PROCEEDINGS A

rspa.royalsocietypublishing.org

Research

Article submitted to journal

Subject Areas:
structural engineering, discrete geometry

Keywords:
kirigami, reconfigurable structures, inverse design

Author for correspondence:
Sergio Pellegrino
e-mail: sergiop@caltech.edu

Kirigami Tiled Surfaces with Multiple Configurations

Charles Dorn[1], Robert J. Lang[2] and Sergio Pellegrino[3]

[1]Graduate Aerospace Laboratories, California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA
[2]Lang Origami, Altadena, CA 91001, USA
[3]Graduate Aerospace Laboratories, California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA. orcid.org/0000-0001-9373-3278

This paper presents new kirigami patterns consisting of *tiles* connected by *sub-folds* that can approximate multiple specified target surfaces. The curvature of the surfaces approximated by the tiles varies as the patterns are folded, allowing access to a wide range of curvatures. A numerical framework is developed for the synthesis of the fold patterns that approximate a given set of target surfaces. The pattern synthesis process is framed as a tile placement problem, where compatible tile arrangements associated with each target surface are computed by solving a constrained optimization problem. After computing a set of tile arrangements, sub-folds are added to connect adjacent tiles. The resulting patterns are rigid foldable with many kinematic degrees of freedom, allowing them to achieve configurations that approximate the specified target surfaces. Kinematic simulations verify the existence of continuous paths between the target surfaces. A prototype pattern with six target surfaces is fabricated using 3D printed components.

THE ROYAL SOCIETY PUBLISHING

1. Introduction

Origami and its variants, such as kirigami, offer a platform for transforming a flat sheet into three-dimensional surfaces by folding. Although rigid folding cannot change the intrinsic Gaussian curvature of a sheet, a set of rigid tiles with "hidden connections" lying in a plane, can approximate surfaces with different Gaussian curvature [1]. The inverse problem of synthesizing fold patterns capable of transforming a flat sheet into a surface with specified curvature has attracted significant interest in recent years. An extension of the inverse problem is to create fold patterns capable of achieving surfaces with different curvatures, which has received little attention despite its broad potential for application. This paper presents the synthesis of kirigami fold patterns capable of approximating multiple specified target surfaces, with different Gaussian curvature.

A significant body of work has focused on synthesizing fold patterns capable of realizing curved surfaces from a flat sheet. The early work of Resch [2] [3] proposed several origami patterns that can achieve curved surfaces. Subsequently, various computational methods have been developed for fold pattern synthesis. One means of achieving surfaces with specified curvature is by generalizing patterns such as the Miura-ori [4] [5] and Resch's patterns [6]. General algorithms such as TreeMaker [7] and Tachi's Origamizer [8] can create fold patterns to achieve an arbitrary 3D shape. Alternatively, kirigami patterns, which allow cuts to be made in the sheet, can be designed to realize arbitrary 3D surfaces [9] [10].

The problem of creating fold patterns capable of achieving multiple configurations, sometimes referred to as *pluripotent* patterns, is less well studied. Existing pluripotent patterns rely on switching the mountain-valley fold assignments to achieve multiple target shapes. Universal origami [11] [12] and kirigami [13] patterns exist that can approximate any arbitrary shape, but they are limited to step-approximations of the shapes. Smoother approximations of multiple target surfaces are achieved by [14], where quadrilateral crease patterns are systematically designed to realize multiple target shapes. While a generic quadrilateral mesh is rigid, a generic triangular lattice is not only flexible with many degrees of freedom, but the number of distinct branches of motion increases exponentially with the number of vertices in the pattern [15].

This paper presents a way to design kirigami patterns that can achieve multiple target surfaces, with different Gaussian curvature. The patterns consist of an arrangement of tiles connected by sub-folds and, although the patterns cannot vary their intrinsic curvature, the *apparent surfaces* approximated by the tiles can achieve a wide range of Gaussian curvatures. Unlike existing pluripotent patterns, the physical mechanism for the transformation used in this research is changing the angles of the sub-folds.

The geometric synthesis of these patterns is framed as a tile arrangement problem, involving identical tiles that are required to approximate all of the target surfaces. A set of geometric constraints is formulated to ensure compatibility between the arrangements corresponding to the target surfaces. Once the tile arrangements have been computed, the fold pattern synthesis is completed by adding sub-folds between adjacent tiles in any one of the arrangements. Then, since the tile arrangements corresponding to each target surface are already compatible, the pattern can be reconfigured into all of the target arrangements. The resulting patterns are rigidly foldable with many kinematic degrees of freedom, allowing rigid folding paths between the target surface configurations. Kinematic simulations are presented to demonstrate the reconfiguration between the target surfaces. The fold patterns have a relatively simple geometry, which is desirable from a manufacturing standpoint. A demonstration prototype is constructed that can achieve six target shapes. The prototype is built from 3D printed plates connected by steel pin hinges.

The patterns proposed in this work open the door to applications that require multiple operating configurations. Origami-inspired structures that use folding to morph between two target configurations have been developed for medical stents [16] and deployable structures [17]. There is a wide range of applications for morphing structures that can adapt to different operating conditions. Examples include antennas [18] [19], optical sensors [20], and other electronics [21].

The remainder of the paper is organized as follows: Section 2 introduces the general concept of the proposed patterns and Section 3 discusses the mechanism for changing apparent curvature. Constraints on gaps between tiles are derived in Section 4. Section 5 presents the pattern synthesis method. Section 6 explores pattern connectivity followed by examples in Section 7. A prototype pattern is presented in Section 8. Finally, Section 9 concludes the paper.

2. Preliminaries

Given a set of target surfaces, Fig. 1(a), consider a set of *tiles* connected by *sub-folds*, Fig. 1(b) where the tiles are highlighted in green and the sub-folds in gray.

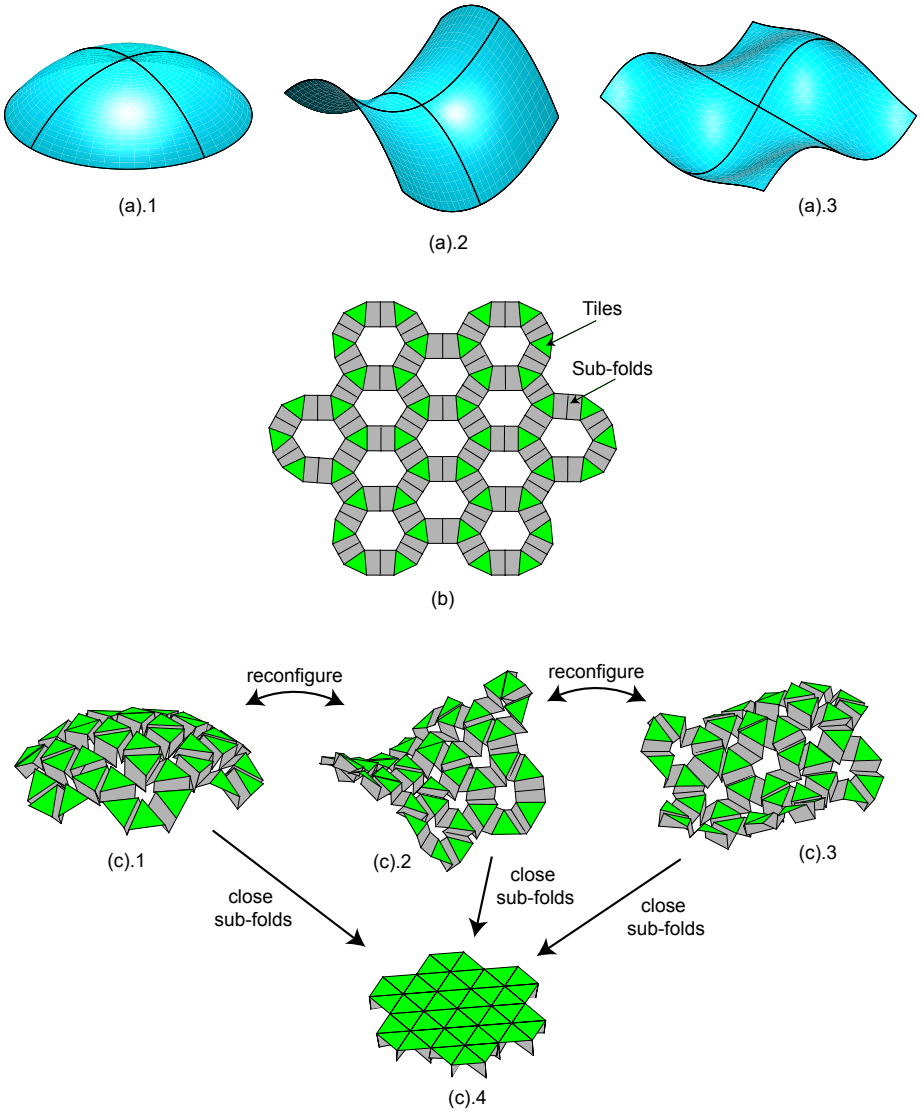


Figure 1: (a) Target surfaces with different Gaussian curvature; (b) Kirigami pattern that can approximate these surfaces; (c) reconfigurations of kirigami pattern into target surfaces and a flat configuration.

This chosen architecture results in loops of tiles connected by sub-folds, and allows the pattern to approximate a wide range of surfaces. The *apparent surface* is defined as the surface tangent to the tiles, i.e. disregarding the sub-folds. The pattern of tiles and sub-folds is designed such that varying the sub-fold angles allows the apparent surface to approximate each target surface, Fig. 1(c). By closing the sub-folds, a flat apparent surface is achieved where the edges of adjacent tiles are coincident. Note that, although in the present paper it has been chosen to design patterns that are flat when all sub-folds are closed, different choices could also be made.

The presence of holes inside the loops plays an important role in the choice of this pattern, by introducing many kinematic degrees of freedom (DOF), and allowing a rich configuration space accessible by rigid folding of the sub-folds. Including holes in a pattern increases the number of boundary edges in the pattern and increases the degrees of freedom. The equation for the number of internal kinematic degrees of freedom of a general fold pattern was derived by Tachi [22]:

$$\text{DOF} = B - 3H + S - 3 - \sum_{k\geq 4}(k-3)P_k, \tag{2.1}$$

where B is the number of edges on the boundary, H is the number of holes, S is the number of redundant constraints (states of self stress), and P_k is the number of k-gon faces in the pattern.

This paper considers patterns with triangular or square tiles that are connected by trapezoidal sub-folds. For such patterns, Eq. (2.1) can be expressed as

$$\text{DOF} = B - 3H + S - 3 - 2N_g - N_{t,sq}, \tag{2.2}$$

where N_g is the number of sub-folds and $N_{t,sq}$ is the number of square tiles in the pattern.

Consider, for example, a loop of four square tiles connected by sub-folds, as shown in Fig. 2. There are $B = 24$ boundary edges (16 on the external boundary and 8 on the internal boundary), $H = 1$ hole, $N_g = 4$ sub-folds, and $N_{t,sq} = 4$ square tiles. Generically, there are no states of self stress ($S = 0$) in the loop, which can be verified using the kinematic model in Appendix A. Thus, Eq. (2.2) gives:

$$\text{DOF} = 24 - 3 \times 1 + 0 - 3 - 2 \times 4 - 4 = 6 \tag{2.3}$$

and hence the loop has 6 internal degrees of freedom. Note that the presence of the hole has increases the number of degrees of freedom since there are 8 boundary edges along the hole, which is larger than the $-3H$ term associated with introducing the hole. In other words, if the hole was filled in, the DOF would be reduced.

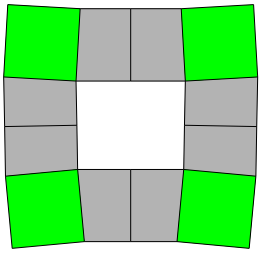


Figure 2: A loop of square tiles connected by sub-folds with 6 kinematic degrees of freedom.

