Folding Functions: Origami Corrugations from Equations

Uyen Nguyen[1], Ben Fritzson[2], and Marcus Michelen[3]

[1]New York, New York, USA; win@winwin.fashion
[2]Philadelphia, Pennsylvania, USA; ben.fritzson@gmail.com
[3]Department of Mathematics, University of Pennsylvania, USA; marcusmi@sas.upenn.com

Abstract

We present an algorithm for designing flat-foldable origami corrugations that approximate functions and other continuous curves. We discuss the types of curves for which our algorithm works and where it breaks down, giving reasons why and suggesting changes that can be made to rectify problematic functions. We also give examples of modifications that can be made to the algorithm for aesthetic variation.

Introduction

In origami, a corrugation is a style of design that showcases the entire surface of the paper such that every fold is visible[5, 1]. Origamist Ray Schamp coined the term corrugation to differentiate the style from tessellations [4]; Schamp's works also served as inspiration in developing the algorithm for this paper. We describe a method of creating a flat-foldable corrugation that models a raised approximation of a curve and compresses said curve to a single point when fully flat-folded. Figure 1 shows an example corrugation of piecewise semicircles in flat and expanded configurations. If the paper had zero thickness, the flat configuration would exist in one plane. However, since it has nonzero thickness, the compressed curve is represented as a line rather than a point. All models shown were folded from Stardream paper, which has a thickness of approximately 0.16mm.

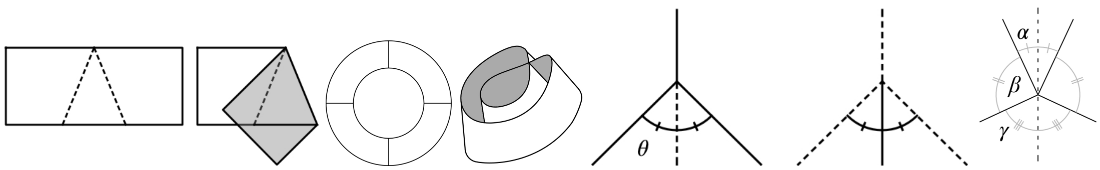
Figure 1: *A flat-foldable origami corrugation.*

Flat-Foldability

A model can be deemed flat-foldable if all the uncreased parts of the paper lay in a set of parallel planes. This definition allows for the thickness of the paper (or other material) and for the minute curvature involved in even the flattest of creases. For an origami design to be flat-foldable, it must satisfy three criteria: Maekawa's theorem, Kawasaki's theorem, and Justin's non-crossing conditions [3, 6].

Maekawa's and Kawasaki's theorems need only apply to intersections within the boundary of the sheet and not to those on the perimeter. Maekawa's theorem states that in order for a given vertex to be flat-foldable, the difference between the number of mountain folds and valley folds must be two. We denote mountain and valley folds by solid and dashed lines, respectively. Kawasaki's theorem states that in a flat-foldable vertex, if we number the angles between adjacent folds and subtract the sum of the even-numbered angles from the sum of the odd-numbered angles, the result will be zero.

Justin's non-crossing conditions describe valid and invalid stacking orders of the layers of paper for a given origami design. Invalid stacking orders would involve self-intersection, where part of the sheet passes through itself. This must be avoided in order for a given model to be possible, let alone flat-foldable. Figure 2a shows an example of a folding pattern that has no valid stacking order. Once the first crease is made, it is impossible to fold the second crease without self-intersection unless a third crease is added. Unlike Maekawa's and Kawasaki's theorems, which are easily checked local conditions, Justin's non-crossing conditions imposes both local and global restrictions. Figure 2b shows an example of a ring with four mountain folds. In order for it to fold flat, the stacking order of its facets would have to include a crossing (Figure 2c), thereby violating Justin's non-crossing conditions.



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

Figure 2: *Satisfying Justin's non-crossing conditions.*

Our corrugations are comprised of four-fold and six-fold vertices. Every four-fold vertex has one of the two crease assignments shown in Figure 2d, and every six-fold vertex has the crease assignment shown in Figure 2e. These crease assignments satisfy Maekawa's theorem, and since they are mirror-symmetric, they satisfy Kawasaki's theorem. To locally satisfy Justin's non-crossing conditions, θ cannot exceed 90° and neither α nor γ may exceed β.

