



# Article DDSM: Design-Oriented Dual-Scale Shape-Material Model for Lattice Material Components

Chao Feng, Rui Yang, Bin Niu \* and Xiangpeng Meng

School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China \* Correspondence: niubin@dlut.edu.cn

Abstract: This paper proposes a new CAD model for the design of lattice material components. The CAD model better captures the user's design intent and provides a dual-scale framework to represent the geometry and material distribution. Conventional CAD model formats based on B-Rep generate millions of data files, which also makes design intent and material information missing. In the present work, a new shape-material model for lattice material components is proposed. At the macroscopic scale, a compact face-based non-manifold topological data structure is proposed to express the lattice shape-material information without ambiguity. At the microscopic scale, implicit function is adopted for the representation of lattice material components. Numerical experiments verify that the proposed CAD model provides a powerful support for design intent with minor space costs. Meanwhile, the representation method supports solid modeling queries of geometric and material information on each scale.

Keywords: lattice material components; dual-scale model; non-manifold topology; implicit representation



Citation: Feng, C.; Yang, R.; Niu, B.; Meng, X. DDSM: Design-Oriented Dual-Scale Shape-Material Model for Lattice Material Components. *Materials* 2022, *15*, 7428. https:// doi.org/10.3390/ma15217428

Academic Editor: Evgeny V. Shilko

Received: 23 September 2022 Accepted: 20 October 2022 Published: 23 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

# 1. Introduction

Different from conventional components made of the same material uniform, lattice material component (LMC) refers to objects with spatially homogeneous material and different periodic microstructures. The LMC is widely used in the lightweight design of aerospace structures due to its excellent mechanical properties (e.g., high specific strength, stiffness [1] and shock absorption [2]). Based on homogenization theory, the LMC is regarded as a component consisting of multiple homogeneous materials at the macroscopic scale. At the microscopic scale, the LMC is composed of a periodic array of different types of unit cells. The LMC provides more design freedom for designers to further control the distribution of equivalent properties of materials at the macroscopic scale by controlling both material compositions and their microstructures [3,4].

The CAD model, as an explicit reflection of the user's design intent, is the basis for subsequent simulation and manufacturing. It is crucial to propose a new CAD model for the LMC dual-scale characteristics. Conventional CAD models are not efficient at representing lattice structures [5]. Since the manufacturing of lattice structures is mostly achieved by additive manufacturing [6], about 80% of lattice files are stored in stereolithography format (STL) [7]. However, the triangular mesh-based CAD model can only reflect the geometric information of the components discretely, ignoring the dual-material information (i.e., homogenized equivalent materials and manufacturing materials). In addition, the STL file will be useless when the microstructures need to be modified. Consequently, a CAD model that can completely and efficiently represent and handle the shape-material information of the LMC is extremely critical. The shape information includes geometric information and topological relationships. Additionally, the material information includes homogenized equivalent materials and manufacturing materials. Therefore, there are two basic challenges which require a solution to the CAD model. Firstly, the model needs to reflect clear design intent. Secondly, it needs to contain the complete shape-material information with minor space costs.

Spatial decomposition as a CAD model scheme includes the voxel model and the mesh model [8]. A voxel-based method was proposed for the generation of trimmed lattice structures [9]. The overall material distribution is determined by specifying a material composition to each voxel directly. However, the problems with this method are the large memory overhead [10] and the complexity of the material information query [11]. In terms of the volume mesh-based model, the shape-material model is represented as a collection of polyhedrons [12–14]. Each polyhedron is represented as a list of vertices, where the geometric position, as well as the material composition information, can be stored. However, the homogenized equivalent material information makes internal boundary vertices appear ambiguous in the storage process.

In contrast to the spatial decomposition, the analytical models separate geometric and material information. The representations based on the analytical models can be divided into two categories. In the first analytical model representation, the boundary representation (B-Rep) is used to represent geometric information. The material information is described by some specific functions (e.g., explicit functions [15] and distance function [16]). For example, The B-spline volume is utilized to represent the shape-material model. Additionally, the material information of the model is attached to each control point [17,18]. Second, the implicit functions are used to represent point set geometry and the material distribution. Specifically, the implicit function-based models reduce memory consumption compared to the STL files [19]. However, the dual-material information distributed on two scales is not represented by such an analytical model.

The composite model is used to represent components with two or more types of material distributions. The composite model is proposed based on the idea of spatial decomposition. There are two main modeling methods as follows. Firstly, the component is modeled by regularized operators of sub-objects, such as difference, intersection, union, etc. However, this approach faces the problem of heavily data redundant [20], cumbersome for regularized Boolean operations [21], and material ambiguity [22]. Secondly, the non-manifold geometric representation is used to produce a composite model [23]. However, it requires complex data structures and algorithms to construct the topological relationships for sharing the same boundary.

According to the existing material component modeling methods, the LMC CAD model representation has the following problems to be solved.

- 1. Lack of dual-material model representation: Due to the dual-material properties on the lattice CAD model, the geometric model based on STL file format does not provide a complete representation of the material information. Moreover, this renders the model computationally expensive for the material query. Therefore, there is an urgent need to propose a dual-scale shape-material model, which supports material information queries on two scales.
- 2. Large storage cost. As mentioned before, most lattice structures are stored in the STL format. A large number of hollowed-out structures produce huge triangular facets to store geometry information. These triangular facets take up large memory and make the information redundant extremely.
- 3. Missing design intent: In the design-oriented process, designers prefer to focus on the overall equivalent material properties (elastic modulus, Poisson's ratio, etc.) embodied in design domain, rather than on the specific configuration and manufacturing material of the unit cell. However, the existing CAD tools and model representation format (e.g., Delaunay triangulation) lack equivalent macroscopic material properties, which makes CAD models poorly designable and revisable.

