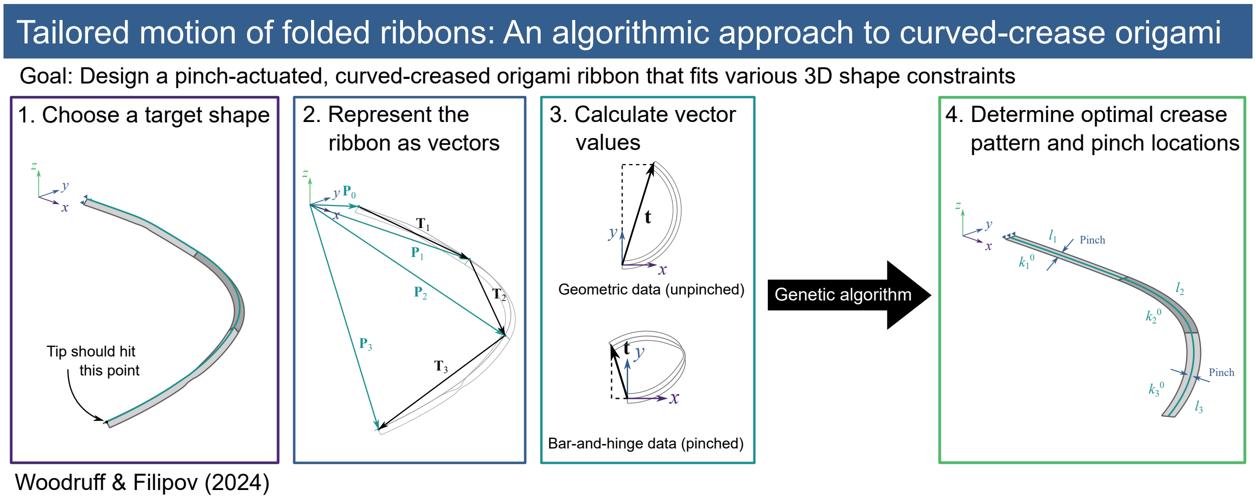
Graphical Abstract

Tailored motion of folded ribbons: An algorithmic approach to singly curved-crease origami

Steven R. Woodruff, Evgueni T. Filipov



Tailored motion of folded ribbons: An algorithmic approach to curved-crease origami

Goal: Design a pinch-actuated, curved-creased origami ribbon that fits various 3D shape constraints

1. Choose a target shape

2. Represent the ribbon as vectors

3. Calculate vector values

4. Determine optimal crease pattern and pinch locations

Genetic algorithm

Woodruff & Filipov (2024)

z
y
x
Tip should hit this point
P0
T1
P1
T2
P2
P3
T3
t
Geometric data (unpinched)
Bar-and-hinge data (pinched)
l1
Pinch
k1 0
l2
k2 0
k3 0
l3

Highlights

Tailored motion of folded ribbons: An algorithmic approach to singly curved-crease origami

Steven R. Woodruff, Evgueni T. Filipov

- Developed a genetic algorithm for optimal curved-crease origami design and actuation.
- Introduced vector representation of creases, improving shape approximation and deflection.
- Applied the algorithm to inverse-design tasks, achieving precise 3D targeting and shape morphing.
- Demonstrated the engineering potential of curved-crease origami in complex tasks.

Tailored motion of folded ribbons: An algorithmic approach to singly curved-crease origami

Steven R. Woodruff[a,], Evgueni T. Filipov[b]

[a] *Department of Engineering, James Madison University, 801 Carrier Drive, Harrisonburg, 22807, VA, USA*
[b] *Department of Civil and Environmental Engineering, The University of Michigan, 2350 Hayward Street, Ann Arbor, 48109, MI, USA*

Abstract

This study introduces a computational framework for the inverse design of curved-crease origami, addressing challenges in shape fitting and functional morphing of flexible, curved-crease ribbon structures. A novel vector-based representation is formulated to describe crease geometries composed of both pinched and unpinched curved segments, enabling efficient and accurate reconstruction of three-dimensional folded shapes. The proposed method integrates analytical geometry with bar-and-hinge simulation data, forming a hybrid forward model that captures the coupled bending and twisting behavior of curved folds. An inverse-design algorithm based on a genetic optimization scheme is developed to solve non-convex, non-smooth design problems involving (A) planar curve approximation, (B) maximization of tip deflection, and (C) three-dimensional target fitting. The computational framework automates the determination of optimal crease parameters and actuation schemes, achieving high precision with minimal manual intervention.

Email address: woodrusr@jmu.edu (Steven R. Woodruff)

Compared to prior approaches that rely on periodic origami tessellations, this method demonstrates the versatility of a single curved crease for achieving controlled shape change. The results highlight the framework's potential for broad computational mechanics applications, including morphing materials, soft robotics, and adaptive structural systems.

Keywords: Curved-crease origami, shape fitting, inverse design, morphing structures

1. Introduction

Shape fitting in origami involves determining a crease pattern that, when folded, approximates a specified three-dimensional form. In origami engineering, this process ranges from simple geometric mapping, such as aligning a folded sheet to specific points, to complex surface reconstruction tasks. Shape-fitting algorithms enable efficient transitions from a flat, easily fabricated sheet to a functional folded structure with prescribed geometry and mechanical performance.

Lang [1] pioneered this domain with the TreeMaker algorithm, which produced patterns to fold intricate origami designs, such as a deer or different insects. Subsequent advancements from Tachi [2] led to crease patterns that fit complex polyhedral forms. More recent innovations have explored topology-optimized origami [3, 4] and the challenge of fitting origami to surfaces with non-zero Gaussian curvature using repeated Miura-Ori [5] and Eggbox cells [6]. Furthermore, there has also been exploration in the alignment of rigid-foldable origami with space curves and metamaterials, as well as computational approaches for the inverse design and automated assignment of folds [7, 8, 9]. Related advances in kirigami and ori-kirigami metamaterials have introduced analytical and computational frameworks for lattice-based and multistable systems, enabling tailored mechanical responses through combined folding and cutting strategies [10, 11].

Despite progress with straight-crease patterns, research on curved creases remains scarce. Few examples exist, such as using curved-crease origami for conical and cylindrical structures [12, 13], approximating 2D curves (such as alphabetical letters) [14], tight wrapping of thin sheets using curved creases [15], or for approximating surfaces like vases [16, 17]. Additionally, Liu and James [18] have explored using the Lagrangian method to design curved-crease tessellations starting from a flat state. However, these primarily target artistic or mathematical endeavors, rather than engineering applications.

The potential of curved-crease origami remains vast. This field of origami promises the ability to fit intricate shapes and, due to the curved surfaces introduced by folding, could allow for more exact surface approximation. Furthermore, recent work has shown that the curvature and torsion of a curved-crease origami structure can be programmed using local actuations (i.e., pinches), which shows promise for shape control with low actuation complexity [19].

This work introduces a computational algorithm for the inverse design of a singly curved-creased sheet (i.e., a curved-creased ribbon) using a vector-based geometric representation combined with a genetic optimization framework. The formulation captures both folded and pinched configurations of a curved-crease ribbon and enables rapid numerical evaluation of candidate geometries. The resulting algorithm solves three representative inverse-design problems: (A) approximating a prescribed planar curve, (B) maximizing tip deflection through targeted actuation, and (C) reaching a specified point in three-dimensional space. The approach addresses the challenges of non-convexity and non-smoothness inherent to these design spaces, demonstrating robust convergence using stochastic optimization.

From a computational standpoint, the novelty of this study lies in (1) a unified vector representation that enables efficient calculation of three-dimensional deformations from discrete arc parameters, bridging geometric and mechanics-based modeling; (2) integration of bar-and-hinge simulation data to extend analytical solutions to pinched configurations, forming a hybrid analytical–numerical forward model; and (3) an inverse-design algorithm based on genetic optimization, which automates the determination of crease geometry and actuation schemes for prescribed shape objectives.

This paper introduces an innovative algorithmic framework for designing curved-crease origami structures capable of precise shape morphing and targeted deformation. By formulating a novel vector-based representation of both pinched and unpinched crease segments, the method enables accurate approximation of complex planar curves, maximized deflection through strategic actuation, and precise targeting of points in three-dimensional space. Unlike previous approaches reliant on repeating origami cells, this study uniquely demonstrates the versatility and effectiveness of single, curved-creased ribbons. These ribbons offer substantial advantages for engineering applications requiring adaptable, lightweight, and robust structures. Potential applications of this methodology span soft robotics, deployable structures, and adaptable components in infrastructure, highlighting the broad impact of tailored motion through curved-crease origami in advanced engineering design.

2. Formulating the Forward Process

Our work focuses on the inverse design of curved-crease origami strips, as shown in Figure 1. To craft a precise crease pattern and actuation scheme (CP/AS) for a desired shape, we first need a method to compute the shape of the crease based on its pattern parameters and pinch status. Our forward process seeks to develop a function that efficiently returns the Cartesian coordinates of the crease for given CP/AS parameters. Speed in computation ensures our inverse-design algorithm remains agile during numerous iterations, especially if we consider curved-crease strips consisting of multiple segments. Here, we will define the origami geometry underpinning our further analysis.

2.1. Geometry of a curved-crease strip

Considering the extensive CP/AS possibilities in curved-crease origami, it's practical to narrow down our model's geometry. We adopt a simple yet non-trivial origami geometry apt for diverse shape-fitting challenges. Our selected curved-crease geometry comprises a singular crease made of n discrete circular arc segments, as illustrated in Figure 1(a). The beginning of the first crease segment has a specific orientation to emulate structural restraints. The choice of arc segments facilitates variation in the curvature of the crease over its length, granting flexibility in shape adjustments. Every circular arc segment possesses an initial curvature, k_i^0, a length, l_i, and a pinch status, $f_i \in \{0, 1\}$ with 0 for no pinch and 1 for pinched. The dihedral

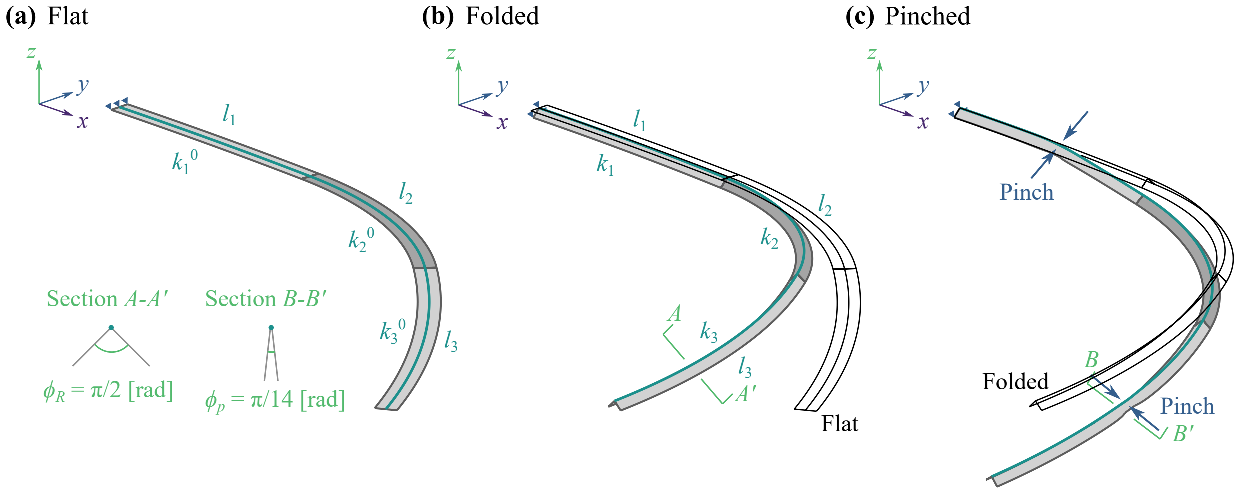


Figure 1: Representative geometry and deformations of curved-crease origami. **(a)** Discrete circular arc segments form the origami strip. Each segment has an initial curvature, k_i^0, and length, l_i. **(b)** Folding this strip increases the curvature, k_i (for segments where $k_i^0 \neq 0$). **(c)** Pinches at the middle of segments result in bends and twists of the origami that modify its final form.

angle after folding is $\phi_R = \pi/2$ [rad] or 90°. For $f_i = 1$, the segment gets pinched at its midpoint to $\phi_p = \pi/14$ [rad] or 12.9°, gradually unfolding to ϕ_R towards its ends. All segments have a sheet width, $w = 5$ [mm], on either side of the crease.

