

# Self-Supporting Rhombic Infill Structures for Additive Manufacturing

Jun Wu<sup>\*1,4</sup>, Charlie C.L. Wang<sup>2,3</sup>, Xiaoting Zhang<sup>3</sup>, and Rüdiger Westermann<sup>4</sup>

<sup>1</sup>*Department of Mechanical Engineering, Technical University of Denmark, Denmark*

<sup>2</sup>*Department of Design Engineering, Delft University of Technology, The Netherlands*

<sup>3</sup>*Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, P. R. China*

<sup>4</sup>*Department of Informatics, Technische Universität München, Germany*

*\*Corresponding Author. Tel: +45 4525 4255. Fax: +45 4525 1961. Email: junwu@mek.dtu.dk*

## Abstract

Recent work has demonstrated that the interior material layout of a 3D model can be designed to make a fabricated replica satisfy application-specific demands on its physical properties, such as resistance to external loads. A widely used practice to fabricate such models is by layer-based additive manufacturing (AM), which however suffers from the problem of adding and removing interior supporting structures. In this paper, we present a novel method for generating application-specific infill structures on rhombic cells so that the resultant structures can automatically satisfy manufacturing requirements on overhang-angle and wall-thickness. Additional supporting structures can be avoided entirely in our framework. To achieve this, we introduce the usage of an adaptive rhombic grid, which is built from an input surface model. Starting from the initial sparse set of rhombic cells, via numerical optimization techniques an objective function can be improved by adaptively subdividing the rhombic grid and thus adding more walls in cells. We demonstrate the effectiveness of our method for generating interior designs in the applications of improving mechanical stiffness and static stability.

**Keywords:** infill, self-supporting, geometric algorithm, optimization, additive manufacturing

## 1. Introduction

The shape of 3D models can be designed by a variety of geometric modeling tools. To create a physical replica of such shapes, instead of manufacturing the object with a solid interior, a typical practice in AM is to use a *uniform* infill with an adjustable percentage of voids. The purpose is to balance the mechanical quality and the cost associated with material consumption and fabrication time. To serve specific purposes such as load bearing [1] or standing [2], a number of studies have been devoted to the generation of *optimized* interior.

However, in many (if not all) cases, the optimized structures cannot be directly fabricated by layer-based AM. Specifically, if the shapes have a relatively large overhang, the manufacturing material cannot be deposited on a layer where there is no material below it. Recent developments in AM address this problem by automatically generating support structures below the layers with overhang (e.g., [3, 4, 5]).

A support structure can be fabricated by either a dissolvable material or the same material as the main body. If a dissolvable material is employed, exterior supports can be removed automatically in a post-process (ref. [6]). The removal procedure has to be performed manually if single-material manufacturing is employed (e.g., SLA and the low-cost FDM machines<sup>1</sup>).

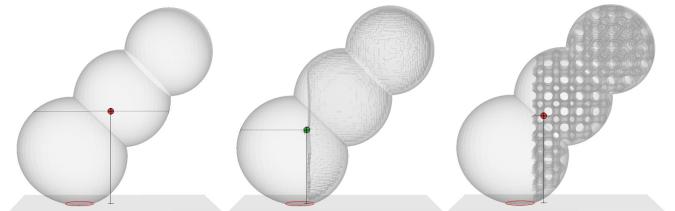


Figure 1: An optimized shape can be affected by support structures if they are considered separately: (a) an unbalanced model, (b) an optimized model that can stand, (c) the model becomes unbalanced again when support structures are added.

In some recent works, methods have been proposed to reduce the effort involved in the manual removal process by optimizing the direction of fabrication [7, 8, 9] and the shape of models [10, 11]. However, the existence of support structures becomes more problematic when an optimized structure has interior voids, because they cannot be removed at all. As a consequence, these additional material structures counteract the goal in shape optimization to create shapes with a particular physical property. For example, since additional mass is added, the result of mechanical stiffness optimization where the volume (and thus the total weight) is controlled becomes invalid [12]. Moreover, for an optimization targeting at static stability [2], additional interior support can make an optimized model lose its balance (see Fig. 1 for an example).

<sup>1</sup>SLA and FDM stands for *Stereolithography Apparatus* and *Fused Deposition Modeling*, which are two widely used approaches in AM.

Therefore, the effect of support structures must be considered simultaneously in the optimization process, or the need for such structures must be avoided entirely, for instance, by limiting the maximum allowed overhang during topology optimization (ref. [13]). This approach, however, cannot completely ensure that no overhang is generated. Moreover, adding the respective constraints into the optimization process increases the problem complexity and deteriorates the convergence of the employed numerical simulation schemes.

To overcome the aforementioned problem in interior structure optimization, we present a new method for infill optimization that ensures the resultant structures are *self-supporting*. A structure is said to be self-supporting if its surfaces have an overhang-angle smaller than a prescribed maximum overhang-angle, and thus the structure can be fabricated without adding supports below all its surfaces. Our method restricts the computational domain to a special class of spaces which are self-supporting and let control the size of the smallest features. Specifically, we advocate the use of rhombic cell structures as composites of the interior, where the maximum overhang-angle and the minimum wall-thickness can be specified explicitly. With these composites, we carry out an optimization process to gradually update the interior structures of a given model towards different specific usages, at the same time enforcing manufacturability of the results. The specific contributions of our work are as follows:

- We account for manufacturability explicitly by restricting topology optimization to generate structures living in a manufacturability-ensured space, i.e., represented by a hierarchical grid of rhombuses.
- We demonstrate the use of grid refinement and cell-to-shell operators to efficiently perform the optimization process.
- We demonstrate the effectiveness of our approach for the optimization of mechanical stiffness and static stability.

To the best of our knowledge, our proposed approach is the first that can ensure the manufacturability (i.e., bounds of the maximum overhang-angle and the minimum wall-thickness) of optimized interior structures.

The remainder of this paper is organized as follows. After reviewing work that is related to ours in Section 2, the overall methodology of our infill optimization framework is introduced in Section 3. The use of adaptive rhombic structures as the computational domain is discussed in Section 4. The formulation for optimizing the mechanical stiffness and the static stability is given in Section 5 to exemplify the use of our method. Results are shown in Section 6 to verify the effectiveness of our approach.

## 2. Related Work

With the development of additive manufacturing techniques in recent years, more and more research has been devoted to

geometric and physical modeling for AM (also called 3D printing). The purpose of this section is to discuss the strengths and limitations of approaches most closely related to ours, rather than providing a comprehensive survey. For the latter, let us refer to [14, 15].

**Interior Shape Optimization** The interior structure of a model can be optimized to meet different objectives on physical properties. The static balance has been considered in [2, 16], which is later extended to dynamic balance [17] by changing both the surface of an input model and its interior infills. Targeting a variety of stability objectives, Musalski et al. [18] optimize the interior by proposing a reduced-order parametrization of offset surfaces. Wu et al. [19] present an analysis of the optimal topology regarding static and rotational stability, and propose a reduced yet accurate optimization based on ray-reps. A heuristic method is developed in [20] to construct meso-structure inside a model to change its mechanical property. A skin-frame structure is optimized in [21] for enhancing the mechanical stiffness of objects fabricated by 3D printing. Lu et al. [1] compute the interior structure for a similar purpose but use different structures – honeycomb-cells. Among all these approaches, except [16] which fills the interior void by pre-defined regular patterns, interior structures generated via optimization faces the problem of large overhangs. As a result, the interior structure cannot be fabricated directly without support structures. We tackle this problem by using self-supporting rhombic structures in the optimization process.

**Support Structures during 3D Printing** Adding supports to a model leads to the problems of material waste, longer fabrication time and lower surface quality. Approaches have been developed to reduce the usage of support structure to resolve these problems. For example, an optimal printing direction is searched in [4] according to the total area of facing down regions (i.e., places where support structures need to be added). As a result, the total volume of the used support structure and, thus, additional material cost can be reduced. In the recent work of Zhang et al. [8], a perceptual model is developed using a training-and-learning strategy to consider multiple influences that will be given on a fabricated model by support structures. The best printing directions are determined by composing all the factors including contact area, visual saliency, viewpoint preference, and smoothness entropy. Given a fixed printing direction, the shape of an input model is optimized in [10, 11] to reduce the area of facing down regions so that less support structures are needed. Some approaches consider another aspect of supports – the stability. Unlike the tree structures used in [4, 8], bridge structures are suggested in [3, 5, 22]. Different from such techniques for reducing the usage of external supports, we optimize interior structures with the goal to ensure that they are self-supporting and thus can completely eliminate the usage of additional interior supports.