In this paper, we strive to develop a new CAD shape-material model to solve the above problems. We refer to our model as Design-oriented Dual-scale Shape-material Model (DDSM). The main contributions of this work are summarized as: First, a new model representation method is proposed to meet the modern LMC design needs. Second, a new non-manifold topological data structure is used to represent macroscopic shape-material model. Third, an implicit function-based representation of truss-like struts is introduced

at the microscopic scale. Consequently, this new dual-scale representation provides more intuitive the design intent and more efficient storage than conventional CAD model.

The remainder of the paper is organized as follows. Section 2 presents a mathematical framework of DDSM. Section 3 proposes a new data structure for finite non-manifold characteristic at the macroscopic scale. Section 4 provides geometric representation based on implicit functions at the microscopic scale. Section 5 proposes the DDSM formulation and advantages. Section 6 is the conclusions of this research.

## 2. Mathematical Frameworks for the DDSM

The CAD models of lattice structure are usually converted into STL files. Therefore, it is less convenient for designers to capture the design intents. Firstly, for the design process of the LMC, the ideal design should be optimized for performance based on the specific functional requirements. Then, each sub-design domain obtains the optimal material distribution such as elastic modulus [24,25], Poisson's ratio [26], shear modulus [27] etc. Finally, the unit cell configuration is chosen based on macroscopic material distribution. The ideal design process for the LMC is shown in Figure 1. The corresponding CAD model needs to reflect the ideal design process of the LMC and not just the discrete geometric information.



Figure 1. The CAD model of the ideal lattice material component.

Depending on the design intent of the lattice material component, a material space is introduced based on the geometric space. The material space is made up of equivalent material properties with corresponding unit cell configurations. The LMC model representation is essentially a product space consisting of the geometric space as the base space and the material space as the bundle space. Thus, the information of any point in space can be expressed by a dual pair of geometric space and material space. The dual pairs realize the partitioning of the geometric space. The partitioning results in a collection of sub-objects, and each sub-object corresponds to a material space. From the perspective of topology, the point set partitioning forms a finite number of continuous open subsets, and the intersection of each open subset is empty. The union of all open subsets is the original geometric region. Therefore, the CAD model *S* for the LMC can be composed of several sub-objects. Each sub-object is described by geometric information and material information, i.e.,

$$S = \{S_1, S_2, \dots, S_n\}$$
  

$$S_i = \{(P, M) | P \in \Omega_g^{(i)}, M \in \Omega_m^{(i)}, 1 \le i \le n\}$$
(1)

where  $S_i$  is the *i*-th sub-object, *P* is the geometric description of the sub geometric space  $\Omega_g^{(i)}$ , and *M* is the material description in the sub material space  $\Omega_m^{(i)}$ .

In summary, the LMC CAD model describes geometric and material information in two dimensions: macroscopic scale and microscopic scale. The dual-scale CAD model needs to consider the material information while representing the geometric information on each scale. The correspondence in mechanical behavior between the two scales is expressed in terms of equivalent material properties computed by the unit cell [28,29].

#### 3. Macroscopic Scale: A Compact Face-Based Topological Data Structure

# 3.1. Finite Non-Manifold Characteristic

The geometric information, as the base space of the material information, is the key to the LMC CAD model. However, the conventional model representation based on Regular Set and Manifold Theory mathematically faces ambiguous representation of boundary materials as shown in Figure 2. For example,  $C_1$  and  $C_2$  correspond to different material space. Points on the internal boundary MN cannot define their equivalent material attribution when storing material information by B-Rep. The ambiguity of material information causes the typical non-manifold of CAD models at the macroscopic scale.



Figure 2. The ambiguity of inner boundary. MN is the edge shared by adjacent material spaces.

According to Ellul [30] "in 3D, for the manifold to be valid, the neighborhood of each point within the sub-space must be able to be deformed into a sphere. Thus, self-intersecting surfaces are non-manifold". It usually consists of a finite number of manifolds sharing boundaries as shown in Figure 3.



Figure 3. A non-manifold object formed by sharing (a) a vertex; (b) an edge; (c) a face.

Due to the internal boundary material ambiguity, the LMC model appears as a "finite non-manifold", i.e., it is simply limited to the presence of shared boundaries between material regions as shown in Figure 3c. Our goal is to design efficient and simple topological data structures for the finite non-manifold characteristic. The current non-manifold data structures are classified as the edge-based data structure and the face-based data structure. The edge-based data structures often utilize pseudo-manifold representation for non-manifold geometry. For example, a non-manifold shape with shared edges can be transformed into a manifold with two edges infinitely close to each other [31]. The face-based data structure was proposed for non-manifold and non-regular simplicial complexes. However, the storage of topological information in intermediate dimensions causes unnecessary memory overhead [32].

## 3.2. A compact Half-Face Data Structure Design

A well-defined CAD data structure defines the topological relationships of geometric entities completely. For the macroscopic LMC model, it is desirable to meet the following requirements:

- Generality: support for macroscopic non-manifold geometry CAD representations.
- Time efficiency: support specific topology queries without performing global search.
- Memory overhead: require a minimal amount of storage.

In this paper, we propose a mesh-based data structure to represent the LMC macroscopic model. The data structure establishes complete topological relationships. We refer to our data structure as the Compact Array-based Half-Face (CAHF) data structure oriented to finite non-manifold. First, we define several entity concepts. If a face has two incident elements, we refer to such face as twin half-face. A face without any twin is a border half-face. Vertices incident on only one element are called manifold border vertices.