The first crease originates at $\mathbf{P}_0$, a position vector in Cartesian space. For our analyses, this start point is the origin, $\mathbf{P}_0 = \mathbf{0}$. The initial orientation aligns the crease tangent with the x-axis, transformed by the rotation matrix $\mathbf{R}_0$. When $\mathbf{R}_0 = \mathbf{I}$, the tangent parallels the global x-axis, as is consistent in our analyses, but not a strict requirement. The ending edges of preceding segments and the starting edges of subsequent ones have matching tangents and spatial locations. They are also oriented to ensure adjacent crease sheets form a continuous surface between segments.

2.2. Vector Representation of Curved-Crease Origami

The position at the end of a crease segment, $\mathbf{P}_j$, constructed from n discrete circular arc segments is determined by summing vectors as shown in Figure 2(c):

$$\mathbf{P}_j = \mathbf{P}_0 + \sum_{i=1}^{j} \mathbf{T}_i, \quad j = 1, 2, \ldots, n. \tag{1}$$

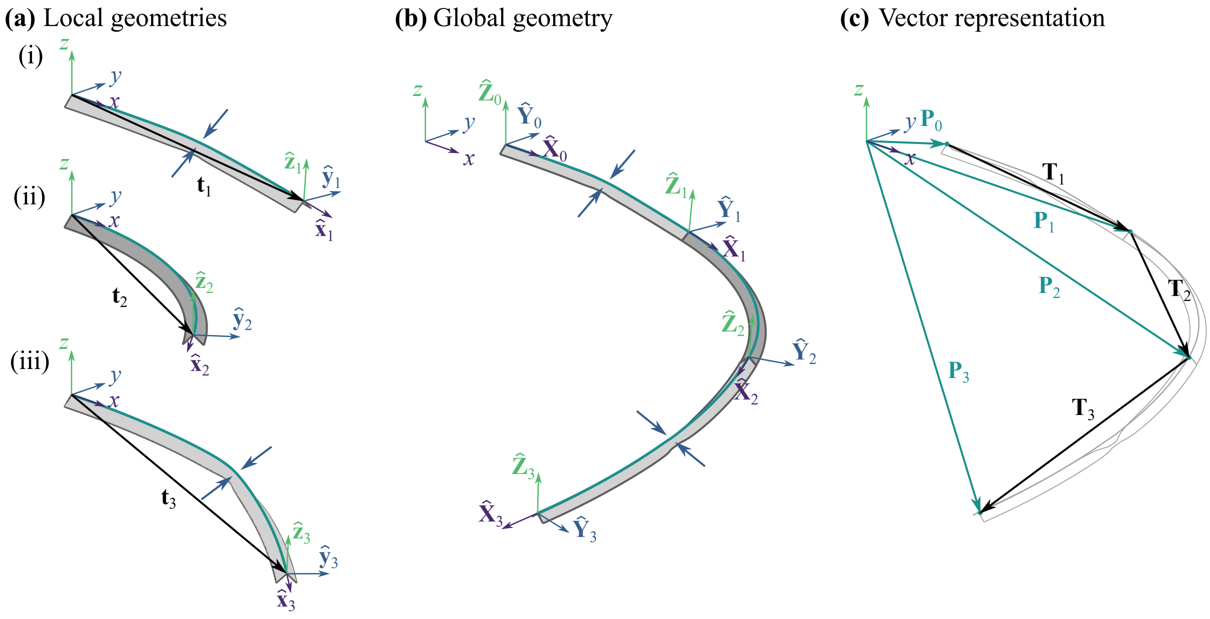
Here, $\mathbf{T}_i$ represents the global translation vector for the i^{th} segment's crease. This vector is influenced by the CP/AS parameters of the segment itself and those parameters of preceding segments.

The global translation vector components are computed by first examining individual segment deformations, then adjusting the segment positions within the global coordinate system, as depicted in Figure 2(b). For consistent analysis, we employ a local coordinate system where segments, irrespective of their specifications, share a common starting orientation and location, showcased in Figure 2(a). Each segment begins at the origin with a tangent aligned with the local x-axis. The local translation vector, $\mathbf{t}_i$, quantifies the translation from each segment's start to end.

The global vector $\mathbf{T}_i$ is determined by rotating its local counterpart:

$$\mathbf{T}_i = \mathbf{R}_{i-1}\mathbf{t}_i, \tag{2}$$

where $\mathbf{R}_{i-1}$ describes the rotation of the starting edge of the segment with respect to the global frame, which is equivalent to the rotation at the end



(a) Local geometries

(b) Global geometry

(c) Vector representation

Figure 2: Vector representation of segments in local and global spaces. **(a)** Individual segments, whether pinched or not, are assessed in local coordinates, beginning at the origin with the initial tangent on the x-axis. **(b)** The local vectors and end orientations facilitate the global assembly of the strip. **(c)** The global deformation vectors, combined with rotation matrices, determine the end location of each segment.

of the preceding segment. See Appendix A for details on how the rotation matrix is compiled based on the preceding segment geometries.

The local translation and rotation matrices for each segment depend on its specific attributes. For segments without pinches, the geometry alone can provide analytical solutions. The range of translations and rotations given by this analytical process is shown in Figure 3. For pinched segments, we employed a bar-and-hinge mechanical model [20] (using a sheet thickness, $t = 0.1$ [mm], modulus of elasticity, $E = 4$ [GPa], length scale, $L^* = 0.5$, and sheet width on either side of the crease, $w = 5$ [mm]), compiling a lookup table of deformed shapes for the optimization process. The range of translations and rotations given by the bar-and-hinge model is shown in Figure 4.

The bar-and-hinge formulation used here has been validated for curved-crease systems through comparisons to structural-mechanics theory, finite-element reference checks in high-fold regimes, and laser-scanned paper models, including guidance on twist tendencies and recommended mesh aspect ratios [20]. That prior work also identified in-plane strains local to creases in highly folded or pinched states, which the present formulation captures [19]. In all studies herein we adopt an aspect ratio, $\alpha = 5$, which lies within the validated range for folding and twist-sensitive predictions.

Further details on determining local rotation and translation matrices and how these are used to compute the global shape are available in the supplementary materials (see Appendix A). Moreover, we briefly compare the bar-and-hinge models with facet isometric mappings to situate this mechanics-based surrogate among common kinematic approaches.

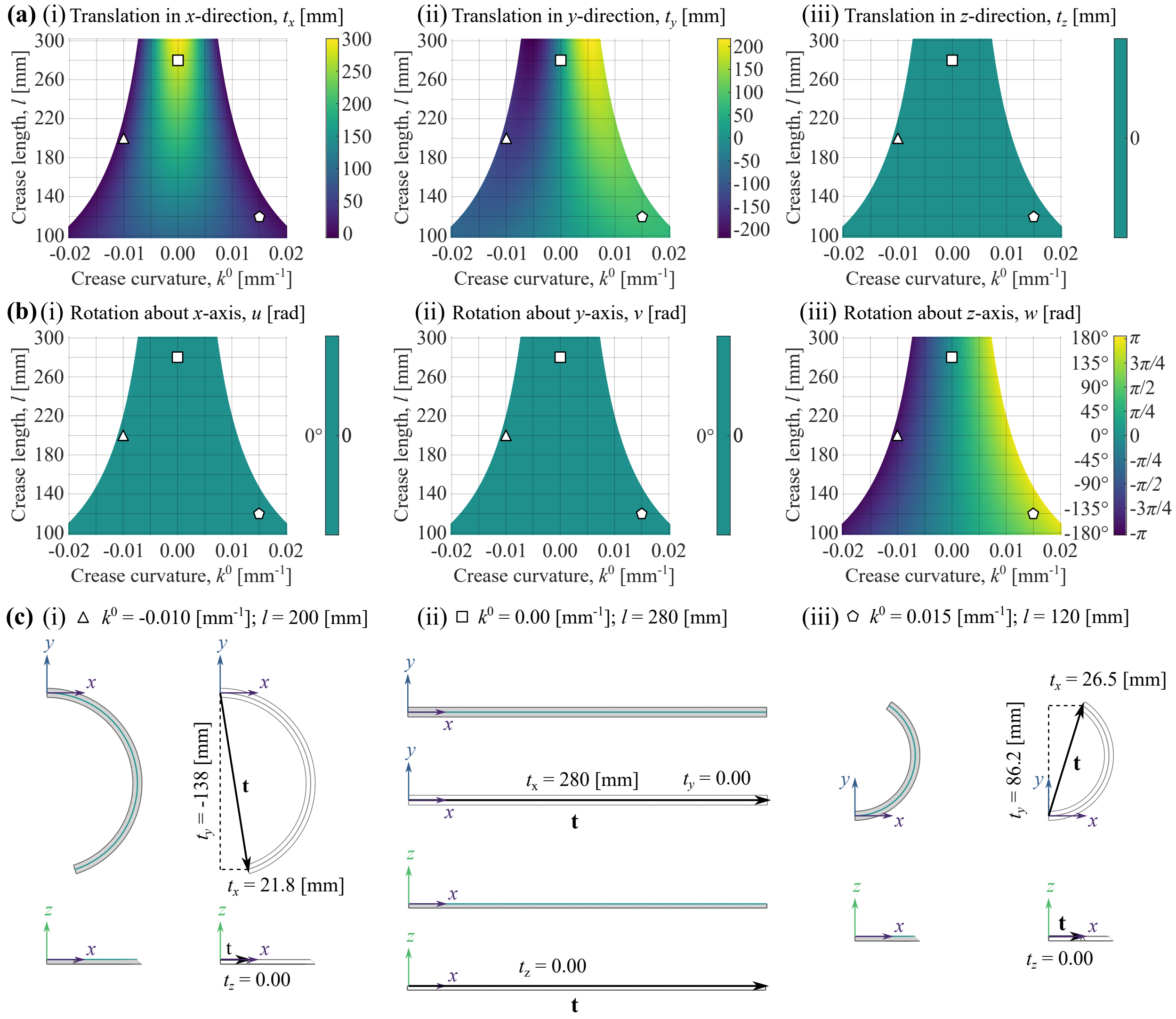


Figure 3: The translations and rotations at the end of unpinched segments. **(a)** The magnitudes and direction of translation from an unfolded to a folded state. **(b)** The rotation angles that occur from an unfolded to a folded state. **(c)** Three examples showing the folded state of segments with different curvatures and lengths. The translations and rotations due to folding are shown with corresponding symbols in (a) and (b).

