

November 2023

When to Hold and When to Fold: Studies on the topology of origami and linkages

Mary Elizabeth Lee
University of Massachusetts Amherst

Follow this and additional works at: https://scholarworks.umass.edu/dissertations_2



Part of the [Geometry and Topology Commons](#), and the [Statistical, Nonlinear, and Soft Matter Physics Commons](#)

Recommended Citation

Lee, Mary Elizabeth, "When to Hold and When to Fold: Studies on the topology of origami and linkages" (2023). *Doctoral Dissertations*. 3011.
<https://doi.org/10.7275/35895143> https://scholarworks.umass.edu/dissertations_2/3011

This Open Access Dissertation is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

**WHEN TO HOLD AND WHEN TO FOLD:
STUDIES ON THE TOPOLOGY OF
ORIGAMI AND LINKAGES**

A Dissertation Presented

by

M. E. V. LEE-TRIMBLE

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2023

Department of Physics

© Copyright by M. E. V. Lee-Trimble 2023

All Rights Reserved

**WHEN TO HOLD AND WHEN TO FOLD:
STUDIES ON THE TOPOLOGY OF
ORIGAMI AND LINKAGES**

A Dissertation Presented

by

M. E. V. LEE-TRIMBLE

Approved as to style and content by:

Christian D. Santangelo, Chair

Gregory M. Grason, Member

Benjamin Davidovitch, Member

Narayanan Menon, Member

Tony D. Dinsmore, Chair of the Faculty
Department of Physics

DEDICATION

To the artist in every scientist and the scientist in every artist.

ACKNOWLEDGMENTS

No science is truly done in a vacuum, and this dissertation certainly was not. I owe thanks to so many people, in and out of science, and I will do my best to acknowledge everyone here. Some names will appear multiple times as I go through the groups; if that's you then you've certainly earned the extra thanks.

I owe a special thanks, of course, to Chris, for mentorship, guidance, and for making me the scientist I am today. I consider myself very lucky to have had you for an advisor. Thanks as well to my committee members, Benny Davidovitch, Greg Grason, and Narayanan Menon, as well as Shubha Tewari for keeping an eye out for me at UMass after Chris moved to Syracuse.

A thanks as large as this document to everyone who provided comments on it. In particular, to Nat DeNigris and Casey Lee-Trimble for comments on the introduction; to Daria Atkinson, Manu Mannatil, and Itay Griniasty for comments on the manuscripts that became the papers that became Chapter 1; to Nick Hackney for comments on Chapter 2; and Manu Mannatil for comments on the manuscript of the paper that became Chapter 3. A special thanks as well to Michelle Berry, with whom two of these papers were written. This work was supported by the UMass Spaulding-Smith Fellowship, the National Science Foundation for support for this work through grant DMR-1822638 and the Graduate Research Fellowship under Grant No. 451512, and the Army Research Office through grants W911NF-19-1-0348 and W911NF-21-1-0068.

To group members, past and present, for good discussions and group meetings, in order of getting to know you: Salem Al-Mosleh, Bryan Chen, Carlos Duque,

Daria Atkinson, Michelle Berry, Manu Mannatil, Sourav Roy, Kyungeun Kim, Samay Narasimhamurthy, Gentian Muhaxheri, Alexia Chatzitheodorou, and Chance Bagget. I hope someday I'll actually get to meet the last five of you in person. This dissertation happened during the absolutely wild ride that was 2020 and the subsequent years, and so I owe an additional thanks to the community and will to continue the work provided by my home and virtual office-mates: Nat DeNigris, Casey Lee-Trimble, Daria Atkinson, Nick Hackney, and everyone else in the remote work accountability Discord.

To friends, without whom life would be that much more dreary: Special thanks to Emily Pruc, Daria Atkinson, and Nat DeNigris for the weekly Zoom calls and shenanigans, I love you all and appreciate your friendship. Also no, I did not rewrite this section multiple times and delete everything too sentimental, you're lying. To Thomas Dellaert for the visits and longest continuing friendship and the ability to pick up the phone and continue right where we left off. To the Honk family Discord. Honk.

To family: I'm struggling to put into words how much your support means. For phone calls and putting up with my not-as-frequent-as-we-all-would've-liked visits, thank you and love to you all. Mom, this is your birthday present this year. Sorry.

Finally to Casey, I definitely could not have done this without you. I love you bunches and I can't wait for our next adventure.

ABSTRACT

WHEN TO HOLD AND WHEN TO FOLD: STUDIES ON THE TOPOLOGY OF ORIGAMI AND LINKAGES

SEPTEMBER 2023

M. E. V. LEE-TRIMBLE

B.Sc., GEORGIA INSTITUTE OF TECHNOLOGY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Christian D. Santangelo

Linkages and mechanisms are pervasive in physics and engineering as models for a variety of structures and systems, from jamming to biomechanics. With the increase in physical realizations of discrete shape-changing materials, such as metamaterials, programmable materials, and self-actuating structures, an increased understanding of mechanisms and how they can be designed is crucial. At a basic level, linkages or mechanisms can be understood to be rigid bars connected at pivots around which they can rotate freely. We will have a particular focus on origami-like materials, an extension to linkages with the added constraint of faces. Self-actuated versions typically start flat and when exposed to an external stimulus - such as a temperature change or magnetic field - spontaneously fold. Since these structures fold all at once, and the number of folding patterns accessible to a given origami are exponential, they are prone to folding to a configuration other than the desired one. Other work has suggested methods for avoiding this misfolding, but it assumes ideal, rigid origami.

Here, we expand on these models to account for the elasticity of real structures and introduce methods for accounting for Gaussian curvature in them. We also explore how to find and set an upper bound on minimal forcing sets, or the minimum set of folds required to force an origami, and present a graph theory algorithm for finding them in arbitrary origami. Taken altogether, these origami studies give insight into how the physical properties of origami influence folding and a new set of tools for avoiding misfolding. Next, we turn back to a more fundamental study of linkages and present a new method for finding the manifold of their critical points. We then demonstrate a design protocol that utilizes this manifold to create linkages with tunable motions, before turning to several example structures, including the four-bar linkage and the Kane-Lubensky chain.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	v
ABSTRACT	vii
LIST OF TABLES	xii
LIST OF FIGURES.....	xiii
CHAPTER	
INTRODUCTION	1
1. NON-IDEAL ORIGAMI	5
1.1 Non-Euclidean Origami ¹	5
1.1.1 Mathematical Formulation	7
1.1.2 Single vertices	15
1.1.3 Discussion	21
1.2 Elastic Origami ²	22
1.2.1 Folding rigid origami	24
1.2.2 Elastic origami	25
1.2.2.1 Estimate of the stretching energy of elastic origami	28
1.2.2.2 The bending modulus of the folds	30
1.2.2.3 Stiffness ratios	31
1.2.3 The origami “bird’s foot”	32

¹This section is adapted from Ref. [1] with permission from the American Physical Society.

²This section is adapted from Ref. [2] with permission from the Royal Society of Chemistry

1.2.3.1	Phase Diagrams	36
1.2.3.2	Experimental Methods	39
1.2.4	Folding complex origami	44
1.2.5	Discussion	48
2.	OTHER ORIGAMI PROBLEMS	50
2.1	Face-Forced Origami	50
2.1.1	Connectivity Matrix	51
2.1.2	Disallowing Bending in the Vertex Forces	54
2.1.3	Predicting and Designing Face-Forced Origami	56
2.1.4	Discussion	59
2.2	Maximal Minimal Forcing Sets	59
2.2.1	Forcing an arbitrary single vertex	60
2.2.2	Finding a 2-Subgraph	62
2.2.3	Spin System Version	66
2.2.4	Simulations	68
2.2.5	Discussion	73
3.	DESIGNING LINKAGES	74
3.1	Critical Points in Mechanisms	75
3.1.1	Mechanism Rigidity	75
3.1.2	Shape of the configuration space at critical points	77
3.1.3	Shape of the configuration space near critical points	81
3.2	Controlling configuration space topology	85
3.2.1	Properties of the tangent form	86
3.2.2	The geometry of the critical configuration set	90
3.2.3	The geometry of the critical value set	94
3.2.4	Three rotor system	95
3.3	Gating the Kane-Lubensky Chain	98
3.4	Discussion	104
4.	CONCLUSIONS AND OUTLOOK	106
	APPENDIX: MATHEMATICA CODE AND	
	DOCUMENTATION	110

BIBLIOGRAPHY	118
--------------------	-----

LIST OF TABLES

Table		Page
2.1	Simulation data for a few random origami with both only the minimal forcing set programmed with target angles and with all folds programmed. The average fold difference is calculated using Eq. 2.25 and represents the distance from the target angles in <i>radians</i> ² . Each line consists of around 500 iterations for each set of target angles.	71

LIST OF FIGURES

Figure	Page
1.1 (a) A generic non-Euclidean origami structure. The vertex Gaussian curvature is defined by $K = 2\pi - \sum_i \alpha_i$ (b,c) Degree four vertices with positive and negative Gaussian curvatures respectively necessarily buckle out of the plane.	6
1.2 A nearly flat origami structure can be projected to a fold pattern in the xy -plane. In-plane and out-of-plane displacements are unambiguously decomposable.	8
1.3 The intersection of a sphere with a vertex at its center is a spherical polygon, which we decompose into triangular slices as shown. (a) The dihedral angle of the i^{th} fold is $\theta_i^+ + \theta_i^-$. (b) The side lengths are the planar angles $\alpha_{i,i+1}$ and the angle the folds make with respect to the xy -plane, ψ_i	13
1.4 Notation for the vicinity of a single vertex when there are multiple vertices.	14
1.5 The configuration space of a symmetric five-fold vertex (a) near the flat state with zero (b), positive (c), and negative (d) Gaussian curvature projected onto the fold angles $(\theta_{02}, \theta_{03}, \theta_{04})$. The fraction of red and blue, $[r, b]$, in the coloring is determined by $[(\theta_{01} - \pi)/(2\pi), (\theta_{05} + \pi)/(2\pi)]$	16
1.6 (left) a cross-section of a trilayer origami face showing the thicknesses and three-dimensional Young's moduli. (right) the area of each face adjacent to an edge that is closer to that edge than any other.	30
1.7 Bending a face or a fold to have constant curvature R^{-1} . The angle θ is identical to the apparent fold angle of the fold. When bending a face (left), we assume the face bends along the midsurface whereas for an active fold, we assume the surface bends along the stiffest layer.	31

1.8	Schema, configuration spaces, and energy landscapes for the birdsfoot origami. (A) A schematic of the birdsfoot. The folds are marked in solid lines, while the folds added in the model to imitate face bending are marked with dashed lines. The folds used to define fold space are highlighted in red. The face we have “pinned” to a plane is highlighted in yellow, then the heights of vertices 4 and 6 above this plane defines the height space. (B) The configurations that correspond to each branch. (C-D) The branches for a rigid origami in both fold and height space, respectively. In (D), the heights are non-dimensionalized using the characteristic fold length ℓ . The dashed lines show the linearized trajectories between the branches at two magnitudes. The height space projection of the trajectory takes advantage of the linearity between height and fold space at small heights. In (C), note that the branches in fold space are perpendicular. (D) also shows the shape of branch 2 in the approximate energy used to draw the energy landscapes in Fig. 2.	33
1.9	Energy landscapes in height space for the bird’s foot at different values of the control parameter A , for the small magnitude trajectory between branches shown in Fig. 1.8(C-D), at $K_{face} = 10^{-2}$ and $K_{fold} = 10^{-4}$. The white dots represent local minima and the red dots represent saddle points. The dashed line represents the projection of the trajectory between branches into height space (as in Fig. 1.8(D)), with the circle denoting the location on the trajectory of the landscape. Notice that as A increases, the original minimum moves toward the flat state. Between $A = 0$ and $A = 0.37$, a saddle point and the minimum for branch 2 are created, then after $A = 0.56$ the saddle point and the minimum for branch 1 annihilate each other. The created minimum also moves out away from the flat state as A increases. Note that the contours and color scheme are on a log scale and inconsistent between landscapes to emphasize features.	35
1.10	Slices of a four-dimensional phase diagram defined by K_{fold} , K_{face} , control parameter A , and target fold angle magnitude M at four different values of K_{face} and K_{fold} . The regions in light purple represent the region of bi-stability, where both minima are present. The purple and tan represent the regions where only the branch 1 minimum and branch 2 minimum are present, respectively.	37

1.11	Phase diagrams for the birdsfoot with axes of the magnitude of the target angles, M , and the control parameter A , as described in the main body of the paper. The blue and red regions represent only one branch appearing while the center pink region represents the bistable region. The left column uses the linear spring edge model for face stretching while the right column uses the elastic polygon model. The top row has torsion spring constants corresponding to experiment with default faces while the bottom row corresponds to softened faces.	40
1.12	Experimental schema and results for folding the birdsfoot with and without weakened faces. (A) Schematic of the tri-layer origami structure with perforated faces. (B) The percentage of samples folded to the second branch for both non-perforated and perforated faces. Each point corresponds to 10 samples. Error bars are from the rule of three. (C) The folded samples with perforated faces at control parameters (i) $A = 0.33$ and (ii) $A = 0.67$. At $A = 0.33$ (i), you can see the two samples that folded to branch 1 while the rest shallowly folded to branch 2.	43
1.13	(left) the Randlett bird with true folds in black and added face folds in lighter blue with vertices numbered. (right) the programmed fold angles used in the simulations. Folds are denoted by their end vertices.	44
1.14	(A) An optical image of the experiment before folding. (B) Some examples of folded and mis-folded structures. The experiment folds correctly at a rate of 0.45 ± 0.15	45
1.15	(A) A density plot of the number of different configurations seen from near the flat state for given K_{fold}/K_S and K_{face}/K_S . Each point represents a minimum of either 500 simulated folds or 10 times the number of different states observed, whichever was larger. The number of configurations reflects whether the origami is mono- or multistable, but may not predict the precise number of possible fold configurations for the given parameters. Note that the color scale is a log scale to emphasize features at lower configuration numbers. The gray region represents the region in which the pre-folded, initial numerical minimization fails. (B) A density plot showing the percentage of the simulated birds that fold into the target state. This does not represent a prediction of experiment, rather that the basin of attraction for the correct state is larger when the degree of multistability is less.	46

2.1	A sample coordinate system for relating the torques to the forces for a triangular face. If the face is in the plane of the page, the forces are perpendicular to the page.	52
2.2	Two different versions of the birdsfoot origami, one with parallelogram faces and one with triangulated faces, with vertices labeled in black and faces labeled in blue.	53
2.3	An example bipartite expansion of a vertex with degree 4 as part of the algorithm for finding a 2-subgraph on an origami in order to find a minimal forcing set. The red dashed lines demonstrate a matching that would include the $d - 2$ vertex set as end points, and the red dashed vertices show the two vertices that would be free to be included in a maximal matching over the whole graph.	64
2.4	The method for finding the maximal minimal forcing set using Tutte's 2-subgraph algorithm using a simple two vertex origami as an example. (a) shows the original origami while (b) is its corresponding bipartite expansion. The red highlighted edges in (c) show the maximal matching on the full graph while in (d) they represent the chosen unassigned folds. Correspondingly, the black internal folds in (d) are the maximal minimal forcing set.	65
2.5	Comparisons of different torsional spring stiffness constants for the assigned and unassigned folds for the folding simulations. Each cluster of points represents 100 folded origami.	69
2.6	(Left) The randlett bird with the unassigned folds marked in red, assigned folds in black, and face folds in blue. (Right) The average difference between the final fold angles of the simulation and the programmed target angles. Blue represents the minimally programmed version while orange represents the fully programmed version.	72
3.1	Schematic of how a configuration space with a branch point split into one of two types of smooth, disconnected configuration spaces. The choice of sign is arbitrary.	80

3.2	(a) Schematic of the planar, four-bar linkage with variables defined. (b) Projection of the configuration space of the two rotor mechanism with $L_1 = L_2 = L_3 = a$ projected into (θ_1, θ_2) plane (black). This choice of lengths has three branch-like critical points. Deforming the length of L_2 results in a smooth configuration space with either one (red) or two (blue) components. The arrows indicate the direction of the tangent form $t_i(\mathbf{u})$ from Eq. (3.36).	85
3.3	(a) The critical value set for the four bar linkage, plotted in terms of (L_1, L_2, L_3) in units of a . The points c-e show the locations of the configuration spaces in (c-e). (b) shows a different view of the surface with a cutout on the $L_1 = 0$ plane showing the shape of one of the enclosed volumes. There is one critical point in the configuration space along any smooth portion of the set (e). Self intersecting lines indicate choices with two critical points (d) and the triply self intersection point at $(L_1, L_2, L_3) = (a, a, a)$ (c) is the unique choice with three critical points. (c-e) The configuration spaces corresponding to the points in (a), showing how to split different branch points. (f) shows the volumes labeled with their branch point types, corresponding to the standard engineering classifications.	93
3.4	(a) Schematic of the planar, three rotor linkage with variables defined. (b) A cross-section of the critical value set for $r_1 = r_2 = r_3 = a$. (c) The 3D configuration space of the three rotor linkage with $r_1 = r_2 = r_3 = L_{13} = L_{23} = a$, corresponding to the red (gray) point in (b), contains twelve individual critical points. Arrows indicate the orientation of each configuration space segment.	97
3.5	A schematic of programming the three rotor system. (a) A map showing how changing the length of r_1 , L_1 , and L_2 leads to different ways to split the branch points from Fig. 3.4. The shaded and unshaded regions correspond to the two ways for the branch points to split. The red (light gray) dot, corresponding to a change in length $r_1 = 1.05a$, leads to the red (light gray) curve in (b). In order to change the topology of the configuration space by changing how branch points 4 and 6 split, an additional change to $L_1 = 0.9a$ can be effected, shown by the blue (dark gray) dot. The new configuration space is shown in (b) as a blue (dark gray) curve.	99

3.6	(a) A Kane-Lubensky chain with a gating mechanism attached. The gating state is controlled by the length D ; the red (light gray) bar and the blue (dark gray) bar correspond to an ungated and gated state, respectively. (b) A cross-section of the critical set with $r = \ell = 3a/2$ and $L_1 = 2a$. There is a critical point at $L_2 = 2a$ and $D = 5a/2$. (c) At the critical point, changing the position of the third rotor or the lengths of two beams can switch the type of branch split (gray and transparent regions). The red (light gray) point corresponds to branch splitting such that the mechanism is ungated, while the blue (dark gray) point demonstrates that changing D can switch to a gated configuration. (d) The configuration space at and near the critical point as a function of the three rotor angles, and the projection of that configuration space onto the θ_1 - θ_2 plane. The gap in the blue (dark gray) projection shows that the full range of angles is not accessible in that configuration, and thus the chain is gated.	102
3.7	Top row (red): Ungated device made from LEGOs with the corresponding simulation. This device can continue rotating and return back to its initial position, as indicated by the arrow. Bottom row (blue): Gated device made from LEGOs with the corresponding simulation. This device gets stuck in the configuration shown in the last frame and is forced to reverse direction in order to continue moving.	104
A.1	Mathematica function for finding the face coordinates given a mechanism and a set of faces.	111
A.2	Mathematica function for finding the face coordinates given only a mechanism.	111
A.3	Mathematica function for finding the face-vertex connectivity matrix of Sec. 2.1.1	112
A.4	Mathematica function for finding vertical forces that result in non-tringular faces bending.	113
A.5	Part of the Mathematica function for finding the maximal minimal forcing set. This portion	114
A.6	Part of the Mathematica function for finding the maximal minimal forcing set. This portion	114
A.7	Part of the Mathematica function for finding the maximal minimal forcing set. This portion	115

A.8	Mathematica function to find possible programmable angles for an origami.	115
A.9	Mathematica function to generate origami programmed only with the maximal minimum forcing set.	116
A.10	Mathematica function to repeatedly minimize an origami structure based on the conjugate gradient method.	117
A.11	Mathematica function to find the average difference between the fold angles of an origami and a target set of folds angles.	117

INTRODUCTION

In the past few decades, we’ve seen a dramatic rise in man-made materials that can change their shape, behavior, or general physical properties when an external stimulus is applied. This class of materials gets called by a variety of names, from self-programmable materials to shape-changing smart materials. The possibilities presented by these materials are nearly endless: 3D structures that can be shipped compactly and deployed at their destinations are less expensive to ship and a holy grail for space exploration; mechanical computers that perform calculations under external conditions don’t need power sources and can be used in extreme environments; metamaterials with tunable properties can create multi-use devices; self-actuated, untethered robots that do not need an operator have applications from medicine to package delivery to defense. In addition to these applications, they also present a new perspective through which to view, probe, and understand the entwined relationship between a material’s geometry and its behavior. In this thesis, we will explore a limited class of these new materials that can be described as linkages or mechanisms, with a special focus on origami-like materials.

A linkage is a graph whose edges have a fixed length but whose vertices are otherwise freely rotating joints. This superficial simplicity belies behavior that can be surprisingly complex and rich. Indeed, in addition to having been prolific in engineering since the beginning of automation, they also serve as prototypical mechanical models for many different physical systems, including animal limbs and joints [3–5], polymer physics [6], protein allostery [7–11], DNA rigidity [12, 13] origami [14–16] and jamming [17–19]. Meanwhile, their complexity is particularly apparent in the recent rise of mechanical metamaterials [20] - materials whose mechanical response

arises from their geometry - the discovered topological-protection of soft modes in iso-static lattices [21], and the anomalous rigidity of undercoordinated spring and fiber networks [22].

To understand the difficulties of designing linkages, consider Kempe’s universality theorem, which showed that a linkage can be designed such that a given vertex traces out a portion of any rational algebraic curve [23, 24]. However, the results of following the proof’s procedure can be unwieldy for even simple curves. A planar curve of degree d requires no more than $3d + 2$ edges where d is the degree of the curve [25] (there is a different bound in three dimensions [24]), but these edges must be arranged in very specific geometries, and consequently the tolerances required in their fabrication is unclear. Moreover, if one wants to change the motion or proscribed path of an existing mechanism - desirable when designing programmable materials - there is no guarantee that the result can be achieved without changing the connectivity of the linkage itself.

We start this thesis by considering materials inspired by origami and kirigami, which here we will use to refer to materials with discrete, solid faces connected at folds or folds and holes in the kirigami case. When forces are applied to these origami-like materials, the deformations in the materials are distributed to the folds and the structures fold out of their initial plane. We can model them as we do for linkages, with the added considerations of faces defined by the edges of the graph of the origami. This specification of linkages has been proposed as a framework to engineer new materials with complex mechanical responses [26–30]. Self-actuating or self-folding origami-like structures are found throughout nature [31, 32] and have inspired a number of engineering applications from medicine [33, 34], to solar panel deployment [35], to robotics [36, 37].

Past work on modeling these origami-like structures has been largely focused on flat, rigid origami, or origami with rigid faces that do not stretch or bend and starts

from a flat sheet with no curvature (Gaussian or otherwise) [38–40]. Even for this highly constrained case, the space of configurations accessible to a rigid origami structure becomes increasingly complicated as the fold pattern itself becomes more complex, with the number of possible configurations increasing exponentially with the number of internal vertices of an origami [41, 42]. In the rigid case, the configuration space of these structures takes the form of branches, or paths along which the origami can fold without deforming the faces, connected at the flat state [41]. In order to reach a different branch, the origami would have to unfold to the flat state.

Allowing face bending, however, allows access to configurations away from these branches that would otherwise be impossible in purely rigid origami [27, 28, 43–46]. Physical origami systems have elasticity, and thus are able to access this larger range of configurations. Additionally, many physical systems have limitations in how they can be programmed [2, 47]. These features lead to multistability through the proliferation of local minima of the energy [48–50], sometimes resulting in origami that does not fold easily or repeatably into the target shape, impacting device performance. Various methods to avoid misfolding (the folding of an origami to a configuration other than the target configuration) have been introduced, including biasing the vertices [51] or fine tuning individual fold stiffnesses [39, 52], but the actual mechanisms behind misfolding are still not well understood.

In this thesis, we will explore these topics starting with the specific case of origami before expanding outward to address linkage design problems. In Chapter 1, we will study non-ideal origami, or origami that is not rigid or flat or both, focusing on non-Euclidean structures (those with Gaussian curvature at their vertices) and elastic origami (those with faces that are allowed to deform and edges that are allowed to stretch). Both of these additions to rigid models expose some of the factors that contribute to misfolding and present new options for avoiding it in physical realizations.

In Chapter 2, we will continue to explore origami but for two new cases. In the first, we will consider self-actuating origami-like structures that are actuated on the faces rather than at the folds. Most experimental realizations of these structures are actuated at the fold, but an increasing number achieve actuation at the faces. Most models do not apply to these newer actuation methods. We present a method for relating torques on faces to forces on vertices, from which we can use previous methods, and along the way gain new insight into how non-triangulated faces contribute to the mechanisms' allowed motions. Second, we address the problem of how many folds are required to fully actuate an origami and present a new method for using tools from graph theory to identify these minimum fold sets. Using simulation methods developed in Chapter 1, we also see that the minimum fold sets can be used to design origami that minimize misfolding.

Finally, in Chapter 3 we will turn back to generalized linkages and address how to design linkages that change general behavior without changing their connectivity. To do so, we introduce a new formalism that represents the tangent to the configuration space for a given connectivity of a mechanisms as a function of its bar lengths. Finding the points where this tangent field changes sign gives the bar lengths at critical points - the only points across which a qualitative motion change can occur. Moreover, we can use this tangent field formalism to find entire critical value surfaces in the space of bar lengths. By categorizing these surfaces and the spaces delineated by them, we can find bar length changes corresponding to desired motion changes in the linkage. We demonstrate this design principle for both the four bar linkage, a classic engineering example, and to gate a Kane-Lubensky chain, a mechanism that can propagate signals along its length.

In Chapter 4 we wrap up our discussions and conclude the previous three chapters. We also examine how these seemingly disparate projects interweave, their contributions to the field, and the outlook for future projects built off of these.

CHAPTER 1

NON-IDEAL ORIGAMI

Past modeling of origami has been primarily focused on flat, rigid origami - origami that starts as a flat sheet and has rigid faces and folds, meaning the origami does not stretch and its faces do not bend. In this chapter we will discuss some of this past work and seek to fill some of these gaps, first by considering origami with Gaussian curvature at the vertices then by considering elastic origami.

A note on collaboration in this chapter: The first section was done with fellow graduate student Michelle Berry, with my primary contribution being the bar length expansion and decomposition that leads to Eq. 1.10. The second section was done with Professor Ryan Hayward and his postdoc Ji-Hwan Kang, who designed and performed the experiments, although all theoretical and numerical work was done by me.

1.1 Non-Euclidean Origami ¹

Though most examples of origami structures are foldable from an initially flat sheet, two threads of research suggest a need to understand the motions of a broader class of “curved” origami. First, kirigami structures, initially flat structures with holes which can be glued together along their free edges to create intrinsically buckled structures [30]. Second, newer origami fabrication methods have enabled vertices with Gaussian curvature and curved faces [53–56].

¹This section is adapted from Ref. [1] with permission from the American Physical Society.

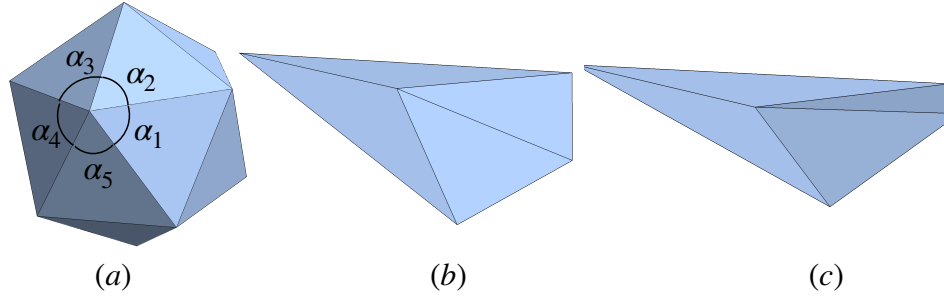


Figure 1.1. (a) A generic non-Euclidean origami structure. The vertex Gaussian curvature is defined by $K = 2\pi - \sum_i \alpha_i$ (b,c) Degree four vertices with positive and negative Gaussian curvatures respectively necessarily buckle out of the plane.

This section analyzes the kinematics of non-Euclidean origami in the limit that it is almost flat. By “non-Euclidean origami,” we mean that faces are flat, but that the vertices have Gaussian curvature (Fig. 1.1 a–c). This Gaussian curvature manifests as either a deficit or excess angle when summing the sector angles around the internal vertices (Fig. 1.1a). By “almost flat,” we mean that both the sum of sector angles around internal vertices is near 2π and that the dihedral angles of the folds are nearly π . In this limit, we will develop a general framework for studying origami motions, and make contact with both the kinematics of flat origami structures [41] and continuum equations governing the small deformations of elastic sheets [57].

Understanding whether an origami fold pattern can be folded without tearing is NP-hard [58]. More generally, when mapping out the space of possible configurations of a given origami fold pattern, the configuration space defined by the angles of the folds can be geometrically complex. Additionally, these spaces can undergo topological changes as the fold pattern changes that lead to changes in the mechanical properties of origami [14]. Here, we show that vertex Gaussian curvature can induce a topological change in the configuration space of origami structures. In particular, we will show that origami with positive Gaussian curvature vertices have configuration

spaces that become disconnected, and that such disconnection need not (and likely does not) occur for negative Gaussian curvature.

1.1.1 Mathematical Formulation

We model origami by a collection of polygonal faces meeting at point-like vertices and joined along rigid edges, as shown in Fig. 1.1 for triangular faces. We find it useful to distinguish internal vertices, whose number we will denote V_i , from boundary vertices, whose number is V_b . Note that in traditional origami nomenclature a “vertex” denotes only the internal vertices. Similarly, we denote the internal and boundary edges by E_i and E_b , respectively. The internal edges are the folds in the origami literature.