The local numbering conventions in the CAHF data structure is shown in Figure 4. Moreover, the elements in the volume mesh carry more topology information. For example, the order of storing vertices implicitly expresses the order of faces(2-D) and edges(1-D), which does not be stored but referenced implicitly. This representation makes full use of the implicit topology information between vertices and faces within the element. Additionally, this effectively reduces memory overhead.

| _     | Half-face index     | Vertices             |   | Edge index | Vertices       |
|-------|---------------------|----------------------|---|------------|----------------|
|       | $f_1$               | $v_1, v_2, v_3, v_4$ | • | $e_1$      | $v_1, v_2$     |
| 8 6 5 | f.                  | V V V V.             |   | $e_2$      | $v_2, v_3$     |
|       | J <sub>2</sub><br>f | ,1,,2,,5,,6          |   | $e_3$      | $v_{3}, v_{4}$ |
| ④ ③   | $J_3$               | $V_2, V_3, V_6, V_7$ |   | $e_4$      | $v_4, v_1$     |
| 3 2 2 | $f_4$               | $v_3, v_4, v_7, v_8$ |   |            |                |
|       | $f_5$               | $v_1, v_4, v_5, v_8$ |   |            |                |
| 4 1   | $f_{6}$             | $v_5, v_6, v_7, v_8$ |   |            |                |

**Figure 4.** Local numbering conventions and implicit topology information for 3-D elements. The circled numbers correspond to local face IDs, and the unmarked ones correspond to local vertex IDs in the standard 3-D element.

More specifically, each face ID is composed of a pair of numbers  $\langle E_{id}, F_{id} \rangle$ , where  $E_{id}$  denotes the element ID (starting from 1), and  $F_{id}$  denotes the local face ID. The number of elements in a complex mesh model can be in the millions, and the element ID requires a large range of integer data. Additionally, the maximum local face ID is six. To avoid the space consumption caused by memory alignment, the face ID is coded as a single 32-bit unsigned integer. The element ID is stored in the first twenty-nine bits and the last three bits store the local face ID [33]. Therefore, about 500 million elements can be stored based on the half-face coding.

Based on the above topological conventions, the CAHF data structure oriented to finite non-manifold characteristic is proposed as shown in Figure 5. Volume meshes contain element connectivity and node coordinates. The element connectivity and node coordinates are used as input. These are stored in Class *Element*. Class *Element* also opens a one-dimensional array that stores the twin half-face of each face in the element. Class *HalfFacet* is an encoding and decoding operation for half-face information. Class *VtxHalfface* is used to store the topological relationships by Mesh Topology Reconstruction.



**Figure 5.** Class diagram of the CAHF volume mesh model. Let *CellIndex*, *FaceIndex* and *VtxIndex* denote the ID of element, half-face and vertex, respectively. *SmallIndex* is the local ID of the face. The type of these variables is unsigned int.

# 3.3. Macroscopic CAD Model Topology Reconstruction

For the DDSM macroscopic model to handle geometric information, it is desirable to reconstruct the topological relationships. Therefore, we define the arrays for the CAHF data structure similar to that in Dyedov et al. [34]:

- *m2hfs*: Map each half-face to the ID of its twin half-face;
- *v2hfs*: Map each non-manifold vertex to its incident half-face;
- *b2hfs*: Map manifold border vertex to its incident element ID and its local ID;
- *bhfsm*: Map each border half-face to its incident element ID and its local ID.

Specifically, the topology reconstruction consists of the following two algorithms. Algorithm 1 describes the identification of twin half-faces. The identification starts by traversing the faces on each element and storing the index of the largest vertex on each face in the temporary array *v2fs*. Then, the adjacent vertices of the largest vertex on each face are stored in the temporary array *v2adj*. Finally, the twin half-face is matched through two temporary arrays.

| Algorithm 1: Construction of twin half-faces                       |  |  |  |  |
|--|--|--|--|--|
| Input: element connectivity, CElement                              |  |  |  |  |
| <b>Output:</b> mutual mapping of twin half-faces, <i>m2hfs</i>     |  |  |  |  |
| begin  |  |  |  |  |
| Allocate the initial <i>Element</i> and define <i>m2hfs</i> zeros  |  |  |  |  |
| for each <i>e</i> in <i>CElement</i> do                            |  |  |  |  |
| $f \leftarrow$ traverse each face in $e$ ;                         |  |  |  |  |
| $ID_{max} \leftarrow$ get the maximum node identification on $f$ ; |  |  |  |  |
| set mapping $ID_{max}$ to the corresponding face into $v2fs$ ;     |  |  |  |  |
| set mapping $ID_{max}$ to adjacent vertices into $v2adj$ ;         |  |  |  |  |
| end for  |  |  |  |  |
| for each <i>e</i> in <i>CElement</i> do                            |  |  |  |  |
| $f \leftarrow$ traverse each face in $e$ ;                         |  |  |  |  |
| if $m2hfs$ (f) = 0 then  |  |  |  |  |
| $ID_{max} \leftarrow$ get the maximum node identification on $f$ ; |  |  |  |  |
| get candidate faces of mutual mapping on <i>f</i> ;                |  |  |  |  |
| match twin half-face of $f$ in $v2adj$ ;                           |  |  |  |  |
| end if   |  |  |  |  |
| end for  |  |  |  |  |
| end  |  |  |  |  |

Algorithm 2 implements the mapping between the different dimensional entities. Specifically, non-manifold vertices are mapped in the array *v2hfs*. This is a one-to-many mapping. The manifold border vertices are stored in the array *b2hfs*. Meanwhile, it is useful to extract the geometric boundaries for numerical analysis. Therefore, the map *b2hfsm* is used to store all the manifold faces.

