# Automated generation of flat tileable patterns and 3D reduced model simulation

Iason Manolas[a,b], Francesco Laccone[b], Gianmarco Cherchi[c,*], Luigi Malomo[b], Paolo Cignoni[b]

[a] *ISTI-CNR, Pisa, Italy*
[b] *University of Pisa, Italy*
[c] *University of Cagliari, Italy*

A B S T R A C T

The computational fabrication community is developing an increasing interest in the use of patterned surfaces, which can be designed to show ornamental and unconventional aesthetics or to perform as a proper structural material with a wide range of features. Geometrically designing and controlling the deformation capabilities of these patterns in response to external stimuli is a complex task due to the large number of variables involved. This paper introduces a method for generating sets of tileable and exchangeable flat patterns as well as a model-reduction strategy that enables their mechanical simulation at interactive rates. This method is included in a design pipeline that aims to turn any general flat surface into a pattern tessellation, which is able to deform under a given loading scenario. To validate our approach, we apply it to different contexts, including real-scale 3D printed specimens, for which we compare our results with the ones provided by a ground-truth solver.

## 1. Introduction

With the increasing availability of CNC machines and 3D printers, the fabrication of physical artifacts and their visual appearance have become trending research topics in the Computer Graphics community. In recent years, several workflows have been developed to streamline the digital fabrication process to overcome material, size, and geometric limitations, and to speed up the reproduction and the prototyping phase [1]. In addition to high-resolution reproductions, new approaches based on stylized fabrication techniques have acquired attention. Hence, the objects are realized in an artistic manner [2] using a broad spectrum of techniques to reinterpret the shape but preserve its main geometric features. The use of surface segmentation and patterns is very common since a target geometry is often not reproducible in the real world and needs to be decomposed or simplified due to material or manufacturing constraints. Practically, the raw material is always limited in size and shape; and the manufacturing tools can usually perform some operations in a prescribed space and adopt specific tolerances. Fabrication-aware methods include these constraints in the earliest phases of geometry definition.

It quickly became apparent that these techniques could also be directed towards the production of objects that look or perform in a desired way, e.g., when subject to a particular external stimulus (force, deformation, heat, light etc.). A class of objects exploiting geometry abstraction to achieve a specific response to an external stimulus makes use of the so-called mechanical metamaterials.

Mechanical metamaterials are artificial structures that rely on a specific internal organization of their elements to achieve a required behavior in terms of deformation, stress or energy [3]. While in general the material's structure at the micro- or nano-scale usually determines the macroscopic behavior of an element, a metamaterial is instead governed by its geometry for a given constituting material. This feature has expanded the design space provided by traditional materials and elements, and

*Corresponding Author:
*e-mail:* g.cherchi@unica.it (Gianmarco Cherchi)

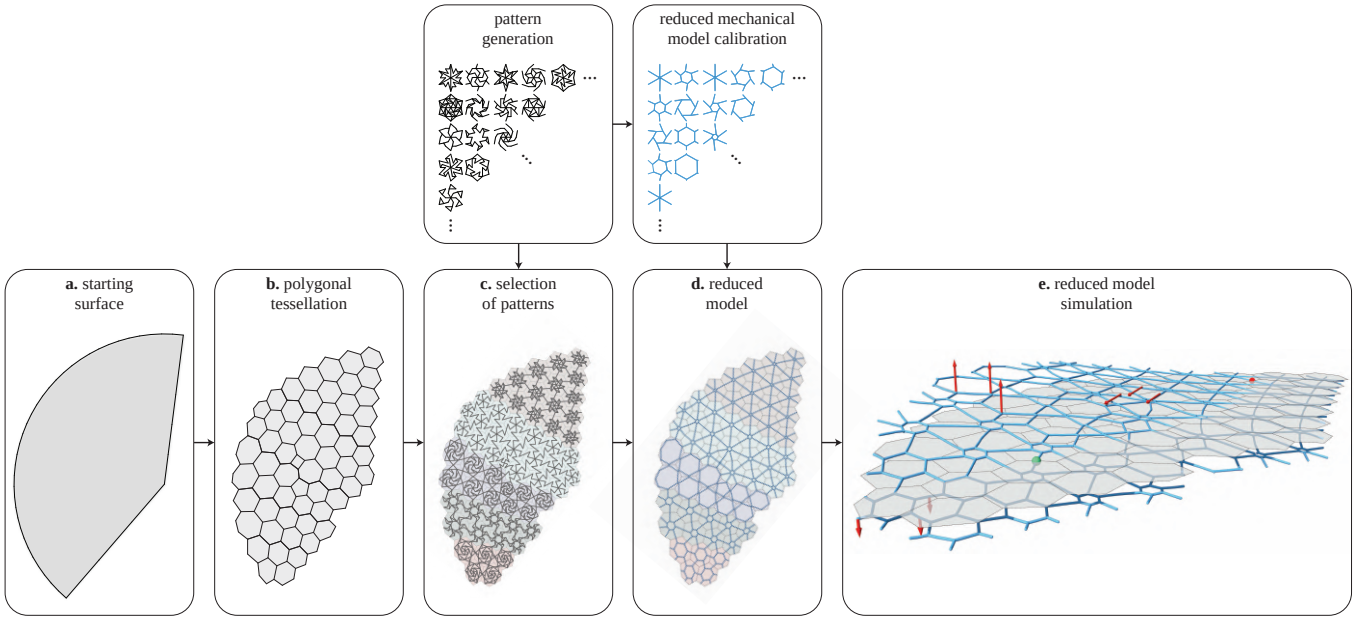



Fig. 1: Overview of the proposed framework.

has shifted the design problem from the discrete assembly of different material to the meso-scale structure design. At the same time, the possibility of defining a custom geometry within a given volume or surface has increased the complexity of the design problem, which in turn requires computational methods to be successfully performed.

When dealing with metamaterials, the usual design and simulation workflows are based on the repetition of a representative element (or families of elements) and on the approach of homogenization, i.e., in which their mechanical response can be simplified using low parameters continua models. However, the mapping between the metamaterial, or simply geometric pattern, and the continua space is constrained on one side to small geometric alterations and, on the other, on simple mechanical models; otherwise the problem quickly becomes intractable.

Instead of contributing in the wake of this continua-mapping work field, we tackled the problem from a different perspective by providing a method for generating generic tileable 2D pattern geometries, which can be arranged heterogeneously over a surface, and a tool to simulate the generic deformation behavior of such patterned surfaces. Our strategy opens up a broader design space of flat patterns that can be deformed in 3D, which are hard to cover using homogenization methods, also overcoming the pattern geometry limitations of other approaches based on reduced model calibration [4, 5]. Restricting the problem to 2D domains carries significant advantages for the fabrication process: the shape can be manufactured in a flat configuration, employing common additive and subtractive techniques and an ample spectrum of materials, such as plastics, wood, etc.

In this paper, we present three main contributions (Fig. 1). Firstly, we provide a method to generate a wide set of flat patterns made of uni-dimensional beam-like segments using a topological enumeration algorithm. Secondly, we introduce a single reduced mechanical model whose parameters can be calibrated to simulate each of these patterns. This model reduction relies on an efficient beam-like element formulation that outperforms the simulation time of using an accurate model. Contrarily to state-of-the-art strategies [6, 7], we do not target only in-plane deformations but rather out-of-plane bending behaviors and 3D deformations. Thirdly, we embed our reduced model in a design pipeline (bottom row of Fig. 1) that is suitable for any generic input flat shape (Fig. 1a). Starting from this shape, we derive a polygonal tessellation (Fig. 1b), in which each polygon can be filled with any pattern from our set to produce complex assemblies with varying aesthetic and mechanical properties (Fig. 1c). The patterns are replaced with their corresponding reduced models (Fig. 1d) to be rapidly simulated under generic loading and boundary conditions (Fig. 1e). The pattern set complies with mechanical and fabrication constraints, and constitutes a *palette* for the designer to interactively *paint* on the tessellated surface. We validate our approach on several application scenarios, i.e., tiling of patterns and accompanying simulations, and on 3D printed physical examples.

## 2. Related work

Patterns have been used for a long time in art and architecture for their aesthetic properties. In light of recent advancements in fabrication technologies, patterns have been extensively applied in numerous works (see the façades of MuCEM or Texoversum and the roof of Louvre Abu Dhabi). Several works use ornamental patterns to approximate a desired 3D shape with discrete tiles [8, 9, 10]. However, patterns are not only used for aesthetic purposes, but also to obtain a desired mechanical behavior [11, 12].