Designing the Corrugation

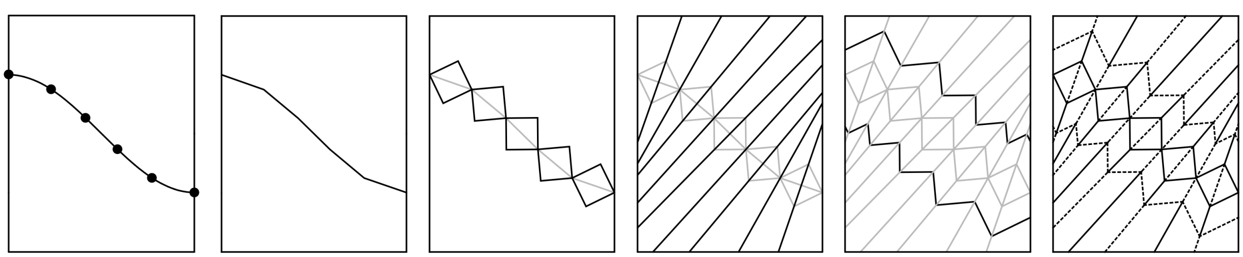

(a) **(b)** **(c)** **(d)** **(e)** **(f)**

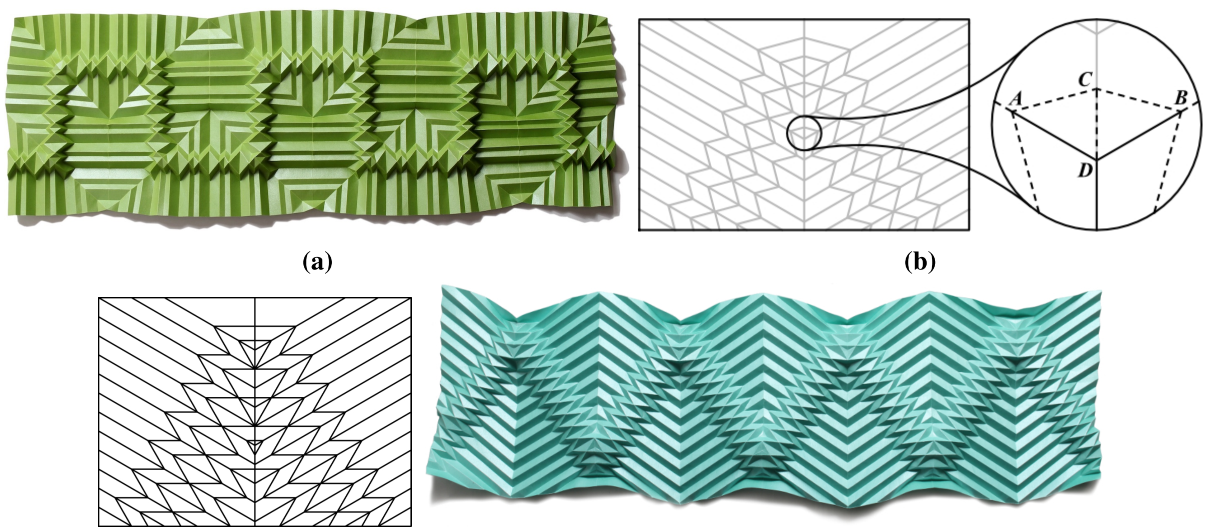
Figure 3: *Structures of the crease pattern.*

To generate a crease pattern, take a curve and divide the length into segments whose linear distances are equal (Figure 3a). If the curve has mirror symmetry or periodicity, the endpoints of the segments should reflect that. Draw a line connecting the endpoints of each segment to form a piecewise linear approximation of the original curve (Figure 3b). Construct a series of rhombi using each of the new lines as a diagonal of a rhombus (Figure 3c). For most examples, we use a square; other angles can work, but all rhombi on a curve must use the same angle in order to satisfy Kawasaki's theorem. Construct angle bisectors between each pair of line segments and perpendicular bisectors of each segment (Figure 3d). If the original curve has any lines of symmetry, extend the bisectors that follow those lines away from the curve to the edges of the sheet. Then extend all bisector lines until each reaches either the edge of the sheet or a line of symmetry. At this point, the line segments approximating the original curve can be ignored. Around each rhombus, draw a similar rhombus that has been enlarged (in most of our examples we do so by a factor of two) and trim each to limit it to the space within each of the neighboring angle bisector lines (Figure 3e); hereafter, this collection of lines surrounding the original rhombi are referred to as the ripple. When folding the result, the starting rhombi are mountain folded and the ripple is valley folded (Figure 3f). The remaining bisectors are folded such that they satisfy the crease assignments of Figure 2d and 2e.