3. Intrinsic vs. Apparent Curvature

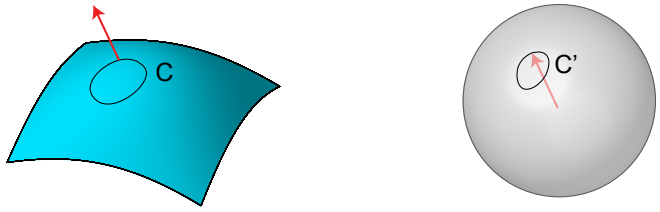
Gauss's Theorema Egregium states that the Gaussian curvature of a surface is invariant under isometric transformations [23], which applies generally to both smooth and discrete surfaces. In the context of discrete surfaces, rigid folding is an isometric transformation and hence the discrete Gaussian curvature of a fold pattern is an intrinsic property, which cannot be changed through folding.

However, the patterns presented in this work are capable of approximating surfaces with different Gaussian curvature, which motivates the distinction between the *intrinsic* and the *apparent* discrete Gaussian curvature of a fold pattern.

Before defining the discrete curvature, the Gaussian curvature of a smooth surface can be defined by considering the Gauss map of a contour on the surface. Consider a closed loop C on a smooth surface, as shown in Fig. 3a. The Gauss map of C is constructed by mapping the surface normal vector along C onto the unit sphere, shown in Fig. 3b. Then, the Gaussian curvature at a point P on the surface is defined as the ratio of the spherical area $A(C')$ enclosed by C' on the Gauss map to the area $A(C)$ enclosed by C on the surface, as C approaches P [23]:

$$K_P = \lim_{C \to P} \frac{A(C')}{A(C)}. \tag{3.1}$$

Note that $A(C')$ is the signed area and the direction of the mapping matters.



(a) (b)

Figure 3: Definition of the Gauss map. (a) Contour C on a smooth surface, with the normal vector shown at one point. (b) Gauss map C' of the contour with the corresponding normal vector at one point shown.

The definition of Gaussian curvature in Eq. (3.1) does not directly apply to discrete surfaces; since the curvature of a discrete surface is concentrated at points, the ratio of $A(C')$ to $A(C)$ is singular in the limit of a shrinking contour. This motivates the definition of the *discrete Gaussian curvature* K as the area on the Gauss map of the trace of a closed loop on the surface:

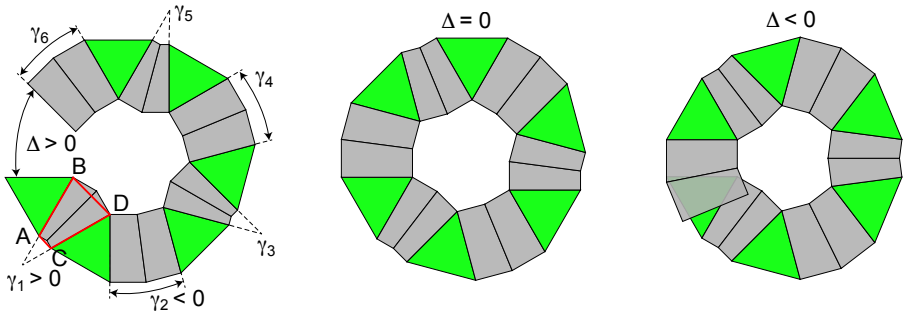
$$K = A(C'). \tag{3.2}$$

A discussion of the discrete Gaussian curvature can be found in [24] and Gauss maps in the context of origami are discussed in [25]. This definition of the discrete Gaussian curvature is meaningful even for discrete surfaces with holes, which is relevant to the patterns presented in this paper. An example of the Gauss map of a set of flat tiles forming a loop is shown in Fig. 4. Note that this definition of discrete Gaussian curvature is an integrated quantity, not a point-wise measure of curvature. Reference [24] discusses estimates of point-wise curvature from discrete curvature.

The discrete Gaussian curvature of Eq. (3.2) can be used to define the intrinsic curvature of the presented kirigami patterns by examining the loops in a pattern. The area enclosed by the Gauss map of a discrete loop is equal to the angular defect Δ of the loop [26], which is obtained by cutting and flattening the loop, as shown in Fig. 5 for example loops of tiles connected by sub-folds with positive, zero, and negative intrinsic curvature. The case with zero intrinsic curvature, shown in Figure 5b, is *developable* since it can be flattened without cutting. Developability is often desirable for manufacturing, since many fabrication techniques involve cutting an initially flat sheet.

(a) (b)

Figure 4: Gauss map of a loop on a discrete surface. (a) Discrete loop C, with the normal vector of one face shown. (b) Gauss map C' of the loop, with the corresponding normal vector of one face shown.



(a) (b) (c)

Figure 5: Loops of tiles with different intrinsic curvature. (a) Cut and flattened loop with positive intrinsic curvature. The isosceles trapezoid formed by gap 1 is highlighted in red. (b) Flattened loop with zero intrinsic curvature (developable). (c) Cut and flattened loop with negative intrinsic curvature.

It will be shown in Section 4 that for all of the kirigami patterns developed in the present study the *edges of adjacent tiles are coplanar in all configurations and form isosceles trapezoids*. These trapezoidal gaps, as well as the tiles themselves, define the intrinsic and apparent Gaussian curvatures of a loop.

For the flattened loops shown in Figure 5 the intrinsic Gaussian curvature can be obtained by considering the tiles as well as the geometry of the gaps between the tiles. In Figure 5a, the trapezoid for gap 1 is defined by the tile vertices A, B, C, and D, highlighted in red. The orientation of this gap determines how it contributes to the angular defect Δ of the loop. If the short side of the trapezoid lies on the outside of the loop, the gap contributes positively to the angular defect, which is the case for the highlighted trapezoid. If the gap is a rectangle, the contribution to Δ is zero. Finally, if the short side of the trapezoid lies on the inside of the loop, its contribution to Δ is negative.

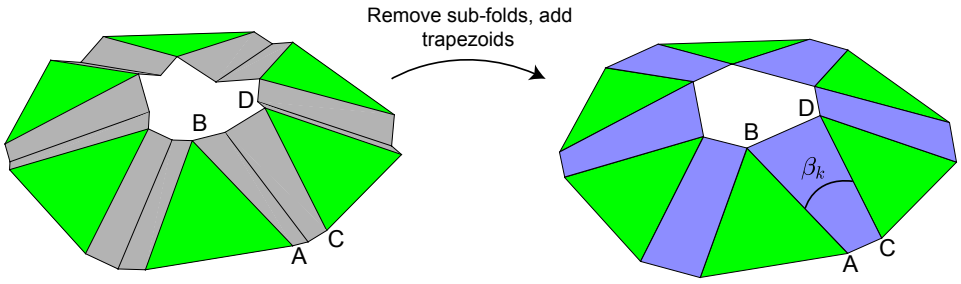
For loops with six equilateral triangle tiles the angular defect is zero, and hence the intrinsic discrete Gaussian curvature of the loop, K, is related only to the six gaps and can be obtained from:

$$K = \Delta = \sum_{k=1}^{6} \gamma_k \tag{3.3}$$

where γ_k is the angle of the isosceles trapezoid corresponding to gap k, as defined in Fig. 5. As previously noted, the sign of each angle depends on the orientation of the trapezoid: $\gamma_k > 0$ if $||\boldsymbol{BD}|| > ||\boldsymbol{AC}||$ and $\gamma_k < 0$ if $||\boldsymbol{BD}|| < ||\boldsymbol{AC}||$.

Therefore, to form a developable loop, either all of the gaps must be rectangles, in which case $\gamma_k = 0\ \forall k$, or there must be present gaps of both orientations, such that the sum of the angles is zero. This is the case of greatest interest, because it allows a range of apparent curvatures to be achieved, and is discussed next.

While the intrinsic discrete Gaussian curvature is based on the angles of the sub-folds, and hence is invariant, the *apparent discrete Gaussian curvature* is the curvature of the surface formed by the tiles joined by trapezoids across the gaps, disregarding the sub-folds. Figure 6a shows a loop of tiles connected by sub-folds. In Fig. 6b the sub-folds have been removed and the resulting gaps have been filled with isosceles trapezoids. These added faces are highlighted in blue and the points A, B, C, D that correspond to the tile edges for one gap are labeled.



(a) (b)

Figure 6: The apparent curvature is defined after filling the gaps between tiles with trapezoidal faces, highlighted in blue.

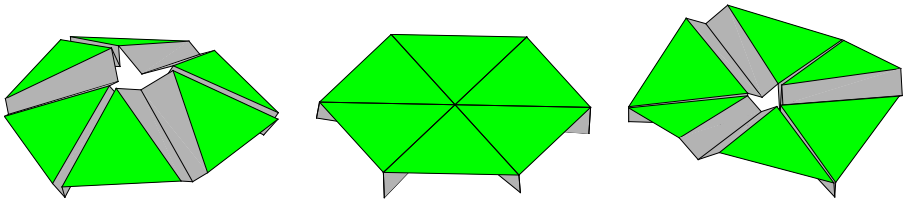
The apparent angular defect Δ_a of a loop of tiles, connected by the blue trapezoids, corresponds to the apparent discrete Gaussian curvature K_a of the loop and is defined by a natural extension of the intrinsic curvature in Eq. (3.3):

$$K_a = \Delta_a = \sum_{k=1}^{6} \beta_k \tag{3.4}$$

where β_k is the angle of the trapezoid corresponding to gap k, which is defined in Figure 6. The sign of β_k is positive for trapezoids with the shorter side out and negative for the opposite orientation.

Note that the sign of β_k matches the sign of γ_k for all values of k and, therefore, each gap contributes either positively or negatively to both the intrinsic and the apparent curvature. However, the magnitude of β_k varies as the angle of the sub-fold is changed. Therefore, the apparent curvature can be changed by varying the sub-fold angles. Furthermore, if a loop contains gaps of both orientations, the apparent curvature can assume both positive and negative values.

Figure 7 shows the same loop with positive, zero, and negative apparent curvature, illustrating the mechanism of apparent curvature change. This loop contains three gaps of each orientation. In Fig. 7a, the gaps with $\beta_k < 0$ are nearly closed ($\beta_k \approx 0$) while the gaps with $\beta_k > 0$ are kept open, so the sum of the gap angles, which is the apparent curvature, is positive. If all sub-folds are closed, then $\beta_k = 0$ for all gaps and a zero apparent curvature is obtained, as shown in Fig. 7b. Finally, Fig. 7c shows a configuration where the gaps with $\beta_k < 0$ are open, leading to a negative apparent curvature.



(a) (b) (c)

Figure 7: Three configurations of the same loop of tiles, with apparent curvature changes achieved by varying the sub-fold angles. (a) Positive , (b) zero and (c) negative apparent curvatures.

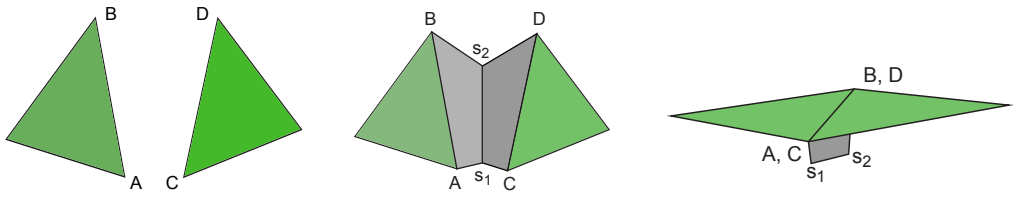
This mechanism for curvature change can be replicated for patterns consisting of many loops, which can be used to design surfaces with multiple curvatures. Each individual loop in the pattern can vary its apparent curvature, allowing a wide range of apparent surfaces, with different curvatures, to be created.

4. Gap Conditions

The geometry of the gaps between the tiles has to satisfy two types of compatibility conditions. The first condition ensures that a sub-fold fits between a pair of adjacent tiles, such that the edges of the tiles become coincident when the sub-fold is closed. The second condition ensures the compatibility of two gaps, meaning that the same sub-fold can span both gaps by varying only the fold angle. These conditions on the gaps between two tiles are used, in Section 5, to constrain the tile arrangement synthesis.

(a) Single sub-fold conditions

Consider two flat rigid tiles in three-dimensional space. The relative position of the tiles is such that a single sub-fold can be fitted in the gap between the tiles and when the sub-fold is closed the edges of the tiles become coincident. Figure 8 shows an example of tiles that satisfy these conditions, when the sub-fold is fully closed and AB and CD coincide, Fig. 8c.



