

An additive framework for kirigami design

Received: 23 September 2022

Accepted: 10 April 2023

Published online: 25 May 2023

 Check for updates

Levi H. Dudte^{1,5}, Gary P. T. Choi^{2,5}, Kaitlyn P. Becker^{1,3} & L. Mahadevan^{1,4}✉

We present an additive approach for the inverse design of kirigami-based mechanical metamaterials by focusing on the empty (negative) spaces instead of the solid tiles. By considering each negative space as a four-bar linkage, we identify a simple recursive relationship between adjacent linkages, yielding an efficient method for creating kirigami patterns. This allows us to solve the kirigami design problem using elementary linear algebra, with compatibility, reconfigurability and rigid-deployability encoded into an iterative procedure involving simple matrix multiplications. The resulting linear design strategy circumvents the solution of a non-convex global optimization problem and allows us to control the degrees of freedom in the deployment angle field, linkage offsets and boundary conditions. We demonstrate this by creating a large variety of rigid-deployable, compact, reconfigurable kirigami patterns. We then realize our kirigami designs physically using two simple but effective fabrication strategies with very different materials. Altogether, our additive approaches present routes for efficient mechanical metamaterial design and fabrication based on ori/kirigami art forms.

Kirigami and origami are ancient art forms that have been rejuvenated as paradigms of a class of structural mechanical metamaterials derived from thin sheets. Kirigami works with cuts to create a globally coordinated, articulated, deployable structure that can lead to spectacular patterns in two and three dimensions, while origami works with creases to create a similar effect, although almost always in three dimensions. Both kirigami and origami can lead to complex structures that have been studied extensively in art, mathematics, science and engineering^{1,2}. However, the research literature on these subjects has had only a few points of convergence. Indeed, both artistic and technical books on origami^{3–5} scarcely, if ever, mention, kirigami, and vice versa⁶. From a mathematical perspective, although there are deep similarities between kirigami and origami, there are some subtle differences that require qualitatively different approaches to understand the implications of the local constraints on the global properties that result. In particular, both origami and kirigami exploit coordinated rotations of individual facets to achieve collective geometric changes of the connected structures; that is, they use piecewise deformations (with strong discontinuities along cuts/creases) to affect complex shape-shifting abilities. However, the nature of the discontinuity differs between origami and kirigami. In origami, creases are locations

identified with a discontinuous geometry (slope jump), whereas in kirigami, cuts are identified with a topological discontinuity (position jump).

Numerous studies have explored different kirigami-based deployable structures, with a focus that is primarily on the forward problem, that of predicting the geometric and mechanical response of given designs^{7–13}. The solution of the inverse problem typically involves setting up a global constrained optimization problem that encodes the geometric constraints in terms of the angles and lengths of the constituent tiles^{14–18}. Although such an approach is capable of producing patterns and mechanisms with different shape changes, the process of finding a solution to the nonlinear global constrained optimization problem for large patterns is both computationally difficult and hard to control. Inspired by biological morphogenesis driven by growing matter¹⁹, and its synthetic analog, additive manufacturing²⁰, one is naturally led to the question of whether it is possible to simplify the design of origami/kirigami metamaterials by switching from a global optimization perspective to some locally constrained additive process. In recent work, we showed that the use of local marching rules allows for an additive origami design approach²¹, converting a difficult non-convex global optimization problem²² into a simpler, but still nonlinear, local problem.

¹School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. ²Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA. ³Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA. ⁴Departments of Physics, and Organismic and Evolutionary Biology, Harvard University, Cambridge, MA, USA. ⁵These authors contributed equally: Levi H. Dudte, Gary P. T. Choi. ✉e-mail: lmahadev@g.harvard.edu

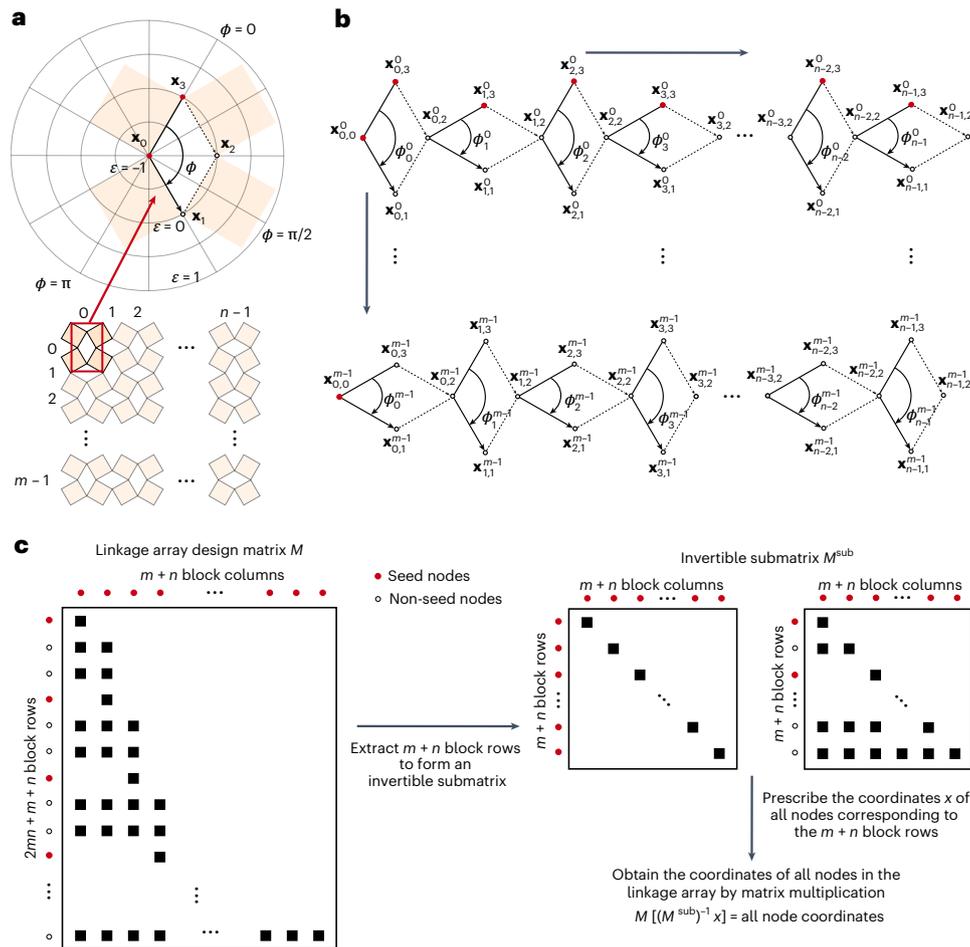


Fig. 1 | Marching construction for quad kirigami design. **a**, Geometric design of a four-bar linkage representing a negative space in an $(m + 1) \times (n + 1)$ quad kirigami pattern. Given the coordinates of two seed nodes \mathbf{x}_0 and \mathbf{x}_3 (red), the linkage can be parameterized by a deployment angle ϕ and an edge-length parameter (offset) ϵ . **b**, Each linkage strip consists of a series of four-bar linkages each parameterized by ϕ_j and ϵ_j . The coordinates of the non-seed nodes (open dots) are dependent on the seed nodes (red dots), as indicated by the arrows in the linkages. One can grow the linkages following the direction of the large arrows to obtain m linkage strips each with n linkages, thereby forming an $m \times n$ linkage array with nodes $\{\mathbf{x}_{i,j}^k, i \in \{0, 1, \dots, m - 1\}, j \in \{0, 1, \dots, n - 1\}, k \in \{0, 1, 2, 3\}\}$. **c**, Considering the relationship between all nodes in the linkage array, we obtain a design matrix M with all parameters $\{\phi_j^i\}$ and $\{\epsilon_j^i\}$ encoded in it (left).

Note that M consists of $2mn + m + n$ block rows and $m + n$ block columns, where each block (black square) is a 2×2 matrix corresponding to the xy coordinates of each node (also equation (3)). One can then extract $m + n$ block rows suitably to form a $(2m + 2n) \times (2m + 2n)$ invertible submatrix M^{sub} . In particular, choosing exactly the $m + n$ block rows corresponding to all seed nodes (red dots) will form an identity matrix $M^{\text{sub}} = I_{2m+2n}$ (middle). Alternately, one can choose block rows corresponding to some seed nodes (red dots) and some non-seed nodes (open dots) to form M^{sub} (right). In either case, we can then prescribe the coordinates of all nodes corresponding to the chosen $m + n$ block rows, and then invert the submatrix M^{sub} to solve for the seed node coordinates. Finally, by a direct matrix multiplication of M with the solved seed node coordinates, we obtain the entire linkage array.

In this Article we show that kirigami also lends itself to a simple additive approach that identifies and exploits a linear marching construction to connect local growth rules to a global form. The subtle differences in the nature of the geometric and topological constraints between origami and kirigami imply that there are qualitative differences in these marching algorithms that lead to a much simpler algorithm for kirigami, despite it being a far richer paradigm for structural material design. Furthermore, we show that we can control the flexibility of encoding contractibility, compact reconfigurability and rigid-deployability for the design of quad kirigami in any contracted or deployed space using a simple design matrix characterized by a combination of certain edge-length and angle parameters. This substantially simplifies the process of designing kirigami patterns as matrix multiplications. Our approach of decoupling the length field, angle field and the boundary node constraints allows us to systematically analyze the degrees of freedom in the design space of different types of quad kirigami tessellations.

Finally, to realize these designs in practice, we describe two manufacturing approaches using additive manufacturing and casting/molding for robust kirigami structures with soft, fabric hinges.

Results

Additive design framework

We tackle the kirigami design problem by working our way up a ladder of hierarchical complexity, beginning with the smallest building block, a single four-bar linkage representing the negative space of a unit cell of four quads, before moving on to strips of linkages and finally linkage arrays. At each level, our strategy will be to describe a geometric marching construction, the forward process, which can be cast as matrix multiplication, allowing us then to convert the marching algorithm into a flexible and intuitive design framework using linear inverse design techniques.