We are primarily interested in determining the isometries of a given origami fold pattern, *i.e.* the motions that preserve the length of all edges and the angles between any two adjacent edges on the same face. In the case of triangular faces, the angle constraint is redundant – once the length of all the edges are known, the angles between edges are already uniquely determined.

In this section, we will focus mainly on origami with triangular faces. This is not very restrictive; we will see that the configuration space of an origami structure with polygonal faces can be obtained by taking a lower dimensional slice through the configuration space of a suitable triangulated origami fold pattern. In Section 1.2, we will additionally show that this can account for face deformations. To define the discrete Gaussian curvature of an internal vertex, we measure the sector angles, α_i , between adjacent folds with one end on a given vertex (Fig. 1.1a). The Gaussian curvature of that vertex is then $K_n = 2\pi - \sum_i \alpha_i$ [59].

One of the primary features of triangulated origami is that the number of infinitesimal isometries is almost precisely balanced by the number of constraints. This is true for any Gaussian curvature though it manifests in different ways when $K_n = 0$

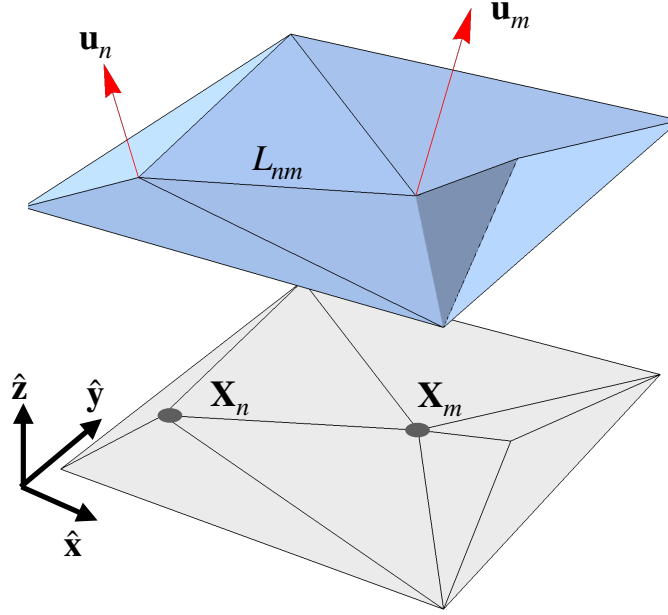


Figure 1.2. A nearly flat origami structure can be projected to a fold pattern in the xy -plane. In-plane and out-of-plane displacements are unambiguously decomposable.

on each internal vertex. Understanding this distinction turns out to be important to developing a fuller picture of the origami configuration space so we review it here. If \mathbf{X}_n denotes the three dimensional position of the n^{th} vertex, then any pair of vertices joined by an edge induces a geometrical constraint,

$$(\mathbf{X}_n - \mathbf{X}_m)^2 = L_{nm}^2, \quad (1.1)$$

where L_{nm} is the length of the edge between n and m . We then write \mathbf{u}_n (Fig. 1.2b) as the displacement of the n^{th} vertex, and find that, to first order, motions are governed by the linear equations

$$2(\mathbf{X}_n - \mathbf{X}_m) \cdot (\mathbf{u}_n - \mathbf{u}_m) = 0. \quad (1.2)$$

There is one equation of this type for each edge (n, m) , where n and m represents the vertices connected by the edge.

To understand the generic behavior of Eq. (1.2), we note that there are $E_i + E_b$ constraints, one for each edge and $3V_i + 3V_b$ naive degrees of freedom associated with the three-dimensional displacements of the vertices. A triangulated origami fold pattern also satisfies both Euler's theorem, $F - E_i - E_b + V_i + V_b = 1$, where F is the number of faces, and satisfies the $2E_i + E_b = 3F$ to account for the fact that each face is associated to three edges but each internal edge joins two faces. Similarly, we have $E_b = V_b$ because the boundary of the fold pattern is a polygon. Taken together, these equations imply $E_i = V_b + 3V_i - 3$ and so naive counting suggests that the dimension of the configuration space of origami is $D = V_b + 3$. Six of these degrees of freedom are Euclidean motions.

Though this generic counting should be valid for most configurations, it fails when the origami is flat because the constraints at first order are not all independent. In that case, only the in-plane deformations are fixed by the length constraints: any vertex can be displaced vertically without causing a first-order change in the edge lengths. Though this suggests that $D = V_i + V_b + 3$, it turns out that there are additional constraints at quadratic order in the lengths. If we define $\mathbf{h} = (h_1, h_2, \dots)$ as a vector specifying the vertical displacement of each of the vertices above the xy -plane, then a necessary and sufficient condition for a motion to be an isometry to second order is

$$\mathbf{h}^T \mathbf{Q}_n \mathbf{h} = 0, \tag{1.3}$$

for each internal vertex, n , where the matrix \mathbf{Q}_n depends on the sector angles of internal vertex n [41]. The left-hand side of Eq. (1.3) is the Gaussian curvature of internal vertex n induced by the height changes [41] so Eq. (1.3) is simply the statement that no infinitesimal deformation can change the Gaussian curvature of the internal vertices. There are precisely enough quadratic constraints, one for each internal vertex, to recover the generic result, $D = V_b + 3$.

We now wish to modify Eq. (1.3) to allow for internal vertices to have a small but nonzero Gaussian curvature. In this regime, the vertices continue to remain almost planar, although will not typically lie flat. Our problem, then is to reconcile the linear and quadratic length-preserving motions. Hence, we would expect them to be well-described by the linear equation Eq. (1.2). As the Gaussian curvature goes to zero, however, quadratic constraints must somehow emerge.

As before, we will approach the analysis of the possible motions by expanding around the flat state. We expect this expansion to be valid so long as the Gaussian curvature of the vertices is sufficiently small. Denoting the planar angles around any vertex with α_n , the discrete Gaussian curvature is $K = 2\pi - \sum_n \alpha_n$. We imagine that the deformation of a structure is governed by an expansion of the form

$$\mathbf{X}_n = \mathbf{X}_n^{(0)} + \mathbf{u}_n^{(1)} + \mathbf{u}_n^{(2)} \quad (1.4)$$

where $\mathbf{X}_n^{(0)}$ is the position of a flattened origami structure and the superscript of \mathbf{u} represents the order in a formal expansion of the displacement from the flat state.

Because we are expanding the deformations around an otherwise flat structure, the equilibrium lengths of the edge connecting vertex n and m will not be represented by the distances between the planar vertex positions, $\mathbf{X}_n^{(0)}$. Instead, we let $\Delta_{nm} = L_{nm}^2 - (\mathbf{X}_n^{(0)} - \mathbf{X}_m^{(0)})^2$ measure the deviation of the equilibrium edge lengths from the lengths of the edges when projected to the xy -plane. We denote Δ the vector formed by concatenating the components Δ_{nm} for each edge. We similarly write $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ as the concatenation of the vertex displacements at first and second order. Finally, introduce a quadratic function, $\mathbf{f}(\mathbf{u})$ with components $(\mathbf{u}_n - \mathbf{u}_m)^2$ for each edge, (n, m) . Then we have

$$\Delta = \mathbf{R}\mathbf{u}^{(1)} + \mathbf{R}\mathbf{u}^{(2)} + \mathbf{f}(\mathbf{u}^{(1)}), \quad (1.5)$$

where \mathbf{R} is the compatibility matrix mapping vertex displacements to linear changes in the edge lengths [60, 61].

To linear order, one should solve $\Delta = \mathbf{R}\mathbf{u}^{(1)}$. However, this linear equation can only have a solution if the left-hand side of the equation lies in the image of \mathbf{R} . We denote the projection of a vector into the image of \mathbf{R} with a subscript \parallel , and a projection into the orthogonal complement \perp . Therefore, Eq. (1.5) decomposes into the pair

$$\Delta_{\parallel} = \mathbf{R}\mathbf{u}^{(1)} + \mathbf{R}\mathbf{u}^{(2)} + \mathbf{f}_{\parallel}(\mathbf{u}^{(1)}) \quad (1.6)$$

$$\Delta_{\perp} = \mathbf{f}_{\perp}(\mathbf{u}^{(1)}) \quad (1.7)$$

Eq. (1.6) can now be solved order by order. To first order, $\mathbf{u}^{(1)} = \mathbf{u}_{\parallel} + \mathbf{h}$, where \mathbf{u}_{\parallel} is any solution of $\mathbf{R}\mathbf{u}_{\parallel} = \Delta_{\parallel}$, and \mathbf{h} is in the right null space of \mathbf{R} . At the next order, we obtain a correction $\mathbf{R}\mathbf{u}^{(2)} = -\mathbf{f}_{\parallel}(\mathbf{u}_{\parallel} + \mathbf{h})$.

Since we are expanding around a flat origami structure, we can further restrict the structure of \mathbf{u}_{\parallel} and \mathbf{h} . Particularly, it must be that \mathbf{h} can only involve the three in-plane Euclidean motions and the vertical displacements of all of the vertices. Consequently, \mathbf{u}_{\parallel} can be chosen so that the vertex displacements lie in the xy -plane and \mathbf{h} can then contain only vertex displacements along the $\hat{\mathbf{z}}$.

Eq. (1.7) is not dispensed with so easily. It remains a quadratic constraint on \mathbf{h} of the form

$$\Delta_{\perp} = \mathbf{f}_{\perp}(\mathbf{u}_{\parallel} + \mathbf{h}) = \mathbf{f}_{\perp}(\mathbf{u}_{\parallel}) + \mathbf{f}_{\perp}(\mathbf{h}). \quad (1.8)$$

The last equality follows from the fact that \mathbf{u}_{\parallel} is perpendicular to \mathbf{h} and \mathbf{f} is quadratic. Finally, we neglect $\mathbf{f}_{\perp}(\mathbf{u}_{\parallel})$ since it is quadratic in $|\Delta_{\parallel}|$. This is valid when $|\Delta_{\perp}| \sim |\Delta_{\parallel}|$.

To interpret Eq. (1.8), we let $\{\sigma_1, \sigma_2, \dots\}$ be the basis of wheel stresses of $\ker \mathbf{R}^T$ described in Ref. [41]. In this basis,

$$\boldsymbol{\sigma}_n \cdot \mathbf{f}(\mathbf{h}) \approx \boldsymbol{\sigma}_n \cdot \boldsymbol{\Delta}, \quad (1.9)$$

where the left-hand side can be interpreted as the discrete Gaussian curvature at vertex n , K_n , or alternatively as a quadratic form, $\mathbf{h}^T \mathbf{Q}_n \mathbf{h}$ [41]. This allows us to arrive at, finally,

$$\mathbf{h}^T \mathbf{Q}_n \mathbf{h} = \boldsymbol{\sigma}_n \cdot \boldsymbol{\Delta} \equiv K_n. \quad (1.10)$$

We note that, when $K_n = 0$, Eq. (1.10) reproduces the results of Chen *et al.* [41] for flat origami. Notice that the right-hand side of Eq. (1.10) involves only lengths of the bonds, encoded through $\boldsymbol{\Delta}$. This is then a discrete version of Gauss' *theorema egregium*, which relates the Ricci curvature on a surface – a completely intrinsic quantity – to the Gaussian curvature – an extrinsic quantity.

Eq. (1.10) can be contrasted to the equations governing the small isometries of a continuum elastic sheet, which are governed by the approximate equations [57]

$$K = -\frac{1}{2} \sum_{ijkl=1}^2 \epsilon_{ik} \epsilon_{jl} \partial_i \partial_j h \partial_k \partial_l h, \quad (1.11)$$

where ϵ_{ij} is the antisymmetric Levi-Civita symbol with $\epsilon_{12} = 1$, $h(x, y)$ is the vertical height of the elastic sheet above the xy -plane, and $K(x, y)$ is the Gaussian curvature. Eq. (1.11) is accurate in the limit of small slopes $|\partial_i h| \ll 1$, which is precisely the same limit of our discrete formulation. In that sense, Eq. (1.10) is a discrete analogue to the better known continuum result of Eq. (1.11).

We can also rewrite the Gaussian curvature around any internal vertex in closed form. To do so, we need to introduce some additional notation. We denote the central vertex with 0 and number the boundary vertices from $n = 1$ to N . Denote $\alpha_{n,n+1}$ as the angle between fold n and $n + 1$, interpreted assuming $\alpha_{N,N+1} = \alpha_{N,1}$, and assume that $\alpha_{n,n+1}$ is always between 0 and π . Since we are interested in single vertices near the flat state, it is useful to change variables from the vertex heights to the angles

made by the folds with respect to the $\hat{\mathbf{z}}$ axis, oriented with respect to the reference z -axis: $\psi_n = \pi/2 + (h_0 - h_n)/L_{n0}$ where L_{n0} is the length of fold n .

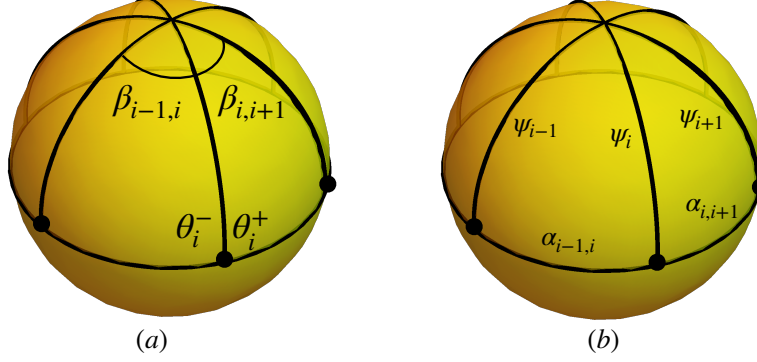


Figure 1.3. The intersection of a sphere with a vertex at its center is a spherical polygon, which we decompose into triangular slices as shown. (a) The dihedral angle of the i^{th} fold is $\theta_i^+ + \theta_i^-$. (b) The side lengths are the planar angles $\alpha_{i,i+1}$ and the angle the folds make with respect to the xy -plane, ψ_i .

It is well known that a single vertex can be interpreted as a spherical polygon in which the side lengths are given by the planar angles $\alpha_{n,n+1}$ and the dihedral angles by the interior angles of the polygon (Fig. 1.3); this connection has been used to explore the full configuration space of single origami vertices in general [62, 63]. Fig. 1.3 shows that such a polygon can be decomposed into triangular slices. Spherical trigonometry then allows one to write the dihedral angles entirely in terms of the ψ_n . For small deformations, these N angles $\psi_n = \pi/2 + \delta\psi_n$ where $\delta\psi_n = (h_n - h_0)/L_{n0}$. Finally, we define θ_n as the dihedral angle made by the n^{th} fold; the diagram in Fig. 1.2 shows that $\theta_n = \theta_n^+ + \theta_n^-$. Finally, we let $\theta_n = \pi - \delta\theta_n$ and assume $\delta\theta_n$ is small.

Expanding to quadratic order, we obtain the linear relationship

$$\delta\theta_n = \sum_m M_{nm} \delta\psi_m \quad (1.12)$$

where

$$\begin{aligned}
M_{nm} = & -\csc \alpha_{n,n+1} \delta_{n,m+1} - \csc \alpha_{n-1,n} \delta_{n,m-1} \\
& + (\cot \alpha_{n,n+1} + \cot \alpha_{n-1,n}) \delta_{nm}.
\end{aligned} \tag{1.13}$$

Expanding the angles $\beta_{n,n+1}$ around $\alpha_{n,n+1}$ and using $\sum_n \beta_{n,n+1} = 2\pi$, we also find an expression for the Gaussian curvature of the vertex, $K = 2\pi - \sum_n \alpha_{n,n+1}$,

$$K = -\frac{1}{2} \sum_{nm} \delta\psi_n \delta\psi_m M_{nm}. \tag{1.14}$$

Comparing Eq. (1.14) to Eq. (1.10) provides a connection between the matrix \mathbf{Q} governing the configuration space in terms of the vertex heights to the matrix \mathbf{M} , having components M_{nm} , governing the configuration space in terms of angles $\delta\psi_n$. In particular, while \mathbf{Q} should have an additional zero eigenvalue from global translations of the vertex in the $\hat{\mathbf{z}}$ direction, it shares the same number of positive and negative eigenvalues as \mathbf{M} [41].

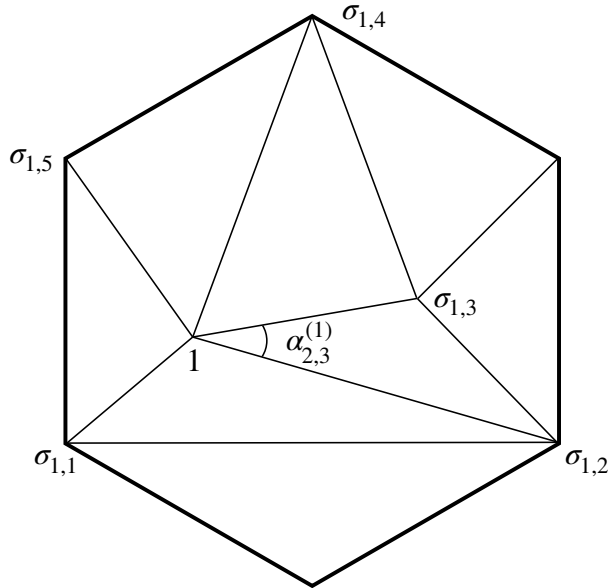


Figure 1.4. Notation for the vicinity of a single vertex when there are multiple vertices.

To rewrite this and expand to multiple internal vertices, we introduce even more notation, the illustration for which can be seen in Fig. 1.4. Let $\sigma(n, 1)$ through $\sigma(n, N(n))$ be the vertices connected to an internal vertex n in counterclockwise order, where $N(n)$ is the number of edges with n at one end. We also denote L_{nm} as the length of the edge joining vertex n to m . Then,

$$K_n = -\frac{1}{2} \sum_i^{N(n)} \sum_j^{N(n)} \left(\frac{h_{\sigma(n,i)} - h_n}{L_{\sigma(n,i)n}} \right) M_{ij}^{(n)} \left(\frac{h_{\sigma(n,j)} - h_n}{L_{\sigma(n,j)n}} \right), \quad (1.15)$$

where the matrix $M_{ij}^{(n)}$ is the same matrix as in 1.13 but re-indexed for clarity. Note that it is an $N(n) \times N(n)$ square matrix depending on the sector angles around each internal vertex. Additionally, note that for sector angles smaller than π , the matrices $\mathbf{M}^{(n)}$ have two zero eigenvalues, one negative eigenvalue, and the remaining eigenvalues are positive (see Ref. [62] or Appendix C of Ref. [41] for a detailed proof).

1.1.2 Single vertices

To better understand Eq. (1.15), consider an origami structure with one internal vertex from which N folds emerge (Fig. 1.5a). This case has been analyzed in some depth due to the correspondence between origami vertices of degree N and spherical linkages with N segments [62, 63]. We denote the height of the central vertex h_0 and the heights of the surrounding vertices h_1 through h_N , and we explicitly eliminate rigid body motions by fixing the heights of vertices $(h_0, h_1, h_2) = (0, 0, 0)$. One quadratic constraint remains on the remaining heights, h_3 to h_N , leaving $N - 3$ distinct degrees of freedom.

So what does the configuration space of a single non-Euclidean vertex look like? We suppose $\mathbf{e}_{-,i}$ are the components of the normalized eigenvector corresponding to the negative eigenvalue, $-\lambda_-$, of $\mathbf{M}^{(0)}$. We then suppose $\mathbf{e}_{n,i}$ are the components of

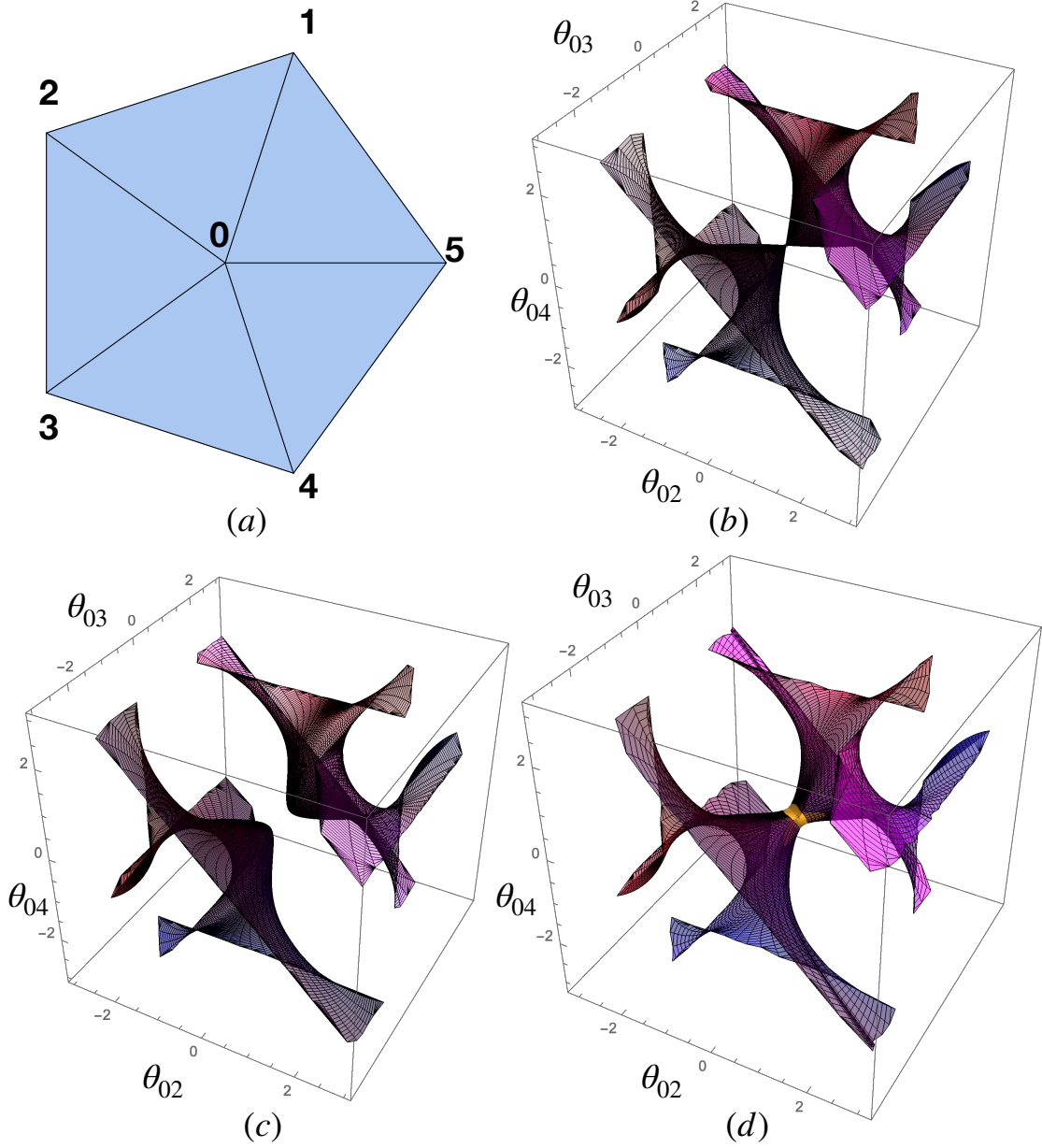


Figure 1.5. The configuration space of a symmetric five-fold vertex (a) near the flat state with zero (b), positive (c), and negative (d) Gaussian curvature projected onto the fold angles $(\theta_{02}, \theta_{03}, \theta_{04})$. The fraction of red and blue, $[r, b]$, in the coloring is determined by $[(\theta_{01} - \pi)/(2\pi), (\theta_{05} + \pi)/(2\pi)]$.

the n^{th} normalized eigenvector with positive eigenvalues, λ_n . We can then attempt to solve Eq. (1.15) with the ansatz

$$\frac{h_i - h_0}{L_{i0}} = \frac{1}{\sqrt{2\lambda_-}} c_- \mathbf{e}_{-,i} + \sum_n \frac{1}{\sqrt{2\lambda_n}} c_n \mathbf{e}_{n,i}. \quad (1.16)$$

We find that $c_-^2 - \sum_n c_n^2 = K_0$, where K_0 is the Gaussian curvature of vertex 0.

When $K_0 = 0$, we recover the results of Ref. [41]: the solution forms a cone described by the equation $c_- = \pm \sqrt{c_n^2}$ with a singularity at $c_- = c_n = 0$. Each nappe of the solution space is characterized by the sign of c_- (called branch signs in [41]). When $K_0 > 0$, we must instead solve,

$$c_-^2 = K_0 + \sum_n c_n^2, \quad (1.17)$$

showing that $|c_-| \geq K_0$. This would seem to imply that the two nappes have split into two disconnected components characterized by the sign of c_- . Finally, we turn to $K_0 < 0$, for which

$$c_-^2 + |K| = \sum_n c_n^2. \quad (1.18)$$

Here, it is clear that there is no obstruction to $c_- = 0$. Instead, $\sum_n c_n^2 \geq |K|$. We conclude that the conical configuration space is one in which both nappes remain connected near the flat state but are connected by a neck (Fig. 1.5d). This is quite different than what happens for degree four vertices [64].

In Fig. 1.5, we numerically plot the configuration space of a symmetric degree-5 vertex. To do this, we compute radial trajectories from a known configuration of the origami vertex. Each point of the radial trajectory is found in a sequence of steps. For each step, we solve Eq. (1.2) to identify the infinitesimal isometries from any configuration that is not flat and project the previous tangent direction onto the

new tangent space. After finding a new configuration using the linear isometry, we numerically minimize the energy functional,

$$E = \frac{1}{2} \sum_{nm} [(\mathbf{X}_n - \mathbf{X}_m)^2 - L_{nm}^2]^2, \quad (1.19)$$

where the sum is over edges joining vertex n to m , using the BFGS (“QuasiNewton”) algorithm in Mathematica 11. This prevents numerical errors in the linear isometries from building up as the integration proceeds. This process proceeds until one of the fold angles exceeds π or $-\pi$, indicating that a face has come into contact with an adjacent face. Finally, the trajectories are assembled into a mesh to produce a surface.

Generically, we find that the configuration space near the flat state follows the analytical results we obtained. Specifically, it appears that the configuration space decomposes into two nappes with the topology of a disk which are either touching at one point ($K = 0$), disconnected ($K > 0$), or connected by a narrow neck ($K < 0$). At first glance, this appears to contradict Streinu and Whitely [63], who showed that the configuration space of single vertices with $K_0 > 0$ is always connected. However, in their analysis, faces can pass through each other; whereas in Fig. 1.5, fold angles must remain strictly between $-\pi$ and π . Anecdotally, it does appear that when faces are allowed to pass through each other, isometric trajectories can pass from one nappe to the other for any K . In this case, however, the surfaces become difficult to plot, even more difficult to understand, and, in any case, are unphysical.

Degree-four vertices, those with only four folds emerging from a central vertex, are a special case that has been recently explored [65]. The configuration space of a degree-four vertex can be obtained from Fig. 1.5 by considering a particular planar slice. For example, if we create a degree-four vertex by removing fold θ_{02} from Fig. 1.5a, the configuration space of the degree-four vertex is the intersection of the surfaces in Fig. 1.5 with the plane $\theta_{02} = 0$. This configuration space is, therefore, one

dimensional and the two nappes become disconnected for both positive and negative Gaussian curvature.

This reasoning can also be used to explore the configuration spaces of non-triangulated origami. If we are given an arbitrary origami fold pattern, any non-triangular faces can be triangulated, introducing new fold angles, (ϕ_1, \dots, ϕ_M) . The proper isometries of the non-triangulated origami are then the intersection of the triangulated configuration space with the hyperplane defined by $(\phi_1, \dots, \phi_M) = 0$. Therefore, the dimension of the configuration space becomes $D = V_b - 3 - M$, where M is the number of diagonals added to triangulate the fold pattern. Because these hyperplanes pass through the origin (where the origami is unfolded), they do not change the fundamental topology of the configuration spaces of triangulated origami shown in Fig. 1.5.

The configuration spaces in Fig. 1.5b – d give us a first picture of the interplay between origami energetics and kinematics. If we imagine that a torsional spring of stiffness κ has been placed on each fold of Fig. 1.5a, the energy functional would be $E = (\kappa/2) \sum_{i=1}^N \theta_{ni}^2$. The equi-energy surfaces are given by spheres centered on the state with $\theta_{0i} = 0$ and so the ground state is the configuration (or configurations) that are closest to the flat state. The previous section provided some mathematical machinery to expand this discussion to general origami fold patterns, which we will continue to use here. In addition to determining the kinematics of an origami structure near the flat state, the matrix $M_{ij}^{(n)}$ also determines the fold angles as a function of the vertex heights through

$$\theta_{\sigma(n,i)n} = \sum_j^{N(n)} M_{ij}^{(n)} \left(\frac{h_{\sigma(n,j)} - h_n}{L_{\sigma(n,j)n}} \right), \quad (1.20)$$

where $\theta_{\sigma(n,i)n}$ is the fold angle connecting vertex $\sigma(n,i)$ to vertex n . Note that the quadratic terms in Eq. (1.20) actually vanish so this equation is accurate to quadratic

order as well. Using Eq. (1.20) we can write an energy functional for a nearly-flat origami structure as

$$E = \frac{1}{2} \sum_n \sum_{ijk}^{N(n)} \kappa_{n\sigma(n,i)} M_{ij}^{(n)} M_{ik}^{(n)} \times \left(\frac{h_{\sigma(n,j)} - h_n}{L_{\sigma(n,j)n}} \right) \left(\frac{h_{\sigma(n,k)} - h_n}{L_{\sigma(n,k)n}} \right) \quad (1.21)$$

where the sum over n is over internal vertices only. Any fold that joins an internal vertex n to a boundary vertex k has torsional stiffness κ_{nk} whereas a fold connecting internal vertex n to internal vertex m has stiffness $2\kappa_{nm}$ because such folds are double counted in Eq. (1.21). Thus, for a single vertex with equal fold stiffness κ and zero equilibrium fold angles, the decomposition of deformations in terms of collective variables c_- and c_n yields an energy

$$E = \frac{1}{2} \kappa \left[\lambda_- (c_-)^2 + \sum_n \lambda_n (c_n)^2 \right]. \quad (1.22)$$

For $K_0 > 0$, we introduce a new collective variable ξ such that $c_- = \pm K_0 \cosh \xi$ and $c_n = K_0 n_n \sinh \xi$, where n_n are the components of a unit vector. When $K_0 < 0$, we instead use $c_- = |K| \sinh \xi$ and $c_n = \pm |K| n_n \cosh \xi$. Therefore,

$$E = \frac{\kappa K^2}{2} \begin{cases} \lambda_- \cosh^2 \xi + \sinh^2 \xi \sum_n \lambda_n (n_n)^2, & K_0 > 0, \\ \lambda_- \sinh^2 \xi + \cosh^2 \xi \sum_n \lambda_n (n_n)^2, & K_0 < 0. \end{cases} \quad (1.23)$$

There is an obvious generalization of Eq. (1.23) to the case when the fold stiffnesses are not all equal.

