Design and Optimization of Conforming Lattice Structures

Wu, Jun; Wang, Weiming; Gao, Xifeng

DOI
10.1109/TVCG.2019.2938946

Publication date
2021

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Visualization and Computer Graphics

Design and Optimization of Conforming Lattice Structures

Jun Wu, Weiming Wang, Xifeng Gao

Abstract—Inspired by natural cellular materials such as trabecular bone, lattice structures have been developed as a new type of lightweight material. In this paper we present a novel method to design lattice structures that conform with both the principal stress directions and the boundary of the optimized shape. Our method consists of two major steps: the first optimizes concurrently the shape (including its topology) and the distribution of orthotropic lattice materials inside the shape to maximize stiffness under application-specific external loads; the second takes the optimized configuration (i.e. locally-defined orientation, porosity, and anisotropy) of lattice materials from the previous step, and extracts a globally consistent lattice structure by field-aligned parameterization. Our approach is robust and works for both 2D planar and 3D volumetric domains. Numerical results and physical verifications demonstrate remarkable structural properties of conforming lattice structures generated by our method.

Index Terms—Lattice structures, topology optimization, homogenization, 3D printing.

1 Introduction

The design of lightweight structures by optimization is a classical and still active topic in engineering. Stimulated by the increasingly high flexibility and resolution offered by 3D printing, there has been a growing interest in optimizing structures that are composed of delicate microstructures [1], [2]. These approaches assume that the microstructures are aligned with a prescribed regular grid. This simplifies modelling, simulation and optimization. It, however, limits the solution space, and thus the achievable structural performance. The microstructures are typically anisotropic (e.g. a hollowed cubic cell with uniform thickness is stiffer along its axes than along its diagonals). It is known that the orientation of anisotropic materials in stiffness-optimal structure coincides with the principal stress directions resulting from forces acting on these materials [3]. Furthermore, axis-aligned microstructures do not match the curved surfaces of 3D shapes, which may create problems in assembly of mechanical components.

To address the above issues, in this paper we propose an efficient and robust method to generate *conforming* lattice structures. A *lattice* is a connected array of struts. The lattice structure generated by our method is conforming in two aspects: the struts align with principal stress directions, maximizing structural stiffness; and, struts on the boundary capture the curved surface of the optimized shape. We note that the shape, according to design options accessible to the user, is allowed to evolve together with the optimization of lattice distribution, i.e. the optimized shape is a subset of the design domain.

Our method has two major steps, in line with the homogenization-based optimization method proposed by Bendsøe and Kikuchi [4] and the post-treatment of the homogenization proposed by Pantz and Trabelsi [5] which was recently revisited [6], [7]. In the first step of our method, both the shape and the spatially-varying orientation of lattices inside the shape evolve simultaneously according to stress analysis and numerical optimization. Rather than relying on extremely high-resolution finite elements to capture the evolving lattice geometric details, we develop a homogenization-based topology optimization method which allows to efficiently simulate and optimize the lattice material distribution on a relatively coarse level. By introducing a novel parameterization of the unit cell, our method ensures a uniform thickness of struts while allowing a sufficient degree of lattice anisotropy. The second step, which we call *lattice compilation*, extracts a globally consistent lattice structure from the optimized, locally-defined lattice configuration, including orientation, porosity, and anisotropy. We address the challenging problem of extracting connected lattices across cells with spatially-varying orientation, by extending field-aligned meshing techniques. This extension allows a fast and robust lattice compilation where anisotropic geometric features are incorporated.

The specific contributions of our paper include:

- A novel workflow for designing, in both 2D and 3D, conforming lattice structures based on homogenization-based topology optimization and field-aligned parameterization.
- A simple and effective parameterization of the unit cell for allowing structural anisotropy while ensuring a uniform thickness of struts.

- *Jun Wu and Weiming Wang are with the Department of Design Engineering, Delft University of Technology, Delft, The Netherlands.*
- *Xifeng Gao is with the Department of Computer Science, Florida State University, US.*
- *Weiming Wang is also with the School of Mathematical Sciences, Dalian University of Technology, China.*
- *Corresponding Author: Jun Wu, E-mail: j.wu-1@tudelft.nl*

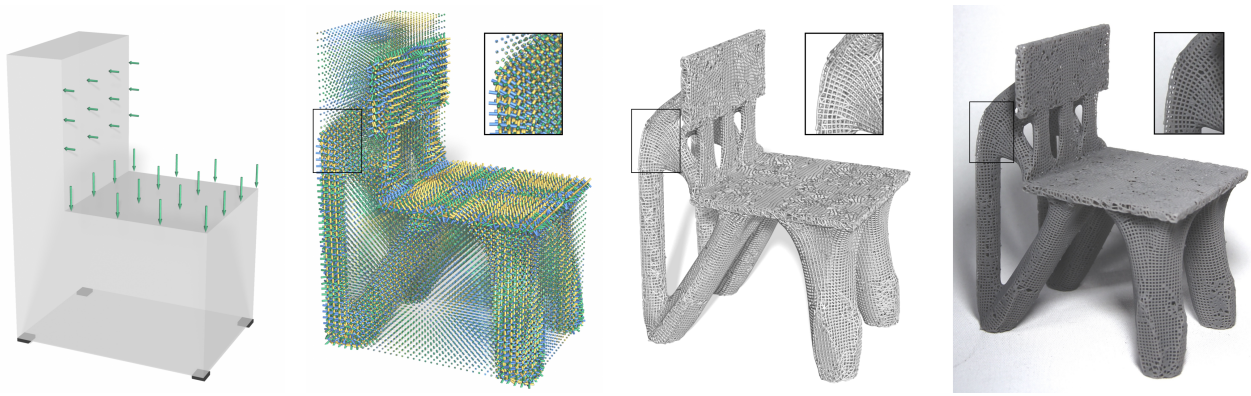


Fig. 1: From left to right: Given a design domain with specified external loads, our method optimizes the distribution of lattice materials for maximizing stiffness. From the optimized, locally-defined lattice configuration, a globally connected lattice structure is compiled, and fabricated by 3D printing.

- A new formulation to allow simultaneous optimization of the shape and the lattice distribution.
- A novel approach for extracting globally consistent lattice structures that accommodate anisotropy and heterogeneity.

Our method generates highly detailed lattice structures. The optimized lattice chair in Figure 1, for instance, consists of $178,291$ struts, achieved on a simulation resolution of $140 \times 100 \times 200$.

The remainder of the paper is organized as follows. In the next section we review related work. In Section 3 we give an overview of the proposed method. In Sections 4 and 5 the two major steps of our method, lattice optimization and compilation, are presented. Results and analysis are presented in Section 6, before the conclusions are given in Section 7.

2 Related Work

2.1 Structural Optimization for 3D Printing

In the era of 3D printing (and more broadly, digital fabrication), structural optimization becomes increasingly relevant in computational design [8]. Skin-frame structures [9], honeycomb-like Voronoi structures [10], tree-like supporting structures [11], and bone-inspired porous structures [12] have been optimized as lightweight infill for prescribed 3D shapes. Guided by outputs from structural optimization, Martinez et al. proposed to use graded orthotropic foams as a parameterized metamaterial to fill a prescribed shape [13], [14]. In contrast to design and optimize internal structures for prescribed shapes, our method optimizes concurrently the shape and its internal microstructures for application-specific loads. Different from two-scale structural optimizations (e.g. [15], [1]) which assume axis-aligned microstructures, our method optimizes the orientation of microstructures, in particular, to align it with spatially-varying stress directions. We restrict our design method to lightweight microstructures that are composed of struts, i.e. lattice structures. We note that lattice structures are less optimal than closed-walled shell structures regarding stiffness, yet they have potential benefits regarding, among others, structural stability and manufacturing [16].

Lattice structures are typically aligned with a regular grid [17], [18]. Rosen and his co-authors proposed a method to design lattice structures that conform with the boundary surface of a prescribed 3D shape [19], [20]. Our method optimizes concurrently the shape and align the lattices with stress directions. The alignment of structures along principal stress directions improves structural performance [21], [3]. This principle has been applied to 2D planes (e.g. [22]) and curved surfaces [23], [24], [25]. The appealing 2D results are achieved by tracing stress directions or based on a ground structure approach [26], [27]. Due to their inherent challenges associated with the initialization of samples/nodes, a direct extension of these methods to 3D volumetric lattices is not applicable. Our method constructs stress-aligned 3D volumetric lattices, relying on homogenization-based topology optimization and field-aligned meshing.

Our method is among recent efforts on structural analysis and optimization for 3D printing. Stava et al. proposed a method to detect and correct structural defects [28]. Recent efforts include worst-case structural analysis [29], [30], and stochastic structural analysis [31]. Chen et al. proposed a solver for inverse elastic shape design [32]. Ulu et al. optimized structures under force location uncertainty [33]. Our method, targeting on stiffness maximization of lattice structures under certain static loads, is complementary to these efforts. Yet the integration goes beyond the scope of this paper.

2.2 Homogenization-based Topology Optimization

Topology optimization is an important design method for 3D printing, as it effectively leverages the fabrication flexibility to create structures with exceptional (mechanical) properties. Topology optimization transforms optimal shape design as a material distribution problem. In their seminal work, Bendsøe and Kikuchi proposed a homogenization method, which optimizes the distribution of square unit cells with variable rectangular holes [4]. Due to the lack of manufacturing means for such microstructures back then, the homogenization method was replaced by density-based approaches (e.g. SIMP [34]) which have since been widely used in industry and in many academic contributions (e.g. large scale optimization [35], [36], [37]).

In light of the capability of 3D printing to fabricate microstructures, the homogenization method was recently revisited to design structures with manufacturable mi-

crostructures [6], [7], [38], based on the rectangular hole model [4]. A challenge is to compile a continuous structure from hollowed cells that are defined on a regular grid, and that, after optimization, have different orientations. To this end, a projection approach proposed by Pantz and Trabelsi [5] was improved to connect the orthotropic microstructures [6], [39]. The output structure is represented by high-resolution pixels or voxels.

Our approach to design conforming lattice structures follows the homogenization approaches, but differs in three aspects. First, we propose a new parameterization of cells to ensure that the variable cells have a constant thickness, while allowing a large degree in anisotropy. Our intention of creating uniform thickness is to simplify all downstream operations including surface mesh creation, process planning, fabrication, surface treatment, inspection and qualification. Uniform thickness is a common practice in (metal) additive manufacturing of (axis-aligned) lattices [17], [18]. Such structures can also be fabricated by a direct extrusion in 3D [40], or by robotic fabrication [41], [42]. We note that variable thickness is not impossible with 3D printing. Second, our method simultaneously optimizes the lattice distribution and the shape, achieved using multiple design variables. Last but not least, while existing works exploited projection methods for generating high-resolution pixel or voxel models, we develop a novel approach based on field-aligned meshing to compile the lattice structure. The optimized structure is compactly represented by a graph. This direction shares a similar goal with the recent work by Arora et al. [43]. In contrast to the design approach [43], our method unlocks a large solution space by optimizing the porosity, anisotropy, and orientation of lattices.

2.3 Field-aligned Parameterization

We develop a lattice compilation method based on field-aligned parameterization which has been researched intensively in the past decade, especially for generating quadrilateral (quad-) mesh. We review briefly the more recent works on hexahedral (hex-) meshing, and for quad-meshing we refer an interested reader to the survey by Bommes et al. [44].