This interplay between mechanical behavior and aesthetics gave birth to new challenges related to the joint optimization of material and appearance. These challenges are often entwined with digital fabrication, which rendered feasible and cheap the production of highly complex objects with intricate geometric

features. At the same time, patterns with load-bearing purposes can be engineered up to a small scale. Significant research effort has been spent on the computational design of mechanical metamaterials for achieving a desired deformation capacity. Bickel et al. [13] used solenoid microstructures to expand the deformation gamut of multi-material compositions. The works in [14, 15] computationally design tileable cubic microstructures which can exhibit a range of elastic properties. Our pattern generation is inspired by the construction and filtering of the basic topologies used in [14], but concerns flat patterns tiled on a surface instead of cubic structures embedded in a volume.

Flat pieces have particularly appealing properties since they can be easily transported and manufactured in different sizes using a wide range of fabrication techniques. A representative work [4] approximates an input surface using flat parametrized spiraling patterns. Although this method has been employed to produce both small-scale and architectural-scale objects [16, 17], the pattern geometry is constrained to a four-arm spiral pattern; and the reduced representation provided in [4] is only applicable to that specific pattern. We provide both a wide set of flat patterns as well as their corresponding reduced representations, which can be for example useful in form-finding optimization loops.

Other lines of research focus on the design of *auxetic* structures. Auxetics are metamaterials that have a negative Poisson's ratio and thus expand laterally when stretched. The authors in [18] allow inextensible materials to uniformly stretch by introducing a trihexagonal cutting pattern onto flat panels. Due to the regularity of the hexagonal pattern, the fabricated structure can deform into an infinite family of surfaces. The method in [19] enables the spatial variation of the cutting pattern. Thus, they can uniquely encode the target shape, although their method still requires external actuation forces for retaining the 3D shape. This limitation is addressed in [20]. This method proposes a framework for computationally designing bistable auxetic structures, which retain their shape after an initial actuation.

Another line of work focuses on the design of irregular small-scale structures. The authors in [21] propose a method for generating irregular auxetic networks, primarily founded on geometric criteria. The generation of stochastic and irregular microstructures was previously explored also in [22, 23, 24, 25]. Recent works on irregular structures focus on enabling the smooth variation of 2D patterns and their corresponding mechanical properties. Martinez et al. [6] generate 2D irregular tiled geometries computed from Voronoi diagrams of regular lattices under star-shaped distance functions. The resulting microstructures can be interpolated and smoothly vary their mechanical properties. Similarly, Tozoni et al. [7] created a set of parametric rhombic microstructures with a continuous mapping between their geometric parameters and their mechanical properties. In [26] a homogenization approach for model reduction was proposed to reduce the cost of yarn simulations.

In this context, simulation models are essential to assess the mechanical behavior of pattern arrangements, and can be extremely slow when dealing with complex patterns. To solve this problem, a line of work relies on model reduction, e.g. homogenization theory [27, 28]. The goal of these works is to find the parameters of a low-resolution discretization that best approximates the behavior of a more complex one. Although [7, 6] provide homogenized models for pattern tilings, these are only applicable for in-plane scenarios and are not directly applicable to 3D.

In this work, we approximate a surface using planar patterns targeting out-of-plane (3D) deformation of flat materials. With a similar scope, Leimer and Musialski [5, 29] propose an approach for decoupling the pattern complexity from its simulation complexity, by encoding the mechanical behavior into a reduced geometry as well as in the parameters of a physical simulation model. Parametric spring-like assemblies are approximated with a reduced set of discrete elastic rods [30, 31], as also seen in FlexMaps [4]. Differently from the works [5, 29], we provide a method for producing a large pattern variety using a small set of parameters. Additionally, we provide an approach for approximating each pattern using a single parametric reduced model, which results in efficient simulations.

## 3. Method overview

This paper proposes a method for the generation and simulation of flat patterns, as shown in Fig. 2. These patterns are composed of a 2D arrangement of edges, which behave like a network of solid beams. Moreover, they are defined to be embedded into generic polygonal domains according to two fundamental properties of *exchangeability*, i.e., any pattern can be embedded into any polygon of a generic surface tessellation, and *tileability*, since any pattern can be connected to its adjacent ones.

To provide a fast simulation, we propose a single, parametric reduced model that is able to approximate the mechanical behavior of each individual pattern. Using ground-truth simulation results, from a set of prescribed loading scenarios applied to each pattern, we fit the reduced model to each pattern. Eventually, this set of reduced model parameters allows us to implement an approximate simulation strategy, which is capable of interactively handling very large tessellations, composed of hundreds of different patterns.

Essentially, our research strategy can be summarized as follows:

- we generate a family of flat and tileable 2D patterns using a topological enumeration strategy. These patterns are constituted of a network of beams;
- we introduce a method to optimize a reduced mechanical model to match the deformation behavior of each generated pattern;
- we develop a design pipeline for tiling a general flat surface with our patterns. Subsequently, leveraging the computed reduced models, we simulate the mechanical response of a patterned tessellation for given loading and boundary conditions;
- we implement the previous step in a visual tool, providing a user interface for interactive editing of the desired settings;

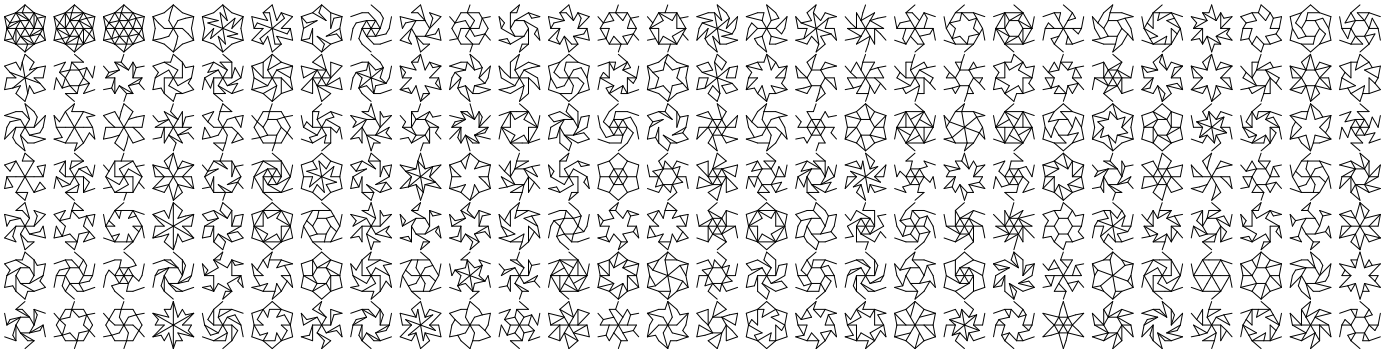


Fig. 2: A subset of the patterns generated with our method.

- we validate our approach by comparing the mechanical behavior of various pattern tessellations against their reduced model counterparts.

## 4. Pattern generation

To design a set of flat patterns of large aesthetic and behavioral variety with the aim of discretely changing their distribution on the surface as desired, we define two requirements:

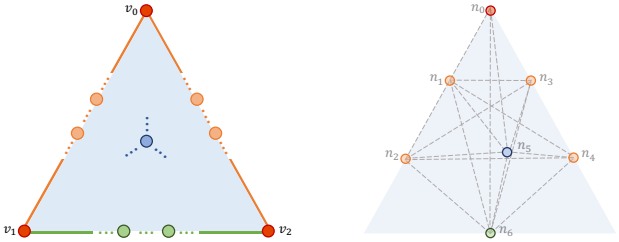
- *exchangeability*, i.e. a pattern can be embedded into any generic polygon;
- *tileability*, i.e. a pattern can be connected to its adjacent ones.

We enforce the exchangeability of patterns by design, by embedding patterns individually into a discrete surface region, namely a flat polygon. We also enforce tileability by imposing boundary compatibility between adjacent patterns. Each pattern is made of a graph-based combination of solid straight beams sharing endpoints. However, for a specific beams' arrangement to be meaningful, its geometry has to fulfill geometric criteria considering both fabrication and mechanical constraints.

### 4.1. Pattern enumeration

Searching all possible patterns in a generic polygonal domain is a computationally infeasible task. However, its complexity can be reduced by adopting a smaller domain and a finite setup. We decompose each polygon into a triangle fan tessellation, whose shared vertex is its centroid. Thus, the basic configuration of a pattern can be defined on a single base triangle and then replicated by rotational symmetry around the polygon's centroid. We fix a finite number of nodes within the base triangle domain constituting a set of possible vertices. Finally, we consider all possible graphs connecting these nodes as candidate patterns.

Given a set of nodes $n_i \in \mathcal{N}$ on a base triangle, we enumerate all possible graphs $g_i \in \mathcal{G}$ that emerge by different connections among $n_i$. For varying the complexity and aesthetics of the resulting patterns, different set of graphs $g_i \in \mathcal{G}$ can be obtained by setting different number of nodes or by modifying their position on the base triangle. However, the set of nodes must comply with some basic constraints in order to guarantee tileability and exchangeability.