| Algorithm 2: Construction of topological relationship                                       |  |
|---|--|
| Input: element connectivity, CElement   |  |
| mutual mapping of twin half-faces, <i>m2hfs</i>   |  |
| <b>Output:</b> vertex to incident half-faces, <i>v2hfs</i>                                  |  |
| border vertex to incident half-faces, <i>b2hfs</i>  |  |
| border half-faces mapping, <i>bhfsm</i>   |  |
| begin   |  |
| Allocate array <i>marked</i> initialized to false and it is the same size as <i>m2hfs</i> ; |  |
| for each e in CElement do   |  |
| $f \leftarrow$ traverse each face in $e$ ;  |  |
| if not <i>m</i> 2 <i>h</i> fs (f) and not <i>marked</i> (f) then                            |  |
| set mapping vertices on face to <i>f</i> into <i>v2hfs</i> ;                                |  |
| end if  |  |
| end for   |  |
| for each <i>e</i> in <i>CElement</i> do   |  |
| $f \leftarrow \text{traverse each face in } e;$   |  |
| if not m2hfs(f) then  |  |
| set mapping <i>f</i> into <i>bhfsm</i> ;  |  |
| $v \leftarrow \text{traverse each vertex in } f;$   |  |
| $V \leftarrow$ set of elements incident on $v$ in $v2hfs$ ;                                 |  |
| if not V then   |  |
| append <i>f</i> to <i>b2hfs</i> ;   |  |
| end if  |  |
| end if  |  |
| end for   |  |
| end   |  |

3.4. Topology Information Query

The purpose of reconstructing topological relationships is to enable efficient querying and processing of mesh information. We summarize the basic queries as follows.

- 1. Adjacency query:
- Given an edge, return vertex-connected adjacent edges;
- Given a face, return edge-connected adjacent faces;
- Given an element, return face-connected adjacent elements;

Algorithm 3 describes adjacency query for the given element as an example. Figure 6 shows the example of the operation. The complete topological relationship supports the adjacent query time complexity of O(1).

| Algorithm 3: Adjacency query for shared sub-entity                |  |  |  |  |  |
|---|--|--|--|--|--|
| Input: element identification number, <i>elem</i>                 |  |  |  |  |  |
| mutual mapping of twin half-faces, <i>m2hfs</i>                   |  |  |  |  |  |
| Output: adjacent three-dimensional entities, <i>cadj</i>          |  |  |  |  |  |
| begin   |  |  |  |  |  |
| <b>for each</b> face <i>f</i> in <i>elem</i> <b>do</b>            |  |  |  |  |  |
| if <i>m2hfs</i> (f) then  |  |  |  |  |  |
| $e \leftarrow$ decode to obtain face-connected adjacent elements; |  |  |  |  |  |
| append <i>e</i> to <i>cadj;</i>                                   |  |  |  |  |  |
| end if  |  |  |  |  |  |
| end for   |  |  |  |  |  |
| end   |  |  |  |  |  |
|   |  |  |  |  |  |
| (a) (b)   |  |  |  |  |  |



**Figure 6.** An example of a 3-D element adjacency query. For (**a**) a given blue element, it shows (**b**) adjacent red elements.

- 2. Incidence query:
- Given a vertex, return incident elements;
- Given an edge, return incident elements;

As an example, Algorithm 4 for querying the incident elements of an edge is as follows. Figures 7 and 8 show examples of these operations.

| Algorithm 4: Upscaling incidence query                                 |  |  |  |  |
|--|--|--|--|--|
| <b>Input:</b> An edge defined by two vertices, <i>s</i>                |  |  |  |  |
| vertex to incident half-faces, v2hfs                                   |  |  |  |  |
| border vertex to incident half-faces, b2hfs                            |  |  |  |  |
| Output: incident three-dimensional entities, cinc                      |  |  |  |  |
| begin  |  |  |  |  |
| $v1 \leftarrow$ vertex with larger ID in <i>s</i> ;                    |  |  |  |  |
| $V1 \leftarrow$ set of elements incident on $v1$ found in $v2hfs(v)$ ; |  |  |  |  |
| if not V1 then   |  |  |  |  |
| $V1 \leftarrow$ set of elements incident on v1 found in $b2hfs(v)$ ;   |  |  |  |  |
| end if   |  |  |  |  |
| $v2 \leftarrow adjacent vertex to v1 in s;$                            |  |  |  |  |
| $V2 \leftarrow$ set of elements incident on $v2$ found in $v2hfs(v)$ ; |  |  |  |  |
| if not V2 then   |  |  |  |  |
| $V2 \leftarrow$ set of elements incident on $v2$ found in $b2hfs(v)$ ; |  |  |  |  |
| end if   |  |  |  |  |
| <i>cinc</i> —the intersection of V1 and V2;                            |  |  |  |  |
| end  |  |  |  |  |



Figure 7. An example of incidence queries (red) for a non-manifold vertex (blue).



Figure 8. The example of incidence queries (red) for a given non-manifold edge (blue).

The topology query results show that the proposed CAHF data structure is effective. It can achieve topology reconstruction and topology queries within a limited complexity. The CAHF exploits the implied topological relationships between geometries and greatly reduces the memory overhead. This data structure is a solution to the finite non-manifold characteristic of adjacent geometries sharing faces. Therefore, the LMC can be viewed as a component consisting of the volume mesh at the macroscopic scale. Each mesh corresponds to an equivalent material. The CAHF data structure constructs the complete topology relationships so that the boundary geometric information between different materials can be achieved easily.