In Fig. 1 we present an overview of the marching construction for quad kirigami design. The key idea is to identify the constraints in the negative spaces in a deployed configuration, and use them to form a design matrix that is nonlinearly dependent on the deployment angles and edge lengths. However, once the matrix is determined, the coordinates of all nodes can be obtained by a simple matrix multiplication step, yielding a linear design method for quad kirigami.

Linkage design. To simplify our discussion, we start by considering the design of a negative space surrounded by four quads represented as a parallelogram four-bar linkage in \mathbb{R}^2 given by four points $\mathbf{x}_k = (x_k, y_k)$, where $k = 0, 1, 2, 3$, with the deployment angle $\phi = \angle(\mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_3)$ at \mathbf{x}_0 (Fig. 1a). We parameterize its construction according to

$$\begin{bmatrix} I-Q & Q \\ -Q & I+Q \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (1)$$

where $Q = (1 + \epsilon)R(-\phi)$ is a scaled rotation matrix, with ϵ being a scalar that we refer to as the offset, $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $R(-\phi) = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix}$ are the identity and rotation matrices in \mathbb{R}^2 , respectively, and

$\begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ x_j \\ y_j \end{bmatrix}$ is a column vector of coordinates of two points in the

parallelogram. Extending equation (1) yields an 8×4 linkage design matrix D that takes two points \mathbf{x}_0 and \mathbf{x}_3 (Fig. 1a, red nodes) as input and returns as output four points $\mathbf{x}_k = (x_k, y_k)$, $k \in \{0, 1, 2, 3\}$ to form the parallelogram:

$$D \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} I & 0 \\ I-Q & Q \\ -Q & I+Q \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} \quad (2)$$

Our choice of parameterization measures a kind of eccentricity of the parallelogram (its departure from a rhombus) by the offset ϵ and forms a counterclockwise polar coordinate system ($r = 1 + \epsilon, -\phi$) centered at \mathbf{x}_0 with $\phi = 0$ in the direction of $\mathbf{x}_3 - \mathbf{x}_0$. When $\epsilon = 0$, the parallelogram is a rhombus, and when $\epsilon = -1$, the parallelogram degenerates to two equal collinear line segments with $\mathbf{x}_0 = \mathbf{x}_1$ and $\mathbf{x}_2 = \mathbf{x}_3$. Otherwise, so long as $\phi \neq 0, \pi$, the generated points \mathbf{x} form a parallelogram in \mathbb{R}^2 for all values of ϵ . The points are ordered counterclockwise when $\phi < \pi$, clockwise when $\phi > \pi$ and are collinear when $\phi = \pi$. Holding ϵ constant keeps each edge length in the parallelogram constant, while varying ϕ casts the parallelogram as a four-bar linkage. This allows for a one-dimensional (1D) deployment path, whereby two of the opposite interior angles in the linkage are ϕ and the other two are $\pi - \phi$ (Supplementary Section 1.1 provides more details). There are two major advantages for using parallelogram negative spaces. First, as described above, parallelogram slits have a simple parameterization involving only one parameter ϕ and one offset parameter ϵ , substantially simplifying the final formulation for the kirigami design problem for large linkage arrays. Second, using parallelogram slits makes it easy to create rigid-deployable patterns that can morph from a closed and compact state to any deployed state, as we will discuss in Methods.

Linkage strips and linkage arrays. Now we will analyze the design of two parallelogram four-bar linkages having one common point and generalize this system to strips and arrays of connected linkages. Consider the first two adjacent linkages in the first row of Fig. 1b, composed of seven total nodes, so that one node is shared by both. The linkage points can be labeled $\mathbf{x}_{j,k}^0$, where $j \in \{0, 1\}$ is the linkage index, $k \in \{0, 1, 2, 3\}$ is the point in the linkage index, and $\mathbf{x}_{0,2}^0 = \mathbf{x}_{1,0}^0$ denotes the shared node. The linkages each have their own deployment angles ϕ_j^0 and offsets ϵ_j^0 . It can be observed that once $\mathbf{x}_{0,0}^0, \mathbf{x}_{0,3}^0, \phi_0^0$ and ϵ_0^0 in the first linkage are given, $\mathbf{x}_{0,1}^0$ and $\mathbf{x}_{0,2}^0$ are uniquely determined. In particular, as $\mathbf{x}_{0,2}^0$ is already determined, it suffices to prescribe $\mathbf{x}_{1,3}^0, \phi_1^0$ and ϵ_1^0 to uniquely determine all points in the second linkage.

We can apply this insight to a more complex linkage strip of n linkages $\mathbf{x}_{j,k}^0$, as shown in the first row of Fig. 1b, where $j \in \{0, \dots, n-1\}$ is

the linkage index, $k \in \{0, 1, 2, 3\}$ is a point in the linkage index, and $\mathbf{x}_{j,2}^0 = \mathbf{x}_{j+1,0}^0$ denotes a shared node. In other words, we can obtain a series of linkage design matrices $D_0^0, D_1^0, \dots, D_{n-1}^0$ that encode the recursive dependency of the nodes on the previously determined nodes via dynamic programming (Supplementary Section 1.2). Once the relationship in a linkage strip is determined, we can proceed to consider the relationship in a linkage array consisting of multiple linkage strips analogously, noticing that the (i, j) th linkage in the array is dependent on all linkages (\tilde{i}, \tilde{j}) with $\tilde{i} \leq i$ and $\tilde{j} \leq j$ recursively (Supplementary Section 1.3 provides more details). More specifically, the entire linkage array is dependent of the seed nodes $\{\mathbf{x}_{0,0}^0, \mathbf{x}_{0,3}^0, \mathbf{x}_{1,3}^0, \dots, \mathbf{x}_{n-1,3}^0, \mathbf{x}_{1,0}^0, \mathbf{x}_{2,0}^0, \dots, \mathbf{x}_{m-1,0}^0\}$ consisting of all the left boundary points and all the top boundary points in the linkage array (Fig. 1b, red nodes), the deployment angle field $\{\phi_j^i\}$ and the offset field $\{\epsilon_j^i\}$. We remark that the deployment angles $\{\phi_j^i\}$ have to satisfy certain additional conditions to yield a kirigami pattern with different physical properties such as contractibility and compact reconfigurability, which will be discussed in Methods.

Global linear inverse design via matrix operations. Putting all of these pieces together, for an array of $m \times n$ planar parallelogram four-bar linkages with nodes $\{\mathbf{x}_{j,k}^i\}$, $i \in \{0, 1, \dots, m-1\}$, $j \in \{0, 1, \dots, n-1\}$, $k \in \{0, 1, 2, 3\}$, we obtain a full linkage array design matrix M of size $(4mn + 2m + 2n) \times (2m + 2n)$. In particular, we have

$$M \begin{bmatrix} \mathbf{x}_{0,0}^0 \\ \mathbf{x}_{0,3}^0 \\ \mathbf{x}_{1,3}^0 \\ \vdots \\ \mathbf{x}_{n-1,3}^0 \\ \mathbf{x}_{0,0}^1 \\ \mathbf{x}_{0,0}^2 \\ \vdots \\ \mathbf{x}_{0,0}^{m-1} \end{bmatrix} = \begin{bmatrix} D_0^0 \\ D_1^0 \\ \vdots \\ D_{n-1}^0 \\ D_0^1 \\ D_1^1 \\ \vdots \\ D_{n-2}^{m-1} \\ D_{n-1}^{m-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{0,0}^0 \\ \mathbf{x}_{0,3}^0 \\ \mathbf{x}_{1,3}^0 \\ \vdots \\ \mathbf{x}_{n-1,3}^0 \\ \mathbf{x}_{n-1,3}^0 \\ \mathbf{x}_{0,0}^1 \\ \mathbf{x}_{0,0}^2 \\ \vdots \\ \mathbf{x}_{0,0}^{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{0,0}^0 \\ \mathbf{x}_{0,1}^0 \\ \mathbf{x}_{0,2}^0 \\ \mathbf{x}_{0,3}^0 \\ \mathbf{x}_{1,1}^0 \\ \mathbf{x}_{1,2}^0 \\ \mathbf{x}_{1,3}^0 \\ \vdots \\ \mathbf{x}_{n-1,1}^{m-1} \\ \mathbf{x}_{n-1,2}^{m-1} \\ \mathbf{x}_{n-1,3}^{m-1} \end{bmatrix} \quad (3)$$

with

$$M = \begin{bmatrix} D_0^0 \\ D_1^0 \\ \vdots \\ D_{n-1}^0 \\ D_0^1 \\ D_1^1 \\ \vdots \\ D_{n-2}^{m-1} \\ D_{n-1}^{m-1} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ G_{0,0}^{0,0} & G_{0,0}^{0,1} & 0 & 0 \\ G_{0,0}^{1,0} & G_{0,0}^{1,1} & 0 & 0 \\ \vdots & I & 0 & 0 \\ G_{0,1}^{0,0} G_{0,0}^{0,0} & G_{0,1}^{0,0} G_{0,0}^{0,1} & G_{0,1}^{0,1} & 0 \dots \\ G_{0,1}^{1,0} G_{0,0}^{1,0} & G_{0,1}^{1,0} G_{0,0}^{1,1} & G_{0,1}^{1,1} & 0 \\ \vdots & 0 & I & 0 \\ G_{0,2}^{0,0} G_{0,1}^{1,0} G_{0,0}^{1,0} & G_{0,2}^{0,0} G_{0,1}^{1,0} G_{0,0}^{1,1} & G_{0,2}^{0,0} G_{0,1}^{1,1} & G_{0,2}^{0,1} \\ G_{0,2}^{1,0} G_{0,1}^{1,0} G_{0,0}^{1,0} & G_{0,2}^{1,0} G_{0,1}^{1,0} G_{0,0}^{1,1} & G_{0,2}^{1,0} G_{0,1}^{1,1} & G_{0,2}^{1,1} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & I \end{bmatrix} \quad (4)$$