**Topology Optimization** Topology optimization has been widely used for product and structure design, and very compact code has been developed for well-posed problems (e.g., [23]). Recently, Wu et al. [24] have developed a high-throughput sys-

tem to improve the efficiency of topology optimization on 3D solids. Unlike the prior method which computes optimized shape and topology on a fixed structure of grids, Wang et al. [12] represent the structural boundary by a level-set model that is embedded in a scalar function of a higher dimension. The parametric control is realized in [25] by presenting level-sets of a higher-dimensional function derived from B-splines and combining primitives with R-functions. Different representations have been developed to control the domain of computation, including the B-spline space [26], the medial zone [27], and the dynamically changed simplicial complex [28]. Our research is related to these works in the sense that we follow the idea of restricting the computational domain to fully control geometric features. In particular, our framework computes on an adaptive rhombic grid to ensure the manufacturability constraints including self-support and a minimum thickness constraint for wall structures.

**Self-Supporting Structures** In computer graphics, the term “self-supporting” has been used in related-but-different contexts other than concerning the overhang-angle. Specifically, self-supporting in architectural geometry refers to structures which as a whole stand in static equilibrium without external support. The structure of self-supporting masonry was studied in [29, 30], and nonlinear optimization was used in [29] to approximate an arbitrary surface by such structures. Instead of assembling self-supporting structures, Whiting et al. [30] proposed an approach to guide the design process so that a self-supporting shape is formed. The self-supporting surface can also be constructed with the help of regular triangulation as studied in [31]. The assembly of building blocks to form a self-supporting structure is investigated in [32] to govern the process of construction. Recently, self-supporting shape is computed on parametric surfaces, which is a standard representation of freeform surfaces in CAD systems (ref. [33]). However, only the shape of a surface (but not topology) is optimized in most of these approaches.

### 3. Methodology

To ensure the manufacturability of infill structures, our method performs the computation using a hierarchical grid representation. In particular, we develop a tree structure with rhombic cells that is self-supporting and with controlled smallest feature sizes (see Section 4.1). Topology of the tree structure can be changed using adaptive spatial refinement similar to an octree, yet it subdivides rhombic cells.

Given a fabrication direction and the maximally allowed overhang-angle  $\alpha$ , the orientation and aspect ratio of the rhombic cells are first determined to initialize a root rhombus bounding the input model. Starting from this root rhombus, the grids are adaptively refined to construct a hierarchy so that every leaf node corresponds to a rhombus completely inside the given input model, as shown in Fig. 2. Details can be found in Section 4.3. By applying so called *carving* operations (Section 4.2) to each leaf node of the rhombic tree, the rhombic cell is converted into a rhombic *shell* with a given wall-thickness.

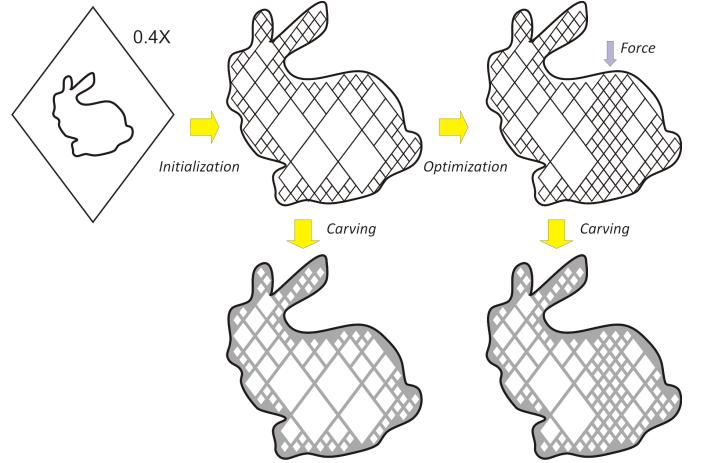


Figure 2: Overview of infill optimization using a rhombic structure. An initial model-covering rhombus (left), i.e., the root, is refined adaptively (top-middle) to fit the geometry of an input model. Only the leaf nodes completely inside a model are retained. The rhombic tree is then converted into a self-supporting structure formed by rhombic shells (bottom-middle). During the optimization for a specific physical objective, structures (and material) are added by refining recursively the rhombic cells represented by the affected leaf nodes (right).

The rhombic cells and their corresponding shells form the dual-representation employed in our optimization method. We subsequently call the composition of rhombic shells the *rhombic structure*. By setting appropriate geometric parameters, manufacturability of a rhombic structure can be guaranteed.

The optimization process for infills works on the rhombic structure by using a *splitting* operator to refine the rhombic grid adaptively and, thus, to reduce the value of the objective function. Applying the splitting operator to a rhombic cell results in eight cells with smaller size, yet the total volume and weight of the rhombic structure increases due to the added cells (Section 4.2). Starting from the given rhombic tree, an iterative optimization process is performed to modify the rhombic structure towards application-specific purposes. In each iteration, we perform a structural or stability analysis. Based on this analysis, the rhombic cells are subdivided adaptively using a greedy strategy. Details about this procedure are given in Section 5.

### 4. Adaptive Rhombic Structures

Given a polygonal mesh representing the boundary of a solid, we construct a set of rhombuses as composites of the object’s interior and exploit the geometry of these composites for satisfying manufacturing constraints. The representation of rhombic cells and shells, the associated geometric operators, and the construction method for the adaptive rhombic structure are presented in this section.

#### 4.1. Rhombic cells and shells

For the sake of simplicity, in the following we describe our proposed method in a 2D setting, yet the extension to 3D is straightforward and can be realized by extrusion.

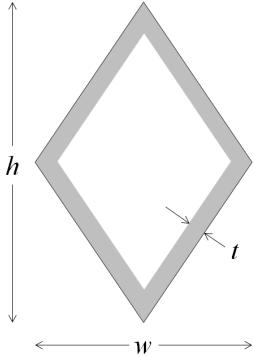


Figure 3: Illustration of the rhombic cell which is used as a self-supporting element in our computation. Manufacturability of this cell can be ensured by tuning the thickness  $t$  and the aspect ratio  $\frac{h}{w}$  according to different materials and machines in additive manufacturing.

A rhombus ( $\diamond$ ) is a parallelogram with four equal edges that can be represented uniquely by a tuple  $(w, h)$ , with  $w$  and  $h$  denoting the width and the height respectively. The geometry of a *hollowed* rhombic cell is referred to as a rhombic shell. It is represented by a triple  $(w, h, t)$ , where  $t$  indicates the thickness of the shell in 3D space. As shown in Fig. 3, the gray-shaded region is interpreted as solid (i.e., where material will be accumulated), and the white region inside the rhombic cell is a cavity. In our dual-representation, a rhombic grid stands for a lattice of rhombic cells with no thickness, while a rhombic structure is composed of a set of rhombic shells with thickness.

**Self-support** A rhombic shell is said to be *self-supporting* if it can be fabricated without adding support structures in the cavity. It is intuitive that the condition to ensure self-support of a rhombic shell is

$$\frac{h}{w} \geq \tan\left(\frac{\pi}{2} - \alpha\right), \quad (1)$$

where  $\alpha$  is the allowed maximum angle of overhangs, which is a device-dependent parameter taking different values depending on the used 3D printers.

**Wall-thickness** The wall-thickness of a rhombic structure is  $2t$  because each wall is shared by two adjacent cells. Therefore, according to the minimum plate thickness  $t_{\min}$  that can be fabricated by a particular 3D printer, the following condition must be satisfied to ensure the manufacturability of a rhombic structure:

$$t \geq \frac{1}{2}t_{\min}. \quad (2)$$

To ensure that the least possible material is added via the splitting operator and fine control over the material accumulation is achieved, the optimized structure should be as sparse as possible. To achieve this goal, we let  $\frac{h}{w} = \tan\left(\frac{\pi}{2} - \alpha\right)$  and  $t = \frac{t_{\min}}{2}$  when constructing the rhombic structures.

**Adaptivity** A rhombus can be divided by a regular 1:4 split, i.e., by inserting new vertices at the mid points of each edge and at the cell center (see Fig. 4). This results in four equally-sized smaller rhombuses. The new cells are self-supporting since they

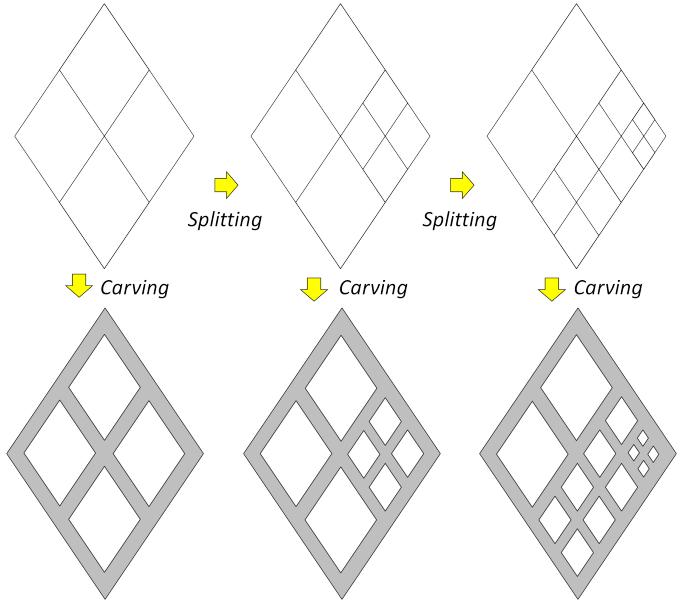


Figure 4: Illustration of adaptive rhombic subdivision and the dual rhombic representation: (top row) adaptively refined hierarchy of rhombic grids, and (bottom row) corresponding rhombic structures. The two geometric operators—carving and splitting—are also illustrated.

inherit the aspect ratio  $(\frac{h}{w})$  from the parent cell. Note that while the size of the new cells is halved along each dimension in every subdivision step, the same thickness is maintained to ensure manufacturability. As a consequence, the volume of solid increases as the rhombic cell is subdivided.