For both signs of K_0 , Eq. (1.23) has a minimum at $\xi = 0$. When $K_0 < 0$, this implies $E = \kappa K^2 \lambda_- c_-^2 / 2$ and is independent of the choice of n_n or the values of λ_n . There are two energy minima corresponding to the two points closest to the flat state

in Fig. 1.5c, independent of any other details of the shape. When $K_0 < 0$, on the other hand, the component of n_n corresponding to the smallest eigenvalue λ_n will be 1 and the remaining components will be 0. Hence, $E = \kappa K^2 \lambda_{n_{min}} c_{n_{min}}^2 / 2$, where n_{min} is the index of the smallest eigenvalue.

1.1.3 Discussion

To conclude, we have derived the form of the configuration space of non-Euclidean origami for small amounts of Gaussian curvature near the flat state. For single positive Gaussian curvature vertices, the configuration is characterized by nappes that are separated near the flat state, whereas for negative Gaussian curvature, the configuration space remains connected. Though we have analyzed the case of a single degree- N vertex in detail, the procedure we have used can be applied to explore the kinematics and energetics of more complex, nearly flat origami structures with or without Gaussian curvature. We first consider the case of multiple vertices with $K_n > 0$. Around each vertex, Eq. (1.15) establishes a single equation for h_n as a function of the heights of the vertices surrounding it. We further assume that this equation has two distinct real solutions for h_n . Then the analysis in the previous section establishes that no matter how we deform the boundary vertices, there is no way for the configuration of this vertex to pass from one configuration space nappe to the other. The conclusion is that distinct branches of the configuration space of a complex, origami fold pattern that are distinguished by a $K > 0$ vertex being on different nappes are topologically disconnected – if they were not, there would be also be a way of passing from one nappe to the other on a single vertex. Unfortunately, it is difficult to determine whether or not every combination of nappes can be realized when $K > 0$. The case for $K < 0$ is murkier because, while a single vertex remains connected, there is no reason that global constraints might not lead to disconnected components of the configuration space. Indeed, this must be possible in principle, as

triangulated fold patterns with disconnected configuration spaces, albeit rare, have been found [50].

We also note that this work provides a new mechanism by which the mechanical response of an origami metamaterial sheet can be molded. In principle, an initially flat structure could be stiffened by imposing a small amount of positive Gaussian curvature. Moreover, Gaussian curvature provides a new means of controlling how a responsive origami structure self-folds by separating the individual nappes so that misfolding is significantly less likely. A more thorough investigation of the interplay between self-folding and Gaussian curvature could be done through simulation. In the next section we will discuss simulating self-folding in more detail, but here we will simply point out that inducing Gaussian curvature at a single vertex within a structure is as simple as lengthening or shortening the equilibrium lengths of the folds connecting to that vertex while maintaining the original equilibrium lengths of the rest of the structure.

1.2 Elastic Origami ²

In this section we will more directly address misfolding and its causes and possible prevention.

Before moving on, it is useful to introduce previously introduced methods to prevent misfolding. Tachi and Hull have proposed a method that takes advantage of the branched structure of the origami configuration space [39]. They assume each fold is a torsional spring and adjust the torques induced by the springs to force the origami in a direction in configuration space perpendicular to all undesirable folding pathways. Unfortunately, due to the high dimensionality of the configuration space, there is often no choice of torques that satisfies all of these requirements [41]. On the

²This section is adapted from Ref. [2] with permission from the Royal Society of Chemistry

other hand, Stern *et al.* explored a large class of origami structures made from only quadrilateral faces and, even in this restricted set, found a proliferation of energy minima [42, 52]. However, most rigid quadrilateral origami cannot be folded at all [44], and so these energy minima represent configurations involving stretching rather than distinguishing between several valid branches. This suggests a more careful treatment of elasticity in origami is crucial to uncovering the mechanisms of misfolding.

In this section, we compute energy landscapes of weakly folded origami using a bar-and-hinge model of self-folding that includes both face stretching and face bending [16, 43, 44, 50]. Energy landscapes provide a detailed picture of the vicinity of the flat state, where multiple origami branches meet. We show that the mechanisms governing the formation of competing local energy minima are poorly captured by the assumption of rigid, unstretchable origami. Instead, the undesirable energy minima that compete with the target configuration are regulated by saddle-node bifurcations nucleated near the unfolded state, even when the target configuration is very folded. Our model allows us to determine how the “foldability” of an origami design is determined both by the stretching and bending moduli of the faces: more bendable faces allow additional folding pathways while more stretchable faces induce saddle-node bifurcations that reduce the number of local energy minima. Critically, the reduction in the number of energy minima does not arise from transitions between branches that induce large strains in the faces but is, nevertheless, enabled by small amounts of strain while the origami is barely folded. Our analysis leads to new insights on the robustness of the target folding pathway to programming errors in the target fold angles.

To go beyond our theoretical analysis, we also demonstrate these effects experimentally on self-folding origami structures using a previously-published trilayer swelling gel system [47, 51]. These experiments demonstrate the possibility of using

face stiffness to tune the metastability of self-folding origami. In contrast, the few methods that have been proposed to prevent misfolding require a more careful tuning of fold angles and stiffnesses [39, 42, 52]. We demonstrate that robust folding can still be induced in systems where such precise control may not be possible.

1.2.1 Folding rigid origami

Rigid origami, having both unbendable and unstretchable faces, can be modeled as a triangulated surface with V vertices joined by N edges of fixed length spanned by F polygonal faces with additional torsional springs. Because each face is decomposed into triangles, the length constraints of the edges also preserve the sector angles of the faces. We assume there are N_B edges that are adjacent to a single face which we dub boundary edges to distinguish them from the N_F edges that adjoin a pair of faces, which we refer to as folds. Many origami fold patterns do not have triangular faces, however. In those cases, we decompose each face into triangular subfaces along their shortest diagonals [27, 28, 44]. This suggests a further division of folds into “face folds”, those folds spanning a rigid face, and “active folds”, which drive the self-folding of the origami.

Since the faces are triangular, the state of any origami structure can be represented completely by its fold angles, $(\rho_1, \dots, \rho_{N_F})$, where each angle ρ_i is the supplement of the corresponding dihedral angle made by the faces adjacent to the edge. We introduce self-folding by incorporating torsional springs on the folds of the form

$$E_B = \frac{1}{2} \sum_{I \leq N_F} \kappa_{B,I} (\rho_I - \bar{\rho}_I)^2, \quad (1.24)$$

where $\kappa_{B,I}$ is the torsional modulus of the I^{th} fold and $\bar{\rho}_I$ the equilibrium angle of the fold. For face folds, we require $\bar{\rho}_I = 0$ so that Eq. (1.24) penalizes bending of the faces. On active folds, however, $\bar{\rho}_I \neq 0$ which imposes a bending torque that drives the origami to fold along its active folds.

For models of this type, there are singular configurations where several branches of allowed configurations meet [1, 39, 41]. Each branch has a tangent space where it meets the singular configuration and many such branches meet at this point [41]. Tachi and Hull have proposed that misfolding can be prevented when the torque, defined by $\tau_I = -\kappa_{B,I}\bar{\rho}_I$, is perpendicular to the tangent space of each branch [39] in the space of folds. Indeed, examining Eq. (1.24) shows that the Tachi-Hull condition is precisely the condition that there is no direction along an undesirable branch along which the energy decreases.

It is notable that the Tachi-Hull condition can be impossible to satisfy if the origami fold pattern is sufficiently complicated, as the number of branches grows exponentially with the number of vertices while the number of folds grows linearly [41]. If there are several branches along which E_B decreases, each of these branches must have at least one local energy minimum. Consequently, the number of potential competing energy minima in a rigid, self-folding origami system can be quite sensitive to even small errors in the programmed torques τ_I . The question of stability and metastability of an origami folding becomes even more complex when one considers Eq. (1.24) along an entire origami trajectory, and such an analysis has only been undertaken for some single origami vertices [49].

1.2.2 Elastic origami

Part of the sensitivity of competing minima to torques arises from the singular nature of the unfolded, flat origami. To study this further, we augment our model to allow for stretching. We supplement Eq. (1.24) with additional terms [43] of the form,

$$E_S = \frac{1}{2} \sum_{i \leq N} \kappa_{S,i} \gamma_i^2, \quad (1.25)$$

where $\kappa_{S,i}$ the stiffness of edge i and the elastic strain is given by

$$\gamma_i = \frac{1}{2} \left(\frac{L_i^2}{\bar{L}_i^2} - 1 \right). \quad (1.26)$$

Note that small deformations of the edges $\Delta \ll \bar{L}_i$ gives $\gamma_i \approx \Delta/\bar{L}_i$ as does the slightly more common form for the elastic strain $\gamma_i = L_i/\bar{L}_i - 1$.

By formulating the energy in terms of a dimensionless strain, $\kappa_{S,i}$ has the same units as $\kappa_{B,I}$ evaluated on the same edge ($i = I$). We set $\kappa_{S,I} = Y_{2D}\bar{A}_I$ where Y_{2D} is the two dimensional Young's modulus of the origami faces and \bar{A}_I is a characteristic face area. Here, we will set \bar{A}_I to one third the total area of the faces adjoining edge I , which implies that $A_I \propto L_I$, and that edges can be subdivided without changing the energy cost of a given strain. There are more complex choices for $\kappa_{S,I}$ that are expected to capture more detailed features of the stretching deformations [45]. As an alternative, we also consider a more realistic model in which the faces themselves are elastic polygons that deform affinely, finding good agreement with our simpler model. We will go into more detail about both of these points in the subsequent sections. The advantage of Eq. (1.25) is that it allows us to make contact with the rigidity theory of frameworks on which the analysis of branched configuration spaces has been done [41, 49].

Eq. (1.25) also provides a convenient geometrical interpretation of the stretching energy of weakly-folded origami in terms of the Gaussian curvature of the vertices [1]. In the limit that $\kappa_{S,I} \gg \kappa_{B,I}$, we obtain an approximate expression for the energy valid when the fold angles are small. To do so, we note that vertical motions of the vertices off the xy -plane preserve L_I to lowest order. Therefore, after accounting for rigid body motions we can express E_S as a function of the $V_B + V_I - 3$ vertex heights only, where V_B is the number of vertices adjoining a boundary edge and V_I are the number of vertices adjoining only folds. This expression for the energy, quartic in the vertex heights and can be expressed as

$$E_S = \frac{1}{8} \sum_{n \leq V_B + V_I - 3} (\mathbf{h}^T \mathbf{Q}_n \mathbf{h})^2 \quad (1.27)$$

for a vector of vertex heights $\mathbf{h} = (h_1, \dots, h_{V_I})$ where h_i is the height of the i^{th} vertex and \mathbf{Q}_n a symmetric matrix which encodes the geometrical constraints associated with the branches as well as the stiffnesses of the origami. One can show that $E_S = 0$ if and only if the discrete Gaussian curvature of each origami vertex vanishes, and that the matrices \mathbf{Q}_n have two zero eigenvalues, one eigenvalue of either positive or negative sign, and the rest of the opposite sign [41].

To derive the elastic energy for nearly flat origami in the small strain limit in Eq. 1.27, we assume we have already added face folds so that the origami is built from only triangular faces. We then define a vector function of the vertex positions

$$f_i(\mathbf{u}) = \frac{\sqrt{K_i}}{2} \left(\frac{L_i^2}{\bar{L}_i^2} - 1 \right) \quad (1.28)$$

where $\mathbf{u} = (\mathbf{X}_1, \dots, \mathbf{X}_V)$ is a vector containing the position of all V vertices. Then the stretching energy is written as

$$E_S = \frac{1}{2} \sum_{i=1}^E f_i(\mathbf{u})^2. \quad (1.29)$$

We now expand $f_i(\mathbf{u})$ around the flat state \mathbf{u}_0 to find

$$f_i(\mathbf{u}_0 + \delta \mathbf{u}) \approx \partial_n f_i(\mathbf{u}_0) \delta u^n + \frac{1}{2} \partial_n \partial_m f_i(\mathbf{u}_0) \delta u^n \delta u^m \quad (1.30)$$

We next construct an orthonormal basis in the space of possible edges indexed by i , $\{\sigma_{1,i}, \dots, \sigma_{S,i}, e_{1,i}, \dots, e_{E-S,i}\}$ where $\sum_i \sigma_{N,i} \partial_n f_i(\mathbf{u}_0) = 0$. The $\sigma_{N,i}$ are, therefore, the components of the self-stresses of the linkage representing the origami structure.

Now the energy can be written as

$$E_S = \frac{1}{2} \sum_{N=1}^{E-S} \left[\sum_i e_{N,i} \left(\partial_n f_i(\mathbf{u}_0) \delta u^n + \frac{1}{2} \partial_n \partial_m f_i(\mathbf{u}_0) \delta u^n \delta u^m \right) \right]^2 \quad (1.31)$$

$$+ \frac{1}{2} \sum_{N=1}^S \left(\sum_i \sigma_{N,i} \frac{1}{2} \partial_n \partial_m f_i(\mathbf{u}_0) \delta u^n \delta u^m \right)^2$$

Finally, we drop higher order contributions to the first term to obtain

$$E_S = \frac{1}{2} \sum_{N=1}^{E-S} \left(\sum_i e_{N,i} \partial_n f_i(\mathbf{u}_0) \delta u^n \right)^2 + \frac{1}{8} \sum_{N=1}^S \left(\sum_i \sigma_{N,i} \partial_n \partial_m f_i(\mathbf{u}_0) \delta u^n \delta u^m \right)^2. \quad (1.32)$$

The first term is the harmonic contribution to the energy. For flat origami, we know that these correspond to the in-plane motions. On the other hand, the second term corresponds to the out-of-plane motion of the vertices. Finally, we note that the number of self-stresses S is given by the number of internal vertices V_I .

For small strains, we assume that the first term is zero so that only out-of-plane deformations can occur. Finally, we obtain an approximate energy for flat origami near the flat state as a sum of terms quartic in the vertical displacements of the vertices,

$$E_S \approx \frac{1}{8} \sum_{N=1}^{V_I} \left(\sum_{n=1}^V \sum_{m=1}^V Q_{Nnm} h_n h_m \right)^2, \quad (1.33)$$

where h_n is the height of the n^{th} vertex above the xy -plane and

$$Q_{Nnm} = \sum_i \sqrt{K_{S,i}} \sigma_{N,i} \left. \frac{\partial}{\partial h_n} \frac{\partial}{\partial h_m} \gamma_i \right|_{h_n=0}, \quad (1.34)$$

where γ_i is the strain of the i^{th} edge defined in the main text.

1.2.2.1 Estimate of the stretching energy of elastic origami

We now want to consider how to set the relative magnitudes of the torsional spring moduli in our energy. As a model of self-folding origami, we consider a trilayer

polymer system described previously [47] and shown in Fig. 1.6, in which faces are characterized by a hydrogel of thickness h_N sandwiched between two stiffer layers $h_P \ll h_N$ and active folds are induced by cutting trenches in either the top or bottom of the two stiff layers of a given width. To estimate the bending rigidity for the faces, we imagine that the bending energy arises from bending along a cylinder oriented along each fold of characteristic width W_I . Active folds, intuitively, have a W_I of the width of the cuts in the stiff layer, while we assume that the face folds have a W_I determined by the width of the vertices and thus are determined by the W_I of the active folds. For the rest of the paper, we will neglect the small changes in W_I between different folds. We will estimate the elastic moduli for our trilayer system origami based on the estimates $Y_N/Y_p \sim 5 \times 10^{-4}$ and $h_p/h_N \sim 0.04$.

We estimate the elastic energy of a face according to

$$E = \frac{1}{2} \int dA \left[\int_{-h_N/2-h_P}^{-h_N/2} dz Y_p + \int_{-h_N/2}^{h_N/2} dz Y_N + \int_{h_N/2}^{h_N/2+h_P} dz Y_p \right] \gamma^2, \quad (1.35)$$

where Y_N and Y_p are the three dimensional Young's moduli and h_N is the thickness of the hydrogel layer, h_P is the thickness of each polymer layer, and γ is a dimensionless strain.

Then assuming that γ is approximately constant across a face and assuming $Y_p h_P \gg Y_N h_N$, we obtain

$$E \approx Y_p h_p A \gamma^2. \quad (1.36)$$

For the area, A and an edge surrounded by two faces, we use the area in Fig. 1.6, which is conveniently one third the total area of the two adjoining faces. For edges on the boundary, the corresponding stretching energy is obtained from a single face. Comparing this to our spring energy,

$$E = \frac{1}{2} \kappa_S \gamma^2, \quad (1.37)$$

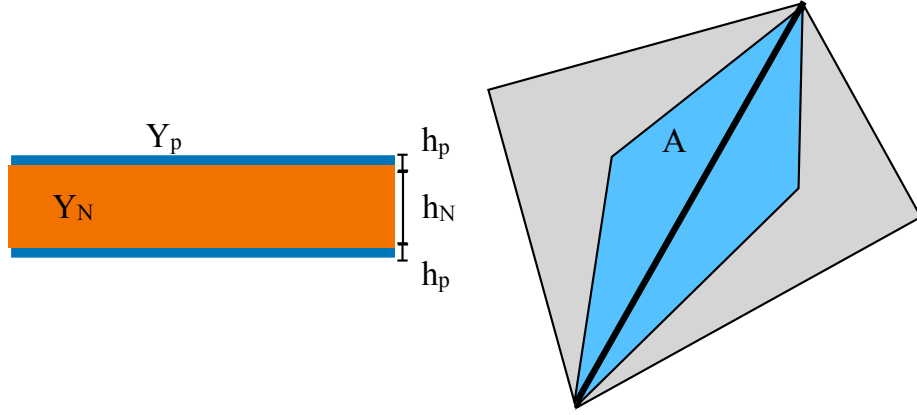


Figure 1.6. (left) a cross-section of a trilayer origami face showing the thicknesses and three-dimensional Young's moduli. (right) the area of each face adjacent to an edge that is closer to that edge than any other.

we obtain an estimate $\kappa_{S,I} \approx 2Y_p h_p A_I$ for the stretching modulus associated with edge I , where A_I is the appropriately chosen area.

1.2.2.2 The bending modulus of the folds

For an active fold, we assume the fold is bent along a width W_I to a constant curvature R , so that $R = W_I/\theta$. Therefore, the bending energy of a face can be approximately computed as

$$E_B = \frac{1}{2} \frac{Y_N}{1-\nu^2} W L \int_{-h_N/2}^{h_N/2} dz \frac{z^2}{R^2} + \frac{1}{2} \frac{Y_p}{1-\nu^2} W L \int_{-h_N/2-h_P}^{-h_N/2} dz \frac{z^2}{R^2} + \frac{1}{2} \frac{Y_p}{1-\nu^2} W L \int_{h_N/2}^{h_N/2+h_P} dz \frac{z^2}{R^2} \quad (1.38)$$

$$\approx \frac{1}{2} \theta^2 L \frac{Y_p h_N^2 h_P}{2W(1-\nu^2)} \quad (1.39)$$

Thus, $\kappa_{face,I} \approx L_I/W_I(Y_p h_N^2 h_P)/(2(1-\nu^2))$. For an active fold we obtain

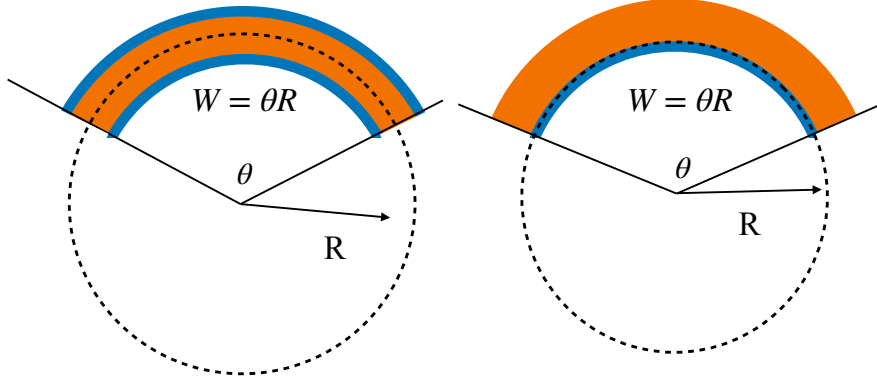


Figure 1.7. Bending a face or a fold to have constant curvature R^{-1} . The angle θ is identical to the apparent fold angle of the fold. When bending a face (left), we assume the face bends along the midsurface whereas for an active fold, we assume the surface bends along the stiffest layer.

$$E_B = \frac{1}{2} \frac{Y_N}{1 - \nu^2} W L \int_{h_P/2}^{h_N + h_P/2} dz \frac{z^2}{R^2} + \frac{1}{2} \frac{Y_p}{1 - \nu^2} W L \int_{-h_P/2}^{h_P/2} dz \frac{z^2}{R^2} \quad (1.40)$$

$$\approx \frac{1}{2} \theta^2 L \frac{Y_N h_N^3}{3W(1 - \nu^2)} \quad (1.41)$$

Thus, $\kappa_{fold,I} \approx L_I / W_I Y_N h_N^3 / (3(1 - \nu^2))$. We expect that the width of an active fold is set by the size of the cut used to create the folding face whereas the width of a fold associated with bending a face is set by the size of a vertex which is also the width of the trenches. Therefore, we assume W_I is the same for both types of folds.

1.2.2.3 Stiffness ratios

In our numerical calculations, we divide all the moduli by $2Y_p h_p A$ where A is the characteristic area. Neglecting the Poisson ratio, we use

$$\begin{aligned}
\kappa_{S,I} &\approx A_I/A \\
\kappa_{fold,I} &\approx \frac{L_I}{\ell} \left(\frac{Y_N}{Y_p} \frac{\ell}{W} \frac{h_N^3}{6h_p A} \right) \\
\kappa_{face,I} &\approx \frac{L_I}{\ell} \left(\frac{h_N^2 \ell}{4WA} \right)
\end{aligned} \tag{1.42}$$

where ℓ is the characteristic length of a fold. For $h_P \approx 0.2\mu m$, $h_N \approx 5\mu m$, $Y_N/Y_p \approx 5 \times 10^{-4}$, $W \approx 44\mu m$, $A \approx 2 \times 10^4 \mu m^2$, and $\ell \approx 260\mu m$, we obtain $\kappa_{fold,I} \approx 2 \times 10^{-5} L_I/\ell$ and $\kappa_{face,I} \approx 2 \times 10^{-3} L_I/\ell$, or $K_{fold} = 2 \times 10^{-5}$ and $K_{face} = 2 \times 10^{-3}$

For the theoretical numerics and simulations, we will nondimensionalize the spring moduli by a characteristic linear spring moduli, $\kappa_{S,c}$, letting us combine the material parameters and define $\kappa_{face,I}/\kappa_{S,c} \approx K_{face} \bar{L}_I/\ell$ and $\kappa_{fold,I}/\kappa_{S,c} \approx K_{fold} \bar{L}_I/\ell$. We will also define $\kappa_{S,I}/\kappa_{S,c} \approx K_S \bar{L}_I^2/\ell^2$. The characteristic fold length ℓ is introduced to keep K_{face} , K_{fold} , and K_S nondimensional and independent of edge.

1.2.3 The origami “bird’s foot”

Eq. (1.27) provides a means of computing and plotting energy landscapes for weakly-folded origami. We start our study of the folding and mis-folding of elastic origami with the simplest non-trivial example, the self-folding “bird’s foot” origami (Fig. 1.8A). The bird’s foot is a single origami vertex from which four folds emerge. We supplement these four folds with two additional face folds, shown as dashed lines in Fig. 1.8A. It is well known that there are two folding pathways possible which can be characterized by the relative signs of the fold angles between vertices 4 and 1, ρ_1 and ρ_3 (Fig. 1.8B).

For the rigid case, in the space of fold angles (ρ_1, \dots, ρ_4) the trajectories of the fold angles are perpendicular and can be projected conveniently to just a pair of angles as in Fig. 1.8C. In the elastic case, if we orient the bird’s foot so that vertices 1, 2 and 7 per Fig. 1.8A lie on the xy -plane, Eq. (1.27) suggests plotting the energy landscape

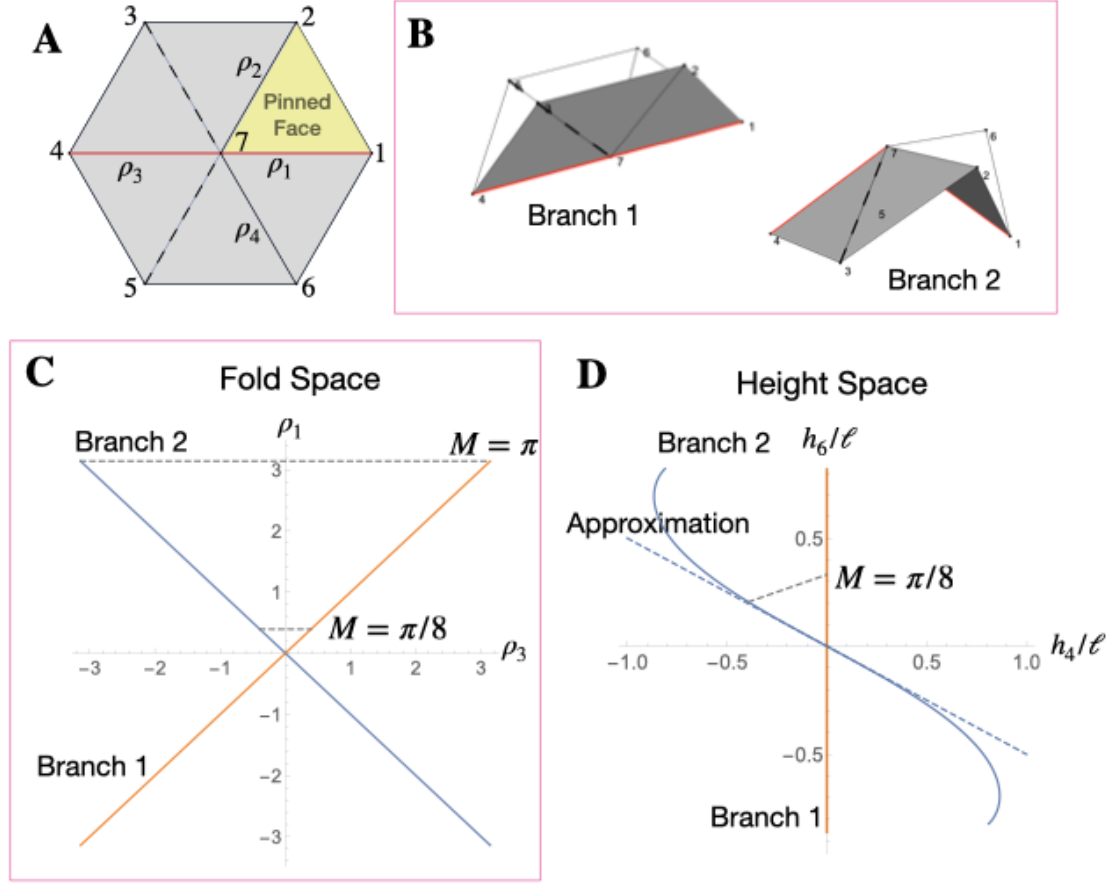


Figure 1.8. Schema, configuration spaces, and energy landscapes for the birdsfoot origami. (A) A schematic of the birdsfoot. The folds are marked in solid lines, while the folds added in the model to imitate face bending are marked with dashed lines. The folds used to define fold space are highlighted in red. The face we have “pinned” to a plane is highlighted in yellow, then the heights of vertices 4 and 6 above this plane defines the height space. (B) The configurations that correspond to each branch. (C-D) The branches for a rigid origami in both fold and height space, respectively. In (D), the heights are non-dimensionalized using the characteristic fold length ℓ . The dashed lines show the linearized trajectories between the branches at two magnitudes. The height space projection of the trajectory takes advantage of the linearity between height and fold space at small heights. In (C), note that the branches in fold space are perpendicular. (D) also shows the shape of branch 2 in the approximate energy used to draw the energy landscapes in Fig. 2.

near the unfolded state in terms of the heights of the remaining four vertices above the plane, $\mathbf{h} = (h_3, h_4, h_5, h_6)$ (Fig. 1.8D) rather than the fold angles.

We program target angles according to

$$\vec{\rho} = (1 - A)M\vec{\rho}_{B1} + AM\vec{\rho}_{B2}, \quad (1.43)$$

where $\vec{\rho}_{B1} = (-1, 0, -1, 0)$ and $\vec{\rho}_{B2} = (-1, 1, 1, 1)$ are the fold angles of each branch when folded flat, and M ranges from 0 to π and controls the degree of folding. The parameter A , which lies between 0 and 1, tunes the target angles between the two branches accessible by rigid origami. For values of A other than 0 and 1, the target angles lie between the two branches. It should be noted that $A = 0.5$ is not precisely between the two branches geometrically, and the geometric center, though dependent on the precise value of M , is closer to $A \approx 0.425$. Fig. 1D shows this trajectory in height space for $M = \pi/8$ by taking advantage of the linear relationship between folds and heights to quadratic order [41].