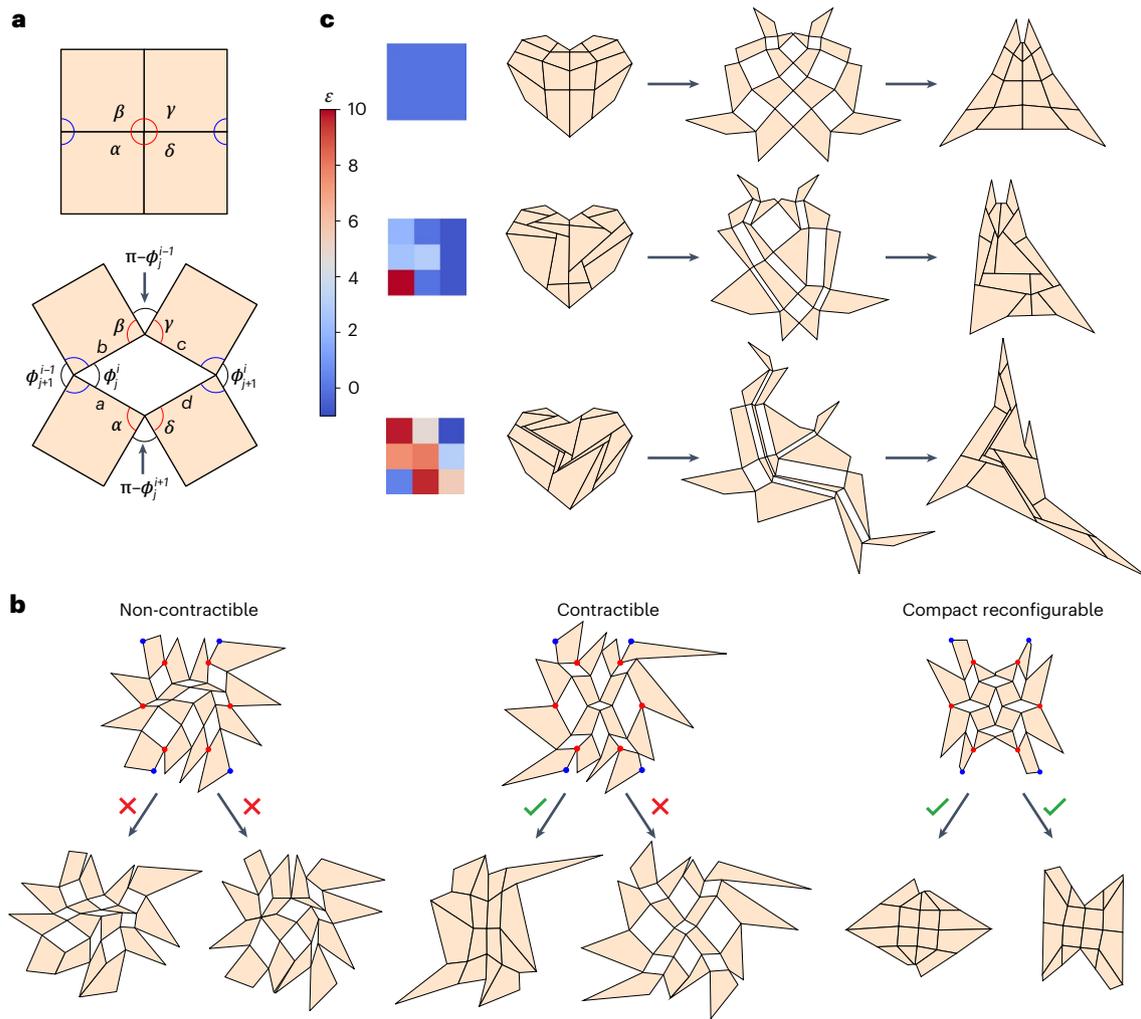


Fig. 2 | Encoding different desired properties in the design matrix. **a**, The angle constraints related to contractibility and compact reconfigurability can all be expressed in terms of the deployment angles of adjacent linkages and encoded in the design matrix. The top panel shows a contracted unit cell of four quads with the four angles $\alpha, \beta, \gamma, \delta$ at the center. The bottom panel shows a deployed configuration of the unit cell, where $\phi_j^i, \phi_{j-1}^i, \phi_{j+1}^i$ are the deployment angles of adjacent linkages. Note that the four angles $\alpha, \beta, \gamma, \delta$ highlighted in red determine the contractibility of the unit cell, while the four angles highlighted in blue determine the compact reconfigurability of it. **b**, Example choices of the deployment angle field $\{\phi_j^i\}$ that lead to a non-contractible pattern (left), a contractible pattern (middle) and a compact reconfigurable pattern (right). The top row shows three patterns obtained using the proposed design method with a given deployment angle field, where the same linkage boundary node

constraints (highlighted in red) and corner constraints (highlighted in blue) are used. In all examples, the offset field $\{\epsilon_j^i\}$ is set to be 0. The bottom row shows the two maximally contracted configurations of each pattern, where the red crosses indicate that the maximally contracted configurations are not closed and compact and the green check marks indicate that they are closed and compact. It can be observed that two closed and compact contracted configurations can be achieved only by using a deployment angle field satisfying the compact reconfigurability condition. **c**, Example choices of the offset field $\{\epsilon_j^i\}$ for the 3×3 negative space in a 4×4 kirigami pattern. Three compact reconfigurable, rigid-deployable heart shapes are obtained by solving the matrix equation, with the offset at each of 3×3 linkages shown in the corresponding square indicators (see the blue/red gradient for the values of each ϵ).

where D_j^i are the linkage design matrices, which can be expressed in terms of the following generator matrices using dynamic programming (Supplementary Section 1.3 provides more details):

$$\begin{bmatrix} G_{i,j}^{0,0} & G_{i,j}^{0,1} \\ G_{i,j}^{1,0} & G_{i,j}^{1,1} \end{bmatrix} = \begin{bmatrix} I - (1 + \epsilon_j^i)R(-\phi_j^i) & (1 + \epsilon_j^i)R(-\phi_j^i) \\ -(1 + \epsilon_j^i)R(-\phi_j^i) & I + (1 + \epsilon_j^i)R(-\phi_j^i) \end{bmatrix} \quad (5)$$

The sparsity of the matrix M forms a simple pattern (Fig. 1c): after the first block row, each of the next n set of three block rows has entries in a new block column unoccupied by all block rows above it in the matrix. For example, the fifth, sixth and seventh block rows in equation (3) contain $G_{0,1}^{0,1}, G_{0,1}^{1,1}$ and I , respectively, as entries in the third

block column, which contains only zero entries in the block rows above the fifth. After that, each of the next $(m - 1)$ set of $(n + 1)$ block rows has entries in a new block column unoccupied by all block rows above it in the matrix. This makes it easy to observe that a submatrix M^{sub} of M composed of any two block rows from the first four block rows, and one block row from each of the subsequent sets of block rows (that is, with a total of $2 + (n - 1) + (m - 1) = m + n$ block rows) will be a $(2m + 2n) \times (2m + 2n)$ square matrix with full rank and hence invertible.

This observation suggests a linear design strategy for quad kirigami patterns. To create a quad kirigami pattern consisting of $(m + 1) \times (n + 1)$ quads, we need to fully determine an array of $m \times n$ planar parallelogram four-bar linkages representing all

negative spaces. This linear inverse design process for the linkage array is summarized as follows:

1. Assemble the full linkage array design matrix M (Fig. 1c, first panel). Note that this requires choosing the deployment angle ϕ_j^i and offset ϵ_j^i for each linkage (i, j) , which will be discussed in detail in the next section.
2. Choose a subset of $(m + n)$ points $\{\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{m+n-1}\}$ in the linkage array such that the corresponding block rows of them in M form a $(2m + 2n) \times (2m + 2n)$ matrix M^{sub} with full rank (Fig. 1c, second panel). The coordinates of these points can be prescribed directly by the designer as boundary conditions. We can then use them to find the seed nodes in equation (3):

$$\begin{bmatrix} \mathbf{x}_{0,0}^0 \\ \mathbf{x}_{0,3}^0 \\ \mathbf{x}_{1,3}^0 \\ \vdots \\ \mathbf{x}_{n-1,3}^0 \\ \mathbf{x}_{0,0}^1 \\ \mathbf{x}_{0,0}^2 \\ \vdots \\ \mathbf{x}_{0,0}^{m-1} \end{bmatrix} = (M^{\text{sub}})^{-1} \begin{bmatrix} \tilde{x}_0 \\ \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_{m+n-2} \\ \tilde{x}_{m+n-1} \end{bmatrix} \quad (6)$$

In particular, if we simply choose $\{\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{m+n-1}\}$ to be the seed nodes (that is, all the top and left boundary points in the linkage array), then M^{sub} is simply the $(2m + 2n) \times (2m + 2n)$ identity matrix.

3. Calculate the full linkage array by a direct matrix multiplication using equation (6) together with equation (3) (Fig. 1c, third panel).
4. For the boundary nodes of the $(m + 1) \times (n + 1)$ kirigami pattern that are not included in the linkage array, one can determine their coordinates uniquely using the linkage array obtained in the last step together with four prescribed corner positions and a set of desired boundary offsets (Supplementary Section 1.4 provides details).

Choices of the deployment angles and the offsets. As discussed in the above subsection on linkage strip and linkage arrays, the full design matrix M is determined by the deployment angles and offsets of every four-bar linkage negative space. Here, the angles $\{\phi_j^i\}$ in the negative spaces are related by certain local angle rules. Depending on the desired properties of the resulting kirigami pattern, the degrees of freedom in them will be different. In particular, note that the angles of the tiles can be expressed in terms of the deployment angles (Fig. 2a). Therefore, we can achieve contractibility¹⁵, reconfigurability¹⁷ and rigid-deployability¹⁷ by enforcing additional constraints on the deployment angles $\{\phi_j^i\}$ (Fig. 2b). The offsets $\{\epsilon_j^i\}$ of all linkages can be chosen independently as long as they do not lead to degeneracies or self-intersections (Fig. 2c). More details are provided in Methods.