**2D-to-3D Extrusion** A 2D rhombic cell can be lifted to 3D by extruding the edges orthogonal to the 2D plane by a certain distance. We extrude about the length of a rhombus’s edge. Since the sparsity of the rhombic structure depends on the orientation of the cells, we construct initial structures with different orientations and choose the sparsest one.

#### 4.2. Geometric operators

Two geometric operators are developed for optimizing the infill on the rhombic structures: 1) carving and 2) splitting.

The carving operator is applied on the leaf nodes of the hierarchical grid. Given the geometry of a rhombus, it can be converted into a rhombic shell by adding the surface of its cavity, which is the offset of the rhombus about a fixed value  $t$ . The orientation of each facet must be inverted to represent a cavity. As a rhombic cavity is carved from a rhombus completely inside the input model  $M$ , there is no intersection between the shell’s surface and the surface of  $M$ . The final model with rhombic structures as infill can be represented by combining the facets of  $M$  and the facets of the rhombic cavities.

The splitting of a grid is realized by our method as an operator for optimization. Specifically, each 3D rhombus is uniformly split into eight sub-rhombuses with equal size. During the optimization process, a dual representation is kept for each instance. As a result, it is easy to convert from a rhombic grid into a rhombic structure to analyze physical properties (e.g.,

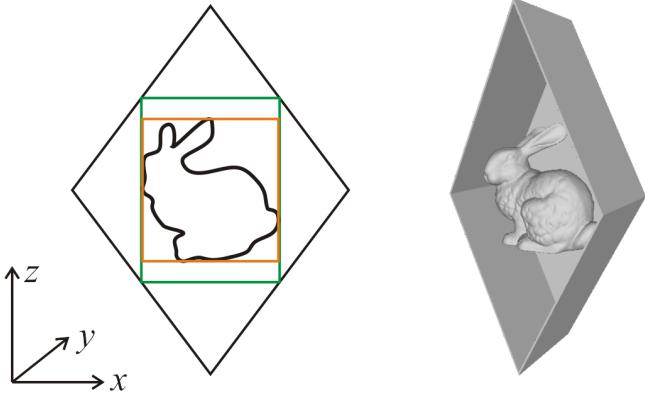


Figure 5: Initialization of the root cell. The bounding box of the input model (yellow) is fully contained within an inner box (green) within the spacial extent of the root cell. A 3D illustration is shown on the right.

stress and strain), and to further refine the grid to adhere to additional optimization constraints.

#### 4.3. Hierarchy construction

Given an input model  $\mathcal{M}$ , we construct an adaptive rhombic grid representing the interior of the model using the device depended parameters  $t_{\min}$  and  $\alpha$ . The idea is to first initialize a large rhombic cell covering the entire  $\mathcal{M}$  and satisfying the constraints of manufacturability. Using this cell as a root, the hierarchical rhombic grid structure can be constructed by recursive subdivision.

**Initialization** The root cell is positioned at the centroid of the bounding box of the input model. The height-to-width ratio is determined by the overhang-angle as  $h/w = \tan(\frac{\pi}{2} - \alpha)$ . The orthogonal extrusion distance is also set to  $w$ . To ensure that the root cell covers the entire model  $\mathcal{M}$ , we create the root cell by letting the bounding box of  $\mathcal{M}$  with dimensions  $(L_x, L_y, L_z)$  fit into an inner box of the rhombus (see the green one in Fig. 5). As the dimensions of the inner box are  $(w/2, w, h/2)$ , we should have

$$(w/2, w, h/2) \geq (L_x, L_y, L_z), \quad (3)$$

which leads to

$$w = \min \left\{ 2L_x, L_y, 2L_z / \tan\left(\frac{\pi}{2} - \alpha\right) \right\}. \quad (4)$$

**Recursive Subdivision** By applying the splitting operator, a rhombic cell on the  $i$ -th level is subdivided regularly into eight cells on level  $(i+1)$ . The resulting finer cells can be either partially or fully inside or outside  $\mathcal{M}$ . Cells fully outside  $\mathcal{M}$  are excluded from further computations. Those partially inside are further refined, and the subdivision stops at the cells which are fully inside  $\mathcal{M}$  and stored as leaf nodes of the tree. Moreover, the subdivision is also stopped when the distance between opposite faces in a rhombus is less than  $2t_{\min}$ , as no cavity can be formed that preserves the minimal thickness of the walls. This procedure results in fine cells along the boundary and coarse cells in the interior (see Fig. 2 for an example).

**Classification** A key geometric computation in the construction process is to determine the membership of a rhombus – whether it is *in*, *out* or *intersected* with  $\mathcal{M}$ . To facilitate such a classification, we rely on a high-resolution and regular Cartesian discretization of the domain. We build a distance field storing the signed distance from the centroid of each Cartesian grid cell (also known as a *voxel*) to  $\mathcal{M}$ , with a sign representing inside ('+') or outside ('-'). Based on this grid, a rhombus is classified as *inside* if all voxels it covers have a distance value larger than  $\tau$ , or as *outside* if all covered voxels have negative values. Otherwise, the rhombus is considered as intersecting the boundary zone of  $\mathcal{M}$ . To incorporate the width  $\varpi$  of voxels and the thickness  $t_{\text{bnd}}$  ( $t_{\text{bnd}} \geq t_{\min}$ ) of boundary hollowing, the value of  $\tau$  can be assigned as  $\tau = t_{\text{bnd}} + \frac{\sqrt{2}}{2} \varpi$ .

## 5. Infill Optimization

Starting from an adaptive rhombic grid, the optimizer iteratively modifies the grid by subdividing selected cells, based on the evaluation of a specific objective function. In particular, we demonstrate the optimization with respect to mechanical stiffness and static stability. The basic idea behind the optimization process is that subdividing a cell can improve its mechanical stiffness since more material is allocated, and can shift the center of gravity of the input model towards the centroid of the subdivided cell. Rather than subdividing the entire grid uniformly, we adaptively subdivide cells which are selected by a greedy strategy. We ensure that the subdivision does not violate the manufacturability of the 3D printed shapes, and the resulting model can be obtained without modifying the boundary mesh of the input model.

### 5.1. Mechanical stiffness

Given an initial rhombic grid, we want to determine whether a leaf cell should be subdivided. Encoding this as a Boolean-valued design variable  $\beta_c$  per leaf cell  $C$ , and discretizing the computational domain by a hexahedral finite element scheme, the optimization problem is formulated as follows,

$$\underset{\beta}{\text{minimize}} \quad E = \frac{1}{2} u^T K(\rho) u, \quad (5)$$

$$\text{subject to} \quad K(\rho) u = f, \quad (6)$$

$$V(\rho) = \sum_e \rho_e \leq V^*, \quad (7)$$

$$\rho_e(\beta) = \begin{cases} 1.0 & e \in \text{solid}, \\ \rho_{\min} & e \in \text{cavity}, \end{cases} \quad (8)$$

$$\beta_c \in \{0, 1\}, \forall C. \quad (9)$$

Here the objective is to minimize the strain energy  $E$  – i.e., which corresponds to the maximization of the stiffness. We use a finite element discretization and assemble the stiffness matrix  $K$  from elementary stiffness matrices  $K_e = \int_{\Omega_e} B^T D B dx$ , where  $\Omega_e$  is the domain of the finite element,  $B$  is the element strain matrix, and the linear material law is applied in  $D$ . The displacement vector  $u$  is calculated from the equilibrium equation

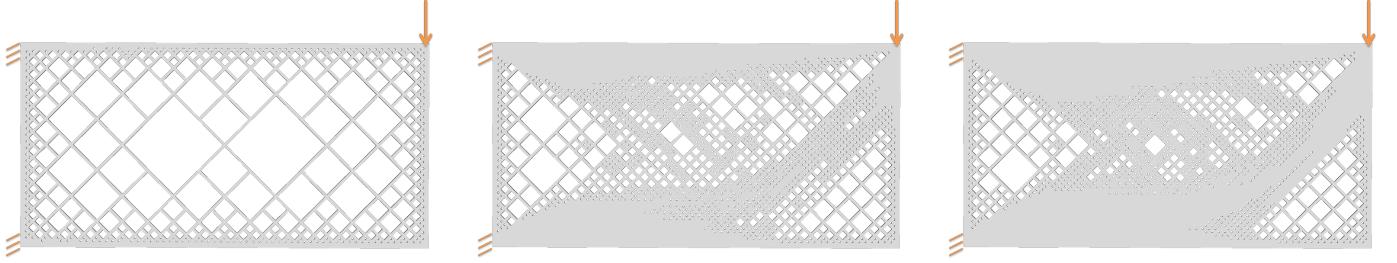


Figure 6: The initial rhombic structure (left) is iteratively optimized with respect to the given boundary conditions (yellow). The optimized structures with 70% volume and 78% volume are shown in the middle and right, respectively.