Since h_3 and h_5 are the heights of vertices associated with face folds, in order to plot the energy landscapes, we numerically minimize $E(h_3, h_4, h_5, h_6)$ with respect to h_3 and h_5 to express the energy in terms of only (h_4, h_6) . Contours of the energy obtained this way are shown in Fig. 1.9 for various values of A and for $M = \pi/8$. The minima of the energy are depicted as closed white circles and saddle points are shown in red, while the target point is denoted by an open white circle.

All of the theoretical figures in this paper were created using a package developed for creating and manipulating origami structures and other similar mechanisms in Mathematica. This package is located on GitHub at <https://github.com/cdsantan/mechanisms>. Mathematica notebooks for each figure and the associated data are also located on GitHub at <https://github.com/meleetricimble/robust-folding-paper-support>.

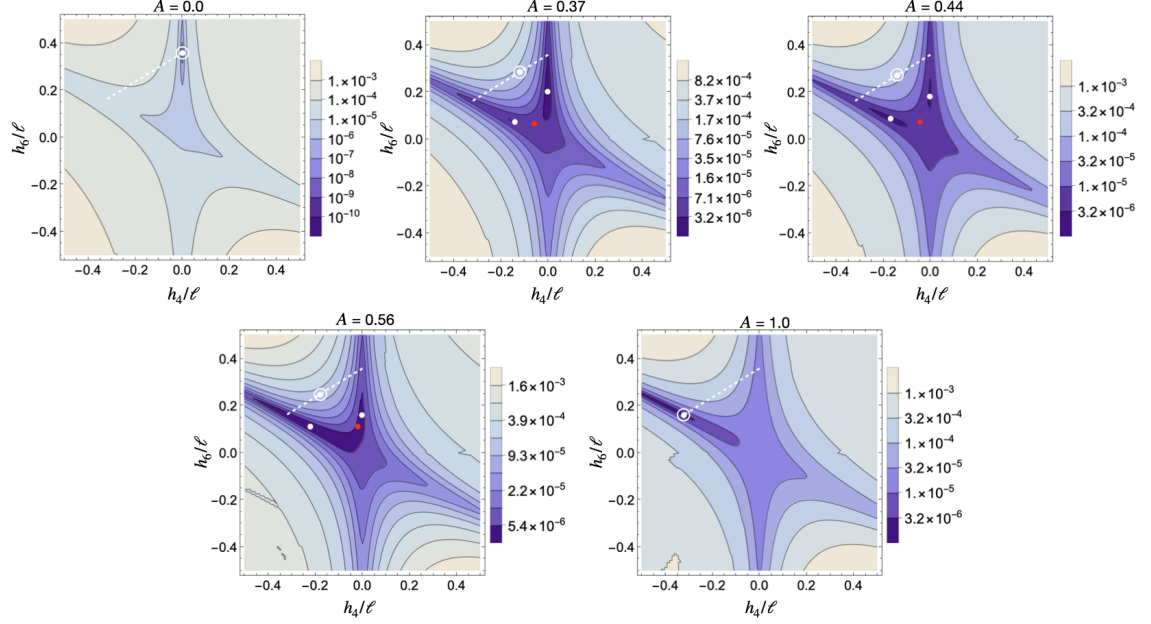


Figure 1.9. Energy landscapes in height space for the bird's foot at different values of the control parameter A , for the small magnitude trajectory between branches shown in Fig. 1.8(C-D), at $K_{face} = 10^{-2}$ and $K_{fold} = 10^{-4}$. The white dots represent local minima and the red dots represent saddle points. The dashed line represents the projection of the trajectory between branches into height space (as in Fig. 1.8(D)), with the circle denoting the location on the trajectory of the landscape. Notice that as A increases, the original minimum moves toward the flat state. Between $A = 0$ and $A = 0.37$, a saddle point and the minimum for branch 2 are created, then after $A = 0.56$ the saddle point and the minimum for branch 1 annihilate each other. The created minimum also moves out away from the flat state as A increases. Note that the contours and color scheme are on a log scale and inconsistent between landscapes to emphasize features.

As seen in Fig. 1.9, which shows the energy landscapes of the birds foot with contours on a log-scale, the configuration space of the rigid origami lies along the bottom of steep valleys defined by Eq. (1.27). Because the torsional springs are weaker than the stretching springs, as A changes from 0 to 1 at fixed $M = 1/8$, the energy minimum on the first branch moves inward along the energy valley. At a critical value of $A > 0$, a new minimum and saddle point nucleate near the flat state and as the new minimum moves outward along the other branch, the old minimum eventually approaches and annihilates with the saddle point.

As the stretching energy is increased, the energy valleys become steeper but the shape of the energy landscape near the flat state remains the same. As K_S increases and we approach the rigid limit, the critical A at which a new minimum forms decreases. Yet for any finite value of K_S , the energy landscape is monostable near $A \approx 0$ and $A \approx 1$.

1.2.3.1 Phase Diagrams

We can determine the size of the region of bistability for different values of K_{face} , K_{fold} , and prescribed fold angle using the full elastic energy. To do so, we start at one end of the linearization we have introduced and find the minimum at that point. Then we increase the linearization parameter A by one step, and repeat the minimization using the minimum just found as the initial position. We continue taking the next step in the parameterization, using the previous minimum, and minimizing until the other branch has been reached. To see where both minima are present, we repeat this process but instead start from the opposite branch and follow the parameterization backwards. The regime in which both minima are present across both directions is the bi-stable regime.

To get the full idea of the bistable regime between the two branches, we repeat the method described above for different values of M , which represents the magnitude of

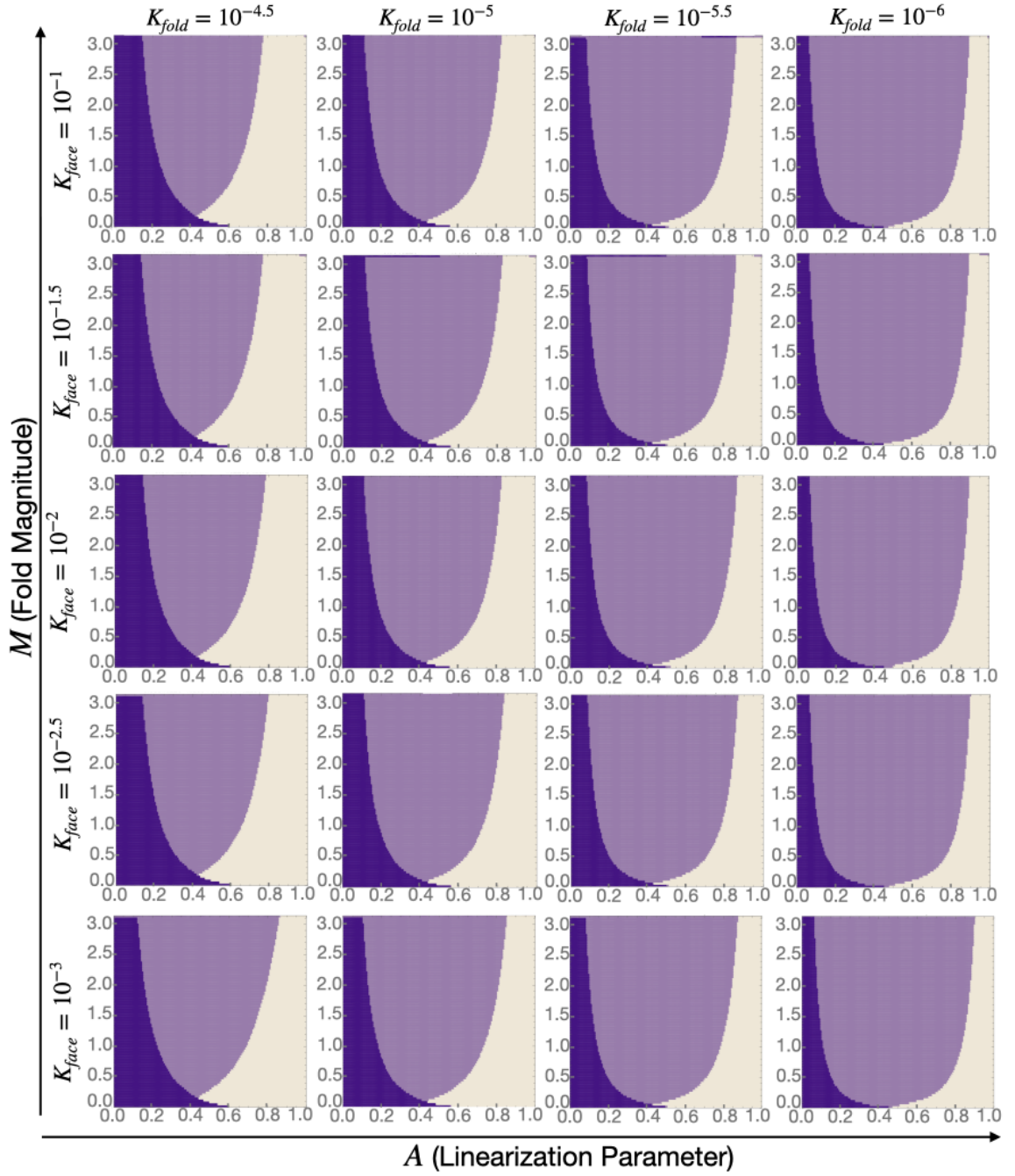


Figure 1.10. Slices of a four-dimensional phase diagram defined by K_{fold} , K_{face} , control parameter A , and target fold angle magnitude M at four different values of K_{face} and K_{fold} . The regions in light purple represent the region of bi-stability, where both minima are present. The purple and tan represent the regions where only the branch 1 minimum and branch 2 minimum are present, respectively.

folding, to draw the phase diagram between the two branches as a function of M and A for given elastic moduli.

In Fig. 1.10, we show the region of bistability as a function of M and A for four different values of K_{fold} and K_{face} . The plots are asymmetric and, in particular, shifted toward values of $A < 0.5$. This is consistent with the midpoint between both branches being at $A \approx 0.425$ rather than $A = 0.5$. Overall, we see two separate trends: decreasing K_{fold} widens the region of bistability with more widening seen at lower values of M , while more surprisingly decreasing K_{face} also widens the region of bistability but with more growth seen at higher values of M .

These results implicate the balance of in-plane stretching and torsional spring moduli in governing bistability. In particular, when K_{fold} is small, indicating that the system is approaching the inextensible limit, we see that even a small error in programming the fold angles can lead to multistability. This is, in fact, entirely consistent with Ref. [39] which argues that for rigid origami, for which $K_{fold}/K_S \rightarrow 0$, a metastable minimum exists unless the vector with components $\kappa_{B,I}\bar{\rho}_I$ is perpendicular to a branch.

It is important to note that the change in bistability occurs even though $K_{fold} \ll K_S$, indicating that in-plane strains are still small and Eq. (1.27) remains a reasonable approximation. Indeed, in our simulations the energy from stretching is typically 1% of the total energy. This is also consistent with the energy landscapes in Fig. 1.10, which show that the bistability arises from the nucleation of additional minima near the flat state and not far out along a branch even when M is large, precisely where we expect our theoretical analysis of elastic origami to be most accurate.

As an additional check, we also did a limited run of bistability diagrams with affine elastic triangles for faces rather than our spring model.

The in-plane elastic energy for a Hookean, isotropic two-dimensional solid can be written as

$$E_{el} = \frac{1}{2}\lambda \left(\sum_i \gamma_{ii} \right)^2 + \mu \sum_{ij} \gamma_{ij}^2, \quad (1.44)$$

where $\gamma_{ij} = \partial_i u_j + \partial_j u_i$ and λ and μ are the Lamè coefficients. We assume that each triangular face has an energy of the form of Eq. (1.44). The in-plane elastic deformations u_i are determined by assuming the face has deformed affinely. For a triangular face on the xy -plane, this uniquely determines the displacement and allows us to estimate the elastic energy of arbitrarily deformed triangular faces.

To compare this to our linear spring edge model, we use the same method for plotting the phase diagrams for the birdsfoot as in the main body of the paper but with the above energy. Fig. 1.11 shows a side-by-side comparison of the resulting plots generated by the two different energies at the experimental values for default and softened faces. Minimizing the more complicated elastic polygon energy is more computationally costly, so the grid size has been reduced and grid squares with black lines denote points that failed to converge, with their color assigned based on neighboring squares. The agreement between models overall is good, even without fitting parameters.

1.2.3.2 Experimental Methods

We next turn to a discussion of self-folding in a trilayer, thermo-responsive system [47], adapted from our previous report [51]. In brief, self-folding origami was prepared by using a bilayer bending mechanism of polymer films. P(*p*MS-BP-RhB) (poly(*p*-methylstyrene-benzophenone-rhodamine B) and P(DEAM-AA-BP-RhB) (poly(diethylacrylamide-acrylic acid-benzophenone-rhodamine B) were used as a stiff layer and a thermosensitive hydrogel layer with a lower critical solution temperature at around 30°C, respectively. Pendant groups of benzophenone contained in both pre-synthesized co-polymers were utilized as a photoreactive cross-linker for multi-layer patterning. First, the bottom stiff layer was deposited by spin coating of toluene

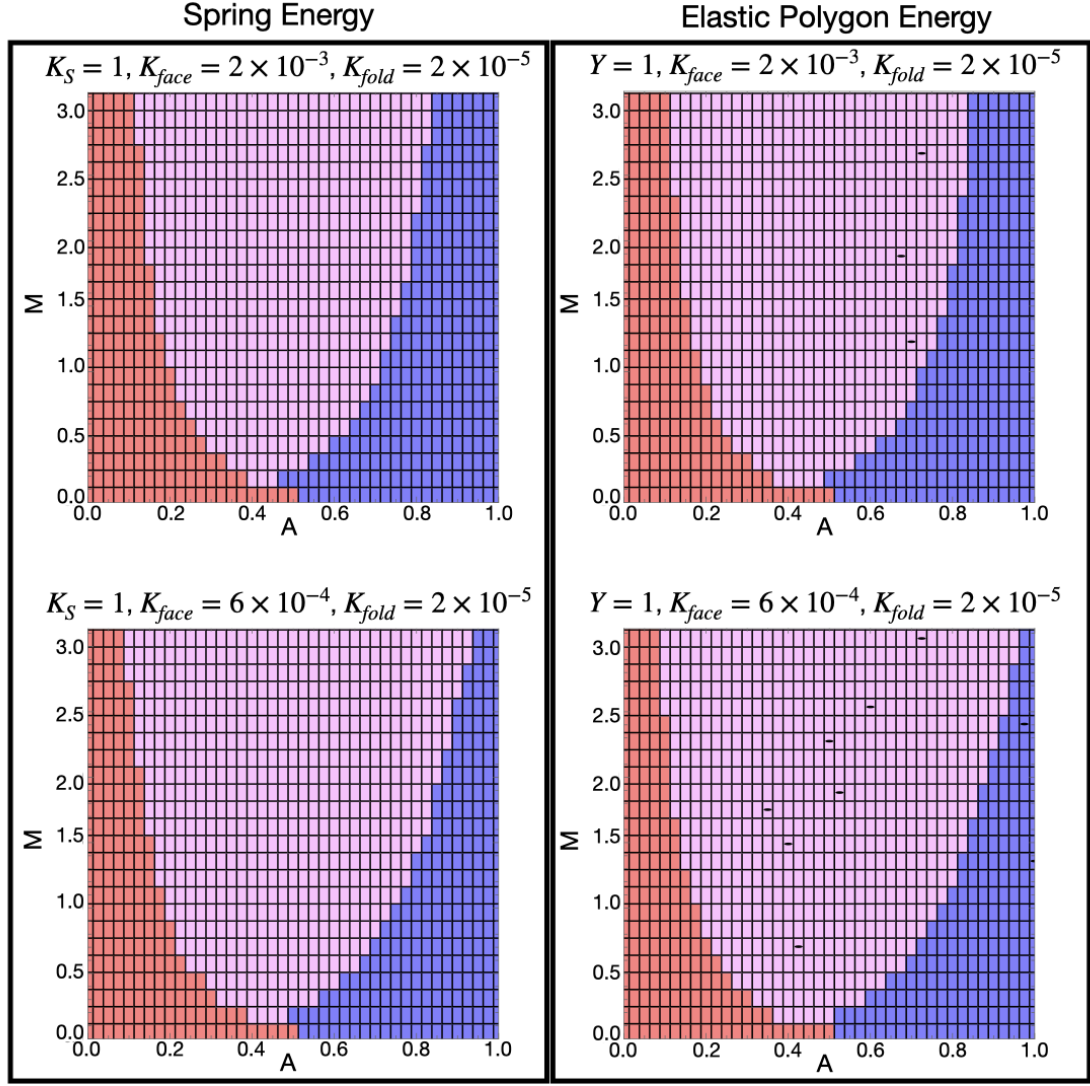


Figure 1.11. Phase diagrams for the birdsfoot with axes of the magnitude of the target angles, M , and the control parameter A , as described in the main body of the paper. The blue and red regions represent only one branch appearing while the center pink region represents the bistable region. The left column uses the linear spring edge model for face stretching while the right column uses the elastic polygon model. The top row has torsion spring constants corresponding to experiment with default faces while the bottom row corresponds to softened faces.

solution of P(pMS-BP-RhB) on a silicon wafer with a water-soluble sacrificial layer of poly(vinyl alcohol) (PVA, Aldrich).

To create a microscale crease pattern, UV-light (365 nm, pE-100, CoolLED) was projected on the layer of P(pMS-BP-RhB) by an inverted optical microscope (Nikon Eclipse Ti, 10x objective lens) equipped with a digital micromirror device (DMD). Pixelated UV illumination for each layer of birdsfoot pattern was obtained by the Mathematica notebook provided from Robert J. Lang (Tessellatica 11.1d7)[66] based on the folding angle calibration at a fixed temperature, 20°C. After cured, a typical development process was followed by stripping uncured area of the film with a marginal solvent (e.g., mixture of toluene and hexane with 1:3 vol%). Next, a few-micron thick hydrogel layer was deposited on the sample pattern by casting a chlorobenzene polymer solution and slowly drying in the dark chamber. Patterned UV curing with computer-controlled alignment was then followed for crosslinking of the mid layer on top of the bottom layer. Finally, another thin layer of P(*p*MS-BP-RhB) was photo-patterned as a top stiff layer by using the same procedure as the bottom layer. For folds with a target angle of zero, a series of square holes was additionally applied to all three layers as a perforated line between the vertices of the crease pattern, as shown in Fig. 1.12A. Because the perforations align on both sides, the resulting face folds have a target angle of 0 but are stiffer than the active folds represented by slits.

The resultant trilayer origami was fully dried before further use. To release the origami as flat from the substrate, the sample was dipped in the pre-heated buffer solution (pH 7.0 PBS, 60°C). After full dissolution of PVA layer, the water bath was cooled down to induce programmed folding of the crease patterns, which was observed by using the optical microscope (Zeiss AxioTech Vario, with 2.5x objective lens).

We can now have a small amount of control over K_{face} by utilizing the perforated 0 angle folds explained above. Perforating the faces decreases the amount of stiff

trilayer by a factor of 3 to 4, and since the stiff layer provides the majority of the bending modulus we expect K_{fold}/K_{face} to decrease by the same factor.

We created batches of 10 bird's foot origami both with and without perforated faces for several values of A , corresponding to different target fold angles and controlled by the width of the cuts in the stiff layer, between the two branches. Fig. 1.12B shows the fraction of bird's foot samples that folded to branch $\vec{\rho}_{B2}$ with non-perforated samples (circles) and perforated samples (squares). In the non-perforated samples, we see a sharp transition between branch 1 at small A and branch 2 at large A , with a small region of values near $A \approx 0.5$ that show some bistability. In the perforated samples we see this bistable region widen, with both states observed in the $A \approx 0.33$ samples.

Some care must be taken in interpreting the results quantitatively. Because the experimental system folds slowly, we expect the number of minima to be governed to some degree by the small M portion of Fig. 1.10, even when the programmed fold angles are large, since we expect that a structure that has found a stable configuration will tend to remain in that configuration as it folds. In addition, failure to see misfolded states does not indicate that those states do not exist; in contrast, even a small number of misfolds indicates metastability. Finally, the programmed fold angles are controlled by the width of the long cuts in either the top or bottom rigid layers. The cuts that lead to folding then also affect the torsional stiffness of the folds. This effect is negligible for most folds, except for those with zero fold angle (those remaining flat), which must be cut on both top and bottom surfaces. This leads to folds that are weaker than active folds, as is the case at the $A = 0.5$ point. This point does, however, highlight that using target angle tuning to avoid misfolding can be more complicated to realize in experiment than in theory.

To compare our experimental and theoretical results, we use the estimates $K_S \approx 1$ on the folds and boundary edges, $K_{face} \approx 2 \times 10^{-3}$ and $K_{fold} \approx 2 \times 10^{-5}$ (see SI), and

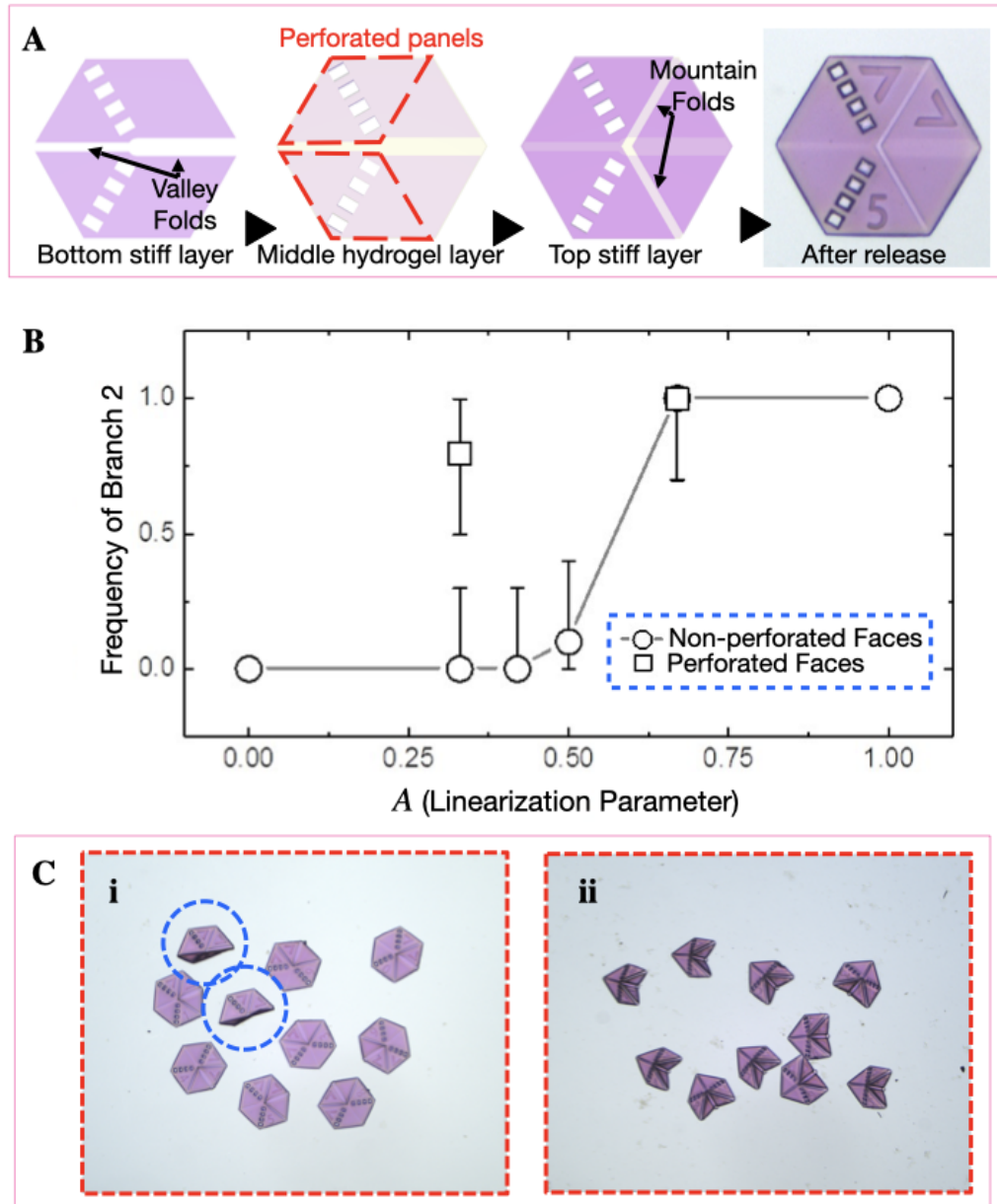


Figure 1.12. Experimental schema and results for folding the birdsfoot with and without weakened faces. (A) Schematic of the tri-layer origami structure with perforated faces. (B) The percentage of samples folded to the second branch for both non-perforated and perforated faces. Each point corresponds to 10 samples. Error bars are from the rule of three. (C) The folded samples with perforated faces at control parameters (i) $A = 0.33$ and (ii) $A = 0.67$. At $A = 0.33$ (i), you can see the two samples that folded to branch 1 while the rest shallowly folded to branch 2.

use the factor stated above when faces are weakened ($K_{face} \approx 6 \times 10^{-4}$). According to the relevant region of plots in Fig. 3, perforating the faces should widen the bistable region by weakening the face folds and this effect is seen quite prominently in Fig. 1.12B, as well as the bistable region occurring for smaller A . In Fig. 1.12C, we show a representative batch of 10 origami structures. It is also notable that the misfolded configurations in 1.12C are quite shallow, as we expect from our theoretical analysis. Though the experiments are in qualitative agreement with our theoretical model, the effect of softening the torsional moduli of the faces affects the stability of experimental bird foot origami rather dramatically whereas the theory shows more subtle effects. The origin of this discrepancy remains unclear.

1.2.4 Folding complex origami

Finally, we turn to a more complex fold pattern, the “Randlett bird” [67], (Fig. 1.14), which we have previously explored with the trilayer, self-folding origami system [47]. Here, we use the same programmed fold angles from Ref. [51] for both the experiment and the simulations. They can be seen in Fig. 1.13. .

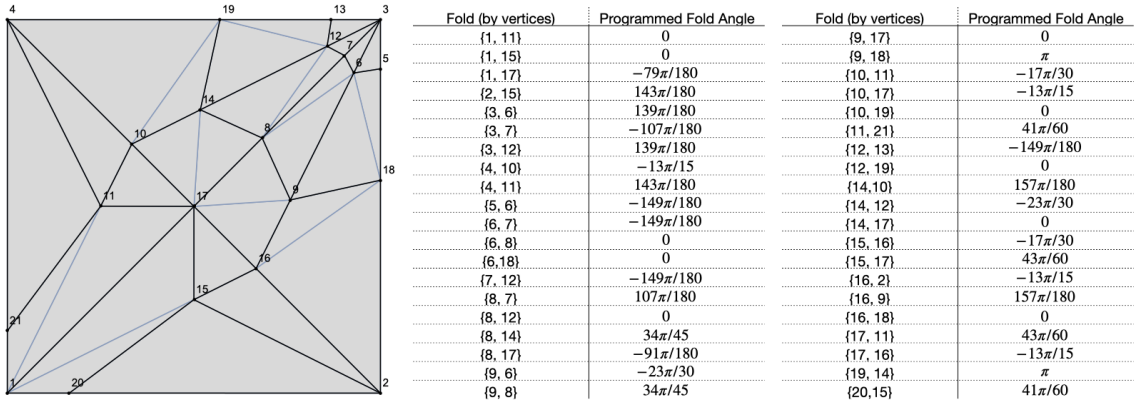


Figure 1.13. (left) the Randlett bird with true folds in black and added face folds in lighter blue with vertices numbered. (right) the programmed fold angles used in the simulations. Folds are denoted by their end vertices.

We previously reported that self-folding trilayer Randlett birds misfold at a rate of 0.55 ± 0.15 [51]. Some examples of both correctly and incorrectly folded birds can be seen in Fig. 1.14.

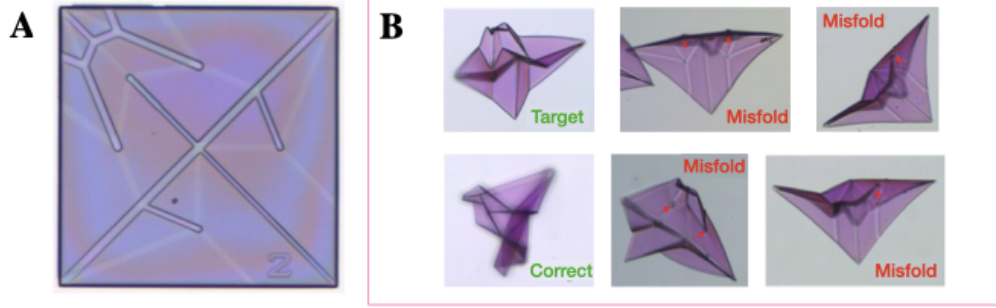


Figure 1.14. (A) An optical image of the experiment before folding. (B) Some examples of folded and mis-folded structures. The experiment folds correctly at a rate of 0.45 ± 0.15 .

Unlike the bird’s foot origami, the Randlett bird is not foldable without bending faces. If we introduce face folds across the shortest diagonal of the faces, however, we expect the Randlett bird to have 2048 branches each with 6 degrees of freedom (as predicted by formulas in Ref. [41]). The high dimensionality of this enlarged configuration space makes direct visualization of the energy landscape impossible. Instead, we will apply a statistical analysis to the folded minima.