(a) (b) (c)

Figure 8: Two tiles with a gap that satisfies the single sub-fold conditions. (a) Definition of edges AB and CD (top view). (b) A single sub-fold (gray) connecting edges AB and CD (top view). (c) When the sub-fold is closed, edges AB and CD become coincident (perspective view).

To develop general constraints on the relative positions of the tiles, it is required that points C and D coincide with points A and B after a rotation about some axis s through an angle ϕ, which closes the sub-fold. Consider an arbitrary point s_1 on the axis of rotation s, the length of vector $\boldsymbol{A}\mathbf{s}_1$ from A to s_1 and the length of vector $\boldsymbol{C}\mathbf{s}_1$ must be equal since these segments coincide when

the sub-fold is closed. The same argument can be made for a point s_2, leading to the conditions:

$$\begin{aligned} ||\boldsymbol{A}\mathbf{s}_1|| &= ||\boldsymbol{C}\mathbf{s}_1|| \\ ||\boldsymbol{B}\mathbf{s}_2|| &= ||\boldsymbol{D}\mathbf{s}_2||. \end{aligned} \tag{4.1}$$

Therefore, s must lie in the planes P_1 and P_2, which perpendicularly bisect line segments AC and BD, respectively.

There are three possible cases concerning the planes P_1 and P_2, as shown in Fig. 9. In case 1, P_1 and P_2 intersect along a line. In this case, the only candidate axis of rotation is the line of intersection of the two planes. However, different rotation angles ϕ are generally required to bring C to A and D to B. In case 2, planes P_1 and P_2 are parallel, and hence there is no possible rotation axis. Finally, in case 3 planes P_1 and P_2 are coincident, and hence there are infinitely many candidate rotation axes, which is the case of interest.

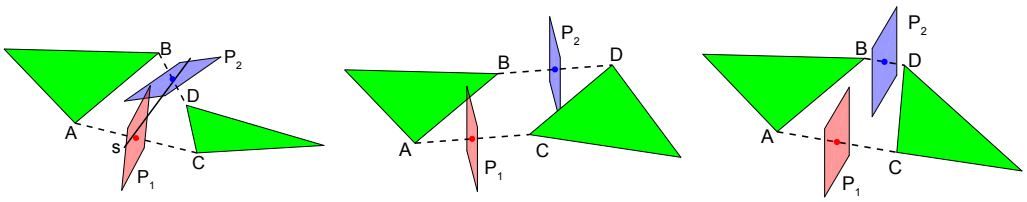


Figure 9: Three cases of a pair of tiles in 3D space. (a) Case 1: planes P_1 and P_2 intersect along a unique line s. (b) Case 2: planes P_1 and P_2 do not intersect. (c) Case 3: planes P_1 and P_2 coincide.

Two conditions are needed to ensure that P_1 and P_2 coincide. First, points $A,\ B,\ C$, and D must be coplanar, which can be enforced by requiring the triple product of three vectors meeting at vertex A is zero:

$$\boldsymbol{BA} \times \boldsymbol{CA} \cdot \boldsymbol{DA} = 0. \tag{4.2}$$

This condition requires the volume of the parallelepiped generated by $\boldsymbol{BA}, \boldsymbol{CA}$, and $\boldsymbol{DA}$ is zero. Equation (4.2) could be equivalently expressed in terms of three vectors meeting at points B, C, or D.

The second condition arises from symmetry. Within the plane of points $A,\ B,\ C$, and D, the edges AB and CD must be mirror symmetric with respect to the bisecting plane. Therefore $A,\ B,\ C$, and D form an isosceles trapezoid, which is equivalent to stating:

$$||\boldsymbol{AD}|| = ||\boldsymbol{BC}||. \tag{4.3}$$

The conditions in Eqs. (4.2) and (4.3) ensure that the edges AB and CD can be made to coincide through the rotation of a sub-fold.

(b) Gap compatibility

Two gaps are defined as compatible if they can be obtained by rotation of the same sub-fold. In the pattern synthesis process, this will be the key condition that allows multiple shapes to be achieved with the same fold pattern.

Consider a pair of tiles connected by a sub-fold. Figure 10a shows a top view of the same pair of tiles, with two different sub-fold angles. To establish a condition on A, B, C, D, C' and D' that ensures the compatibility of Gap 1 and Gap 2, consider the perpendicular projections of Gap 1 and Gap 2 onto a plane perpendicular to the sub-fold axis, s, Fig. 10c. The solid lines from s to A and from s to B, and from s to C and s to D in Gap 1 are the projections of the faces of the

sub-fold. Similarly, for Gap 2 the solid lines are the projections of the sub-fold faces. The dotted lines representing segments AC, BD, AC', and BD' are true lengths, since they are parallel to the projection plane.

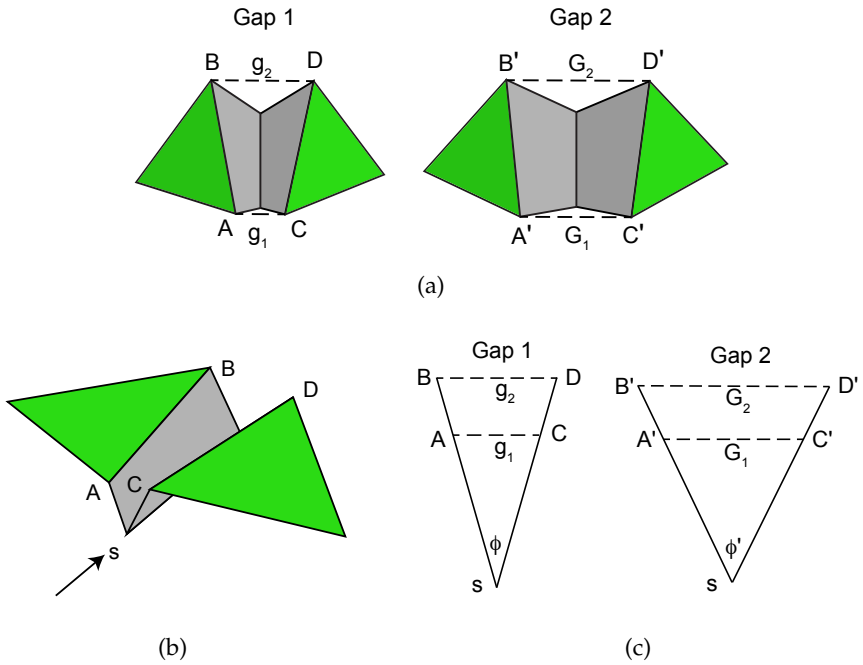


Figure 10: Diagrams for deriving gap compatibility constraints. (a) Two gaps from the top view. (b) Perspective view one gap. (c) Diagram of the sub-fold viewed along the sub-fold axis, looking along the arrow shown in (b). Note that the dashed lines are true length as they are parallel to the viewing plane. Solid lines are projections onto the viewing plane.

By examining Gap 1 in Figure 10c, it is observed that

$$\sin\left(\frac{\phi}{2}\right) = \frac{g_2 - g_1}{2||\boldsymbol{AB}||_p} \tag{4.4}$$

$$\sin\left(\frac{\phi}{2}\right) = \frac{g_2}{2||\boldsymbol{sB}||_p} \tag{4.5}$$

where $||\ ||_p$ denotes the length of the projection. Similarly, by examining Gap 2,

$$\sin\left(\frac{\phi'}{2}\right) = \frac{G_2 - G_1}{2||\boldsymbol{AB}||_p} \tag{4.6}$$

$$\sin\left(\frac{\phi'}{2}\right) = \frac{G_2}{2||\boldsymbol{sB}||_p}. \tag{4.7}$$

Assuming that Gap 1 and Gap 2 are two different configurations of the same sub-fold, for different angles ϕ and ϕ', the projected lengths $||\boldsymbol{AB}||_p$ and $||\boldsymbol{sB}||_p$ are the same for Gap 1 and Gap 2. By combining Eqs. (4.4)- (4.7), the following relation between the two gaps is obtained:

$$\frac{g_2 - g_1}{g_2} = \frac{G_2 - G_1}{G_2}. \tag{4.8}$$

The condition in Eq. (4.8) is equivalent to the statement that $\frac{g_2-g_1}{g_2}$ is invariant as the sub-fold changes angle. Equation (4.8) can be used as a condition on any two gaps, to ensure that they can be realized with the same sub-fold.

5. Pattern Synthesis

The objective is to compute a single tile arrangement that can be reconfigured into multiple target surfaces with different Gaussian curvature, by varying the the sub-fold angles. The tiles are simultaneously arranged on all of the target surfaces, while ensuring that the following conditions are satisfied for all target configurations:

(i) Adjacent tiles can be connected by a single sub-fold (Eqs (4.2) and (4.3)).
(ii) Corresponding gaps between tiles in each target configuration are compatible (Eq. (4.8)).

An overview of the pattern synthesis process is presented in Fig. 11. First, a set of initial guesses of the tile arrangements is generated, for each target surface. The initial guesses can be generated systematically using any discretization of the target surfaces. A flat target surface is included, to achieve a developable pattern. The constrained optimization problem uses the initial guesses as a starting point, and outputs valid arrangements of the tiles on each target surface. Then, the sub-folds are added to the arrangement corresponding to the flat target surface, completing the developable fold pattern.

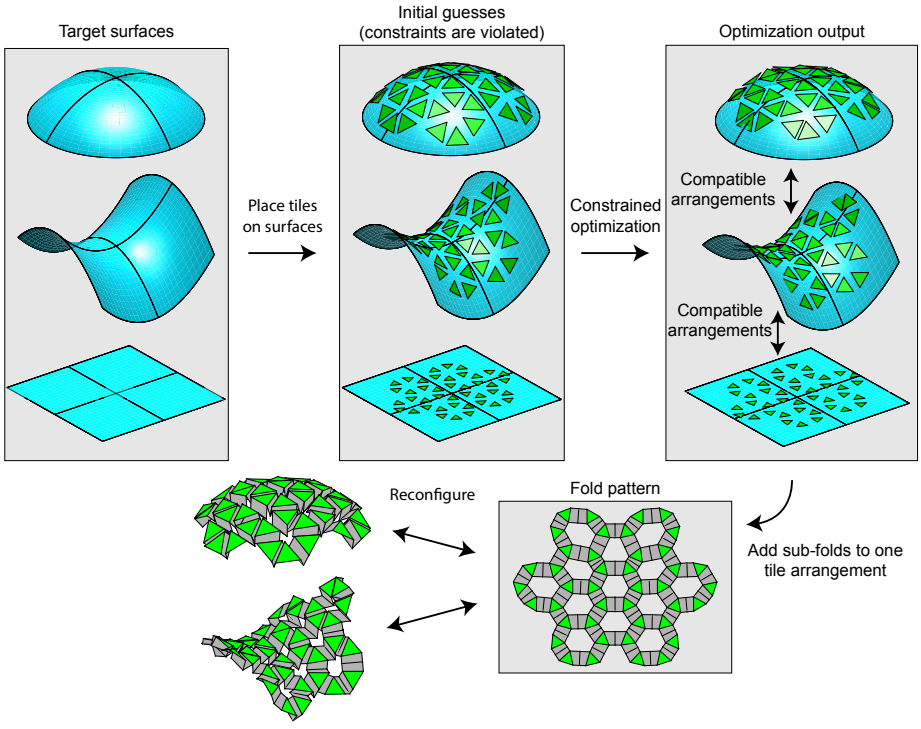


Figure 11: Overview of pattern synthesis process. First, an initial guess is created by arranging tiles on each target surface. By solving a constrained optimization problem, compatible tile arrangements corresponding to each target surface are computed. Then, sub-folds are added to connect adjacent tiles in one of the tile arrangements. By adding sub-folds to a flat target surface arrangement, a developable pattern is created that can be reconfigured to achieve the other tile arrangements.

(a) Constrained optimization problem

The set of tile arrangements is parameterized by

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_j \\ \vdots \\ \boldsymbol{v}_{N_s} \end{bmatrix} \tag{5.1}$$

where $\boldsymbol{V} \in \mathbb{R}^{3n_{vt}N_tN_s}$ is a vector of tile vertex coordinates in all of the N_s arrangements. The vector $\boldsymbol{v}_j \in \mathbb{R}^{3n_{vt}N_t}$ contains the vertex coordinates of an arrangement of N_t identical tiles corresponding to the target surface $T_j(x, y) : \mathbb{R}^2 \to \mathbb{R}$. Various tile shapes can be considered and the number of vertices per tile is n_{vt}. There are N_s tile arrangements, each corresponding to one of the N_s target surfaces. Each arrangement of tiles has N_g gaps, which determines the connectivity of the tiles.

