# Symmetric Implicit Surface Microstructures for Hierarchical Design

RUIQI CHEN, HARDIK KABARIA, and ADRIAN J. LEW, Stanford University, USA

We present a framework for generating hierarchical structures using a palette of symmetric implicit surface microstructures. The framework utilizes an unstructured mesh composed solely of triangles, quadrilaterals, tetrahedrons, or hexahedrons to serve as a scaffold for microstructural unit cells to tile, where each unit cell is represented by an implicit surface and defined using a set of symmetric basis functions derived from Fourier series. Any arbitrary input geometry may be converted into a symmetric implicit surface using an Fast-Fourier-Transform-based method as long as inside/outside queries can be performed on the input geometry. Using our framework, we demonstrate blending together different unit cells to create new ones as well as spatial transitions between different unit cells. In addition, we present a method to constrain unit cells to intersect its bounding polygon/polyhedron at a right angle, which we postulate is necessary for $C^1$ continuity. Finally, we briefly mention some mechanical applications that may benefit from our framework.

CCS Concepts: • **Applied computing** → **Computer-aided design**; Aerospace; • **Computing methodologies** → **Shape modeling**.

Additional Key Words and Phrases: hierarchical structure, implicit surface, lattice, microstructure, symmetry

**ACM Reference Format:**
Ruiqi Chen, Hardik Kabaria, and Adrian J. Lew. 2025. Symmetric Implicit Surface Microstructures for Hierarchical Design. *ACM Trans. Graph.* 37, 4, Article 111 (August 2025), 4 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

### 1.1 Contributions

To the best of our knowledge, this work is the first to present a framework for generating hierarchical structures using *unstructured* scaffold meshes composed solely of one element type: triangles, quadrilaterals, tetrahedrons, and hexahedrons. Furthermore, this work is the first to present a method of transitioning between arbitrarily chosen unit cells within the unstructured scaffold mesh. By supporting the most common unstructured mesh types, we ensure that our framework is compatible with a wide range of off-the-shelf mesh generators and can be used to generate hierarchical structures from complex design domains that are practically unsuitable for structured meshing.

In short, this paper makes the following contributions:

- We develop a novel method to constrain the unit cell to intersect its bounding polygon/polyhedron at a right angle by formulating a constrained convex optimization problem that is always feasible.

Authors' Contact Information: Ruiqi Chen, rchensix@alumni.stanford.edu; Hardik Kabaria, hardikk@alumni.stanford.edu; Adrian J. Lew, lewa@stanford.edu, Stanford University, Stanford, California, USA.


ACM 1557-7368/2025/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

## 2 Related Works

## 3 Methodology

### 3.1 Symmetric Implicit Surfaces

Implicit surfaces represent a geometric object through an equation of the form $\Phi(\mathbf{x}) = 0$ where $\Phi : \mathbb{R}^d \to \mathbb{R}$ is some function that evaluates a query point $\mathbf{x}$ and returns zero if $\mathbf{x}$ is on the surface, returns a negative value if $\mathbf{x}$ is inside the surface, or returns a positive value if $\mathbf{x}$ is outside the surface (note that this convention is not universal and other authors may use the opposite convention, i.e. points inside the surface have positive return values). The magnitude of the return value of the implicit surface function may have some geometric meaning, but this is not required. For example, signed distance functions are a subset of implicit surface equations where the magnitude of the return value denotes the distance from the query point $\mathbf{x}$ to the surface. In this work, we are interested in implicit surface functions that represent 2D and 3D geometries, so we assume that $d = 2$ or $d = 3$.

A symmetric implicit surface is defined in this work to be an implicit surface equation that has been constrained such that the geometry that it represents is unchanged after applying any transformation from a chosen symmetry group $\mathcal{S}$. In other words, $\Phi(\mathbf{x}) = \Phi(\mathbf{Q}\mathbf{x})$, where $\mathbf{Q} : \mathbb{R}^{d\times d} \in \mathcal{S}$ is an orthogonal matrix that represents a linear transformation of a point $\mathbf{x}$ about the origin. Note that for this work, we are restricting ourselves to linear transformations and excluding affine transformations. We want to constrain the implicit surface to match the symmetry group of the cell type in the unstructured scaffold mesh to tile: dihedral group of degree 3 and order 6 ($D_3$) for triangle meshes, dihedral group of degree 4 and order 8 ($D_4$) for quad meshes, achiral tetrahedral symmetry ($T_d$) for tetrahedron meshes, and achiral octahedral symmetry ($O_h$) for hexahedron meshes. The list of symmetry operators for these four symmetry groups is given in Appendix A. The necessity of symmetric implicit surfaces to tile the unstructured scaffold mesh is explained in detail in Section 3.4, but in brief, it is the symmetry that ensures $C^0$ continuity at the boundaries of the hierarchical structure. Furthermore, symmetry is crucial for enabling the spatial transitions described in Section 3.7.