(a) (b)

Fig. 3: Pattern enumeration setup on the base triangle: (a) domain and region identification for building the nodes set; (b) our adopted enumeration setup.

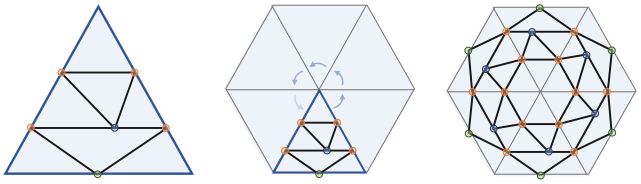
With reference to Fig. 3a, let us divide the base triangle $v_0v_1v_2$, in which $v_0$ is the original polygon centroid, into four regions: the radial edges (colored in orange), the interface edge (green), the triangle face (light blue), and the triangle vertices (red). The subset of nodes laying in each region will be subject to different constraints.

The two radial edges define a unique region, so only pairs of nodes that are equidistant from the vertex $v_0$ can populate it. Regardless of the number of pairs, this prerequisite avoids the generation of unconnected patterns once the graph is replicated within the polygon.

The interface edge is a shared element between polygons, which can possibly be filled with different patterns. To comply with the tileability constraint, every graph $g_i$ is required to share the same nodes on the interface edge. This constraint practically implies that the positioning of the nodes has to be symmetric with respect to the midpoint. Moreover, all these nodes must necessarily be included in each graph $g_i$ to guarantee that each pattern is connected to its adjacent. The number and the position of the subset of nodes on the base triangle face can be freely defined.

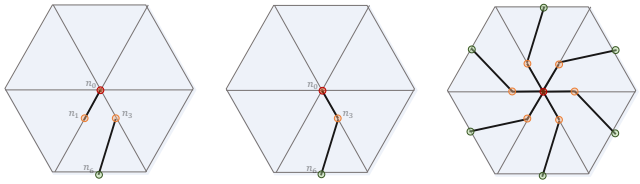
Lastly, concerning the triangle vertices, a distinction should be made between the central node $v_0$ and the others. Referring to the enumeration in Fig. 3a, the vertex $v_0$ has the same properties as any face node, and can be part of the graph $g_i$ or not. Vertices $v_1$ and $v_2$ are interface nodes and they are subject to the same constraints (can be either used both or none of them). In our setup, their use is restricted due to tileability reasons.

In order to generate a pattern $p_i$, we replicate each triangle's graph $g_i$ on all the triangles of a polygon by assembling an edge mesh. Each graph $g_i$ fits any triangle and fan tessellation since



(a) (b) (c)

Fig. 4: A pattern example generated with a chosen set of edges on the base triangle. We refer to the graph in the base triangle (a) as base triangle pattern. For tessellating a polygon with our pattern we use barycentric interpolation and polar repetition around the polygon's centroid (b). This produces what we refer to as the fan configuration of the pattern (c).



(a) (b) (c)

Fig. 5: Two base triangle configurations (a) and (b) with two distinct graphs, which result in the same fan configuration (c). The cause of equivalence is the radial edges in building the fan configuration.

we rely on barycentric interpolation for replicating the pattern. As a result the approach is scale- and shape-invariant, and thus can be applied even to irregular tessellations. Although the pattern embedding strategy works for generic polygons, for the sake of clarity in our illustrations we always use an equilateral triangle, which results in a hexagonal fan tessellation.

Since the enumerable patterns is exponential in the number of nodes, in our setup we limit the number of generated patterns. As shown in Fig. 3b we rely on 7 nodes $n_i$ on the base triangle: the $v_0$ vertex ($n_0$), two equally spaced nodes on the radial edges ($n_1$, $n_2$, $n_3$, and $n_4$), one on the triangle face ($n_5$), and a single interface node at the midpoint of the edge $v_1$,$v_2$ ($n_6$). Therefore, every pattern configuration is obtained from a $g_i$ including a set of edges chosen among the ones in Fig. 3b. An example of pattern $p_i$ generated from this setting can be seen in Fig. 4.

In the following, we will denote as $\mathcal{P}$ the set of all possible patterns $p_i$ that we generate. As described in Sec. 4.2, not all the generated patterns in $\mathcal{P}$ will be admissible.

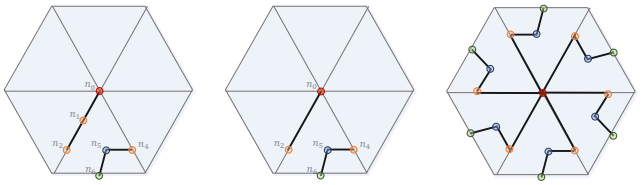
### 4.2. Filtering

The patterns must possess geometric properties to be meaningful from both a fabrication and a mechanical point of view. An *a posteriori* filtering process identifies and discards non-compliant graphs. As a result, each graph $g_i$ is labelled as valid or invalid, producing a subset of valid patterns $\mathcal{P}' \subseteq \mathcal{P}$. The conditions to mark a pattern as invalid are: (i) geometric or mechanical equivalence, i.e., presence of duplicates and/or inactive sub-parts; (ii) topology irregularity, i.e., intersecting edges and multiple-connected components; (iii) unfeasible fabrication details. All of them can be expressed as geometric limitations to be adopted for graphs.

*Duplicate patterns.* One of the desired properties of the pattern set $\mathcal{P}'$ is to be composed of unique patterns, avoiding duplication. Duplicate patterns occur when two different graphs, namely graphs that connect different nodes on the base triangle, result in the same fan configuration. Duplicate patterns may be of two kinds: edge meshes with the same topology and edge meshes with different topologies resulting in the same shape.

An example of the first one can be seen in Fig. 5. Duplicate edge meshes with the same topology are caused by the equivalence generated by repeating the graph segments on one (Fig. 5a) or the other (Fig. 5b) radial triangle edge in assembling the fan configuration. In our setup, we force the enumeration algorithm to not use edges that lay on the right radial triangle edge (namely $\{n_0, n_3\}$, $\{n_0, n_4\}$ and $\{n_3, n_4\}$).

An example of the second kind can be seen in Fig. 6. Different topologies that result in the same shape cause duplicate edge meshes with a mechanical equivalence. Indeed, introducing more nodes within a beam does not alter its structural response. Therefore, we do not consider candidate edges that include nodes other than its endpoints. For example, the edge $\{n_0, n_2\}$ is never considered in the pattern generation (Fig. 6).



(a) (b) (c)

Fig. 6: Two different base triangle configurations with two distinct graphs (a) and (b), which result in the same shape, i.e., same structure (c).

*Dangling parts.* A graph can be marked as invalid if it builds a pattern that is mechanically equivalent to another one in our set. However, apart from the trivial duplicate shape condition applied in the previous paragraph, another equivalence condition may occur if parts of the pattern do not contribute to the transferring of forces. Therefore, these parts are deemed to remain unstressed if the small contribution provided by their self-weight is negligible.

Mechanically equivalent patterns are produced in two cases. In the first case, the pattern contains at least a node with valence one which will result in *dangling edges* (Fig. 7a). In the second case, the pattern contains articulation points, which signal the existence of generic *dangling component* (Fig. 7b). The second condition is more general and it is therefore sufficient to check the pattern graph for articulation points to filter them out.

*Interface connectivity.* Patterns that have no connection to the interface edge are marked invalid, as they will not be connected to adjacent patterns in their final tessellated configuration. Specifically, we discard all base graphs $g_i$ that do not include at least an edge connected to the interface node $n_6$.

*Multiple-connected components.* Patterns that do not result in a single connected component are discarded (see Fig. 7c). Multiple-connected components patterns are not significant from a mechanical point of view, since independent (or rigid) movements between parts of the structure are allowed. By also

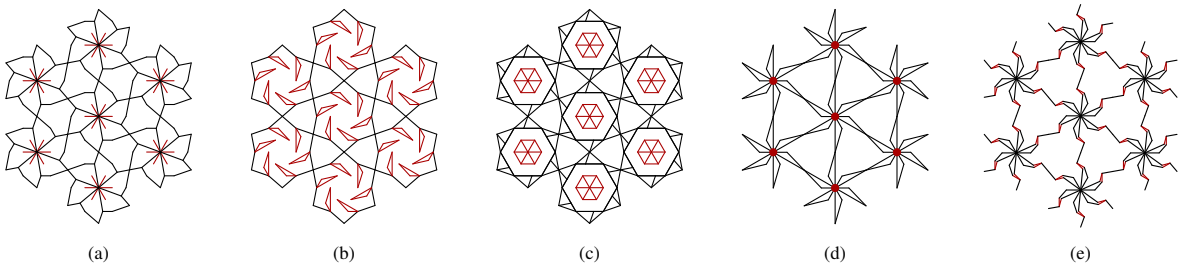



Fig. 7: Arrangements of seven hexagonal patterns that show pattern filtering to enforce fabrication and mechanical constraints. Red-colored parts violate one constraint producing: (a) dangling edges; (b) articulation points; (c) multiple-connected components; (d) high valence nodes; (e) small angles between incident edges.

forcing every base triangle pattern to be connected to its interface (previous condition), we ensure that any patterned surface form a single connected component.