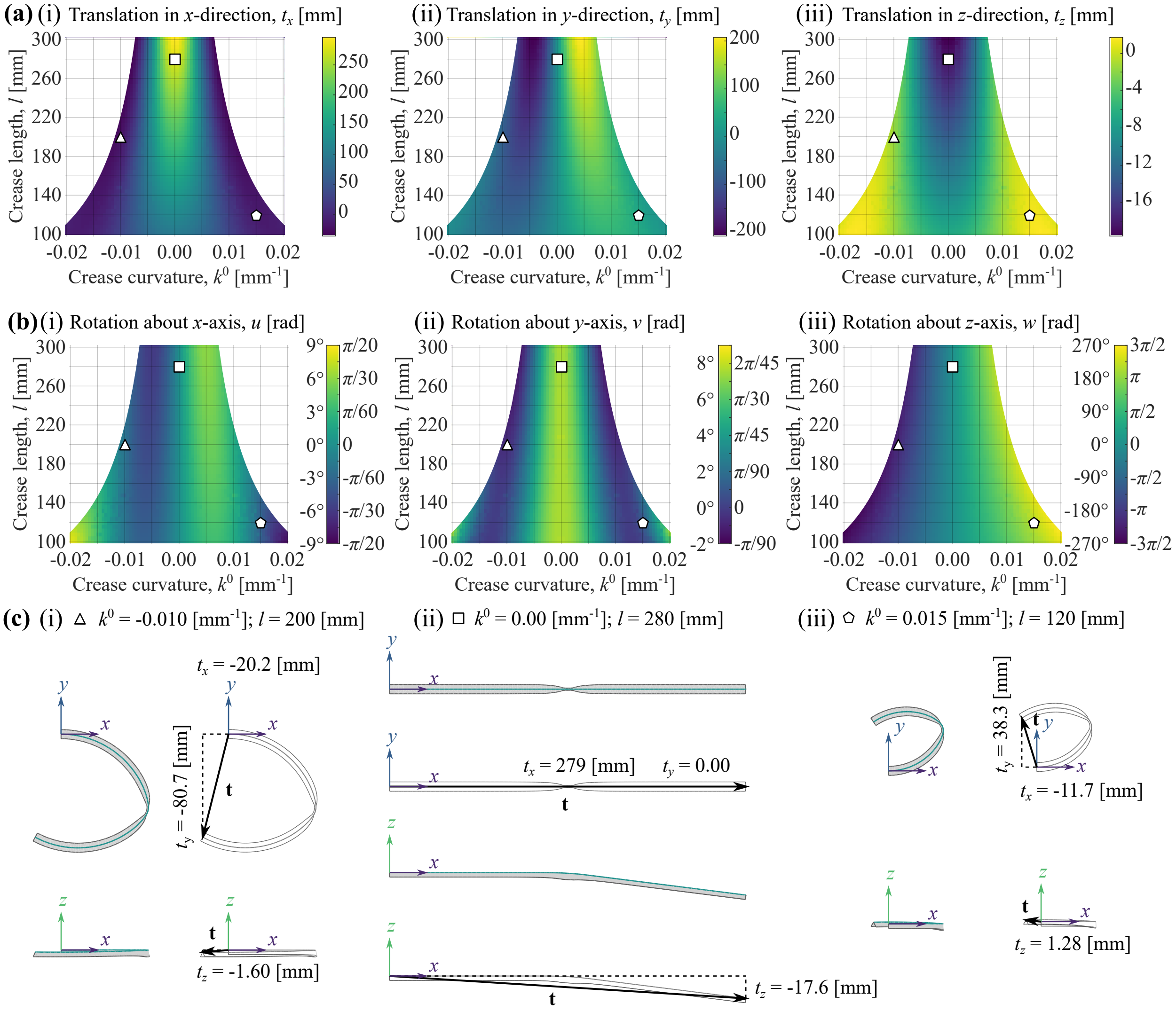


Figure 4: The translations and rotations at the end for folded-then-pinched segments. **(a)** The magnitude and direction of translation from an unfolded to a folded-then-pinched state. **(b)** The rotation angles that occur from an unfolded to a folded-then-pinched state. **(c)** Three examples showing the folded-then-pinched state of segments with different curvatures and lengths. The translations and rotations due to folding then pinching are shown with corresponding symbols in (a) and (b).

2.3. Comparison to facet-based isometric mappings

Facet-based isometric mappings idealize a folded sheet as piecewise-developable facets linked by straight creases and assume no in-plane stretch. This approximation is fast and effective for moderate folds and polyhedral geometries, but it diverges in the high-fold, pinched regimes central to our inverse-design problems. Near a pinch the sheet departs from isometry: Bending gives way to in-plane stretching, local crease curvature decreases and flattens within the pinched zone as the dihedral nears closure, and coupled bending–twisting emerges from gradients in fold rate and generator direction. Finite width also matters, because the spacing to edges sets a lever arm that increases global bending and produces a smaller, nonlinear increase in twist. These behaviors have been documented in our prior curved-crease studies [19] and are captured by the present bar-and-hinge surrogate, which includes bending, folding, and in-plane stretching with geometric nonlinearity and has been validated against structural-mechanics benchmarks and laser-scanned specimens. For the optimization problems here, which require thousands of forward evaluations in high-fold, twist-sensitive states, the mechanics-based model provides the required fidelity while remaining computationally tractable, whereas facet isometry omits the stretch-dominated transitions and width effects that govern response in pinched configurations.

3. Implementing the Inverse Design

Having established the forward process, we can compute the geometry of a creased strip consisting of multiple segments with various curvatures and pinch statuses. The solution space is composed of at least three dimensions (crease curvature, k_1, crease length, l_1, and pinch status, f_1) and expands based on the crease segment count, n. Using this design space, we can envision numerous inverse-design problems for shape fitting or functional shape morphing enabled by pinching actuation. In this section, we present three sample inverse design problems for creased strips.

3.1. Problem A - Fitting an Unpinched Origami to Planar Curves

In this problem, we restrict ourselves to folded, but unpinched crease segments, which constrains the target deformed crease to the xy-plane. Such a crease geometry can be determined analytically without invoking deformation data for the strip. The curved creases provide an unbounded domain for initial crease curvatures and a lower bound of zero for crease lengths. However, to prevent self-intersection, the post-folding sector angle, $|k_i|l_i = |k_i^0|l_i/\sin(\phi/2)$, must not exceed 2π [rad].

This extensive solution domain facilitates solving complex shape-fitting problems. We aim to identify a crease pattern that, when folded, approximates a planar curve, $\mathbf{c}_{\text{tar}}$. We choose target curves, including a parabolic segment, $\mathbf{c}_{\text{tar}}^{\text{par}}$, a logarithmic spiral, $\mathbf{c}_{\text{tar}}^{\text{log}}$, and a cosine wave, $\mathbf{c}_{\text{tar}}^{\text{cos}}$, defined parametrically as:

$$\mathbf{c}_{\text{tar}}^{\text{par}} = \left\{ s,\ \frac{2s^{3/2}}{30},\ 0 \right\}^T \ [\text{mm}], \quad 0 \leq s \leq 211.6, \tag{3}$$

$$\mathbf{c}_{\text{tar}}^{\text{log}} = \left\{ -e^s \cos s + 1,\ e^s \sin s,\ 0 \right\}^T \ [\text{mm}], \quad 0 \leq s \leq 5.362, \tag{4}$$

$$\mathbf{c}_{\text{tar}}^{\text{cos}} = \left\{ s,\ 20 \cos\left(\frac{s}{45.63}\right) - 20,\ 0 \right\}^T \ [\text{mm}], \quad 0 \leq s \leq 286.7. \tag{5}$$

Each curve has an arc length of 300 [mm], starts at the origin (when $s = 0$) and is depicted in Figure 7(b-d).

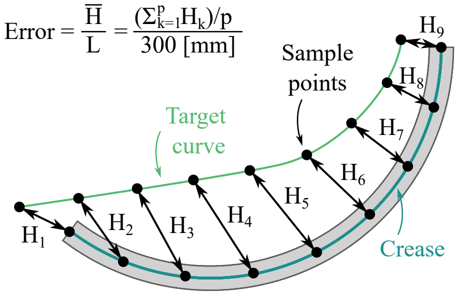

Error = $\frac{\overline{H}}{L} = \frac{(\Sigma_{k=1}^{p} H_k)/p}{300\ [mm]}$
Sample points
Target curve
H_1
H_2
H_3
H_4
H_5
H_6
H_7
H_8
H_9
Crease

Figure 5: Error quantification between a crease and a target curve. The normalized mean distance between two equidistant sets of p points represents the error that the optimization seeks, in Equation 6, to minimize.

Our goal is to minimize the error between the folded origami and target curves. The error, expressed as the average distance between two sets of equidistant sample points, $\bar{H}$, is normalized by the target crease's length ($L = 300$ [mm]), illustrated in Figure 5. For each curve, we performed optimization to obtain crease patterns that minimized this error. The shape fitting was repeated for varying crease segment numbers to evaluate how segment count impacts approximation accuracy. In essence, our optimization problem for fitting unpinched origami to a planar curve is:

$$\begin{aligned} &\text{minimize} \quad \frac{\bar{H}(k_i, l_i)}{L}, \\ &\text{subject to :} \\ &\qquad (1) \quad l_i \geq 0, \\ &\qquad (2) \quad \frac{k_i^0 l_i}{\sin(\phi_R/2)} \leq 2\pi \text{ [rad]}, \ \phi_R = \pi/2 \text{ [rad]}, \\ &\quad \text{for } i = 1, 2, \ldots, n. \end{aligned} \tag{6}$$

3.2. Problem B - Maximizing Tip Deflections After Pinching

Our objective in this problem is to identify a crease pattern and actuation scheme (CP/AS) maximizing the deflection between the folded and pinched states of an origami with n crease segments. The distance between the folded state, $\mathbf{P}_n^{\text{folded}}$, and the pinched state, $\mathbf{P}_n^{\text{pinched}}$, serves as our objective function. The goal is to minimize the negative of the distance between these points.

Given that pinching is allowed, we use the bar-and-hinge surrogate data, such as that presented in Figure 4(a-b), to determine the translation and rotation of individual segments and the system as a whole. Our search space is now limited to a range of crease lengths and curvatures that were initially simulated (i.e., the colored space in Figure 4(a-b)). Interpolation allows approximating values between individual simulations.

Potential self-intersections in the crease pattern can complicate fabrication and cause unmodeled displacements. To address this, we incorporated two constraints: an upper bound of π [rad] on the sector angle of a crease segment, and an algorithm detecting intersections between crease segments. This intersection algorithm evaluates potential collisions by projecting global translation vectors onto the xy-plane and checking for segment intersections.

The optimization problem is as follows:

$$
\begin{aligned}
&\text{minimize} \quad -\left|\mathbf{P}_n^{\text{pinched}} - \mathbf{P}_n^{\text{folded}}\right|, \text{subject to :}\\
&(1) \quad -0.02 \leq k_i^0 \leq 0.02 \ [\text{mm}^{-1}];\\
&(2) \quad 100 \leq l_i \leq 300 \ [\text{mm}];\\
&(3) \quad \frac{k_i^0 l_i}{\sin(\phi_R/2)} \leq \pi \ [\text{rad}], \quad \phi_R = \pi/2 \ [\text{rad}];\\
&(4) \quad \text{intersection} = \text{false},\\
&\text{for} \quad i = 1, 2, \ldots, n.
\end{aligned}
\tag{7}
$$

3.3. Problem C - Reaching a Target Point in Three Dimensions

Here, we aim to determine a CP/AS that results in the origami tip reaching a specific target in three-dimensional space, $\mathbf{P}_{\text{tar}} \in \mathbb{R}^3$. As this problem permits targeting points outside the xy-plane, we need to include pinching and again use simulated data. The objective function for this problem is the distance between the origami's end after actuation and the target point. We again employ the sector angle and intersection algorithm constraints to manage potential self-intersections. The optimization problem is:

$$
\begin{aligned}
&\text{minimize} \quad |\mathbf{P}_n - \mathbf{P}_{\text{tar}}|, \text{subject to :}\\
&(1) \quad -0.02 \leq k_i^0 \leq 0.02 \ [\text{mm}^{-1}];\\
&(2) \quad 100 \leq l_i \leq 300 \ [\text{mm}];\\
&(3) \quad \frac{k_i^0 l_i}{\sin(\phi_R/2)} \leq \pi \ [\text{rad}], \quad \phi_R = \pi/2 \ [\text{rad}];\\
&(4) \quad \text{intersection} = \text{false},\\
&\text{for} \quad i = 1, 2, \ldots, n.
\end{aligned}
\tag{8}
$$