For a given 3D closed shape, field-aligned hex/hex-dominant meshing techniques typically consist of three steps [45], [46], [47], [48], [49], [50]. It starts by estimating the gradients of a volumetric parameterization using a directional field [51], [52], where the field is discretized per vertex or per tetrahedron and smoothly interpolated within the volume under a boundary alignment constraint. This is followed by computing a parameterization aligned with the estimated gradients by fitting. Finally it extracts the hex-mesh in the parametric space [53]. Robust hex-meshing remains a challenging problem. A promising direction is to topologically correct the directional fields [47], [48], [54], [55]. Lei et al. introduced a hex-mesh generation method based on surface foliation theory [56]. This approach, however, requires heavy topological pre-processing of the input.

The field-aligned parameterization pipeline is primarily used for generating semi-regular meshes. To ensure the validity of the mesh, complex geometric and topological computations are involved. In this paper we make use of field-aligned parameterization to generate lattice structures. This new application differs from mesh generation, as lattices are encoded by graphs rather than meshes. This goal sidesteps the numerical stability issue and geometrical and topological complexities typically occurred during mesh extraction from the parameterization.

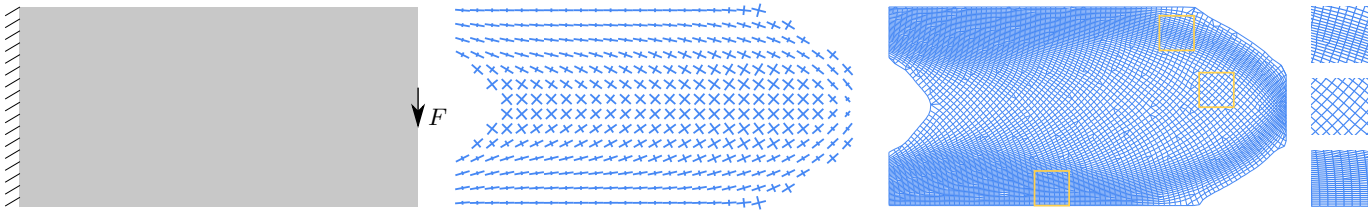
To efficiently and robustly extract consistent lattice structures, we extend the robust meshing approach that was proposed by Jacob et al. [57] and further developed by Gao et al. [50]. The per vertex local parameterization from [50] fits our purpose well since the local parameterization aligns exactly with the direction field by construction and permits fast and scalable computations. The extension proposed in this paper allows to incorporate anisotropy and heterogeneity.

The recent work by Arora et al [43] shares a similar goal as ours, i.e. to extract field aligned struts from stress directions. Our approach takes the optimized stress fields as input, without a field smoothing operation that compromises the accuracy of input fields. During lattice compilation, while they extract the struts by tracing stress lines and simplifying the duplicated ones, our approach directly generates struts by simple and efficient graph operations. This makes our approach fast and scalable, taking a couple of minutes for an input with tens of millions of tetrahedral elements (see Table 2).

3 Overview

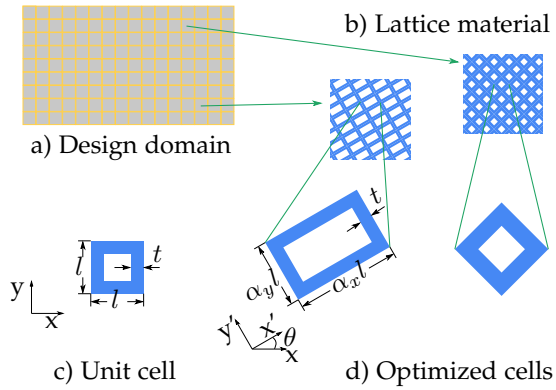
Given a design domain and application-specific loads, our method generates a lattice structure that maximizes structural stiffness. The struts in the optimized lattice structure conform with principal stress directions. Moreover, the struts on the boundary span a smooth surface faithfully approximating the optimized shape.

As illustrated in Figure 2 for 2D and Figure 1 for 3D, our approach consists of two steps. The first optimizes the shape (including its topology) and the distribution of lattice material within the shape. The input includes a design domain and boundary conditions (Figure 1 left and Figure 2a), as well as design specifications such as the target fraction of solid material. The design domain in 3D is represented by a closed triangle surface mesh. This mesh is voxelized, generating finite elements for simulation and optimization. The output is a set of fields, indicating, per element, the occupancy of lattice material, and the orientation and anisotropy of lattice material (Figure 1 second left and Figure 2b). A surface mesh is then reconstructed using Marching Cubes, representing the optimized shape, i.e. the interface between elements that are filled with lattice material and that are empty. The shape enclosed by this reconstructed surface mesh (or, the input surface mesh which defines the design domain, in case that the entire design domain is to be filled with lattices) is tetrahedralized. The optimized fields are then interpolated on the vertices of the tetrahedral model. The vertices, including their connectivity and their associated field values, serve as input for the second step, which compiles a globally connected conforming lattice structure composed of struts (Figure 1 second right and Figure 2c). The output lattice structure is encoded by a graph.



a) Design domain b) Optimized lattice distribution c) Continuous lattice structure

Fig. 2: A 2D example, illustrating the pipeline of our approach. Given the design specification (a), the first step optimizes the distribution of lattice materials (b). The second step extracts a continuous lattice structure corresponding to the optimized lattice configuration (c).



c) Unit cell d) Optimized cells

Fig. 3: The design domain (a) is discretized into bilinear quadrilateral elements. Each element is filled with lattice material (b), i.e. a block of periodic cells (d). The cells are adapted by scaling and rotating a unit cell (c).

4 Lattice Optimization

The goal of our optimization is to find the optimal distribution of lattice material that maximizes structural stiffness, subject to a number of design constraints. To this end, the design space is discretized into a regular grid of bilinear square elements in 2D or trilinear cubic elements in 3D. As illustrated in Figure 3 for a 2D rectangular design domain, each element is to be filled by repeating a unique, rectangular-shaped cell. The cells are adapted from a unit cell by scaling and rotation. The scaling factors and rotation matrices are to be optimized. The scaling factors for the cell in element e are denoted by α_e, and in 2D by $(\alpha_{e,x}, \alpha_{e,y})$ and in 3D by $(\alpha_{e,x}, \alpha_{e,y}, \alpha_{e,z})$. The rotation matrix for element e is denoted by R_e.

The unit cell in 2D is a hollowed square with a side length of l and a thickness of t, which are specified by the user. In 3D, the cubic unit cell consists of the union of all 12 edges with square cross section of thickness t. During scaling the side length of cells (l) is elongated, while the thickness (t) is kept constant. We note that this treatment differs from the standard scaling where the thickness is also scaled. This creates cells with gradation in the fraction of solid material (v_e),

$$v_e(\alpha_e) = 1 - \frac{(\alpha_{e,x} l - 2t)(\alpha_{e,y} l - 2t)}{\alpha_{e,x}\alpha_{e,y} l^2}. \tag{1}$$

This gradation allows the optimization to place adapted cells with a smaller fraction of solid material in regions where the stress is relatively small. Furthermore, per axis elongation allows to increase the mechanical anisotropy of cells. This is beneficial since the stress tensors are typically anisotropic.

Besides a scaling factor per axis and a rotation matrix, each element is assigned a variable φ_e, to indicate whether the element is empty ($\varphi_e = 0$) or filled ($\varphi_e = 1$) with lattice material. The set of elements that are filled with lattice material defines the overall shape of the optimized structure. To allow for gradient-based numerical optimization, the variable φ_e is relaxed to take intermediate values, i.e. $\varphi_e \in [0, 1]$. This variable is akin to the density variable in classical density-based topology optimization, which in that context indicates the fraction of *solid* material. In the context of lattice optimization, this variable shall be interpreted as the fraction of *lattice* material. The fraction of solid material per element (ρ_e) depends on φ_e and the fraction of solid material within an adapted cell (v_e), i.e.

$$\rho_e(\varphi_e, \alpha_e) = \varphi_e v_e(\alpha_e). \tag{2}$$

As the design space is parameterized by the fraction of lattice material ($\boldsymbol{\varphi}$), scaling factor ($\boldsymbol{\alpha}$), and orientation matrix ($\boldsymbol{R}$), the optimization problem is given as

$$\begin{aligned}
\min_{\boldsymbol{\varphi},\boldsymbol{\alpha},\boldsymbol{R}} \quad & J = \mathbf{F}^{\mathsf{T}}\mathbf{U}(\boldsymbol{\varphi}, \boldsymbol{\alpha}, \boldsymbol{R}) && \text{(3a)}\\
s.t.: \quad & \textstyle\sum_e \rho_e(\varphi_e, \alpha_e) \le \overline{v} N && \text{(3b)}\\
& \varphi_e \in [0.0,\ 1.0],\ \forall e && \text{(3c)}\\
& \alpha_{e,k} \in [\underline{\alpha}_k,\ \overline{\alpha}_k],\ k \in \{x, y, z\}, \forall e. && \text{(3d)}
\end{aligned}$$

Here the objective is to minimize the work done by the external force, which is equivalent to minimize compliance (i.e. stiffness maximization). $\mathbf{F}$ denotes the force vector that is applied to the design domain. The force vector is constant. $\mathbf{U}$ denotes the displacement vector of the shape when it comes to its static equilibrium under the external force $\mathbf{F}$. The first constraint, Eq. 3b, restricts the amount of solid material, where $\overline{v}$ is the fraction of available solid material, and N is the number of elements in the design domain. The second constraint, Eq. 3c, sets bounds for the fraction of lattice material ($\boldsymbol{\varphi}$). The third constraint, Eq. 3d, sets bounds for the scaling factors ($\boldsymbol{\alpha_x}, \boldsymbol{\alpha_y}, \boldsymbol{\alpha_z}$). The lower and upper bounds of the scaling factors are user-defined.

The novelty of this formulation is two-fold. First, by optimizing the scaling factors rather than the thickness of hollowed cells, it ensures that all struts in the optimized structure have the same thickness. As discussed in Section 2.2, this eases the control of the 3D printing process. Second, we assign an additional variable φ to indicate the occupation of lattice material. This makes the formulation more general. Prescribing $\varphi = 1$ leads to optimized lattices

that fill the entire design domain. This is useful as infill for prescribed shapes. Allowing φ to be decided by the optimization enables both the shape and the lattice to evolve simultaneously, achieving a higher stiffness.

4.1 Stiffness Matrix for Lattices

The objective function, Eq. 3a, involves the displacement vector ($\mathbf{U}$), which is related to the external force ($\mathbf{F}$). The unknown $\mathbf{U}$ is computed by solving the equilibrium equation with finite element analysis,

$$\mathbf{K}(\boldsymbol{\varphi}, \boldsymbol{\alpha}, \boldsymbol{R})\mathbf{U} = \mathbf{F}. \tag{4}$$

Here the stiffness matrix, $\mathbf{K}$, is assembled from per element stiffness matrix, $\mathbf{K}_e(\varphi_e, \alpha_e, R_e)$.

In standard finite element analysis of solids [58], the element stiffness matrix $\mathbf{K_e}$ is computed by integrating over the domain of an element, Ω_e,

$$\mathbf{K}_e = \int_{\Omega_e} B^\mathsf{T} D_e B dx, \tag{5}$$

where B is the element strain-displacement matrix for linear basis functions [58]. D_e represents the fourth order elasticity tensor, computed based on the Young's modulus and Poisson's ration of the solid material.

