacturing. RPJ. 2018;24:351–360. doi:10.1108/RPJ-07-2016-0122
[25] Tonelli D, Pietroni N, Puppo E, et al. Stability of statics aware voronoi grid-shells. Eng Struct. 2016;116:70–82. doi:10.1016/j.engstruct.2016.02.049
[26] Cresswell ND, Ameri AAH, Wang J, et al. Characterization and modelling of triply periodic minimum surface (TPMS) lattice structures for energy absorption in automotive applications. In: Z Peng, M Zhang, J Li, B Li, SN Monteiro, R Soman, J-Y Hwang, YE Kalay, JP Escobedo-Diaz, JS Carpenter, AD Brown, S Ikhmayies, editors. *Characterization of minerals, metals, and materials 2024*. Cham: Springer Nature Switzerland; 2024. p. 295–305. doi:10.1007/978-3-031-50304-7_28
[27] Li D, Liao W, Dai N, et al. Optimal design and modeling of gyroid-based functionally graded cellular structures for additive manufacturing. Comput-Aid Des. 2018;104:87–99. doi:10.1016/j.cad.2018.06.003

[28] Maskery I, Sturm L, Aremu AO, et al. Insights into the mechanical properties of several triply periodic minimal surface lattice structures made by polymer additive manufacturing. Polymer. 2018;152:62–71. doi:10.1016/j.polymer.2017.11.049

[29] Defanti S, Giacalone M, Mantovani S, et al. Dimensional and mechanical assessment of gyroid lattices produced in aluminum by laser powder bed fusion. Meccanica. 2025;60:633–646. doi:10.1007/s11012-024-01854-7

[30] Tamburrino F, Graziosi S, Bordegoni M. The design process of additively manufactured mesoscale lattice structures: a review. J Comput Inf Sci Eng. 2018;18:040801-01–040801-16. doi:10.1115/1.4040131

[31] Syam WP, Jianwei W, Zhao B, et al. Design and analysis of strut-based lattice structures for vibration isolation. Prec Eng. 2018;52:494–506. doi:10.1016/j.precisioneng.2017.09.010

[32] Mantovani S, Giacalone M, Merulla A, et al. Effective mechanical properties of AlSi7Mg additively manufactured cubic lattice structures. 3D Print Addit Manuf. 2022;9:326–336. doi:10.1089/3dp.2021.0176

[33] Azman AH, Vignat F, Villeneuve F. CAD tools and file format performance evaluation in designing lattice structures for additive manufacturing. J Teknol. 2018;80:87–95. doi:10.11113/jt.v80.12058

[34] Savio G, Rosso S, Meneghello R, et al. Geometric modeling of cellular materials for additive manufacturing in biomedical field: a review. Appl Bionics Biomech. 2018;2018:1–14. doi:10.1155/2018/1654782

[35] Rosen DW. Computer-Aided design for additive manufacturing of cellular structures. Comput Aid Des Appl. 2007;4:585–594. doi:10.1080/16864360.2007.10738493

[36] Dong G, Tang Y, Zhao YF. A survey of modeling of lattice structures fabricated by additive manufacturing. J Mech Des. 2017;139:100906. doi:10.1115/1.4037305

[37] Aremu AO, Brennan-Craddock JPJ, Panesar A, et al. A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing. Addit Manuf. 2017;13:1–13. doi:10.1016/j.addma.2016.10.006

[38] Omigbodun FT, Oladapo BI. AI-Optimized Lattice structures for biomechanics scaffold design. Biomimetics. 2025;10:88. doi:10.3390/biomimetics10020088

[39] Yüksel N, Eren O, Börklü HR, et al. Mechanical properties of additively manufactured lattice structures designed by deep learning. Thin-Walled Struct. 2024;196:111475. doi:10.1016/j.tws.2023.111475

[40] Wang H, Chen Y, Rosen DW. A hybrid geometric modeling method for large scale conformal cellular structures. In: Volume 3: 25th Computers and Information in Engineering Conference, Parts A and B, ASMEDC, Long Beach, California, USA; 2005. p. 421–427. doi:10.1115/DETC2005-85366

[41] Vongbunyong S, Kara S. Rapid generation of uniform cellular structure by using prefabricated unit cells. Int J Comput Integr Manuf. 2017;30:792–804. doi:10.1080/0951192X.2016.1187303

[42] Montalti A, Ferretti P, Santi GM. From CAD to G-code: strategies to minimizing errors in 3D printing process. CIRP J Manuf Sci Technol. 2024;55:62–70. doi:10.1016/j.cirpj.2024.09.005

[43] Adam A. Converting a 3D logical array into an STL surface mesh; 2021. https://www.mathworks.com/matlabcentral/fileexchange/27733-converting-a-3d-logical-array-into-an-stl-surface-mesh (accessed February 17, 2021).