As the paper is folded, the bisectors allow the paper to collapse much like an accordion or a bellows, while the rhombi's shared corners are elevated to illustrate an approximation of the original curve. The approximation improves as the number of rhombi used increases.

Piecewise Linear Functions and Fractals

At each of the bends in a piecewise linear curve, the bisector between two neighboring segments is a line of mirror symmetry and the rhombi, bisectors, and ripple described in Figure 3 terminate when they reach that line of mirror symmetry. This line alternates between mountain and valley folds as it passes through each intersection such that it satisfies Figure 2d and 2e. We can apply this method to various segmented shapes, such as square waves (Figure 4a) and triangle waves, but depending on the angle between segments, it can be necessary to adjust the proportions of the rhombi involved.



(a) **(b)**

(c) **(d)**

Figure 4: *Square wave and triangle wave.*

For example, trying to construct square rhombi along segments of a triangle wave with 60° angles will result in a model that is not flat-foldable (Figure 4b). If we examine the enlarged region, vertices A and B are part of the ripple and have set crease assignments. For vertex C to be flat-foldable, $\overline{CD}$ must be a mountain fold. For vertex D to be flat-foldable, $\overline{CD}$ must be a valley fold. As there is no way to satisfy Maekawa's theorem without self-intersection, the design cannot be flat-folded. If we instead use 60°-120° rhombi (Figure 4c and 4d), points A, B, and C will be collinear and the assignment of line $\overline{CD}$ can be in agreement with that of vertex

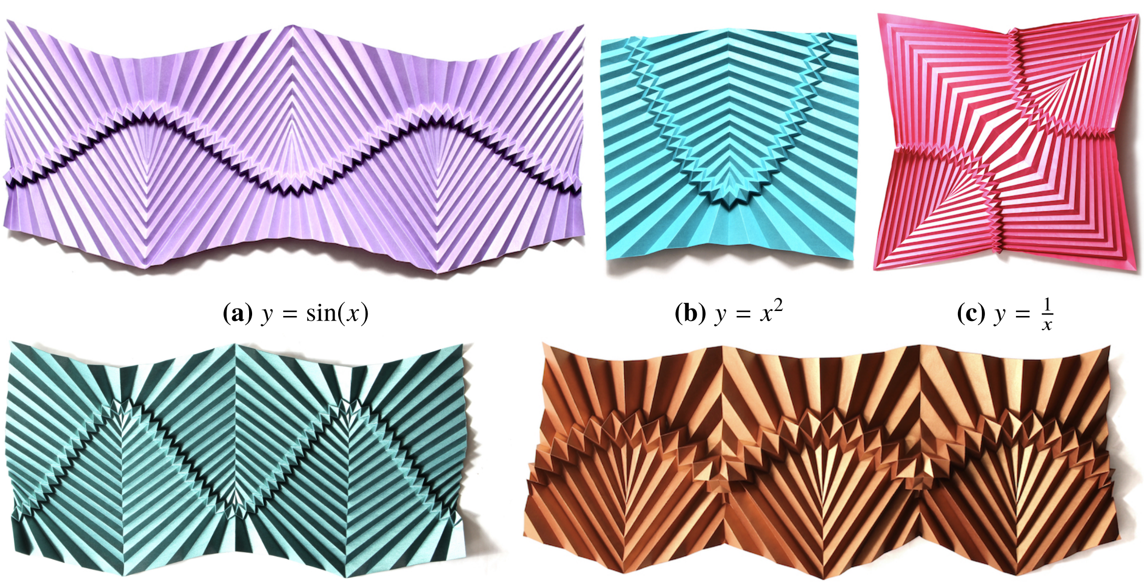
D. Using thinner rhombi will place point C below points A and B, in which case the crease assignments for vertices C and D will also be in agreement.

Our method has allowed for shapes as complex as a second order Hilbert curve to be folded (Figure 5a); however, due to nonzero paper thickness it cannot fold all the way flat (Figure 5b) - there are too many layers for the last fold to fully compress the Hilbert curve. We designed and simulated a third order Hilbert curve using Amanda Ghassaei's Origami Simulator [2] (Figure 5c), but we have not attempted to fold it. Note that in the square wave and Hilbert curves, the size of the rhombi is not uniform. This is an aesthetic variation discussed later in the paper.