To generate a set of tile arrangements, a constrained optimization problem is posed to minimize a cost function E subject to a set of equality constraints:

$$\min_{\boldsymbol{V}} E(\boldsymbol{V}) \quad \text{subject to} \quad C_{tiles}(\boldsymbol{V}),\ C_{gaps}(\boldsymbol{V}),\ C_{target}(\boldsymbol{V}) \tag{5.2}$$

where the choice of E depends on the specific application. A suitable option for the cost function is proposed in Section 5c.

The constraints have been divided into three sets: C_{tiles}, C_{gaps}, C_{target}. For brevity of notation, it is omitted that all constraints are functions of $\boldsymbol{V}$.

The first set of constraints, C_{tiles}, defines the shape of the tiles. In the case of equilateral triangle tiles, denote by L_{mj} the length of tile edge m in arrangement j. The length of all tile edges in all arrangements is set equal to L_0 and hence,

$$C_{tiles}^{triangle} = \{L_{mj} = L_0, \quad m = 1, ..., 3N_t, \ \ j = 1, ..., N_s\}. \tag{5.3}$$

Additional constraints are needed to define square tiles. Again, the length of edge m in arrangement j is set equal to L_0. The length of the diagonal L_{ij}^D of each tile i in arrangement j is set equal to $\sqrt{2}L_0$. An additional constraint is added to enforce the coplanarity of the four vertices of each tile. Therefore, for square tiles with edge length L_0 and corner points (q_1, q_2, q_3, q_4), the constraint set is:

$$C_{tiles}^{square} = \{L_{mj} = L_0, \qquad m = 1, ..., 4N_t, \ \ j = 1, ..., N_s \tag{5.4}$$

$$L_{ij}^D = \sqrt{2}L_0, \qquad i = 1, ..., N_t, \ \ j = 1, ..., N_s \tag{5.5}$$

$$(\mathbf{q}_2\mathbf{q}_1 \cdot \mathbf{q}_3\mathbf{q}_1 \times \mathbf{q}_4\mathbf{q}_1)_{ij} = 0, \qquad i = 1, ..., N_t, \ \ j = 1, ..., N_s\}. \tag{5.6}$$

Equations (5.4) and (5.5) define the side lengths and diagonal length of the square, and Eq. (5.6) enforces the planarity of the tile. Other tile shapes would be formulated in a similar way.

The second set of constraints, C_{gaps}, ensures a valid set of gaps in all tile arrangements. Following Section 4, C_{gaps} ensures that a single sub-fold can be inserted in the gap between pairs of adjacent tiles and that corresponding gaps in each of the target surface tile arrangements are compatible. The gap constraints can be written as

$$C_{gaps} = \{\, (\boldsymbol{BA} \times \boldsymbol{CA} \cdot \boldsymbol{DA})_{kj} = 0, \qquad k = 1, ..., N_g, \ \ j = 1, ..., N_s, \tag{5.7}$$

$$(||\boldsymbol{AD}|| = ||\boldsymbol{BC}||)_{kj}\,, \qquad k = 1, ..., N_g, \ \ j = 1, ..., N_s, \tag{5.8}$$

$$\left(\frac{g_{2k} - g_{1k}}{g_{2k}}\right)_j = \left(\frac{g_{2k} - g_{1k}}{g_{2k}}\right)_{j=1}, \qquad k = 1, ..., N_g, \ \ j = 2, ..., N_s\} \tag{5.9}$$

where k indexes the gaps in target arrangement j. Equations (5.7) and (5.8) enforce the condition that a single sub-fold fits to each gap. Equation (5.9) ensures the compatibility of corresponding

gaps in the arrangement of the tiles for each target surface. The third set of constraints, C_{target}, ensures that each target surface is closely approximated by its corresponding tile arrangement configuration. Sections 5b-5c, below, discuss appropriate target surface constraints.

(b) Ideal constraints

A tile arrangement that achieves the closest possible approximation of the target surfaces is defined as *ideal*; each tile is tangent to the target surface at the tile center. Then, the constraint set C_{target}^{ideal} is defined to strictly enforce that the arrangements are all ideal.

The centroid coordinates of tile i in target arrangement j are $(\bar{x}_{ij}, \bar{y}_{ij}, \bar{z}_{ij})$. The unit normal vector of the tile is $\hat{\mathbf{n}}_{ij}^t \in \mathbb{R}^3$. The target surface j, defined by the function $T_j(x, y)$, has unit normal vector $\hat{\mathbf{n}}_j^s(x, y) \in \mathbb{R}^3$. In this work, it is assumed that analytical expressions for the target surfaces are known, although similar constraints could be formulated in terms of a set of numerically defined target surfaces. The constraints that force all arrangements to be ideal approximations of the target surfaces are:

$$C_{target}^{ideal} = \{\bar{z}_{ij} = T_j(\bar{x}_{ij}, \bar{y}_{ij}), \qquad i = 1, ..., N_t, \quad j = 1, ..., N_s \tag{5.10}$$

$$\theta_{ij} = 0, \qquad i = 1, ..., N_t, \quad j = 1, ..., N_s\} \tag{5.11}$$

where the angle θ_{ij} is the angle between the unit normal vector $\hat{\mathbf{n}}_{ij}^t$ of tile i in arrangement j and the unit normal vector $\hat{\mathbf{n}}_j^s$ of surface T_j, evaluated at the tile center:

$$\theta_{ij} = \cos^{-1}\left(\hat{\mathbf{n}}_{ij}^t \cdot \hat{\mathbf{n}}_j^s(\bar{x}_{ij}, \bar{y}_{ij})\right). \tag{5.12}$$

The first constraint, Eq. (5.10), ensures that the tile centers lie on the target surface. Equation (5.11) ensures that the tile normal is aligned with the surface normal at the center of the tile, so that the tile is tangent to the surface.

Although it is difficult to make general statements about the existence of feasible solutions to systems of nonlinear equations, a constraint and free variable counting argument can be used to determine if the system is generically overdetermined or underdetermined. When subject to the ideal constraints, each tile effectively has three free parameters to fix its orientation in space. The centroid coordinates $\bar{x}$ and $\bar{y}$ of each tile are free, but $\bar{z}$ follows from Eq. (5.10). If Eq. (5.11) is satisfied, the only remaining free parameter is a rotation about the tile normal vector. Therefore, considering all N_t tiles in N_s arrangements, there are effectively $3N_sN_t$ free variables. Since C_{tiles} and C_{target} have already been accounted for while counting the free variables, the number of free variables can be compared to the number of gap constraints. From Eqs. (5.7) - (5.9), there are $(3N_s - 1)N_g$ gap constraints. If the number of gap constraints is greater than the number of free variables, the system is overdetermined and in general there will be no solution.

Table 1 summarizes the number of free variables versus gap constraints if the ideal target constraints are enforced. The freedom in the tile arrangement problem, denoted by F, is defined as the number of free variables minus the number of gap constraints. If $F > 0$, the problem is underdetermined and feasible solutions should exist, while $F < 0$ indicates that the problem is overdetermined.

Whether or not the system is overdetermined depends on the connectivity of the tiles, particularly the number of tiles N_t, the number of gaps N_g, and the number of target surfaces N_s. Section 6 analyzes the freedom in the tile arrangement problem for tiles with different connectivity. The analysis suggests that the ideal target constraints are often too restrictive and tend to over-constrain the system. This motivates the relaxation of some target constraints, discussed in the next sub-sections.

(c) Relaxed constraints and cost function

Depending on the connectivity of the tiles, the ideal target constraints may over-constrain the tile arrangement problem. In such cases it is of interest to consider relaxing the constraints to obtain feasible solutions. A relaxed target constraint set is proposed that maintains the constraint that

Table 1: Count of free variables and constraints for three target constraint cases.

	Free variables	Gap constraints	F = Free variables − Gap constraints
Ideal target constraints	$3N_sN_t$	$(3N_s-1)N_g$	$3N_sN_t-(3N_s-1)N_g$
Relaxed target constraints	$5N_sN_t$	$(3N_s-1)N_g$	$5N_sN_t-(3N_s-1)N_g$
Relaxed target constraints and developability	$5(N_s-1)N_t+3N_t$	$(3N_s-1)N_g$	$5(N_s-1)N_t+3N_t-(3N_s-1)N_g$

the tile center must lie on the target surface, but relaxes the constraint that the tile normal must be aligned with the surface normal. The resulting constraint set is then,

$$C_{target}^{relax}=\{\bar{z}_{ij}=T_j(\bar{x}_{ij},\bar{y}_{ij}),\quad i=1,...,N_t,\ \ j=1,...,N_s\}. \tag{5.13}$$

A similar constraint counting argument to that presented in Section (b) can be applied to the relaxed constraints. The placement of each tile has five free variables: two free variables to describe the tile center coordinates $\bar{x}$ and $\bar{y}$ and three angular orientation variables, since the tile normal vector is no longer constrained. The count of free variables and gap constraints for the relaxed target constraint case is included in Table 1.

Since the tile normal vectors are no longer required to be aligned with the surface normal vector, the deviation between the tile and the surface normals can be incorporated in the cost function, in Eq. (5.2). A suitable cost function is:

$$E=E_{normal}+wE_{gap}. \tag{5.14}$$

Here, the first term in the cost function is

$$E_{normal}=\left|\left|\left[\theta_{11},...,\theta_{ij},...,\theta_{N_tN_s}\right]\right|\right|, \tag{5.15}$$

which is the norm of a vector whose entries are the angles between the tile normals and the surface normals, θ_{ij}, defined in Eq. (5.12), for all tiles and in all target arrangements. This term penalizes the misalignment between the tiles and the target surface. However, including only E_{normal} in the cost function would allow optimal solutions with undesirable tile spacing. For example, some gaps between adjacent tiles could be very large and others very small. To address this issue, a second term, E_{gap}, has been introduced in Eq. (5.14) to control the gap sizes. This term is defined as:

$$E_{gap}=||\boldsymbol{G}-\boldsymbol{G}_{tar}||, \tag{5.16}$$

where $\boldsymbol{G}\in\mathbb{R}^{2N_gN_s}$ is a vector of gap lengths g_1 and g_2 for each gap of each arrangement. It is given by:

$$\boldsymbol{G}=\left[g_{111},\ g_{211}\ \ ...,\ \ g_{1kj},\ g_{2kj},\ \ ...,\ \ g_{1N_gN_s},\ g_{2N_gN_s}\right], \tag{5.17}$$

where k indexes gaps of arrangement j. Therefore, the term E_{gap} penalizes the deviation from a specified vector of target gap sizes $\boldsymbol{G}_{tar}\in\mathbb{R}^{2N_gN_s}$. The relative weighting between the two terms in the cost function in Eq. (5.14) is the parameter w.

(d) Including the developability target

A developable pattern can be obtained by including a flat surface among the targets. However, it should be noted that in the case of the relaxed target constraints, the tile arrangement corresponding to the flat target surface may not be exactly planar. Therefore, a third set of target constraints $C_{target}^{relax,dev}$ is proposed that treats the flat target surface with ideal constraints and

all other target surfaces with relaxed target constraints. Let target surface 1 be flat; the relaxed developable constraint set is then:

$$C_{target}^{relax,dev} = \{\bar{z}_{ij} = T_j(\bar{x}_{ij}, \bar{y}_{ij}), \qquad i = 1, ..., N_t, \;\; j = 1, ..., N_s \tag{5.18}$$

$$\theta_{ij} = 0, \qquad i = 1, ..., N_t, \;\; j = 1\}. \tag{5.19}$$

Again, the free variables and gap constraints can be counted. By requiring that tile arrangement 1 is satisfied exactly (ideal target), there are $3N_t$ free variables associated with the tiles in arrangement 1 and $5N_t(N_s - 1)$ free variables associated with the other target arrangements. The count of free variables versus gap constraints is presented in Table 1.