*Intersecting edges.* Graphs that include intersecting edges are discarded. By design, our patterns are composed of edges that are exclusively connected at the nodes of the base triangles. Thus, such graphs can be automatically converted into a network of beam elements. Instead, intersecting edges do not make this conversion accurate as two intersecting beams would physically behave as if they were independent.

*Fabrication constraints.* In order to reduce the discrepancies between simulations and the manufactured pattern's behavior, the fabricated elements have to be as close as possible to their digital representation. When fabricating such beam-like structures, a well-known issue is the accumulation of material at the nodes, which results in a local stiffness increase. This effect occurs, for example, if two incident edges are too close to each other, merging in a unique element with a larger cross section. Moreover, this effect is amplified due to the fabrication machine tolerances. Conversely, the model employed in beam simulations and thus also in this work (see Section 5) is based on uni-dimensional elements and dimensionless nodes. Thus, the higher the valence of a node, the higher its stiffness when manufactured. And accordingly, the higher is the discrepancy from the real behavior with respect to the simulated one. To cope with this issue, we filter out patterns having incident edges joining at a small angle and patterns having high valence nodes. In our experimental setting, we adopt a minimum angle of 15° and maximum valence of 6.

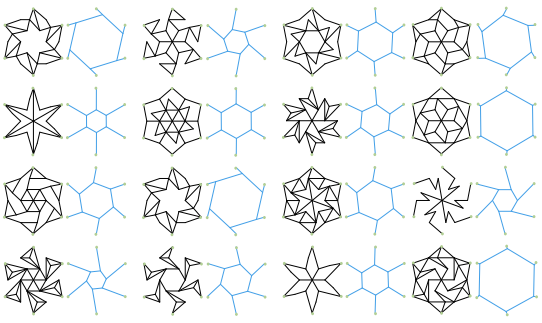


Fig. 8: Examples of full patterns with their corresponding reduced models.

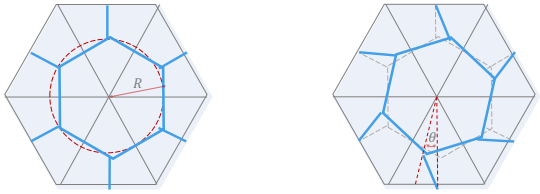
## 5. Reduced model calibration

Our pattern set is widely heterogeneous from a geometrical point of view (see Fig. 2). Simulating such pattern assemblies can be extremely costly. The simulation cost is linked to the number of elements, i.e., nodes and edges. Regardless of the mechanical model employed, the simulation cost grows when the pattern complexity increases, and the number of tiled patterns increases. We tackle this problem at the pattern level by proposing a unique reduced model, which provides a generalized representation of the mechanical behavior of any single pattern and can approximate it accordingly. The reduced model must possess the same properties as regular patterns namely, tileability and exchangeability, such that reduced models can be seamlessly used in place of their patterns.

Intending to maintain a standard simulation cost for each pattern, we adopted a reduced model with a constant topology (Fig. 8). Then, we search for its best geometric and mechanical parameters to match the mechanical response of each pattern. The problem of fitting the reduced model parameters to every single pattern is formulated as a non-linear constrained optimization problem with respect to several standard simulation scenarios to provide a general and satisfactory description of each pattern's mechanics. The response of the pattern to each simulation scenario is computed by using an accurate yet costly ground-truth simulation model [32]. Instead, we compute the reduced model's response using a simplified linear model [33]. Note that both mechanical models used in this work use discrete beams modeled as uni-dimensional elements. Each beam endpoint merges into a structural node and has six degrees of freedom (DOFs): $x, y, z$ and $\phi_x, \phi_y, \phi_z$, used to describe translations and rotations, respectively. The underlying assumption is that every pattern in $\mathcal{P}'$, which is stored as an edge mesh, can be converted unequivocally into a structural model: edges and nodes become beam centerlines and structural nodes, respectively. Moreover, each beam is equipped with both geometric and material cross-section properties. An overview of the mechanical models is available in the additional material.

### 5.1. Reduced model overview and characterization

The general objective of a reduced model is to provide a compressed and simplified representation of a more complex structural assembly. In the present case, this representation is composed of a network of beams, a set of cross-section properties and a linear mechanical solver [33].




(a) (b)

Fig. 9: We define the reduced model's geometry as a hexagon with attached edges at its vertices (light blue). During the optimization of the reduced pattern, we allow it to change shape in two ways: by altering the inner hexagon's size (a), and by allowing it to rotate around its center (b).

The network of beams builds on a globally adaptable unique topology defined as a hexagonal chain of edges centered in the centroid of the hexagonal polygon, whose vertices are attached via edges to the midpoint of each interface edge. In our setting, the reduced model and the pattern share the same interface nodes (green nodes in Fig. 8) and their barycenters. In order to describe a large variety of patterns' behavior, we characterize this topology with two geometric parameters: $R \in (0, l_e\sqrt{3}/2)$, which describes the circumradius of the hexagon (Fig. 9a), where $l_e$ is the length of the polygon's side; and $\theta \in (-\pi/6, \pi/6)$ which describes the rotation of the link edge (Fig. 9b). We choose this geometry for our reduced model by observing three main types of deformation when the patterns are stretched: a symmetric deformation ($\theta \approx 0$, e.g. second row, first column in Fig. 8), and a polar rotational deformation in a counterclockwise ($\theta < 0$, e.g. last row, first column in Fig. 8) and clockwise direction ($\theta > 0$, e.g. first row, second column in Fig. 8).

For capturing the pattern stiffness, we introduce mechanical parameters to characterize the beams of the reduced model. We completely define the beam by selecting 4 parameters: the cross-section area of the beam $A$, the bending inertia values $I_2$ and $I_3$ along the main axes of the beam section, and the torsion inertia $J$. We fixed a unique Young's modulus since it results redundant (i.e., it behaves as a scaling factor for the other parameters). We empirically found this choice of parameters performing best in approximating the patterns' behavior. Physically, all these parameters are interrelated and depend on the cross-section type and geometry. However, since the reduced model is intended to be solved with the linear Euler-Bernoulli theory, they can be considered independent variables, broadening the spectrum of the reduced model behavior across scenarios. Eventually, this reduced model is defined by selecting 2 geometric parameters $R$, $\theta$ and 4 cross-section parameters $A$, $I_2$, $I_3$, $J$, since all beams in our reduced model share the same cross-section properties.

### 5.2. Simulation Scenarios

For the general applicability of the patterns, we propose five standard scenarios that cover a wide range of deformations to abstract the mechanical behavior of the pattern. Our goal is to activate the main ways of loading 2D patches, namely stretch, shear, and bending, as independently as possible to adopt weighting strategies in the reduced model calibration phase (Sec. 5.3).

Axial (Fig. 10a) and shear (Fig. 10b) are planar scenarios that are obtained by fixing two adjacent interface nodes and by moving the opposite ones. In the axial case, the forces are directed towards the supports, while in the shear case, the force is orthogonal. The remaining three scenarios simulate the bending behavior and are distinguished by the Gaussian curvature $K$ assumed by the deformed shape. An almost cylindrical shape with $K \approx 0$ is obtained by fixing three consecutive interface nodes and pulling the rest in the normal direction (Fig. 10c). A $K > 0$ dome shape is obtained by applying opposite moments on two pairs of opposite interface nodes and forcing the remaining pair of opposite nodes towards the interior (Fig. 10d). A $K < 0$ saddle-like shape is created by applying opposite-deforming moments and forces (Fig. 10e).

From the five simulation cases described above, we derive a larger set of simulation scenarios by varying the force magnitude. We do this in order to evaluate the behavior of the patterns on a broader force spectrum. We uniformly sample the force magnitudes by scaling down a maximum value, which is uniquely defined for each of the five scenarios. These magnitudes are empirically set and limit the range of pattern behaviors, used to calibrate the reduced model, to realistic load cases. Eventually, for each pattern, the reduced model calibration is performed considering a finite set of $s_k \in \mathcal{S}$ scenarios. Since our reduced model calibration is performed on regular hexagons tessellations having polar symmetry, the mechanical response will be the same if in every scenario the geometry is rotated by 60°. Therefore, to compute every reduced model, in-plane rotated patterns are not included, favorably reducing the number of simulation scenarios to a minimum.