(a) **(b)** **(c)**

Figure 5: *Second and third order Hilbert curves.*

Automating the Algorithm



(a) $y = \sin(x)$ **(b)** $y = x^2$ **(c)** $y = \frac{1}{x}$

(d) $y = 2\sin(x)$ **(e)** $y = |\sin(x)|$

Figure 6: *Equations with changing curvature modeled with our algorithm.*

For equations of constant curvature (lines and circles), constructing crease patterns is relatively easy without automation, as symmetry (rotational, translational, and mirror) can be exploited to expedite the process. In the case of equations whose curvature is not constant, it can be very difficult and time consuming to apply our algorithm using CAD software alone, so we wrote a Python program to automate the process. Some examples of curves folded using the program are shown in Figure 6.

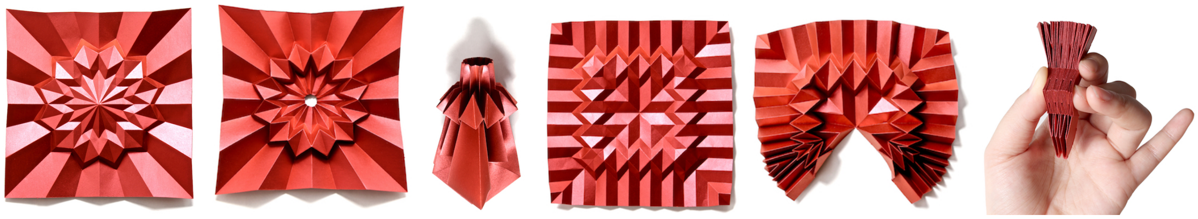
The program implements a bisection root-finding algorithm to lay out points that are an even linear length apart and connects adjacent pairs linearly. The perpendicular bisectors of each segment are added, and congruent rhombi are added. The remaining bisectors are computed by bisecting the angles between the rhombi at the vertices that lie on the curve. The ripple is subsequently drawn by offsetting the peaks of each rhombi in the direction of the corresponding bisectors and drawing line segments parallel to the sides of the rhombi.

The inputs to the program include the function to be modeled, start and end x-values, number of rhombi, spike factor (the ratio of diagonal lengths of the rhombi), and ripple factor (related to the distance between the rhombi and the ripple). The program outputs a .dxf file that can then be edited using AutoCAD or similar software. When using the program, if the input equation is symmetric or periodic, only the smallest symmetric interval should be used, as the program is not equipped to deal with intersecting bisectors. We can complete a larger portion of the curve in AutoCAD by mirroring the crease pattern until the curve is complete or the desired number of periods is reached. The program also does not properly deal with endpoints on the curve unless they lie on a normal parallel to the y-axis. If the function doesn't have an interval that starts and ends on vertical normals, it may be desirable to input a slightly larger interval than needed and trim the endpoints in AutoCAD. The program is available for download in our supplementary materials.

Limitations

Having the automated program greatly expedited the process of creating crease patterns, which enabled us to explore more types of curves and, in turn, let us discover new limits of our algorithm where we encountered problem areas. There are multiple ways in which the algorithm's results can fail to be flat-foldable.

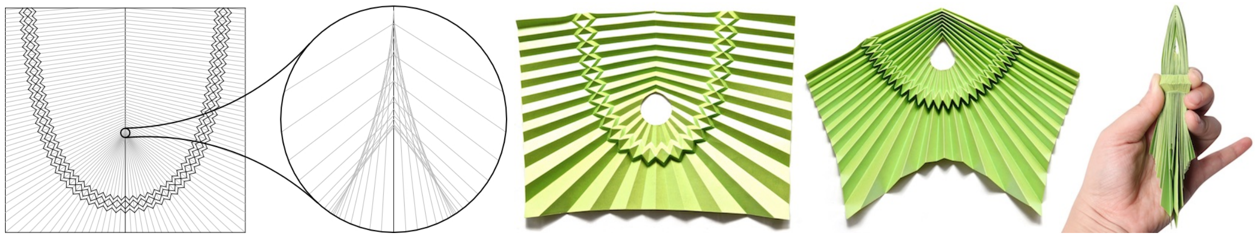
We folded $y = \frac{1}{x}$ as an example of a discontinuous curve but encountered a Maekawa's theorem violation at the central vertex (Figure 6c). All curves that have intersecting lines of mirror symmetry will have this problem, as the vertex created where they meet must have creases that either are of the same orientation or alternate between mountain and valley in order to remain mirror symmetric across all axes, thereby violating Maekawa's theorem. Folding $y = |\frac{1}{x}|$ would not have this problem.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 7: *Closed loops.*