Since the flat target arrangement is restricted to lie exactly in the plane, a developable pattern can be made by adding sub-folds to a flat tile arrangement. This results in a planar configuration that defines the fold pattern.

(e) Implementation

The constrained optimization problem of Eq. (5.2) can be implemented numerically to compute the optimal arrangements that satisfy the constraints. Standard nonlinear optimization algorithms can be used to perform the optimization. The *fmincon* utility in Matlab was used to compute optimal tile arrangements in the examples presented in this paper, and the specific codes that were developed are included in the supplementary material.

First, the type of tile, the number of tiles and their connectivity are chosen. Then, depending on the chosen connectivity and whether or not the pattern is to be developable, the appropriate target constraint set is chosen, following Sections 5b-5d.

Next, the numerical part of the pattern synthesis process begins. An initial guess of the tile arrangements for each target surface is made. Since the initial guess may violate the constraints, it can be created by placing tiles on any arbitrary discretizations of the target surfaces. First, a uniform grid of equilateral triangles or squares, depending on the tile shape, is created in the xy-plane. A discretization of each target surface can be created by mapping the vertices of that grid to each target surface $T_j(x, y)$. Then, a tile is placed in the center of each face of the discretized target surface to produce an initial guess of the tile arrangement for each target surface. If needed, the tiles in the initial guess can be repositioned based on trial and error.

In all of the examples of Section 7, the cost function proposed in Eq. (5.14) was used to derive optimal tile arrangements. The target gap size $\boldsymbol{G}_{tar}$ is specified for each gap. Upon computing an optimal arrangement of tiles, the sub-folds must be added to one of the tile arrangements to complete the pattern.

To create a developable pattern, the sub-folds can be added to the flat arrangement of tiles in the plane of the tiles. In this case, care should be taken to ensure that the gaps of the flat tile arrangement are larger than corresponding gaps in the other arrangements, which can be achieved by setting a larger target gap size in the cost function for the flat tile arrangement.

To create a non-developable pattern, sub-folds can be added to any of the tile arrangements. In this case, the size of each sub-fold can be chosen freely. For example, the distance from point s_1 of Fig. 8 to the plane of the tile edges can be chosen.

After inserting the sub-folds, the pattern synthesis is complete and numerical simulations can be used to verify that the resulting pattern can be reconfigured into each of the target surface arrangements.

In conclusion, it is noted that, since the constrained optimization problem involves computations with nonlinear constraints, a limitation of the design process is that it is difficult to predict if feasible and high quality solutions will exist. For example, if the optimization algorithm fails to compute a feasible design, it is difficult to establish if no solution exists or if a more robust algorithm is needed. Additionally, if a feasible solution is found, there is no guarantee that it is locally or globally optimal. While it is difficult to make generic statements about the existence of

solutions, through the constraint counting analysis of Section 6, we build insights into when a feasible solution is expected for various tile connectivities.

6. Tile Shape and Connectivity

The shape and connectivity of the tiles determine the number of constraints in the optimization problem of Eq. (5.2), which have to be decided before implementing the pattern synthesis process. This section provides insight into the role of tile connectivity by analyzing the freedom F in the tile arrangement problem in relation to the number of the degrees of freedom of patterns of different size, with either equilateral triangle or square tile patterns.

The number of target surfaces, N_s, the number of tiles, N_t, and the number of gaps, N_g, determine whether the tile arrangement problem is overdetermined or underdetermined. From Table 1, for the ideal target constraint set:

$$F_{ideal} = 3N_sN_t - (3N_s - 1)N_g \tag{6.1}$$

and for the relaxed constraints with the developability constraint,

$$F_{relax,dev} = 5(N_s - 1)N_t + 3N_t - (3N_s - 1)N_g. \tag{6.2}$$

The ratio F/N_s has been plotted in Fig. 12 for various sizes of six-fold rotationally symmetric patterns with equilateral triangle tiles, beginning with a single loop of tiles. For example, the patterns with 7 and 13 loops are shown in Fig. 12a. In Figure 12b, F/N_s has been plotted for both the ideal constraints and the relaxed constraints including the developability constraint. The two limiting cases of two target shapes ($N_s = 2$) and an infinite number of target shapes have been considered. In both cases, the ideal constraint set leads to an overdetermined system ($F < 0$) for all of the patterns considered, with the exception of a single loop. However, for the relaxed constraints $F > 0$ and its value increases as the pattern size is increased. Note that the relaxed constraints without the strict developability constraint would provide even greater freedom. Even in the limiting case as the number of target surfaces N_s goes to infinity, there is freedom in the tile arrangement problem with the relaxed constraints. This suggests that solutions may be possible for a high number of target shapes, though the quality of the possible solutions would need a separate study.

Square tiles arranged in a square grid have also been considered. Example patterns with 4 loops and 9 loops are shown in Fig. 13a. Like the triangular tile cases considered previously, the ideal target constraints lead to an overdetermined constraint set except for the single loop, as shown in Fig. 13b. However, this time the relaxed constraints including the developability constraint lead to an underdetermined constraint set only for the smaller patterns. This is because for patterns with square tiles the number of gaps N_g grows more rapidly than the number of tiles N_t.

Along with the freedom in solving the tile arrangement problem, the connectivity of the tiles determines also the number of kinematic degrees of freedom of the fold pattern, which was computed with Eq. (2.2) assuming that there are no states of self stress. In fact, it was verified, using the bar-hinge kinematic model in Appendix A, that the patterns presented in this paper generically have no states of self stress. The number of kinematic degrees of freedom versus the number of loops have been plotted in Figs 12c and 13c for the triangle and square tile patterns. The figures show that, for both tile patterns, the number of degrees of freedom increases as the pattern increases in size.

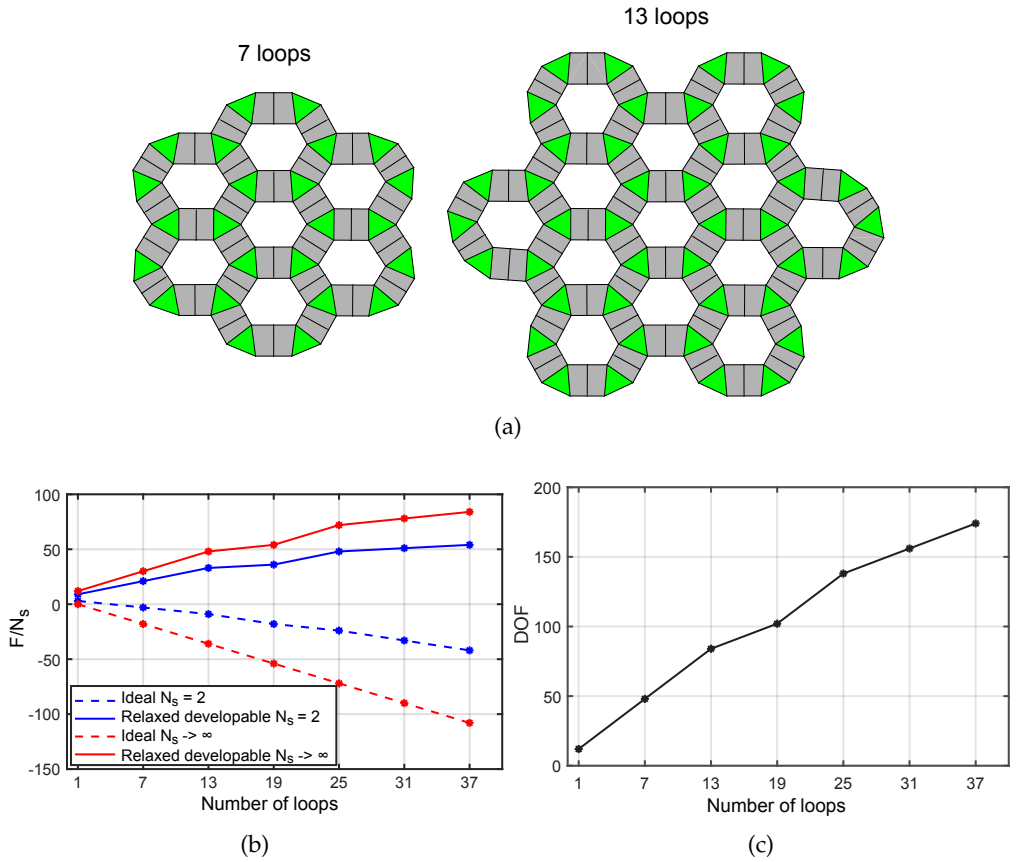


Figure 12: Analysis of connectivity of equilateral triangle tile patterns with six-fold rotational symmetry. (a) 7 loop and 13 loop patterns. (b) Ratio between freedom of tile arrangement problem and number of target shapes, plotted against number of loops. (c) Kinematic degrees of freedom plotted against number of loops.

7. Examples

The pattern synthesis process is demonstrated for a set of six target surfaces, shown in Fig. 14 and with the following equations:

$$
\begin{aligned}
T_1: &\quad \hat{z} = 0 \\
T_2: &\quad \hat{z} = \sqrt{20 - \hat{x}^2 - \hat{y}^2} \\
T_3: &\quad \hat{z} = \frac{1}{6}\left(\hat{x}^2 - \hat{y}^2\right) \\
T_4: &\quad \hat{z} = \frac{4}{3}\sin\left(\frac{3}{4}\hat{x}\right)\cos\left(\frac{3}{4}\hat{y}\right) \\
T_5: &\quad \hat{z} = \frac{1}{32}(\hat{x}^3 - 3\hat{x}\hat{y}^2) \\
T_6: &\quad \hat{z} = 5\sqrt{\frac{3}{2} - \frac{\hat{x}^2}{32} - \frac{5\hat{y}^2}{68}}
\end{aligned}
\tag{7.1}
$$

where $\hat{x} = x/L_0,\ \hat{y} = y/L_0, \hat{z} = z/L_0$.

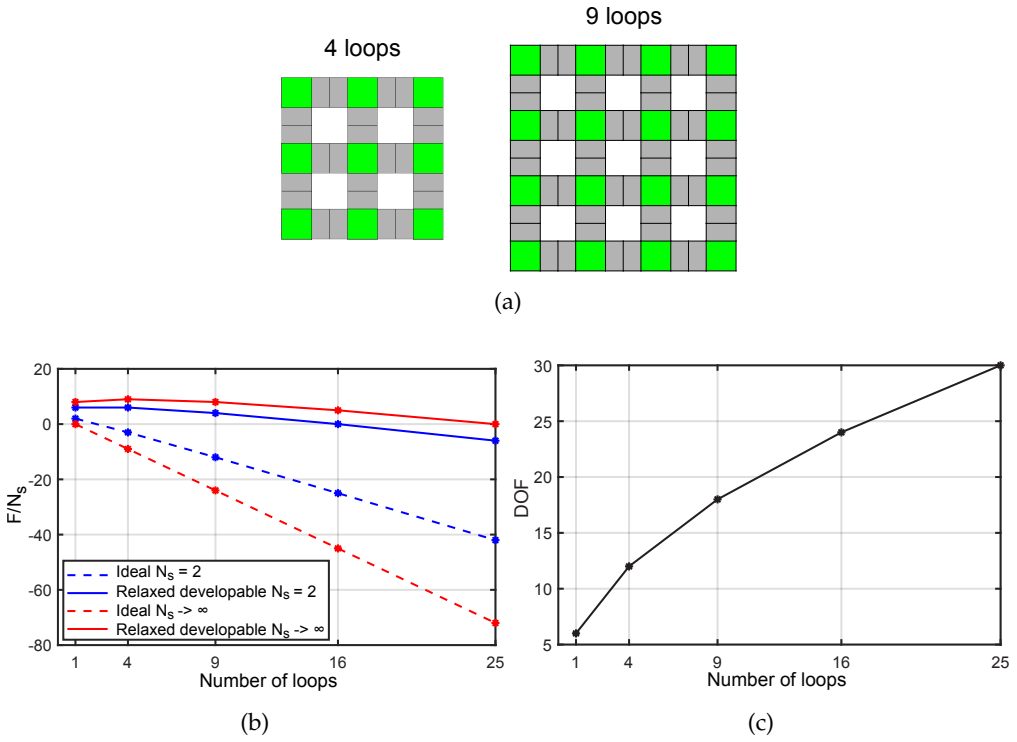


Figure 13: Analysis of connectivity of square tile patterns with four-fold rotational symmetry. (a) 4 loop and 9 loop patterns. (b) Freedom of tile arrangement problem plotted against number of loops. (c) Kinematic degrees of freedom plotted against number of loops.