To construct a symmetric implicit surface $\Phi(\mathbf{x})$, we first construct a basis function that is inherently symmetric. The symmetric basis function is constructed from some other conventional basis function $\psi(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ by iterating over the symmetry operators $\mathbf{Q} \in \mathcal{S}$, feeding the transformed point $\mathbf{Q}\mathbf{x}$ as input to $\psi$, and summing over all values of $\mathbf{Q}$:

$$\phi(\mathbf{x}) = K \sum_{\mathbf{Q}\in\mathcal{S}} \psi(\mathbf{Q}\mathbf{x}) \tag{1}$$

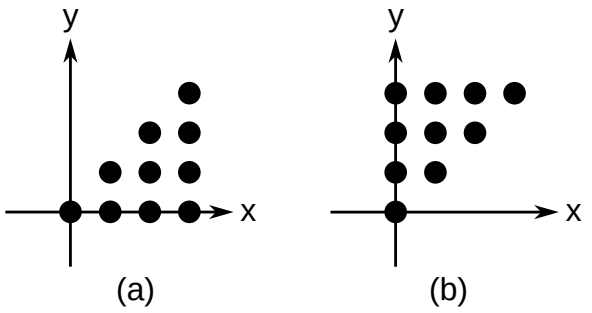


Fig. 1. A subset $\mathcal{T}$ of the $\mathbb{Z}^2$ lattice defines unique spatial frequencies that can be used to generate unit cells with $D_4$ symmetry. Both (a) and (b) are valid representations for $\mathcal{T}$, and it is a matter of user preference to choose.

where $K$ is some normalization factor to make the basis function orthonormal. In this work, the $d$-dimensional Fourier series

$$\psi_{\mathbf{f}}(\mathbf{x}) = e^{2\pi i \mathbf{f}^T \mathbf{x}} \tag{2}$$

is used, where $\mathbf{f} : \mathbb{R}^d \in \mathcal{T}$ is an integer-valued vector that represents spatial frequencies in the Cartesian directions, and $\mathcal{T} \subset \mathbb{Z}^d$ is a subset of the $d$-dimensional integer lattice that is specific to the chosen scaffold mesh type. In other words, $\mathcal{T}$ contains all of the *unique* spatial frequencies that may be used to construct a symmetric basis functions. Any spatial frequencies not in $\mathcal{T}$ do not need to be considered due to symmetry. For example, suppose the scaffold mesh is a quad mesh, then one representation for $\mathcal{T}$ is given by

$$\mathcal{T} = \left\{ \begin{bmatrix} f_x \\ f_y \end{bmatrix} \in \mathbb{Z}^2 | f_y \geq 0 \text{ and } f_x - f_y \geq 0 \right\} \tag{3}$$

This representation of $\mathcal{T}$ for a quad mesh is depicted in Figure 1(a). Note that if all of the symmetry operators in $S$ are applied to $\mathcal{T}$, then we would recover $\mathbb{Z}^2$. Also note that the representation of $\mathcal{T}$ is not unique; Figure 1(b) shows an alternative representation that is equally valid. Representations of $\mathcal{T}$ used in this work for triangles, quadrilaterals, tetrahedrons, and hexahedrons are given in Appendix A. Additionally, note that by writing the Fourier series using complex exponentials (as expressed in Equation 2), Equation 1 is, in general, a function that takes in real-valued $d$-D points and returns complex valued scalars, i.e. $\phi_{\mathbf{f}}(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{C}$. However, for the purpose of generating implicit surfaces, only the real or imaginary part (but not both) of the final expression for $\Phi(\mathbf{x})$ will be used.