Our algorithm breaks down when modeling a closed loop as well; the central vertex has an equal number of mountain and valley folds and thus cannot fold flat (Figure 7a and 7d). If we cut a hole in the paper such that the problematic vertex is no longer part of the sheet (Figure 7b) it still cannot fold flat and instead folds onto a cylinder (Figure 7c). However, if we cut a slit from the center of the failing intersection to the edge of the paper (Figure 7e), the design can fold flat (Figure 7f). The central vertex, now on an edge, is no longer fully subject to flat-folding rules and the model is now able to collapse freely. This is because cutting a hole does not allow for alternative layer-stacking orders, but cutting a slit does.

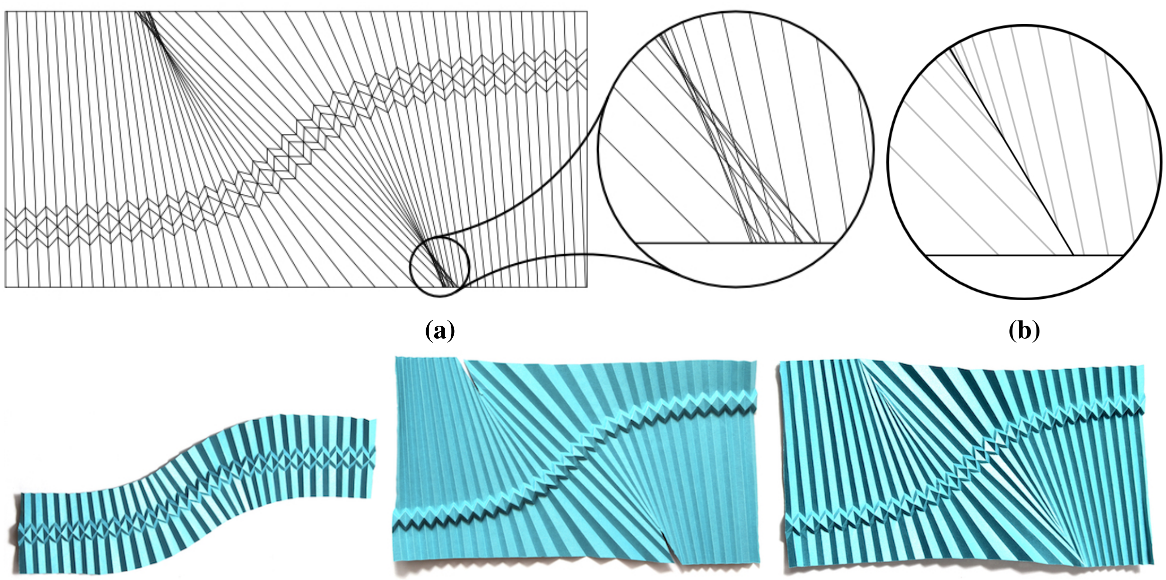


(a) (b)

Figure 8: *Folding modifications for a secant curve.*

We found a different type of problem when testing half a period of secant, between -90° and 90°. While it has broadly the same structure as a parabola (Figure 6b), it fails in a new way. The bisectors emanating from the curve do align along the axis of symmetry in tidy pairs that would satisfy flat-foldability rules, but they first converge with one another (Figure 8a). The resulting intersections do not satisfy any flat-folding conditions, meaning that secant cannot be flat-folded without modification. However, unlike closed loops, with secant it is possible to simply excise the small section of the pattern with the conflicts and render it flat-foldable (Figure 8b).

While a closed loop does not meet Justin's non-crossing conditions on a global level, the problematic intersections for the secant curve only affect the immediate surrounding area. Thus cutting a hole will not resolve a global problem, but it may resolve local violations.



(a) (b)

(c) (d) (e)

Figure 9: *Folding modifications for the arctangent function.*

A similar problem appears when we try to apply the process to arctangent. As it lacks a line of symmetry for stray bisectors to meet at, each bisector converges with a neighboring one on the concave sides, resulting in a region in each concavity that is dense with intersections that don't satisfy flat-folding conditions (Figure 9a). Unlike with secant, these regions extend infinitely away from the curve.