For analyzing elements that are filled with lattice material, the elasticity tensor D_e is not constant but rather depends on design variables α_e and R_e. Let us first consider an element that is filled lattice with $\varphi_e = 1$. The stiffness matrix for lattices is calculated by

$$\mathbf{K}_e(1, \alpha_e, R_e) = \int_{\Omega_e} B^\mathsf{T} D_e(\alpha_e, R_e) B dx, \tag{6}$$

The elasticity tensor of a rotated lattice cell, $D_e(\alpha_e, R_e)$, is computed by rotating the elasticity tensor of this cell in its local coordinate system, $D_e(\alpha_e)$. In engineering notation, D_e is represented as a 3×3 matrix for 2D problems or a 6×6 matrix for 3D. The rotation of tensor is realized by

$$D_e(\alpha_e, R_e) = \overline{R}_e(R_e) D_e(\alpha_e) \overline{R}_e^\mathsf{T}(R_e), \tag{7}$$

where the tensor rotation matrix $\overline{R}$ is given in the Appendix.

The effective elasticity tensor of an elongated cell, $D_e(\alpha_e)$, is evaluated by numerical homogenization. We make use of the Matlab code provided in [59] and [60] for homogenization in 2D and 3D, respectively. Given the scaling factors, the domain of the elongated unit cell is discretized by square/cube finite elements with bilinear/trilinear basis functions. To avoid performing homogenization for every α_e during the optimization process, we precompute D_e for regularly sampled α values. In 2D, we fit a surface for every non-zero entry in D over the 2D domain of $[\underline{\alpha}_x, \overline{\alpha}_x] \times [\underline{\alpha}_y, \overline{\alpha}_y]$. In 3D we use trilinear interpolation. The derivative of each non-zero entry in D with respect to α is evaluated using the interpolation.

For elements with φ_e between 0 and 1, we use the power law from density-based approaches [34] to interpolate,

$$\mathbf{K}_e(\varphi_e, \alpha_e, R_e) = \varphi_e^p \mathbf{K}_e(1, \alpha_e, R_e), \tag{8}$$

where the parameter p (typically $p = 3$) is introduced to penalize intermediate values in φ_e, and consequently the optimization steers φ_e towards either 0 or 1.

4.2 Solving

The optimization problem (Eq. 3) is solved in an iterative process, as in [6]. In each iteration the following computational steps are performed, until the maximum change in design variables is smaller than a threshold (or the maximum number of iterations is reached).

First, the equilibrium equation (Eq. 4) is solved, obtaining the displacement vector, $\mathbf{U}$. From the element displacement vector ($\mathbf{U}_e$), strain ($\overline{\epsilon}_e$) and stress ($\overline{\sigma}_e$) per element, in engineering notation, are calculated by $\overline{\epsilon}_e = B\mathbf{U}_e$ and $\overline{\sigma}_e = D_e(\alpha_e, R_e)\epsilon_e$, respectively.

Second, design variables φ and α are updated using a gradient-based solver. We make use of the method of moving asymptotes (MMA) [61]. To avoid checkerboard patterns, the design variables are regularized into $\tilde{\varphi}$ and $\tilde{\alpha}$ using the so-called density filter. $\tilde{\varphi}$ is then projected into $\overline{\tilde{\varphi}}$ by a (smoothed) Heaviside operation, to approach a 0-1 solution. The filter and Heaviside operator are widely used in density-based approaches, e.g. in [62], [12].

Third, the orientation of each element (R_e) is updated based on the associated stress tensor (σ_e). The stress tensor is symmetric positive-definite. By eigendecomposition we obtain three mutually orthogonal principal stress directions (v_1, v_2, v_3). The eigenvectors are ordered by respective eigenvalues in ascending order, i.e. $\gamma_1 \leq \gamma_2 \leq \gamma_3$. As shown by Pedersen [3], the optimal orientation of an orthotropic material coincides with the principal stress directions, hence the element is rotated by $R_e = [v_1^\mathsf{T}; v_2^\mathsf{T}; v_3^\mathsf{T}]$.

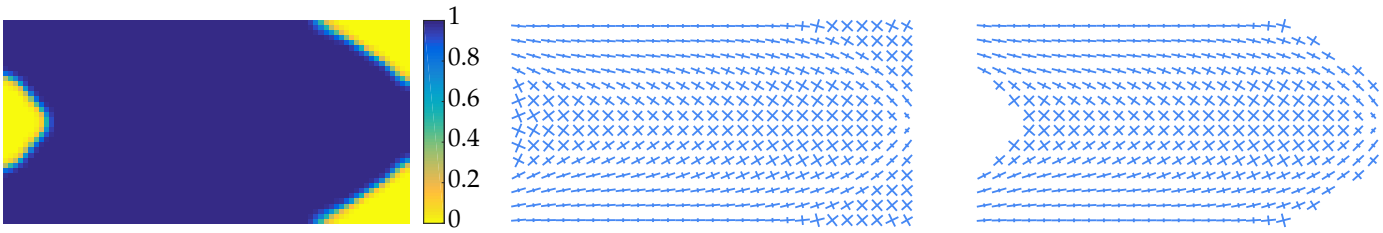
Fourth, the stiffness matrices of lattices, K_e, are recalculated based on the updated orientation (R_e) and regularized variables $\overline{\tilde{\varphi}}$ and $\tilde{\alpha}$, according to Section 4.1.

4.3 Example

The output of our optimization is a set of fields defined on the design domain. Figure 4 visualizes these fields for a rectangular domain, which is discretized by a grid of 80×40 elements. The unit cell has $l = 10t$. The maximum fraction of solid material is 0.15. Figure 4a shows the optimized lattice fraction field. The field contains values very close 0 or 1 (cf. the colorbar). Even with a fraction of solid material as small as 0.15, the lattice covers a large portion of the design space. This is due to the fact that the unit cell has a small fraction of solid material (i.e. 36%, with $l = 10t$). Figure 4b visualizes the orientation of optimized cells. Here the rotated frame is elongated according to the respective scaling factor per axis. For clarity the frames are shown for regularly-spaced samples. On the right, the frame field is visualized for elements which have a fraction of lattice material (φ_e) that is larger than a threshold (0.5). We note that since the optimized field φ contains values close to 0 or 1, Figure 4c is independent (almost) of the value of the threshold.

5 Lattice Compilation

Up to this step, we have equipped with a design domain with a set of fields including fraction of lattice infill, orientation, and scaling, that are optimized for the prescribed external loads (cf Figure 4). Since a region with a low fraction implies that little material is required, we extract



a) Occupancy of lattice materials (φ) b) Lattice orentiation (R) and scaling (α) c) R and α, for $\varphi_e \geq 0.5$

Fig. 4: Visualization of the optimized fields. In (b) and (c) the frame is elongated according to the respective scaling per axis (α), and then rotated according to the orientation (R).

a sub-area (volume) from the design space by thresholding (≥ 0.5) out low infill regions. With the actual shape being extracted, we now focus on generating a lattice structure that conforms to both the boundary of the shape and the directional and scaling fields.

Our problem setting differs from the typical meshing problem in that both of our input and output are quite relaxed from the conditions of being a mesh. For the input, we put no constraints on its geometrical quality (i.e. angles, edge ratios, etc) nor its topological correctness (i.e. permitting non-manifold features, holes, and intersections). This maximizes the scope of the problem but poses a great challenge to the design of a robust solution. For the lattice output, it does not require face (solid) elements, making complex topological operations in most of the meshing methods unnecessary for our purpose. Moreover, considering that it is not a hard requirement for our lattice structure to be all-hex cells for the designed structure to function, we choose the parameterization optimization in [50] that can be easily adapted to handle graphs and propose a simple extraction strategy to generate a lattice structure. The produced lattice structure contains mostly quad (hex)-like connections while allowing certain irregularity to adapt for rapid changes in the directions and scales.

Our method takes a graph with vertices of the optimized shape as the input: $G = (V, E)$, where every vertex $\mathbf{v} \in V$ has a position $\mathbf{x} \in \mathbb{R}^k$ (k is 2 for 2D and 3 for 3D), an orientation matrix $\mathbf{R} \in \mathbb{R}^{k\times k}$ encoding the cross directions and also denoting a local coordinate system, and a scaling vector $\boldsymbol{\alpha} \in \mathbb{R}^k$ composed of scales for the k axes of the local coordinate system. Our goal is to extract a lattice structure, which is another graph $G' = (V', E')$ that (1) reproduces the input direction and anisotropy as much as possible, and (2) has a resolution that can be flexibly controlled by a target edge length h.

In the following, we first describe the parameterization optimization that incorporates anisotropic orientations, and then present the lattice structure extraction.

5.1 Parameterization

Given an orientation field $\mathbf{O}$ that includes the cross directions for all the vertices, we want to compute a parameterization $\mathbf{P}$ with the gradient aligned to $\mathbf{O}$. Methods that compute a global parameterization with the gradient aligning to the orientation field in a least-square sense (e.g. [63], [64], [43] and [5], [7], [6], [39]), rely on non-linear optimization solvers which are not scalable to large datasets. We instead compute a parameterization for the input graph by representing it with a set of local parameterizations and minimizing an energy between the local parameterizations of adjacent vertices [57], [50]. The local nature of the parameterization makes it easily parallelizable and scalable to large inputs.

As illustrated in Figure 5a, the local parameterization of a vertex in 2D plane (or 3D volume) can be uniquely determined by its origin $\mathbf{p}$, the orientation matrix $\mathbf{R}$, and unit lengths $h\cdot\boldsymbol{\alpha}$, where h is the user-defined global target edge length. The unit lengths are fixed through the entire process. Unlike the previous approaches [50], [57] that treat directions as a 4 rotational symmetric field in 2D or a 24 rotational symmetric field in 3D, since the unit length varies for different axis, our coordinate system is mutable only by flipping the signs of each axis. The origin with a random initialization, is the variable we need to optimize.

Given the above setting, the optimization energy of the parameterization $\mathbf{P}$ is defined as the summation of all the squared differences of local parameterizations for each edge:

$$E(\mathbf{P}) = \sum_{i\in V}\sum_{j\in N(i)} ||\mathbf{p}_i - (\mathbf{M}_{ij}\mathbf{t}_{ij} + \mathbf{p}_j)||^2, \qquad (9)$$

where $N(i)$ is a set of all the vertices sharing an edge with vertex i, $\mathbf{M}_{ij}$ is an interpolation of $\mathbf{M}_i$ and $\mathbf{M}_j$ where $\mathbf{M} = \mathbf{RS}$ and $\mathbf{S}$ is the scaling matrix converted from $h\cdot\boldsymbol{\alpha}$, and $\mathbf{t}_{ij} \in \mathbb{Z}^k$ encodes the integer translations of $\mathbf{p}_j$. $\mathbf{M}_{ij}\mathbf{t}_{ij}+\mathbf{p}_j$ translates $\mathbf{p}_j$ by integer moves to the nearest position to $\mathbf{p}_i$, effectively avoiding the integer jumps between the two local parameterization and only the difference of their fractional parts is measured. The computation of $\mathbf{M}_{ij}$ requires interpolating the directions and scales separately,

$$\mathbf{M}_{ij} = ortho(\mathbf{R}_i + \mathbf{R}_j r(\mathbf{R}_i, \mathbf{R}_j)) \cdot \tfrac{(\mathbf{S}_i+\mathbf{S}_j)}{2}, \qquad (10)$$

where $ortho$ denotes the orthogonalization of a matrix, and $r(a, b)$ is the closest matching that gives the smallest difference between two coordinate systems which can be computed efficiently by enumerating all the cases. Note that, there is no scaling involved when computing the matching between orientations, which is similar to the one in [50]. The only difference is that in [50], there are 8 cases to compare for 2D and 48 cases for 3D, while we need to consider 4 and 8 cases in 2D and 3D respectively.