### 5.3. Optimization framework for calibrating the reduced model parameters

The parameters of the reduced model $x$ are calibrated from a comparison between the ground-truth simulation of the pattern and the reduced model simulation. This comparison is made based on the interface nodes' displacements for each simulation scenario $s_k \in S$. For clarity, the quantities related to the pattern are marked by a $p$ subscript (e.g., $d_p$ for the displacement), to be distinguished from the reduced model for which we use a $r$ subscript (e.g., $d_r$). For a given pattern we compute its optimal reduced model parameters $x_{opt}$ by solving the non-linear constrained optimization problem defined as:

$$x_{opt} = \min_x(E(x)) \tag{1}$$

$$s.t. \quad M_p \cdot a_p + r_p = F_k \qquad \forall s_k \in S \tag{2}$$

$$K_r(x) \cdot d_r = F_k \qquad \forall s_k \in S \tag{3}$$

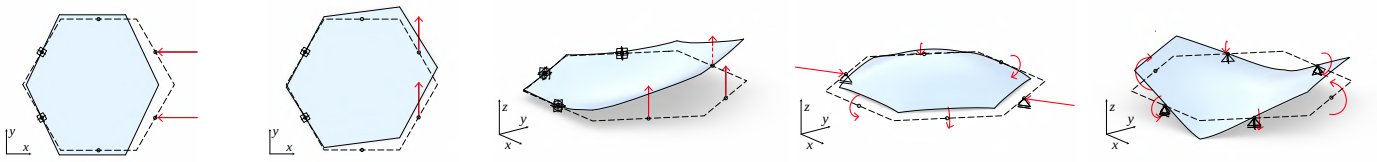
$$x = [A, I_2, I_3, J, \theta, R] \tag{4}$$

$$A, I_2, I_3, J \in \mathbb{R} \tag{5}$$

$$-\pi/6 \leq \theta \leq \pi/6 \tag{6}$$

$$0 < R \leq l_e\sqrt{3}/2 \tag{7}$$

The objective function $E(x)$ quantifies the error between the pattern and the reduced model node displacements. It depends on the vector $x = [A, I_2, I_3, J, \theta, R]$, which contains the reduced model parameters (defined in Sec. 5.1) as problem variables.



(a) (b) (c) (d) (e)

Fig. 10: Standard simulation scenarios adopted in the reduced model calibration: (a) axial; (b) shear; (c) bending; (d) dome; (e) saddle.

The cross-section parameters are free to take any real value. In practice, to minimize this objective, we limit variables extent to multiples of the pattern's constitutive material values in the range $[0.001, 1000]$. Eq. (6) and Eq. (7) define the range of reduced model geometry parameters. Eq. (2) expresses the non-linear static equilibrium according to [32], where $M_p$ is the structure mass matrix, $a_p$ is the accelerations vector, $r_p$ is residual force vector and $F_k$ is the force vector of scenario $k$ (which is unique for both cases). Eq. (3) states the static equilibrium of the reduced pattern adopting the linear beam model [33], where $K_r$ is the stiffness matrix expressed as function of $x$.

For the objective function $E(x)$, we require the definition of an error metric that can be used to evaluate the choice of the reduced pattern variables $x$. This function is the sum of the errors $e_k(\bar{x})$ of each scenario $s_k$ in terms of displacements, for a given vector of variables $\bar{x}$:

$$E(\bar{x}) = \sum_{s_k \in S} w_k \cdot e_k(\bar{x})^2 \tag{8}$$

Each scenario is weighted using a term $w_k$. When simulating the scenarios, using both the non-linear ground-truth model and the reduced linear model, the displacement vector for each of the interface nodes is a 6-dimensional vector made of three translational components and three rotational components. To compute the error between these models, we decouple and normalize the translation contribution $e_{k,trans}(\bar{x})$ from the rotations $e_{k,rot}(\bar{x})$. Additionally, we introduce a distinct weight for either terms $w_{trans}$ and $w_{rot}$, for tuning their importance within the reduced model calibration:

$$e_k(\bar{x}) = w_{trans} \cdot e_{k,trans}(\bar{x}) + w_{rot} \cdot e_{k,rot}(\bar{x}) \tag{9}$$

For computing the terms in $e_k(\bar{x})$, we define the set of pairs $\{v_p, v_r\} \in Q$ as the set of matching interface nodes between the pattern and reduced model, respectively. We compute $e_{k,trans}$ as the sum of the Euclidean distances over pairs in $Q$:

$$e_{k,trans}(\bar{x}) = \frac{\sum_{\{v_p,v_r\} \in Q} \left\| d_{p,k} - d_{r,k}(\bar{x}) \right\|}{\max\left( \sum_{v_p} \left\| d_{p,k} \right\|, \epsilon_{trans} \right)} \tag{10}$$

Where $d_{p,k}$ is the 3-dimensional translational displacement of the interface vertex $v_p$ of the pattern for the simulation scenario $s_k$. Similarly, $d_{r,k}(\bar{x})$ is the displacement of the corresponding reduced model vertex $v_r$ as function of the parameters $\bar{x}$. To obtain an error independent from the displacement range, and thus from the pattern stiffness, this function is normalized with respect to the sum of norms of the interface vertices' displacements $v_p$. To avoid numerical issues for stiff patterns (i.e. with very low displacements) the normalization value is clamped to a minimum threshold $\epsilon_{trans}$.

For computing the rotational error $e_{rot}$ we quantify the error between $Q$ pairs as the minimum angle that is needed for their locally-rotated frames $R_p$ and $R_r$ to match:

$$e_{k,rot}(\bar{x}) = \frac{\sum_{(v_p,v_r) \in Q} \alpha(R_{p,k}, R_{r,k}(\bar{x}))}{\max\left( \sum_{v_p} \alpha(R_{p,k}, R_{p0}), \epsilon_{rot} \right)} \tag{11}$$

Where $\alpha(R_0, R_1)$ is the angle for rotating an orthogonal frame $R_0$ into another one $R_1$ according to Euler's rotation theorem. $R_{p,k}$ is the local frame of the interface vertex $v_p$ in the deformed state of scenario $s_k$. $R_{r,k}(\bar{x})$ is the rotated frame of the corresponding reduced model vertex $v_r$ as a function of the parameters $\bar{x}$. Similarly to Eq. (10), we normalize this function using the sum of angles $\alpha(R_{p,k}, R_{p0})$ between the vertex frames in the deformed state and the rest frames $R_{p0}$. As for the translation error, we also introduce a minimum rotation threshold $\epsilon_{rot}$.

For solving the problem of Eqs. (1)-(7), due to the complexity of our function, we use a derivative-free global optimization method (simulated annealing provided by ensmallen [34]).

## 6. Tiling and interactive tool

All generated patterns and their corresponding reduced models can be tiled onto a variety of surfaces. As a first step, these surfaces are processed with an isotropic triangular remesher. The dual of the obtained triangular mesh is the polygonal mesh used as the base for the tiling. Isotropic triangles produce regular polygons and uniformly distributed cells, making this step compliant with the model reduction strategy. The higher the regularity of the polygonal tessellation, the more accurate the reduced model simulation will be. Additionally, since our reduced model calibration method targets polygons of a specific size, we size the polygonal mesh such that its mean edge size matches the interface edge adopted in the model calibration.

Subsequently, each polygon tile can be filled with any desired pattern from the subset of valid patterns $\mathcal{P}'$. Then, any loading and boundary conditions can be applied to the polygonal mesh and interactively simulated using the reduced models. These conditions can also be modified and the patterns substituted. To further favor interactivity, we provide the user with a tool to interactively create pattern tessellations and set large simulation cases. This tool imports a polygon mesh on which the user can 'paint' the patterns over the polygons. The result is a mapping between each surface polygon and one pattern (with its accompanying reduced model). For a video demonstration of the tool, the reader is referred to the additional material of the paper.