There are many reasons, both mathematical and practical, as to why the Fourier series is an excellent choice for the foundational basis function, but the primary reasons are: Fourier series are an orthonormal basis, Fourier series have a very simple expression for the gradient (this is used in Section 3.3); there are fast, robust algorithms available (most notably, the fast Fourier transform) for computing the cofficients of a Fourier expansion; and finally, several famous triply periodic minimal surfaces (TPMS) that are commonly used to construct hierarchical lattice structures (e.g. Schwarz primitive, gyroid) may be approximated using Fourier series [Wohlgemuth et al. 2001].

Substituting Equation 2 into Equation 1, the symmetric basis function may be expressed as

$$\phi_{\mathbf{f}}(\mathbf{x}) = K_{\mathbf{f}} \sum_{\mathbf{Q} \in \mathcal{S}} e^{2\pi i \mathbf{f}^T \mathbf{Q} \mathbf{x}} \tag{4}$$

where $K_{\mathbf{f}}$ is the normalization coefficient given by

$$K_{\mathbf{f}} = \frac{1}{\sqrt{\iiint_{\Omega} \left( \sum_{\mathbf{Q}_1, \mathbf{Q}_2 \in \mathcal{S}} e^{2\pi i \mathbf{f}^T (\mathbf{Q}_1 - \mathbf{Q}_2) \mathbf{x}} \right) dV}} \tag{5}$$

where the region of integration $\Omega$ is either the unit square (with side length of 1) centered at the origin for $d = 2$ or the unit cube (with side length of 1) centered at the origin for $d = 3$. In the integrand in the denominator of Equation 5, due to the orthonormal property of Fourier series, only terms where $\mathbf{f}^T(\mathbf{Q_1} - \mathbf{Q_2}) = \mathbf{0}$ contribute to the integral.

Just like the Fourier series, the resulting basis function in Equation 4 is orthonormal, i.e.

$$\begin{aligned} \iiint_{\Omega} \phi_{\mathbf{f}}(\mathbf{x}) \overline{\phi_{\mathbf{g}}(\mathbf{x})} dV &= 0 \\ \iiint_{\Omega} \phi_{\mathbf{f}}(\mathbf{x}) \overline{\phi_{\mathbf{f}}(\mathbf{x})} dV &= 1 \end{aligned} \tag{6}$$

for some arbitrary $\mathbf{f} \in \mathcal{T}$, $\mathbf{g} \in \mathcal{T}$, and $\mathbf{f} \neq \mathbf{g}$. The implicit surface $\Phi(\mathbf{x})$ may now be expressed in terms of a sum of the basis functions:

$$\Phi(\mathbf{x}) = \sum_{\mathbf{f} \in \mathcal{T}} c_{\mathbf{f}} \phi_{\mathbf{f}}(\mathbf{x}) \tag{7}$$

where $c_{\mathbf{f}}$ are the weight coefficients for each basis function.

Given an arbitrary input implicit surface $g(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$, which may or may not possess tetrahedral symmetry, a tetrahedrally symmetric approximation to $g(\mathbf{x})$ may be computed by projecting $g(\mathbf{x})$ onto the basis functions to recover the weight coefficients:

$$c_{\mathbf{f}} = \iiint_{\Omega} g(\mathbf{x}) \phi_{\mathbf{f}}(\mathbf{x}) dV \tag{8}$$

In this case, because the input function $g(\mathbf{x})$ is real-valued, the output function $\Phi(\mathbf{x})$ will also be real-valued. In practice, Equation 8 will be evaluated numerically, and the input geometry to approximate does not have to be an implicit surface equation, as demonstrated in the following section.

### 3.2 Converting Geometry Representations to Implicit Surfaces

Frequently, a user will choose a unit cell geometry based on some criteria (e.g. aesthetics, porosity, mechanical performance, etc.) to create the hierarchical structure. The input unit cell to tile might not be an implicit surface; furthermore, it might not have *any* symmetry either. To accomodate nearly arbitrary incoming geometry representations (e.g. triangle meshes, NURBS, voxel grids), the only requirement is that the input representation has a method to determine whether a query point is inside or outside the geometry. For example, inside/outside queries on oriented point clouds and triangle meshes/soups can be computed extremely robustly using the fast winding number method [Barill et al. 2018].