Given the complexity of the design space and the binary nature of the

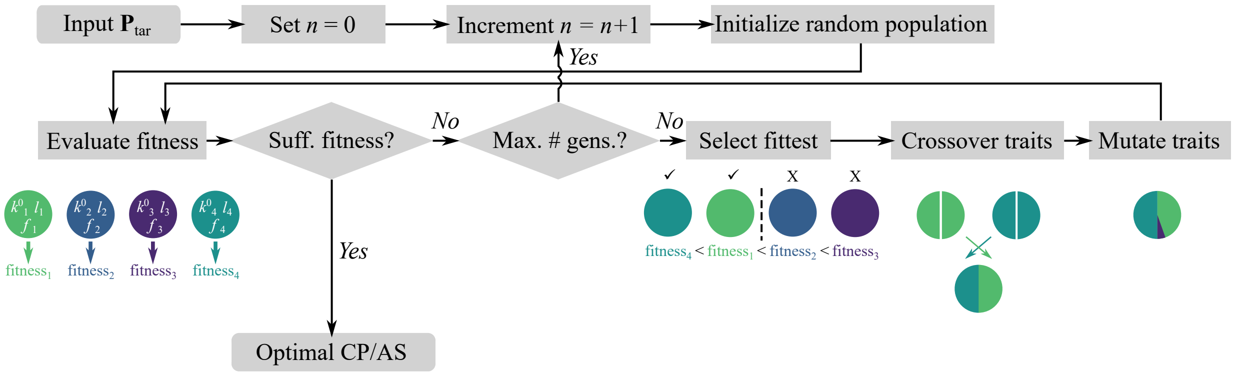


Figure 6: Inverse-design scheme to reach a target point in three dimensions (Problem C) using a genetic algorithm. The user begins by inputting a target point for the crease to reach at its end. Then, the algorithm loops through an increasing number of crease segments, using the genetic algorithm (GA) to find a crease pattern and actuation scheme (CP/AS) that gives a sufficient fitness ($|\mathbf{P}_n - \mathbf{P}_{\text{tar}}| \leq 1$ [mm]). This algorithm excels at finding solutions in a non-linear, non-convex, and non-smooth fitness function, such as the one given in this problem.

pinch status, we used a genetic algorithm (GA) to solve these optimization problems. Figure 6 shows the GA process followed by our code. Additional details on the GA and the convexity of the problems are given in supplementary materials (see Appendix B and Appendix C)

4. Results and Discussion

While the problems we aim to solve are complex, the genetic algorithm (GA) is capable of finding a sufficient solution to all three. Because the GA is not gradient-based, it is able to find optimal solutions in the non-convex and non-smooth design space. The GA is useful for solving all three inverse-design problems with minimal alteration, despite the drastic differences in the three problems. Furthermore, the GA provides solutions that even someone with experience working with curved-crease origami would not be able to predict on their own. The algorithm reliably converged to a sufficient solution without needing to alter the GA's parameters, and the results were identical when the algorithm was run multiple times with the same initial conditions.

The greatest limitation of the GA is its large computational expense, relative to typical gradient-based optimization algorithms. Alternatives such as combining a gradient-based search method with Monte Carlo randomized initial conditions could also be feasible, but given the size and complexity of the design space, may provide a less repeatable solution. The computational costs associated with using the GA is justified by the stability and repeatability we have found in using it. Moreover, the GA was run on a typical desktop computer (13th Gen Intel Core i7-13700K (3.40 GHz) CPU and 64 [GB] of RAM using parallel processing with 16 cores and 24 threads) and converged to a solution within minutes, so the computational costs are still practical.

4.1. Problem A - Fitting an Unpinched Origami to Planar Curves

For the target curves $\mathbf{c}_{\text{tar}}^{\text{par}}$, $\mathbf{c}_{\text{tar}}^{\text{log}}$, and $\mathbf{c}_{\text{tar}}^{\text{cos}}$, the GA minimized the error between the target curve and the folded creased strip for different numbers of segments, $n = 1, 2, \ldots, 6$. Figure 7 presents the results. For all target curves, single crease segments produced greater errors than multiple segments. This stems from the constant curvature of a single segment, which poorly approximates target curve curvatures. The logarithmic spiral, with its significant curvature slope, is especially challenging to approximate using constant curvature segments.

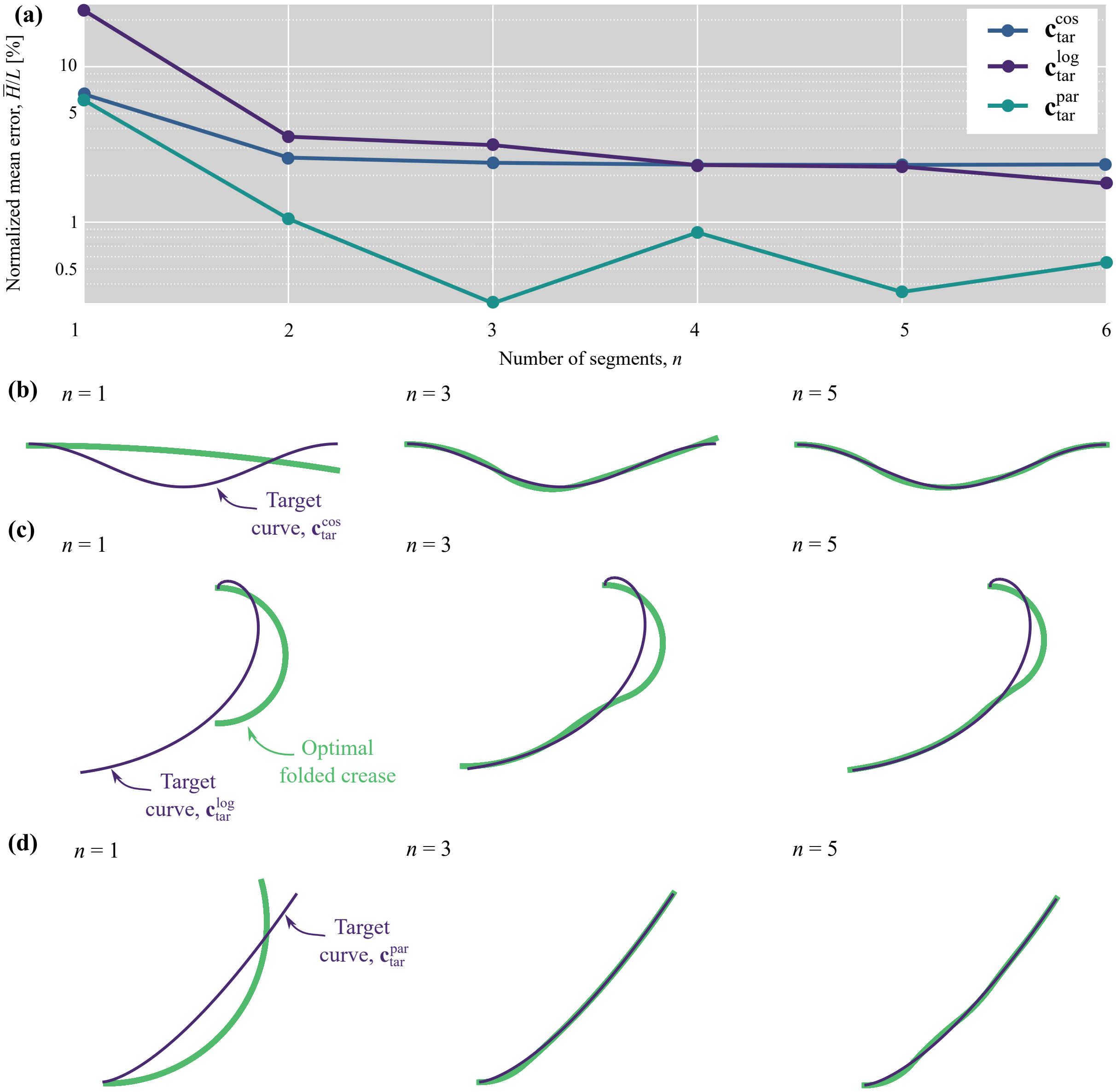


Figure 7: Shape fitting of unpinched creases to planar curves using the genetic algorithm. (a) Convergence of the three target shapes with average error quantified by $\bar{H}/L$ (the error is shown on a logarithmic scale). (b) Visual convergence for the cosine wave. (c) Visual convergence for the logarithmic spiral. (d) Visual convergence for the parabolic curve.

In contrast, the parabola, with its milder curvature slope, is more readily approximated by constant curvature arcs. Adding more segments generally reduces errors, but the improvement diminishes as segments increase. Most discrepancies arise in areas with rapidly changing curvature. Overall, our approach effectively approximated the shapes of planar curves.

4.2. Problem B - Maximizing the Tip Deflection After Pinching

Transitioning to three-dimensional shapes, the GA identified the crease pattern and actuation scheme (CP/AS) maximizing tip deflection from folded to pinched states for $n = 1, 2, 3$, and 4 segments, as shown in Figure 8. The maximal displacement for a single segment demands a balance of bending and twisting from a pinch (see supplementary materials for more information). A crease with excessive curvature and length limits z-direction deflection. A multi-segment CP/AS results in greater displacement than single segments. A recurrent result is that the first segment is moderately curved and long, always pinched, whereas subsequent segments are near straight and long. The bending and twisting of the initial segment causes a global rotation for the remainder of the system, while the other segments amplify this rotation into large global displacements. Figure 8(c) affirms that tip displacement grows nearly linearly with additional segments.

Although our method does not model the nonlinear trajectory during pinching, it establishes the maximum possible deflection, setting a benchmark for functional origami structure applications.

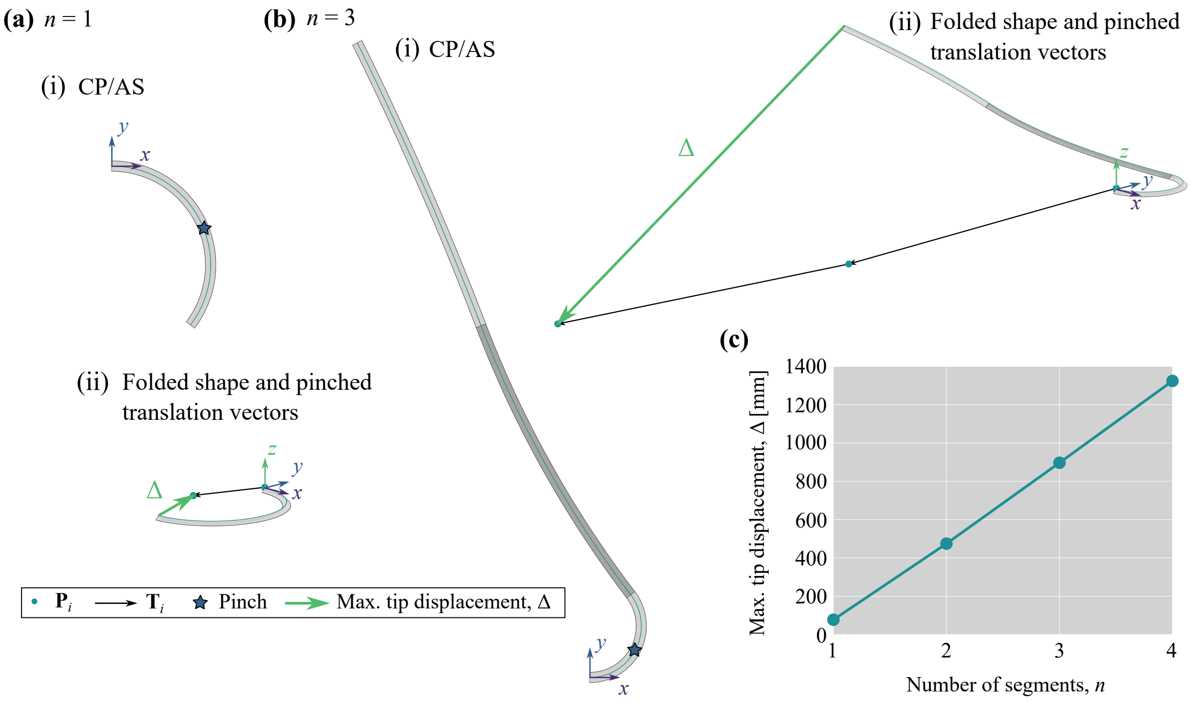


Figure 8: Maximizing tip displacement for curved strips with different segments. **(a)** A single segment with moderate initial curvatures achieves the largest displacement. **(b)** When more segments are included, only the first is pinched and has moderate curvature which leads to large global displacements. **(c)** Maximum displacements increase proportional to the number of segments.