Linear and nonlinear inverse design

With the proposed additive design approach, we now have full control of the design of quad kirigami patterns. In particular, we have characterized the design space of contractible quad kirigami patterns as well as that of rigid-deployable, compact reconfigurable quad kirigami patterns. We can inverse design all such patterns by prescribing the desired interior properties via the deployment angle and offset fields in the design matrix, specifying certain boundary conditions, solving a matrix equation to find the corresponding seed

coordinates, and finally using direct matrix multiplication to obtain the resulting patterns. Moreover, as the deployment angle field and the boundary conditions are set in two separate steps, this linear design approach suggests that it is possible to control the target shape of a kirigami pattern (related to the boundary conditions) and the state of deployment (related to the deployment angle) separately. To illustrate this idea, in Fig. 3a we design two rigid-deployable, compact reconfigurable kirigami patterns that deploy to a circle at the deployed state with $\phi = \pi/2$ (that is, all negative spaces are rectangular) and a trapezium at the deployed state with $\phi = \pi/4$ (that is, all negative spaces are parallelograms with the acute angle being $\pi/4$), respectively, from which it can be observed that we have precise control of the shape matching and the deployment angle (Supplementary Section 3 presents more results).

A natural next question is whether it is possible to match multiple target shapes at multiple stages of deployment. Since the boundary conditions have already been used for enforcing the shape of the pattern at a certain stage, we cannot use them to control the shape at another state. Nevertheless, there are still DOFs in the angles and edge lengths in setting the design matrix. This suggests a nonlinear inverse kirigami design approach for approximating multiple prescribed shapes at multiple states: given a certain nonlinear objective function that quantifies the optimality of a kirigami pattern, we can search for an optimal pattern effectively by solving the optimization problem over the remaining set of the design parameters. For example, we can solve for a second contracted target shape by fixing the deployment angle as $\phi = \pi$ and optimizing the offset parameters $\{\epsilon_j^i\}$ (Supplementary Section 2 provides more details). Figure 3b shows various nonlinear inverse design results of rigid-deployable, compact reconfigurable kirigami patterns that morph from a closed and compact square to a second closed and compact target shape. For example, similar to the results of our recent work that uses a complementary global optimization approach¹⁷, our current method is capable of producing a square-to-circle shape change. However, our method can go further as we can design many more different target shapes with different curvature properties at the second contracted state (Supplementary Section 3), owing to the simplified parameter space in the proposed additive design formulation.

With the above linear and nonlinear inverse design results, one may ask about the limit of the shapes achievable by introducing cuts within a square. As the design space of the rigid-deployable, compact reconfigurable quad kirigami patterns is fully characterized by our framework, we can easily generate a large number of such patterns by setting the design matrix parameters randomly and perform a statistical analysis. Here we fix the linkage array size as $m \times n = 10 \times 10$ and generate 10,000 kirigami patterns on a unit square with random offsets $\{\epsilon_j^i\}$. More specifically, for each trial, the boundary nodes of the pattern are enforced to lie on a unit square with uniform spacing, and the deployment angle ϕ is set to be 0. For the offset ϵ_j^i of each linkage (i, j) , we first randomly sample $r_j^i \in [-1, 1]$ and then set $\epsilon_j^i = 10^{r_j^i} - 1$, thereby ensuring that $\epsilon_j^i \in [-0.9, 9]$ and that there is no degeneracy. We then use the proposed method with the above parameters to obtain a rigid-deployable, compact reconfigurable square kirigami pattern, and then assess the geometric properties of its fully deployed configuration ($\phi = \pi/2$) and its second contracted configuration ($\phi = \pi$) such as diagonal ratio and area as functions of ϕ (Supplementary Section 4 provides more details). Specifically, by considering the ratio r_d between the two diagonal lengths of a structure at different states, we can assess the shear of the overall shape as ϕ increases. As shown in Fig. 3c, the diagonal ratio at the fully deployed state ($r_d(\pi/2)$) and that at the second contracted state ($r_d(\pi)$) forms a linear relationship. It is also noteworthy that the reconfigured diagonal ratio is always larger than the deployed diagonal ratio, which suggests a natural limitation on the possible shape change achievable throughout the deployment.

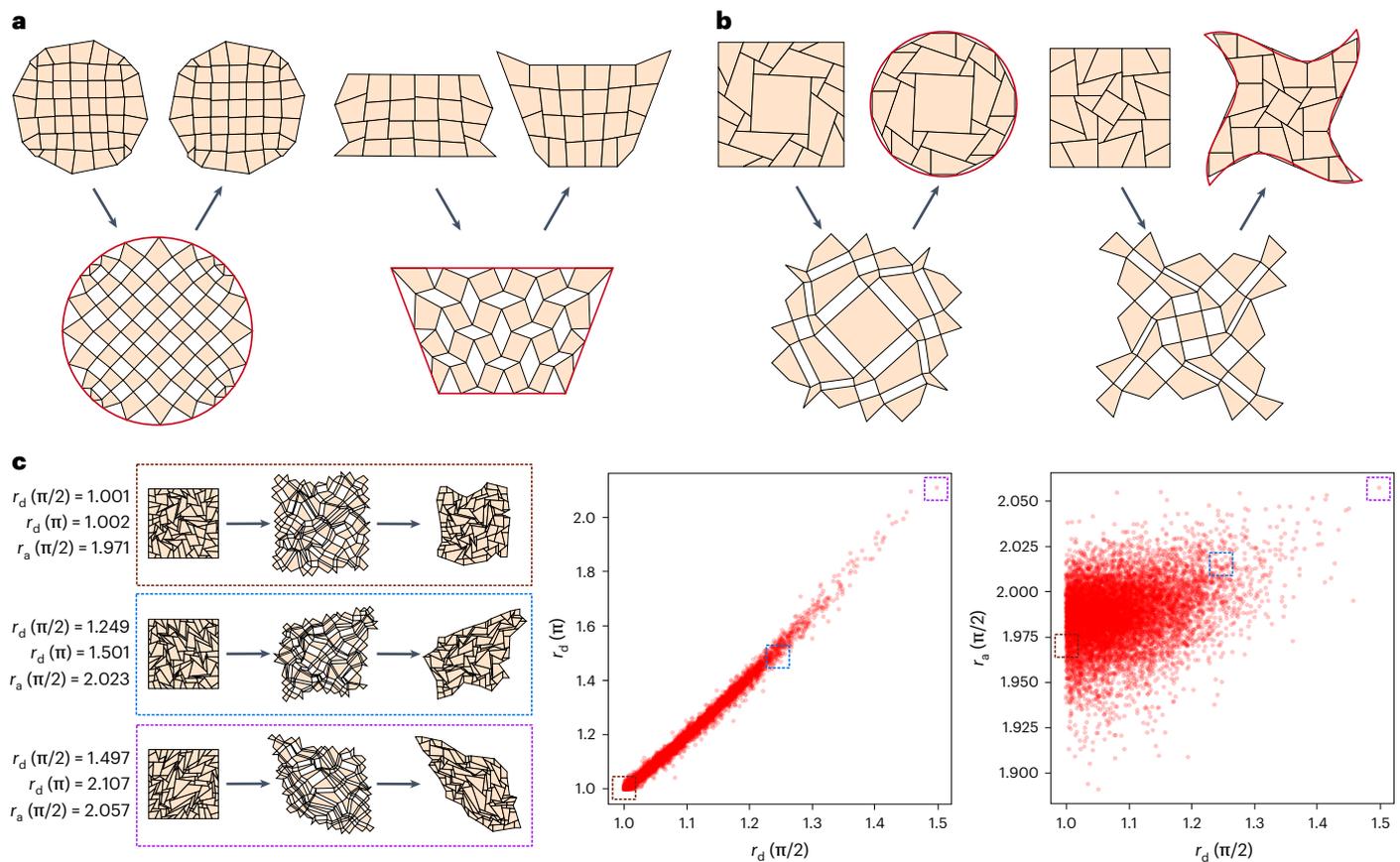


Fig. 3 | Linear and nonlinear inverse design of quad kirigami patterns using the proposed framework. a, Rigid-deployable, compact reconfigurable kirigami patterns that approximate a target circle and trapezium at two prescribed deployed states $\phi = \pi/2$ and $\phi = \pi/4$, respectively. **b**, Patterns that morph from a square to a circle (positive curvature) and a star shape (mixed curvature) in the second contracted state. **c**, Statistical analysis of 10,000 compact reconfigurable, rigid-deployable square kirigami patterns with random cuts. The left column

shows the deployment paths of three random patterns. The middle and right plots show the relationship between the diagonal ratio at the fully deployed state $r_d(\pi/2)$, the diagonal ratio at the second contracted state $r_d(\pi)$ and the maximum deployed area $r_a(\pi/2)$ for all 10,000 randomly generated patterns, where each red dot represents a pattern. The positions of the three example patterns on the two scatter plots are boxed.

As for the overall area of the pattern r_a , it can be shown that the maximum deployed area of the pattern is always achieved at $\phi = \pi/2$. Comparing the diagonal ratio and the maximum deployed area of the random patterns, we find that these quantities are positively correlated. One can also consider other quantities such as the side length ratio r_l and study the relationship between them (Supplementary Section 4). We remark that as the proposed design framework only involves simple matrix operations, the generation and analysis of each pattern are highly efficient and take less than 0.5 s on a laptop.

Physical model fabrication

Moving from the additive design of kirigami structures to physically realizing rigid-deployable, compact reconfigurable kirigami patterns requires a careful treatment of the hinges between the tiles to enable compact reconfigurability of the patterns in practice. Although it is possible to use tape joints to connect the tiles as illustrated in our earlier work¹⁷, the fabrication is time-consuming and the hinges are likely to fatigue and break. To circumvent both issues, we propose two fabrication techniques for manufacturing robust kirigami models: an additive manufacturing (3D printing) method (Fig. 4a–d) and a molded elastomeric composite method (Fig. 4e–h). More details are provided in Methods.