#### 4. Microscopic Scale: Lattice Unit Cell Based on Implicit Representation

#### 4.1. Implicit Function of Truss-like Struts

At the microscopic scale, we propose an implicit representation of truss-like strut. In this work, the composition of the material at the microscopic scale can be considered as homogeneous. The CAD model based on implicit function greatly supports material evaluation. Specially, the shape is determined by the implicit function  $\Phi(X)$ . Given a geometric point  $X_i$ ,  $\Phi(X_i) \leq 0$  means that material is present at that point. When  $\Phi(X_i) > 0$ , it means that there is no material at that point.

As shown in Figure 9, the shape of the strut can be regarded as a composite of two tangent circles  $C_1$ ,  $C_2$  and a conic curve. Given two points as the center of the circle and radius, the shape is entirely determined by the conic curve. In the modeling coordinate system (Figure 9), the shape can be represented by a quadratic equation after normalization (i.e., b = 1):

$$ax^2 + by^2 + 2cx + d = 0 (2)$$

and it becomes in the matrix form:





Figure 9. Construction of variable truss-like strut.

The circles  $C_1$ ,  $C_2$  are denoted as:

$$C_1: (x+1)^2 + y^2 = (r-k)^2$$
(4)

$$C_2: (x-1)^2 + y^2 = (r+k)^2$$
(5)

The symbols are defined as,  $r = (r_1 + r_2)/2$  and  $k = (r_2 - r_1)/2$ . As the circles are tangent to the curve, we can calculate two parameters of the curve equation. Thus, we take  $C_1$  as an example from (2) and (4). Depending on Equation (6) obtained by the union, its discriminant will be zero. Therefore, we achieve:

$$ax^{2} + 2cx + d = (x+1)^{2} - (r-k)^{2}$$
(6)

$$d = \frac{1}{a-1} \left[ (c-1)^2 - (a-1)(r-k)^2 + (a-1) \right]$$
(7)

similar reasoning with circle  $C_2$  gives:

$$d = \frac{1}{a-1} \left[ (c+1)^2 - (a-1)(r+k)^2 + (a-1) \right]$$
(8)

We can solve *c* from (7) and (8), and achieve the value of *d* into either (7) or (8):

$$c = (a-1)kr \tag{9}$$

$$d = \left(\frac{a}{a-1}\right) + \left[(a-1)r^2 - 1\right]k^2 - r^2$$
(10)

The unknown constant *a* as a free input variable determines the unique strut type as shown in Table 1 [35]. However, it is not very intuitive for unit cell geometric design. Based on this, a new explicit geometric representation is proposed as shown in Figure 10. Specifically, we introduce a third circle  $C_3$  with center coordinates (0, 0) and an initial radius of *r*. Additionally, the offset parameter *t* is introduced as a substitute for *a*. The circle  $C_3$  with radius r + t is tangent to the curve. A positive offset parameter makes the strut elliptical, whereas a negative offset parameter makes the strut hyperbolic.

**Table 1.** Strut type defined by *a*.

| Value of <i>a</i>                       | Shape           |
|---|-----------------|
| $r^{2}/(r^{2}-1) < a < k^{2}/(k^{2}-1)$ | Hyperbola strut |
| $a = k^2 / (k^2 - 1)$                   | Cone strut      |
| $k^2/(k^2-1) < a < 0$                   | Hyperbola strut |
| a = 0                                   | Parabola strut  |
| 0 < a < (kr + k - r)/(kr + k - r - 1)   | Ellipse strut   |
|   |                 |



Figure 10. Profile curve controlled by offset parameter *t*.

By introducing the offset parameter, the conic curve is precisely defined. Therefore, we replace *a* with the offset parameter *t*:

$$a(t) = \frac{k^2 - t^2 - 2rt}{k^2 - t^2 - 2rt - 1}$$
(11)

As shown in Figure 11, the strut shapes controlled by the offset parameter t are discussed based on the given two circles. The maximum and minimum values of the offset parameter can be calculated as follows. When the tangency points are on the *x*-axis, the value of t will be the maximum. The maximum value of t can be calculated from (11).

$$t_{max} = -r + \sqrt{k^2 + (r - k)(r + 1)}$$
(12)



**Figure 11.** The strut shapes controlled by the offset parameter *t*.

The value of *t* will be the minimum when the conic curve becomes an inner common tangent about circles  $C_1$ ,  $C_2$ . The existence condition of the circle  $C_3$  also should be considered (i.e.,  $r + t_{min} > 0$ ). We achieve:

$$t_{min} = k - r \tag{13}$$

As analyzed above, we propose an implicit representation for the variable shape. The parameters  $r_1$ ,  $r_2$  and t are used to replace a, c, d in the equation. Moreover, it is more intuitive and flexible to control the shape of the strut.

## 4.2. Generation of Lattice Unit Cell

The implicit function precisely represents the shape profile of the strut. A unit cell can be viewed as consisting of several struts. Thus, the microscopic CAD shape-material model is represented by the following steps. Firstly, the profile curve is rotated around the *x*-axis (Figure 9) to generate a 3D solid in the local modeling coordinate system (LCS) from (14).

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & c \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ c & 0 & 0 & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$$
(14)

which can be simplified as:

$$(\mathbf{P}\mathbf{X}^{\mathrm{T}}=\mathbf{0}$$
(15)

Next, the model is scaled, rotated and translated in sequence from (16). These operations realize the transformation from the local coordinate system to the global coordinate system (GCS), as shown in Figure 12.