Eq.(6), which describes the static state of the object under the given external force  $f$ . The volume constraint Eq.(7) restricts the material consumption to a desired threshold  $V^*$ , measured in terms of the number of elements. In Eq.(8), hexahedral elements located in the cavities or in the solid region are distinguished by introducing an imaginary density value  $\rho_e$  per finite element  $e$ . A non-zero small constant  $\rho_{\min} = 0.001$  is prescribed for elements in the cavities in order to prevent the singularity of the global stiffness matrix  $K$ .

The optimization is taken as an iterative process. In each iteration, the following three steps are carried out sequentially:

1. Finite element analysis of elasticity (i.e., solve the state equation Eq.(6));
2. Evaluate the sensitivity;
3. Update the rhombic structure by subdividing selected cells.

During optimization, the material consumption gradually increases as a certain number of cells are subdivided in each iteration. The iterative optimization is thus terminated as soon as a further subdivision would lead to a volume exceeding the prescribed volume constraint Eq.(7). A 2D example of the optimization process is given in Fig. 6.

**Finite Element Analysis** We use a hexahedral finite element discretization in our analysis. To fully capture the thin walls in a hexahedral *Finite Element Analysis* (FEA), the discretization is conducted at a very high resolution so that thin walls can be represented by at least 3 voxels. This results in some millions of elements to be simulated by the optimizer. To this end, a memory-efficient multigrid solver as presented in [24] is employed. The elementary stiffness is expressed using the power-law relationship as  $K_e = (\rho_e)^p K_0$ , where  $p = 3$  is a penalization parameter [34], and  $K_0$  is the stiffness matrix of a solid element (i.e.,  $\rho_e = 1.0$ ).

**Sensitivity Analysis** With the displacement vector  $u$  computed by FEA, we can evaluate the functions in Eqs.(5) and (7). We are particularly interested in how much the subdivision of a leaf cell (i.e., change  $\beta_c$  from 0 to 1) affects the strain energy and the volume. Derivatives can be calculated by the chain rule as

$$\frac{\partial E}{\partial \beta_c} = \sum_e \frac{\partial E}{\partial \rho_e} \frac{\partial \rho_e}{\partial \beta_c} \quad \text{and} \quad \frac{\partial V}{\partial \beta_c} = \sum_e \frac{\partial V}{\partial \rho_e} \frac{\partial \rho_e}{\partial \beta_c},$$

with

$$\frac{\partial E}{\partial \rho_e} = -\frac{p}{2}(\rho_e)^{p-1} u_e^T K_0 u_e, \quad (10)$$

$$\frac{\partial V}{\partial \rho_e} = 1.0, \quad (11)$$

$$\frac{\partial \rho_e}{\partial \beta_c} = \begin{cases} 1.0 & \text{if } e \in \{\mathcal{S}_{\beta_c}\}, \\ 0.0 & \text{otherwise.} \end{cases} \quad (12)$$

Here  $\{\mathcal{S}_{\beta_c}\}$  represents the set of elements which become solid if the leaf cell  $C$  is subdivided.

We can then measure the sensitivity of strain energy with respect to the change of volume for each design variable as

$$G_c = \frac{-\partial E / \partial \beta_c}{\partial V / \partial \beta_c}. \quad (13)$$

This sensitivity, which is a positive value for compliance minimization, serves as an indicator of how much the strain energy will be decreased if the material consumption is increased due to the subdivision of the leaf cell  $C$ .

**Design Update** Based on the sensitivity, the optimizer subdivides the rhombic cells in a greedy manner, i.e., the element that leads to the largest reduction is refined first. To this end, we sort all cells by their sensitivity in descending order and the first 2% cells are subdivided successively in each iteration of the optimization process. After subdivision, new leaf cells are created which become design variables for the next iteration. Iteratively subdividing leaf cells leads to a model with strong mechanical stiffness while the total volume (and, thus, also the weight) is controlled. An example is given in Fig. 6 and more can be found in Section 6.

Concerning the granularity of volume change due to the subdivision of cells, let us consider a 3D model discretized by a resolution of  $256^3$ , and the subdivision of a relatively large rhombic cell of the size  $16^3$ . The induced volume increase of this cell subdivision is  $\frac{16^2 \times 3 \times 2}{256^3} < 0.01\%$ . This means that the granularity is very high.

## 5.2. Static stability

A shape stands if its center of gravity, when projected along the gravity direction, falls within the convex hull of its contact points with the ground. Our optimizer thus tries to shift the center of gravity into the convex hull.

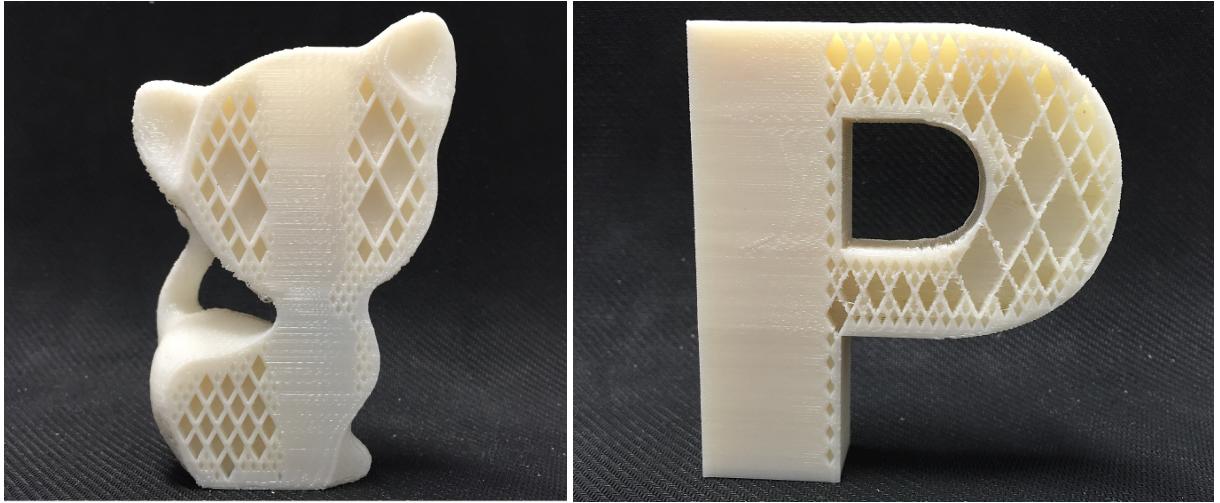


Figure 7: Two models with optimized infill were directly fabricated via AM without additional support structures. Left: A kitten model optimized for mechanical stiffness. Right: A model of the letter 'P' optimized for static stability.

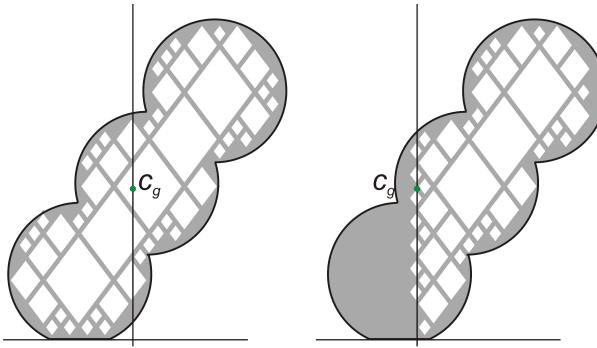


Figure 8: Left: The model would tip over since the projection of the center of gravity  $c_g$  does not fall into the convex hull of its contact points. Right: By subdividing cells on the left hand side of  $c_g$ ,  $c_g$  can be shifted leftwards to satisfy the static stability.