Surface T_1 is flat, with zero Gaussian curvature, and is included in order to ensure that the tile patterns are developable. Surface T_2 is spherical, with positive Gaussian curvature. Surface T_3 is a hyperbolic paraboloid, with negative Gaussian curvature. T_4 is a sinusoidal surface, and hence includes both regions of positive and negative curvature. Surface T_5 is known as a monkey saddle, with negative curvature everywhere except at the origin, where it has zero curvature. Finally, T_6 is an ellipsoidal surface with positive curvature.

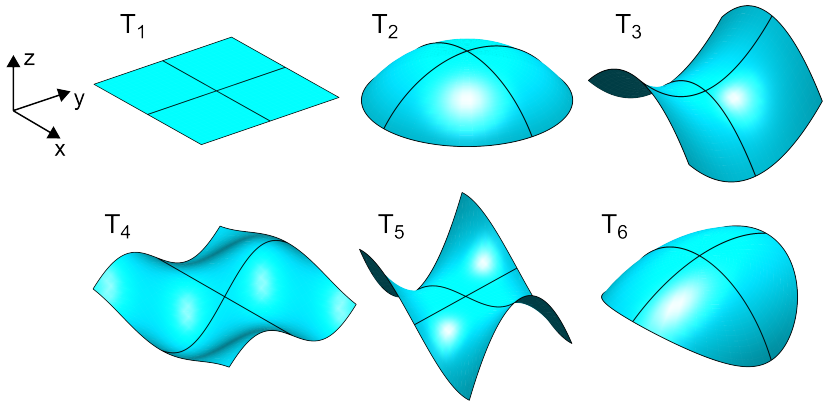


Figure 14: Target surfaces used in the pattern synthesis examples.

Three different kirigami patterns have been synthesized, that match some or all of these target surfaces. After the pattern synthesis process computes a fold pattern, it is verified through kinematic simulations that the pattern is rigidly foldable and continuous kinematic paths connect all target configurations. Appendix A formulates a kinematic bar-hinge model that captures the rigid folding kinematics. Also included in Appendix A is a kinematic pathfinding algorithm used to simulate the folding between target configurations. Animations of the folding simulations for each example are included as supplemental materials.

(a) Triangular tiles, three target surfaces

The first example considers the pattern with 13 loops of equilateral triangle tiles, as shown in Figure 12a, and targets the surfaces T_1, T_2 and T_3. Following the analysis of the freedom in the tile arrangement problem, Section 6, the relaxed developable target constraints are adopted. Subject to these constraints, the free variable and constraint count of Table 1 gives $F/N_s = 38$ free parameters.

The tile arrangement optimization problem was implemented according to Section 5. For T_1, the target gap size was set to $0.8L_0$ where L_0 is the side length of the tiles. A smaller target gap size of $0.4L_0$ was used for T_2 and T_3 to ensure that when the sub-folds are added to the flat tile arrangement, the T_2 and T_3 configurations can be achieved.

To explore the solution space, the weight w in the cost function of Eq. (5.14) was varied from $1/30$ to 30, and the resulting solutions are shown in Fig. 15. The tile arrangement corresponding to the target T_3 is shown in Fig. 15a. For $w = 1/30$, the tiles align closely to the surface, but the tile spacing is irregular. Conversely, $w = 30$ produces closer to uniform tile spacing at the cost of tiles poorly aligned to the target surface. A balance is achieved with $w = 1$. Figure 15b is a plot of the tile alignment and distance to the target gap size of the optimal tile arrangements as w varies. In practice, the selection of w will depend on the specific application. In the present case, $w = 1$ was selected. The corresponding mean and maximum tile misalignment and distance from target gap size are summarized in Table 2 for each target surface tile arrangement.

Table 2: Mean and maximum angles between tiles and surface normals for target surfaces in first example

Target surface	T_1	T_2	T_3
mean(θ) (degrees)	0	6.94	8.50
max(θ) (degrees)	0	10.22	10.91
mean($\lvert G - G_{tar}\rvert$)	$0.09L_0$	$0.07L_0$	$0.08L_0$
max($\lvert G - G_{tar}\rvert$)	$0.29L_0$	$0.30L_0$	$0.37L_0$

After computing the tile arrangements, fully-open (flat) sub-folds were added to the flat tile arrangement, resulting in the planar fold pattern in Figure 16a. This fold pattern can then be folded into the T_2 and T_3 configurations. Upon closing the sub-folds, a flat apparent surface is obtained, denoted as T_0. Note that the T_0 configuration does not correspond to a target surface, but it is guaranteed to exist by the gap constraints of Section 4. The four configurations of the pattern are shown in Fig. 16.

For a more detailed insight, in Fig. 16 one loop of triangles has been highlighted in blue. The pattern can be flattened so the intrinsic curvature of the loop is zero, as shown in the T_1 configuration. To achieve zero intrinsic curvature, trapezoidal gaps of both orientations are present in the loop, as discussed in Section 3, and the angles of the trapezoidal gaps around the loop sum to zero for developability. The apparent curvature of the highlighted loop varies between the target configurations. In the T_2 configuration, the pattern exhibits positive apparent curvature and the blue loop has an apparent angular defect of $\Delta_a = 6.0^\circ$. The T_3 configuration exhibits negative apparent curvature and the blue loop has an apparent angular defect of $\Delta_a = -7.1^\circ$.

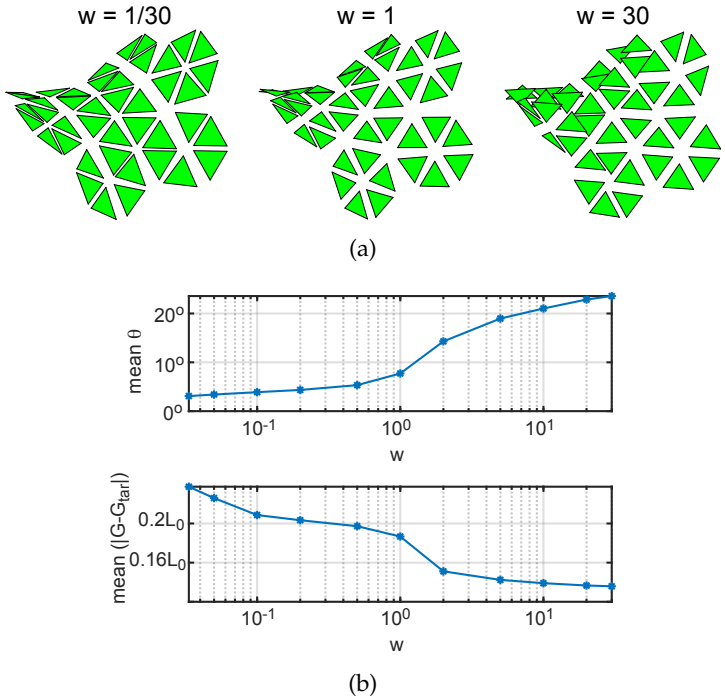


Figure 15: Analysis of optimal tile arrangements with varying cost function weight w for the 13 loop triangular tile example with 3 target surfaces. (a) Optimal tile arrangements corresponding to three values of w. (b) Characterization of the optimal tile arrangement for varying w, measured by the mean misalignment between tile and surface normals and mean distance to target gap size.

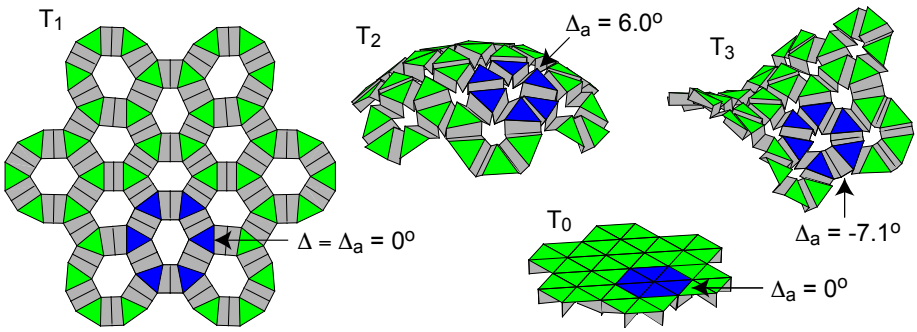


Figure 16: Example of developable kirigami pattern with equilateral triangle tiles designed to achieve target surfaces T_1, T_2 and T_3.

(b) Square tiles, two target surfaces

This second example demonstrates the pattern synthesis for square tiles. A pattern with two target surfaces, T_2 and T_3, was considered, omitting the flat target surface to create a non-developable pattern. A pattern connectivity with nine loops was considered. With $N_t = 16$, $N_g = 24$, and $N_s = 2$, the relaxed constraints (without developability) lead to a tile arrangement problem with $F/N_s = 20$ free parameters.

Target gap sizes of $0.8L_0$ and $1.2L_0$ were used for the spherical and saddle tile arrangement, respectively. The cost function weight was taken as $w = 0.25$, favoring normal vector alignment over gap spacing. In Table 3, the mean and maximum tile misalignment and deviation from the target gap size are listed for the optimal tile arrangements.

Once the tile arrangements had been computed, sub-folds were added to the T_2 arrangement. The size of the sub-folds was determined by specifying the distance from s_1 to plane $ABCD$ in Fig. 8. This distance was chosen as $0.5L_0$ for all sub-folds, resulting in the pattern shown in Fig. 17.

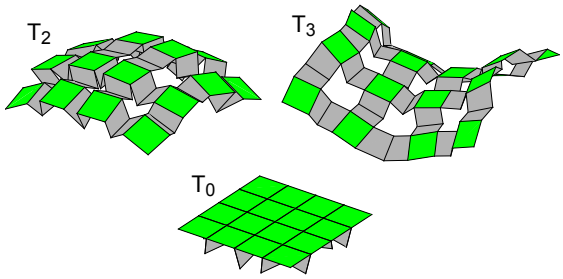


Figure 17: Example of non-developable kirigami pattern with square tiles, designed to achieve two target surfaces.

While the example shown in Fig. 17 exhibits only two target configurations, square patterns with more than two target configurations can also be designed. However, for the surfaces considered in the present study it was observed that the quality of the patterns tends to decrease sharply if more than two target surfaces are considered (e.g., solutions exhibiting large and irregular tile spacing, as well as large tile normal misalignment, were obtained).

Table 3: Mean and maximum angles between tiles and surface normals for the square tile pattern with two target surfaces in the second example

Target surface	T_2	T_3
mean(θ) (degrees)	7.92	10.43
max(θ) (degrees)	9.07	11.95
mean($\lvert G - G_{tar} \rvert$)	$0.23L_0$	$0.25L_0$
max($\lvert G - G_{tar} \rvert$)	$0.49L_0$	$0.76L_0$

(c) Triangular tiles, six target surfaces

In the final example, the capability of the pattern synthesis process to achieve all six target surfaces is demonstrated. Again, triangular tiles forming 13 loops were considered. For the target constraints, the relaxed developable set was used. In the cost function, the target gap size was set to $0.8L_0$ for the T_1 tile arrangement and $0.4L_0$ for all other tile arrangements. The cost function weight was set at $w = 1$. After computing the optimal tile arrangements, fully-open sub-folds were added to the T_1 tile arrangement to ensure developability. The resulting pattern is shown in Fig. 18 in the configurations corresponding to target surfaces T_1 through T_6 along with T_0 where the sub-folds fully closed.