We first initialize the Randlett bird in the folded configuration according to vertices provided by Ref. [38] (described in [68]) and attempt to numerically minimize using the BFGS algorithm [69]. For $K_{face} \geq 10K_{fold}$, this direct numerical minimization of the pre-folded state fails. The gray regions in Fig. 1.15 represents this region. To avoid complications when counting minima, we only use values of K_{face} and K_{fold} for which this minimization produces a reliable energy minimum. Note that, for the trilayer origami system, the expected stiffnesses, $K_{face} \approx 2 \times 10^{-3}$ and $K_{fold} \approx 2 \times 10^{-5}$, are within this region. A Mathematica notebook for this subfig-

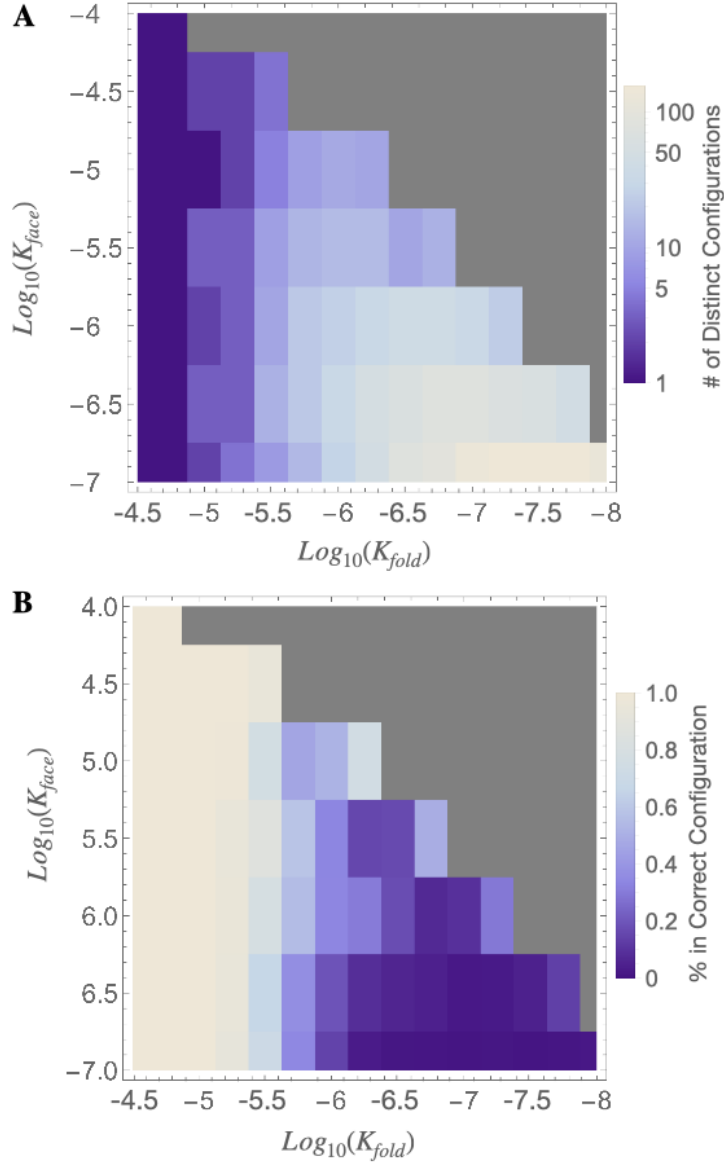


Figure 1.15. (A) A density plot of the number of different configurations seen from near the flat state for given K_{fold}/K_S and K_{face}/K_S . Each point represents a minimum of either 500 simulated folds or 10 times the number of different states observed, whichever was larger. The number of configurations reflects whether the origami is mono- or multistable, but may not predict the precise number of possible fold configurations for the given parameters. Note that the color scale is a log scale to emphasize features at lower configuration numbers. The gray region represents the region in which the pre-folded, initial numerical minimization fails. (B) A density plot showing the percentage of the simulated birds that fold into the target state. This does not represent a prediction of experiment, rather that the basin of attraction for the correct state is larger when the degree of multistability is less.

ure is located on GitHub at <https://github.com/meleetricimble/robust-folding-paper-support>.

For a given set of K_{face} and K_{fold} , we start by generating a sample of 300 randomly perturbed Randlett birds at a given set of K_{face} and K_{fold} by moving each vertex of the bird out of its flat starting position by a normal distribution with a width of 50% of the shortest fold in the origami. We then minimize each bird’s energy and discard any results that fail to find a minimum to within a target accuracy goal. We continue to generate further samples until we reach a total of the larger of 500 successful minimizations or 10 times the number of distinct minima found. We then identify distinct folded states by first determining the optimal alignment by a least-squares minimization of the distance between corresponding vertices of a pair of birds with respect to Euclidean motions, then determining whether all corresponding vertices are closer to each other than a threshold value. This threshold value is chosen so that the number of distinct minima does not change when the threshold value is changed.

Finally, we count the number of distinct states, each representing a mechanically stable state. While there is no way to guarantee that this procedure finds every metastable state, we expect the relative number of energy minima found to scale with the actual number of metastable minima. We then also extract the percentage of samples folded to the target state. We perturbed the simulated samples from the flat state using a normal distribution, so the initial birds represent a uniform cloud of initial states in position space. Thus, this percentage does not represent the folding rate of experiment, but rather the relative size of the basin of attraction for the target minima.

Fig. 1.15A shows the resulting number of minima we find as a function of K_{face} and K_{fold} on a log scale to emphasize the points that have only a single minima. Fig. 1.15B shows the percentage of samples folded to the target minimum for the same data. In both plots, each point represents at least the larger of 500 birds or 10 times

the number of distinct states seen. The two plots together show that a lesser degree of multistability leads to the basin of attraction for the correct minimum increasing. This implies that there is a relationship between the number of minima and the robustness of the folding origami.

Overall, we see the same effect for the Randlett bird that we saw for the bird’s foot: multistability increases with both decreasing K_{fold} and decreasing K_{face} . The method to arrive at this result for the Randlett bird can be generalized for any origami, and we would expect the same general result.

1.2.5 Discussion

We have introduced a simple model to study self-folding origami that accounts for the finite elasticity of the origami. With finite elasticity, a more complicated picture of the energy landscapes and folding of these structures arises than in rigid origami. Though the energy landscape is characterized by deep valleys along the configuration space of the rigid structures (so that strains while folding are still typically small) we find that the number of energy minima changes with the elastic moduli of the folds through a series of bifurcations near the flat configuration. Because these bifurcations occur near the flat configuration, where finite elasticity dominates the shape of the energy landscape, they are not well-captured by analyses of rigid origami.

We demonstrated two methods for using this model to examine the stability of origami for different stretching and bending parameters: first one that can be applied to simple origami with low-dimensional configuration spaces that can be easily represented, and a second method that can be applied to much more complicated origami. In both cases, we saw that weakening both faces and folds results in an increase in the degree of multistability of the structures. In other words, thicker, elastic origami self-folds better than idealized origami with infinitely stiff faces and floppy folds.

Both Tachi and Hull[39] and Stern *et al.* [52] proposed methods for avoiding misfolding that utilize tuning the target fold angles and fold stiffnesses to avoid misfolding. Both methods require a more precise fine-tuning of fold stiffnesses and angles that are often difficult to achieve in many experimental platforms. Tuning the in-plane and out-of-plane stiffnesses of the faces themselves, either by weakening as suggested here or stiffening by adding additional layers, is an additional simple tool to avoid misfolding even when geometric constraints are still dominant.

CHAPTER 2

OTHER ORIGAMI PROBLEMS

In this chapter, we will briefly discuss two other origami problems. First, we will discuss self-actuating origami with non-standard forcing mechanisms that result in torques on faces rather than torques about the folds and present a set of tools for designing these mechanisms. Second, we will introduce minimal forcing sets, or the minimum set of folds required to be assigned to fold to cause the entire origami to fold. By first finding the minimum number of folds needed to fold an arbitrary single vertex, we can then minimize the total number of assigned folds across the entire origami. Using tools from graph theory, we will do so and set an effective upper bound on the size of minimal forcing sets.

2.1 Face-Forced Origami

Most self-actuated origami is driven at the fold - ranging from trilayer origami, like the swelling gel origami discussed in detail last chapter, where the incompatibilities between the layers is forced into the folds [47, 51, 70], to capillary origami where the folds are the capillaries[71], to more robotics-inspired origami where mechanical actuators are placed on the folds[72, 73]. As such, almost all modeling of origami structures have emphasis placed on the folds as actuators, from energetically designed spring models like our own to models that focus on forces or torques about the folds [15, 39, 42, 52]. More recently, there have been an emerging class of origami where torques are exerted on the faces. Zhang et. al.[74], for example, achieve self-folding by placing magnetic dipoles on the faces and then applying a magnetic field. The

dipoles then attempt to align with the magnetic field, exerting a torque from where they are placed on the faces. This presents a new challenge - how do we determine how these structures fold? What angles will they fold to?

In this section, we will present a method for predicting how a structure will fold under torques on the faces. Assuming rigid origami, we will utilize the geometry of the origami to develop a framework for relating torques on faces to forces on vertices, which can then be used to determine how the structure folds. We will also provide some additional tools for realizing physical versions, including how to avoid face bending and achieve experimentally realizable torques.

2.1.1 Connectivity Matrix

Consider the origami in the flat state. As we discussed in Section 1.1.1, near the flat state the folding problem can be reduced to quadratic order in the heights. This means that only forces in the direction out of the initial plane of the origami will result in folding motions, so we will consider only the in-plane components of the torques. For clarity, let's consider the initial plane of the origami to be the x-y plane and the z-direction to be out of plane.

Now let's take advantage of the geometry of the origami, and work backwards from the torques to get the forces on the vertices, from which we can eventually get fold angles. Starting with triangular faces, we are assuming we have a known torque in the x-y plane and a set of three unknown forces in the z direction, one for each vertex. Then, choose a coordinate system such that $(0,0)$ lies inside the triangle at the center of mass, as seen in Fig. 2.1. This gives a geometric relationship between the torque and forces,

$$\vec{\tau}_{face} = (y_1 f_1 + y_2 f_2 + y_3 f_3)\hat{x} + (x_1 f_1 + x_2 f_2 + x_3 f_3)\hat{y}. \quad (2.1)$$

Repeating this process for every face - assigning each face its own center-of-mass-centered coordinate system and finding this relationship - leads to a natural way to write the relationship between the torques and forces,

$$\vec{\tau} = \mathbf{C}\vec{f}, \quad (2.2)$$

where in order to maintain matrix notation $\vec{\tau} = \{\tau_{1,x}, \tau_{1,y}, \dots, \tau_{F,x}, \tau_{F,y}\}$ where F is the number of faces and $\vec{f} = \{f_{1,z}, \dots, f_{V,z}\}$ where V is the number of vertices. We will then call the matrix \mathbf{C} the face-vertex connectivity matrix. Note that \mathbf{C} has dimension $2F \times V$.

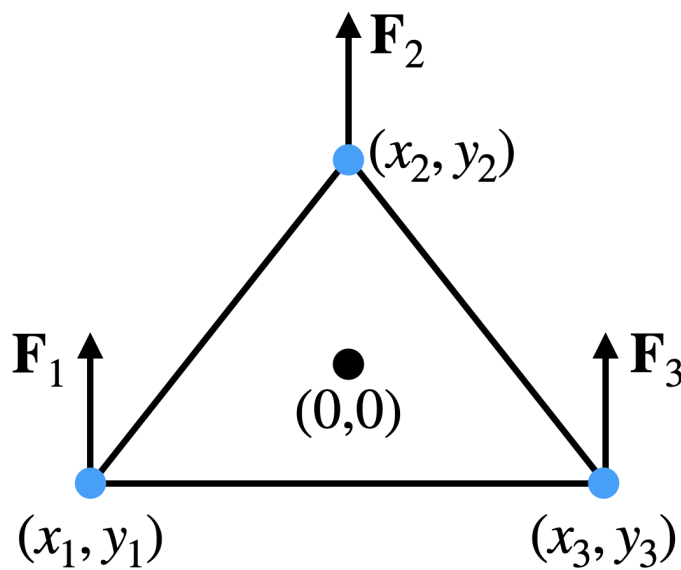


Figure 2.1. A sample coordinate system for relating the torques to the forces for a triangular face. If the face is in the plane of the page, the forces are perpendicular to the page.

While we have used a triangular face as an example, this method is agnostic to face geometry. We can utilize computational methods for finding the center of mass of arbitrary faces, such as the Mathematica function `RegionCentroid[]` which

finds the centroid of an arbitrary shape by integrating over the specified region and normalizing by the region's measure. We can then use these centroids to construct **C**. Code and documentation for finding the face-vertex connectivity matrix can be found in Appendix A.

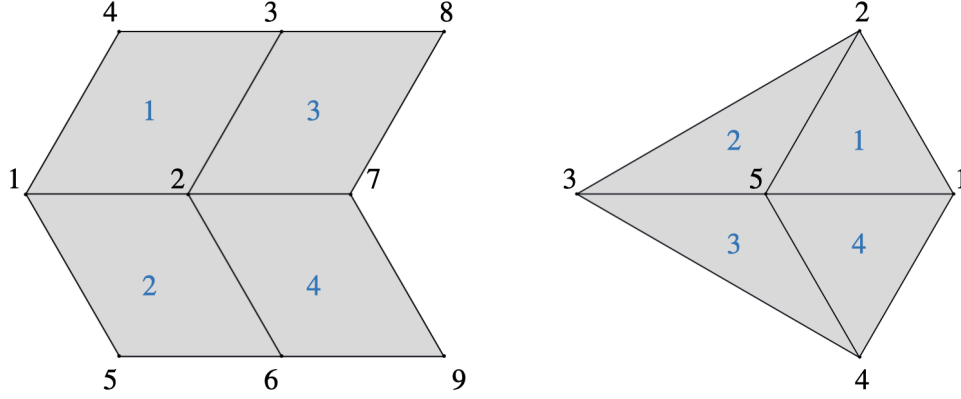


Figure 2.2. Two different versions of the birdsfoot origami, one with parallelogram faces and one with triangulated faces, with vertices labeled in black and faces labeled in blue.

As examples, we will write the face-vertex connectivity matrices for the birdsfoot in the case of parallelogram and triangular faces. The parallelogram case yields the matrix

$$\begin{pmatrix} -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{12}(3+\sqrt{3}) & \frac{1}{12}(-3+\sqrt{3}) & \frac{1}{12}(-3-\sqrt{3}) & \frac{1}{12}(3-\sqrt{3}) & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{12}(3+\sqrt{3}) & \frac{1}{12}(-3+\sqrt{3}) & 0 & 0 & \frac{1}{12}(-3-\sqrt{3}) & \frac{1}{12}(3-\sqrt{3}) & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{12}(3+\sqrt{3}) & \frac{1}{12}(-3+\sqrt{3}) & 0 & 0 & 0 & \frac{1}{12}(-3-\sqrt{3}) & \frac{1}{12}(3-\sqrt{3}) & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & \frac{1}{12}(3+\sqrt{3}) & 0 & 0 & 0 & \frac{1}{12}(-3+\sqrt{3}) & \frac{1}{12}(-3-\sqrt{3}) & 0 & \frac{1}{12}(3-\sqrt{3}) \end{pmatrix} \quad (2.3)$$

while the triangular case yields

$$\begin{pmatrix} -\frac{1}{2\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 & 0 & -\frac{1}{2\sqrt{3}} \\ -\frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{2\sqrt{3}} & 0 & -\frac{1}{2\sqrt{3}} \\ 0 & -\frac{2}{3} & \frac{5}{6} & 0 & -\frac{1}{6} \\ 0 & 0 & \frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{2\sqrt{3}} \\ 0 & 0 & \frac{5}{6} & -\frac{2}{3} & -\frac{1}{6} \\ \frac{1}{2\sqrt{3}} & 0 & 0 & -\frac{1}{\sqrt{3}} & \frac{1}{2\sqrt{3}} \\ -\frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}. \quad (2.4)$$

Note the relative dimensions and overall structure. We could have constructed these matrices in two separate parts, one a traditional connectivity matrix as seen in graph theory with only 0 and 1 entries and a second matrix that accounted for the exact geometry.

Before we move on to using this matrix to find torques, forces, or design origami we need to account for the non-triangulated faces. Triangular faces are rigid under forces on the vertices in 3D (i.e. any applied forces will move or rotate the triangle but never cause it to bend or flex) while faces with more edges are not. The face coordinate system introduced here lends itself to a new way to understand isometries in nontriangulated origami. First, we find the allowed motions of each face in the face-centered coordinates. Then we find the null space of a matrix constructed from these allowed motions, resulting in the force vectors that would bend each non-triangular face. We can then use these vectors to construct a projection matrix that takes a force vector and projects it into only allowed motions.

2.1.2 Disallowing Bending in the Vertex Forces

From the flat state, where we define the x-y plane to be the initial plane of the origami, we can reduce the allowed motion of an origami to each face either being able to translate in the z-direction or rotate around the x or y axis. We can write the translation vector for the i th face as

$$\vec{T}_i = \sum_n B_{i,n} e_n \quad (2.5)$$

where $B_{i,n}$ is a matrix that encodes the connectivity of the faces, i , to the vertices, n , taking the value 1 if n bounds i and 0 if not. The sum and unit vector e_n represent that the vector is in the space of forces in the z-direction on the vertices.

Then the face coordinates can be used to find the two rotation vectors,

$$\vec{R}_{x,i} = \sum_n B_{i,n}(\hat{x} \times \vec{r}_{i,n})e_n \quad (2.6)$$

$$\vec{R}_{y,i} = \sum_n B_{i,n}(\hat{y} \times \vec{r}_{i,n})e_n \quad (2.7)$$

where \times is a cross product, $\vec{r}_{i,n}$ represents the position vector to the n th vertex from the center of the i th face in that faces' coordinates, and the sum again represents that the vectors are in the space of forces on the vertices with zero components when the vertex is not connected to the face.

We can now use these allowed motions to construct a matrix for each face. The nullspaces of these matrices then represent the force vectors that would result in the faces bending. By the rank-nullity theorem, and since the three vectors are linearly independent as defined, a triangular face has no force vectors that correspond to the face bending while a quadrilateral face has one, a five-sided face has two, etc. To return to our birdsfoot example in Fig. 2.2, the triangular faced example has no bending forces while following this procedure for the quadrilateral example returns the vectors $(-1, 1, -1, 1, 0, 0, 0, 0, 0)^T$, $(1, -1, 0, 0, -1, 1, 0, 0, 0)^T$, $(0, 1, -1, 0, 0, 0, -1, 1, 0)^T$, and $(0, 1, 0, 0, 0, -1, -1, 0, 1)^T$.

To use this information in designing or predicting origami, we can construct a projection operator that when acting on a force vector will remove the components that would result in face bending,

$$\mathbf{P}_B = \mathbf{I}_V - M_B(M_B^T M_B)^{-1} M_B^T \quad (2.8)$$

where \mathbf{I}_V is the identity matrix of dimension $V \times V$ where V is the number of vertices and M_B is the basis matrix constructed with the calculated bending force vectors as its columns. By construction the basis is linearly independent but not normalized, but this definition for the projection operator accounts for this. We could have orthogonalized our basis, but keeping it in this form preserves the physical interpretation of the components as forces on the vertices. The code for this process for arbitrary origami can be found in Appendix A.

Now that we have all of these pieces, we can address designing and predicting the behavior of face-forced origami.

2.1.3 Predicting and Designing Face-Forced Origami

Let's start with the case where we know the torques and want to predict how the origami will fold. Since we are already using computational methods, the easiest process is to use a psuedo inverse to solve the system of linear equations in Eq. 2.2. At least one psuedo inverse exists for every rectangular matrix, and different types of pseudo inverses have different properties. For the scope of this problem, we will focus on Moore-Penrose inverses, which satisfy the following properties,

$$AA^+A = A \tag{2.9}$$

$$A^+AA^+ = A^+ \tag{2.10}$$

$$(AA^+)^* = AA^+ \tag{2.11}$$

$$(A^+A)^* = A^+A \tag{2.12}$$

where A is an arbitrary matrix and A^+ is its Moore-Penrose inverse. Since all of our matrix components represent a physical geometry and thus take real values, we can ignore the last two conditions as they will always hold. To construct these matrices,

we use singular value decomposition, the method used by the Mathematica function `PseudoInverse[]`. Then the linear equation $Ax = b$ has a solution when

$$AA^+b = b, \quad (2.13)$$

in which case the most general form of the solution is

$$x = A^+b + (I - A^+A)y \quad (2.14)$$

where y is arbitrary.

For our case then, after finding the pseudo inverse for the face-vertex connectivity matrix \mathbf{C}^+ the naive solution is simply to check if the given torques satisfy $\mathbf{C}^+\mathbf{C}\vec{\tau} = \vec{\tau}$. However, this is not true for many torque choices and ignores the physics of the problem. The column space of \mathbf{C} is the orthogonal complement of its left nullspace, so we can decompose the torque vectors into its components in the range of \mathbf{C} and those in its left nullspace. Physically, we can think of this as being the components that would result in forces on the vertices which cause the origami to fold and those that are counteracted by internal forces in the origami due to geometric constraints encoded by \mathbf{C} .

We then have a few choices for finding this decomposition. If we represent the j th vector in the nullspace of \mathbf{C} with σ_j , then the components of the torques that are compatible with \mathbf{C} are given by

$$\vec{\tau}_C = \vec{\tau} - \sum_j (\sigma_j \cdot \vec{\tau}) \sigma_j. \quad (2.15)$$

This construction is presented to draw parallels between self-stresses and the internal forces imposed by the geometry of the origami encoded in the face-vertex connectivity matrix. We also find it slightly more transparent than the next option for decomposition.

Our other option is to utilize $\mathbf{C}\mathbf{C}^+$, which for the Moore-Penrose inverse can be interpreted as a projection operator on the column space of \mathbf{C} along the nullspace of \mathbf{C}^+ per Corollary 2.7 in Ref. [75]. Since, by construction via singular value decomposition, the nullspace of \mathbf{C}^+ is the left nullspace of \mathbf{C} , $\mathbf{C}\mathbf{C}^+$ acting on any $\vec{\tau}$ will yield the projection of $\vec{\tau}$ on the column space of \mathbf{C} along its nullspace, i.e.

$$\vec{\tau}_C = \mathbf{C}\mathbf{C}^+\vec{\tau}. \quad (2.16)$$

This is the same result as in Eq. 2.15, although we find the construction of Eq. 2.15 more transparent. This version, however, makes it clear that $\vec{\tau}_C$ satisfies Eq. 2.13 (using Eq. 2.10) and is thus a solution to our matrix equation, Eq. 2.2.

Now that we have a method for extracting the compatible components of our torques, we can use the compatible forms to calculate the forces on the vertices. Additionally, we can use \mathbf{P}_B to project the forces into those that won't bend the faces. From there, we can use the force vectors to predict the branch that the origami will choose. We have a few choices for how to do this depending on the origami.

For rigid origami where the branches have unique M-V assignments, we can treat the magnitudes of the forces as infinitesimal initial z-displacements and convert these heights directly to fold angles. Near the flat state, the heights are linearly related to the folds to quadratic order [41], and the matrix to convert between the two can be found using the `TorsionalFoldMatrix[]` function in the `mechanisms` package. From these near-flat fold angles we can extract the M-V assignments, giving us the branch information. For one degree of freedom origami, this information with the forces, which also encode the "pop" direction of each vertex, will usually uniquely define a branch with a few notable exceptions [41].

For elastic origami or origami that do not meet these conditions, we can instead utilize the energy introduced in Section 1.2. Using the forces to create an initial

displacement of the vertices in the z-direction, we can then minimize the energy to identify the branch.

To design origami, if we have a known set of target angles we can find the corresponding vertex positions and use the heights of the vertices for a naive choice for the force vector \vec{f} . Then we can project out the bending components and simply multiply this resulting vector by \mathbf{C} to find a set of torques. Here is where the problem becomes more complicated - most physical implementations of this type of origami are restricted in the torques that can be applied. Following similar logic as the projection of the torques into compatible form, we can add elements of the nullspace of \mathbf{C}^+ to the torques without affecting the resulting forces. This presents a method for designing torques for any origami system, although at this point we would need to work with an experimental group on exact implementation.

2.1.4 Discussion

In this section we presented a set of tools for handling origami that are actuated due to torques on the faces rather than at the folds. While this may seem to only apply to this specific class of origami, the face coordinate system introduced can be used to understand the isometries of non-triangular faces. To continue further with this project, however, we would need to work with an experimental group to understand the limitations of programming their specific systems.

2.2 Maximal Minimal Forcing Sets

Forcing sets, or sets of folds that when actuated will cause the rest of the origami to fold, were first introduced by Ballinger et. al. [76]. Minimal forcing sets are then relevant for self-folding as they present the fewest number of engineered folds needed. This is particularly of interest for tethered origami or origami with mechanical actuators [72, 73]. So far there is no procedure for finding minimal forcing sets for

general origami, although they have been established for 1D origami [77], the special Miura-Ori fold pattern [76], and flat-foldable single vertices [78].

In this section, we will first present a condition for the folding of any given single vertex regardless of its exact geometry. We will then show how to globally apply this condition to an arbitrary origami while minimizing the number of forced folds both by using a graph theory treatment and a Monte Carlo method. Comparing our condition to the results on flat-foldable single vertices from Ref. [78], we know that this condition is not the minimum condition for every single vertex, but overall we present a way to find an upper bound on the forcing set and what we will term a *maximal minimal forcing set*.

2.2.1 Forcing an arbitrary single vertex

Consider a single vertex origami with E_I folds. We also have that the number of external edges E_B and boundary vertices V_B are equivalent to E_I , $E_B = E_I = V_B$. We have previously shown[41] that an origami has 2^{V_I} branches, so a single vertex origami has 2 branches. We will also introduce the terminology *assigned* to refer to folds that have a prescribed target angle and thus drive the folding of the motion. We will use *unassigned* to refer to folds that are not driving the folding motion, although later we will need to address that for almost every experimental realization a non-driving fold typically does not want to fold and thus can be understood as having a target angle of zero.

To find minimal forcing sets, let's consider the fold space of a single vertex origami. The dimension of fold space is simply the number of folds, $D_F = E_I$. For a given origami, the dimension of its branches, which is also its degree of freedom, is given by $D_B = V_B - 3$, as discussed in Sec. 2.1.1. Naively, this would suggest that the number of assigned folds needed for a single vertex to fold robustly would be $N_* = E_I - 3$. However, the branching of origami configuration spaces means that there is not a

single, smooth coordinate system. An additional constraint needs to be added to essentially specify the branch, leading to our final condition,

$$N_* = E_I - 2. \quad (2.17)$$

Note that this additional fold assignment is not truly selecting a branch, just as a set of $E_I - 3$ folds would not choose a position on a branch. The condition above is general and does not require any consideration of the way origami branches are structured.

Alternatively, we can consider the height space of the origami. We typically define height space by the heights of an origami's vertices relative to a chosen reference face. Correspondingly, the dimension of height space is the total number of vertices less 3, $D_H = V_B + V_1 - 3$. For a single vertex origami, we can reduce this to $D_H = E_I - 2$. With constraints from the geometry of the origami, the height space is actually a full description of the origami at a lower dimension than fold space, so this immediately yields our condition $N_* = E_I - 2$.

It should be noted, however, that the condition for robust folding for a single vertex as presented above is purely an upper bound. Abel et. al. [78] have shown that there are some flat-foldable - origami that when folded return to a plane - single vertex fold patterns that require fewer than $n - 2$ forced folds due to symmetries, but these are often only true for specific mountain-valley specifications for those origami. For example, the birdsfoot origami as introduced in Section 1.2.3 has been shown to only need one assigned fold to completely fold. However, this can only be used to fold into the branch where no folds have an equilibrium angle of zero (branch 2 as seen in Fig. 1.8b), and two folds still need to be assigned to fold into the other branch (branch 1 as seen in Fig. 1.8b). We will discuss accounting for these in more detail in Section 2.2.5, but for now will focus on this condition.

2.2.2 Finding a 2-Subgraph

Instead of the more intuitive approach of tracking or finding the number of assigned folds, the relationship above indicates that what we want to track instead is the number of *unassigned* folds. Every vertex has 2 unassigned folds, regardless of its actual degree. If we are thinking of an origami as a graph, by reducing it to its vertices and edges, then what we are attempting to find is then instead a 2-subgraph on its internal vertices.

Before moving on, it will be useful to introduce some graph theory definitions. First, a *graph* is simply a set of vertices and their adjoining edges. The *degree* of a vertex is the number of edges that adjoin a vertex. A *subgraph* is collection of vertices and edges that belong to a larger graph that are in themselves a graph. An *independent edge set* or a *matching* is a set of edges that do not share vertices. The largest matching possible for a given graph is called a *maximal matching*. A *perfect matching* is a maximal matching that is able to include all of the vertices as end points for the edge set.

There are many different types of graphs that have been defined because of their unique properties. One type that will come into our discussion here are *bipartite*. Bipartite graphs are characterized by having two distinct sets of vertices where no two vertices within each set are connected to each other. As an example, the subgraph consisting of the two sets of vertices highlighted in blue and the edges adjoining the two sets in Fig. 2.3 is a bipartite graph.

Our problem is then reduced to finding a 2-subgraph, meaning a subgraph of the original origami where each internal vertex is now degree 2. The problem of finding a 2-subgraph on a general network is related to finding Hamiltonian cycles, or a sequence of edges that visit every vertex exactly once, a classical graph theory problem. Proving the existence of a Hamiltonian cycle or a 2-subgraph is NP-complete, but - depending on the problem and graph - finding them can be NP-hard[79, 80].

Tutte[81] proposed a polynomial-time algorithm for finding 2-subgraphs by changing the problem to a maximal matching problem [80], after which it can be solved using known algorithms, such as the blossom algorithm[79, 82]. Briefly, Tutte’s method is to first replace each internal vertex n of degree d with the bipartite graph $G_n = K_{d,d-2}$. Then preserve the original edges, i.e. edge $E_{n,m}$ should now connect graphs G_n and G_m , on the d sides of the bipartitions. This process is shown for a degree 4 vertex in Fig. 2.3. Finally, find a perfect matching over the new graph. In order to include the vertices within the bipartite graphs as part of the matching, the algorithm must select $d - 2$ edges within the bipartite graph, leading to only two edges leaving the expanded vertex to be included in the perfect matching. This is shown in Fig. 2.3 with dashed red lines and vertices.