To numerically compute the implicit surface approximation $\Phi(\mathbf{x})$, first, the input geometry should be centered at the origin and be fully

contained by the unit square (for $d = 2$) or the unit cube (for $d = 3$). The input geometry is then sampled by querying the inside/outside status on an $N \times N \times N$ grid of query points. The query points should equispaced sampled on the domain $[-0.5, 0.5)$ in every Cartesian direction, meaning the spacing between adjacent points along the Cartesian directions is $\frac{1}{N}$. Inside points are assigned a scalar value of $-1$ while outside points are assigned a scalar value of $+1$. Next, the fast Fourier transform (FFT) is applied to the inside/outside samples. Finally, the weight coefficients $c_{\mathbf{f}}$ may be computed from the FFT result by noting that the right hand side of Equation 8 is simply a summation of FFT coefficients due to the fact that $\phi_{\mathbf{f}}(\mathbf{x})$ is a summation of 3D complex exponentials.

It is worth noting that if the input geometry is band limited (i.e. the input geometry is generated from a 3D Fourier series implicit equation with a finite set of spatial frequencies), then using the FFT to compute the weight coefficients $c_{\mathbf{f}}$ will be exact up to limitations in floating point computations, as long as $N$ is large enough to satisfy the Nyquist-Shannon sampling theorem, i.e. $h < \frac{N}{2}$ where $h \in \mathbb{Z}$ is the maximum magnitude integer frequency in any Cartesian direction of the input geometry. However, if the input geometry is not band limited (e.g. there are sharp edges and corners), then this method will only yield approximate values of $c_{\mathbf{f}}$ due to aliasing. In practice, given some arbitrary input geometry that may or may not be band limited, one can choose some reasonable starting value for $N$ such that the FFT can compute relatively quickly (for example $N = 2^7$), compute the implicit surface approximation to the input, and use some stopping criteria to determine if $N$ is large enough. One example of stopping criteria is the Hausdorff distance between the input geometry and output approximation. Furthermore, choosing powers of two for values of $N$ is practical due to exploiting efficiencies in the FFT algorithm. The topic of convergence of the Fourier series is well studied in literature. Since the main purpose of this paper is to develop methods to generate hierarchical structures, i.e. the main purpose is *not* about the convergence of Fourier series, most examples in this paper will use some arbitrarily chosen value for $N$.

The preceding section utilized inside/outside queries of the input geometry to develop the implicit surface approximation. More useful results can be achieved if the signed distance of the input representation can be computed at query points. This enables finding a symmetric implicit equation approximation to the signed distance function (SDF) of the input geometry. For example, once the SDF approximation is computed, it is trivial to compute approximate dilations and erosions (see Section 3.5). Using the sampled SDF as input will also yield a smoother approximation since the inputs are no longer binary. Because of this, most examples in this paper use the sampled SDF to compute the implicit surface approximation.

## 3.3 Face Normal Surfaces

In addition to symmetry constraints, the implicit surface unit cell can be further constrained to intersect with its bounding polygon/polyhedron at a right angle. We postulate this is an important property to have in order to generate hierarchical structures with $C^1$ continuity in higher order meshes with $C^1$ continuity constraints applied to control points (see Section 3.4.1). To apply this face normal constraint, we first note that the outward normal vector of an implicit surface is given by its gradient:

$$\mathbf{N}(\mathbf{x}) = \nabla\Phi(\mathbf{x}) \tag{9}$$

To constrain the implicit surface to intersect its bounding polygon/polyhedron at a right angle, we need its outward normal to be at a right angle to the outward normal $\mathbf{N}_p$ of the polygon/polyhedron face. Thus, the dot product between these two vectors must be zero when evaluated on the polygon/polyhedron faces:

$$\nabla\Phi(\mathbf{x}) \cdot \mathbf{N}_p | \mathbf{x} \to \mathbf{x}_p = 0 \tag{10}$$

where $\mathbf{x}_p$ is a point on the bounding polygon/polyhedron face. Note that due to the symmetry constraint already imposed from our usage of symmetric basis functions, we only need to satisfy Equation 10 on one face of the bounding polygon/polyhedron, as it will be automatically applied to the other faces. We can rewrite $\mathbf{x}_p$ as

$$\mathbf{x}_p = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{11}$$

where $\mathbf{A} : \mathbb{R}^{d \times d}$ is a projection matrix and $\mathbf{b} : \mathbb{R}^d$ is an offset term. Equation 11 can be geometrically interpreted as projecting a point $\mathbf{x}$ onto the extended plane of the chosen face of the bounding polygon/polyhedron. Using Equation 4 and 7, the gradient term can be written as:

$$\begin{aligned} \nabla\Phi(\mathbf{x}) &= \sum_{\mathbf{f} \in \mathcal{T}} c_{\mathbf{f}} \nabla\phi_{\mathbf{f}}(\mathbf{x}) \\ &= 2\pi i \sum_{\mathbf{f} \in \mathcal{T}} c_{\mathbf{f}} K_{\mathbf{f}} \sum_{\mathbf{Q} \in \mathcal{S}} \mathbf{Q}^T \mathbf{f} e^{2\pi i \mathbf{f}^T \mathbf{Q}\mathbf{x}} \end{aligned} \tag{12}$$