As illustrated in Fig. 8 for a 2D case, it is intuitive that subdividing cells on the left hand side of the gravity center  $c_g$  will shift it leftwards, i.e., closer to the convex hull of its contact points. This heuristic can be derived numerically. We aim to minimize the distance between the center of gravity  $c_g$  and the center of the convex hull  $c_h$ , projected along the direction of gravity,

$$\underset{\beta}{\text{minimize}} \quad E_s = \|(c_g - c_h)^{\perp g}\|_2^2, \quad (14)$$

$$\text{subject to} \quad \beta_c \in \{0, 1\}, \forall C. \quad (15)$$

The derivative of this objective function with respect to the design variable  $\beta_c$  is

$$\frac{\partial E_s}{\partial \beta_c} = \frac{2m_c}{m_c + m_g} (c_g - c_h)^{\perp g} \cdot (c_c - c_g)^{\perp g}, \quad (16)$$

where  $m_c$  represents the mass that is introduced by subdividing a cell under consideration,  $m_g$  is the mass of the current shape, and  $c_c$  denotes the geometric center of the cell under consideration. This derivative indicates that if the sign of the dot product

of  $(c_c - c_g)^{\perp g}$  and  $(c_g - c_h)^{\perp g}$  is negative, subdividing the cell decreases the distance and should thus be favored.

To update the rhombic structure, we sort the cells in ascending order of their derivatives (Eq.(16)), and sequentially subdivide those cells with small values until stability is reached. Cells with positive derivatives are also included in the subdivision process. While subdividing these cells does not improve the stability immediately, it enhances the flexibility (by generating more child cells) in the consequent subdivisions to better control the location of the center of mass.

Note that the center of the model,  $c_g$ , is updated incrementally after each subdivision. In principle, the derivative values should be immediately updated. However, since the change is very small, we follow the delayed update scheme similar to that proposed in [2], i.e., updating the derivatives and re-sorting the cells are performed after a certain number of subdivision iterations. In particular, we update after 2% of the cells have been subdivided. The effectiveness of this delayed update scheme has been validated by our tests.

## 6. Results and Discussion

Our proposed approach has been used for the optimization of several models with different geometric complexities. To demonstrate the manufacturability of the optimized shapes, we have fabricated some models using a consumer-level 3D printer, i.e., a MakerBot Replicator 2X using fused deposition modeling. Fig. 7 shows two of such models which were printed along the upright direction as prescribed for orienting the self-supporting rhombic structures. No additional inner supports were used during fabrication.

### 6.1. Mechanical stiffness

**Examples** Our first example is a kitten model as shown in the first row of Fig. 9. On the most left is the initial rhombic structure with an aspect ratio of  $h/w = 2.0$ , which consumes

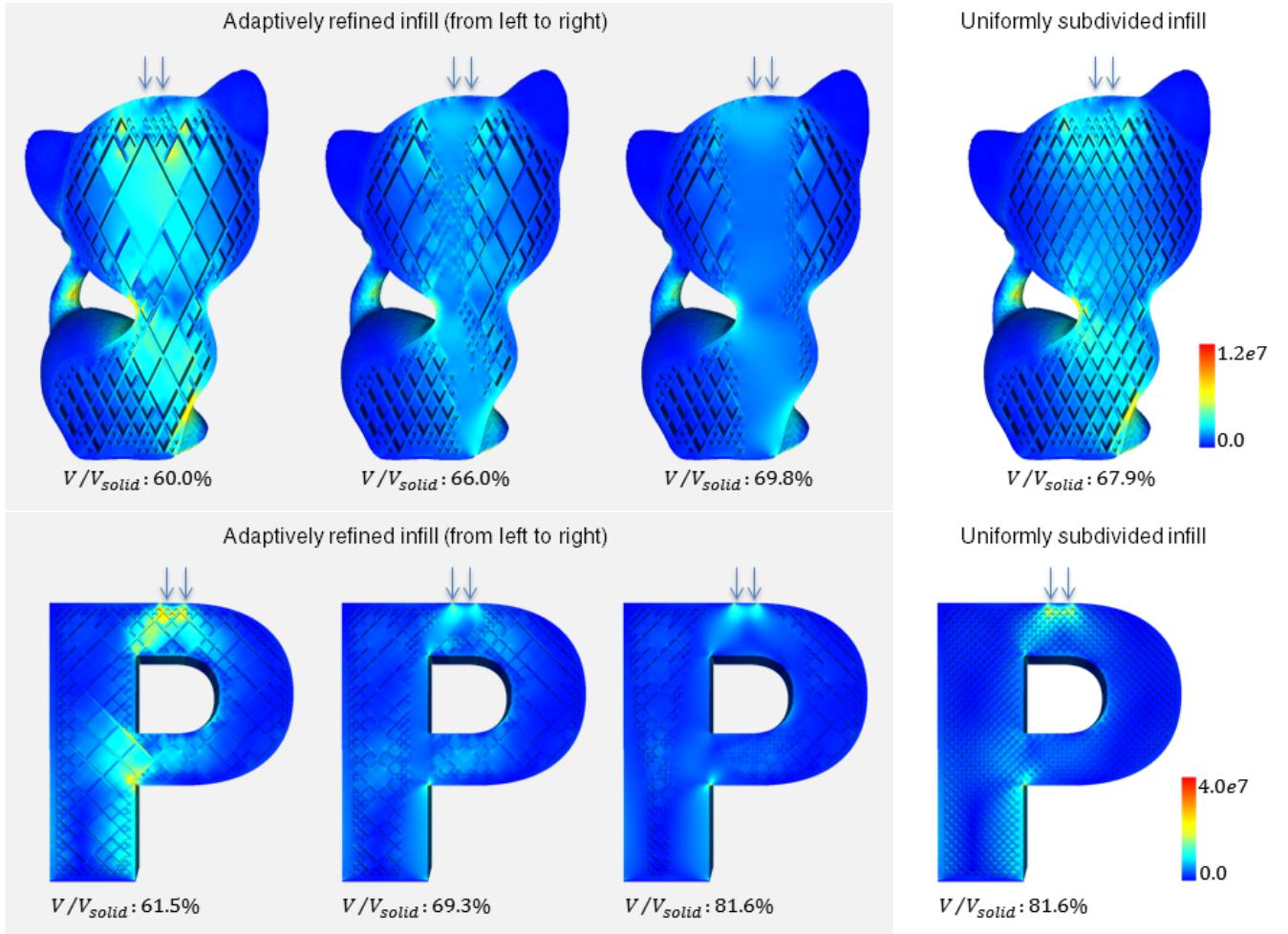


Figure 9: Comparison of the stress distribution in adaptively vs. uniformly refined rhombic structures on the kitten model (top row) and the letter ‘P’ model (bottom row). In the results of adaptive refinement, progressive results with increased volume are listed from left to right starting from the initial rhombic structures.

60.0% volume of a solid shape. From the stress distribution, it can be found that the external loads are transmitted from the top, where external forces are applied, to the bottom mostly through the neck. This initial structure is then adaptively subdivided by our method, which results in a gradual increase of volume consumption (see the middle left and the middle right of Fig. 9). The visualization of the stress distribution clearly shows the improvement of the overall mechanical stiffness due to the subdivision of cells with high stresses. For comparison, we also generate a uniformly refined rhombic structure, which is shown on the right of Fig. 9. Larger stress is observed on the model with uniformly refined structure. In other words, our adaptive refinement framework results in infill with better mechanical property.

The same test is also performed on a model of the letter ‘P’ (the second row of Fig. 9), where the rhombic cells have an aspect ratio of 1.0. Again, the optimized infill is mechanically stronger than a uniform infill.

Fig. 10 shows tests performed on the horse head model using rhombic structures with different aspect ratios. Fig. 10 (a) and (c) correspond to the initial rhombic structures with aspect ratio 1.0 and 2.0, respectively. The optimized structures (see Fig. 10

(b) and (d)) have reduced internal stress in both cases.

These tests are performed on a standard desktop PC equipped with an Intel Xeon X5560 processor running at 2.80 GHz, 8 GB of RAM, and an NVIDIA Tesla C2070 graphics card with 6 GB memory. The total optimization time for each model is less than 12 minutes, with the finite element analysis of the infill structure being the computational bottleneck. The finite element grids used for optimizing the kitten model, the letter ‘P’, and the horse head have a resolution of  $218 \times 198 \times 334$ ,  $219 \times 63 \times 251$ , and  $250 \times 179 \times 247$ , respectively.

**Parameter Analysis** We perform additional tests to analyze some parameters of the rhombic structure, and in particular the aspect ratio and the direction of extrusion. These parameters affect the sparsity of the initial structure and thus the corresponding optimization space.