Discussion

Our additive perspective for kirigami design circumvents the difficulty of non-convex optimization in favor of a much simpler recursive method based on linear algebra. By parameterizing each negative space by two parameters ϕ and ϵ and formulating the kirigami design problem as a matrix multiplication based on the relationship between adjacent linkages, our method decouples the fields of deployment angles and lengths of the cut patterns and hence gives a better understanding of the design space of quad kirigami. From the structure of the design matrix, one can also precisely identify the degrees of freedom in the control of the overall pattern shape from the viewpoint of matrix rank. Altogether, our proposed formulation effectively characterizes the design space of quad kirigami patterns for a range of problems in science and engineering. Complementing this, our combination of 3D printing and casting approaches provides a relatively inexpensive and efficient way of manufacturing kirigami models with fatigue-resistant hinges.

A limitation of our work is that our approach currently only focuses on 2D quad kirigami design with parallelogram negative spaces. Possible future directions include extending the additive design formulation to more general kirigami patterns with non-parallelogram negative spaces¹⁵ or other polygons, as well as generalizations to non-planar linkages in 3D, which would require the consideration of additional geometric constraints.

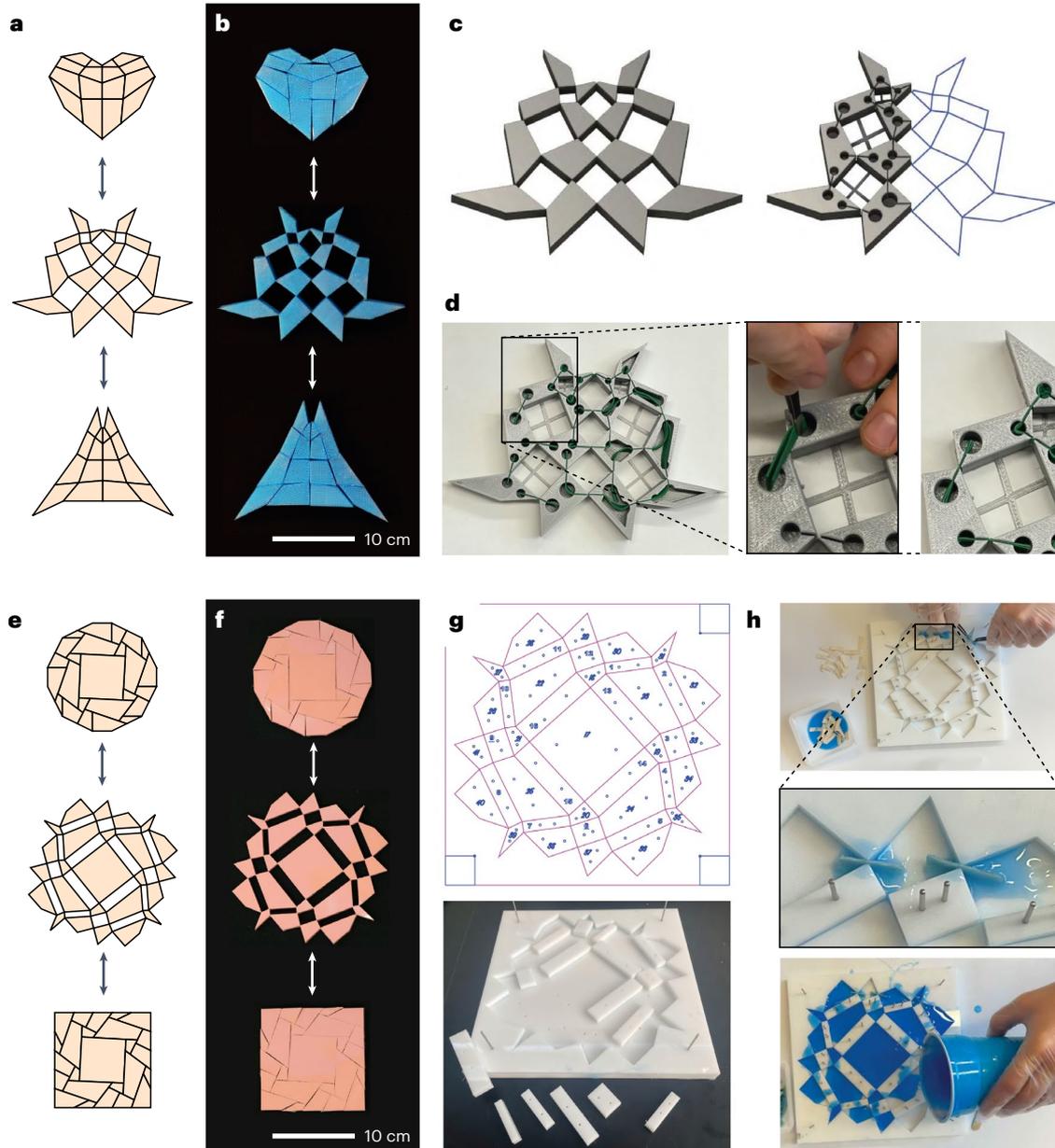


Fig. 4 | Physical model fabrication for rigid-deployable kirigami fabricated via a 3D printing-based method and a cast silicone composite method. **a,b**, Snapshots of the deployment of the heart pattern in Fig. 2c (a) and a corresponding 3D-printed model (b). **c**, A vector file of the pattern mid-deployment imported into 3D modeling software and extruded to a chosen thickness. Slots are digitally cut out for installing textile hinges, and temporary trusses added to fix deployment angles for hinge installation. Larger holes make it easier to insert hinges, and patterns can be processed in symmetric subunits for efficiency and later

mirrored or rotated. **d**, Once the rigid tiles are 3D-printed, fabric hinges are inserted in the slots using tweezers, as shown in the zoom-in image. The hinge insertion holes can later be covered for esthetic purposes. **e,f**, Snapshots of the deployment of the circle-square pattern in Fig. 2c (e) and a corresponding pink elastomeric model (f). **g**, A laser-cut mold based on a deployed configuration (bottom) is created with the vector file of the pattern shown in the top panel. **h**, Fabric pieces are inserted in the gaps between mold pieces using tweezers (top), and the mold is filled with blue silicone rubber to form the final soft kirigami sheet (bottom).

Methods

Choice of deployment angles and offsets

Contractibility. For a quad kirigami pattern to be contractible, each unit cell consisting of four adjacent tiles has to satisfy an edge-length constraint and an angle-sum constraint¹⁵. Specifically, suppose the four edges of a negative space have length a, b, c, d and the four angles of the tiles are $\alpha, \beta, \gamma, \delta$ as shown in Fig. 2a, then the following conditions must be satisfied:

$$a + d = b + c \tag{7}$$

$$\alpha + \beta + \gamma + \delta = 2\pi \tag{8}$$

In our formulation of the parallelogram four-bar linkages, we have $a = c$ and $b = d$ and hence the edge-length constraint in equation (7) is automatically satisfied (note that here we do not require the four angles to meet at a common point, as we have the flexibility to change the offset). Now, we rewrite the angle-sum constraint using the deployment angles of the (i, j) th four-bar linkage and its adjacent linkages. As shown in Fig. 2a, we have

$$\alpha + \delta + (\pi - \phi_j^i) + (\pi - \phi_j^{i+1}) = 2\pi \tag{9}$$

$$\beta + \gamma + (\pi - \phi_j^i) + (\pi - \phi_j^{i-1}) = 2\pi \tag{10}$$

Therefore, the angle-sum constraint in equation (8) can be rewritten as

$$\phi_j^{i-1} + 2\phi_j^i + \phi_j^{i+1} = 2\pi \tag{11}$$

As the deployment direction of adjacent linkages is alternating, for the entire $m \times n$ linkage array, we have

$$\begin{cases} \phi_j^{i-1} + 2\phi_j^i + \phi_j^{i+1} = 2\pi & \text{if } i+j \text{ is odd,} \\ \phi_{j-1}^i + 2\phi_j^i + \phi_{j+1}^i = 2\pi & \text{if } i+j \text{ is even} \end{cases} \tag{12}$$

where $i = 0, \dots, m - 1$ and $j = 0, \dots, n - 1$. Note that some of the above equations involve ghost points with out-of-range indices, namely $\phi_j^{-1}, \phi_j^m, \phi_{-1}^i, \phi_n^i$, which are free variables that can be prescribed arbitrarily. Once the values of them have been prescribed, equation (12) has full rank and gives a unique set of ϕ_j^i values for all i, j . In other words, the number of degrees of freedom in the space of deployment angles of a contractible $m \times n$ linkage array is exactly $2(m + n)$.

Figure 2b (left) shows an example deployed pattern obtained by the linear inverse design approach, where the offset field $\{e_j^i\}$ is set to be uniformly 0 in the design matrix and the deployment angle field $\{\phi_j^i\}$ is set arbitrarily. The coordinates of certain linkage array boundary nodes (red) are prescribed as the boundary conditions for solving the seed coordinates to obtain the linkage array, and then the four corners of the kirigami pattern (blue) are further prescribed to uniquely determine the remaining boundary nodes of the resulting kirigami pattern. It can be observed that the resulting pattern is non-contractible, with the two maximally contracted states containing holes. By contrast, Fig. 2b (middle) shows an example pattern obtained by the linear inverse design approach with the contractibility constraint in equation (12) enforced in the design matrix and all other parameters being the same as the previous example. This time, we can see that the resulting pattern admits a closed and compact contracted configuration.