However, it is possible to fold arctangent with some design modifications. A curved crop can be applied to the paper to exclude the problematic regions regions (Figure 9c), but it can be challenging to preserve boundaries that respect the edges of the curve's range and domain. Alternatively, a slit can be cut on each side of the curve (Figure 9b and 9d), from the start of the irregular intersections to the edge of the paper thus preventing the subsequent intersections from needing to be resolved; each bisector (grey) would terminate on the slit (black) rather than intersecting with another bisector. Lastly, it is possible to preserve the rectangular perimeter for aesthetic purposes, but the result is not flat-foldable (Figure 9e); by subtly changing the angles involved, the lines, which are no longer bisectors, can be made to meet in pairs along a false line of symmetry. Doing so satisfies Maekawa's theorem, but violates Kawasaki's theorem.

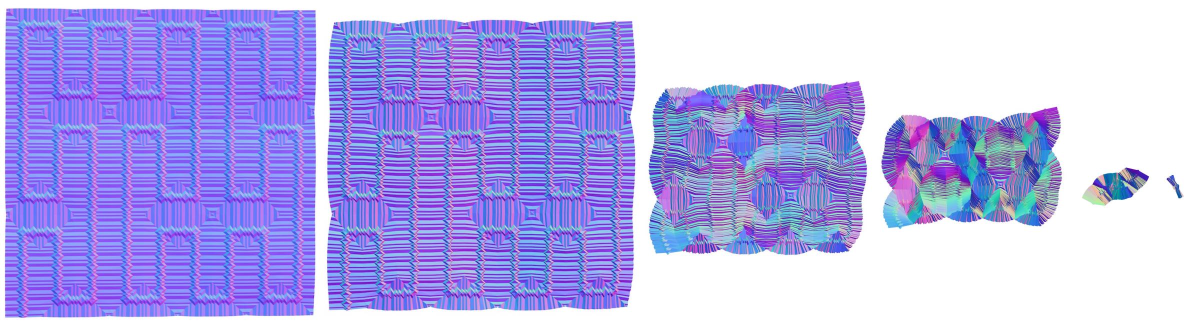
Figure 10: *Algorithmic variations along a sine wave.*

Aesthetic Variations and Conclusion

The algorithm we describe is only one of many ways to fold origami corrugations of various allowable equations. Even small changes to our approach can produce diverse and sometimes striking results. While most of the figures we've shown assume square rhombi of uniform size, by changing the algorithm we can alter the appearance of the final paper sculpture, while still holding true to the design concept (Figure 10).

For example, we can represent the curve using only ripples and no rhombi (Figure 10a), increase or decrease the distance between the rhombi and the ripple and thus changing the curve's elevation off the rest of the sheet (Figure 10b and 10f), have the angles of the ripple differ from those of the rhombi causing sections of the sheet to form a tent (Figure 10c), use a wider or taller aspect ratio for the rhombi (Figure 10d and 10e), use multiple sizes of rhombi so long as the angles in them are preserved (Figure 10f and 10g), and create multiple rows of rhombi using reflected ripples (Figure 10h). Additionally, kites can be used in place of rhombi provided the vertices along the curve maintain mirror symmetry (Figure 1). Variations beyond this spectrum of examples are possible; however, while subtle changes should work, extreme changes can make fully flat-folding a design difficult or even impossible.

As a parting note, we end with Figure 11, which depicts Origami Simulator's [2] output of a design for a second order Peano curve.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 11: *Folding progression to a flat state of the Peano curve.*

Acknowledgements

We thank Jonathan Schneider, Richard Fritzson, Ninh Tran, Doug Caine, Mike Reca, and Tom Hull for useful discussions.

References

[1] "Geometrics." *Origami Universe*, Chimei Museum Foundation, 2017, pp. 214.

[2] A. Ghassaei. "Origami Simulator." http://apps.amandaghassaei.com/OrigamiSimulator/.

[3] R. Lang. *Twists, Tilings, and Tessellations: Mathematical Methods for Geometric Origami*. CRC Press, 2018, pp. 20–29.

[4] B. Peyton and U. Nguyen. "Origami Universe: An Exhibition of Art and Innovation Through Folding." *Origami Universe*, Chimei Museum Foundation, 2017, pp. 31.

[5] "Pleats and Corrugations." *Surface to Structure: Folded Forms*, 2014, pp. 141.

[6] J. Schneider. "Flat-Foldability of Origami Crease Patterns." 2005. https://www.sccs.swarthmore.edu/users/05/jschnei3/brinkmann.pdf.