[44] Azman AH, Vignat F, Villeneuve F, et al. Creation of lattice structures with skeleton model for additive manufacturing. Int J Interact Des Manuf. 2021;15:381–396. doi:10.1007/s12008-021-00767-z

[45] Di Angelo L, Di Stefano P, Guardiani E. An advanced GCode analyser for predicting the build time for additive manufacturing components. ACTA IMEKO. 2020;9:30. doi:10.21014/acta_imeko.v9i4.728

[46] Chougrani L, Pernot J-P, Véron P, et al. Lattice structure lightweight triangulation for additive manufacturing. Comput-Aid Des. 2017;90:95–104. doi:10.1016/j.cad.2017.05.016

[47] Paul Chew L. Constrained delaunay triangulations. Algorithmica. 1989;4:97–108. doi:10.1007/BF01553881

[48] Lorensen WE, Cline HE. Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput Graph. 1987;21:163–169. doi:10.1145/37402.37422

[49] Li Y, Cheng Y, Hu Q, et al. The influence of additive manufacturing on the configuration of make-to-order spare parts supply chain under heterogeneous demand. Int J Prod Res. 2019;57:3622–3641. doi:10.1080/00207543.2018.1543975

[50] McMillan ML, Jurg M, Leary M, et al. Programmatic generation of computationally efficient lattice structures for additive manufacture. Rapid Prototyp J. 2017;23:486–494. doi:10.1108/RPJ-01-2016-0014

[51] Cignoni P, Callieri M, Corsini M, et al. MeshLab: an open-source mesh processing tool. In: V Scarano, RD Chiara, U Erra, editors. Eurographics Italian Chapter Conference. The Eurographics Association; 2008. doi:10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136

[52] Maltsev E, Popov D, Chugunov S, et al. An accelerated slicing algorithm for frep models. Appl Sci. 2021;11:6767. doi:10.3390/app11156767

[53] Savchenko VV, Pasko AA, Okunev OG, et al. Function representation of solids reconstructed from scattered surface points and contours. Comput Graph Forum. 1995;14:181–188. doi:10.1111/1467-8659.1440181

[54] Chen R, Wang S, Wu Z, et al. Compressive enhancement gyroid lattice with implicit modeling implementation and modified G-A model property prediction. Mater Des. 2023;232:112153. doi:10.1016/j.matdes.2023.112153

[55] Tang Y, Dong G, Zhao YF. A hybrid geometric modeling method for lattice structures fabricated by additive manufacturing. Int J Adv Manuf Technol. 2019;102:4011–4030. doi:10.1007/s00170-019-03308-x

[56] Fryazinov O, Vilbrandt T, Pasko A. Multi-scale space-variant FRep cellular structures. Comput-Aid Des. 2013;45:26–34. doi:10.1016/j.cad.2011.09.007

[57] Pasko A, Fryazinov O, Vilbrandt T, et al. Procedural function-based modelling of volumetric microstructures. Graph Models. 2011;73:165–181. doi:10.1016/j.gmod.2011.03.001

[58] Al-Ketan O, Abu Al-Rub RK. MSLattice: a free software for generating uniform and graded lattices based on triply periodic minimal surfaces. Mat Des Process Comms. 2021;3(6):e205, 1–10. doi:10.1002/mdp2.205