Typically, this method would also be applied to the boundary vertices of the graph as well, but we do not need (or want) the 2-subgraph to extend to the boundary vertices or edges. Instead, we will separate the internal edges leading to a boundary vertex, creating additional vertices where needed. This first step is shown in Fig. 2.4b. Here, we diverge from Tutte’s algorithm. Introducing additional vertices with degree 1 results in a perfect matching no longer existing for almost all origami patterns [81]. A maximal matching, however, in most cases will still behave as in the original algorithm, selecting $d-2$ edges inside the bipartite graphs and two edges in the original origami for each original vertex. A maximal matching for our example origami can be seen in Fig. 2.4c. The Mathematica functions and documentation for this procedure for arbitrary origami can be found in the code appendix.

There are notable exceptions. When there are too few interior vertices relative to exterior vertices, and thus too many 1-degree vertices, the maximal matching will sometimes choose to highlight additional edges leading to these 1-degree vertices rather than the connections within the bipartite graphs. Luckily, determining whether a generated set of folds meets the requirement that two edges at each vertex be

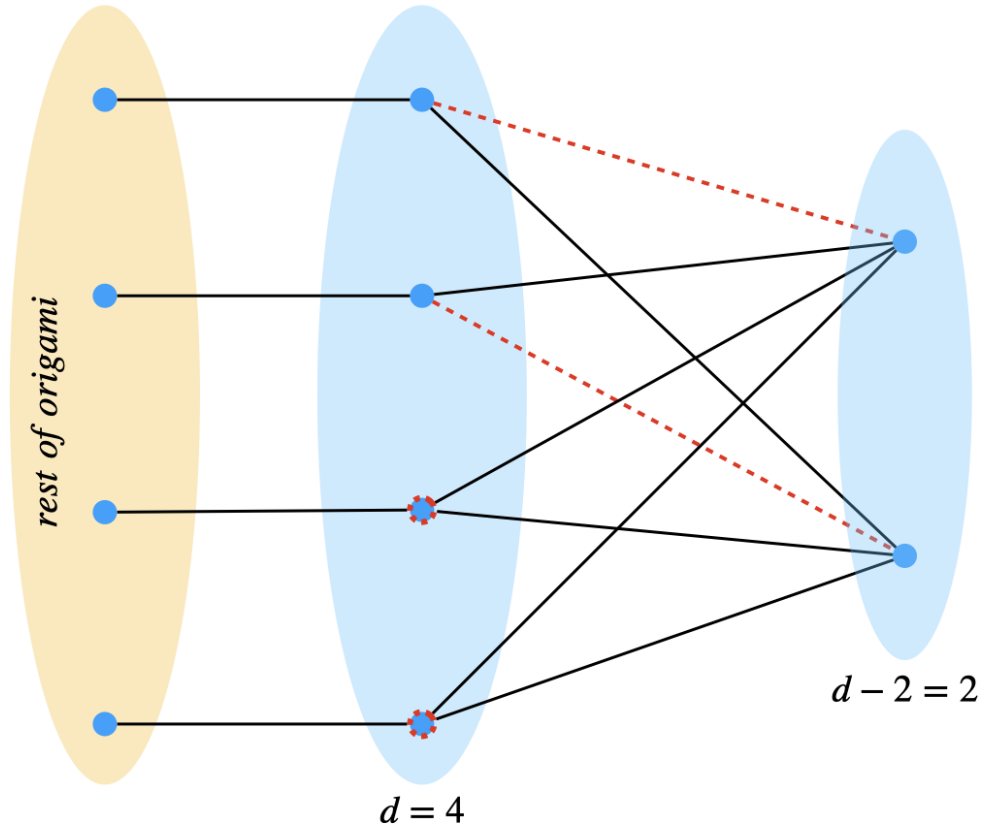


Figure 2.3. An example bipartite expansion of a vertex with degree 4 as part of the algorithm for finding a 2-subgraph on an origami in order to find a minimal forcing set. The red dashed lines demonstrate a matching that would include the $d - 2$ vertex set as end points, and the red dashed vertices show the two vertices that would be free to be included in a maximal matching over the whole graph.

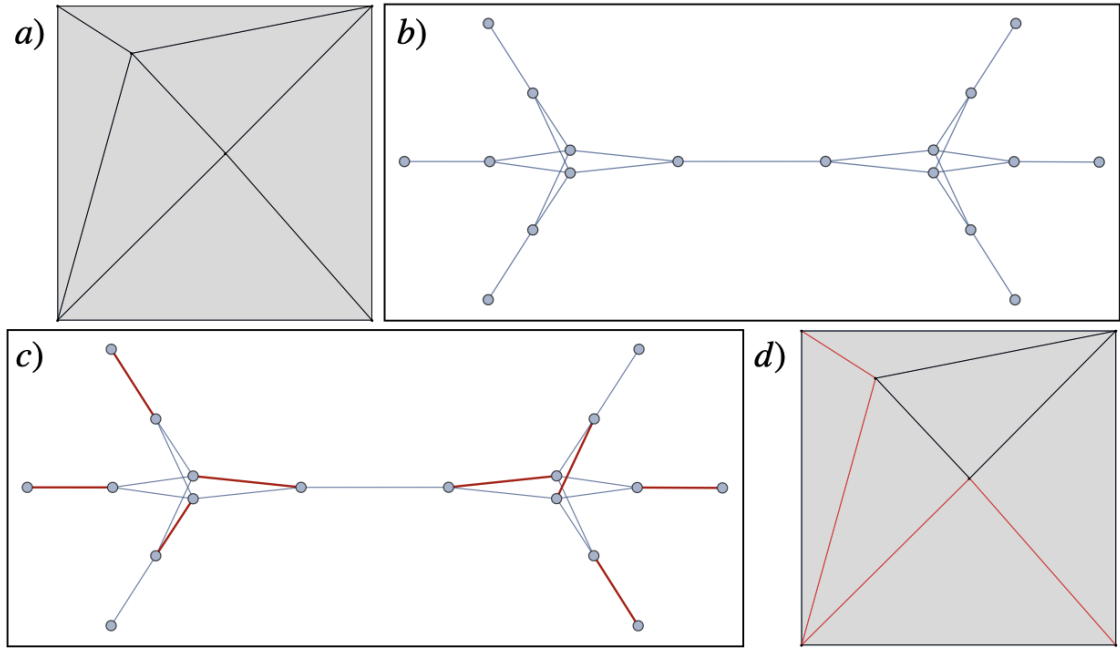


Figure 2.4. The method for finding the maximal minimal forcing set using Tutte's 2-subgraph algorithm using a simple two vertex origami as an example. (a) shows the original origami while (b) is its corresponding bipartite expansion. The red highlighted edges in (c) show the maximal matching on the full graph while in (d) they represent the chosen unassigned folds. Correspondingly, the black internal folds in (d) are the maximal minimal forcing set.

included in the edge set is easy to check. The difference between the number of edges in the maximal edge set of the graph and the number of selected folds in the origami should be $d - 2$ summed over all vertices, i.e. in every bipartite vertex expansion $d - 2$ edges should be selected.

To understand how often this occurs, we generate random origami with four boundary vertices and 3, 4, 5, and 6 internal vertices. Generating 1000 random origami for each number of internal vertices, running each origami through our procedure for finding the forcing set, and then checking that each one is indeed a forcing set gives us an idea of how common these edge cases are. We found that 16.1% of the generated 3 vertex origami, 2.9% of the 4 vertex origami, 0.2% of the generate 5 vertex origami, and none of the 6 vertex origami returned correct minimal edge sets.

2.2.3 Spin System Version

This section is largely included for completion, although it can also be used as an alternative method for the cases where the above graph theory-based algorithm fails. We introduce a spin-system inspired energy for the number of assigned folds at each vertex, with whether a fold is assigned or unassigned mapped to spins and the vertices as interaction sites. We can then borrow Monte Carlo method techniques developed for Ising models to generate the maximal minimal folding set, although it is more computationally expensive and should only be used when the other method fails.

Let's start by rewriting our single vertex folding condition. From Section 2.2.1, at every vertex n we want the assigned folds to satisfy the condition

$$N_{n*} - E_n + 2 = 0 \tag{2.18}$$

where N_{n*} is the ideal number of assigned folds and E_n is the number of folds at vertex n . This equation represents that at every vertex we want precisely two of the

folds to be unassigned. To turn this into an energy, we want too many and too few assigned folds to be equally costly. This leads us to the Hamiltonian

$$H = \sum_n (N_n - E_n + 2)^2, \quad (2.19)$$

where N_n is the number of currently assigned folds (a variable).

To turn this into something that looks like a spin system or Ising model, let's define the incidence matrix I_{ne} to be 1 when edge e connects to vertex n and 0 otherwise. Next, place spins on the edges, and take σ_e to have a value of 1 if a fold is assigned, meaning it has a prescribed target angle, and 0 otherwise. This leads to an expression for the number of assigned folds,

$$N_n = \sum_e I_{ne} \sigma_e, \quad (2.20)$$

as well as a site interaction matrix equivalent,

$$J_{e_1, e_2} = \sum_n I_{ne_1} I_{ne_2}, \quad (2.21)$$

which takes a value of 1 if e_1 and e_2 interact through n and 0 otherwise.

Now, we can expand H

$$H = \sum_n N_n^2 + 4 \sum_n N_n - 2 \sum_n N_n E_n + \sum_n (E_n^2 - 4E_n + 4). \quad (2.22)$$

The first term becomes $\sum_{e_1, e_2} J_{e_1, e_2} \sigma_{e_1} \sigma_{e_2}$, the second two terms can be combined,

$$\sum_n ne(4I_{ne}\sigma_e - 2E_n I_{ne}\sigma_e) = 2 \sum_e \left(\sum_n (2I_{ne} - E_n I_{ne}) \right) \sigma_e, \quad (2.23)$$

and the final term is simply a number. Putting this all together, we get

$$H = \sum_{e_1, e_2} J_{e_1, e_2} \sigma_{e_1} \sigma_{e_2} + 2 \sum_e h_e \sigma_e \quad (2.24)$$

where $h_e = \sum_n (2I_{ne} - E_n I_{ne})$. Now, this looks like an Ising model complete with a “magnetic field”. Simple tests using a Metropolis-Hastings algorithm to find the minimal forcing set were successful but slow.

2.2.4 Simulations

As further evidence that our maximal minimum forcing set truly forces the origami, we performed simulations on random origami and compared the cases where every fold is programmed to have a target angle and where only the folds chosen by the maximal minimum forcing set are programmed. For each generated random origami we found sets of target angles by generating an uniform distribution of vertex displacements with a width of the smallest fold angle and applying these displacements to the origami before minimizing its energy. We could then select target angle sets from this list.

From here, we followed a similar procedure as that used for the Randlett bird simulations in Sec. 1.2.4. After either assigning all of the target angles or only those of the minimal forcing set, we generated an uniform distribution of vertex displacements with a width of 20% of the smallest fold length and applied these displacements to the origami. Then we used this distribution as the initial conditions for the energy minimizations. After minimizing all of the origami, we then found the average difference squared between each folded origami and the target fold angles,

$$\Delta = Avg[(\alpha_{target,i} - \alpha_{actual,i})^2] \quad (2.25)$$

where α_{target} represents the programmed angle, α_{actual} represents the minimized angle, and i indexes over all folds. The code and documentation for this process can be found in the Appendix A.

Some care must then be taken when assigning fold stiffness constants and interpreting the results. In an experiment, a truly unassigned fold does not exist. For

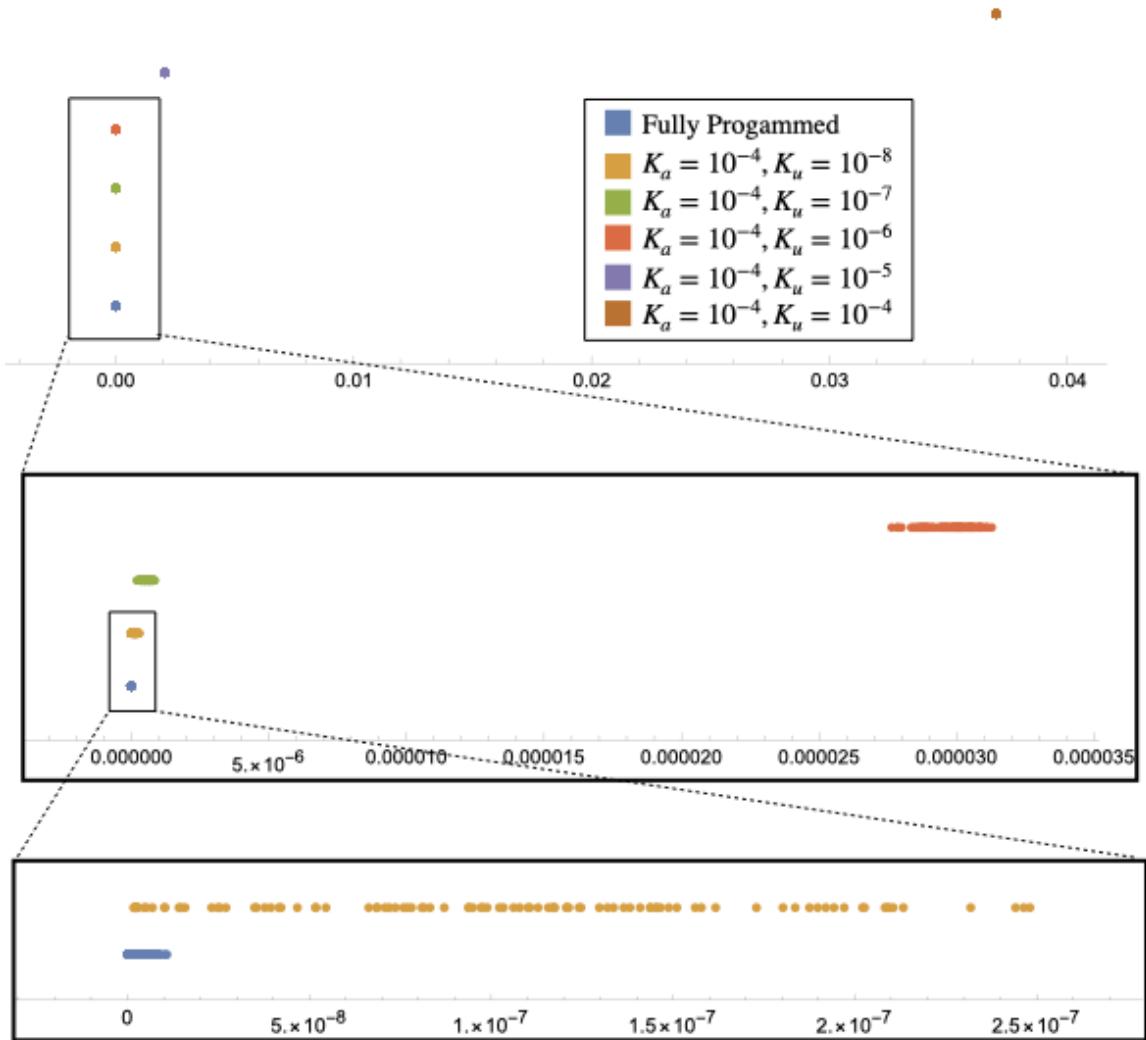


Figure 2.5. Comparisons of different torsional spring stiffness constants for the assigned and unassigned folds for the folding simulations. Each cluster of points represents 100 folded origami.

example, if we consider the trilayer swelling gel origami introduced in Chapter 1, the closest we can get to an unassigned fold is to cut both sides of the stiff layer rather than just one. The fold is then biased to be neither a mountain or valley fold and rather wants to stay flat, although it does have a reduced stiffness. In the simulations, this means that while we can set the torsional spring stiffness of an unassigned fold to zero, a more accurate choice would be to set its target angle to zero and its spring stiffness to a small but nonzero value. Because these unassigned folds are still contributing to the energy, they will act against the total folding. To understand the effect of spring constant choices, we ran the same random 3 vertex origami through our minimization with different choices for the torsional spring constants and compared to the case where every angle is precisely programmed. This data can be seen in Fig. 2.5.

We then chose to use values of 10^{-4} for the assigned fold stiffness constants and 10^{-5} for the unassigned fold stiffness constants and iterated the described process. Out of 50+ randomly generated origami with a range of internal vertices from 3 to 7, each with a minimum of 3 different target angle sets we saw only one origami and angle set that resulted in the origami not folding. This case was a 3 vertex origami with two angles that nearly bisected the origami and the set of fold angles corresponded to only those two fold angles actuating. The partially simulated version simply did not fold and remained flat. In every other case the origami folded and was close to the target values. A representative sampling of 5 origami and angle sets can be seen in Table 2.1. We see that the partially programmed origami does not get quite as close to the target angles as the fully programmed origami, but both versions have the same mountain-valley assignment, which is the only requirement for the traditional definition of the forcing set [76].

Additionally, this comparison of fully and partially programmed origami is not entirely representative of how physical origami will respond to a reduction in the

Table 2.1. Simulation data for a few random origami with both only the minimal forcing set programmed with target angles and with all folds programmed. The average fold difference is calculated using Eq. 2.25 and represents the distance from the target angles in *radians*². Each line consists of around 500 iterations for each set of target angles.

Number of Internal Vertices	Avg Fold Difference Fully Programmed	Avg Fold Difference Partially Prog.	Max. Difference Part. Prog.
3	1.2086×10^{-9}	0.0219024	0.0219284
	1.1663×10^{-9}	0.0015457	0.0015536
	1.4058×10^{-9}	0.0070801	0.0070852
	9.2121×10^{-10}	0.17632	0.17633
	2.5356×10^{-10}	0.043531	0.043557
4	5.0384×10^{-10}	0.043210	0.043247
	8.3471×10^{-10}	0.0060604	0.0060736
	1.0800×10^{-9}	0.0038227	0.0038313
	4.5105×10^{-10}	0.00030619	0.00030783
	3.7775×10^{-10}	0.056115	0.056137
5	5.4227×10^{-10}	0.089906	0.089916
	3.1493×10^{-9}	0.0021736	0.0021793
	1.0517×10^{-9}	0.10964	0.10964
	3.8708×10^{-10}	0.0021234	0.0021314
	9.2448×10^{-10}	0.0082082	0.0082351
6	2.7602×10^{-9}	0.065087	0.065089
	2.5157×10^{-10}	0.034605	0.034629
	2.0346×10^{-9}	0.060375	0.060445
	6.1892×10^{-10}	0.0052119	0.0052010
	6.1697×10^{-10}	0.082349	0.082351
7	4.2413×10^{-10}	0.028590	0.028599
	7.5415×10^{-10}	0.0021709	0.0021755
	5.0676×10^{-10}	0.0032008	0.0032050
	4.7600×10^{-10}	0.027133	0.027148
	3.7314×10^{-10}	0.049496	0.049516

number of programmed folds. Physical origami has limits in what its angles can and cannot be programmed to, and certainly does not have precision in angle programming to fractions of radians. To get a preliminary idea of how a physical origami will behave, we need to know the experimental target fold angles, not those directly on the target branch. In Section 1.2.4, we used the example of the Randlett bird, for which we have experimental target angles. We also previously investigated how choosing face and fold stiffnesses effects the number of branches seen and so additionally have guidance for which stiffnesses to use. Accordingly, we choose $K_{face} \approx 10^{-5}$ and $K_{fold} \approx 10^{-6}$. Then we will consider the unassigned folds to be set to zero by removing both stiff layers along the folds leading to a choice of $K_{unassigned} \approx 10^{-7}$. Following the same procedure as explained for randomly generated origami, we can find how closely the simulated origami will fold to the target angles. A histogram comparing the minimally and fully assigned versions can be seen in Fig. 2.6.

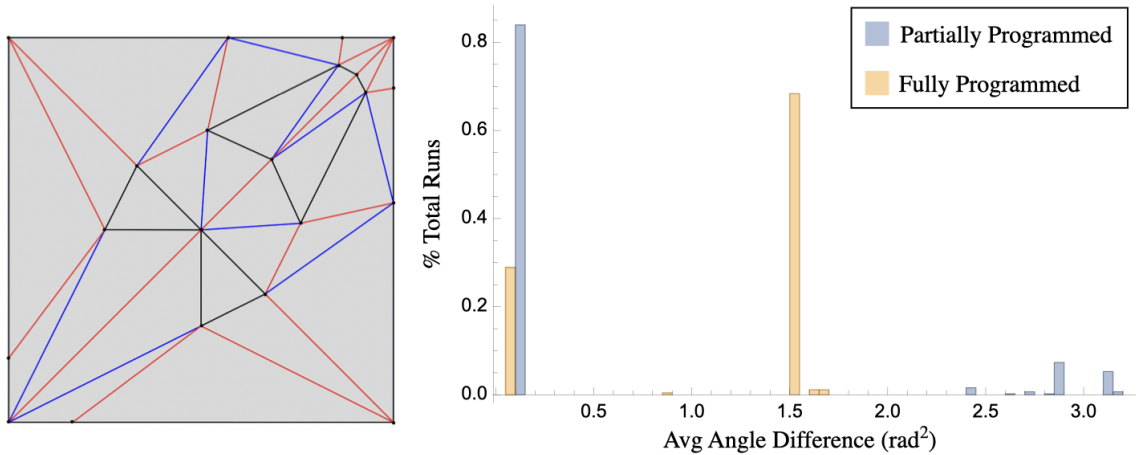


Figure 2.6. (Left) The randlett bird with the unassigned folds marked in red, assigned folds in black, and face folds in blue. (Right) The average difference between the final fold angles of the simulation and the programmed target angles. Blue represents the minimally programmed version while orange represents the fully programmed version.

The simulation results are quite promising. We do see a similar effect to the random origami where the minimally programmed version does not fold quite as

closely to the target angles as the fully programmed version when comparing the states closest to the targets. However, we do see a drastic change in the relative percentage of target fold patterns reached, with fully programmed version misfolding drastically more than the partially programmed version.

2.2.5 Discussion

Taken altogether, here we have presented a method for finding an upper bound on the minimal forcing set for arbitrary origami. Our graph theory based procedure will, for most origami, select for a 2-subgraph on the internal edges of an origami, maximizing the number of unassigned folds while still meeting the condition for every vertex to fold robustly. While origami with low internal connectivity may cause problems, it is easy to check that each generated maximal minimal forcing set meets our condition on every vertex. Additionally, we presented a Monte Carlo based method that can be used in these cases. We then checked our results with simulations for random origami.

This still is not, however, the true minimal forcing set for any origami, as noted earlier. By classifying the single vertex geometries with fewer needed forced folds found in Ref. [78], we could account for these edge cases by introducing an alternative energy condition into the spin-system method for finding the forcing set. To reduce computational time, we can start by finding the maximal minimal forcing set using the graph-theory-based algorithm above and feeding it as an initial state to the spin-system Monte Carlo. We may also be able to incorporate these special cases as weightings on the graph, but it would likely involve writing out own maximal matching algorithms and will be outside of the scope of this thesis.

CHAPTER 3

DESIGNING LINKAGES

This chapter arises from the following observation: there are many applications where the precise motion of the vertices of a linkage is less important than the motion's qualitative features. In these cases, not only can we achieve the desired effect, but we can do so with simpler linkages. Moreover, we demonstrate that select small changes in the linkage geometry can drive large, qualitative changes in the behavior of the mechanism, without changing connectivity.

A note on collaboration in this chapter: The work on this chapter was done in collaboration with fellow graduate students Michelle Berry, whose main focus was on the numeric and computational aspects, and David Limberg who designed and performed the Lego experiments. I contributed to theoretical basis and analysis, with my main contribution being the interpretation of the regions of space demarcated by the critical value surfaces as being regions of distinct behavior.

In this chapter, we introduce an approach to linkage design that takes advantage of the critical points of a configuration space [84, 85]. At a critical point, a mechanism has an anomalously large class of potential linear motions available to it, but higher order corrections from the mechanical constraints restrict the motion to a subset of these motions [86–89]. Many linkages of this type have branched configuration spaces, meaning that many different qualitative motions are accessible. For example, generic origami and kirigami mechanisms have highly branched configuration spaces, as we

This chapter is adapted from Ref. [83] with permission from the American Physical Society.

saw in previous chapters, leading to pluripotency [41, 72]. Similar branched configuration spaces have been used to design mechanical logic devices [90] and kinematotropic mechanisms that can change how many degrees of freedom they can access [91, 92].

Critical points are delicate; even small perturbations of the mechanism geometry will destroy them. However, by controlling those perturbations, we demonstrate that the configuration space topology of a mechanism can be controlled. As a result, we obtain linkages with switchable motion using only a small number of tunable components. We demonstrate this extreme switchability in celebrated Kane-Lubensky (KL) chain [21], a series of rotors joined by springs, which supports the propagation of a soliton called a “spinner” in which each rotor, in turn, rotates a full 360° degrees [93, 94]. We will apply our design methodology to the KL chain. By replacing one of the unit cells with a designed mechanism, we show that the propagation of the spinner soliton can be controllably gated. Because our method relies on topological programming, the mechanisms are robust to flexibility and fabrication imperfections.

In Sec. 3.1, we review relevant parts of rigidity theory and mechanisms. In Sec. 3.2, we describe mathematical tools that provide a geometrical interpretation to critical points. This interpretation will provide the basis of our design methodology, which we will illustrate with an example containing five bars. Finally, in Sec. 3.3, we will use our formalism to explicitly design a mechanism to gate the KL chain. Importantly, the operation of the resulting gate is robust with respect to small perturbations. Finally, we conclude with a brief discussion highlighting new directions enabled by this work.

3.1 Critical Points in Mechanisms

3.1.1 Mechanism Rigidity

In this section, we review the basic mathematical description of mechanisms. Though we focus on linkages, which are constructed entirely from free-rotating joints

and inextensible bars, the formalism can be generalized to mechanisms with other holonomic constraints. We define a linkage as a collection of V vertices in d dimensions joined by E rigid bars. The configuration of a linkage can then always be represented by a point, \mathbf{u} , in the space of vertex positions, which we will denote as \mathcal{M} , and has dimension $M = Vd$. We assume there are E bars in the linkage and denote the length of the α^{th} bar, $\ell_\alpha(\mathbf{u})$. The configuration space of the linkage can then be represented by the family of equations,

$$\ell_\alpha^2(\mathbf{u}) = L_\alpha^2 \quad (3.1)$$

where L_α is the target length of the α^{th} bar. Note that Eq. (3.1) is written using the square of $\ell_\alpha(\mathbf{u})$ so that it can be an analytic function everywhere. By replacing $\ell_\alpha(\mathbf{u})$ with a more general class of functions in Eq. (3.1), we can also describe mechanisms with more complex components beyond rigid bars.

Rather than analyzing the configuration space for specific values of L_α , we will instead analyze the entire family of configuration spaces that can occur with a fixed network topology by changing the L_α . Between Kempe's universality theorem and the potential arbitrariness of $\ell_\alpha(\mathbf{u})$, however, it is indeed difficult to say a great deal more about the configuration space with any kind of generality. Therefore, we assume that $\ell^2(\mathbf{u})$ is an analytic function of \mathbf{u} and that $E \leq Vd$. With these assumptions, the Jacobian matrix, whose components are

$$J_{\alpha i}(\mathbf{u}) = \frac{\partial \ell_\alpha^2(\mathbf{u})}{\partial u_i}, \quad (3.2)$$

provides critical information about the mechanism. Naively, one would expect the configuration space of the mechanism to be $D = M - E$ (for $M > E$). Indeed, the inverse function theorem implies that the configuration space is a smooth D dimensional manifold in any open set of \mathcal{M} in which the Jacobian matrix is full rank.

At such a configuration \mathbf{u} , the tangent space coincides with the right null space of $J_{\alpha i}(\mathbf{u})$,

$$\sum_i J_{\alpha i}(\mathbf{u}) \delta u_i = 0. \quad (3.3)$$

The solutions δu_i of Eq. (3.3) are called zero modes.

Any point \mathbf{u}_C at which the Jacobian fails to be full rank, on the other hand, we call a critical point, and the corresponding edge lengths $\ell_\alpha^2(\mathbf{u}_C)$ we call a critical value. Critical points are characterized by self stresses, σ_α , which are elements of the left null space of $J_{\alpha i}(\mathbf{u}_C)$,

$$\sum_\alpha \sigma_\alpha J_{\alpha i}(\mathbf{u}_C) = 0. \quad (3.4)$$

Because of their relation to critical points, we will see that self stresses play an important role in the topology of the configuration space.

Sard's theorem ensures that critical values (but not necessarily critical points) are a set of measure zero. In that sense, most choices of edge lengths lead to a configuration space that is a smooth D dimensional manifold. Consequently, any change in the configuration space's topology that occurs as the L_α change must happen at a critical point. Thus, these critical points also govern the overall topology of the configuration space of a mechanism.

In the next section, we proceed to analyze the geometry of the configuration space at and near such critical points.

3.1.2 Shape of the configuration space at critical points

To understand the shape of the configuration space, we expand $\ell_\alpha^2(\mathbf{u} + \delta \mathbf{u})$ for small deformations, $\delta \mathbf{u}$ having components δu_i , around the critical point, obtaining

$$0 = \sum_i J_{\alpha i} \delta u_i + \frac{1}{2} \sum_{ij} \frac{\partial^2 \ell_\alpha^2(\mathbf{u}_C)}{\partial u_i \partial u_j} \delta u_i \delta u_j + \mathcal{O}(\delta u^3). \quad (3.5)$$

It is common at this stage to write a formal series expansion, $\delta \mathbf{u} = \delta \mathbf{u}^{(1)} + \delta \mathbf{u}^{(2)} + \dots$, and substitute it into Eq. (3.5). One finds $\delta u^{(1)}$ is a zero mode of the Jacobian satisfying [88]

$$\frac{1}{2} \sum_{\alpha} \sum_{ij} \sigma_{\alpha}^{(n)} \frac{\partial^2 \ell_{\alpha}^2(\mathbf{u}_C)}{\partial u_i \partial u_j} \delta u_i^{(1)} \delta u_j^{(1)} = 0, \quad (3.6)$$

where $\{\sigma_{\alpha}^{(1)}, \sigma_{\alpha}^{(2)}, \dots\}$ is a basis for the space of self stresses at \mathbf{u}_C .