Table 4 lists the mean and maximum offsets between tiles and target surface normal vectors, along with the mean and maximum distances from the target gap size. Compared to the case with only three target shapes, the normal vector offset and the distance to the target gap size of T_1, T_2, and T_3 are slightly larger. Additionally, it is observed in Table 4 that T_4 and T_5 exhibit

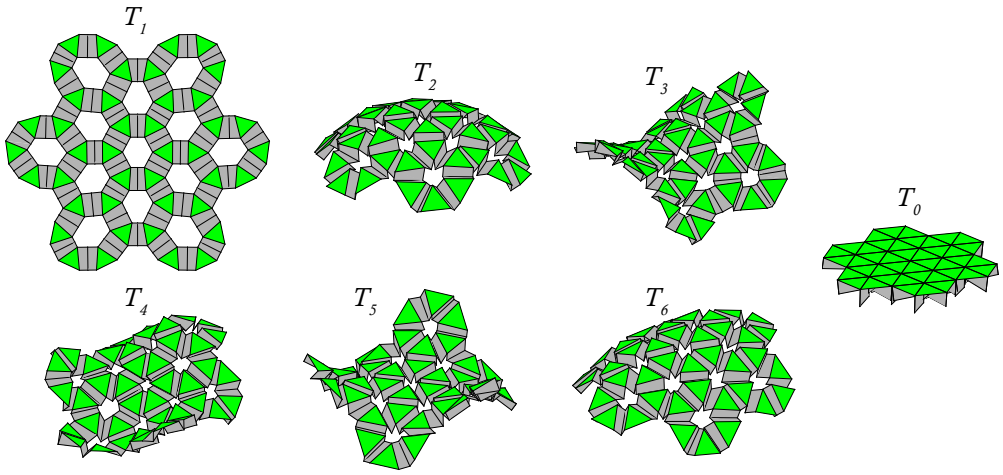


Figure 18: Pattern with equilateral triangle tiles designed to approximate six target surfaces, T_1, T_6.

larger errors than the other surfaces, indicating that complex curvature distributions are more difficult to approximate. Since both problems had been set up with identical target gap sizes and cost function weights, this result indicates that adding extra target shapes sacrifices the quality of the tile arrangements. However, this generally would depend on the specific target surfaces, tile connectivity considered, and cost function parameters.

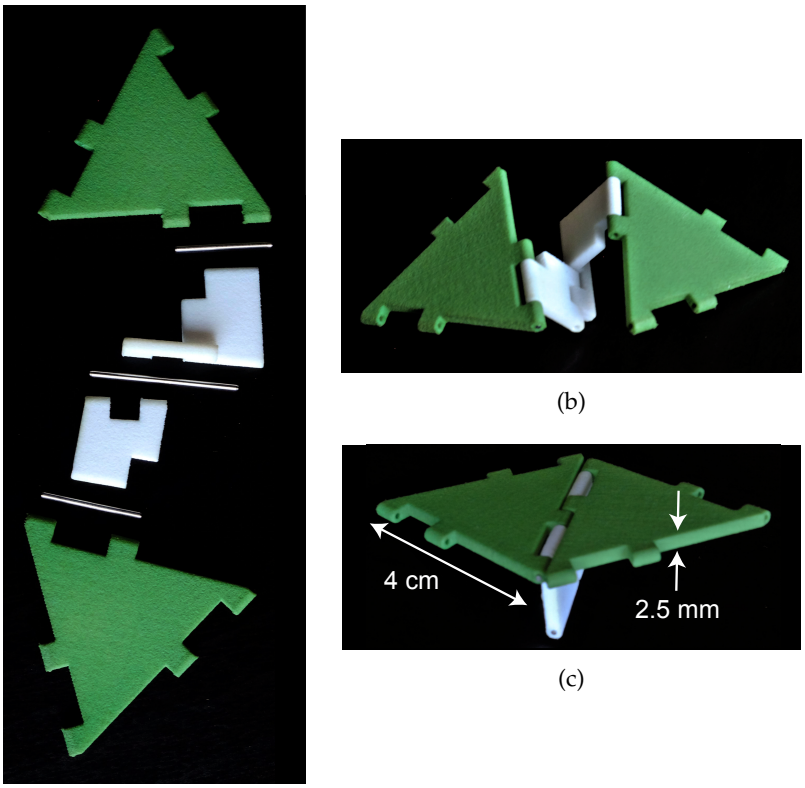
Table 4: Mean and maximum angle between tile and surface normal for the six target pattern

Target surface	T_1	T_2	T_3	T_4	T_5	T_6
mean(θ) (degrees)	0	7.59	8.55	11.15	9.38	6.25
max(θ) (degrees)	0	11.18	14.04	19.33	15.99	11.29
mean($\lvert \boldsymbol{G} - \boldsymbol{G}_{tar} \rvert$)	$0.12L_0$	$0.07L_0$	$0.08L_0$	$0.10L_0$	$0.11L_0$	$0.06L_0$
max($\lvert \boldsymbol{G} - \boldsymbol{G}_{tar} \rvert$)	$0.31L_0$	$0.27L_0$	$0.38L_0$	$0.36L_0$	$0.26L_0$	$0.20L_0$

8. Prototype

A physical prototype of the triangular tile pattern with six target surfaces was designed and built, with the aim of capturing the folding kinematics and demonstrate its reconfigurability. The model was constructed from 3D printed (Nylon 12) plates connected by steel pin hinges. This fabrication concept is appealing because it offers clean rigid-folding kinematics since the plates are stiff and the pin hinges have a well-defined axis of rotation. Although the entire model consists of many individual pieces, assembly simply requires sliding in the pin connections. It should also be noted that while this prototype is developable, the fabrication concept is not restricted to developable patterns.

Photos of the hinge design are shown in Fig. 19 for two tiles connected by a sub-fold. The pins have diameter of 1 mm and the plates are 2.5 mm thick, making the plates sufficiently stiff to exhibit nearly-rigid-folding kinematics. To accommodate the thickness of the plates, both faces of the sub-fold are split in half. A slender arm reaches from one face of the sub-fold to the other, Fig. 19(a), to ensure alignment of the hinge. With this design, the sub-fold is capable of fully closing and the zero-thickness kinematics assumed in the design are preserved despite using plates of finite thickness in the prototype.



(a) (b) (c)

Figure 19: Photos of hinge between two tiles connected by a sub-fold. (a) Exploded view of 3D printed plates and pin hinges; (b) sub-fold partially open; (c) sub-fold closed.

A prototype of the entire pattern was created by applying the sub-fold design concept of Fig. 19 to each sub-fold in the pattern. The prototype is shown in Fig. 20 in each of the six target surface configurations, as well as the T_0 configuration with the sub-folds fully closed.

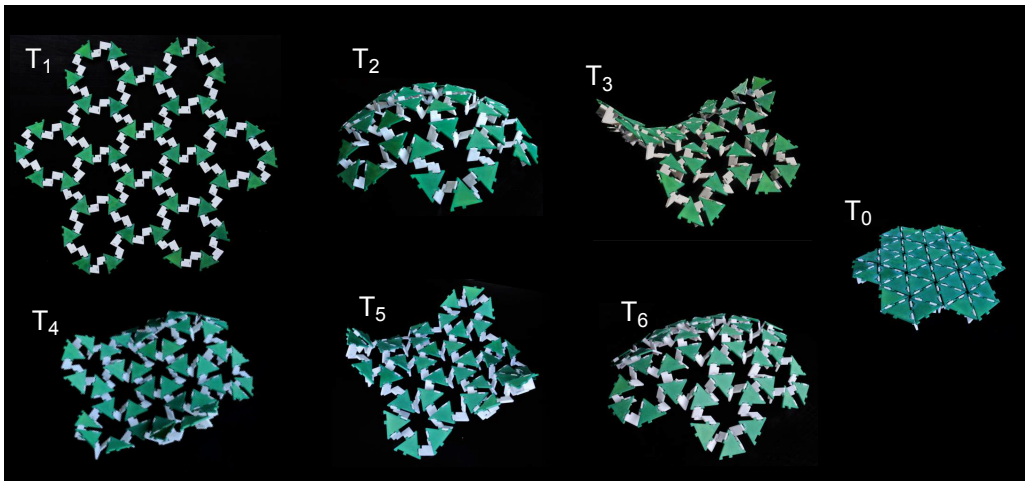


Figure 20: Photos of prototype in six target surface configurations and T_0 configuration with the sub-folds closed.

Since the pattern has 84 kinematic degrees of freedom, positioning the pattern into each target surface configuration is a challenge. Therefore, a 3D printed support structure for each configuration was used. The support structure consists of triangular posts that support each tile in

the target configuration, through slots on the underside of each tile. Figure 21 shows the support structure with and without the prototype for the T_3 (saddle) configuration. Once the prototype had been properly positioned on the support, a drop of glue was placed on each hinge to lock its angle, thus allowing the prototype to hold its configuration when it was removed from the support structure. The bond of the glue on each hinge was then be snapped, freeing the hinges so the next target configuration could be achieved.

(a) (b)

Figure 21: A 3D printed support structure was used to position the prototype in a target surface configuration. (a) Support structure for the T_3 (saddle) configuration. (b) Prototype positioned on the support structure.

9. Conclusions

This paper has presented the synthesis of a new family of pluripotent kirigami patterns. These patterns are capable of approximating multiple target surfaces of different Gaussian curvature. The patterns consist of a set tiles connected by sub-folds, and the apparent curvature of the surface comprised of the tiles only can be reconfigured. Changing the sub-fold angles serves as a mechanism for changing the apparent curvature of the pattern, leading to patterns capable of achieving a large range of apparent curvatures.

A numerical framework for pattern synthesis has been presented, in which arrangements of the tiles corresponding to each target surface are simultaneously computed by solving a constrained optimization problem. The constraints are formulated to ensure that sub-folds can be added between adjacent tiles and that each arrangement is compatible with the others. Additional constraints are included to ensure the arrangements closely approximate the target surfaces and both developable and non-developable patterns can be created. Upon computing a tile arrangements, the sub-folds are added to connect adjacent tiles in one arrangement, completing the pattern synthesis. Since all tile arrangements are compatible, the pattern can be reconfigured into each of the target surface tile arrangements.

The fold patterns that are synthesized in this way are relatively simple and are straightforward to manufacture, as demonstrated by the 3D printed prototype that has been presented. Although no guarantee is provided by the theory of Section 5 regarding the possibility that reconfiguration between all target configurations can be achieved continuously and without any physical interference, in all patterns presented in this paper this behavior was observed from the kinematic simulations and was also verified in the prototypes.

A major challenge that remains at the end of this study is the automatic reconfiguration of the presented structures. Due to their many degrees of freedom, the patterns have a rich space of configurations, which makes their reconfiguration more challenging. This topic will be addressed in a follow-on publication. At this stage, it is noted that the actuation of foldable structures with many degrees of freedom is an open and active area of research [27]. Several concepts have been explored, using smart materials [28], pressurization [29], and multi-stability [30]. The patterns

proposed in this paper further motivate further work investigating means for actuating patterns with many kinematic degrees of freedom.

Variations of the proposed design problem that could be explored in future work include relaxed tile constraints that allow for irregular tile shapes and thus provide more freedom in the design, and thickness accommodation, for example, through inequality constraints on the minimum allowable fold angles.

Data Accessibility. Matlab codes to generate the patterns presented in this paper are included in the supplemental materials.

Authors' Contributions. All authors jointly conceived the ideas and contributed to mathematical formulations. Numerical implementation and initial writing was performed by C.D. under guidance from S.P. All authors conceptualized the prototype, which was built by C.D. All authors contributed to the final manuscript.

Competing Interests. The authors have no competing interests.

Funding. This research was funded by the Air Force Office of Scientific Research under the MURI award number FA9550-18-1-0566 directed by Dr. Ken Goretta.