Reconfigurability. For a kirigami pattern to be compact reconfigurable, that is, admitting two closed and compact contracted states, it should satisfy the above contractibility constraints as well as a set of dual edge-length constraints $a + b = c + d$ and dual angle-sum constraints¹⁷. In the formulation of the parallelogram four-bar linkages, the dual edge-length constraint is again automatically satisfied as we have $a = c$ and $b = d$ in Fig. 2a. Note that the dual angle-sum constraint at every four-bar linkage (i, j) is simply the dual case in equation (12). In other words, for a compact reconfigurable quad kirigami pattern, the corresponding linkage array should satisfy

$$\begin{cases} \phi_j^{i-1} + 2\phi_j^i + \phi_j^{i+1} = 2\pi, \\ \phi_{j-1}^i + 2\phi_j^i + \phi_{j+1}^i = 2\pi \end{cases} \tag{13}$$

for every pair of $(i, j) \in \{0, 1, \dots, m - 1\} \times \{0, 1, \dots, n - 1\}$. This implies that

$$\begin{cases} \phi_j^i = \phi & \text{if } i+j \text{ is even,} \\ \phi_j^i = \pi - \phi & \text{if } i+j \text{ is odd} \end{cases} \tag{14}$$

where ϕ is the deployment angle of the first linkage in the array. In other words, there is exactly one degree of freedom in the entire deployment angle field $\{\phi_j^i\}$. In particular, this degree of freedom captures the state of deployment of the pattern. Note that this agrees with the observation in our recent study¹⁷. Figure 2b (right) shows an example pattern obtained by solving the matrix equation with equation (13) enforced, where the boundary condition is given by the same set of boundary linkage vertices (red) and corner vertices (blue) as in the previous examples. It can be observed that the resulting pattern admits multiple closed and compact contracted configurations. We also note that, due

to the difference in the flexibility of the angle field, the resulting patterns in Fig. 2b are substantially different in shape.

Rigid-deployability. For the structure to be able to morph from a closed and compact contracted state to a deployed state with no tile strain, we need to satisfy certain conditions on the edge lengths and angles of the kirigami tiles¹⁷. Adapting this idea to the current additive design framework, we note that rigid-deployability depends on the geometry of the parallelogram four-bar linkages as well as the contractibility and compact reconfigurability discussed above. For any non-contractible pattern, for example the case in Fig. 2b (left), the pattern is always rigid-deployable between the two maximally contracted states, as the parallelogram four-bar linkages can deploy and contract without any geometrical frustration. For any contractible pattern, for example the case in Fig. 2b (middle), rigid-deployability near the fully contracted state depends on the geometry of the four-bar linkages. If all four-bar linkages are not rhombi, they will always form a straight line at the fully contracted state and hence the kirigami pattern is rigid-deployable between the fully contracted state and the second maximally contracted state. However, if some of the four-bar linkages are rhombi, they do not necessarily form a straight line at the fully contracted state and hence the pattern will generally not be rigid-deployable unless certain extra angle constraints are enforced, as described in ref. 17. In fact, it can be observed that the contractible kirigami pattern in Fig. 2b (middle) contains negative spaces that do not form a straight line in the fully contracted state, which indicates that the pattern is not fully rigid-deployable but only rigid-deployable within a certain range of angles (Supplementary Section 1.5 provides a more detailed discussion).

For any compact reconfigurable pattern created by our approach, such as in Fig. 2b (right), the parallelogram four-bar linkage assumption together with the angle constraint in equation (14) will automatically satisfy the conditions for rigid-deployability in ref. 17. Therefore, the compact reconfigurable kirigami patterns created by our approach are always rigid-deployable.

Offset field. We note that the offsets $\{e_j^i\}$ of all linkages can be chosen independently as long as they do not lead to degeneracies or self-intersections. As shown in Fig. 2c, different choices of the offsets can be used for creating a compact reconfigurable, rigid-deployable heart structure. Setting a uniform offset $e_j^i = 0$ at all linkages leads to a more regular and symmetric second contracted state (top row), while setting a large offset $e_j^i \gg 1$ for one particular linkage creates a more irregular shape locally (middle row). Setting a large offset at multiple linkages produces a much more irregular second contracted state (bottom row). This naturally raises the question of what conditions would yield admissible kirigami patterns without self-intersection, just as in origami²³. A detailed analysis (Supplementary Section 1.6) allows us to establish a theorem ensuring no self-intersections, and also provides empirical insights relevant for practical applications.

Implementation

The proposed framework is implemented in Python. Note that our framework involves solving a matrix equation to determine the deployment angle field $\{\phi_j^i\}$ (either using the contractible formulation or the compact reconfigurable formulation), and a matrix equation to find the seed positions in the linkage array from the prescribed node positions. These matrix equations are solved using the function `numpy.linalg.solve`.

For generating patterns that approximate a target shape, we compute the projection of the boundary points onto the target shape using the Python package `shapely` and evaluate the distance between them. The nonlinear optimization is done using Python's optimization and root finding toolbox `scipy.optimize`. We set all $e_j^i = 0$ as the

initial condition and search for the optimal $\epsilon_j^i \in (-1, \infty)$ for all linkages. Empirically, we do not observe self-intersections in the optimization results, even without imposing any additional constraints, possibly due to the fact that the occurrence of self-intersections is related to extreme values of ϵ_j^i , which are far from the initial condition. However, if one wants to ensure that there is no self-intersection, it is possible to further restrict all ϵ_j^i to a smaller range or introduce additional inequality constraints enforcing that all tile and linkage orientations are consistent (by considering the direction of the outward normal vector obtained using the cross product of two edges).

Physical model fabrication

We have introduced two fabrication approaches for producing rigid-deployable kirigami structures, namely the 3D printing approach and the rubber molding approach. Although fabrication is not limited to these two approaches, the purpose of introducing two methods is to give multiple viable options for construction, which come with the trade-offs from variations in the fabrication strategy, as well as differing materials compatible with the two approaches. The 3D printing approach presented below can be expanded to incorporate more complex geometries more easily than the casting approach. The printed method also can be leveraged for more modular assembly strategies and inline customization. The molding approach can be modified to achieve a higher throughput than would be possible with the printing approach. Using a soft resin in the casting method can also enable some designs that must pass through geometric frustration, as well as simplifying the mold design by not requiring draft angles and relief features. Casting could also be done with a more rigid resin, or a combination of rigid and soft materials.

Overview. To illustrate the 3D printing method, we fabricated the heart pattern obtained by the linear inverse design using this approach (Fig. 4a–d). The rigid tiles were created by 3D printing, with hollow slots where fatigue-resistant fabric hinges could be glued in place (Fig. 4d). Although the tiles can be printed individually and arranged for maximal packing density on a printer bed, we chose to print the tiles in a configuration halfway deployed between the two target design configurations so as to facilitate hinge insertion and balancing tension in the assembly. A vector file of the tile pattern was imported into a CAD software (Fusion360) and extruded to the desired thickness (20 mm for the heart example in Fig. 4b). Slots, 0.75 mm in width, were cut into the tiles of the CAD file to create space for the insertion of a fabric hinge, and allow for the tolerance of the printer (Flashforge Creator Pro), as well as the printing settings used to create the tiles. The additional circular openings visible in Fig. 4d were included to enable the use of tweezers to quickly place the fabric pieces. The slots and circular pocket cut in the CAD were extended to 2 mm above the bottom face of the tiles, leaving connecting material on the bottom for structural integrity and to hide the hinge attachment for esthetic purposes. Similarly, a 2-mm cap was printed and glued in place to cover the insertion holes and slots for the fabric hinges. As a final CAD step, temporary trusses were added in the negative spaces between tiles to guide tile alignment for hinge insertion and gluing. Each truss was -2 mm wide and 0.5 mm tall and extended between opposite tile faces inside the negative spaces. The trusses were aligned with the bottom face to be printed to avoid adding support material. The heart example was divided in half and mirrored to both fit the build platform of the fused filament fabrication (FFF) printer used and to reduce the amount of modeling work and modification required. Although this CAD preparation was done manually, it is possible to automatically generate parametric models, which are then used to create an STL and tool path that could be sent to a printer.

Once printed, 20-mm strips of fabric were cut and placed inside each of the slots with the help of tweezers. The fabric was then glued in place with a few drops of cyanoacrylate (superglue) placed at the

intersection of the circular pocket and hinge slots, with care taken to not flood the fabric exposed in the hinge with glue as this would rigidify the hinge. After the hinges were installed, the trusses were trimmed away with an exacto blade. The caps mentioned above were then glued over the side of the tiles with the exposed hinge slots and circular pockets for esthetic purposes. This approach yields deployed shapes that match the computational result very well; additionally, the resulting rigid-deployable, compact reconfigurable kirigami structures have high bending stiffness.

We can also create much softer kirigami sheets using a molding approach (Fig. 4e–h). We start by producing a laser-cut acrylic molds based on a deployed configuration of the kirigami patterns obtained by our framework. A vector file of the pattern can be minimally edited in vector editing software to add alignment pin holes, add an outer border to the mold, and prepare the file to be received by commercial laser cutters. Alternately, processing of the patterns to a laser-cutter-compatible file could also be automated with Python script. The mold could also be printed or machined, but laser-cut acrylic is quick to fabricate, laser cutters are relatively prevalent in industrial and academic fabrication spaces, and the acrylic is inexpensive and compatible with the molding process. As an added benefit, the laser-cut acrylic does not require mold release. The mold can be glued together, but we used pins with a light press fit (less than 10 pounds of force) to both align the mold pieces and hold them together. The press fit pins are inexpensive, provide sufficiently precise locating between parts, and can be assembled by hand or with a pair of pliers.

Once the mold is assembled, we use tweezers to insert textile pieces into the mold between mold pieces, as shown in Fig. 4h. These textile pieces will form the hinges. Before placing them in the mold, the textile pieces are coated in the two-part liquid silicone rubber that will fill the mold to form the soft kirigami sheet in which the hinges will be embedded. The physical model in Fig. 4f was made with Elastosil (Wacker Chemie, M4601AB) with cheesecloth hinges (McMaster-Carr, PN8808K11), and the physical model in process in Fig. 4h was made with DragonSkin 20 (SmoothOn) and generic herringbone cotton twill tape (0.25-inch ribbon) for the hinges. Dipping the textile hinges in rubber before installing them into the mold ensures saturation of the textile and full mechanical incorporation in the rubber, as well as helping to minimize bubble entrapment during the mold-filling process. We install our rubber-coated fabric hinges in place before the rubber cures, then fill the rest of the mold with a silicone rubber to produce a physical model. This approach facilitates the fabrication of large arrays and allows us to minimize the size and stiffness of the hinges. For illustration, we fabricated the square-to-circle pattern obtained by the nonlinear inverse design (Fig. 4e) and observed that the deployment of the physical models (Fig. 4f) matched that of the computational results very well.