[59] nTop. 2025. https://www.ntop.com/software/products/ (accessed February 14, 2025).
[60] Bacciaglia A, Ceruti A, Liverani A. Structural analysis of voxel-based lattices using 1D approach. 3D Print Addit Manuf. 20219(5):365–379. doi:10.1089/3dp.2020.0178
[61] Jense GJ. Voxel-based methods for CAD. Comput-Aid Des. 1989;21:528–533. doi:10.1016/0010-4485(89)90061-4
[62] Matlab; 2025.
[63] Vega G, Paz R, Gleadall A, et al. Comparison of CAD and voxel-based modelling methodologies for the mechanical simulation of extrusion-based 3D printed scaffolds. Materials (Basel). 2021;14:5670. doi:10.3390/ma14195670
[64] Wang SW, Kaufman AE. Volume-sampled 3D modeling. IEEE Comput Grap Appl. 1994;14:26–32. doi:10.1109/38.310721
[65] Adam A. Mesh voxelisation; n.d. https://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation (accessed May 28, 2025).
[66] Wallach JC, Gibson LJ. Mechanical behavior of a three-dimensional truss material. Int J Solids Struct. 2001;38:7181–7196. doi:10.1016/S0020-7683(00)00400-5
[67] Deshpande VS, Ashby MF, Fleck NA. Foam topology: bending versus stretching dominated architectures. Acta Mater. 2001;49:1035–1040. doi:10.1016/S1359-6454(00)00379-7
[68] Solidworks; n.d.
[69] Cloud Compare; 2025. https://www.cloudcompare.org/ (accessed February 19, 2025).
[70] Moerman KM. GIBBON: the geometry and image-based bioengineering add-on. JOSS. 2018;3:506. doi:10.21105/joss.00506
[71] Dawson-Haggerty et al. Trimesh; 2019. https://trimesh.org/ (accessed January 31, 2025).
[72] Fuji Tsang C, Shugrina M, Lafleche JF, et al. Kaolin: a pytorch library for accelerating 3D deep learning research; 2024. https://github.com/NVIDIAGameWorks/kaolin.
[73] Braam D. Ultimaker Cura; n.d. https://ultimaker.com/it/software (accessed February 21, 2025).
[74] Ceruti A, Ferrari R, Liverani A. Design for additive manufacturing using LSWM: a CAD tool for the modelling of lightweight and lattice structures. In: G Campana, RJ Howlett, R Setchi, B Cimatti, editors. *Sustainable design and manufacturing 2017*. Cham: Springer International Publishing; 2017. p. 756–765. doi:10.1007/978-3-319-57078-5_71
[75] Volume lattice; 2024. https://superhivemarket.com/products/volume-lattice/?ref=4241 (accessed May 12, 2025).
[76] Bastos L. Lattice design for additive manufacturing in solidworks; 2021.
[77] Brandts J, Korotov S, Křížek M. On the equivalence of regularity criteria for triangular and tetrahedral finite element partitions. Comput Math Appl. 2008;55:2227–2233. doi:10.1016/j.camwa.2007.11.010
[78] General Electric Company. GE jet engine bracket challenge; 2013. https://grabcad.com/challenges/ge-jet-engine-bracket-challenge (accessed February 25, 2025).
[79] Creality. Cr Scan-01; 2020. https://rb.gy/54pdum (accessed March 13, 2023).
[80] Bacciaglia A, Falcetelli F, Di Sante R, et al. Indoor replication of outdoor climbing routes: fidelity analysis of digital manufacturing workflow. Prog Addit Manuf. 2023;9:1811–1823. doi:10.1007/s40964-023-00540-6

Appendix

A.1. STL to Graded Voxel Conversion

Inputs:

- STL_file: Path to STL surface mesh
- voxel_size: Desired voxel resolution (edge length)
- grading_direction: Unit vector defining direction of grading
- grading_function: Scalar field function (λ(s)) along grading direction or scalar value

Outputs:

- VoxelGrid: 3D logical array indicating voxel occupancy
- Nx, Ny, Nz: Number of voxels along x, y, z
- origin: Lower-bound corner of the STL bounding box
- bbox: Bounding box of the STL geometry (min/max corners)

Procedure:

1. Load STL mesh and extract vertex coordinates;
2. Compute bounding box of the STL;
3. Set the origin of the bounding box dimensions
4. Compute number of voxels in each direction Nx, Ny, and Nz;
5. Voxelize the STL using the Adam's function [65]:

VoxelGrid ← VOXELISE(Nx, Ny, Nz, STL_file, ray_directions = 'xyz');

6. Generate voxel centre coordinates:

For each voxel index (i, j, k), compute centre position (x, y, z);

7. Project all voxel centres onto the grading direction:

s_i ← dot(center_i, grading_direction)

8. Normalise scalar projections using minimum and maximum values;
9. Evaluate local scale factor:

If grading_function is scalar:
λ_i ← 1
Else:
λ_i ← grading_function(ŝ_i)

10. Compute scaled voxel centres:

center_i_scaled ← center_i + (λ_i - 1) · (s_i · grading_direction)

11. Mask out voxels outside original bounding box after scaling:
For each voxel:
If center_i_scaled ∉ bbox:
VoxelGrid(i, j, k) ← 0

12. Remove isolated voxels:
For each voxel:
If voxel is active and all 26 neighbours are inactive:
Set voxel to inactive

Return:
- VoxelGrid (logical 3D array)
- Nx, Ny, Nz
- origin
- bbox

A.2. Single Unit Cell Wireframe Generation

Inputs:
- Length of the unit cell edge
- Lattice type ('Box', 'BCC', 'FCC', 'Octet', 'Octahedron', 'CVC')
- origin: Lower corner coordinate of the unit cell

Outputs:
- N: Unique node coordinates of the unit cell
- M: Connectivity matrix (list of node index pairs forming beam elements)

Procedure:

1. Set unit cell size
2. Based on type, define node coordinates (nodes) using origin and unit cell geometry
3. Remove duplicate nodes to get N
4. Initialise M and, based on unit cell type, loop over all node pairs (n, m) to define beam connections and save in M