4.3. Problem C - Reaching a Target Point in Three Dimensions

Here, we explored how to design crease strips that when folded and actuated can move their end position to a target point in three dimensional space. Utilizing the inverse-design scheme, we evaluated 605 target points surrounding the crease's starting position, $\mathbf{P}_0 = \mathbf{0}$. For all these points, a crease pattern and actuation scheme (CP/AS) meeting the constraints was determined. The target points were defined as: $P^x_{\text{tar}}, P^y_{\text{tar}} \in \{-100, -80, \ldots, 80, 100\}$ [mm], and $P^z_{\text{tar}} \in \{-10, -5, 0, 5, 10\}$ [mm]. Each solution provided details such as the number of segments, initial curvatures, lengths, and pinch statuses. We verified that each design ended within 1 [mm] of its intended target. Figure 9(a) visually presents the solutions, with the cell center denoting the target point and the color indicating the total crease length. The displayed plots manifest symmetry about the P^x_{tar}-axis, stemming from the alignment of the initial crease tangent. Recognizing this symmetry proves beneficial for minimizing future inputs to the design scheme.

The plane $P^z_{\text{tar}} = 0$ is the same plane as the crease's initial position. Solutions on this plane lack pinched segments. The shortest crease lengths are typically found in the positive P^x_{tar}-quadrants, highlighted by their darker shades. The lighter shaded cells (signifying longer crease lengths) in the positive P^x_{tar}-quadrants lie in target areas that are closer to the origin. Since the lower bound for crease lengths was set at 100 [mm], a single crease could not reach these points. For this plane, the negative P^x_{tar}-quadrants tend to present longer creases, a result of the applied constraints.

For planes differing from the xy-plane, pinching becomes a requisite to deviate from the initial plane. The character and location of the pinches are

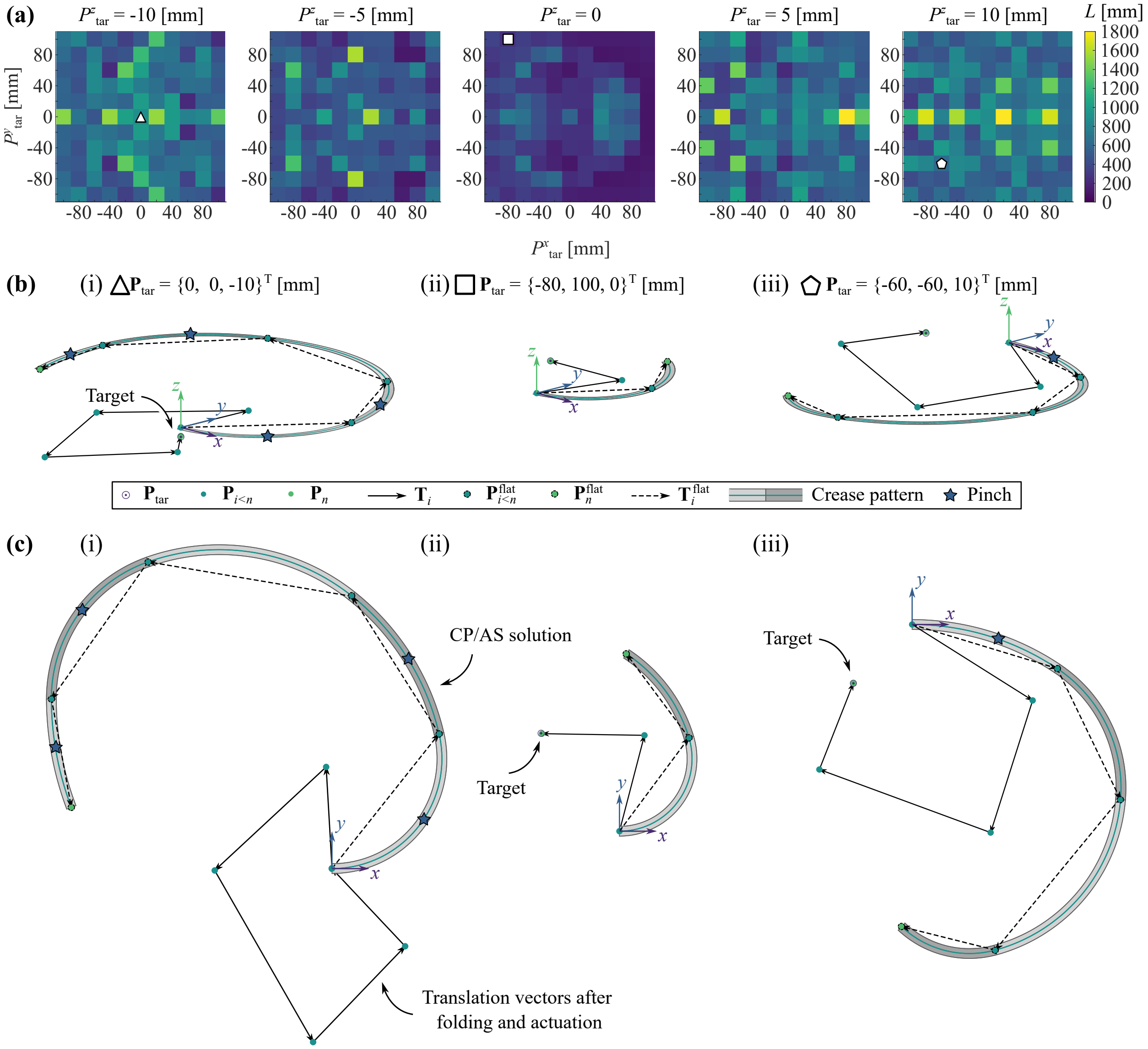


Figure 9: Finding a crease pattern for target points in $\mathbb{R}^3$. **(a)** Cell centers denote target positions with colors indicating the total length for an optimized crease. **(b)** Isometric views and **(c)** top views of translation vectors and solution crease patterns for three selected targets.

diverse, as showcased in Figure 9(i-iii). Patterns within these planes suggest that pinched single creases can only yield downward deflections resulting in several short lengths for $P^z_{\text{tar}} = -5$ [mm] and $P^z_{\text{tar}} = -10$ [mm]. On the other hand, sufficient twisting in multiple segments is needed for upward deviations in the P^z_{tar}-plane.

Crease lengths appear primarily arbitrary due to our non-emphasis on it in our fitness function. Constraining the GA to a mere 100 generations for time conservation might have precluded solutions with fewer segments. To prioritize resource conservation, incorporating crease length into the optimization would be imperative.

A distinguishing trait of pinched curved-crease origami is the gradual increase in twisting and bending with pinching intensity (see Appendix C and [19]). Incorporating this versatility could further increase the solution space, necessitating supplementary data for the GA. An advanced design could trace the crease end during pinching using our methodology, supplemented with this additional data.

The current solutions predominantly target proximate points to the crease origin. This poses challenges due to the self-intersection constraints. Augmenting the self-intersection algorithm might enable solutions for further P^z_{tar} points, potentially reducing the total crease length.

5. Conclusions

In this work, we present a computational framework for the inverse design of curved-crease origami, combining geometric modeling, simulation data, and optimization into a unified algorithmic workflow. The method introduces a vector-based representation of folded ribbons composed of discrete circular arcs, accommodating both folded and pinched configurations. Unpinched segments are resolved analytically, while pinched segments are captured through bar-and-hinge mechanical simulations, allowing accurate prediction of coupled bending and torsion effects. Building upon this forward model, an inverse-design algorithm employing a genetic optimization strategy efficiently solves non-convex design problems that are otherwise intractable using gradient-based techniques.

Through three representative problems (planar curve approximation, maximization of actuation-induced deflection, and spatial point targeting) the algorithm demonstrated strong convergence, computational stability, and design versatility. These results validate the framework's capability to automatically determine crease geometries and actuation schemes that achieve prescribed shape objectives with high precision.

From a computational perspective, the significance of this work lies in its generalizable numerical formulation for solving inverse-design problems in geometrically nonlinear systems. The integration of simulation-informed lookup tables with analytical relations represents a hybrid modeling strategy applicable beyond origami mechanics. Future extensions may incorporate continuous actuation modeling, multi-crease interactions, and dynamic response prediction.

Ultimately, this framework advances the automation of structural design for highly deformable systems, offering a foundation for next-generation simulation tools in morphing materials, deployable structures, and adaptive architectures.

CRediT authorship contribution statement

Steven R. Woodruff: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review and editing. **Evgueni T. Filipov:** Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing - review and editing.

Declaration of competing interest

The authors assert that they have no known competing financial interests or personal relationships that could have influenced the work reported in this article.

Data statement

The data that support the findings of this study are available from the corresponding author, Steven R. Woodruff, upon reasonable request.

Acknowledgments

The authors thank the Office of Naval Research for their financial support (Grant No. N00014–20–5–B001 #13209604). Additionally, the first author thanks the National Science Foundation for their financial support through the Graduate Research Fellowship Program (Grant No. DGE 1841052).

Appendix A. Calculating the global rotation matrices for pinched and unpinched creases

Appendix A.1. Determining the rotation matrices using Tait-Bryan angles

The global rotation matrix is calculated as a series of local rotations. The local rotations, calculated in the local coordinate system, determine the rotational transformation between the crease segment cross-section at the start of the crease to the end. We represent the orientation at the end of the crease segment in local coordinates with a set of basis vectors,

$$\mathbf{r}_i = \left\{ \hat{\mathbf{x}}_i, \quad \hat{\mathbf{y}}_i, \quad \hat{\mathbf{z}}_i \right\}, \tag{A.1}$$

where $\mathbf{r}_i$ is the local rotation matrix and $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$, and $\hat{\mathbf{z}}_i$ are the basis vectors at the end of the crease (see Figure 2(a)). Once the local rotation matrices are calculated for each of the structure's segments, we can calculate the value of the global rotation matrices with,

$$\mathbf{R}_i = \mathbf{R}_{i-1}\mathbf{r}_i, \tag{A.2}$$

Note that $\mathbf{R}_0$ is a given value that determines how the crease is oriented at the start of the crease (at $\mathbf{P}_0$). In this article, we choose to set the initial rotation matrix as the identity matrix ($\mathbf{R}_0 = \mathbf{I}$) in all cases, but any initial rotation can be chosen. It is also useful to note that the order of multiplication has a significant effect on the value of the global rotation matrix. Here, the prior global rotation matrix, $\mathbf{R}_{i-1}$, is right-multiplied by the local rotation matrix, $\mathbf{r}_i$, since the current global rotation matrix is an intrinsic rotation that occurs about the global basis vectors at the end of each crease, $\hat{\mathbf{X}}_{i-1}$, $\hat{\mathbf{Y}}_{i-1}$, and $\hat{\mathbf{Z}}_{i-1}$ (see Figure 2(b)).

By calculating values for the local translation vector and rotation matrix for each of the n segments in a structure, one can determine the position at the end of the crease. However, doing so requires saving twelve values per crease segment (three translation components and nine rotation components). We can reduce the number of values per crease segment required to model the transformations, which will be useful when employing surrogate data later on.