Digital preparation of 3D-printed physical models. An overview of the process for preparing the kirigami design for the 3D printing is shown in Supplementary Fig. 20, and the post-processing, hinge insertion, and assembly of the printed kirigami tiles are shown in Supplementary Fig. 21. The kirigami is first saved in a vector format (such as a dxf file) to be imported into a 3D modeling software package. The result of importing a dxf file of the heart structure into a sketch within Fusion360 (Autodesk's 3D modeling software) is shown in Supplementary Fig. 20a, but this approach of preparing the kirigami pattern for printing can be accomplished in various other 3D modeling software packages or automated via a Python script. The workflow, which may be adapted to various software packages, begins with a vector sketch of the kirigami pattern, which is extruded perpendicularly to the sketch plane, as shown in Supplementary Fig. 20b, to create a 3D structure. Designs with symmetric subunits can be divided, as shown in Supplementary Fig. 20c, for later mirroring or rotation,

thus saving time and allowing larger assemblies to be made on small printer platforms.

For this rigid-deployable kirigami pattern, we add open slots in the extruded tiles into which textile hinges can be added after printing. These slots are indicated on an individual tile in Supplementary Fig. 20d. Circular clearance pockets are added to the ends of the slots in the tiles, and the pockets are large enough to allow the use of tweezers for hinge insertion. Alternately, larger portions of the tile interior can be removed, as shown in the second half of the deployable heart in Supplementary Fig. 21. Removal of additional material maximizes clearance and reduces overall material usage in the printing process. For structural integrity and esthetic purposes, however, the hinge slots and clearance pockets do not extend through the full thickness of the tiles. The thickness of the slot will depend on the hinge textile used, the printer tolerance and the overall scale of the assembly. For this particular instantiation of the design, the tiles were 15 mm thick, the slots were 1.5 mm wide (t_{slot}) and the height of the slots was 12 mm (h_{slot}), thus leaving 3 mm of thickness on the bottom of the tile. The center line of the slot intersects the corner of the individual tiles, but it is not necessary for the slots to form a straight line between tiles, nor is it necessary for the slot to bisect the insertion angle of the tile corners, as is visible in Supplementary Fig. 20d,e.

As a final step before printing, temporary trusses are added in the negative spaces to hold the tiles in a fixed, partially deployed configuration. As shown in Supplementary Fig. 20f, the trusses hold the alignment of the tiles for hinge insertion and are later removed. Although the trusses are helpful to the assembly process, they are not necessary and could be eliminated to allow for greater packing density on the printer bed. The cross-section of the trusses is $-0.5 \text{ mm} \times 2 \text{ mm}$ and they are oriented such that they are printed against the bed, with the first few layers of the model. The trusses should be thick enough that they are not broken when the tiles are removed from the printer platform or when the hinges are installed, but they are also thin enough that they can be easily trimmed with a flush cutter or a knife.

3D printing of physical models and hinge installation. The prototype shown in Supplementary Fig. 21 was printed on an FFF printer (Flashforge Creator pro, Zhejiang Flashforge 3D Technology Co.), using 1.75-mm polylactic acid (PLA) filament (Hatchbox), and can be easily translated to other printers and printed materials. By printing the tiles in a deployed configuration and not in either of the fully compact states, we minimize the processing after printing that needs to be done to separate the tiles. An example of a single textile hinge before installation is shown in Supplementary Fig. 21a. The length of the hinge does not need to be precise, but a hinge is easier to insert with the use of one or two tweezers, in which case the hinge must be long enough to span two of the clearance pockets. After the hinges are in place and positioned as desired, superglue can be used to tack the fabric to the inside of the tiles, adding a few drops to the slot from the inside of the clearance pockets, leveraging the low viscosity of the glue to wick into the slot. Care should be taken to not get glue on the span of hinge exposed between tiles, as this would rigidify the hinge textile and prevent the hinge from bending freely.

Finishing details for 3D-printed models. For esthetic purposes, we use a set of 3D-printed caps to cover the holes and form the closed tiles of the kirigami structure. The caps are the same profile as the tiles and can be glued or press-fit into place. In the example shown in Supplementary Fig. 21, the caps are 2 mm thick and have 2-mm-thick circular extrusions on the inner surface that line up with the clearance pockets. These extrusions are not necessary but assist with alignment. Depending on the printer and material used, as well as the press fit desired, the extrusions may vary from oversized to undersized for the fit with the clearance pockets. In the example shown, the physical model is printed in PLA with 30% infill, three outer solid layers, a line-to-line fit and a chamfer.

The purpose of the chamfer is to help with insertion. The result was a light press fit capable of holding most of the caps in place. The tolerance and print quality was such that a couple of the caps did not stay in place without light applied pressure, but permanent installation of the caps was accomplished by gluing them in place and thus the tolerances were sufficient. To fix the caps in place, a bead of hot glue was piped into the clearance pockets and the cap was pressed on top. The hot glue can also reinforce the interior attachment between the hinges and tiles or may replace the superglue if there is enough contact area of the textile, glue and tile interior. Given the low melting temperature of PLA, care must be taken with applying hot glue. This can also be improved by printing parts in a higher-temperature plastic such as nylon, acrylonitrile butadiene styrene (ABS) or stereolithography (SLA) resin.

Living hinges as a 3D-printed alternative. A construction technique using living hinges could be used in place of the textile hinge installation process described above. For a living hinge, the hinge material is the same as that of the tile in a thin enough geometry that it can bend without breaking. The textile was chosen for our physical models because it is less likely to fatigue or tear while reaching a near-zero pivot radius.

Mold preparation for molded rubber physical models. Details of the fabrication procedure of the rubber molding approach are given below, with illustrations of the mold preparation in Supplementary Fig. 22 and pictures of the molding and hinge installation in Supplementary Fig. 23. The mold is cut from a 6.35-mm (0.25 inch) sheet of acrylic. The mold consists of two layers that are held together with 0.0625-inch-diameter stainless-steel pins that are press-fit into holes that span both layers. The mold pieces could also be glued or bolted together. We find that the press-fit attachment is easy to deconstruct and the ability to quickly pull rigid mold tiles from the assembly without any tools required facilitates removal from the mold after the rubber has fully cured.

To create a mold, we start with a vector file of the kirigami design and turn it into a vector file like the ones shown in Supplementary Fig. 22a,b to laser-cut an acrylic sheet. As shown, we use a configuration approximately halfway through the transition between fully compact states to minimize small angles and small parts that could be difficult to assemble. Laser-cutting the transition state creates tiles of both the positive tile shapes and negative spaces of the four-bar linkage network that makes up the kirigami sheet. Assembling a mold of the tiles that represent negative spaces creates a positive model of the intended design when filled with a rubber or plastic. For this fabrication approach, the necessary gaps for installing hinges are created by the laser kerf, and the design and dimensions are not altered from the vector pattern exported from the script described above other than uniformly scaling the entire assembly. An example molding process is shown for the physical model of the rigid-deployable square-to-circle kirigami pattern in Fig. 3b, for which the assembly was scaled to fit within a square mold with 8-inch sides.

File preparation will vary for a given laser cutter, but a file set-up similar to the one shown in Supplementary Fig. 22a,b was used on both a Versalaser and an Epilog laser cutter to make suitable mold from 6-mm-thick acrylic sheets. For this example, Adobe Illustrator was used to create the vector files to run the laser cutter. There are two copies of the pattern, one for the top layer of the mold (shown in Supplementary Fig. 22a) and one for the bottom layer (shown in Supplementary Fig. 22b). A bounding box around the pattern was added to cut the exterior border of the mold, and circles were added throughout the pattern to create holes where press-fit pins would hold a tile defining a negative space to the bottom of the mold. The color coding in the file allows for different power settings between cuts and control over the order in which different paths are cut. In the vector files shown, pink and red are used for through-cuts with two different speed settings, and the blue lines are etched into the acrylic as references to guide the mold assembly but are not cut through.

The only critical power setting to tune and test before laser-cutting the mold parts is for the circular holes that are used for the press-fit pins that hold the various parts of the mold together. The press fit of the pins was designed to be tight enough so that they would not fall out if the mold is inverted but also loose enough that they could be readily pulled out without the aid of pliers. In this size range (1/16-inch- or 1.59-mm-diameter pins) this can be done by specifying the hole to be ~ 50 – $200\ \mu\text{m}$ smaller in the vector file than the physical pin diameter and adjusting the power and speed settings of the laser cutter to adjust the kerf. (To be clear, the kerf or cut width of the laser has a non-zero thickness, and directing the laser around a circular path of a given diameter will result in a hole of a larger diameter than the guiding circle.) Slower cuts and higher power cuts will remove more material, creating a larger kerf, larger hole and looser fit. We found that a test piece to calibrate the pin press-fit with several holes at increments of 25 or 50 μm was sufficient to calibrate the fit for a given laser cutter. The cut speed can also be used to fine-tune the kerf and fit. The recommended order of cutting operations is for the press-fit pin holes to be cut first. Otherwise, if the laser cutter bed is not flat and the tiles are cut first, the tiles may shift or drop lower once cut free from the piece of acrylic. When a tile shifts, this alters the laser focus for when the press-fit holes are then cut.