The first test is performed on the kitten model by using different aspect ratios to construct the initial rhombic structures (i.e.,  $\alpha = 1.0, 2.0$ , and  $4.0$  respectively). As shown in the left of Fig. 11, the horizontal axis represents the percentage of volume w.r.t. a complete solid model, and the vertical axis refers to the compliance of a model with rhombic infill structures nor-

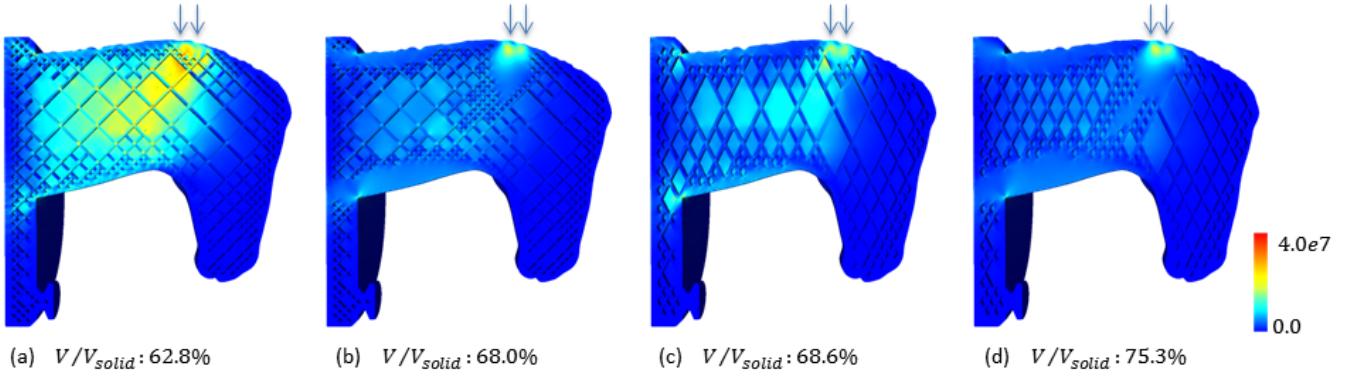


Figure 10: Comparison of the stress distribution in rhombic structures with different aspect ratios: (a) an initial structure with aspect ratio 1.0, (b) the optimized infill by adaptive refining (a), (c) an initial rhombic structure with aspect ratio 2.0, and (d) the optimized result of (c). In both cases, maximal stresses are reduced.

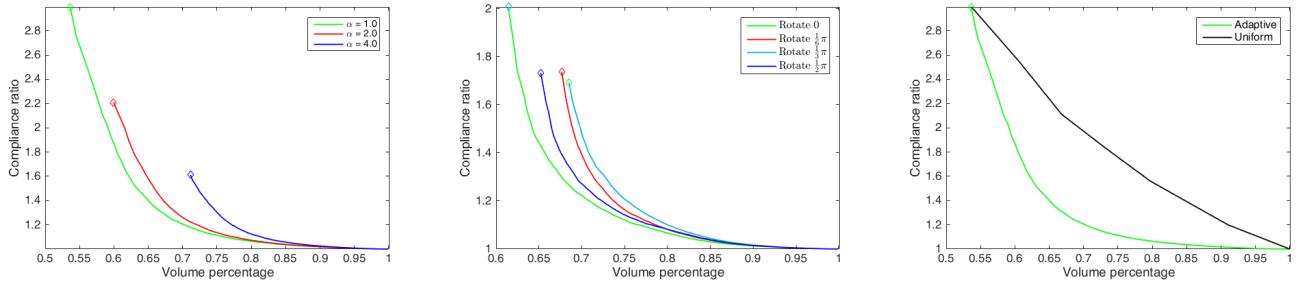


Figure 11: Study of convergence in computation: (left) tested on the kitten model when having different sparsity caused by using different aspect ratios and (middle) tested on the letter ‘P’ model with different sparsity according to different orientation of rhombic cells. Comparison of convergence using adaptive vs. uniform refinement on the kitten model is also given in the right.

malized over the compliance of a fully solid model. The initial structures are indicated by a  $\diamond$  symbol in the plot. The ending structure of refinement in all cases is the same fully solid model. The initial structures using different aspect ratios consume different amounts of volume with  $\alpha = 1.0$  consuming the least and  $\alpha = 4.0$  consuming the most amount. At a particular volume percentage on the horizontal axis (for instance at  $V/V_{solid} = 0.75$ ) the green curve (with  $\alpha = 1.0$ ) is lower than both the red curve (with  $\alpha = 2.0$ ) and the blue curve (with  $\alpha = 4.0$ ). This verifies that, when constraining the volume of material usage (i.e., letting  $V/V_{solid} = \text{constant}$ ), our framework results in a better structure by using an aspect ratio close to 1.0 (i.e., with a more sparse initial structure).

We then study the behavior of optimization when extruding rhombic structures along different orientations on the letter ‘P’ model. The input model is rotated along the  $z$ -axis before it is loaded into the optimization routine. The four initial structures consume different amounts of volume (see the middle of Fig. 11, where the initial status is indicated by the  $\diamond$  symbols). Similar to the previous test on different aspect ratios, the ending structure is the same solid. At a particular volume percentage (e.g.,  $V/V_{solid} = 0.75$ ), the curve of computation starting from the most sparse structure has the smallest compliance ratio while the curves starting from denser structures monotonically have larger values in compliance.

The above study verifies our heuristic that a sparser initial structure (i.e., a smaller initial volume consumption) should be

favored, since the remaining optimization space is larger and the optimized compliance is usually smaller when using the same volume of materials.

The progress of the iterative process to reach a prescribed volume is affected by the number of elements subdivided in each iteration. In general, subdividing a smaller number of elements in each iteration leads to structures with a smaller compliance, at the cost of more iterations. Tests are conducted on the kitten model by subdividing 5%, 2% and 1% elements in each iteration respectively. The initial rhombic structure with an aspect ratio of 1.0 consumes a volume of 53.7%. The process is terminated when a further subdivision would result in a volume exceeding 65.0%. The number of iterations to reach this limit is 10, 26, and 52, respectively, resulting in a relative compliance of 1.424, 1.407, and 1.406, respectively. While this parameter can be specified by the user, we select 2% in our other experiments by considering the trade-off between speed and quality.

**Adaptive or Uniform Refinement** We compare the compliance of adaptively refined rhombic structures with uniform rhombic structures in the right of Fig. 11. The test is performed on the kitten model with an aspect ratio of 1.0. The two curves start from the same initial structure. As more materials are allocated, both the adaptive refinement suggested by the optimization and the uniform refinement can reduce the compliance. However, compliance on the adaptively subdivided structure (the green curve) drops much faster than that of the

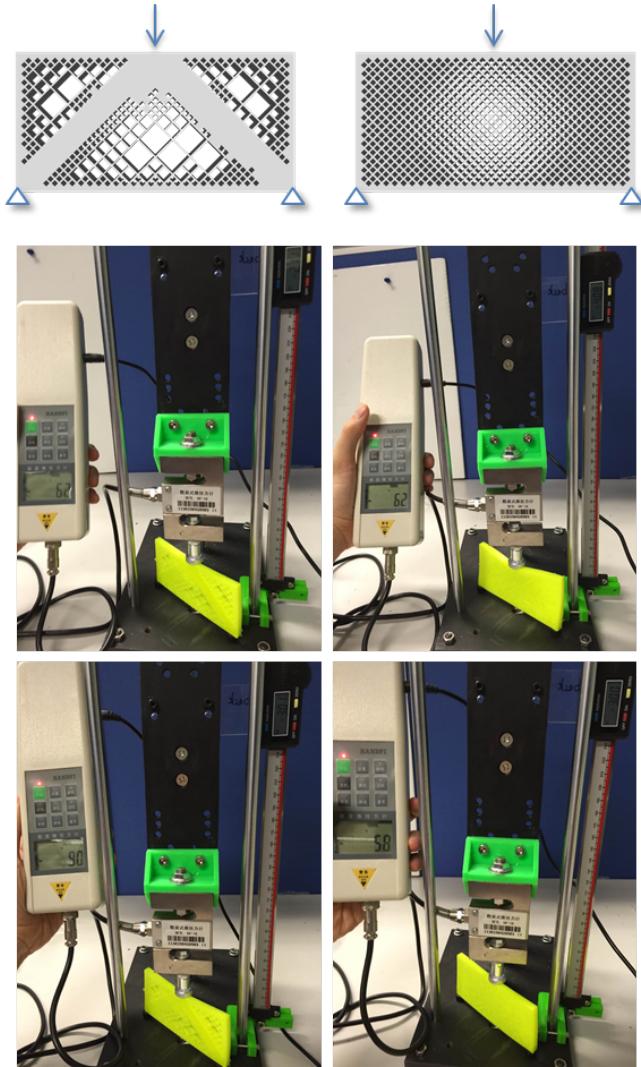


Figure 12: First row: The adaptively refined infill (left) and the uniform infill (right) consume the same amount of material. Second row: Under the same applied force of  $62\text{ N}$ , the optimized infill deforms by  $2.11\text{ mm}$ , while the uniform one has a larger deformation of  $4.08\text{ mm}$ . Third row: When both deform by  $3\text{ mm}$ , the optimized infill supports a force of  $90\text{ N}$ , while the uniform one supports a smaller force of  $58\text{ N}$ .

uniformly subdivided structure (the black curve). As a result, the optimized structure generated by adaptive refinement has a much smaller compliance when having the same value of volume percentage on the horizontal axis.