Reducing the required number of values describing the rotation can happen by representing the local rotation matrices with three rotation angles. Here, we employ Tait-Bryan angles, u_i, v_i, and w_i, that describe the angle the geometry is rotated through about the local x-, y-, and z-axes, respectively [21]. Tait-Bryan angles are similar to Euler angles, but the rotations are extrinsic, meaning the rotations occur about the fixed coordinate space axes, x, y, and z (see Figure A.10). One can calculate the local rotation matrices from the Tait-Bryan rotation angles with,

$$\mathbf{r}_i = \mathbf{r}_i^z \mathbf{r}_i^y \mathbf{r}_i^x = \begin{bmatrix} \cos w_i & -\sin w_i & 0 \\ \sin w_i & \cos w_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos v_i & 0 & \sin v_i \\ 0 & 1 & 0 \\ -\sin v_i & 0 & \cos v_i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos u_i & -\sin u_i \\ 0 & \sin u_i & \cos u_i \end{bmatrix}, \tag{A.3}$$

where $\mathbf{r}_i^z$, $\mathbf{r}_i^y$, and $\mathbf{r}_i^x$ are the rotation matrices for extrinsic, active rotations about the local z-, y-, and x-axes, respectively, for the i^{th} segment. Note that order of rotations occurs first about the x-axis, then the y-axis, and finally about the z-axis. However, the order of multiplication is reversed (left-multiplied), since the rotation is extrinsic.

The next step in the forward analysis is to generate the six translation

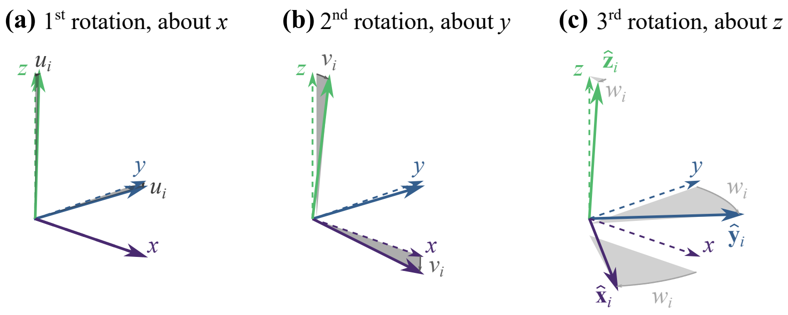

(a) 1st rotation, about x **(b)** 2nd rotation, about y **(c)** 3rd rotation, about z

Figure A.10: The orientation at the end of a crease can be represented using three angles. These Tait-Bryan angles describe the amount an object (here, a set of basis vectors) rotates in three sequential steps. (a) In the first step, the object is rotated about the x-axis by the angle, u_i. **(b)** Next, the object is rotated about the y-axis by the angle, v_i. **(c)** Finally, the object is rotated about the x-axis by the angle, w_i. These angles can be used to calculate the rotation matrix mapping the original x-, y-, and z-axes on to the $\hat{\mathbf{x}}_i$-, $\hat{\mathbf{y}}_i$-, and $\hat{\mathbf{z}}_i$-vectors.

and rotation values for a crease segment given the curvature, length, and pinch status of each segment. Figure A.11 shows a single crease in both the unpinched and pinched forms. Plotting the location at the end of the crease during the pinching process shows that the path is highly nonlinear. We chose to limit the pinch status to a binary, pinched or unpinched, to avoid having to collect data modeling varying degrees of pinching. Since the pinch status is binary, we must assess the deformed shape of pinched and unpinched segments separately, starting with the simpler case of an unpinched segment.

Appendix A.2. Local Analysis of Unpinched Origami

When a creased annulus sector is folded to a uniform dihedral angle and the crease is held planar, the crease retains its circular shape. However, the magnitude of the curvature of the crease increases as the dihedral angle moves

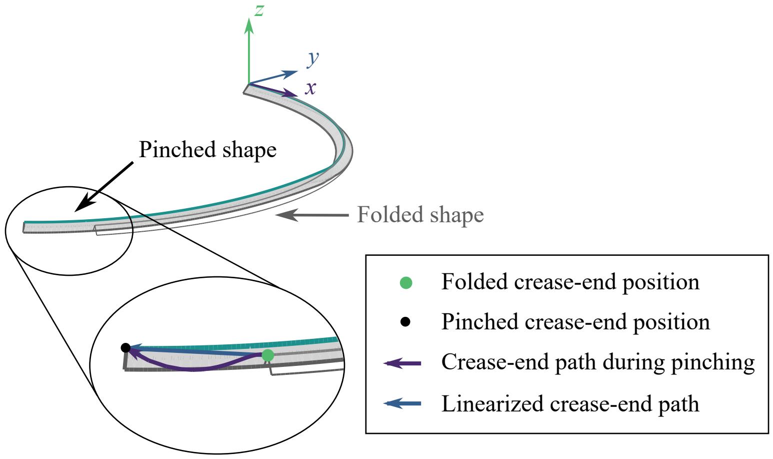

z
y
x
Pinched shape
Folded shape
Folded crease-end position
Pinched crease-end position
Crease-end path during pinching
Linearized crease-end path

Figure A.11: The end of a crease follows a nonlinear path during pinching. Using a linear approximation to predict the end of a crease for varying pinch angles, ϕ_p, would be overly erroneous. Thus, the pinch status is binary, rather than continuous.

away from π [rad] (unfolded). The curvature, with respect to the dihedral angle, can be calculated with,

$$k_i = \frac{k_i^0}{\sin(\phi/2)}, \tag{A.4}$$

where k_i is the signed curvature of the crease after folding, k_i^0 is the signed curvature of the crease before folding, and ϕ is the dihedral angle of the crease (see Figure 1(a-b)).

In equilibrium, a uniformly folded, circular creased annulus sector will exhibit small torsion at the ends of the crease. However, this torsion leads to small deflections from the initial plane of the crease (a distance of about 0.6% the length of the crease [22]). For simplicity, we assume that if the crease is uniformly folded (i.e., unpinched), then the crease will remain planar. Furthermore, we can restrain the origami in such a way that the crease always lies in the original plane (in this case, the local xy-plane). With these assumptions, the crease geometry is simplified to a circular sector. Subsequently, the translation components and rotation angles in the local coordinate system can be calculated using the curvature and length of the crease.

For instance, the local translation vector, $\mathbf{t}_i$, can be calculated with,

$$\mathbf{t}_i = \begin{cases} \left\{ \frac{1}{|k_i|} \sin\left(|k_i| l_i\right), \quad \frac{1}{k_i}\left[1 - \cos(|k_i| l_i)\right], \quad 0 \right\}^{\mathrm{T}}, & k_i \neq 0 \\ \left\{ l_i, \quad 0, \quad 0 \right\}^{\mathrm{T}}, & k_i = 0 \end{cases}. \tag{A.5}$$

Additionally, the rotation angles can be found with,

$$u_i = 0; \qquad v_i = 0; \qquad w_i = k_i l_i. \tag{A.6}$$

Note that the rotation about the z-axis, w_i, is identical to the signed sector angle of the circular arc. With these rotation angle, one can calculate the local rotation matrix, $\mathbf{r}_i$, using Equation A.3. Equations A.5 and A.6 have been evaluated for a range of crease curvatures and lengths, as shown in Figure 3(a,b), respectively.

Uniform folding allows the crease to reach various points in the original, xy-plane (see Figure 7 for examples). However, one must include pinching the origami to reach points off this plane.

Appendix A.3. Local Analysis of Pinched Origami

When determining the deformed shape of pinched origami in equilibrium, we cannot assume that the shape of the crease will be circular. The pinching induces a local bending and global twisting in the crease segment, which is not easily described with common shapes, such as a circle. To determine the transformation data required for our forward process, we need to use a method that goes beyond simple geometric analysis. Here, we use the bar-and-hinge model, a simplified, mechanics-based simulation tool, to calculate the shape of the crease segment after folding and pinching [20].

Given the speed and reliability of the bar-and-hinge model, we can determine the deformed shape of over four thousand crease segments, of various curvatures and lengths, within a couple hours. We can then extract the translation vector and rotation matrix data from the deformed shape nodes and use the values in our forward process.

First, we need to define the model parameters. We chose parameters that mimic those of polyester sheets, with the modulus of elasticity, $E = 4000$ [MPa], sheet thickness, $t = 0.1$ [mm], length scale, $L^* = 0.5$ [mm], width of sheets beside the crease, $w = 5$ [mm], and aspect ratio, $\alpha = 5$. It is important that the aspect ratio is kept constant between all crease segments to ensure a uniform deformation response (since the element stiffness is sensitive to the aspect ratio). To do so, each geometry is discretized with an identical mesh size (the distance between each node on the crease is about 1 [mm]). Therefore, longer crease segments will have more elements than shorter geometries.

Second, we need to determine the domain of the lengths that we will test and the resolution of the dataset. We start by determining a lower bound for the crease lengths. One constraint on this forward process is that the dihedral angle at the ends of each crease segment must always be the same to ensure that the segments will fit together smoothly. We chose to fold the crease segments to $\phi_R = \pi/2$ [rad] $= 90^\circ$, since this is a realistic angle that can be created with paper and polyester sheet prototypes with ease. Thus, the dihedral angle at the ends of any pinched crease segment should be $\pi/2$ [rad]. We found that if the crease length is less than 100 [mm], the dihedral angle at the ends of the segments might not be $\pi/2$ [rad]. Therefore, the lower bound for the crease length is 100 [mm]. We chose the upper bound for the crease length to be 300 [mm] to give a sufficiently large domain, but without adding an overwhelming amount of values to the dataset. We tested every crease length at an increment of 4 [mm] (i.e., $l_i \in \{100, 104, 108, \ \ldots, \ 296, 300\}$ [mm]) so that the mesh at the ends of the crease would be identical and to preserve the aspect ratio, $\alpha = 5$. This gives fifty-one crease lengths to test.

Finally, we need to determine the domain of the initial crease curvatures, considering self-intersection of the crease segment upon pinching. For certain crease curvatures and lengths, pinching might result in such large bending deformations that the ends of the crease intersect. Since the forward process does not account for collisions, such self-intersection is not permissible. We observed that when the sector angle is small enough, $\theta = |k|l = |k^0|l/\sin(\phi/2) \leq \pi$ [rad], self-intersection upon pinching does not occur. With this in mind, we can set the lower bound to a value that will fall within this constraint, considering that the lower bound for the crease lengths is 100 [mm] and the resolution is 4 [mm]. As such, we chose a lower bound of -0.02 [mm^{-1}] for the initial crease curvature. Since the constraint is symmetrical about the initial crease curvature, we chose an upper bound of 0.02 [mm^{-1}]. We tested every initial crease curvature at an increment of 5×10^{-4} [mm^{-1}], a step size that gives a meaningful difference in the deformed shape. This gives eighty-one initial crease curvatures to test.

Overall, there are, at most, 4,131 unique crease segments to fold and pinch in the bar-and-hinge model. However, many of these geometries violate the self-intersection condition and were excluded from the analysis. For the geometries that fall within the constraints, the bar-and-hinge model determined the deformed shape, which was saved for later processing. The next step was to distill the deformed shape data to the six transformation values we needed, so that we could generate a grid of data from which forward-process values could be interpolated.

Since every crease tested in the bar-and-hinge model was positioned and oriented identically (with the crease starting at the origin and aligned with the local x-axis), the only values we need from the bar-and-hinge deformed shape data are the positions of the three nodes at the end of the crease and sheets. Figure A.12 shows an example geometry and the required three nodes: the position at the end of the crease, $\mathbf{p}_i$, the position at the end of the left sheet, $\mathbf{p}_i^l$, and the position at the end of the right sheet, $\mathbf{p}_i^r$.

The local translation vector for each model was found using,

$$\mathbf{t}_i = \left\{ t_x, \quad t_y, \quad t_z \right\}^{\mathrm{T}} = \mathbf{p}_i. \tag{A.7}$$

To determine the rotation angle values, we need to start by defining the plane at the end of the crease segment that must align with the other ends of the segments. We define this plane with a set of basis vectors using the three bar-and-hinge node positions. The first basis vector, $\hat{\mathbf{x}}_i$ is the normal vector to the end plane and is found with,

$$\hat{\mathbf{x}}_i = \left\{ \hat{x}_i^x, \quad \hat{x}_i^y, \quad \hat{x}_i^z \right\}^{\mathrm{T}} = \frac{(\mathbf{p}_i^r - \mathbf{p}_i) \times (\mathbf{p}_i^l - \mathbf{p}_i)}{|(\mathbf{p}_i^r - \mathbf{p}_i) \times (\mathbf{p}_i^l - \mathbf{p}_i)|}. \tag{A.8}$$

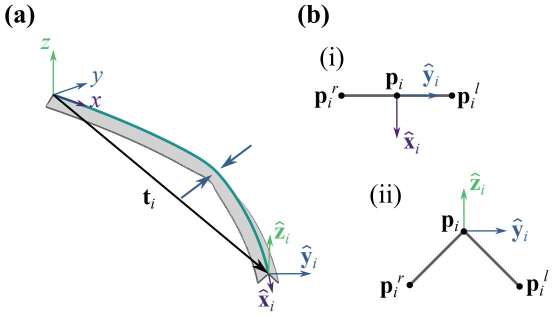


Figure A.12: The end of a pinched crease segment defined using three nodes. **(a)** First, the structure is folded and pinched in the local coordinate system using the bar-and-hinge method. **(b)** Next, the three nodes at the end of the structure are extracted to calculate the basis vectors used to calculate the Tait-Bryan angles for this geometry.