The kerf of a laser cutter is tapered and wider at the top. We found that this effect was not detrimental for 6.25-mm acrylic, but other mold fabrication methods could include milling or 3D printing if the taper proved problematic or thicker tiles would be desirable. Laser cutting was used for this work because it is a fast and inexpensive method with which to make the molds for kirigami patterns with embedded textile hinges. Additional layers could be added to a laser-cut mold for thicker tiles or added complexity. Additionally, although we demonstrate the creation of soft kirigami sheets, a similar process could be used to mold a rigid plastic with textile hinges. The mold would need to be modified to facilitate the release of rigid molded tiles. This can be accomplished with added draft angles, which may be accomplished with the laser kerf, or the mold could be made of a soft material.

As mentioned above, the blue lines visible in Supplementary Fig. 22a,b are etched and not cut through the acrylic sheet. These lines may be omitted but provide visual guidelines for assembling the mold. A partially assembled mold is shown in Supplementary Fig. 22c, where it is difficult to see the etched lines on the white acrylic. To make the lines more easily visible, ink can be wiped into the etched lines, as shown in Supplementary Fig. 22d. The ink used in this case was a silicone dye (Silc Pig, SmoothOn), and the excess was wiped away with a paper towel. The pin holes provide mechanical alignment and the etched numbers and lines help to visually guide manual assembly of the mold, as shown in Supplementary Fig. 22e. The modular press-fit mold is useful for small scale manufacturing but, ultimately, this could be converted to a two-part injection mold for higher-throughput fabrication.

Hinge installation and molding of soft kirigami sheets. Once fully assembled, the acrylic mold is ready for rubber casting and does not require mold release, because silicone rubber will not bond to the acrylic sheet or stainless-steel pins. The molding process is shown in Supplementary Fig. 23, starting with preparation of the textile (Supplementary Fig. 23a) that is used to create the hinges of the kirigami pattern. The pattern can be molded out of rubber alone (without textile hinges) but would then rip very easily. The textile reinforces the hinge while still allowing a full range of motion and transformation of the mechanism. The fabric used in the rigid-deployable square-to-circle kirigami model in Fig. 4 was a cotton cheesecloth, whereas the fabric used for the model in Supplementary Fig. 23 is a herringbone cotton twill tape. The cheesecloth is cut into ~ 12 -mm squares and folded in half (creating rectangles) before insertion into the mold. The twill tape is ~ 6 mm wide and thus only needs to be cut to length to fit into the mold

of 6.35-mm acrylic. The cotton fibers in either choice of textile have a high surface area and create a strong mechanical bond with the silicone. Synthetic fibers will work but were found delaminate more readily from the cast silicone. As shown in Supplementary Fig. 23b, the twill tape is prepared by cutting it into pieces ~ 6 – 12 -mm in length, long enough to span into neighboring cast tiles.

A two-part silicone rubber is mixed and then placed under vacuum to remove bubbles. While the rubber is degassing a small portion is set aside, as shown in Supplementary Fig. 23b, to start coating the cut pieces of twill tape. By coating the textile hinges (twill tape or cheesecloth) before inserting them, we found we could improve the saturation of the rubber in the textile, reduce bubbles, and improve the mechanical integration between the textile and rubber. The rubber used in the model shown in Fig. 4 was a two-part silicone rubber, Elastosil m4601 A/B (Wacker). The rubber used in Supplementary Fig. 23 is DragonSkin 20 (SmoothOn), with blue dye added (Silc Pig, SmoothOn) to help monitor the mixing and improve visibility in pictures for documentation. Silicone is an appealing material to use because of its durability, elastic strain behavior, and ease of use, but a large variety of rubbers would work to create these structures. A soft silicone rubber is also relatively easy to pull out of the rigid mold and does not require mold release. A rigid plastic or urethane could also be cast-molded with relatively minor mold modifications.

Once coated with rubber, the textile pieces are inserted into the mold to form the hinges (Supplementary Fig. 23c) and covered with a liquid rubber to form the kirigami structure (Supplementary Fig. 23d,e). In the example shown (Supplementary Fig. 23c), the textile is pinched between the corners of the acrylic mold tiles that define the open spaces of the kirigami structure to create a cotton and silicone composite hinge. The laser kerf is sufficient to create a gap between the mold tiles for the textile. Two pairs of tweezers, one on either side of the hinge, are very helpful for the process of inserting the textile, as shown in Supplementary Fig. 23c. The hinges can be inserted before or after the rubber is poured into the mold, as shown in Supplementary Fig. 23c–e. Once the rubber is fully cured, the soft kirigami sheet can be peeled out of the mold and trimmed of excess rubber, as shown in Supplementary Fig. 23f. If the excess rubber is not trimmed, it can over-constrain the kirigami sheet, preventing the hinges from pivoting through the intended motion. The excess rubber can also be prevented with careful pouring, by wiping the excess before curing, or by creating a fully enclosed mold with a precise or compliant interface. The reconfiguration of the hinges can be seen in the translucent soft kirigami sheet as it is transformed from a square (Supplementary Fig. 23g) to a mid-deployment state (Supplementary Fig. 23h) and to a circle (Supplementary Fig. 23i).

Data availability

Source data for Fig. 3c are available with this manuscript. Kirigami pattern files are available on Zenodo²⁴.

Code availability

The kirigami design codes are available on Zenodo²⁴.

References

1. Callens, S. J. & Zadpoor, A. A. From flat sheets to curved geometries: origami and kirigami approaches. *Mater. Today* **21**, 241–264 (2018).
2. Zhai, Z., Wu, L. & Jiang, H. Mechanical metamaterials based on origami and kirigami. *Appl. Phys. Rev.* **8**, 041319 (2021).
3. Lang, R. J. *Origami Design Secrets: Mathematical Methods for an Ancient Art* (CRC Press, 2012).
4. Demaine, E. D. & O'Rourke, J. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra* (Cambridge Univ. Press, 2007).
5. Hull, T. C. *Origami: Mathematical Methods in Paper Folding* (Cambridge Univ. Press, 2020).

6. Temko, F. *Kirigami, The Creative Art of Paper Cutting* (Platt & Munk, 1962).
7. Borcea, C. & Streinu, I. Geometric auxetics. *Proc. R. Soc. A* **471**, 20150033 (2015).
8. Chen, B. G.-g et al. Topological mechanics of origami and kirigami. *Phys. Rev. Lett.* **116**, 135501 (2016).
9. Bertoldi, K., Vitelli, V., Christensen, J. & Van Hecke, M. Flexible mechanical metamaterials. *Nat. Rev. Mater.* **2**, 17066 (2017).
10. Celli, P. et al. Shape-morphing architected sheets with non-periodic cut patterns. *Soft Matter* **14**, 9744–9749 (2018).
11. Singh, N. & van Hecke, M. Design of pseudo-mechanisms and multistable units for mechanical metamaterials. *Phys. Rev. Lett.* **126**, 248002 (2021).
12. Zheng, Y., Niloy, I., Celli, P., Tobasco, I. & Plucinsky, P. Continuum field theory for the deformations of planar kirigami. *Phys. Rev. Lett.* **128**, 208003 (2022).
13. Czajkowski, M., Coulais, C., van Hecke, M. & Rocklin, D. Conformal elasticity of mechanism-based metamaterials. *Nat. Commun.* **13**, 211 (2022).
14. Konaković-Luković, M., Panetta, J., Crane, K. & Pauly, M. Rapid deployment of curved surfaces via programmable auxetics. *ACM Trans. Graph.* **37**, 106 (2018).
15. Choi, G. P. T., Dudte, L. H. & Mahadevan, L. Programming shape using kirigami tessellations. *Nat. Mater.* **18**, 999–1004 (2019).
16. Jiang, C., Rist, F., Pottmann, H. & Wallner, J. Freeform quad-based kirigami. *ACM Trans. Graph.* **39**, 1–11 (2020).
17. Choi, G. P. T., Dudte, L. H. & Mahadevan, L. Compact reconfigurable kirigami. *Phys. Rev. Res.* **3**, 043030 (2021).
18. Dang, X., Feng, F., Duan, H. & Wang, J. Theorem for the design of deployable kirigami tessellations with different topologies. *Phys. Rev. E* **104**, 055006 (2021).
19. Thompson, D. W. *On Growth and Form* (Cambridge Univ. Press, 1942).
20. Truby, R. L. & Lewis, J. A. Printing soft matter in three dimensions. *Nature* **540**, 371–378 (2016).
21. Dudte, L. H., Choi, G. P. T. & Mahadevan, L. An additive algorithm for origami design. *Proc. Natl Acad. Sci. USA* **118**, e2019241118 (2021).
22. Dudte, L. H., Vouga, E., Tachi, T. & Mahadevan, L. Programming curvature using origami tessellations. *Nat. Mater.* **15**, 583–588 (2016).
23. Lang, R. J. & Howell, L. Rigidly foldable quadrilateral meshes from angle arrays. *J. Mech. Robot.* **10**, 021004 (2018).
24. Dudte, L. H., Choi, G. P. T., Becker, K. P. & Mahadevan, L. 'Additive kirigami'. *Zenodo* <https://doi.org/10.5281/zenodo.7763613> (2023).

Acknowledgements

This work was supported in part by the National Science Foundation under grants nos. DMS-2002103 (G.P.T.C.), DMR 20-11754 (L.M.), DMR 19-22321 (L.M.) and EFRI 18-30901 (L.M.), and the Simons Foundation (L.M.) and the Henri Seydoux Fund (L.M.).

Author contributions

L.H.D. and L.M. conceived the research. L.H.D., G.P.T.C., K.P.B. and L.M. designed the research. L.H.D. and G.P.T.C. derived the theoretical results, implemented the algorithms and conducted the numerical experiments. K.P.B. built the physical models. L.H.D., G.P.T.C., K.P.B. and L.M. analyzed the results and wrote the manuscript. L.M. supervised the research.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s43588-023-00448-9>.

Correspondence and requests for materials should be addressed to L. Mahadevan.

Peer review information *Nature Computational Science* thanks Xianqiao Wang, Alberto Corigliano and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Kaitlin McCardle, in collaboration with the *Nature Computational Science* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2023