The integer translation between two connecting vertices in the parameterization space, $\mathbf{t}_{ij}$, is computed by a rounding operation,

$$\mathbf{t}_{ij} = round[\mathbf{M}_{ij}^{-1}(\mathbf{p}_i - \mathbf{p}_j)]. \qquad (11)$$

By doing so, the energy between the two vertices will be minimized.

We minimize $E(\mathbf{P})$ in a Gauss-Seidel style by iteratively visiting every vertex and smoothing the origin of each vertex by computing an average of all its neighbors. The pseudo code of the optimization is provided in Algorithm 1.

Algorithm 1 Optimize-Parameterizations ($\mathbf{P}$)

1: **for** $i = 1, \ldots, n$ **do**
2: $\quad \mathbf{p}'_i \leftarrow \mathbf{p}_i$, $d \leftarrow 0$
3: $\quad$ **for all** $j \in \mathcal{N}(i)$ **do**
4: $\quad\quad \mathbf{p}'_i \leftarrow \mathrm{d}\mathbf{p}'_i + \mathbf{p}_j + \mathbf{M}_{ij}\mathbf{t}_{ij}$
5: $\quad\quad d \leftarrow d + 1$
6: $\quad\quad \mathbf{p}'_i \leftarrow \mathbf{p}'_i / d$
7: $\quad$ **end for**
8: $\quad \mathbf{p}_i \leftarrow \mathbf{p}'_i + \mathbf{M}_i round\left[\mathbf{M}_i^{-1}(\mathbf{x}_i - \mathbf{p}'_i)\right]$
9: **end for**

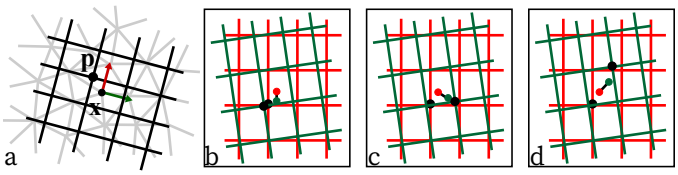


Fig. 5: (a) The local parameterization of a vertex in 2D plane. (b-d) Two local parameterizations of two close vertices in the input graph where they are not necessarily close in the parameterization space.

The last step in line 8 rounds each origin of a local parameterization $\mathbf{p}_i$ to the integer position closest to the vertex position $\mathbf{x}_i$. Consequently, each component of $\mathbf{t}_{ij}$ becomes -1, 0, or 1. For example, as illustrated in Fig. 5, $\mathbf{t}_{ij} = (0, 0)$ for Fig. 5b, $\mathbf{t}_{ij} = (\pm 1, 0)$ or $(0, \pm 1)$ for Fig. 5c, and $\mathbf{t}_{ij} = (\pm 1, \pm 1)$ for Fig. 5d. We note that this approach requires an input graph where the length of edges is much smaller than the desired length of the lattice struts, otherwise the integer translations after rounding could be larger than $|\pm 1|$.

After the rounding step at line 8 of Algorithm 1, if vertex i is on the input boundary, then $\mathbf{p}_i$ is projected onto the tangent plane of i. This projection step ensures the conformity of the finally extracted lattice structure.

To speed up the optimization, similar to [50], we construct a hierarchical structure of the input graph by halving the number of vertices for each level and perform the optimization on each level of the hierarchy by 50 iterations for 2D and 200 iterations for 3D.

5.2 Graph Extraction

In the input graph G, each vertex $\mathbf{v}$ has a smoothed local parameterization. The origin $\mathbf{p}$ of $\mathbf{v}$ provides a guidance for the vertex position in the output graph $G' = (V', E')$. Besides, the integer translation associated with each edge $(\mathbf{v}_i\mathbf{v}_j)$ of G, $\mathbf{t}_{ij} \in \mathbb{Z}^k (k = 2, 3)$, categorizes this edge as a specific element in G', depending on the L_0 norm of $\mathbf{t}_{ij}$ which is the number of ± 1s in $\mathbf{t}_{ij}$. In 3D ($k = 3$), this number can be

- 0 (i.e. $\mathbf{t}_{ij} = (0, 0, 0)$), indicating that the two vertices are very close in the parameterization space, and thus will be collapsed into a point in G',
- 1 ($\mathbf{t}_{ij} = (\pm 1, 0, 0), (0, \pm 1, 0)$, or $(0, 0, \pm 1)$), meaning that the edge is parallel to one of the stress directions, and thus will be kept in G',
- 2 ($\mathbf{t}_{ij} = (\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1)$, or $(0, \pm 1, \pm 1)$) or 3 ($\mathbf{t}_{ij} = (\pm 1, \pm 1, \pm 1)$), respectively corresponding to a rectangular or cuboid diagonal, which deviates from the stress directions and thus shall not appear in G'.

For example, black edges and green dashed edges in Figure 6 left correspond to $||\mathbf{t}_{ij}||_0 = 1$ and 2, respectively.

By utilizing the positional guidance of $\mathbf{p}$ and the indication of $\mathbf{t}_{ij}$, the graph extraction is straightforward: collapse the edges with $||\mathbf{t}_{ij}||_0 = 0$ (dots in Figure 6 represent the averaged positions of collapsed edges), keep the edges with $||\mathbf{t}_{ij}||_0 = 1$, and remove the diagonal edges (i.e. $||\mathbf{t}_{ij}||_0 = 2$ or 3).

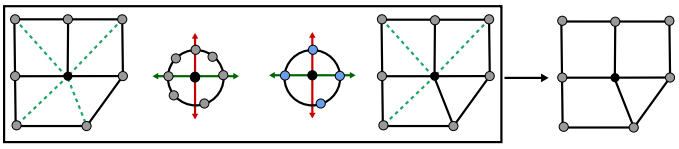


Fig. 6: Left: for a vertex in the graph, the nearest diagonal edges from its rotational directions will be relabelled to be maintained in the final graph if there is no edge representing the corresponding directions. Right: after the relabeling, our final graph is generated by discarding all the diagonal edges.

While the above procedure generates a graph with mostly right angles, we notice T-junctions in the final graph with near flat angles that are suboptimal for the stiffness of the lattice structure. Figure 6 left illustrates a vertex with T-junction in 2D. This can be attributed to the fact that the removal of the diagonal edges is aggressive. The T-junctions appear near singularities of the parameterization (similar to the positional singularities in [50]) which result in elements with non-right angles, for example, triangles and pentagons in 2D, and prisms and general polyhedra in 3D.

To address this issue, we propose to keep some diagonal edges in the final graph. Specifically, right after collapsing edges with $||\mathbf{t}_{ij}||_0 = 0$, we check the configuration of every vertex in the graph and identify critical diagonals. As illustrated in Figure 6, for a vertex in black, the process is done by first normalizing all of its adjacent edge vectors onto a unit circle (sphere in 3D), then computing their nearest directions over 4 rotational-symmetric ones in 2D (6 in 3D), e.g. red and dark green arrows, and finally relabelling a diagonal edge to be an edge with $||\mathbf{t}_{ij}||_0 = 1$ such that each of the 4 (6 in 3D) stress directions is represented by an edge that is close to the direction (Figure 6 right).

In summary, the process to extract the graph G', i.e. a lattice structure, consists of the following steps.

1) Categorize the edges in G based on $||\mathbf{t}_{ij}||_0$.
2) Group vertices in G according to $||\mathbf{t}_{ij}||_0$ such that groups are connected by edges with $||\mathbf{t}_{ij}||_0 \neq 0$. Note that a group might contain only a single vertex.
3) Generate the initial G'. For each group, a new vertex is positioned at the average of the origins of vertices in G. This vertex inherits the edges to new vertices that are converted from neighbouring groups.

4) Categorize the edges in G' based on $||\mathbf{t}_{ij}||_0$.
5) Identify and relabel critical diagonals in G' to avoid T-junctions, and remove other diagonal edges.

6 Results

6.1 Examples

Our method works for both 2D and 3D. Figure 7 shows three optimized 2D lattice structures. In (a), the lattice distributes across the prescribed curved shape, with spatial variations in orientation, porosity, and anisotropy. In (b) and (c), the optimized lattices cover a subset of a rectangular design domain, with variations in orientation in (c) and additionally variations in porosity and anisotropy in (b). The unit cell in 2D is specified with $l = 10t$, $\underline{\alpha} = 1$, and $\overline{\alpha} = 4$.

Figure 8 shows 3D lattice structures optimized by our method. Our method is also applicable to design lattices that spread over a prescribed 3D curved shape, as shown in Figure 9. 3D examples are optimized with a unit cell using $l = 4t$, $\underline{\alpha} = 1$, and $\overline{\alpha} = 2$. Figure 10 shows fabricated bridge and cantilevers. The models are 3D printed with a Formlab Form 2 printer which uses stereolithography (SLA). The dimension of models and the thickness of struts are scaled to comply with the volume and feature size of the printer. The printed femur (Figure 11) has a dimension of $112.4 \times 77.9 \times 133.1\, mm^3$, with a thickness of $0.5\, mm$. The chair (Figure 1) is $110.8 \times 76.6 \times 142.1\, mm^3$, with a thickness of $0.6\, mm$.

6.2 Evaluation

6.2.1 Design options

We evaluate the influence of various design options on the resulting lattice structures using the 2D cantilever (Section 4.3, Figure 4), with the fraction of solid material $\overline{v} = 0.15$, and bounds for the scaling factors $\underline{\alpha} = 1$ and $\overline{\alpha} = 4$. The optimized fields and compiled lattice structures are shown in Figures 12 and 13, respectively.

In the first row of Figure 12, the fraction of lattice is fixed, $\varphi = 1$. Consequently the lattice distributes across the entire rectangular design domain. In (a) the scaling is also fixed, while in (b) the optimization of scaling is enabled. The enlarged solution space leads a decrease in compliance (i.e. improved stiffness), 418.33 (a) vs. 282.62 (b). In (c), the scaling factors along individual axes are decoupled, resulting in a further decrease in compliance to 239.97.

In the second row of Figure 12, the fraction of lattice is optimized. Consequently, a shape evolves from the optimization, corresponding to $\varphi_e \geq 0.5$. Similar to the trend of compliance in the first row, it decreases from (d), to (e), and to (f), along with the increased flexibility in optimization. (f) has the smallest compliance among the six cases. It decreases from (a) by 44.39%. This study, in agreement with similar numerical comparisons for various optimization options [7], confirms the significance of adaptive porosity and anisotropic features for stiffness maximization [65].

As a reference, an axis-aligned uniform lattice structure covering the entire domain (i.e. corresponding to the initialization of Figure 12) is evaluated. Its compliance is 852.30, which is more than twice larger than the design in Figure 12a, and 3.66 times larger than the design in Figure 12f. This comparison confirms the importance of aligning anisotropic microstructures with internal stress directions for stiffness maximization.

6.2.2 Accuracy

To evaluate the accuracy of our lattice compilation method, we perform a comparison of the compliance predicted by homogenization with the compliance of lattice structures by a full-resolution finite element analysis. To this end, the computational domain of the six lattice structures in Figure 13 is discretized by a finite element resolution of 4096×2048, and analyzed using a geometric multigrid elasticity solver [66]. The voids among lattices are approximated by a weak ersatz material with a (relative) Young's modulus of 10^{-2}. The comparison is summarized in Table 1. The difference in compliance is between 2.89% and 6.46%. This demonstrates that our lattice compilation introduces little error to the predicted performance from homogenization-based optimization. We note that homogenization theory assumes infinite periodicity of the cells, while for fabrication the compiled lattice has a finite physical size. This effect has been evaluated and reported in e.g. [6], [67]. Furthermore, we notice that the compiled lattices exhibit a small number of triangles. This also partially explains the discrepancy since pure rectangles are assumed in homogenization.