Two specific instances are shown on the top row in Fig. 9: The adaptively refined and the uniformly refined structures in the kitten model. The optimized structure with a volume percentage of  $V/V_{solid} = 69.8\%$  has a normalized compliance of 1.21, which is much smaller than a compliance of 2.06 on the uniformly refined structure with a close volume percentage of  $V/V_{solid} = 67.9\%$ . The comparison is confirmed as well on the letter 'P' model (Fig. 9 the bottom row).

We further perform physical tests on a beam model. As shown in Fig. 12 (first row), the beam is supported on its bottom at two ends while a downward force is applied on its top in the middle. These two models consume the same amount of mate-

rial. Numerical analysis suggests that the optimized infill (left) has a normalized compliance of 1.25, smaller than 1.94 for the uniform infill (right). The models have a physical dimension of  $100\text{ mm} \times 50\text{ mm} \times 6.25\text{ mm}$ , printed with PLA flexible material on a WiseMaker Pro W600 3D printer. The printing time is comparable – 5.4 hours for the optimized infill and 5.48 hours for the uniform one.

The physical tests are presented in the second and third row of Fig. 12. When the same amount of force ( $62\text{ N}$ ) is applied, the optimized version has a displacement of  $2.11\text{ mm}$  which is almost only one half of  $4.08\text{ mm}$  – the displacement on the uniform version (see the second row of Fig. 12). This difference can be clearly observed from the image. In the third row, we deform both versions by  $3\text{ mm}$ . The optimized version supports a force of  $90\text{ N}$ , which is much larger than  $58\text{ N}$  for the uniform one. These physical tests verify that the optimized infill generated by our method is much stiffer than the uniformed infill.

## 6.2. Static stability

**Examples** Models optimized for standing are collectively shown in Fig. 13. The optimization of each model takes less than one minute. The static stability shown here is verified by using MeshMixer [35] from Autodesk.

## 6.3. Discussion and limitations

**Optimality** We use a greedy strategy to select cells to refine, and thus in theory the approach does not guarantee the global optimality of results, especially in the nonlinear compliance optimization problem. However, given the fact that

- the number of design variables is relatively large, and
- the set of design variables changes during iterations as new cells are created,

an exhaustive search over all possible refinements is impractical. In this sense, our approach provides a reasonable trade-off between optimal and practical.

To infer how close the greedy strategy leads to an optimal solution, we compare the adaptive refinement strategy with a strategy which allows both refinement and coarsening. The bidirectional updating strategy is akin to the voxel-wise discrete optimization method [36], which is renamed as *Sequential Element Rejection and Admissions* by Rozvany et al. in [37]. A comparison of these two different updating schemes is shown in Fig. 14, which is also tested on the kitten model for compliance minimization. In both schemes, we start with a sparse initial rhombic grid and update it until it reaches the configuration of fully solid, which is similar to the tests shown in Fig. 11. The green curve refers to the implementation that only 4% cells having the largest strain energy are refined in each iteration. The red curve refers to an implementation that 4% cells with largest strain energy are refined while another 1% cells with smallest strain energy are coarsened by removing their corresponding walls. In our experiments, the difference between these two curves is rather small. Although the bidirectional strategy does introduce results with slight lower compliance, the scheme with

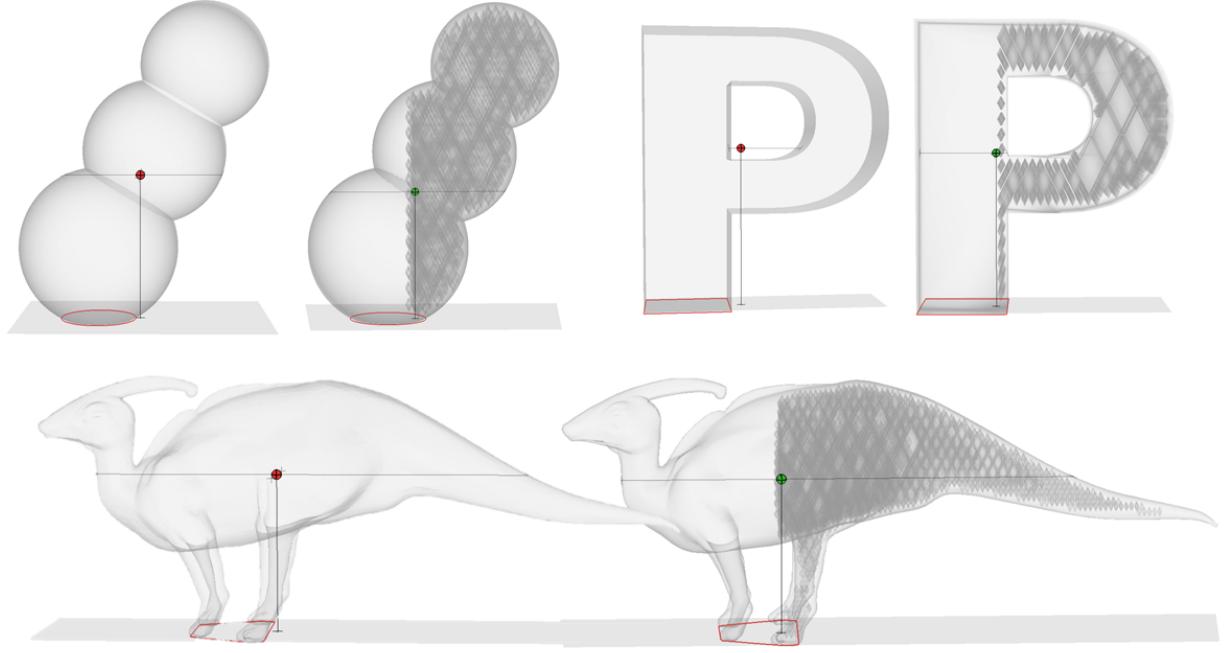


Figure 13: On the left of each group, the initial solid shape would fall over since the projection of its center of mass (red dot) is not located within the convex hull of the contact points (red contour). The right hand side of each group shows the optimized shape, the center of which (green dot) is shifted to a stable configuration.

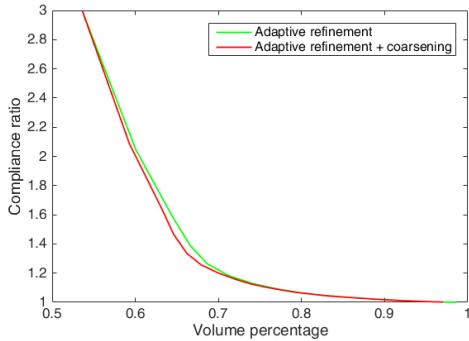


Figure 14: Compliance comparison between the refinement only strategy (in green curve) and the hybrid strategy of both refinement and coarsening (in red curve). The difference between these two schemes is small.

only refinement has less parameters involved and is easier to implement.

**Alternative Design Variables** The optimization problem builds upon a uniform 1:8 subdivision of the leaf cells as design variable. Alternative design variables are possible, for instance, the thickness of the rhombic walls. However, it is unclear to us based on which conditions a subdivision or an increase of thickness should be favored. To this end, it would be necessary to employ optimization algorithms which take different types of design variables into account. Such algorithms have been investigated by Wang et al. [21] for truss optimization.

**Design Space Limitation** Our solution space is limited by the specific subdivision of rhombic cells. Better solutions which

are outside of this solution space cannot be reached. To enlarge the solution space, it is interesting to explore alternative structures exhibiting sparsity and self-support, as well as corresponding update rules.

## 7. Conclusion

In this paper we have presented a new method for infill optimization taking into account the manufacturability of the generated shapes. The overhang constraint is seamlessly satisfied by using rhombic structures as infill so that their self-support is exploited. Our method has been validated in a number of experiments with respect to mechanical stiffness and static stability.

We believe this work opens interesting directions in the area of geometric and physical modeling for 3D printing. As future work, we are particularly interested in extending this work from infill optimization to design domains without a fixed boundary. Furthermore, finding more sparse self-supporting structures will make the method even more effective. Last but not the least, while currently we consider compliance minimization, it is nevertheless relevant in many engineering applications to consider constraints on the maximum stress [38].

## Acknowledgement

We would like to thank the anonymous reviewers for their constructive suggestions, and Zihao Wu for preparing the physically fabricated models. J. Wu is supported by the H.C. Ørsted Postdoc Programme at the Technical University of Denmark, which has received funding from the People Programme (Marie Curie Actions) of the European Union’s Seventh Framework

Programme (FP7/2007-2013) under REA grant agreement no. 609405 (COFUNDPostdocDTU). C.C.L. Wang and X. Zhang were partially supported by Hong Kong RGC General Research Fund (14207414) and Hong Kong ITC Innovation and Technology Fund (ITS/065/14 and ITS/060/13). J. Wu and R. Westermann were partially supported by the European Union under the ERC Advanced Grant 291372 SaferVis – Uncertainty Visualization for Reliable Data Discovery.