References

1. Callens SJ, Zadpoor AA. 2018 From flat sheets to curved geometries: Origami and kirigami approaches. *Materials Today* **21**, 241–264.
2. Resch RD. 1968 Self-supporting structural unit having a series of repetitious geometrical modules. US Patent 3,407,558.
3. Resch RD. 1973 The topological design of sculptural and architectural systems. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition* pp. 643–650.
4. Dudte LH, Vouga E, Tachi T, Mahadevan L. 2016 Programming curvature using origami tessellations. *Nature Materials* **15**, 583–588.
5. Dang X, Feng F, Plucinsky P, James RD, Duan H, Wang J. 2022 Inverse design of deployable origami structures that approximate a general surface. *International Journal of Solids and Structures* **234**, 111224.
6. Tachi T. 2013 Designing freeform origami tessellations by generalizing Resch's patterns. *Journal of Mechanical Design* **135**.
7. Lang RJ. 1996 A computational algorithm for origami design. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry* pp. 98–105.
8. Tachi T. 2009 Origamizing polyhedral surfaces. *IEEE transactions on Visualization and Computer Graphics* **16**, 298–311.
9. Hernandez EAP, Hartl DJ, Lagoudas DC. 2017 Design and simulation of origami structures with smooth folds. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **473**, 20160716.
10. Wang F, Guo X, Xu J, Zhang Y, Chen C. 2017 Patterning curved three-dimensional structures with programmable kirigami designs. *Journal of Applied Mechanics* **84**.
11. Benbernou N, Demaine ED, Demaine ML, Ovadya A. 2009 A universal crease pattern for folding orthogonal shapes. *arXiv preprint arXiv:0909.5388*.
12. Ovadya A. 2010 *Origami transformers: folding orthogonal structures from universal hinge patterns*. PhD thesis, Massachusetts Institute of Technology.
13. Sussman DM, Cho Y, Castle T, Gong X, Jung E, Yang S, Kamien RD. 2015 Algorithmic lattice kirigami: A route to pluripotent materials. *Proceedings of the National Academy of Sciences* **112**, 7449–7453.
14. Dieleman P, Vasmel N, Waitukaitis S, van Hecke M. 2020 Jigsaw puzzle design of pluripotent origami. *Nature Physics* **16**, 63–68.
15. Chen BGg, Santangelo CD. 2018 Branches of triangulated origami near the unfolded state. *Physical Review X* **8**, 011034.
16. Kuribayashi K, Tsuchiya K, You Z, Tomus D, Umemoto M, Ito T, Sasaki M. 2006 Self-deployable origami stent grafts as a biomedical application of Ni-rich TiNi shape memory alloy foil. *Materials Science and Engineering: A* **419**, 131–137.
17. Wilson L, Pellegrino S, Danner R. 2013 Origami sunshield concepts for space telescopes. In

54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference p. 1594.
18. Russo NE, Zekios CL, Georgakopoulos SV. 2019 Origami multimode ring antenna based on characteristic mode analysis. In *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting* pp. 2037–2038. IEEE.
19. Williams DE, Dorn C, Pellegrino S, Hajimiri A. 2020 Origami-inspired shape-changing phased array. In *2020 European Microwave Conference (EuMC)*. IEEE.
20. Zhang K, Jung YH, Mikael S, Seo JH, Kim M, Mi H, Zhou H, Xia Z, Zhou W, Gong S et al.. 2017 Origami silicon optoelectronics for hemispherical electronic eye systems. *Nature Communications* **8**, 1–8.
21. Huang Y, Wu H, Xiao L, Duan Y, Zhu H, Bian J, Ye D, Yin Z. 2019 Assembly and applications of 3D conformal electronics on curvilinear surfaces. *Materials Horizons* **6**, 642–683.
22. Tachi T. 2010 Geometric considerations for the design of rigid origami structures. In *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium* vol. 12 pp. 458–460. Elsevier Ltd.
23. Hilbert D, Cohn-Vossen S. 1999 *Geometry and the Imagination*. Number 87. American Mathematical Soc.
24. Bobenko AI, Sullivan JM, Schröder P, Ziegler G. 2008 *Discrete Differential Geometry* vol. 38. Springer.
25. Lang RJ. 2017 *Twists, tilings, and tessellations: mathematical methods for geometric origami*. CRC Press.
26. Crane K, De Goes F, Desbrun M, Schröder P. 2013 Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses* pp. 1–126.
27. Balkcom DJ, Mason MT. 2008 Robotic origami folding. *The International Journal of Robotics Research* **27**, 613–627.
28. Garza MR, Peraza-Hernandez EA, Hartl DJ. 2019 Self-folding origami surfaces of non-zero Gaussian curvature. In *Behavior and Mechanics of Multifunctional Materials XIII* vol. 10968 p. 109680R. International Society for Optics and Photonics.
29. Tachi T, Masubuchi M, Iwamoto M. 2012 Rigid origami structures with vacuumatics: geometric considerations. *Proc. the IASS-APCS Seoul, Korea, 21–24 May*.
30. Li Y, Pellegrino S. 2020 A theory for the design of multi-stable morphing structures. *Journal of the Mechanics and Physics of Solids* **136**, 103772.
31. Schenk M, Guest SD. 2011 Origami folding: A structural engineering approach. *Origami* **5**, 291–304.
32. Pellegrino S. 1993 Structural computations with the singular value decomposition of the equilibrium matrix. *International Journal of Solids and Structures* **30**, 3025–3035.
33. Tachi T. 2009 Simulation of rigid origami. *Origami* **4**, 175–187.
34. Hernandez EAP, Hartl DJ, Lagoudas DC. 2018 *Active origami: modeling, design, and applications*. Springer.
35. Li Y. 2020 Motion paths finding for multi-degree-of-freedom mechanisms. *International Journal of Mechanical Sciences* p. 105709.

Appendix A

Kinematic simulations are used to verify that the patterns are rigidly foldable and that continuous paths exist between configurations corresponding to each target surface.

A.1 Bar-hinge kinematics

The kinematics of rigid folding structures are captured by a pin-jointed rigid bar model [30] [31]. To develop a set of kinematic constraints, each edge in the fold pattern is treated as a rigid bar. Then, rigid bars are added to triangulate any non-triangular face. The bars added to triangulate the faces are treated as rigid hinges, enforcing planarity of the faces. The corresponding constraint

set is:

$$\boldsymbol{e}(\boldsymbol{x}) = \boldsymbol{l}(\boldsymbol{x}) - \boldsymbol{l}_0 = 0 \tag{A 1}$$

$$\boldsymbol{\theta}(\boldsymbol{x}) = 0 \tag{A 2}$$

where the fold pattern containing n_v vertices is parameterized by vertex coordinates $\boldsymbol{x} \in \mathbb{R}^{3n_v}$. The lengths $\boldsymbol{l} \in \mathbb{R}^{n_b}$ of the n_b bars are fixed at $\boldsymbol{l}_0 \in \mathbb{R}^{n_b}$. In other words, the bar extensions $\boldsymbol{e}$ should be zero. The rotations $\boldsymbol{\theta} \in \mathbb{R}^{n_h}$ about the n_h bars added to triangulate non-triangular faces are also set to zero.

The gradients of the kinematic constraints of Eqs. (A 1) and (A 2) are defined as:

$$(C^b)_{ik} = \frac{\partial l_i}{\partial x_k} \tag{A 3}$$

$$(C^h)_{jk} = \frac{\partial \theta_j}{\partial x_k}. \tag{A 4}$$

The compatibility matrices $\boldsymbol{C}^b$ and $\boldsymbol{C}^h$ for the rigid bars and rigid hinges, respectively, are explicitly computed in [30]. Upon linearization about a configuration $\boldsymbol{x}_0$, the kinematic constraints can be written as:

$$\begin{bmatrix} \boldsymbol{C}^b \\ \boldsymbol{C}^h \end{bmatrix} \boldsymbol{d} = 0 \tag{A 5}$$

where $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{x}_0$ is a displacement from configuration $\boldsymbol{x}_0$ and the compatibility matrices are evaluated at x_0.

The singular value decomposition of the constraint gradient matrix [32] is

$$\begin{bmatrix} \boldsymbol{C}^b \\ \boldsymbol{C}^h \end{bmatrix} = \boldsymbol{U}\boldsymbol{V}\boldsymbol{W}^T. \tag{A 6}$$

The right singular vector matrix $\boldsymbol{W}$ is composed of two sub-matrices:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{r_c} & \boldsymbol{W}_m \end{bmatrix} \tag{A 7}$$

where $\boldsymbol{W}_{r_c}$ has r_c columns and r_c is the rank of the compatibility matrix. The matrix $\boldsymbol{W}_m$ has m columns, which form an orthogonal basis of the m-dimensional null space of the compatibility matrix. The number of kinematic degrees of freedom is defined as m, which corresponds to the number of linearized mechanisms of the structure.

A.2 Pathfinding algorithm

The objective of the kinematic simulations is to verify the existence of a path between configurations of the fold pattern corresponding to each target surface. To simulate rigid folding between configurations, a kinematic pathfinding algorithm is outlined. Several pathfinding techniques have been developed to simulate rigid folding of origami [33], [34], [35]. Specifically, we follow [35] to perform pathfinding simulations.

In the problem at hand, the initial configuration of the pattern $\boldsymbol{x}^1$ is known. A path from $\boldsymbol{x}^1$ to a configuration corresponding to a target surface is desired. The tile vertices $\boldsymbol{v}_t \in \mathbb{R}^{3n_{vt}}$ of the target configuration are known; they are computed for each target surface in the tile arrangement problem presented in Section 5.

Following [35], a step toward the target triangle vertices is given by:

$$\boldsymbol{d}_j^i = \begin{cases} (\boldsymbol{v}_t)_j - \boldsymbol{x}_j^i & j \leq n_{vt} \\ 0 & j > n_{vt} \end{cases} \tag{A 8}$$

By projection of $\boldsymbol{d}^i$ onto the null space of the compatibility matrix, a first-order-compatible displacement is:

$$\boldsymbol{d}_m^i = \boldsymbol{W}_m|_i (\boldsymbol{W}_m)_i^T \boldsymbol{d}^i \tag{A 9}$$

where $\boldsymbol{W}_m|_i$ is $\boldsymbol{W}_m$ evaluated at configuration $\boldsymbol{x}^i$. Finally, an incremental step toward the target surface that satisfies the linearized constraints is:

$$\tilde{\boldsymbol{x}}^{i+1} = \boldsymbol{x}^i + \eta \hat{\boldsymbol{d}}_m^i \tag{A 10}$$

where

$$\hat{\boldsymbol{d}}_m^i = \frac{\boldsymbol{d}_m^i}{||\boldsymbol{d}_m^i||} \tag{A 11}$$

and η is the step size. Since the step was taken within the space of linear mechanisms, it will induce higher order errors. Correction steps can iteratively be applied until the constraints of Eqs. (A 1) and (A 2) are satisfied. The correction step is taken orthogonal to the null space of the constraints,

$$\boldsymbol{d}_c(\tilde{\boldsymbol{x}}^{i+1}) = -\boldsymbol{W}_{rc}|_i \, (\boldsymbol{V}_{rc})_i^{-1} \, (\boldsymbol{U}_{rc})_i^T \begin{bmatrix} \boldsymbol{e}(\tilde{\boldsymbol{x}}^{i+1}) \\ \theta(\tilde{\boldsymbol{x}}^{i+1}) \end{bmatrix}. \tag{A 12}$$

A pathfinding algorithm is summarized in Algorithm 1. The algorithm returns a set of configurations $\{\boldsymbol{x}^i\}$ along the path from the initial to the target configuration.

Algorithm 1 Kinematic pathfinding algorithm

while $||\boldsymbol{d}^i|| > \epsilon_1$ **do**
  Compute $\hat{\boldsymbol{d}}_m^i$ from Eqs. (A 9)-(A 11)
  $\tilde{\boldsymbol{x}}_{i+1} \leftarrow \boldsymbol{x}_i + \eta \hat{\boldsymbol{d}}_m^i$
  while $\left\| \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{\theta} \end{bmatrix} \right\| > \epsilon_2$ **do**
    $\tilde{\boldsymbol{x}}_{i+1} \leftarrow \tilde{\boldsymbol{x}}_{i+1} + \boldsymbol{d}_c(\tilde{\boldsymbol{x}}_{i+1})$
    $\boldsymbol{e} \leftarrow l(\tilde{\boldsymbol{x}}_{i+1}) - \boldsymbol{l}_0$
    $\theta \leftarrow \theta(\tilde{\boldsymbol{x}}_{i+1})$
  end while return $\boldsymbol{x}_{i+1} = \tilde{\boldsymbol{x}}_{i+1}$
  $i \leftarrow i + 1$
end while

The inner while loop increments the steps along the path, terminating when the configuration is at the target configuration. The inner while loop applies a correction after each step to correct for higher order constraint violations.

When applied to the patterns presented in Section 7, the algorithm had no issue finding paths between the target surface configurations. The step size η was manually selected for each path based on trial and error. Animations of the reconfiguration simulations between target surface configurations can be found in the supplemental materials.