TABLE 1: The difference in compliance predicted by homogenization and a full-resolution analysis, for the lattice structures shown in Fig. 13.

	a	b	c	d	e	f
Homo.	418.33	282.62	239.97	332.81	277.27	232.64
Full res.	444.78	300.15	255.48	323.18	292.66	241.94
Diff.	6.32%	6.20%	6.46%	2.89%	5.55%	4.00%

6.2.3 Computational performance

Table 2 presents statistics of 3D model complexity and computational performance. The experiments were run on a standard desktop PC equipped with an Intel Xeon E5-1650 v3 processor (12 cores) running at 3.5 GHz, 64 GB RAM, and an Nvidia GTX1080 graphics card with 8 GB memory. The optimization and compilation together take less than 1 hour even for complex models such as the chair and femur.

The group of columns 2-8 is related to lattice optimization. From the cantilever and bridge examples, it can be observed that with increasing design flexibility the compliance (J_{com}) decreases. This agrees with the 2D tests in Figure 12. The increased design flexibility is also reflected by an increase of time associated with updating stiffness matrices, which is counted in T_{FEA}. The optimization time of the gradient-based solver for $\boldsymbol{\varphi}$ and $\boldsymbol{\alpha}$, T_{Opt}, increases accordingly as well.

The resolution of optimized fields is refined by a regular subdivision (1 element $\rightarrow$ 2^3 elements), followed by trilinear interpolation of the fields. While our lattice compilation algorithm takes a general graph as the input, in our implementation, we use triangle meshes and tetrahedral meshes which are purely for the convenience of computing vertex normal. This step costs 45$\sim$70 seconds (cantilevers, Figure 8) to 4 minutes and 26 seconds (chair, Figure 1). The refinement generates a large number of vertices (#vertex)

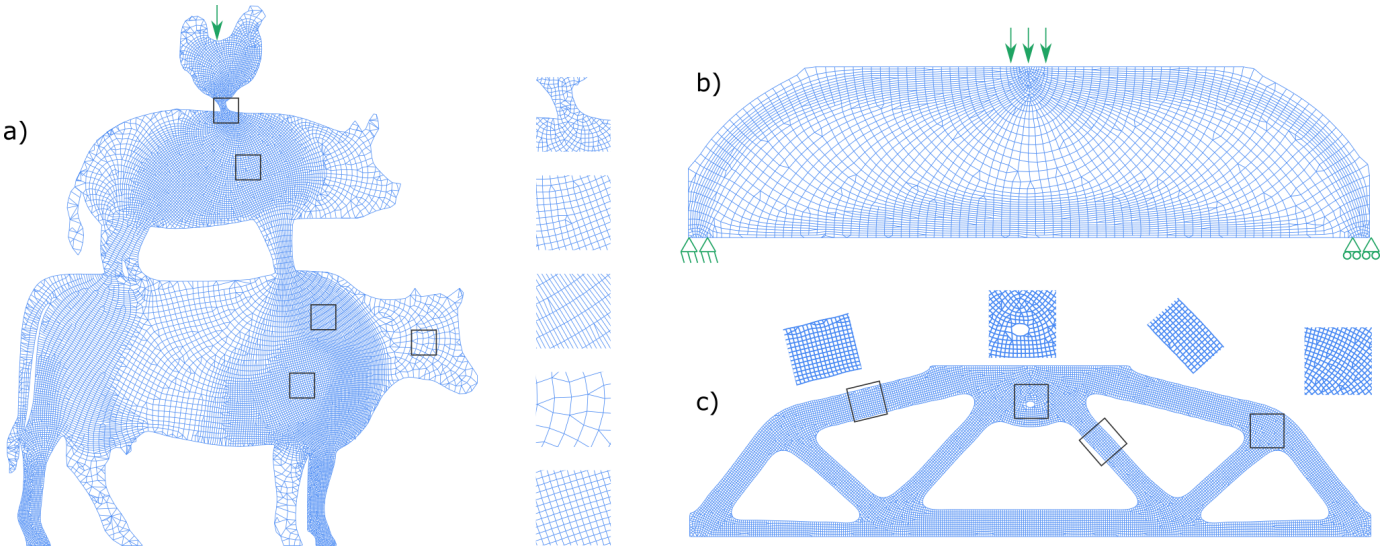


Fig. 7: Optimized 2D lattice structures for a prescribed freeform shape (a) and a rectangular design domain (b and c). The optimized lattice structures possess spatial variations in orientation, porosity, and anisotropy.

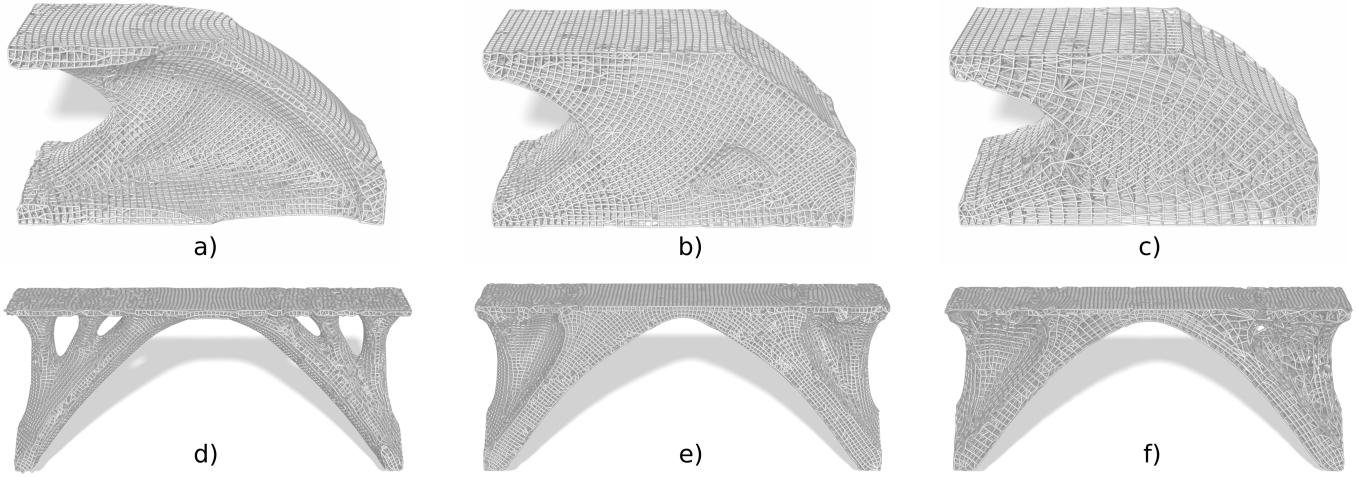


Fig. 8: 3D lattice structures optimized from cuboid design domains, showing spatial variations in orientation, porosity and anisotropy. The design options are: (left) fixed $\alpha = 1$ with design variables R and φ, (middle) design variables R, φ, and α with $\alpha_x = \alpha_y = \alpha_z$, (right) full flexibility. With the increased design flexibility, the compliance reduces from left to right: 110.84 $\rightarrow$ 96.03 $\rightarrow$ 85.85 (cantilever), 230.52 $\rightarrow$ 177.86 $\rightarrow$ 149.96 (bridge).

organized as tetrahedral elements (#tet), supplied to the lattice compilation. The compiled lattice has as many as 305k struts, for the femur model. Timings for pre-processing, i.e. building data structures (T_{pre}), local parameterization (T_{posy}), and graph extraction (T_{extr}) are reported.

In the last two rows, the optimized fields are refined twice (1 element $\rightarrow 4^3$ elements). This creates highly detailed lattice structures as shown in Figure 14.

6.3 Comparison and Discussion

Comparison with solid structures from density-based topology optimization [34] A 2D simply supported beam is optimized using our method and the classic density-based approach – Solid Isotropic Material with Penalization (SIMP). The lattice and solid structure generated by our method and SIMP, are shown in Figure 15 a) and b), respectively. The physical sizes are $294.8 \times 74.2 \times 60\ mm^3$, and the struts have an in-plane thickness of $0.8\ mm$, which is twice the nozzle size. They were fabricated by a Ultimaker S5 printer using flexible TPU (thermoplastic polyurethane) material. While the digital models were designed using the same fraction of solid material, with 3D printing the lattice structure is heavier (52 grams vs. 46 grams) due to the delicate tool-path.

The load condition of the 3D printed specimen is shown in Figure 15c. It is supported at the two ends on the bottom, while a downward force is applied on the top middle. To avoid out-of-plane buckling of these thin specimens, two wooden plates (with open square windows for observation) are placed to clamp the specimen (Figure 15d). The clamping plates are placed with a gap of $62\ mm$, slightly larger than the thickness of the specimen.

The force-displacement plots for multiple tests are shown in Figure 15e. The forces on the solid structure increase more steeply than on the porous structure, meaning that the solid structure from SIMP has a higher stiffness. However, the forces on the solid structure turn down after

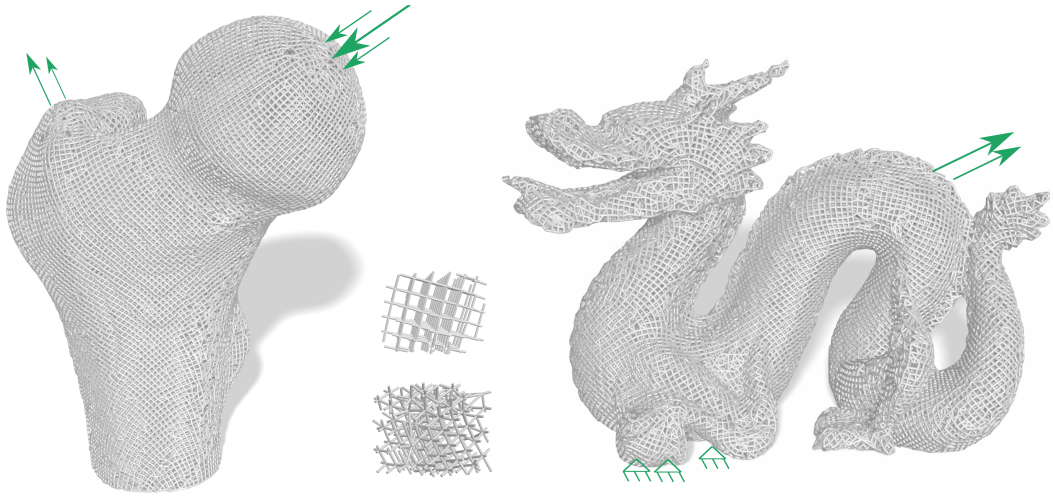


Fig. 9: Optimized 3D lattice structures for prescribed curved shapes. The optimized lattice structures possess spatial variations in orientation. The two samples are taken from inside the femur.