To proceed, we make further assumptions. The most important of these is that *Eq. (3.6) completely characterizes the local geometry of the critical point*. It is well-known that if no solution to Eq. (3.6) exists then the linkage is rigid, but the converse does not necessarily hold. There are mechanisms whose rigidity is only visible at higher order, as well as mechanisms that are rigid at order larger than two but, nevertheless, are mobile [87]. Experience suggests that these examples are rarer than the better behaved examples we consider here, but we are unaware of any results quantifying their rarity or even a simple means to determine when Eq. (3.6) is sufficient to describe the geometry of the critical point accurately. For the scope of this paper, it will prove sufficient to assume we can safely truncate our expansion of $\delta \mathbf{u}$ at second order and check, *post hoc*, that the results produced by our design procedure satisfy our assumptions.

We will make three other assumptions as well:

- 1 *All critical points, \mathbf{u}_C , lying on a configuration space of constant L_{α} are isolated.* There are linkages for which this fails and for which the entire configuration space lies along a sequence of critical points (see, for example, [95]). Note, however, that there are also mechanisms with large D which do satisfy this assumption [41, 96, 97]. In this paper, we will ultimately focus on example mechanisms with only a single degree of freedom, so this will not prove a particularly strong assumption, but in this section we allow D to be general and only specialize to $D = 1$ subsequently.

2 *All critical points have exactly one self stress.* This assumption is certainly not always true. It fails, for example, in flat origami mechanisms [41]. Generally, however, we will see that, qualitatively, critical points with several self stresses appear to require more fine-tuning. This assumption implies that there will be $D + 1$ zero modes at each critical point by the rank-nullity theorem applied to the Jacobian matrix at \mathbf{u}_C .

3 *The matrix*

$$\sum_{\alpha} \sigma_{\alpha} \frac{\partial^2 \ell_{\alpha}^2(\mathbf{u}_C)}{\partial u_i \partial u_j}$$

has nonzero eigenvalues when restricted to the zero modes at \mathbf{u}_C . This assumption allows us to simplify the characterization of the critical points. Notice that without assumption 2, this characterization would be more difficult because the Eq. (3.6) would yield a system of quadratic equations rather than a single equation.

While all of these assumptions will play a role in our analysis, one could relax some of them at the expense of complicating the design procedure. Additionally, they may not be mutually exclusive from our assumption zero that Eq. 3.6 completely characterizes the local geometry of the critical point. Our examples will satisfy them, however, and we leave it for future work to understand which are truly required and which are conveniences.

Suppose we choose a basis for the zero modes at \mathbf{u}_C , $\{\zeta_1, \dots, \zeta_{D+1}\}$, writing $\delta u_i^{(1)} = \sum_n c_n \zeta_{n,i}$. Then Eq. (3.6) becomes

$$\sum_{nm} \mathcal{Q}_{nm} c_n c_m = 0 \tag{3.7}$$

where \mathcal{Q}_{nm} is a symmetric matrix given by

$$\mathcal{Q}_{nm} = \sum_{ij} \sum_{\alpha} \zeta_{n,i} \zeta_{m,j} \sigma_{\alpha}^{(1)} \partial^2 \ell_{\alpha}^2(\mathbf{u}_C) / \partial u_i \partial u_j. \quad (3.8)$$

Under our assumptions, there are just two possibilities. If \mathcal{Q}_{nm} is either positive- or negative-definite, the linkage is rigid: there is no solution to Eq. (3.7) other than $c_n = 0$. If \mathcal{Q}_{nm} has a combination of positive and negative eigenvalues, however, the geometry of the configuration space at \mathbf{u}_C is that of a cone. This is precisely what happens in single-vertex flat origami [1, 41] (Fig. 3.1). We call such a point \mathbf{u}_C a branch point, though this space of possible zero modes is sometimes called a kinematic tangent cone [98].

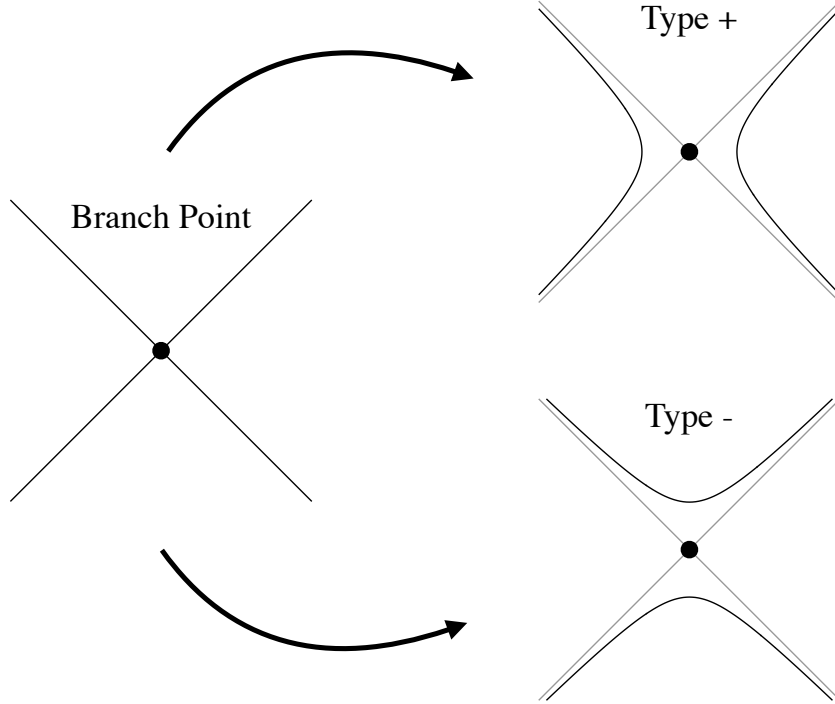


Figure 3.1. Schematic of how a configuration space with a branch point split into one of two types of smooth, disconnected configuration spaces. The choice of sign is arbitrary.

3.1.3 Shape of the configuration space near critical points

We next ask what happens to the configuration space of a mechanism when the lengths are deformed from their critical values, $L_\alpha = L_\alpha^{(c)} + \delta L_\alpha$. We assume we have a mechanism with E edges and V vertices in d dimensions with $dV > E$. We further suppose that the configuration of the mechanism is at a critical point, \mathbf{u}_C , with corresponding critical values $(L_\alpha^{(c)})^2$. Let $\mathbf{u} = \mathbf{u}_C + \delta\mathbf{u}$ and correspondingly $L_\alpha = L_\alpha^{(c)} + \delta L_\alpha$, and expand the squared lengths to quadratic order, using Eq. (3.1),

$$\begin{aligned} 2L_\alpha^{(c)}\delta L_\alpha + \delta L_\alpha^2 &= \sum_i \frac{\partial \ell_\alpha^2(\mathbf{u}_C)}{\partial u_i} \delta u_i \\ &+ \sum_{ij} \frac{1}{2} \frac{\partial^2 \ell_\alpha^2(\mathbf{u}_C)}{\partial u_i \partial u_j} \delta u_i \delta u_j \end{aligned} \quad (3.9)$$

To conclude recapping our assumptions, we assume that Eq. (3.9) completely characterizes the critical point, and that there is one self stress at \mathbf{u}_C , with components σ_α , and two zero modes, with components $\zeta_{1,i}$ and $\zeta_{2,i}$.

It will prove convenient to express Eq. (3.9) using an orthonormal basis in the space of square lengths, $\{\sigma_\alpha, e_\alpha^{(1)}, \dots, e_\alpha^{(E-1)}\}$. We similarly write δu_i in an orthonormal basis $\{\zeta_{1,i}, \zeta_{2,i}, \eta_{1,i}, \dots, \eta_{E-1,i}\}$,

$$\delta u_i = c_1 \zeta_{1,i} + c_2 \zeta_{2,i} + \sum_{I=1}^{E-1} a_I \eta_{I,i}. \quad (3.10)$$

We first contract Eq. (3.9) with σ_α , we obtain an equation that can be expressed as

$$\begin{pmatrix} \mathbf{c}^T & \mathbf{a}^T \end{pmatrix} \begin{pmatrix} \mathcal{Q} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{M} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{a} \end{pmatrix} = \tilde{\Delta}, \quad (3.11)$$

where the components of the matrices are given by

$$\tilde{\Delta} = \sum_\alpha \sigma_\alpha (2L_\alpha^{(c)}\delta L_\alpha + \delta L_\alpha^2), \quad (3.12)$$

$$\mathcal{Q}_{nm} = \frac{1}{2} \sum_{\alpha ij} \zeta_{n,i} \zeta_{m,j} \sigma_{\alpha} \frac{\partial^2 \ell_{\alpha}^2}{\partial u_i \partial u_j}, \quad (3.13)$$

$$\mathcal{M}_{nm} = \frac{1}{2} \sum_{\alpha ij} \eta_{n,i} \eta_{m,j} \sigma_{\alpha} \frac{\partial^2 \ell_{\alpha}^2}{\partial u_i \partial u_j}, \quad (3.14)$$

and

$$\mathcal{B}_{nm} = \frac{1}{2} \sum_{\alpha ij} \zeta_{n,i} \eta_{m,j} \sigma_{\alpha} \frac{\partial^2 \ell_{\alpha}^2}{\partial u_i \partial u_j}. \quad (3.15)$$

We also assume that a_I are the components of the vector \mathbf{a} and that c_1 and c_2 are the components of a two-dimensional vector \mathbf{c} . Finally, we complete the square in Eq. (3.11) to obtain

$$(\mathbf{c} + \mathcal{Q}^{-1} \mathcal{B} \mathbf{a})^T \mathcal{Q} (\mathbf{c} + \mathcal{Q}^{-1} \mathcal{B} \mathbf{a}) = \tilde{\Delta} - \mathbf{a}^T \mathcal{B}^T \mathcal{Q}^{-1} \mathcal{B} \mathbf{a}. \quad (3.16)$$

Note that \mathcal{Q}^{-1} exists because all of the eigenvalues of \mathcal{Q} are nonzero by assumption.

Already, Eq. (3.11) is in the form of a conic section whose form depends on the eigenvalues of \mathcal{Q} . What remains is to show that \mathbf{a} depends only on the length changes (and not \mathbf{c}) to lowest order and, ultimately, to find an expression to determine it.

To do this, we project Eq. (3.9) onto the remaining basis vectors, $e_{\alpha}^{(n)}$, in the space of square lengths. We obtain

$$\begin{aligned} \sum_m \sum_i \sum_{\alpha} e_{\alpha}^{(n)} \frac{\partial \ell_{\alpha}^2(\mathbf{u}_C)}{\partial u_i} \eta_{m,i} a_m + \frac{1}{2} \sum_{ij\alpha} e_{\alpha}^{(n)} \frac{\partial^2 \ell_{\alpha}^2(\mathbf{u}_C)}{\partial u_i \partial u_j} \delta u_i \delta u_j = \\ \sum_{\alpha} e_{\alpha}^{(n)} (2L_{\alpha}^{(c)} \delta L_{\alpha} + \delta L_{\alpha}^2) \end{aligned} \quad (3.17)$$

There are $E - 1$ equations in Eq. (3.17) and $dV - E + 1$ zero modes at the critical point, the space spanned by δu_i^{\perp} is $dV - (dV - E + 1) = E - 1$ dimensional. The matrix appearing in Eq. (3.17) is, consequently, square. Since we have already removed zero

modes and self stresses, it is also invertible. We define a new matrix \mathbf{M} such that its inverse \mathbf{M}^{-1} is given by the components,

$$\mathbf{M}_{nm}^{-1} = \sum_i \sum_\alpha e_\alpha^{(n)} \frac{\partial \ell_\alpha^2(\mathbf{u}_C)}{\partial u_i} \eta_{m,i}. \quad (3.18)$$

This then allows us to solve Eq. (3.17) in powers of both δL_α and \mathbf{c} . To first order in both, we obtain

$$a_n \approx \sum_m M_{nm} \sum_\alpha 2e_\alpha^{(m)} L_\alpha^{(c)} \delta L_\alpha + \mathcal{O}(c\delta L, c^2, \delta L^2). \quad (3.19)$$

We can now put together the results by defining

$$\delta \mathbf{c} = -\mathcal{Q}^{-1} \mathcal{B} \mathbf{a} \quad (3.20)$$

and

$$\Delta = \tilde{\Delta} - \delta \mathbf{c} \mathcal{Q} \delta \mathbf{c} \quad (3.21)$$

to obtain

$$(\delta \mathbf{c} - \delta \mathbf{c})^T \mathcal{Q} (\delta \mathbf{c} - \delta \mathbf{c}) = \Delta \quad (3.22)$$

where Δ and $\delta \mathbf{c}$ depend linearly on the changes in lengths to lowest order. Therefore, small perturbations of the length are seen to produce trajectories that lie on a 2D conic section with a perturbed center.

While this is a rather intricate derivation, we could have obtained the correct answer up to order $\delta \mathbf{u} \sim \delta L^{1/2}$ more simply by assuming $\mathcal{O}(\mathbf{a}) \sim \mathcal{O}(\mathbf{c})$. We have found the full form of Eq. (3.22) to be more useful in perturbing larger linkages, however, as it better captures the case that changes in the bar lengths perturb but do not completely eliminate critical points in the configuration space of a linkage.

Putting all of these pieces together and rewriting for clarity, our final result is

$$\sum_{nm} \mathcal{Q}_{nm}(c_n - \delta c_n)(c_m - \delta c_m) = \Delta, \quad (3.23)$$

where the deformation is along the zero modes at \mathbf{u}_C , $\sum_n c_n \zeta_{n,i}$ as before, and δc_n and Δ are quantities whose value depends linearly on the length changes, δL_α , to lowest order.

We first consider what happens when $\Delta = 0$. In that case, when $\delta c_n = 0$, Eq. (3.23) recovers the results from the previous section: there is either a rigid point or a branch point at $c_n = 0$ corresponding to the critical point \mathbf{u}_C . When $\delta c_n \neq 0$, however, the critical point itself moves by $\approx \sum_n \delta c_n \zeta_{n,i}$.

When $\Delta \neq 0$ and \mathcal{Q} has only positive eigenvalues (the critical point is second order rigid), we have two possibilities: (1) $\Delta > 0$ implies the solution to Eq. (3.23) is an ellipsoid in $D+1$ dimensions (it is almost rigid [99]), and (2) $\Delta < 0$ implies there is no solution to Eq. (3.23). The opposite occurs if \mathcal{Q}_{nm} has only negative eigenvalues.

Finally, we consider the case of a branch point, for which \mathcal{Q}_{nm} has eigenvalues of opposite sign. To develop intuition, it is useful to consider the special case of a branch point when $D = 1$. Then \mathcal{Q}_{nm} is a 2×2 matrix with two eigenvalues of opposite sign. The solutions to Eq. (3.23) take the form of two hyperbolas in the plane spanned by the zero modes at \mathbf{u}_C whose precise configuration depends on the sign of Δ (Fig. 3.1 for characteristic examples for both signs of Δ). For $D > 1$, branch points also break up into smooth surfaces but do so, presumably, in a more complex way that depends on the signature of \mathcal{Q}_{nm} (see Ref. [1] for an example in origami).

As an illustrative example, we turn to the well-studied four-bar linkage shown in Fig. 3.2a. The four-bar linkage is constructed from two rotors of length L_1 and L_3 pinned at one end and joined at the other by a bar of length L_2 . The system configuration can be parameterized as a point in four dimensions with coordinates (x_1, y_1, x_2, y_2) , and the configuration space is one dimensional. When $L_1 = L_2 = L_3 =$

a , there are three branch points each having a single self stress and two zero modes. The configuration space is shown in Fig. 3.2b in terms of the two rotor angles θ_1 and θ_2 . By slightly increasing the length of $L_2 > a$, the branch points all split into a pair of hyperbolas oriented opposite each other in the quadrants spanned by the configuration space when $L_2 = a$. On the other hand, $L_2 < a$ results in the branch point splitting into a pair of hyperbolas in the other pair of quadrants. As a result of switching the orientation of the hyperbolas, the configuration space goes from having a single component for $L_2 > a$ to two disconnected components when $L_2 < a$.

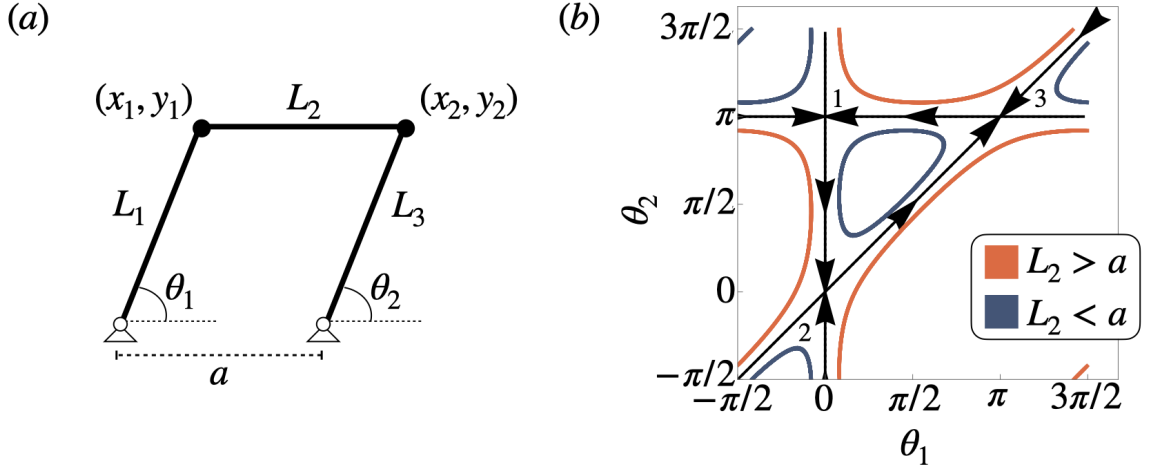


Figure 3.2. (a) Schematic of the planar, four-bar linkage with variables defined. (b) Projection of the configuration space of the two rotor mechanism with $L_1 = L_2 = L_3 = a$ projected into (θ_1, θ_2) plane (black). This choice of lengths has three branch-like critical points. Deforming the length of L_2 results in a smooth configuration space with either one (red) or two (blue) components. The arrows indicate the direction of the tangent form $t_i(\mathbf{u})$ from Eq. (3.36).

3.2 Controlling configuration space topology

We noted earlier that the topology of the configuration space cannot change without passing through an intermediate critical point. If it could, this would contradict the notion that the configuration space is smooth when the Jacobian $J_{\alpha i}$ is full rank. This fact and the analysis of Sec. 3.1 suggests a method for controlling the topology

of the configuration space: (1) find a set of lengths L_α for which the configuration space has many branch points, and (2) perturb the lengths, $L_\alpha \rightarrow L_\alpha + \delta L_\alpha$, such that the branch points split into smooth hyperbolas in the desired configuration. For the four bar linkage in Fig. 3.2b, for example, if we could control how each of the three branch points split independently, we would have complete control over how the configuration space winds around the torus defined by the angles (θ_1, θ_2) as well as the number of components in the configuration space.

3.2.1 Properties of the tangent form

Before moving on to linkages, first we will introduce a new formalism that we call the tangent form. It takes a set of constraints as a function of the relevant coordinates of the system and finds the tangent to the constraints, allowing us in the case of mechanisms to use the bar lengths as constraints rather than specific values. We will go into more detail on using this formalism for mechanisms in the next section, but for now we will introduce it generally.

For a set of N constraints $f_1(\mathbf{u}), \dots, f_N(\mathbf{u})$ where \mathbf{u} are the relevant coordinates of the system, the tangent form is defined as

$$t^{i_1 \dots i_D}(\mathbf{u}) = \sum_{j_1 \dots j_N} \epsilon^{i_1 \dots i_D j_1 \dots j_N} \frac{\partial f_1(\mathbf{u})}{\partial u_{j_1}} \dots \frac{\partial f_N(\mathbf{u})}{\partial u_{j_N}}, \quad (3.24)$$

where $\epsilon^{i_1 \dots i_D j_1 \dots j_N}$ is the antisymmetric Levi-Civita tensor. Next we compute some simple properties of the tangent form.

A. The tangent form is divergence free. This can be seen from the following calculation,

$$\begin{aligned}
\frac{\partial t^{i_1 \dots i_D}(\mathbf{u})}{\partial u_{i_1}} &= \sum_{j_1 \dots j_N} \epsilon^{i_1 \dots i_D j_1 \dots j_N} \frac{\partial^2 f_1(\mathbf{u})}{\partial u_{i_1} \partial u_{j_1}} \dots \frac{\partial f_N(\mathbf{u})}{\partial u_{j_N}} \\
&\dots + \sum_{j_1 \dots j_N} \epsilon^{i_1 \dots i_D j_1 \dots j_N} \frac{\partial f_1(\mathbf{u})}{\partial u_{j_1}} \dots \frac{\partial}{\partial u_{i_1}} \frac{\partial f_N(\mathbf{u})}{\partial u_{j_N}} \\
&= 0
\end{aligned} \tag{3.25}$$

where each term is zero due to the antisymmetry of the Levi-Civita tensor and the symmetry of partial derivatives.

B. For one degree of freedom mechanisms, the tangent form is a vector tangent to the configuration space away from critical points. First, we note that

$$\sum_{i_1} \frac{\partial f_\alpha}{\partial u_{i_1}} t^{i_1 \dots i_D}(\mathbf{u}(s)) = 0 \tag{3.26}$$

which implies that $\frac{\partial f_\alpha}{\partial u_i} t^i(\mathbf{u}(s)) = 0$. Now suppose that $\mathbf{u}(s)$ traces the configuration space in a region where $t^{i_1 \dots i_D}(\mathbf{u}(s))$ is nonzero. Then

$$\sum_i \frac{\partial f_\alpha(\mathbf{u}(s))}{\partial u_i} \frac{\partial u_i(s)}{\partial s} = 0. \tag{3.27}$$

Hence the configuration space is perpendicular to all of the $\partial f_\alpha(\mathbf{u})/\partial u_i$ but $t_i(\mathbf{u})$ is also perpendicular to all of them. Hence, they must be parallel. The more general case for mechanisms with more than one degree of freedom is more subtle but can also be computed.

C. The tangent form is zero at \mathbf{u} if and only if \mathbf{u} is a critical point.

Ultimately, this is a consequence of the fact that the components of $t^{i_1 \dots i_D}(\mathbf{u})$ are the $E \times E$ minors of the Jacobian of $\ell^2(\mathbf{u})$. Nevertheless, we demonstrate it here for completeness. There are E functions

$$\left\{ \frac{\partial f_1(\mathbf{u})}{\partial u_i}, \dots, \frac{\partial f_E(\mathbf{u})}{\partial u_i} \right\}. \tag{3.28}$$

Since the zero modes are defined by the nonzero solutions, δu_i of

$$\sum_i \frac{\partial f_\alpha(\mathbf{u})}{\partial u_i} \delta u_i = 0 \quad (3.29)$$

the zero modes are in the orthogonal complement of the span of the vectors $\partial f_\alpha(\mathbf{u})/\partial u_i$. At a critical point, there must be additional zero modes and so the $\partial f_\alpha(\mathbf{u})/\partial u_i$ span a lower dimensional space and can no longer be linearly independent. Without loss of generality, we can take it to be $\alpha = 1$ so

$$\frac{\partial f_1(\mathbf{u})}{\partial u_i} = \sum_{\beta > 1} c_\beta \frac{\partial f_\beta(\mathbf{u})}{\partial u_i}. \quad (3.30)$$

Substituting this into the definition of $t_{i_1 \dots i_D}(\mathbf{u})$ and using Eq. (3.26), we immediately obtain $t_{i_1 \dots i_D}(\mathbf{u}) = 0$.

Similarly, if $t_{i_1 \dots i_D}(\mathbf{u}) = 0$ then the $\partial f_\alpha(\mathbf{u})/\partial u_i$ cannot all be linearly independent. One way to do see this is to choose D vectors \mathbf{v}_n orthogonal to the $\partial f_\alpha(\mathbf{u})/\partial u_i$ for all α as well as to each other. Then

$$v_{1,i_1} \cdots v_{D,i_D} t_{i_1 \dots i_D}(\mathbf{u}) = \det \begin{pmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_D^T \\ \nabla f_1(\mathbf{u})^T \\ \vdots \\ \nabla f_E(\mathbf{u})^T \end{pmatrix} = 0, \quad (3.31)$$

where ∇ is the gradient in \mathbf{u} and T denotes the transpose. Since the \mathbf{v}_n are orthogonal to the other vectors one of the $\nabla f_\alpha(\mathbf{u})$ must be linearly dependent on the rest of them. We immediately obtain that there is at least one additional linear independent zero mode.

D. Self stresses are orthogonal to the critical value set. The critical set is defined as the set of points \mathbf{u}_C such that $t_{i_1 \dots i_D}(\mathbf{u}_C) = 0$. The critical value set is the image of the critical set under the map $f_\alpha(\mathbf{u}_C)$. Suppose that $\mathbf{u}_C(s)$ is a one-parameter path of points in a smooth portion of the critical set. Then consider its image $F_\alpha(s)$,

$$f_\alpha(\mathbf{u}_C(s)) = F_\alpha(s). \quad (3.32)$$

If the derivative $\partial F_\alpha(s)/\partial s$ is nonzero then it is tangent to the critical value set. Therefore,

$$\frac{\partial F_\alpha(s)}{\partial s} = \sum_i \frac{\partial f_\alpha(\mathbf{u}_C(s))}{\partial u_i} \frac{\partial u_{C,i}(s)}{\partial s}. \quad (3.33)$$

If σ_α is a self stress then $\sum_\alpha \sigma_\alpha \partial f_\alpha / \partial u_I = 0$. Therefore we obtain

$$\sum_\alpha \sigma_\alpha \frac{\partial F_\alpha(s)}{\partial s} = 0. \quad (3.34)$$

Since this is true for any path in the critical value set, it follows that all self stresses are orthogonal to the critical value set.

Though the converse of this is not true – some vectors normal to the critical value set may not be self stresses – if the critical value set has codimension one then there can be only one self stress and the normal vector of the critical value set necessarily corresponds to that self stress.

E. Orientation The tangent form $t_{i_1 \dots i_D}(\mathbf{u})$ carries additional useful geometrical information about the mechanism at regular (non-critical) configurations. When $D = 0$, $t(\mathbf{u})$ is a scalar whose sign was used to compute a topological index in periodic mechanisms [94]. Beyond this, it endows the configuration space with a natural orientation in any dimension. At a regular point on the configuration space of a mechanism, \mathbf{x} , $t_{i_1 \dots i_D}(\mathbf{x}) dx^{i_1} \wedge \dots \wedge dx^{i_D}$ is a differen-

tial form which provides a local orientation: for any basis of tangent vectors $\{\zeta_{1,j}, \dots, \zeta_{D,j}\}$,

$$\text{sgn} \sum_{i_1 \dots i_D} \zeta_{1,i_1} \dots \zeta_{D,i_D} t_{i_1 \dots i_D}(\mathbf{u}) = \pm 1 \quad (3.35)$$

However, note that this local orientation is only defined up to an overall sign, since we can always take one of the constraint functions to have the opposite sign.

Though we do not make a great deal of use of it in this paper, it is worth noting that if one is able to find two regions in which $t^{i_1 \dots i_D}(\mathbf{u})$ has opposite signs, there must be a boundary between those regions for which $t^{i_1 \dots i_D}(\mathbf{u})$ vanishes. That is, in principle we can use the tangent form to verify the existence of critical configurations.

3.2.2 The geometry of the critical configuration set

Now we can apply this to linkages. Since we are interested in understanding how to choose edge lengths, L_α , to control the topology of the configuration space of a linkage, we will consider all possible mechanisms that have the same connectivity but arbitrary values of L_α . Adapting our tangent form to this set of constraints and notation, we arrive at

$$t_{i_1 \dots i_D}(\mathbf{u}) = \sum_{j_1 \dots j_E} \epsilon_{i_1 \dots i_D j_1 \dots j_E} \frac{\partial \ell_1^2(\mathbf{u})}{\partial u_{j_1}} \dots \frac{\partial \ell_E^2(\mathbf{u})}{\partial u_{j_E}}. \quad (3.36)$$

Per item C in Sec. 3.2.1, $t_{i_1 \dots i_D}(\mathbf{u}) = 0$ if and only if \mathbf{u} is a critical point. An alternative explanation for this is because the components of $t^{i_1 \dots i_D}$ are the $E \times E$ minors of the Jacobian matrix. When these all vanish the Jacobian matrix has lower rank. This is of note in the case of linkages because the Jacobian is traditionally used to find critical points in mechanisms. Thus, Eq. (3.36) identifies all possible critical points in mechanisms sharing the same connectivity. Versions of Eq. (3.36) have been studied to identify singularities in robot manipulators [100–104].

The tangent form allows us to define the *critical configuration set* as the locus of points for which

$$t_{i_1 \dots i_D}(\mathbf{u}) = 0. \quad (3.37)$$

In many practical cases, and all of the cases we consider in this paper, it is possible to solve Eq. (3.37) analytically. Note however, that the solutions to Eq. (3.37) only provide the configurations where the Jacobian of the mechanism is not full rank. Therefore, some of the solutions may not satisfy all of our assumptions from Sec. 3.1.2. We conjecture that our assumptions are valid on all but a set of measure zero of the critical configuration set but are not aware of or able to produce a proof of this.

To help understand the geometry of the critical configuration set, we return to our previous example, the planar, four-bar linkage from Fig. 3.2a. In this example, $D = 1$ but $M = 4$ since the mechanism configurations are specified by points (x_1, y_1, x_2, y_2) . If the two pinned vertices are located at $(0, 0)$ and $(a, 0)$ and we restrict $L_\alpha > 0$ (so no bars have zero length), this critical set is described by the two-dimensional manifold of configurations in which all vertices are co-linear, $y_1 = y_2 = 0$.