>

$$\mathbf{X} * \mathbf{TRS} * \mathbf{P} * \mathbf{S}^{\mathsf{T}} \mathbf{R}^{\mathsf{T}} \mathbf{T}^{\mathsf{T}} * \mathbf{X}^{\mathsf{T}} = 0$$
(16)

where **T** is the translation matrix, i.e.,  $\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$ . **R** is Euler rotation matrix, i.e.,  $\mathbf{R} = \begin{bmatrix} c_{\beta} \cdot c_{\theta} & -c_{\beta} \cdot s_{\theta} & s_{\beta} & 0 \\ s_{\alpha} \cdot s_{\beta} \cdot c_{\theta} + c_{\alpha} \cdot s_{\theta} & -s_{\alpha} \cdot s_{\beta} \cdot s_{\theta} + c_{\alpha} \cdot c_{\theta} & -s_{\alpha} \cdot c_{\beta} & 0 \\ -c_{\alpha} \cdot s_{\beta} \cdot c_{\theta} + s_{\alpha} \cdot s_{\theta} & c_{\alpha} \cdot s_{\beta} \cdot s_{\theta} + s_{\alpha} \cdot c_{\theta} & c_{\alpha} \cdot c_{\beta} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ , and the symbols are defined as,  $s_{\alpha} = \sin(\alpha)$ ,  $c_{\alpha} = \sqrt{1-s_{\alpha}^2}$ ,  $s_{\beta} = \sin(\beta)$ ,  $c_{\beta} = \sqrt{1-s_{\beta}^2}$ ,  $s_{\theta} = \sin(\theta)$ , and  $c_{\theta} = \sqrt{1 - s_{\theta}^2}$ . The angles  $\alpha, \beta, \theta$  correspond to the rotation angles of the *x*, *y* and *z* axis, respectively from the GCS to the LCS. **S** is a scaling matrix, i.e.,  $\mathbf{S} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \end{bmatrix}$ .



Figure 12. The illustration of strut expressed in LCS and GCS.

Finally, several struts form a unit cell. Figure 13 shows some unit cell models with different topologies. The topological relationships at the microscopic scale are also based on the conventions in Figure 4. Designers can introduce auxiliary nodes to define more topologically complex unit cells. Moreover, it is more convenient for material evaluation at the microscopic scale due to the implicit representation of geometry and topological conventions.



**Figure 13.** The unit cells of different shapes. (**a**) Body-centered cubic (BCC); (**b**) Simple cubic (SC); (**c**) Octahedron (OCT); (**d**) The BCC with vertical struts (BCCz); (**e**) Face-centered cubic (FCC); (**f**) The FCC with vertical struts and no struts in the horizontal plane (S-FCCz); (**g**) The union of BCC and SC (BCCzxy); (**h**) The union of S-FCC and BCC (S-FBCC); (**i**) Reinforced body-centered cubic (RBCC); (**j**) Octet-truss (OT).

## 5. Dual-Scale Shape-Material Model Representation

5.1. Data Structure of the DDSM

Following the framework of the dual-scale representation of model in Section 2, a new CAD data structure is proposed. For a lattice solid object, *S* is represented by four parts:

$$S = f(P, M, C, U) \tag{17}$$

• Vertices array *P*: The array records the node information of the mesh model at the macroscopic scale. It consists of node index and node coordinates.

$$P = \begin{bmatrix} N_1 & x_1 & y_1 & z_1 \\ N_2 & x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots & \vdots \\ N_n & x_n & y_n & z_n \end{bmatrix}$$
(18)

• Material library *M*: The material library *M* contains homogenized equivalent materials  $M_D$  and unit cell fabrication materials (i.e., microscopic materials)  $M_d$ . It consists of material index and material properties. Depending on the functional requirement, the material properties on each scale can be elastic modulus *E*, Poisson's ratio  $\nu$ , shear modulus *G* and thermal expansion coefficient  $\alpha$ , etc. Additional parameters can be attached to the columns of the array.

$$M_{D} = \begin{bmatrix} M_{1} & E_{1} & \nu_{1} & G_{1} & \alpha_{1} \\ \vdots & & & \\ M_{i} & E_{i} & \nu_{i} & G_{i} & \alpha_{i} \\ \vdots & & \\ M_{m} & E_{m} & \nu_{m} & G_{m} & \alpha_{m} \end{bmatrix}$$
(19)  
$$M_{d} = \begin{bmatrix} M'_{1} & E_{1} & \nu_{1} & G_{1} \\ \vdots & & \\ M'_{j} & E_{j} & \nu_{j} & G_{j} \\ \vdots & & \\ M'_{t} & E_{t} & \nu_{t} & G_{t} \end{bmatrix}$$
(20)

• The macroscopic model *C*: Each mesh has three attributes, the mesh identification number, eight primary nodes, and equivalent material properties. *M<sub>i</sub>* corresponds to equivalent material index from (19). The eight primary node storage order follows the convention in Figure 4.

$$C = \begin{bmatrix} C_1 & M_1 & N_1 & N_2 & \cdots & N_7 & N_8 \\ \vdots & & & & \\ C_i & M_i & N_1 & N_2 & \cdots & N_7 & N_8 \\ \vdots & & & & \\ C_p & M_m & N_1 & N_2 & \cdots & N_7 & N_8 \end{bmatrix}$$
(21)

• The microscopic model U: Each unit cell is composed of  $N_k$  struts and its fabrication materials.  $M'_j$  corresponds to fabrication material index from (20). The strut has two attributes, two nodes ID connecting the strut and strut geometric parameters. The geometry of the strut is determined by the three parameters  $r_1$ ,  $r_2$ , t. All the implicit functions of struts are expressed in LCS.

$$U_{j} = \begin{bmatrix} N_{1} & N_{2} & r_{1} & r_{2} & t \\ N_{2} & N_{2} & r_{1} & r_{2} & t \\ \vdots & & & \\ N_{k} & N_{2} & r_{1} & r_{2} & t \end{bmatrix}$$
(22)

$$U = \begin{bmatrix} M'_1 & \cdots & M'_j & \cdots & M'_t \\ U_1 & \cdots & U_j & \cdots & U_m \end{bmatrix}^T$$
(23)

#### 5.2. Prototype Implementation

In this section, the three benefits of the DDSM model will be discussed over the STL format. Firstly, the DDSM contains dual-material information in addition to geometric information. The STL file format represent only the geometric information of the model. Therefore, it is difficult to complete material evaluation through geometric information. In the dual-scale framework, as shown in Figure 14, the lattice structure can be viewed as volume meshes at the macroscopic scale. The CAHF data structure constructs the CAD model with complete topological relationships, which helps the equivalent material query.