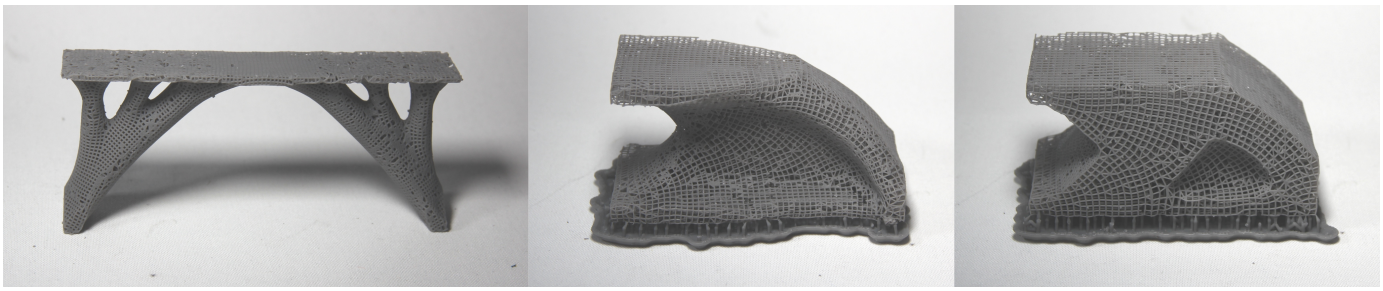


Fig. 10: Optimized lattice structures fabricated by 3D printing.

TABLE 2: Statistics of 3D model complexity and computational performance. The timing is reported in minutes.

Model	Resolution	#Ele.	Vol.	#It.	J_{com}	T_{FEA}	T_{Opt}	#vertex	#tet	#strut	T_{pre}	T_{posy}	T_{extr}	T_{Total}
3D cantilever (Fig. 8a)	$100 \times 50 \times 50$	250k	0.2	60	110.84	3.11	0.85	0.89m	5.0m	48k	1.80	5.28	0.33	11.36
3D cantilever (Fig. 8b)	$100 \times 50 \times 50$	250k	0.2	60	96.03	3.97	1.62	1.65m	9.62m	87k	1.75	9.98	0.70	18.03
3D cantilever (Fig. 8c)	$100 \times 50 \times 50$	250k	0.2	60	85.85	5.83	2.65	1.62m	9.41m	25k	1.83	6.50	0.57	17.38
Bridge (Fig. 8d)	$200 \times 38 \times 88$	644k	0.1	60	230.52	15.13	1.88	1.18m	6.54m	63k	1.41	7.09	0.47	25.97
Bridge (Fig. 8e)	$200 \times 38 \times 88$	644k	0.1	60	177.86	16.76	3.80	2.02m	11.57m	111k	2.84	13.58	1.14	38.12
Bridge (Fig. 8f)	$200 \times 38 \times 88$	644k	0.1	60	149.96	21.10	6.29	1.89m	10.77m	35k	2.40	8.29	0.72	38.79
chair (Fig. 1)	$140 \times 100 \times 200$	1.8m	0.1	60	193.5	30.92	5.03	3.32m	18.60m	178k	4.15	18.66	1.87	60.63
femur (Fig. 9)	$140 \times 93 \times 182$	696k	0.5	6	163.4	0.99	0	5.86m	14.26m	305k	12.36	35.50	5.94	54.79
dragon (Fig. 9)	$200 \times 90 \times 143$	461k	0.5	6	99.4	1.12	0	4.09m	23.31m	200k	5.09	24.84	2.88	33.92
3D cantilever (Fig. 14)	$100 \times 50 \times 50$	250k	0.2	60	110.84	3.11	0.85	6.65m	38.50m	351k	7.19	33.44	6.25	50.84
Bridge (Fig. 14)	$200 \times 38 \times 88$	644k	0.1	60	230.52	15.13	1.88	8.64m	49.63m	462k	12.50	56.56	14.35	100.42

they reach a peak of about 62 N. This is due to the (in-plane) buckling of the compressed bars. In contrast, the lattice structure can support a maximum force that is twice larger before it buckles. This is due to the increased effective cross-section area of the substructures. This test, in agreement with previous physical tests on 3D printed isotropic infill [68], confirms the significance of lattice structures for buckling stability. We note that directly accounting for buckling stability in topology optimization is more expensive than just compliance minimization, due to the less intuitive definition of the buckling mechanism and demanding eigenvalue problems [69]. Lattice structures, although not directly optimized for maximal buckling load, have a very good behaviour against buckling.

Comparison with bone-like porous structures [12] Wu et al. proposed a density-based approach to design bone-like porous structures using constraints on local material volume [12]. Figure 16 compares the porous structure and the lattice structure, generated with the same boundary conditions (see Fig. 2a) and the same fraction of solid material. The porous structure was optimized with a local volume fraction of 0.36, leading to a total volume fraction of 0.288. We then run lattice optimization with this total volume fraction, with the design options of rotation and scaling. The bone-inspired infill was optimized with a finite element resolution of 400×200, while the lattice was obtained with a simulation resolution of 80×40.

The convergence in compliance is plotted in Figure 16

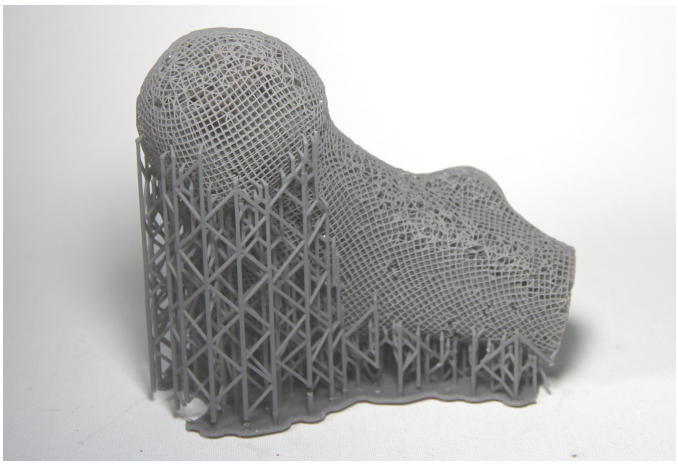


Fig. 11: 3D printed femur with supports.

right. The compliance of bone-like infill and conforming lattice is 184.64 and 177.29, respectively, meaning that the lattice structure is stiffer. Lattice optimization converges much faster, and since it runs on a lower resolution, this leads to a significant speed up. The optimization of lattice took 1 minute and 7 seconds (60 iterations), while the optimization of bone-like infill cost 40 minutes (1000 iterations). Both 2D tests are performed in Matlab.

Discussion of Arora et al. [43] Arora et al. proposed a method to construct a graph with its edges aligned with stress directions from simulation of the solid shape [43]. This approach does not explicitly optimize the porosity nor anisotropy. It relates to the option in our method with fixed φ and α (cf. Figures 12a and 13a). The result in Figure 12f has demonstrated that with optimized porosity and anisotropy, the compliance reduces by 44.39%. We note that under the option of fixed φ and α, after aligning the lattice, our method re-calculates stress directions and updates the lattice orientation. This leads a minor but noticeable decrease in compliance than aligning the lattice with stress directions from the solid shape (420.58 $\rightarrow$ 418.90).

Our lattice compilation approach is scalable, for example, the number of struts is more than two orders of magnitude larger compared with examples shown in [43]. This allows to design highly detailed lattice structures. Figure 14 shows optimized lattice structures with 462k struts (bridge) and 351k struts (cantilever).

7 Conclusions

In this paper we have presented a novel method to design conforming lattice structures by homogenization-based topology optimization and field-aligned parameterization. It can compute not only an optimized lattice structure that occupies certain subregions of regular design domains but also lattices that spread over prescribed (curved) shapes. The optimized lattice structures conform with principal stress directions and the boundary of the (optimized) shape. Our method is scalable and allows to optimize highly detailed lattice structures, which can be fabricated by 3D printing. Numerical analysis on different design options confirms the importance of aligning anisotropic lattice with internal stress directions and the importance of lattice gradation in porosity and anisotropy. The compiled lattice structure, by a full-resolution finite element analysis, has a compliance very close to the compliance predicted by homogenization-based optimization. By physical tests we demonstrate that the optimized lattice structure can support a buckling load twice as large as topology optimized solid structures, at the price of a slight decrease in stiffness. Besides quantified structural performance, the optimized conforming lattice structures look remarkably appealing.

Future work Our method generates lattice structures particularly optimized for mechanical properties. It provides options to steer the optimization by configuring the design variables, and to adapt the output graph resolution in lattice compilation. For designs with lattice spreading across the entire design domain, it is found that the generated lattice, in certain areas where the stress is small, is less regular, e.g. around top right and bottom right corner in Fig. 13 (c), and near the boundary at the back of the pig in Fig. 7. This is attributed to the fact that principal stress directions, and consequently the optimal orientation field, do not coincide with the boundary (see the visualization of the orientation field in Fig. 12c). A potential solution to this problem is to incorporate explicit constraints on the orientation field, i.e. enforcing an alignment of the orientation field with the prescribed boundary in stress minimal regions.

Acknowledgements

The authors gratefully acknowledge the support from the LEaDing Fellows Programme at the Delft University of Technology, which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 707404. Weiming Wang wishes to thank the Natural Science Foundation of China (No. 61702079, U1811463).

Appendix

Denoting a 3×3 rotation matrix by

$$R = \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{pmatrix}, \tag{12}$$

the 6×6 rotation matrix for the elasticity tensor in engineering notation is written as

$$\overline{R} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \tag{13}$$

with

$$A = \begin{pmatrix} l_1^2 & m_1^2 & n_1^2 \\ l_2^2 & m_2^2 & n_2^2 \\ l_3^2 & m_3^2 & n_3^2 \end{pmatrix}, \tag{14}$$

$$B = \begin{pmatrix} 2m_1n_1 & 2n_1l_1 & 2l_1m_1 \\ 2m_2n_2 & 2n_2l_2 & 2l_2m_2 \\ 2m_3n_3 & 2n_3l_3 & 2l_3m_3 \end{pmatrix}, \tag{15}$$

$$C = \begin{pmatrix} l_2l_3 & m_2m_3 & n_2n_3 \\ l_3l_1 & m_3m_1 & n_3n_1 \\ l_1l_2 & m_1m_2 & n_1n_2 \end{pmatrix}, \tag{16}$$

and

$$D = \begin{pmatrix} m_2n_3 + m_3n_2 & n_2l_3 + n_3l_2 & m_2l_3 + m_3l_2 \\ m_3n_1 + m_1n_3 & n_3l_1 + n_1l_3 & m_3l_1 + m_1l_3 \\ m_1n_2 + m_2n_1 & n_1l_2 + n_2l_1 & m_1l_2 + m_2l_1 \end{pmatrix}. \tag{17}$$

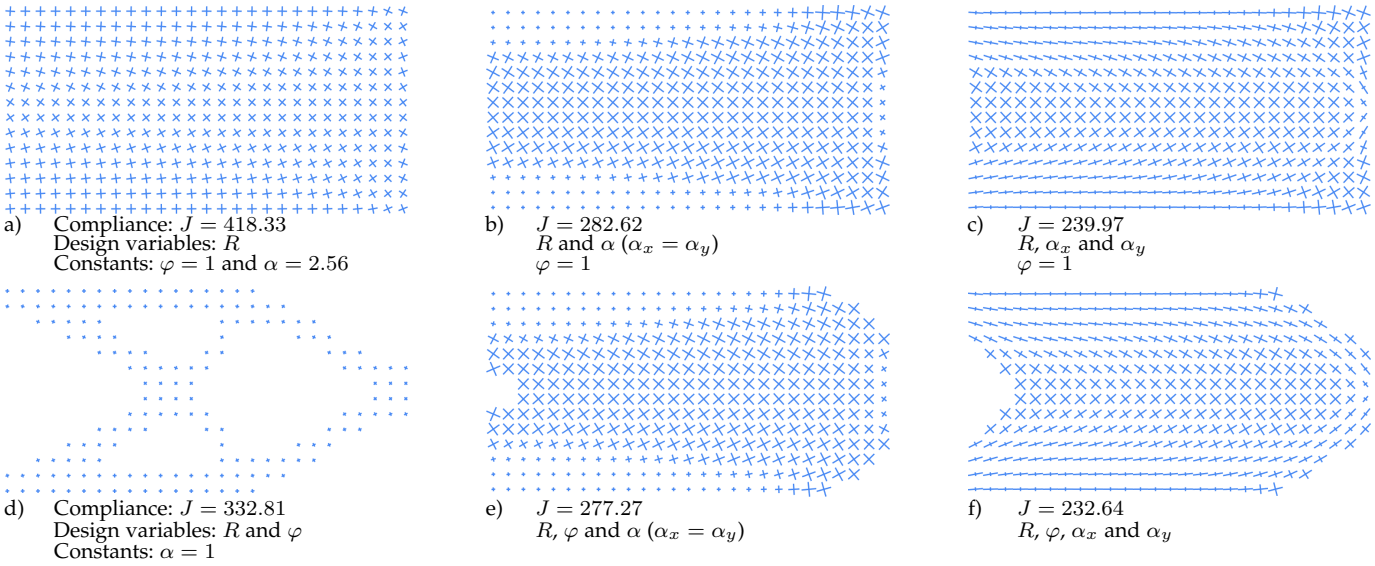


Fig. 12: Visualization of optimized 2D fields corresponding to different design options.