The next basis vector defines how the bottom of the sheets are oriented. We define the $\hat{\mathbf{y}}_i$ vector as the vector aligning with the line segment between the bottom edges of the sheet using,

$$\hat{\mathbf{y}}_i = \left\{ \hat{y}_i^x, \quad \hat{y}_i^y, \quad \hat{y}_i^z \right\}^{\mathrm{T}} = \frac{\mathbf{p}_i^l - \mathbf{p}_i^r}{|\mathbf{p}_i^l - \mathbf{p}_i^r|}. \tag{A.9}$$

Finally, the last basis vector, $\hat{\mathbf{z}}_i$, must satisfy orthogonality. That is,

$$\hat{\mathbf{z}}_i = \left\{ \hat{z}_i^x, \quad \hat{z}_i^y, \quad \hat{z}_i^z \right\}^{\mathrm{T}} = \hat{\mathbf{x}}_i \times \hat{\mathbf{y}}_i. \tag{A.10}$$

Note that the $\hat{\mathbf{x}}_i$ vector is equivalent to the tangent vector at the end of the crease, and the $\hat{\mathbf{z}}_i$ vector is the unsigned binormal vector.

From these three basis vectors, one can define the rotation matrix that rotates the starting plane of the crease segment to the ending plane (i.e., the connection surface). That is,

$$\mathbf{r}_i = \left\{ \hat{\mathbf{x}}_i, \quad \hat{\mathbf{y}}_i, \quad \hat{\mathbf{z}}_i \right\} = \begin{bmatrix} \hat{x}_i^x & \hat{y}_i^x & \hat{z}_i^x \\ \hat{x}_i^y & \hat{y}_i^y & \hat{z}_i^y \\ \hat{x}_i^z & \hat{y}_i^z & \hat{z}_i^z \end{bmatrix}. \tag{A.11}$$

Again, saving all nine values in the rotation matrix would be cumbersome and unnecessary. Therefore, we calculated the Tait-Bryan rotation angles from the rotation matrix using,

$$v_i = -\arcsin(\hat{x}_i^z), \tag{A.12}$$

$$u_i = \begin{cases} \operatorname{atan2}(-\hat{y}_i^x,\, -\hat{z}_i^x), & \hat{x}_i^z = 1 \\ \operatorname{atan2}(\hat{y}_i^x,\, \hat{z}_i^x), & \hat{x}_i^z = -1 \\ \operatorname{atan2}\left(\dfrac{\hat{y}_i^z}{\cos v_i}, \dfrac{\hat{z}_i^z}{\cos v_i}\right), & \text{otherwise} \end{cases}, \tag{A.13}$$

and

$$w_i = \begin{cases} 0, & \hat{x}_i^z = \pm 1 \\ \operatorname{atan2}\left(\dfrac{\hat{x}_i^y}{\cos v_i}, \dfrac{\hat{x}_i^x}{\cos v_i}\right), & \text{otherwise} \end{cases}. \tag{A.14}$$

Note the condition for when $\hat{x}_i^z = \pm 1$. This value represents a crease that is pointing exactly up or down (parallel to the local z-axis). In this instance, the rotation loses a degree of freedom and the value of the w_i angle becomes arbitrary, a phenomenon called Gimbal lock [21]. It is unlikely that a crease segment will fold to such an extreme inclination, but consideration for Gimbal lock is included for completeness.

After running the bar-and-hinge simulations and calculating the six transformation values, we saved the data for the forward-process algorithm to interpolate from later on. These data are plotted in Figure 4.

With Equations A.5 and A.6 and the bar-and-hinge data shown in Figure 4(a,b), we can determine the local translation vector and rotation matrix for any crease segment with $k_i \in [-0.02, 0.02]$ [mm^{-1}], $l_i \in [100, 300]$ [mm], and $f_i \in \{0, 1\}$. We can use these values to find the location at the end of the crease segments using the relationships given in Section 2.2. Therefore, the forward process is defined.

Appendix B. Problem Range and Non-Convexity

With such a broad solution space for all three inverse-design problems, finding the optimal solution can be challenging. Consider the solution space for *Problem B - Maximizing Tip Deflections After Pinching.* Figure B.13(a-b) shows the tip displacement values for a single-segment crease for all admissible crease patterns. These plots were calculated by taking the difference between the local translation values in Figures 4 (pinched crease segments) and 3 (unpinched crease segments). The maximum displacements from a single pinched crease lie along a contour, rather than in an obvious minimum. When multiple creases are added, and the pinch status can be varied, the objective function becomes even more complex.

Figure B.14 shows the variety of points in space that a single crease pattern can reach by varying the location of pinches. The versatility of this forward process gives a large solution space from which one will need to find a crease pattern and actuation scheme (CP/AS) that reaches a given target point.

Additionally, the objective function for point-fitting (Problem C) is, overall, non-convex. When simplifying the problem to just one crease segment with a pinch, as shown in Figure B.15, notice that there are multiple minima for many of the target points (here, we define the objective functions as the distance between the end of the crease, $\mathbf{P}_1$ and the target point). When the target point lies in the xy-plane, and the segment is not pinched, the function

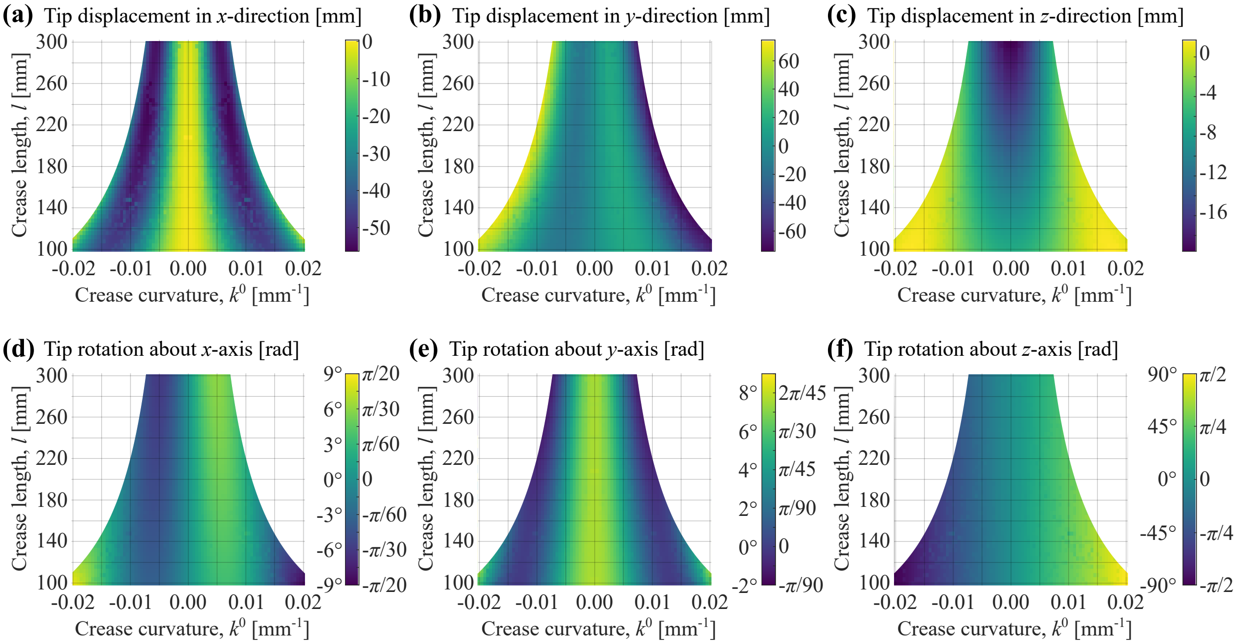


Figure B.13: Tip deflection and rotation data due to pinching. After a single origami crease is folded, pinching deforms the shape of the structure, leading to deflections at the tip **(a)** in the x-direction, **(b)** in the y-direction, and **(c)** in the z-direction, along with rotations of the tip **(d)** about the x-direction, **(e)** about the y-direction, and **(f)** about the z-direction

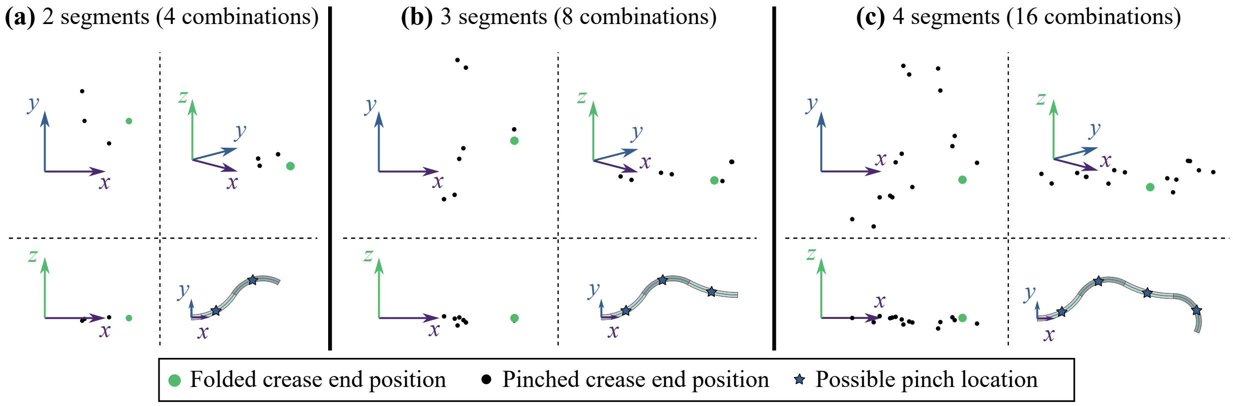


Figure B.14: All crease-end locations possible with various pinching combinations. A single crease pattern can reach a wide range of points in space by varying the combination of pinches on the structure, as shown for **(a)** two, **(b)** three, and **(c)** four crease segments (crease pattern and possible pinch locations shown).

surface might be convex, but this restriction on the target points is far less interesting, since the problem is reduced to finding a circular arc that reaches the target. Including more crease segments and the option of pinching each segment or not makes the problem much more valuable to someone trying to determine a crease pattern the reaches a target in a wide variety of points in three-dimensional space, including the xy-plane.

The nature and size of the solution space poses a problem in our attempt to find the CP/AS that reaches a given target point. Since the objective function is non-convex, one cannot rely on traditional convex optimizers and nonlinear solvers for all target points. As such, we will need to use a more versatile optimizer to achieve our goal.