Table 1: The filtering statistics for our pattern enumeration of Fig. 3b, according to Section 4: the number of edges of the base triangle configuration $g_i$ ($E$), patterns with $E$ edges in the base triangle ($P^E$), patterns containing intersecting edges (*Inters.*), patterns not linked to the interface nodes (*No int. nodes*), patterns with multiple connected components (*Multip. CC*), patterns with at least one articulation point (*AP*), patterns having at least one angle smaller than 15° ($A < 15°$), patterns containing nodes with valence higher than 6 ($V > 6$) and, eventually, valid patterns in the $\mathcal{P}'$ set (*Valid*).

| $E$ | $P^E$ | *Inters.* | *No int. nodes* | *Multip. CC* | *AP* | $A < 15°$ | $V > 6$ | *Valid* |
|---|---|---|---|---|---|---|---|---|
| 2 | 136 | 23 | 55 | 95 | 107 | 5 | 3 | 9 |
| 3 | 680 | 268 | 165 | 359 | 524 | 73 | 43 | 47 |
| 4 | 2380 | 1442 | 330 | 628 | 1669 | 489 | 302 | 172 |
| 5 | 6188 | 4774 | 462 | 722 | 3349 | 1995 | 1334 | 429 |
| 6 | 12376 | 10934 | 462 | 570 | 4438 | 5548 | 4131 | 576 |
| 7 | 19448 | 18460 | 330 | 316 | 4055 | 11150 | 9481 | 405 |
| 8 | 24310 | 23873 | 165 | 120 | 2640 | 16747 | 16302 | 154 |
| 9 | 24310 | 24197 | 55 | 28 | 1228 | 19167 | 20632 | 29 |
| 10 | 19448 | 19435 | 11 | 3 | 392 | 16881 | 18754 | 3 |
| 11 | 12376 | 12376 | 1 | 0 | 77 | 11465 | 12354 | 0 |

Table 2: Weights adopted in our experimental setting.

| *Scenarios* | $w_k$ | $w_{trans}$ | $w_{rot}$ |
|---|---|---|---|
| axial, shear | 0.5 | 1.2 | 0.8 |
| bending, dome, saddle | 1.0 | 1.2 | 0.8 |

## 7. Experimental configuration and Results

To demonstrate the applicability of the proposed methodology for the pattern generation, we adopt the setting depicted in Fig. 3b. Then, we select a set of surfaces (Fig. 11) which we tessellate using the generated patterns and we run a simulation leveraging the computed reduced models. The accuracy of the reduced solution is assessed by comparing it with the solution obtained from the ground-truth solver.

### *7.1. Pattern generation and reduced model calibration*

The set of patterns is generated from all possible graphs obtainable from the configuration of seven nodes of Fig. 3b. We adopt a single interface node on the edge midpoint, and a single node on the triangle face positioned eccentrically. This simple configuration produces a broad range of patterns' complexity with a limited number of nodes. We obtain a set of 121652 patterns that, once filtered, become a set of 1824 valid patterns. The filtering statistics are included in Table 1.

In the experimental setting of model reduction, we adopted different parameters on an empiric base for the material, size and intended use of the patterns to be meaningful. In the error formulation for computing the objective function, we use $\epsilon_{trans} = 0.0003$ meters and $\epsilon_{rot} = 0.1°$. The cardinality of $Q$ is 6 since we embed our base triangle patterns in hexagons.

The weights considered in this setup are listed in Table 2 and have been empirically chosen as well to minimize the error. The scalar $w_k$ has the purpose of restoring an equal contribution of symmetric and non-symmetric scenarios within the optimization objective. The scalars $w_{trans}, w_{rot}$ are set as scenario-invariant quantities with a general objective of shifting the desired accuracy more on translations rather than rotations.

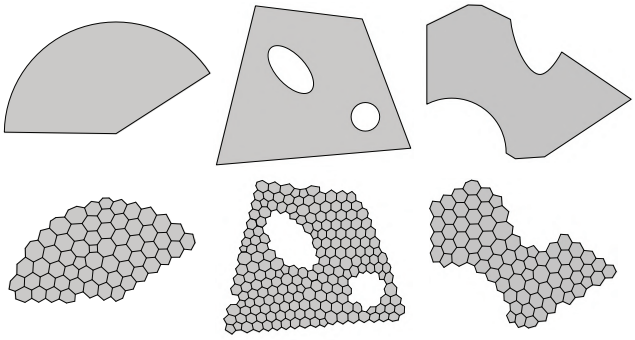


Fig. 11: Plan view of the surfaces adopted in the experiments (top) and their corresponding polygonal tessellations (bottom).

The reduced models are computed from regular hexagons inscribed in a circle of diameter $0.03\sqrt{3}$ meters. We input material properties that are on average found on plastics to be 3D printed [35], namely 1 *GPa* as Young's Modulus and 0.3 Poisson's ratio. For the patterns, we adopted a square cross-section of $2 \times 2$ *mm* that is constant over all the edges of the patterns. For optimizing our pattern set, we used an Intel I9-7920X @ 2.90GHz $\times$ 24 CPUs. The average reduced model calibration time was 1.2 minutes per pattern, in which the majority of time being used for computing the ground-truth data. The average optimal objective value found via our reduced model calibration method was 9.7 with a standard deviation of 9.9. For additional information regarding the parameters used in our implementation we refer the reader to the Appendix in the supplementary material. As additional material for each valid pattern we also included its edge mesh, its reduced model geometry and its optimal parameters.

### *7.2. Experiments and validation*

We used Instant Meshes [36] for generating almost isometric triangular meshes, then we retrieve the polygonal tessellation from their duals. For consistency, to build any simulation scenario, loads and boundary conditions must be shared between both the pattern model and the reduced model. This condition is fulfilled if they are applied on the interface nodes,

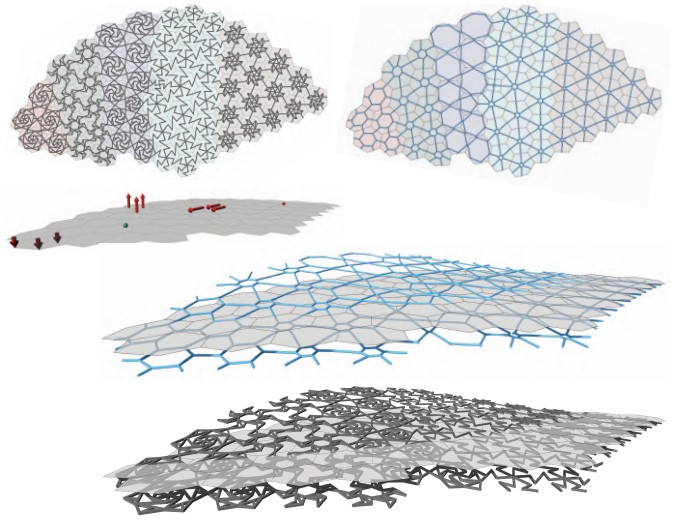


Fig. 12: Example A results: pattern tiling and the corresponding reduced model on top; simulation setup; reduced model deformed shape (blue) and ground-truth deformed shape (grey).

which are shared between patterns and their reduced models. The results of the experiments are included in Fig. 12 (example A), Fig. 13 (example B) and Fig. 14 (example C), each showing a top view of the tiled patterns and their reduced models, the simulation setup, and the deformed state for both configurations. The colours of the polygons in the top views are meant to distinguish different patterns. Forces are represented as arrows whose size is proportional to their magnitude; boundary conditions are spheres, red if all DOFs are constrained and green if only the translations are constrained.

The outcome of our pipeline is the deformed network of the reduced models. It is important to point out that in the rendering, we adopt a constant circular cross-section for all reduced models because the properties obtained from the calibration are all different and may not have a physical meaning. To validate this result, we simulate the tiled patterns within the same setup using the time-consuming ground-truth solver in [32]. For the examples A, B and C, the simulation time was 1.9, 4.2 and 6.3 minutes, respectively. The corresponding reduced simulation times are 80, 110, and 40 milliseconds, respectively. The error between the two results is shown in Table 3, in which for all examples we report the maximum, the mean, and standard deviation values. The error is computed on the interface nodes, and the translation component is distinguished from the rotational component. Overall, the model reduction strategy adopted in this work demonstrates to be accurate enough for exploring various geometries and loading configurations. However, the accuracy diminishes once the displacement range increases, i.e. Example A is less accurate than others due to the limitations of the linear model to well describe large deformations.

### *7.3. Physical tests*

To assess the practical utility of the present methodology, we perform a simple test on 3D-printed cantilever surfaces that adopt different patterns, whose elements have a constant cross-section of $2 \times 2$ *mm*. The material used for the actual physical

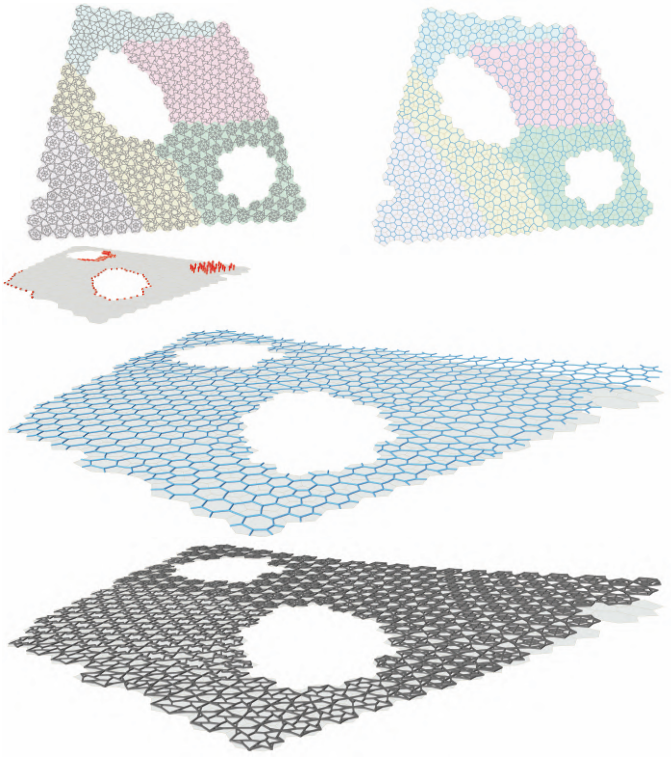


Fig. 13: Example B results: pattern tiling and the corresponding reduced model on top; simulation setup; reduced model deformed shape (blue) and ground-truth deformed shape (grey).