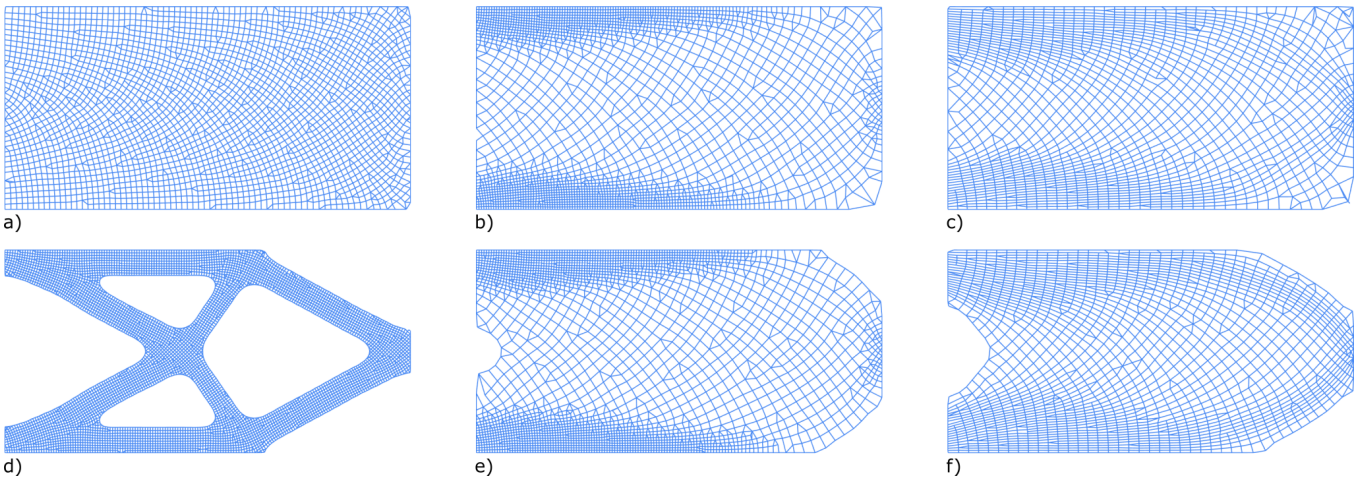


Fig. 13: Compiled lattice structures from the optimized, locally-defined lattice configuration (cf Fig. 12).

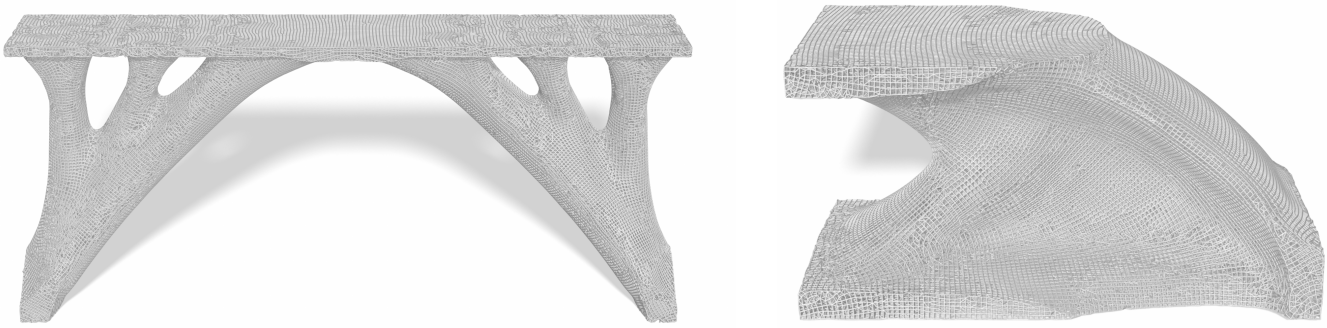


Fig. 14: Optimized lattice structures composed of 462k struts (bridge) and 351k struts (cantilever).

REFERENCES

[1] B. Zhu, M. Skouras, D. Chen, and W. Matusik, "Two-scale topology optimization with microstructures," *ACM Trans. Graph.*, vol. 36, no. 5, Jul. 2017.

[2] J. Liu, A. T. Gaynor, S. Chen, Z. Kang, K. Suresh, A. Takezawa, L. Li, J. Kato, J. Tang, C. C. L. Wang, L. Cheng, X. Liang, and A. C. To, "Current and future trends in topology optimization for additive manufacturing," *Structural and Multidisciplinary Optimization*, vol. 57, no. 6, pp. 2457–2483, Jun 2018.

[3] P. Pedersen, "On optimal orientation of orthotropic materials," *Structural optimization*, vol. 1, no. 2, pp. 101–106, Jun 1989.

[4] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197 – 224, 1988.

[5] O. Pantz and K. Trabelsi, "A post-treatment of the homogenization method for shape optimization," *SIAM Journal on Control and Optimization*, vol. 47, no. 3, pp. 1380–1398, 2008.

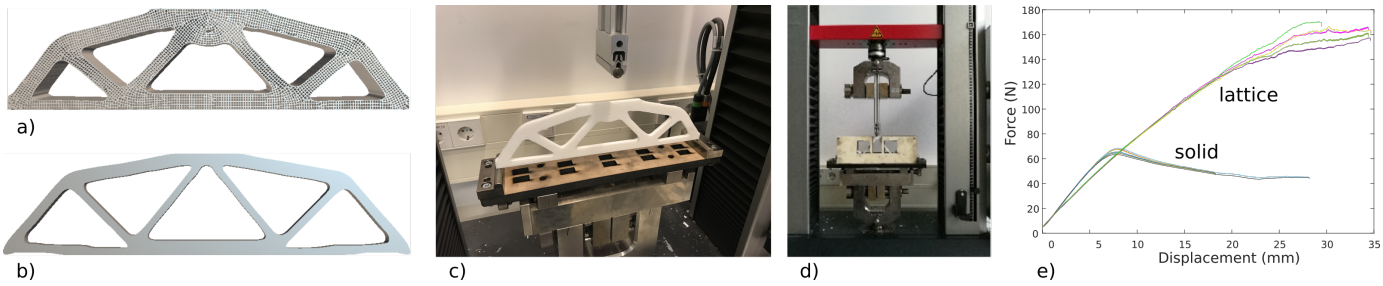


Fig. 15: Physical test comparing an optimized lattice structure (a) with an optimized solid structure (b). (c) and (d) show the experimental setup. From the force-displacement plots for multiple tests (e), it can be observed that, while the lattice structure is slightly less stiff, it supports a compressive force twice larger than the peak force supported by the solid counterpart.

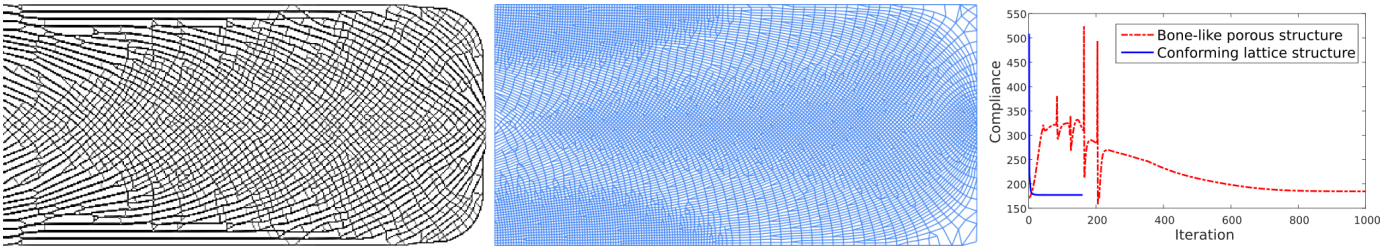


Fig. 16: Left: A bone-like porous structure generated by local volume constraints [12]. Middle: A conforming lattice structure generated by the proposed method. The lattice is stiffer, with a compliance of 177.29, compared to 184.64 of the porous structure. Right: Convergence in compliance for the optimizations.

[6] J. P. Groen and O. Sigmund, "Homogenization-based topology optimization for high-resolution manufacturable microstructures," *International Journal for Numerical Methods in Engineering*, vol. 113, no. 8, pp. 1148–1163, 2018.

[7] G. Allaire, P. Geoffroy-Donders, and O. Pantz, "Topology optimization of modulated and oriented periodic microstructures by the homogenization method," *Computers & Mathematics with Applications*, 2018.

[8] M. Attene, M. Livesu, S. Lefebvre, T. Funkhouser, S. Rusinkiewicz, S. Ellero, J. Martínez, and A. H. Bermano, "Design, representations, and processing for additive manufacturing," *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, vol. 10, no. 2, pp. 1–146, 2018.

[9] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, and X. Liu, "Cost-effective printing of 3d objects with skin-frame structures," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 177:1–177:10, Nov. 2013.

[10] L. Lu, A. Sharf, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D. Cohen-Or, and B. Chen, "Build-to-last: Strength to weight 3D printed objects," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 97:1–97:10, Jul. 2014.

[11] X. Zhang, Y. Xia, J. Wang, Z. Yang, C. Tu, and W. Wang, "Medial axis tree-an internal supporting structure for 3d printing," *Comput. Aided Geom. Des.*, vol. 35, no. C, pp. 149–162, May 2015.

[12] J. Wu, N. Aage, R. Westermann, and O. Sigmund, "Infill optimization for additive manufacturing – approaching bone-like porous structures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 2, pp. 1127–1140, February 2018.

[13] J. Martínez, J. Dumas, and S. Lefebvre, "Procedural voronoi foams for additive manufacturing," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 44:1–44:12, Jul. 2016.

[14] J. Martínez, H. Song, J. Dumas, and S. Lefebvre, "Orthotropic k-nearest foams for additive manufacturing," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 121:1–121:12, Jul. 2017.

[15] L. Xia and P. Breitkopf, "Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework," *Computer Methods in Applied Mechanics and Engineering*, vol. 278, pp. 524 – 542, 2014.

[16] O. Sigmund, N. Aage, and E. Andreassen, "On the (non-)optimality of michell structures," *Structural and Multidisciplinary Optimization*, vol. 54, no. 2, pp. 361–373, Aug 2016.

[17] G. Dong, Y. Tang, and Y. F. Zhao, "A survey of modeling of lattice structures fabricated by additive manufacturing," *Journal of Mechanical Design*, vol. 139, no. 10, p. 100906, 2017.