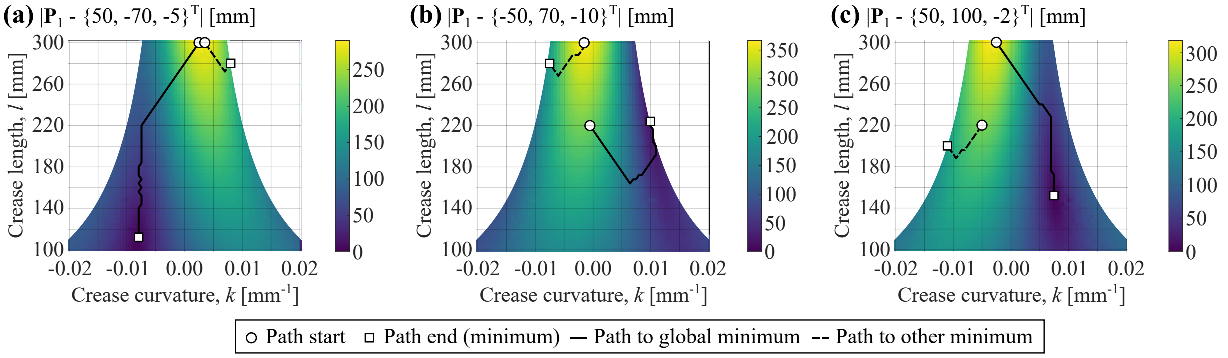


Figure B.15: The objective functions for problems with pinched creases are not usually convex. These plots show the distance between the end of a crease composed of one pinched segment, $\mathbf{P}_1$, and three different target points. The lines show the paths of a discrete gradient descent of the objective function towards different minima. The paths in solid lines converge towards the global minimum, while the dashed lines do not. Multiple minima in the objective means the problem is not convex, complicating the optimization.

Appendix C. Finding Optimal Solutions with a Genetic Algorithm

The objective functions for these problem are all non-linear, often non-convex, and non-smooth for Problems B and C (since the pinch status gives an integer parameter). As such, we will need to use an optimization method capable of searching such a space. The method we chose to use to solve these problems is a guided, random-search method called a genetic algorithm (GA). Unlike traditional, gradient-based optimization methods, GA does not calculate partial derivatives of the objective function (or fitness function, a term commonly used in GA). This allows one to find minima for problems with integer or binary inputs. Furthermore, GA is well suited to find the global minimum of a fitness function, and is robust against non-convex problems [23, 24].

Given the complexity of the fitness functions we are interested in searching, GA is a reasonable option that will find a CP/AS that minimizes each problem. The major limitation of GA is that running the optimization typically takes longer than gradient-based methods. However, because the forward process uses closed-form equations and surrogate data collected from the bar-and-hinge method, the computational cost will be much less than if the deformed shape of the origami had to be analyzed using a mechanics-based method in real-time. Additionally, GA is considered a “black-box" process, so the action of finding the global minimum is more complicated than calculating a gradient. However, the CP/AS found using the GA can be tested with the forward process to ensure the objective has been met and meets the problem constraints. Moreover, GA has been used to solve complex problems in the geometric design and mechanics of metamaterials and thin-sheet structures with great success [25, 26, 27].

GA is analogous to biological evolution, where “survival of the fittest" generates sufficient solutions to problems. Unlike evolution, the user determines how fitness is measured. The fitness functions used in each of our problems are the objective functions described in Equations 6, 7, and 8.

For these problems, the GA works by generating a population of CP/AS with random input values. Then, the population is evaluated for, and sorted by, fitness using the fitness measure. The fittest portion of the population will crossover traits (analogous to sexual reproduction and the exchange of chromosomes between parents) and generate a new generation of CP/AS children with traits from the fittest parents. In these problems, the traits of the individuals are the initial crease curvature, crease length, and pinch status (for Problems B and C) of the CP/AS, encoded into binary strings that exchange bits during crossover. Some children CP/AS will undergo random mutations in their input parameters, allowing the algorithm to explore other parts of the fitness function if the population is converging towards a local minimum that is not the global minimum. The process of generating a new population and evaluating the fitness of each individual continues until either a CP/AS gives a sufficient solution, the change in fitness is negligible, or if the number of generations (loops) exceeds a set value. For Problem A, the algorithm stops when the average error is less than 1 [mm]. For Problem B, there is no sufficient solution limit, and the algorithm stops when the change in fitness is negligible. For Problem C, the algorithm stops when the distance between the target point and crease end are within 1 [mm].

For Problems A and B, we looked for the optimal solution for various crease segment numbers, looking to see how the number of creases impacts the shape-fitting. For Problem C, we looked for a CP/AS that reached the target point, regardless of the number of creases. Although there is no explicit consideration for the length of the crease in the fitness functions, implicitly we limited the crease length by setting the number of crease segments, n, to one, attempting to find a solution with the GA, and adding segments until a solution was found. Figure 6 shows a flowchart explaining the algorithm for Problem C.

We used the `ga()` function in MATLAB to implement the optimization algorithm and find minimal solutions to the three inverse-design problems. The options we used in the analysis follow the recommended, default parameters for each type of problem (e.g., problems with linear constraints, integer

Table C.1: MATLAB GA function parameters used. For parameters not indicated, we used the default values found in [28].

Parameter	Problem A	Problem B	Problem C
`FunctionTolerance`	1×10^{-4}	1×10^{-4}	1×10^{-6}
`ConstraintTolerance`	1×10^{-6}	1×10^{-6}	1×10^{-3}
`MaxGenerations`	1000	1000	500
`FitnessLimit`	$-\infty$	$-\infty$	1

variables, etc.) for the MATLAB function [28]. In cases where an optimal solution that satisfied the minimum fitness was not found, parameters such as the function tolerance and maximum number of generations were altered until an appropriate solution was found. See Table C.1 for specific parameters used in each problem. Note that the `ga()` function automatically checks for solution convergence. Additionally, we visually confirmed convergence for a random sample of tests among the thousands performed.

References

[1] R. J. Lang, A computational algorithm for origami design, in: Proceedings of the 12th Annual ACM Symposium on Computatinoal Geometry, Philadelphia, PA, USA, 1996, pp. 98–105, https://doi.org/10.1145/237218.237249.

[2] T. Tachi, Origamizing polyhedral surfaces, IEEE Transactions on Visualization and Computer Graphics 16 (2) (2010) 298–311, https://doi.org/10.1109/TVCG.2009.67.

[3] K. Fuchi, A. R. Diaz, Origami design by topology optimization, Journal of Mechanical Design 135 (11), https://doi.org/10.1115/1.4025384 (2013).

[4] K. Fuchi, P. R. Buskohl, G. Bazzan, M. F. Durstock, G. W. Reich, R. A. Vaia, J. J. Joo, Origami actuator design and networking through crease topology optimization, Journal of Mechanical Design 137 (9), https://doi.org/10.1115/1.4030876 (2015).

[5] L. H. Dudte, E. Vouga, T. Tachi, L. Mahadevan, Programming curvature using origami tessellations, Nature Materials 15 (5) (2016) 583–588, https://doi.org/10.1038/nmat4540.

[6] H. Nassar, A. Lebée, L. Monasse, Curvature, metric and parametrization of origami tessellations: Theory and application to the eggbox pattern, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473 (2197) (2017) 20160705, https://doi.org/10.1098/rspa.2016.0705.

[7] A. Walker, T. Stankovic, Algorithmic design of origami mechanisms and tessellations, Communications Materials 3 (1) (2022) 1–8, https://doi.org/10.1038/s43246-022-00227-5.

[8] F. Feng, X. Dang, R. D. James, P. Plucinsky, The designs and deformations of rigidly and flat-foldable quadrilateral mesh origami, Journal of the Mechanics and Physics of Solids 142 (2020) 104018, https://doi.org/10.1016/j.jmps.2020.104018.

[9] Y. Chen, L. Fan, Y. Bai, J. Feng, P. Sareh, Assigning mountain-valley fold lines of flat-foldable origami patterns based on graph theory and mixed-integer linear programming, Computers & Structures 239 (2020) 106328, https://doi.org/10.1016/j.compstruc.2020.106328.

[10] R. He, Y. Chen, J. Liang, Y. Sun, J. Feng, P. Sareh, Crystallographically programmed kirigami metamaterials, Journal of the Mechanics and Physics of Solids 193 (2024) 105903, https://doi.org/10.1016/j.jmps.2024.105903.

[11] R. He, Y. Chen, J. Shi, Y. Bai, J. Feng, Programming morphological and mechanical performance of cyclic ori-kirigami via design-feasible parameter space, Thin-Walled Structures 207 (2025) 112706, https://doi.org/10.1016/j.tws.2024.112706.

[12] J. Mitani, A design method for axisymmetric curved origami with triangular prism protrusions, in: P. Wang-Iverson, R. J. Lang, M. Yim (Eds.), Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education, CRC Press, 2011, pp. 437–447, https://doi.org/10.1201/b10971.

[13] E. D. Demaine, K. Mundilova, T. Tachi, Locally flat and rigidly foldable discretizations of conic crease patterns with reflecting rule lines, in: L.-Y. Cheng (Ed.), ICGG 2022 - Proceedings of the 20th International Conference on Geometry and Graphics, Springer International Publishing, 2023, pp. 185–196, https://doi.org/10.1007/978-3-031-13588-0_16.

[14] E. Demaine, M. Demaine, K. Mundilova, et al., Design of circular-arc

curved creases of constant fold angle, in: Bridges 2020 Conference Proceedings, Tessellations Publishing, 2020, pp. 129–136.

[15] N. Lee, S. Close, Curved pleat folding for smooth wrapping, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 469 (2013) 20130152, https://doi.org/10.1098/rspa.2013.0152.

[16] C. C. Leong, Simulation of nonzero Gaussian curvature in origami by curved-crease couplets, in: P. Wang-Iverson, R. J. Lang, M. Yim (Eds.), Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education, CRC Press, 2011, pp. 53–67, https://doi.org/10.1201/b10971.

[17] C. Jiang, K. Mundilova, F. Rist, J. Wallner, H. Pottmann, Curve-pleated structures, ACM Trans. Graph. 38 (6), https://doi.org/10.1145/3355089.3356540 (Nov. 2019).

[18] H. Liu, R. D. James, Design of origami structures with curved tiles between the creases, Journal of the Mechanics and Physics of Solids 185 (2024) 105559, https://doi.org/10.1016/j.jmps.2024.105559.

[19] S. R. Woodruff, E. T. Filipov, Bending and twisting with a pinch: Shape morphing of creased sheets, Extreme Mechanics Letters 52 (2022) 101656, https://doi.org/10.1016/j.eml.2022.101656.

[20] S. R. Woodruff, E. T. Filipov, A bar and hinge model formulation for structural analysis of curved-crease origami, International Journal of Solids and Structures 204-205 (2020) 114–127, https://doi.org/10.1016/j.ijsolstr.2020.08.010.

[21] G. G. Slabaugh, Computing Euler angles from a rotation matrix, http://eecs.qmul.ac.uk/~gslabaugh/publications/euler.pdf (1999).

[22] J. C. Badger, T. G. Nelson, R. J. Lang, D. M. Halverson, L. L. Howell, Normalized coordinate equations and an energy method for predicting natural curved-fold configurations, Journal of Applied Mechanics 86 (7), https://doi.org/10.1115/1.4043285 (Jul. 2019).

[23] S. N. Sivanandam, S. N. Deepa, Introduction to genetic algorithms, Springer, 2008, https://doi.org/10.1007/978-3-540-73190-0.

[24] R. K. Arora, Optimization: Algorithms and applications, CRC Press, New York, 2015, https://doi.org/10.1201/b18469.

[25] A. Jha, I. Dayyani, Shape optimisation and buckling analysis of large strain zero Poisson's ratio fish-cells metamaterial for morphing structures, Composite Structures 268 (2021) 113995, https://doi.org/10.1016/j.compstruct.2021.113995.

[26] D. K. Pokkalla, L. H. Poh, S. T. Quek, Isogeometric shape optimization of missing rib auxetics with prescribed negative Poisson's ratio over large strains using genetic algorithm, International Journal of Mechanical Sciences 193 (2021) 106169, https://doi.org/10.1016/j.ijmecsci.2020.106169.

[27] G. Cimolai, I. Dayyani, Q. Qin, Multi-objective shape optimization of large strain 3D helical structures for mechanical metamaterials, Materials & Design 215 (2022) 110444, https://doi.org/10.1016/j.matdes.2022.110444.

[28] MathWorks, Find minimum of function using genetic algorithm - MATLAB ga, https://www.mathworks.com/help/gads/ga.html (2025).