## 3.4 Tiling Mesh Scaffolds

### 3.4.1 *Higher Order Mesh Scaffolds.*

## 3.5 Dilation, Erosion, and Inversion

Dilating, eroding, and inverting the implicit surface unit cell is trivial.

## 3.6 Hybridization

### 3.6.1 *Weighted Averaging.*

### 3.6.2 *Minkowski Sum and Difference.*

## 3.7 Spatial Transitions

## 3.8 Generalizing to Other Symmetry Groups

# 4 Discussion

## 4.1 Limitations

TODO continuity of implicit surface equations, enforcing continuity, etc.

TODO continuity of hybridization and transitions, how to pick transition path to enforce continuity, etc.

## 4.2 Future Work

TODO optimizing weight coefficients to achieve some target, i.e. porosity, stiffness, etc.

## 5 Conclusion

A publically available implementation of this work is available at https://github.com/rchensix/tetrahedral_implicit_lattice under an MIT License. However, please note that while the implementation itself falls under an MIT License, methods and ideas mentioned in this paper are, at the time of writing, under consideration for a patent [Chen and Kabaria 2022].

## Acknowledgments

TODO

## References

Mois I. Aroyo (Ed.). 2016. *International Tables for Crystallography Volume A: Space-group symmetry.* John Wiley & Sons, Inc., New York, NY, USA. doi:10.1107/9780955360206000114

Gavin Barill, Neil G. Dickson, Ryan Schmidt, David I. W. Levin, and Alec Jacobson. 2018. Fast winding numbers for soups and clouds. *ACM Transactions on Graphics* 37, 4 (July 2018), 1–12. doi:10.1145/3197517.3201337

Ruiqi Chen and Hardik Kabaria. 2022. Systems and methods for constructing lattice objects for additive manufacturing. https://patents.google.com/patent/WO2022212472A1 Patent Application No. WO2022212472A1, Filed March 30, 2022, Published October 6, 2022.

Meinhard Wohlgemuth, Nataliya Yufa, James Hoffman, and Edwin L. Thomas. 2001. Triply periodic bicontinuous cubic microdomain morphologies by symmetries. *Macromolecules* 34, 17 (July 2001), 6083–6089. doi:10.1021/ma0019499

## A Symmetry Groups

All symmetry operators here are written in crystallographic notation, composed of a $d$-tuple representing the result of applying a linear transformation to a point $\mathbf{x} \in \mathbb{R}^d$ (affine symmetry operators are not used in this work). An overbar denotes negation. For example, $x, \overline{z}, y$ represents the linear transformation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ -z \\ y \end{bmatrix} \tag{13}$$

An authoritative source enumerating all space groups is the International Tables of Crystallography [Aroyo 2016].

### A.1 Dihedral Group of Degree 3 and Order 6

The dihedral group of degree 3 and order 6 ($D_3$) is the symmetry group of an equilateral triangle.

### A.2 Dihedral Group of Degree 4 and Order 8

The dihedral group of degree 4 and order 8 ($D_4$) is the symmetry group of a square.

### A.3 Achiral Tetrahedral Symmetry

Achiral tetrahedral symmetry ($T_d$) is the symmetry group of a regular tetrahedron.

$$\mathcal{T} = \left\{ \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \in \mathbb{Z}^3 | f_y + f_z \geq 0 \text{ and} f_y - f_z \geq 0 \text{ and} f_x - f_y \geq 0 \right\} \tag{14}$$

### A.4 Achiral Octahedral Symmetry

Achiral tetrahedral symmetry ($O_h$) is the symmetry group of a cube (as well as that of a regular octahedron, hence the name). There are 48 symmetry operators.

$$\begin{matrix} x, y, z \\ x, \overline{y}, \overline{z} \end{matrix} \tag{15}$$

Received DD MONTH YYYY; revised DD MONTH YYYY; accepted DD MONTH YYYY