[18] F. Tamburrino, S. Graziosi, and M. Bordegoni, "The design process of additive manufactured meso-scale lattice structures: a review," *Journal of Computing and Information Science in Engineering*, vol. 18, no. 4, p. 040801, 2018.

[19] H. Wang, Y. Chen, and D. W. Rosen, "A hybrid geometric modeling method for large scale conformal cellular structures," in *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2005, pp. 421–427.

[20] J. Nguyen, S.-I. Park, D. W. Rosen, L. Folgar, and J. Williams, "Conformal lattice structure design and fabrication," in *Solid Freeform Fabrication Symposium, Austin, TX*, 2012, pp. 138–161.

[21] A. Michell, "The limits of economy of material in frame-structures," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 8, no. 47, pp. 589–597, 1904.

[22] T.-H. Kwok, Y. Li, and Y. Chen, "A structural topology design method based on principal stress line," *Computer-Aided Design*, vol. 80, pp. 19 – 31, 2016.

[23] K.-M. M. Tam, C. T. Mueller, J. R. Coleman, and N. W. Fine, "Stress line additive manufacturing (slam) for 2.5-d shells," *Journal of the International Association for Shell and Spatial Structures*, vol. 57, no. 4, pp. 249–259, 2016.

[24] M. Kilian, D. Pellis, J. Wallner, and H. Pottmann, "Material-minimizing forms and structures," *ACM Trans. Graphics*, vol. 36, no. 6, p. article 173, 2017.

[25] W. Li, A. Zheng, L. You, X. Yang, J. Zhang, and L. Liu, "Rib-reinforced shell structure," *Computer Graphics Forum (Proc. Pacific Graphics)*, vol. 36, no. 7, 2017.

[26] J. Smith, J. Hodgins, I. Oppenheim, and A. Witkin, "Creating models of truss structures with optimization," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 295–301, Jul. 2002.

[27] T. Zegard and G. H. Paulino, "Grand — ground structure based topology optimization for arbitrary 2d domains using matlab," *Structural and Multidisciplinary Optimization*, vol. 50, no. 5, pp. 861–882, Nov 2014.

[28] O. Stava, J. Vanek, B. Benes, N. Carr, and R. Měch, "Stress relief: Improving structural strength of 3d printable objects," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 48:1–48:11, Jul. 2012.

[29] Q. Zhou, J. Panetta, and D. Zorin, "Worst-case structural analysis," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 137:1–137:12, Jul. 2013.

[30] J. Panetta, A. Rahimian, and D. Zorin, "Worst-case stress relief for microstructures," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 122:1–122:16, Jul. 2017.

[31] T. Langlois, A. Shamir, D. Dror, W. Matusik, and D. I. W. Levin, "Stochastic structural analysis for context-aware design and fabrication," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 226:1–226:13, Nov. 2016.

[32] X. Chen, C. Zheng, W. Xu, and K. Zhou, "An asymptotic numerical method for inverse elastic shape design," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 95:1–95:11, Jul. 2014.

[33] E. Ulu, J. Mccann, and L. B. Kara, "Lightweight structure design under force location uncertainty," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 158:1–158:13, Jul. 2017.

[34] O. Sigmund, "A 99 line topology optimization code written in matlab," *Structural and multidisciplinary optimization*, vol. 21, no. 2, pp. 120–127, 2001.

[35] J. Wu, C. Dick, and R. Westermann, "A system for high-resolution topology optimization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 3, pp. 1195–1208, March 2016.

[36] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund, "Giga-voxel computational morphogenesis for structural design," *Nature*, vol. 550, no. 7674, p. 84, 2017.

[37] H. Liu, Y. Hu, B. Zhu, W. Matusik, and E. Sifakis, "Narrow-band topology optimization on a sparsely populated grid," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 251:1–251:14, Dec. 2018.

[38] N. Boddeti, Z. Ding, S. Kaijima, K. Maute, and M. L. Dunn, "Simultaneous digital design and additive manufacture of structures and materials," *Scientific reports*, vol. 8, no. 1, p. 15560, 2018.

[39] P. Geoffroy-Donders, G. Allaire, and O. Pantz, "3-d topology optimization of modulated and oriented periodic microstructures by the homogenization method," " 2018.

[40] S. Mueller, S. Im, S. Gurevich, A. Teibrich, L. Pfisterer, F. Guimbretière, and P. Baudisch, "Wireprint: 3d printed previews for fast prototyping," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '14. New York, NY, USA: ACM, 2014, pp. 273–280.

[41] R. Wu, H. Peng, F. Guimbretière, and S. Marschner, "Printing arbitrary meshes with a 5dof wireframe printer," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 101:1–101:9, Jul. 2016.

[42] Y. Huang, J. Zhang, X. Hu, G. Song, Z. Liu, L. Yu, and L. Liu, "Framefab: Robotic fabrication of frame shapes," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 224:1–224:11, Nov. 2016.

[43] R. Arora, A. Jacobson, T. R. Langlois, Y. Huang, C. Mueller, W. Matusik, A. Shamir, K. Singh, and D. I. Levin, "Designing volumetric truss structures," Oct. 2018, arXiv:1810.00706 [cs.GR].

[44] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin, "Quad-mesh generation and processing: A survey," *Comput. Graph. Forum*, vol. 32, no. 6, pp. 51–76, 2013.

[45] M. Nieser, U. Reitebuch, and K. Polthier, "Cubecover- parameterization of 3d volumes," *CGF*, vol. 30, no. 5, pp. 1397–1406, 2011.

[46] J. Huang, Y. Tong, H. Wei, and H. Bao, "Boundary aligned smooth 3d cross-frame field," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 143:1–143:8, Dec. 2011.

[47] Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo, "All-hex meshing using singularity-restricted field," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 177:1–177:11, Nov. 2012.

[48] T. Jiang, J. Huang, Y. T. Yuanzhen Wang, and H. Bao, "Frame field singularity correction for automatic hexahedralization," *IEEE TVCG*, vol. 20, no. 8, pp. 1189–1199, Aug. 2014.

[49] D. Sokolov, N. Ray, L. Untereiner, and B. Lévy, "Hexahedral-dominant meshing," *ACM Trans. Graph.*, vol. 36, no. 4, Jun. 2016.

[50] X. Gao, W. Jakob, M. Tarini, and D. Panozzo, "Robust hex-dominant mesh generation using field-guided polyhedral agglomeration," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 114:1–114:13, Jul. 2017.

[51] A. Vaxman, M. Campen, O. Diamanti, D. Panozzo, D. Bommes, K. Hildebrandt, and M. Ben-Chen, "Directional field synthesis, design, and processing," *Computer Graphics Forum*, vol. 35, no. 2, pp. 545–572, 2016.

[52] N. Ray, D. Sokolov, and B. Lévy, "Practical 3d frame field generation," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 233:1–233:9, Nov. 2016.

[53] M. Lyon, D. Bommes, and L. Kobbelt, "Hexex: Robust hexahedral mesh extraction," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 123:1–123:11, Jul. 2016.

[54] J. Solomon, A. Vaxman, and D. Bommes, "Boundary element octahedral fields in volumes," *ACM Trans. Graph.*, vol. 36, no. 3, May 2017.

[55] H. Liu, P. Zhang, E. Chien, J. Solomon, and D. Bommes, "Singularity-constrained octahedral fields for hexahedral meshing," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 93:1–93:17, Jul. 2018.

[56] N. Lei, X. Zheng, J. Jiang, Y.-Y. Lin, and D. X. Gu, "Quadrilateral and hexahedral mesh generation based on surface foliation theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 758 – 781, 2017.

[57] W. Jakob, M. Tarini, D. Panozzo, and O. Sorkine-Hornung, "Instant field-aligned meshes," *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA)*, vol. 34, no. 6, Nov. 2015.

[58] K.-J. Bathe, *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[59] E. Andreassen and C. S. Andreasen, "How to determine composite material properties using numerical homogenization," *Computational Materials Science*, vol. 83, pp. 488 – 495, 2014.

[60] G. Dong, Y. Tang, and Y. F. Zhao, "A 149 line homogenization code for three-dimensional cellular materials written in matlab," *Journal of Engineering Materials and Technology*, vol. 141, no. 1, p. 011005, 2019.

[61] K. Svanberg, "The method of moving asymptotesa new method for structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 359–373, 1987.

[62] F. Wang, B. S. Lazarov, and O. Sigmund, "On projection methods, convergence and robust formulations in topology optimization," *Structural and Multidisciplinary Optimization*, vol. 43, no. 6, pp. 767–784, Jun 2011.

[63] D. Bommes, H. Zimmer, and L. Kobbelt, "Mixed-integer quadrangulation," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 77:1–77:10, Jul. 2009.

[64] T. Jiang, X. Fang, J. Huang, H. Bao, Y. Tong, and M. Desbrun, "Frame field generation through metric customization," *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

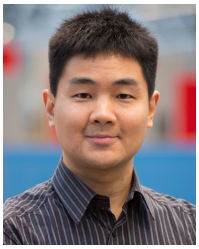
[65] G. Allaire, *Shape optimization by the homogenization method*. Springer Science & Business Media, 2002, vol. 146.

[66] O. Amir, N. Aage, and B. S. Lazarov, "On multigrid-cg for efficient topology optimization," *Structural and Multidisciplinary Optimization*, vol. 49, no. 5, pp. 815–829, May 2014.

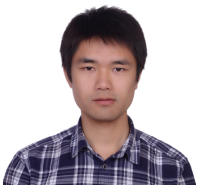
[67] E. Garner, H. M. Kolken, C. C. Wang, A. A. Zadpoor, and J. Wu, "Compatibility in microstructural optimization for additive manufacturing," *Additive Manufacturing*, vol. 26, pp. 65 – 75, 2019.

[68] A. Clausen, N. Aage, and O. Sigmund, "Exploiting additive manufacturing infill in topology optimization for improved buckling load," *Engineering*, vol. 2, no. 2, pp. 250 – 257, 2016.

[69] F. Ferrari and O. Sigmund, "Revisiting topology optimization with buckling constraints," *Structural and Multidisciplinary Optimization*, vol. 59, no. 5, pp. 1401–1415, May 2019.



Dr. Jun Wu is an assistant professor at the Department of Design Engineering, Delft University of Technology, the Netherlands. Before this, he was a Marie Curie postdoc fellow at the Department of Mechanical Engineering, Technical University of Denmark. He obtained a PhD in Computer Science in 2015 from TU Munich, Germany, and a PhD in Mechanical Engineering in 2012 from Beihang University, Beijing, China. His research is focused on computational design and digital fabrication, with an emphasis on topology optimization.



Dr. Weiming Wang is a Marie Curie postdoc fellow at the Department of Design Engineering, Delft University of Technology, the Netherlands, and a lecturer at the School of Mathematical Sciences, Dalian University of Technology, China. He obtained his PhD in the School of Mathematical Sciences in 2016 from Dalian University of Technology, China. His research is focused on digital fabrication and geometry processing.

Dr. Xifeng Gao is an assistant professor of the Computer Science Department at Florida State University. Dr. Gao was a PostDoc for two years at the Courant Institute of Mathematical Sciences of New York University. He received his Ph.D. degree in 2016 and won the best Ph.D. dissertation award from the Department of Computer Science at the University of Houston. Dr. Gao has wide research interests that are related to geometry, such as Computer Graphics, Visualization, Multimedia Processing, Robotics, and Digital Fabrication.