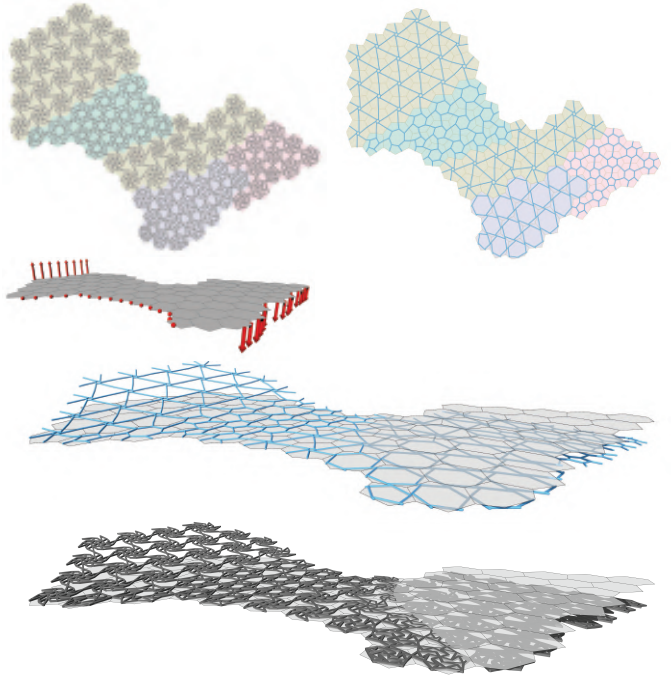


Fig. 14: Example C results: pattern tiling and the corresponding reduced model on top; simulation setup; reduced model deformed shape (blue) and ground-truth deformed shape (grey).

Table 3: Error between the reduced model and the ground-truth simulations on all experimental examples. The table shows the bounding box of the starting flat surface ($size_{BB}(m)$), the maximum displacement norm of the nodes in the ground-truth simulation ($dist^p_{max}$), the maximum distance between an interface node in the pattern and the corresponding node in the reduced ($dist_{max}(m)$), the average distance between the interface nodes in the pattern and reduced configurations ($dist_{avg}(m)$) and its standard deviation ($dist_\sigma$). Then, the same measures computed for the rotation: $rot^p_{max}$, $rot_{max}(rad)$, $rot_{avg}(rad)$ and $rot_\sigma$.

| | $size_{BB}(m)$ | $dist^p_{max}$ | $dist_{max}(m)$ | $dist_{avg}(m)$ | $dist_\sigma$ | $rot^p_{max}$ | $rot_{max}(rad)$ | $rot_{avg}(rad)$ | $rot_\sigma$ |
|---|---|---|---|---|---|---|---|---|---|
| Ex. A (Fig. 12) | 0.71x0.40 | 0.0444 | 0.0072 | 0.0027 | 0.0252 | 0.2247 | 0.0378 | 0.0278 | 0.0844 |
| Ex. B (Fig. 13) | 1.08x0.91 | 0.0454 | 0.0042 | 0.0006 | 0.0219 | 0.1101 | 0.0110 | 0.0035 | 0.0803 |
| Ex. C (Fig. 14) | 0.80x0.64 | 0.0620 | 0.0065 | 0.0011 | 0.0233 | 0.2670 | 0.0241 | 0.0063 | 0.0703 |

models is PLA with a density of 1.24 $g/cm^3$ and stiffness of $E = 2.3465\ GPa$. We rigidly fixed the specimens on one side and let them deform under their own weight. In this test setup, we 3D scan the specimens. We repeated this test on two specimens measuring $18 \times 26\ cm$, tessellated with different densities patterns. We diversify them with respect to their global stiffness and weight in order to achieve different displacements.

We simulate the experimental response of the specimens with both the ground-truth solver and the reduced model. In both cases, all DOFs are restrained on the fixed side and the weight is lumped, i.e., considered as concentrated forces on the interface nodes only. To comply with the real-life material properties, the optimization of the reduced model is re-computed.

We compare the deformed shapes with respect to the experimental results acquired from the 3D scan, as shown in Fig. 15. Remarkably, the present methodology proves to be fairly general and versatile enough to be tailored to real-world needs. Indeed, the ground-truth and the reduced model are in both cases in good agreement. A more significant deviation is found between the ground-truth result and the tested shape. The denser the patterns, the more evident this effect is. Two or more beams sharing a node, in fact, result stiffer than any centerline-based mechanical model can simulate. This effect is caused by the physical size of the node, which, especially for small angles between incident beams, behaves like a wide area of almost null deformation. Being these examples bending-dominant cases, the reduced model shows the same limitations as the linear simulation model [33] (see Appendix in additional material), more precisely, the non-shrinkage along the cantilever length.

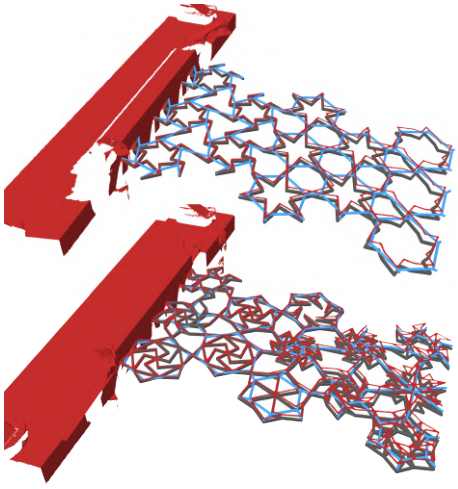


Fig. 15: Physical tests (red) in comparison with deformations of the pattern (dark grey) and the reduced model (blue).

## 8. Conclusions

This paper contributes to building a relationship between geometric graph-based models and their mechanical simulation, targeting bending-dominant behaviors and 3D deformations. We introduced a novel method to generate a set of patterns via graph enumeration. The pattern generation constitutes a fairly general framework that can be used to produce a large set of configurations from a truly basic input, and this setup can be applied in several contexts. To favor the use of such patterns, we provide a strategy for simulating each one with a reduced model so that the actual patterns can be replaced with them to give instantaneous feedback about their deformation behavior. The model reduction is obtained through an optimization procedure that is valid for a specific material and pattern size.

This contribution has been embedded in a design pipeline starting from a general flat surface, which is then converted into a suitable polygonal mesh. The mesh can embed the patterns since they are tileable and exchangeable. In this setup, any loading and boundary conditions can be applied, and the combination of reduced models provides a quick approximate solution. To facilitate design exploration, we developed a graphical tool to interactively test various examples and possible modifications.

Our findings confirm that the adopted strategy could be useful for exploring the performance of a desired combination of flat patterns under a specific regime of displacement. The model reduction framework exploits a linear model. While this is fast and accurate for small/medium deformations, applications with expected large displacements will not be as accurate. Therefore, based on the target deformation, a different simulation model may be needed (such as [30]).

The model reduction framework we employed is currently used for forward simulation, and we did not design it to be used for inverse design problems. It could be interesting to investigate if the present framework could be adapted for such a purpose, i.e., given a flat polygonal tessellation, to find the optimal pattern distribution to match a target shape under prescribed boundary and loading conditions.

From a user perspective, it is hard to select patterns to obtain a precise mechanical property since the employed parameters are not meaningful as real physical quantities. In fact, an interesting line of further research would be to design a different reduced model based on more controllable explicit mechanical parameters. This would allow us to group patterns based on different macro-mechanical properties providing, for example, for each pattern, alternative shapes which mechanically perform equivalently.

Since in our reduced model calibration pipeline we assume a regular hexagon of a specific size, an error is introduced when applying the reduced models on (possibly deformed) non-hexagonal polygons or polygons of different size. A possible approach for handling this limitation would be to jointly optimize the reduced models for a range of target edge sizes, deformations, and edge numbers.

## Acknowledgments

This research was partially funded by the EU H2020 Programme EVOCATION: Advanced Visual and Geometric Computing for 3D Capture, Display, and Fabrication (grant no. 813170). The authors thank Marco Callieri for the 3D scan of the physical experiments. Gianmarco Cherchi gratefully acknowledges the support to his research by PON R&I 2014-2020 AIM1895943-1.