Moreover, the equivalent material property of unit cell takes the CAD model down from the macroscopic scale to the microscopic scale. The microscopic CAD model includes information on the geometry and manufacturing materials of the unit cell. The implicit representation can answer the question of whether the manufacturing material exists at the geometric points.



**Figure 14.** A  $30 \times 30 \times 30$  mm<sup>3</sup> cubic domain consisting of octahedral unit cells.

Secondly, in terms of memory overhead, the DDSM offers complete model information with a small storage overhead. The lattice structures are usually stored in STL format. However, the STL requires to record a large amount of facet geometric information due to Delaunay triangulation. The DDSM records model geometric and material information through two scales. As shown in Figure 14, the result show that the DDSM proposed requires less than 910 KB storage space, whereas coarse STL needs 1.38 GB storage space.

Thirdly, the DDSM reflects the user's design intent, which makes the CAD model revisable easily. Figure 15 shows the ideal design process for the lattice structure represented by the DDSM. Specially, the middle part of the connecting rod has been designed as a lattice structure for the lightweight design. The designer can make equivalent material design to determine macroscopic CAD model, which is represented by the CAHF. The unit cell configurations are obtained from equivalent material properties in *U*. Therefore, the geometric model of the product is thus quickly customized by the DDSM as shown in  $S_1$ . When the unit cell type needs to be modified in certain areas, the designers can simply modify the material distribution of target areas to obtain a new CAD model  $S_2$ . Compared to the DDSM format, the STL file with only facet information is powerless in modifying model. Meanwhile, it is hard and time-consuming to model the geometric again from  $S_1$  to  $S_2$  by the current CAD tools.



Figure 15. The connecting rod CAD model represented by the DDSM.

# 6. Conclusions

This paper presents a new CAD model (DDSM) for lattice representation. The key ingredients of the CAD model are complete shape-material representation on each scale and mapping linking the two scales. Compared to the conventional CAD file format, the DDSM demonstrates advantages in terms of material information representation, memory overhead and the capture of design intent. For different functional requirements, the DDSM embodies trans-scale parallel design. Unit cell configurations at the microscopic scale determine the properties of the equivalent material. Equivalent materials reflect the geometric design intent at the macroscopic scale.

Non-manifold geometry and implicit functions are used to construct the DDSM. The proposed CAHF verifies that it is complete and unambiguous to construct macroscopic CAD models. At the microscopic scale, the LMC is regarded as a collection of unit cells. A new implicit representation model determines the geometry and material distribution. According to the findings of this paper, the LMC represented by the DDSM meets the modern product design needs better than the conventional CAD format. Future work should include CAD model representation of unit cell incompatible connections and slicing information extraction of the DDSM for additive manufacturing.

**Author Contributions:** Conceptualization, R.Y. and C.F.; methodology, C.F.; software, C.F.; formal analysis, C.F.; investigation, C.F.; data curation, X.M.; writing—original draft preparation, C.F.; writing—review and editing, R.Y. and B.N.; visualization, X.M.; supervision, R.Y. and B.N.; funding acquisition, R.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is financially supported by the National Natural Science Foundation of China (Grant No.51975087).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declared no potential conflict of interest.

#### References

- Bai, L.; Yi, C.; Chen, X.; Sun, Y.; Zhang, J. Effective design of the graded strut of bcc lattice structure for improving mechanical properties. *Materials* 2019, 12, 2192. [CrossRef] [PubMed]
- 2. Zhao, M.; Liu, F.; Fu, G.; Zhang, D.Z.; Zhang, T.; Zhou, H. Improved mechanical properties and energy absorption of bcc lattice structures with triply periodic minimal surfaces fabricated by slm. *Materials* **2018**, *11*, 2411. [CrossRef] [PubMed]
- Liu, Y.; Zhuo, S.; Xiao, Y.; Zheng, G.; Dong, G.; Zhao, Y.F. Rapid modeling and design optimization of multi-topology lattice structure based on unit-cell library. J. Mech. Des. 2020, 142, 091705. [CrossRef]
- Liu, Y.; Zheng, G.; Letov, N.; Zhao, Y. A survey of modeling and optimization methods for multi-scale heterogeneous lattice structures. J. Mech. Des. 2020, 143, 040803. [CrossRef]
- Azman, A.H.; Vignat, F.; Villeneuve, F. Cad tools and file format performance evaluation in designing lattice structures for additive manufacturing. J. Teknol. 2018, 80, 87–95. [CrossRef]
- Kladovasilakis, N.; Tsongas, K.; Karalekas, D.; Tzetzis, D. Architected materials for additive manufacturing: A comprehensive review. *Materials* 2022, 15, 5919. [CrossRef]
- 7. Plocher, J.; Panesar, A. Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures. *Mater. Des.* **2019**, *183*, 108164. [CrossRef]
- 8. Savio, G.; Meneghello, R.; Concheri, G. Geometric modeling of lattice structures for additive manufacturing. *Rapid Prototyp. J.* **2018**, 24, 351–360. [CrossRef]
- Aremu, A.O.; Brennan-Craddock, J.P.J.; Panesar, A.; Ashcroft, I.A.; Hague, R.J.M.; Wildman, R.D.; Tuck, C. A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturingt. *Addit. Manuf.* 2017, 13, 1–13.
- Mustafa, S.S.; Lazoglu, I. A new model and direct slicer for lattice structures. *Struct. Multidiscip. Optim.* 2021, 63, 2211–2230. [CrossRef]
- 11. Liu, H. Algorithms for Design and Interrogation of Functionally Graded Material Solids. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000.
- 12. Cirillo, E.; Elber, G. Handling heterogeneous structures and materials using blending schemes in v-reps. *Comput. Aided Geom. Des.* **2020**, *83*, 101942. [CrossRef]