Return:
- List of unique node coordinates
- Beam element connectivity matrix

A.3. Solid Cylinder Implicit Function

Inputs:
- Starting point of the cylinder axis
- Ending point of the cylinder axis
- Radius of the cylinder

Output:
- cylinderImplicit(x, y, z): Function evaluating the implicit solid cylinder field at any 3D point

Procedure:
1. Compute the axis direction vector and normalise it;
2. Define projection of any point $P = (x, y, z)$ onto the axis;
3. Clamp the projection to cylinder height
4. Compute the closest point on the axis:
5. Compute squared radial distance from axis:
6. Define the implicit function for lateral surface:
7. Define implicit function for the bottom cap:
8. Define implicit function for the top cap:
9. Combine all to define the solid cylinder:

cylinderImplicit(x, y, z) ← max(lateral_surface(x, y, z), max (bottom_cap(x, y, z), top_cap(x, y, z)))

Return:

- cylinderImplicit: A scalar field function, ≤ 0 inside the solid cylinder, > 0 outside, = 0 on the boundary

A.4 Graded lattice scaling

Inputs:

- type of grading (none, scalar or function-based)
- gradingDirection
- gradingFunction: Scalar or function handle defining the grading profile
- fv_full: Structure containing vertices and faces of the lattice mesh
- bbox: Target bounding box (min/max corners) of the domain to be filled

Output:

- fv_full: Modified mesh structure with transformed vertex coordinates

Procedure:

1. If type_grading = 'scalar':

Multiply each vertex by a scalar vector: fv_full.vertices ← fv_full.vertices .* gradingFunction
Else if type_grading = 'functi':

a. Project each vertex onto the grading direction: s ← dot(-vertex, gradingDirection)
b. Normalise the projections: ŝ ← (s − s_min) / (s_max − s_min)
c. Evaluate local scale factor: λ ← gradingFunction(ŝ)
d. Compute scaled position along the grading direction: fv_full.vertices ← fv_full.vertices + + (λ − 1) .* (s * gradingDirection)

Else if type_grading = 'nothin':
Leave the vertices unchanged.

2. Rescale the graded lattice to fit inside the target bounding box:
a. Compute bounding box of the transformed lattice
b. Compute lattice size and target size:
c. Compute non-uniform scale factors:
d. Translate the lattice to origin:
e. Apply non-uniform scaling:
f. Translate to target bounding box:

Return:

- fv_full: The graded and spatially aligned lattice mesh, ready for Boolean operations or export

A.5 Main code

1. Input lattice parameters (strut length, strut radius, unit cell type, grading type and direction)
2. Call gradingDefinition(type_grading) to obtain a function handle for the grading
3. Load the STL file of the inner design domain

4. Voxelize the STl domain by calling *STL to Graded Voxel Conversion*
5. Compute the number of unit cells for replication
6. Set the resolution for the implicit function modelling
7. Generate the unit cell wireframe model by calling *Single Unit Cell Wireframe Generation*
8. Subdivide beam bounding box for implicit evaluation and generate 3D meshgrid of evaluation points
9. For each vertex in the unit cell:
   a. Define implicit function for a sphere φ_sphere(x, y, z)
   b. Update the overall unit cell field: combinedVolume ← min(combinedVolume, φ_sphere)
10. For each edge in the unit cell:
   a. Let A ← vertices(i) and B ← vertices(j)
   b. Define implicit function for solid cylinder between A and B by calling *Solid Cylinder Implicit Function*
   c. Update the field: combinedVolume ← min(combinedVolume, φ_cylinder)
11. Define isosurface level (e.g. 0)and extract the isosurface from the implicit field
12. Initialise fv_full, the combined mesh of the entire lattice structure (vertices and faces) with an empty structure
13. Loop through the 3D grid of unit cells. For each (x, y, z) in [0, numCells.x−1] × [0, numCells.y−1] × [0, numCells.z−1]:
   a. Check if the voxel is active
   b. Compute spatial offset for this cell
   c. Translate unit cell vertices
   d. Append new vertices to fv_full.vertices
   e. Shift face indices and append
   f. Update face index offset
14. Validate triangulation with GIBBON toolbox:
   a. Extract triangle definitions from fv_full.faces
   b. Keep only valid triangles (those with 3 unique vertices)
15. Apply lattice grading by calling *Graded lattice scaling* to rescale the lattice mesh based on the defined spatial grading strategy
16. Export the parallelepiped design domain as a graded lattice to STL file, excluded the inactive voxels
17. Compute Boolean intersection between the STL file of graded lattice and design volume using *trimesh* Python library; this ensures that only the portion of the lattice inside the design domain is preserved.
18. Compute Boolean union between the STL file of intersected lattice and shell design domain.