## References

- [1] L. Lu, A. Sharf, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D. Cohen-Or, B. Chen, Build-to-last: Strength to weight 3D printed objects, *ACM Trans. Graph.* 33 (4) (2014) 97:1–97:10. doi:10.1145/2601097.2601168.
- [2] R. Prévost, E. Whiting, S. Lefebvre, O. Sorkine-Hornung, Make it stand: Balancing shapes for 3d fabrication, *ACM Trans. Graph.* 32 (4) (2013) 81:1–81:10. doi:10.1145/2461912.2461957.
- [3] P. Huang, C. C. L. Wang, Y. Chen, Algorithms for layered manufacturing in image space, in: ASME Advances in Computers and Information in Engineering Research, 2014, pp. 377–410. doi:10.1115/1.860328\_ch15.
- [4] J. Vanek, J. A. G. Galicia, B. Benes, Clever support: Efficient support structure generation for digital fabrication, *Comput. Graph. Forum* 33 (5) (2014) 117–125. doi:10.1111/cgf.12437.
- [5] J. Dumas, J. Hergel, S. Lefebvre, Bridging the gap: Automated steady scaffolds for 3d printing, *ACM Trans. Graph.* 33 (4) (2014) 98:1–98:10. doi:10.1145/2601097.2601153.
- [6] K. G. Swift, J. D. Booker, Manufacturing Process Selection Handbook, Elsevier Ltd., 2013.
- [7] P.-T. Lan, S.-Y. Chou, L.-L. Chen, D. Gemmill, Determining fabrication orientations for rapid prototyping with stereolithography apparatus, *Computer-Aided Design* 29 (1) (1997) 53 – 62. doi:10.1016/S0010-4485(96)00049-8.
- [8] X. Zhang, X. Le, A. Panotopoulou, E. Whiting, C. C. L. Wang, Perceptual models of preference in 3d printing direction, *ACM Trans. Graph.* 34 (6) (2015) 215:1–215:12. doi:10.1145/2816795.2818121.
- [9] N. Umetani, R. Schmidt, Cross-sectional structural analysis for 3d printing optimization, in: SIGGRAPH Asia 2013 Technical Briefs, 2013, pp. 5:1–5:4. doi:10.1145/2542355.2542361.
- [10] K. Hu, S. Jin, C. C. Wang, Support slimming for single material based additive manufacturing, *Computer-Aided Design* 65 (2015) 1–10. doi:10.1016/j.cad.2015.03.001.
- [11] K. Hu, X. Zhang, C. Wang, Direct computation of minimal rotation for support slimming, in: Automation Science and Engineering (CASE), 2015 IEEE International Conference on, 2015, pp. 936–941. doi:10.1109/CoASE.2015.7294219.
- [12] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, *Computer Methods in Applied Mechanics and Engineering* 192 (2003) 227–246. doi:10.1016/S0045-7825(02)00559-5.
- [13] A. T. Gaynor, J. K. Guest, Topology optimization for additive manufacturing: Considering maximum overhang constraint, in: 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2014. doi:10.2514/6.2014-2036.
- [14] L. Liu, A. Shamir, C. Wang, E. Whiting, 3D printing oriented design: Geometry and optimization, in: SIGGRAPH Asia 2014 Courses, 2014. doi:10.1145/2659467.2675050.
- [15] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. Wang, Y. C. Shin, S. Zhang, P. D. Zavattieri, The status, challenges, and future of additive manufacturing in engineering, *Computer-Aided Design* 69 (2015) 65–89. doi:10.1016/j.cad.2015.04.001.
- [16] A. N. Christiansen, R. Schmidt, J. A. Bærentzen, Automatic balancing of 3D models, *Computer-Aided Design* 58 (2015) 236–241, Solid and Physical Modeling 2014. doi:10.1016/j.cad.2014.07.009.
- [17] M. Bächer, E. Whiting, B. Bickel, O. Sorkine-Hornung, Spin-it: Optimizing moment of inertia for spinnable objects, *ACM Trans. Graph.* 33 (4) (2014) 96:1–96:10. doi:10.1145/2601097.2601157.
- [18] P. Musalski, T. Auzinger, M. Birsak, M. Wimmer, L. Kobbelt, Reduced-order shape optimization using offset surfaces, *ACM Trans. Graph.* 34 (4) (2015) 102:1–102:9. doi:10.1145/2766955.
- [19] J. Wu, L. Kramer, R. Westermann, Shape interior modeling and mass property optimization using ray-reps, *Computers & Graphics* 58 (2016) 66 – 72, Shape Modeling International 2016. doi:10.1016/j.cag.2016.05.003.
- [20] Y. Chen, S. Wang, Computer-aided product design with performance-tailored mesostructures, *Computer-aided Design and Application* 5 (1-4) (2008) 565–576. doi:10.3722/cadaps.2008.565–576.
- [21] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, X. Liu, Cost-effective printing of 3d objects with skin-frame structures, *ACM Trans. Graph.* 32 (6) (2013) 177:1–177:10. doi:10.1145/2508363.2508382.
- [22] Y. Chen, K. Li, X. Qian, Direct geometry processing for tele-fabrication, *ASME Journal of Computing and Information Science in Engineering* 13 (2013) 041002. doi:10.1115/1.4024912.
- [23] O. Sigmund, A 99 line topology optimization code written in matlab, *Struct. Multidiscip. Optim.* 21 (2) (2001) 120–127. doi:10.1007/s001580050176.
- [24] J. Wu, C. Dick, R. Westermann, A system for high-resolution topology optimization, *IEEE Transactions on Visualization and Computer Graphics* 22 (3) (2016) 1195–1208. doi:10.1109/TVCG.2015.2502588.
- [25] J. Chen, V. Shapiro, K. Suresh, I. Tsukanov, Shape optimization with topological changes and parametric control, *International Journal for Numerical Methods in Engineering* 71 (3) (2007) 313–346. doi:10.1002/nme.1943.
- [26] X. Qian, Topology optimization in b-spline space, *Computer Methods in Applied Mechanics and Engineering* 265 (2013) 15–35. doi:10.1016/j.cma.2013.06.001.
- [27] A. A. Eftekharian, H. T. Ilies, Shape and topology optimization with medial zones, in: ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2011, pp. 687–696.
- [28] A. N. Christiansen, J. A. Bærentzen, M. Nobel-Jørgensen, N. Aage, O. Sigmund, Combined shape and topology optimization of 3D structures, *Computers & Graphics* 46 (2015) 25–35, Shape Modeling International 2014. doi:10.1016/j.cag.2014.09.021.
- [29] E. Vouga, M. Höbinger, J. Wallner, H. Pottmann, Design of self-supporting surfaces, *ACM Trans. Graph.* 31 (4) (2012) 87:1–87:11. doi:10.1145/2185520.2185583.
- [30] E. Whiting, H. Shin, R. Wang, J. Ochsendorf, F. Durand, Structural optimization of 3d masonry buildings, *ACM Trans. Graph.* 31 (6) (2012) 159:1–159:11. doi:10.1145/2366145.2366178.
- [31] Y. Liu, H. Pan, J. Snyder, W. Wang, B. Guo, Computing self-supporting surfaces by regular triangulation, *ACM Trans. Graph.* 32 (4) (2013) 92:1–92:10. doi:10.1145/2461912.2461927.
- [32] M. Deuss, D. Panozzo, E. Whiting, Y. Liu, P. Block, O. Sorkine-Hornung, M. Pauly, Assembling self-supporting structures, *ACM Trans. Graph.* 33 (6) (2014) 214:1–214:10. doi:10.1145/2661229.2661266.
- [33] M. Miki, T. Igarashi, P. Block, Parametric self-supporting surfaces via direct computation of airy stress functions, *ACM Trans. Graph.* 34 (4) (2015) 89:1–89:12. doi:10.1145/2766888.
- [34] M. P. Bendsøe, O. Sigmund, Material interpolation schemes in topology optimization, *Archive of Applied Mechanics* 69 (9) (1999) 635–654. doi:10.1007/s004190050248.
- [35] R. Schmidt, K. Singh, Meshmixer: An interface for rapid mesh composition, in: ACM SIGGRAPH 2010 Talks, ACM, New York, NY, USA, 2010, pp. 6:1–6:1. doi:10.1145/1837026.1837034.
- [36] Y. Xie, G. Steven, A simple evolutionary procedure for structural optimization, *Computers & Structures* 49 (5) (1993) 885 – 896. doi:10.1016/0045-7949(93)90035-C.
- [37] G. I. N. Rozvany, A critical review of established methods of structural topology optimization, *Structural and Multidisciplinary Optimization* 37 (3) (2008) 217–237. doi:10.1007/s00158-007-0217-0.
- [38] K. Suresh, M. Takalloozadeh, Stress-constrained topology optimization: A topological level-set approach, *Struct. Multidiscip. Optim.* 48 (2) (2013) 295–309. doi:10.1007/s00158-013-0899-4.