## References

[1] Hullin, MB, Ihrke, I, Heidrich, W, Weyrich, T, Damberg, G, Fuchs, M. Computational Fabrication and Display of Material Appearance. In: Sbert, M, Szirmay-Kalos, L, editors. Eurographics 2013 - State of the Art Reports. The Eurographics Association; 2013, p. 137–153. doi:10.2312/conf/EG2013/stars/137-153.

[2] Bickel, B, Cignoni, P, Malomo, L, Pietroni, N. State of the art on stylized fabrication. Computer Graphics Forum 2018;37(6):325–342. doi:10.1111/cgf.13327.

[3] Yu, X, Zhou, J, Liang, H, Jiang, Z, Wu, L. Mechanical metamaterials associated with stiffness, rigidity and compressibility: A brief review. Progress in Materials Science 2018;94:114–173. doi:10.1016/j.pmatsci.2017.12.003.

[4] Malomo, L, Pérez, J, Iarussi, E, Pietroni, N, Miguel, E, Cignoni, P, et al. FlexMaps: Computational Design of Flat Flexible Shells for Shaping 3D Objects. ACM Trans Graph 2019;37(6). doi:10.1145/3272127.3275076.

[5] Leimer, K, Musialski, P. Reduced-order simulationx of flexible metamaterials. In: Symposium on Computational Fabrication. SCF '20; Association for Computing Machinery; 2020,doi:10.1145/3424630.3425411.

[6] Martínez, J, Skouras, M, Schumacher, C, Hornus, S, Lefebvre, S, Thomaszewski, B. Star-shaped metrics for mechanical metamaterial design. ACM Trans Graph 2019;38(4). doi:10.1145/3306346.3322989.

[7] Tozoni, DC, Dumas, J, Jiang, Z, Panetta, J, Panozzo, D, Zorin, D. A low-parametric rhombic microstructure family for irregular lattices. ACM Trans Graph 2020;39(4). doi:10.1145/3386569.3392451.

[8] Zhou, S, Jiang, C, Lefebvre, S. Topology-constrained synthesis of vector patterns. ACM Trans Graph 2014;33(6). doi:10.1145/2661229.2661238.

[9] Zehnder, J, Coros, S, Thomaszewski, B. Designing structurally-sound ornamental curve networks. ACM Trans Graph 2016;35(4):1–10. doi:10.1145/2897824.2925888.

[10] Chen, W, Ma, Y, Lefebvre, S, Xin, S, Martínez, J, wang, w. Fabricable tile decors. ACM Trans Graph 2017;36(6):1–15. doi:10.1145/3130800.3130817.

[11] Dumas, J, Lu, A, Lefebvre, S, Wu, J, Dick, C. By-example synthesis of structurally sound patterns. ACM Trans Graph 2015;34(4):1–12. doi:10.1145/2766984.

[12] Martínez, J, Dumas, J, Lefebvre, S, Wei, LY. Structure and appearance optimization for controllable shape design. ACM Trans Graph 2015;34(6):1–11. doi:10.1145/2816795.2818101.

[13] Bickel, B, Bächer, M, Otaduy, MA, Lee, HR, Pfister, H, Gross, M, et al. Design and fabrication of materials with desired deformation behavior. ACM Trans Graph 2010;29(4). doi:10.1145/1778765.1778800.

[14] Panetta, J, Zhou, Q, Malomo, L, Pietroni, N, Cignoni, P, Zorin, D. Elastic textures for additive fabrication. ACM Trans Graph 2015;34(4). doi:10.1145/2766937.

[15] Schumacher, C, Bickel, B, Rys, J, Marschner, S, Daraio, C, Gross, M. Microstructures to control elasticity in 3d printing. ACM Trans Graph 2015;34(4). doi:10.1145/2766926.

[16] Laccone, F, Malomo, L, Pérez, J, Pietroni, N, Ponchio, F, Bickel, B, et al. A bending-active twisted-arch plywood structure: computational design and fabrication of the FlexMaps Pavilion. SN Applied Sciences 2020;2(9):1505. doi:10.1007/s42452-020-03305-w.

[17] Laccone, F, Malomo, L, Pietroni, N, Cignoni, P, Schork, T. Integrated computational framework for the design and fabrication of bending-active structures made from flat sheet material. Structures 2021;34:979–994. doi:10.1016/j.istruc.2021.08.004.

[18] Konaković, M, Crane, K, Deng, B, Bouaziz, S, Piker, D, Pauly, M. Beyond developable: computational design and fabrication with auxetic materials. ACM Trans Graph 2016;35(4):1–11. doi:10.1145/2897824.2925944.

[19] Konaković-Luković, M, Panetta, J, Crane, K, Pauly, M. Rapid deployment of curved surfaces via programmable auxetics. ACM Trans Graph 2018;37(4):1–13. doi:10.1145/3197517.3201373.

[20] Chen, T, Panetta, J, Schnaubelt, M, Pauly, M. Bistable auxetic surface structures. ACM Trans Graph 2021;40(4). doi:10.1145/3450626.3459940.

[21] Bonneau, G, Hahmann, S, Marku, J. Geometric construction of auxetic metamaterials. Computer Graphics Forum 2021;40(2):291–304. doi:doi.org/10.1111/cgf.142633.

[22] Martínez, J, Dumas, J, Lefebvre, S. Procedural voronoi foams for additive manufacturing. ACM Trans Graph 2016;35(4):1–12. doi:10.1145/2897824.2925922.

[23] Martínez, J, Song, H, Dumas, J, Lefebvre, S. Orthotropic k-nearest foams for additive manufacturing. ACM Trans Graph 2017;36(4):1–12. doi:10.1145/3072959.3073638.

[24] Martínez, J, Hornus, S, Song, H, Lefebvre, S. Polyhedral voronoi diagrams for additive manufacturing. ACM Trans Graph 2018;37(4). doi:10.1145/3197517.3201343.

[25] Tricard, T, Tavernier, V, Zanni, C, Martínez, J, Hugron, PA, Neyret, F, et al. Freely orientable microstructures for designing deformable 3D prints. ACM Trans Graph 2020;39(6):1–16. doi:10.1145/3414685.3417790.

[26] Sperl, G, Narain, R, Wojtan, C. Homogenized yarn-level cloth. ACM Trans Graph 2020;39(4). doi:10.1145/3386569.3392412.

[27] Papanicolau, G, Bensoussan, A, Lions, JL. Asymptotic Analysis for Periodic Structures. Elsevier; 1978. ISBN 978-0-08-087526-2.

[28] Kharevych, L, Mullen, P, Owhadi, H, Desbrun, M. Numerical coarsening of inhomogeneous elastic materials. ACM Trans Graph 2009;28(3):1–8. doi:10.1145/1531326.1531357.

[29] Leimer, K, Musialski, P. Analysis of a reduced-order model for the simulation of elastic geometric zigzag-spring meta-materials. Computers & Graphics 2022;102:187–198. doi:10.1016/j.cag.2021.10.007.

[30] Bergou, M, Audoly, B, Vouga, E, Wardetzky, M, Grinspun, E. Discrete viscous threads. ACM Trans Graph 2010;29(4):1–10. doi:10.1145/1778765.1778853.

[31] Pérez, J, Thomaszewski, B, Coros, S, Bickel, B, Canabal, JA, Sumner, R, et al. Design and fabrication of flexible rod meshes. ACM Trans Graph 2015;34(4). doi:10.1145/2766998.

[32] Sakai, Y, Ohsaki, M, Adriaenssens, S. A 3-dimensional elastic beam model for form-finding of bending-active gridshells. International Journal of Solids and Structures 2020;193-194:328–337. doi:10.1016/j.ijsolstr.2020.02.034.

[33] Bauchau, OA, Craig, JI. Euler-Bernoulli beam theory; chap. 5. Springer Netherlands. ISBN 978-90-481-2516-6; 2009, p. 173–221. doi:10.1007/978-90-481-2516-6_5.

[34] Curtin, RR, Edel, M, Prabhu, RG, Basak, S, Lou, Z, Sanderson, C. The ensmallen library for flexible numerical optimization. Journal of Machine Learning Research 2021;22(166):1–6.

[35] Abeykoon, C, Sri-Amphorn, P, Fernando, A. Optimization of fused deposition modeling parameters for improved pla and abs 3d printed structures. International Journal of Lightweight Materials and Manufacture 2020;3(3):284–297. doi:10.1016/j.ijlmm.2020.03.003.

[36] Jakob, W, Tarini, M, Panozzo, D, Sorkine-Hornung, O. Instant field-aligned meshes. ACM Trans Graph 2015;34(6):1–15. doi:10.1145/2816795.2818078.