- 13. Hong, Q.Y.; Elber, G. Conformal microstructure synthesis in trimmed trivariate based v-reps. *Comput. Aided Des.* **2021**, *140*, 11. [CrossRef]
- 14. Mandad, M.; Chen, R.; Bommes, D.; Campen, M. Intrinsic mixed-integer polycubes for hexahedral meshing. *Comput. Aided Geom. Des.* **2022**, *94*, 102078. [CrossRef]
- 15. Zhang, Z.Y.; Chen, D.F.; Hu, J.Q.; Wang, X.P.; Chen, P. Representation and fabrication method for multiple gradients fgm part based on additive manufacturing. *Appl. Mech. Mater.* **2013**, *433-435*, 2076–2080. [CrossRef]
- 16. Tereshin, A.; Pasko, A.; Fryazinov, O.; Adzhiev, V. Hybrid function representation for heterogeneous objects. *Graph. Models* **2021**, *114*, 101098. [CrossRef]
- 17. Zang, T.; Zhu, D.; Mu, G. Inverse construction methods of heterogeneous nurbs object based on additive manufacturing. *Comput. Modeling Eng. Sci.* 2020, 125, 597–610. [CrossRef]
- 18. Li, B.; Fu, J. Solid modeling and slicing process of heterogeneous materials based on trivariate t-splines. *J. Zhejiang Univ. (Eng. Sci.)* **2021**, *55*, 10.
- 19. Kambampati, S.; Jauregui, C.; Museth, K.; Kim, H.A. Geometry design using function representation on a sparse hierarchical data structure. *Comput.-Aided Des.* **2021**, *133*, 102989. [CrossRef]
- Kou, X.Y.; Tan, S.T.; Sze, W.S. Modeling complex heterogeneous objects with non-manifold heterogeneous cells. *Comput. Aided Des.* 2006, 38, 457–474. [CrossRef]
- Qian, X.; Dutta, D. Heterogeneous object modeling through direct face neighborhood alteration. *Comput. Graph.* 2003, 27, 943–961. [CrossRef]
- 22. Mueller-Karger, C.M.; Rank, E.; Cerrolaza, M. P-version of the finite-element method for highly heterogeneous simulation of human bone. *Finite Elem. Anal. Des.* **2004**, *40*, 757–770. [CrossRef]
- 23. Opgenoord, M.M.J.; Willcox, K.E. Design for additive manufacturing: Cellular structures in early-stage aerospace design. *Struct. Multidiscip. Optim.* **2019**, *60*, 411–428. [CrossRef]
- 24. Schumacher, C.; Bickel, B.; Rys, J.; Marschner, S.; Daraio, C.; Gross, M. Microstructures to control elasticity in 3d printing. *ACM Trans. Graph.* **2015**, *34*, 136. [CrossRef]
- Deng, H.; Zhao, J.; Wang, C. Bionic design method of a non-uniform lattice structure for a landing footpad. *Aerospace* 2022, 9, 220. [CrossRef]
- 26. Li, F.; Zhang, Q.; Shi, H.; Liu, Z. A modified three-dimensional negative-poisson-ratio metal metamaterial lattice structure. *Materials* **2022**, *15*, 3752. [CrossRef]
- Liu, P.; Liu, A.; Peng, H.; Tian, L.; Liu, J.; Lu, L. Mechanical property profiles of microstructures via asymptotic homogenization. *Comput. Graph.* 2021, 100, 106–115. [CrossRef]
- 28. Liu, X.; Shapiro, V. Multiscale shape-material modeling by composition. Comput.-Aided Des. 2018, 102, 194–203. [CrossRef]
- 29. Xia, H.; Meng, J.; Liu, J.; Ao, X.; Lin, S.; Yang, Y. Evaluation of the equivalent mechanical properties of lattice structures based on the finite element method. *Materials* **2022**, *15*, 2993. [CrossRef]
- 30. Ellul, C. Functionality and Performance: Two Important Considerations when Implementing Topology in 3D; University of London: London, UK, 2008.
- 31. Boguslawski, P. Modelling and Analysing 3D Building Interiors with the Dual Half-Edge Data Structure; University of South Wales: New Port, UK, 2011.
- Canino, D.; De Floriani, L.; Weiss, K. Ia\*: An adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions. *Comput. Graph.* 2011, 35, 747–753. [CrossRef]
- Alumbaugh, T.J.; Jiao, X. Compact Array-Based Mesh Data Structures. In *Proceedings of the 14th International Meshing Roundtable*; Hanks, B.W., Ed.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 485–503.
- 34. Dyedov, V.; Ray, N.; Einstein, D.; Jiao, X.; Tautges, T.J. Ahf: Array-based half-facet data structure for mixed-dimensional and non-manifold meshes. *Eng. Comput.* 2015, *31*, 389–404. [CrossRef]
- Gupta, A.; Allen, G.; Rossignac, J. Quador: Quadric-of-revolution beams for lattices. *Comput.-Aided Des.* 2018, 102, 160–170. [CrossRef]