For one degree of freedom mechanisms ($D = 1$), Eq. (3.36) provides another way of understanding how the configuration space topology changes with changing lengths near a critical point \mathbf{u}_C . In that case, $t_i(\mathbf{u})$ is a vector field everywhere tangent to the zero modes of the mechanism, which follows from the simple fact that it is always orthogonal to the constraints (Sec. 3.2.1 item B). Thus $t_i(\mathbf{u})$ can be thought of as a local vector field whose integral curves trace out curves of constant L_α . That is, when $t_i(\mathbf{u}) \neq 0$, curves of constant L_α can be parameterized by the solutions

$$\frac{du_i(s)}{ds} = t_i[\mathbf{u}(s)]. \quad (3.38)$$

We show this in Fig. 3.2b using arrows pointing along t_i projected onto the rotor angles. Because $t_i(\mathbf{u})$ is divergence-free (Sec. 3.2.1 item A), each branch point has

two arrows pointing in and two arrows pointing out. Note that $t_i(\mathbf{u})$ provides a way to think about the mechanism configuration space as a dynamical system. This dynamical system should not be confused with the motions of the physical mechanism, however, which can move either parallel or antiparallel to $t_i(\mathbf{u})$ equally well. This is also distinct from the dynamical system approach obtained for a periodic (or nearly periodic) mechanism as an iterated map [105, 106].

Eq. (3.38) also provides an intuitive way to understand the hyperbolas formed by the configuration space near branch points that arise from Eq (3.23). We project $t_i(\mathbf{u})$ near \mathbf{u}_C onto the plane spanned by the two zero modes, $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$. Since $t_i(\mathbf{u})$ is tangent to the configuration spaces, we expect the trajectories approach this plane as they approach \mathbf{u}_C . After projection, we obtain a 2D vector field whose components are,

$$T_n(c_1, c_2) = \sum_i \zeta_{n,i} t_i(\mathbf{u}_C + c_1 \boldsymbol{\zeta}_1 + c_2 \boldsymbol{\zeta}_2). \quad (3.39)$$

The integral curves of T_n then trace the projection of the configuration space onto the plane spanned by the zero modes near the branch point.

In this projection, the constant L_α trajectories are quite limited in how they can appear. We know that $T_n(0, 0) = 0$, but because we assume branch points are isolated, the projected tangent vector $T_n(c_1, c_2) \neq 0$ elsewhere. Now suppose that the critical point is a branch point. The projection of the configuration space on the plane of zero modes will have the form of a hyperbolic fixed point, with a stable and unstable manifold associated with the configuration space branches that solve Eq. (3.7) (Fig. 3.1). Thus, we would generically expect the trajectories near the branch point to appear hyperbolic when projected onto the plane of zero modes. Though we do not work with second order rigid points here, these considerations also limit what the trajectories do near such rigid points [99].

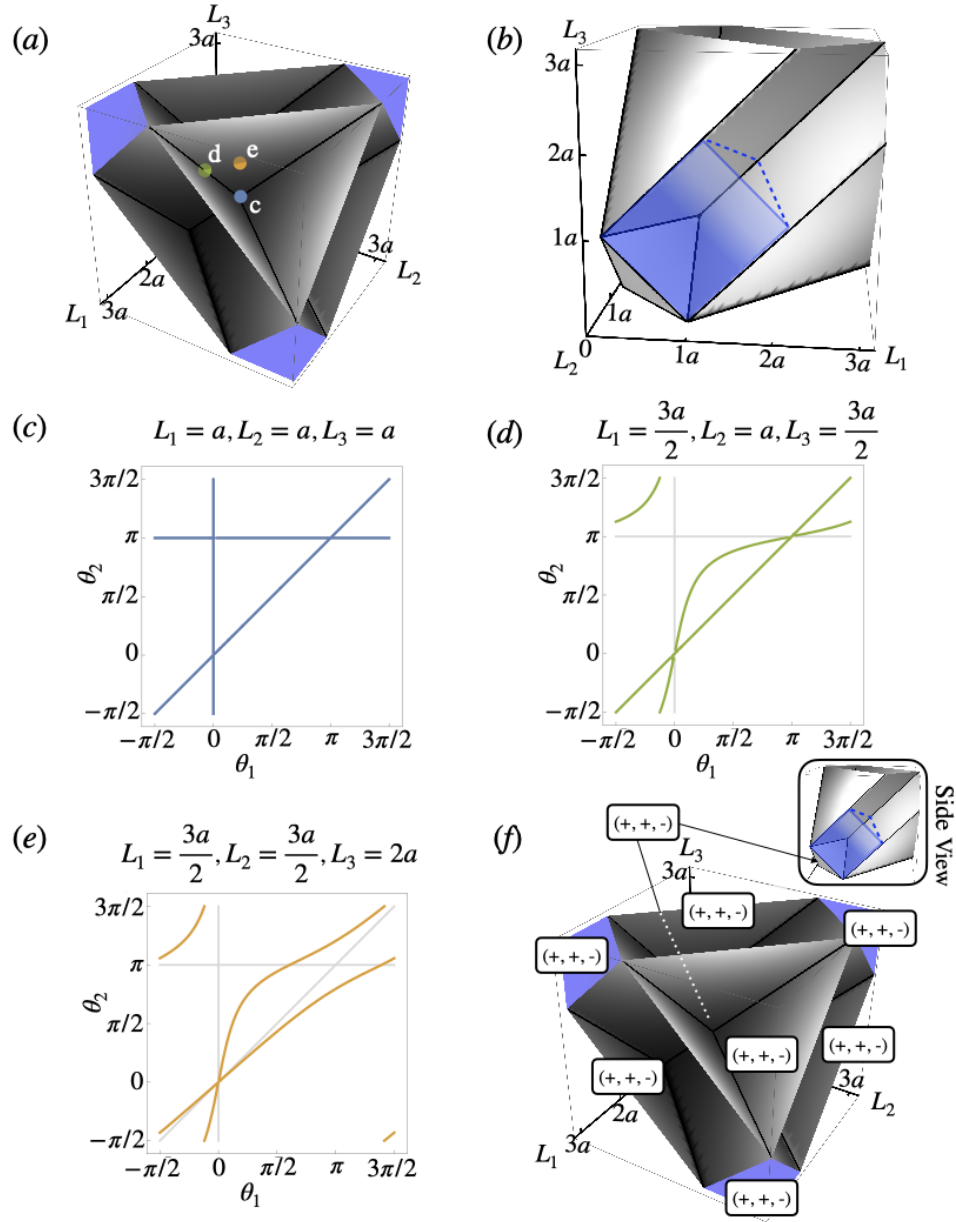


Figure 3.3. (a) The critical value set for the four bar linkage, plotted in terms of (L_1, L_2, L_3) in units of a . The points c - e show the locations of the configuration spaces in (c-e). (b) shows a different view of the surface with a cutout on the $L_1 = 0$ plane showing the shape of one of the enclosed volumes. There is one critical point in the configuration space along any smooth portion of the set (e). Self intersecting lines indicate choices with two critical points (d) and the triply self intersection point at $(L_1, L_2, L_3) = (a, a, a)$ (c) is the unique choice with three critical points. (c-e) The configuration spaces corresponding to the points in (a), showing how to split different branch points. (f) shows the volumes labeled with their branch point types, corresponding to the standard engineering classifications.

3.2.3 The geometry of the critical value set

For any point \mathbf{u}_C in the critical configuration set, $\ell_\alpha^2(\mathbf{u}_C)$ gives its corresponding critical value: the set of squared bar lengths that would be required for the system to be in configuration \mathbf{u}_C . We will call the image of the critical configuration set in the space of squared lengths the *critical value set*. We again illustrate with the four-bar linkage: the set of critical values is a self-intersecting surface $(L_1^2, L_2^2, L_3^2) = (x_1^2, (x_2 - x_1)^2, (a - x_2)^2)$. In Fig. 3.3, we show the critical value set in terms of (L_1, L_2, L_3) rather than the squared lengths to make the surface slightly more compact and easier to understand visually. Note that we have included additional leaves on either the $L_1 = 0$, $L_2 = 0$, or $L_3 = 0$ plane which happen to contain only rigid critical points; this is a natural consequence of the fact that any mechanism with two pinned vertices and one edge having zero length must always be rigid.

Were we to choose the L_α to lie anywhere along the portion of the critical value set in Fig. 3.3, the resulting linkage would have one or more critical points. It is also apparent that Fig. 3.3 self intersects. At such a self-intersection, there will be multiple critical configurations, \mathbf{u}_C , corresponding to the same choices of edge lengths, L_α . Thus, if we choose the L_α along a line of self-intersection, there are two branch points (Fig. 3.3d). If we choose L_α along a smooth portion of the critical value set, there is only one critical point (Fig. 3.3e). Interestingly, Fig. 3.3 shows that at $(L_1, L_2, L_3) = (a, a, a)$ three individual sheets self intersect. Therefore we expect that that choice of L_α is the unique place where three branch points coincide (Fig 3.3c). An animation demonstrating the branch splitting for the 4-bar linkage is included in the Supplementary Material of Ref. [83].

The critical value set contains more information than just the location of critical values. If the critical value set is locally a smooth manifold, the self-stresses at such a critical point are always normal to the critical values (see Sec. 3.2.1 item D). Though the converse is not generally true – normals need not also be self stresses – if the

critical value set is a manifold of codimension one it must necessarily coincide with the single self stress at that point and there can be no other self stresses. We also see that splitting a branch point amounts to choosing δL_α transverse to the critical value set. On one side of the surface in Fig. 3.3, a branch point splits into one pair of smooth branches; on the other side it splits into the opposite pair. This endows the calculation of how branch points split under small perturbations of the lengths with a concise geometrical meaning.

With this understanding of the critical value set, we can classify the distinct configuration spaces of the four-bar linkage in terms of the $2^3 = 8$ individual volumes enclosed by the surface in Fig. 3.3. For completeness, we note that these volumes correspond to standard results for the four-bar linkage found in the engineering literature, which can be classified by the sign of three functions [107]

$$\begin{aligned}\tau_1 &= a - L_1 + L_2 - L_3 \\ \tau_2 &= a - L_1 - L_2 + L_3 \\ \tau_3 &= L_2 + L_3 - a - L_1\end{aligned}\tag{3.40}$$

derived from limits on the angles θ_1 and θ_2 . When one of the τ_i are equal to zero, the configuration space contains the corresponding critical point from Fig. 3.2b. Thus, the critical value set in Fig. 3.3 agrees with the surfaces computed in Ref. [108, 109]. The regions are labeled in Fig. 3f.

3.2.4 Three rotor system

Finally, in this section we will use these considerations to describe a design procedure for configuration space topology. To be concrete, it is helpful to consider a specific example, the three-rotor linkage in Fig. 3.4a. The three rotor linkage has three pinned joints attached to three bars of length r_1 , r_2 , and r_3 (the rotors) and whose opposite ends are joined by bars of length L_1 and L_2 . Therefore, \mathbf{u} is a six

component vector and the five bars provide constraints, $\ell_\alpha^2(\mathbf{u}) = L_\alpha^2$, that limit the configuration space to a single degree of freedom generically.

Since the mechanism has five bars, it is difficult to visualize the critical set and critical value set. Nevertheless, we can still gain insight by restricting ourselves to the cross-section of \mathcal{M} for which $r_1 = r_2 = r_3 = a$. We plot the cross section of the critical value set with the (L_1, L_2) plane in Fig. 3.4b. While this is a cross-section, the open regions in Fig. 3.4b still correspond to structures with different configuration space topologies, with the transitions from one distinct region to another through the critical value set occurring through a branch point. However, it is still a cross section of a higher dimensional space and care must be taken when interpreting the intersections of the critical value set. Choosing $L_1 = L_2 = a$ leads to a configuration space with twelve interconnected branch points, though it appears that only two lines meet at $L_1 = L_2 = a$ in Fig. 3.4b. The proliferation of branch points in this example can be understood from the fact that this linkage contains two pairs of four-bar linkages. Choosing all bars to have length a , therefore, maximizes the branch points of each individual sub-mechanism.

To identify these branch points, we solve $t_i(\mathbf{u}) = 0$ subject to the length constraints $\ell_\alpha^2(\mathbf{u}) = L_\alpha^2$ using Mathematica (Wolfram). At each critical point, we then solve Eq. (3.7) to obtain the tangents to the configuration space. The trajectories in Fig. 3.4c are obtained by first stepping along one of the obtained tangent vectors, then stepping along the configuration space in the direction indicated by $t_i(\mathbf{u})$ with a step size proportional to its magnitude. The step size is adjusted to maintain the edge lengths to less than one percent strain. Finally, the integration for each segment is terminated when the magnitude of $t_i(\mathbf{u})$ falls below a critical threshold, indicating that the integration has reached a point close to the next critical point. Once terminated, we minimize $\sum_i t_i^2$ with respect to the configuration to verify that the integration has

found the next branch point. The directions of integration inherited from $t_i(\mathbf{u})$ are indicated by the arrows in Fig. 3.4c.

Note, however, that the branch points shown in Fig. 3.4c are not all independent. Projecting the configuration space onto the θ_1 - θ_2 plane must give the configuration space of the equivalent four-bar linkage found by ignoring the third rotor. In contrast, removing the first rotor is equivalent to the projection onto the θ_3 - θ_2 plane. Consequently, any branch points that overlap in one of these two projections must, after a deformation, still be identical in projection and such overlapping branches appear or disappear together. From Fig. 3.4, this implies that branch points are paired $\{(1, 2), (3, 5), (4, 6), (7, 8), (9, 11), (10, 12)\}$.

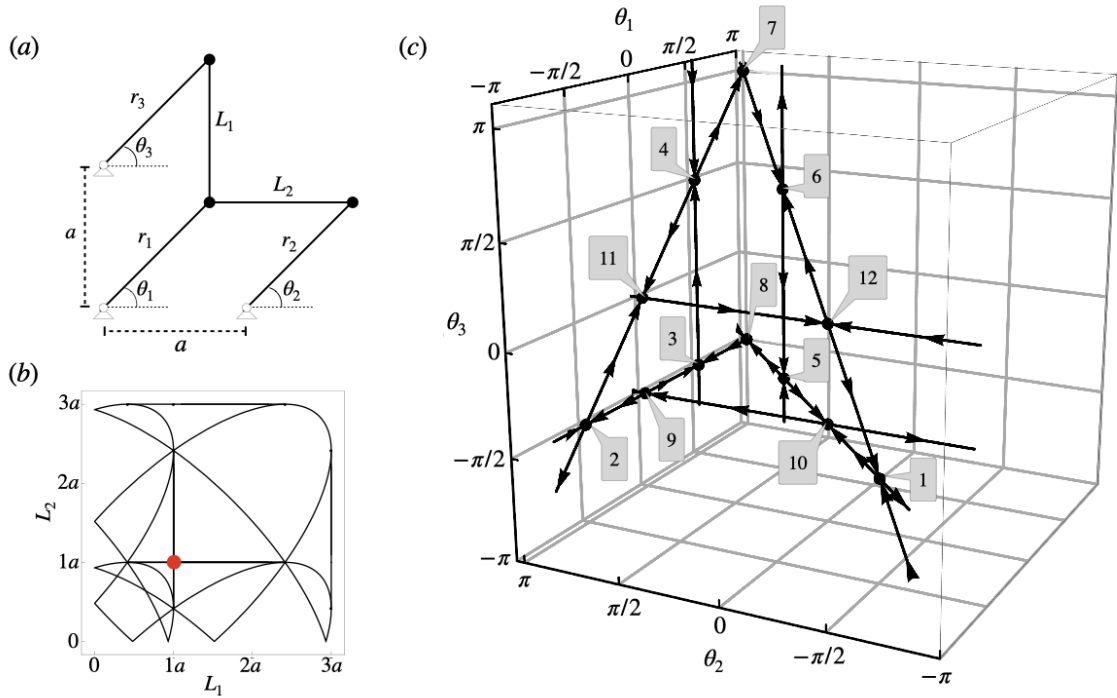


Figure 3.4. (a) Schematic of the planar, three rotor linkage with variables defined. (b) A cross-section of the critical value set for $r_1 = r_2 = r_3 = a$. (c) The 3D configuration space of the three rotor linkage with $r_1 = r_2 = r_3 = L_{13} = L_{23} = a$, corresponding to the red (gray) point in (b), contains twelve individual critical points. Arrows indicate the orientation of each configuration space segment.

We finally consider how to “program” the configuration space by adjusting the lengths of r_1 , L_1 and L_2 away from their critical values. For each critical point, we plot the domain over which $\Delta > 0$ in Fig 3.5 as a table for each branch point. We next choose lengths according to the red (light gray) dot in Fig. 3.5a, which increases r_1 at constant L_1 and L_2 . The resulting configuration space is shown as the red (light gray) curve in Fig. 3.5b. Note that the red (light gray) point was chosen so that the configuration space is smooth but passes near the branch points. If the red (light gray) curve has the topology we want already, we can stop now. If we instead wanted to switch the sign of the branch point pair (4, 6) to obtain a particular configuration space topology. From Fig. 3.5a we see that the three lengths (r_1, L_1, L_2) distinguish this pair of branch points from the rest. Inspection of Fig. 3.5a suggests an additional change in L_1 would switch the way only those two branch points split. The result of this perturbation is the blue (dark gray) curve in Fig. 3.5b. Note that Fig. 3.5b shows that, since each branch point has one self stress, the hyperbolas approach the plane spanned by the two zero modes as expected.

If we limit ourselves to perturbing only the bar lengths (r_1, L_1, L_2) , Fig. 3.5a shows even more redundancy in how the branch points split than expected from our previous analysis that branch points split in pairs. That is, just three control lengths are not sufficient to obtain full control over the way the configuration space splits at the branch points. While it would be difficult to plot Fig. 3.5 using all five bar lengths, there seems to be no mathematical obstacle to generalizing the analysis to distinguish all six pairs of branch points independently.

3.3 Gating the Kane-Lubensky Chain

We finally apply our design methodology to design a mechanism that gates the propagation of a soliton in the Kane-Lubensky (KL) chain [21]. The KL chain is a topologically polarized lattice of rotors that has a zero mode on either the left edge

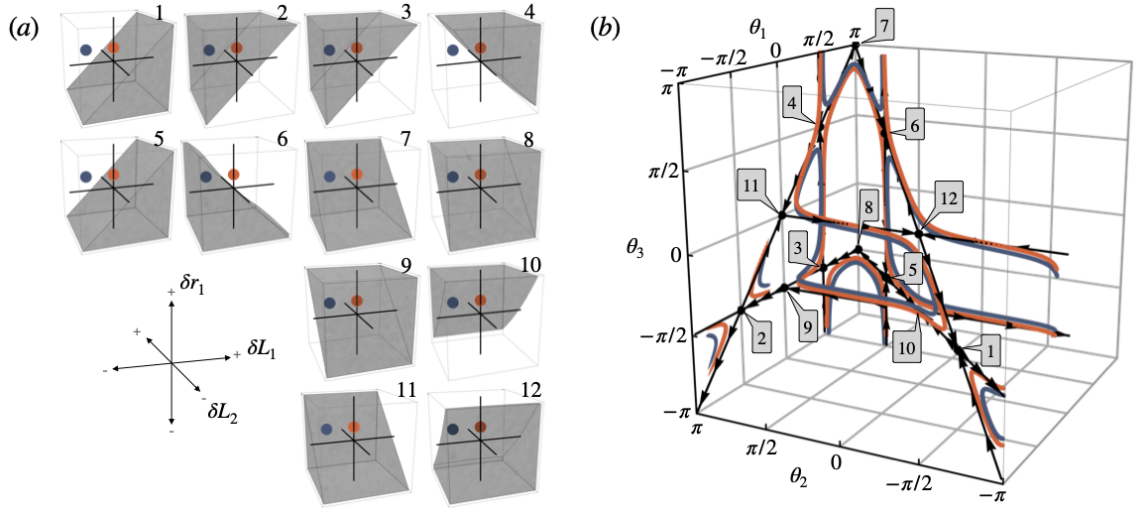


Figure 3.5. A schematic of programming the three rotor system. (a) A map showing how changing the length of r_1 , L_1 , and L_2 leads to different ways to split the branch points from Fig. 3.4. The shaded and unshaded regions correspond to the two ways for the branch points to split. The red (light gray) dot, corresponding to a change in length $r_1 = 1.05a$, leads to the red (light gray) curve in (b). In order to change the topology of the configuration space by changing how branch points 4 and 6 split, an additional change to $L_1 = 0.9a$ can be effected, shown by the blue (dark gray) dot. The new configuration space is shown in (b) as a blue (dark gray) curve.

or the right edge, depending on the choice of bar lengths. It was later discovered that the KL chain actually supported two distinct families of propagating solitons, the “flipper” and the “spinner” [93], that allowed a continuous pathway between the left and right edge modes. The spinner soliton, however, is topologically protected by the shape of the configuration space [93, 94]. In this section, we modify a single unit cell of a spinner-supporting KL chain with rotor length $r = 3a/2$ and $\ell = 3a/2$ by adding an additional two bars and one pinned vertex (Fig. 3.6a).

In the spinner phase of the KL chain, a full cycle consists of the soliton traveling back and forth across the chain once, and the KL chain returning to its initial configuration. After one full cycle, each rotor in the KL chain has rotated by 2π , with each rotor rotating by π each time the soliton passes. Here, we will show that these additional components can act as a gate by opening a gap in the full 2π rotation of the KL chain rotors, thereby obstructing the passage of the soliton.

In addition to the length of the two additional bars, L_1 and L_2 , we also allow the location of the pinned vertex to be set an arbitrary distance D from the KL chain. In order to allow for different positions of the third pinned vertex, we augment \mathbf{u} to include the y coordinate of the 3rd vertex but also augment the constraint functions to pin that vertex’s y position. Thus, we use a constraint map

$$f_\alpha(\mathbf{u}) = \begin{pmatrix} \ell_1^2(\mathbf{u}) \\ \vdots \\ \ell_E^2(\mathbf{u}) \\ D^2(\mathbf{u}) \end{pmatrix}, \quad (3.41)$$

where $D(\mathbf{u})$ is the function that determines the distance between pinned vertex 3 and 2. Thus, the generalized constraints $f_\alpha(\mathbf{u})$ is a smooth function whose solution allows us to pin vertex 3 by setting the length D in Fig. 3.6a to an arbitrary value.

Using this generalized formulation, we can compute a cross section of the critical value set with $r = \ell = 3a/2$ and $L_1 = 2a$ (Fig. 3.6b). Fig. 3.6b shows that there are six distinct regions separated by critical points. The labels on each region correspond to the sign of τ_1 , τ_2 , and τ_3 from Eq. 3.40 with respect to the four-bar linkage between vertices two and three. For concreteness, we choose $L_2 = 2a$ and $D = 5a/2$, on the boundary between the blue (+,+,-) and red (+,+,+) regions, as the initial lengths for our gate, resulting in a configuration space with two critical points (Fig. 3.6d). When $D < 5a/2$, the system is in the red (light gray) regime and when $D > 5a/2$ it is in the blue (dark gray) regime. This choice determines whether the KL chain rotors wind around fully or not. Note that the projection of the configuration space in the θ_1 - θ_2 plane (Fig. 5d) never changes shape, but that the change in how the branch points split into hyperbolas determines whether the full range of angles is accessible to the system or not. To verify that the red (light gray) and blue (dark gray) regimes correspond to ungated and gated behavior of the KL chain device, we use the `mechanisms` package in Mathematica (Wolfram) to calculate the infinitesimal motions of the linkage and animate those motions. As shown in Fig. 3.7, changing D controls whether or not the soliton can complete a full cycle along the KL chain.

To test our design, we constructed the gated KL chain in Fig. 3.6a numerically and constructed a single unit cell and gate from LEGO[™] pieces. The design of the LEGO gate was chosen to be compatible with the LEGO realization of a KL chain shown in [93]. When testing different examples, we pushed on the various bars and rotors in the device to move it through all possible configurations. We tracked how the rotors 1 and 2 moved to determine if the gate was preventing a soliton from propagating. Fig. 3.7 shows a comparison between the simulated and LEGO chains with both D larger and smaller than $5a/2$. Movies of both chains in the gated and

Available on <https://github.com/cdsantan/mechanisms>

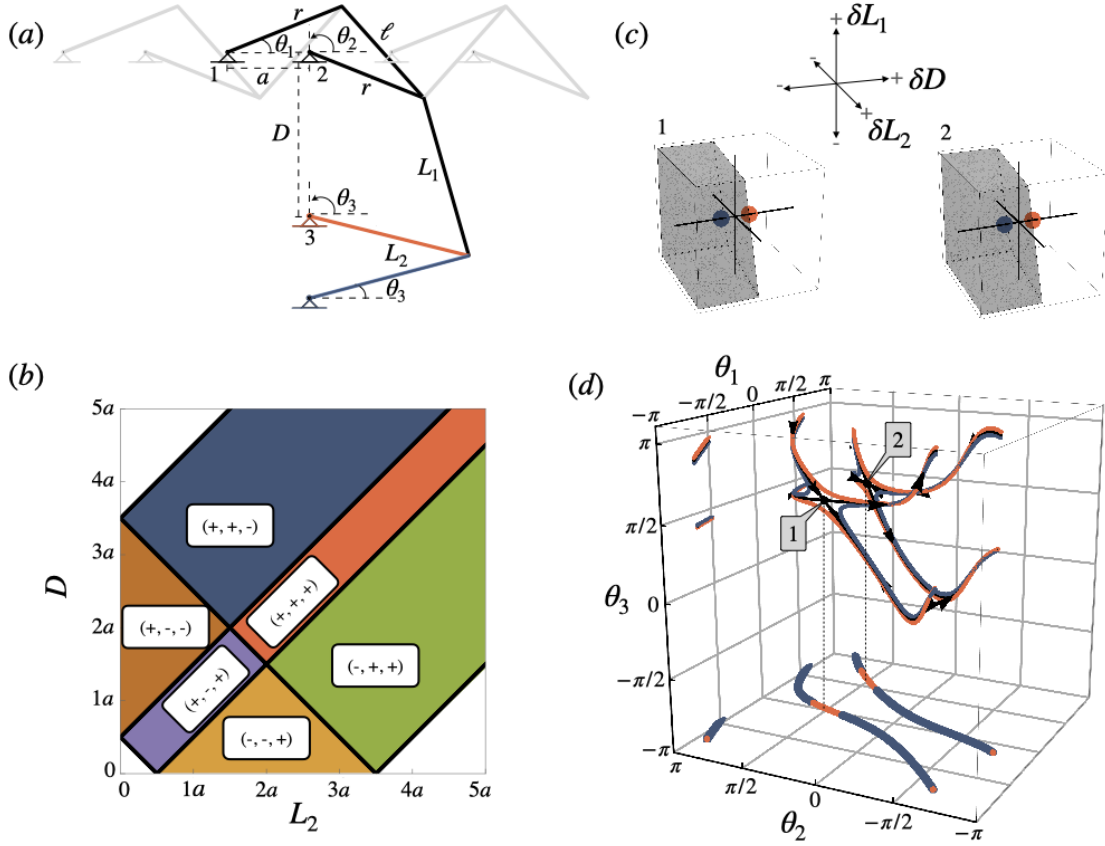


Figure 3.6. (a) A Kane-Lubensky chain with a gating mechanism attached. The gating state is controlled by the length D ; the red (light gray) bar and the blue (dark gray) bar correspond to an ungated and gated state, respectively. (b) A cross-section of the critical set with $r = \ell = 3a/2$ and $L_1 = 2a$. There is a critical point at $L_2 = 2a$ and $D = 5a/2$. (c) At the critical point, changing the position of the third rotor or the lengths of two beams can switch the type of branch split (gray and transparent regions). The red (light gray) point corresponds to branch splitting such that the mechanism is ungated, while the blue (dark gray) point demonstrates that changing D can switch to a gated configuration. (d) The configuration space at and near the critical point as a function of the three rotor angles, and the projection of that configuration space onto the θ_1 - θ_2 plane. The gap in the blue (dark gray) projection shows that the full range of angles is not accessible in that configuration, and thus the chain is gated.

ungated states are provided in the Supplementary Material of Ref[83]. In the case of an ungated chain, the soliton propagates from one end of the chain to the other (and back); for a gated chain the soliton propagates up to the location of the gate but is reflected.

Interestingly, the size of the gap in the projection of the blue (dark gray), gated configuration space in Fig. 3.6d is important for determining how the soliton is reflected from the gate. For very small gaps, which occurs when D is close to its critical value, the soliton can, temporarily, pass the gate but is, ultimately, prevented from completing an entire cycle. For larger gaps, when D is farther from its critical value, the soliton appears to reflect from the gate. From Fig. 3.6c, the same effect can be achieved by changing the size of L_2 instead of D , since the plane dividing the gray and transparent regions, which correspond to the two ways to split the branch, is slightly angled in that direction. Movies of both simulated and LEGO chains that switch between the gated and ungated states by changing L_2 are also provided in the Supplementary Material of Ref [83].

Our analysis shows that the presence of a gap in the (θ_1, θ_2) plane blocks soliton propagation. In the example of Fig. 3.6a, changing the length D moves the device from from the gated (blue, $(+, +, -)$) region to the ungated (red, $(+, +, +)$) region of Fig. 3.6b. However, this is not the only pair of regions that produces a functioning gate. Indeed, the regions indicated in Fig. 3.6b as $(+, +, +)$ (red) and $(-, -, +)$ (yellow) are ungated with respect to propagation of the soliton, whereas the remaining regions are gated. Numerical experiments further show that if we had chosen L_1 to change length as well, we would have found even more regions of both gated and ungated behavior as we extended Fig. 3.6b. It becomes clear that there is a great deal of flexibility when choosing D , L_1 , and L_2 to produce the desired dynamics of the final KL chain and gate system.

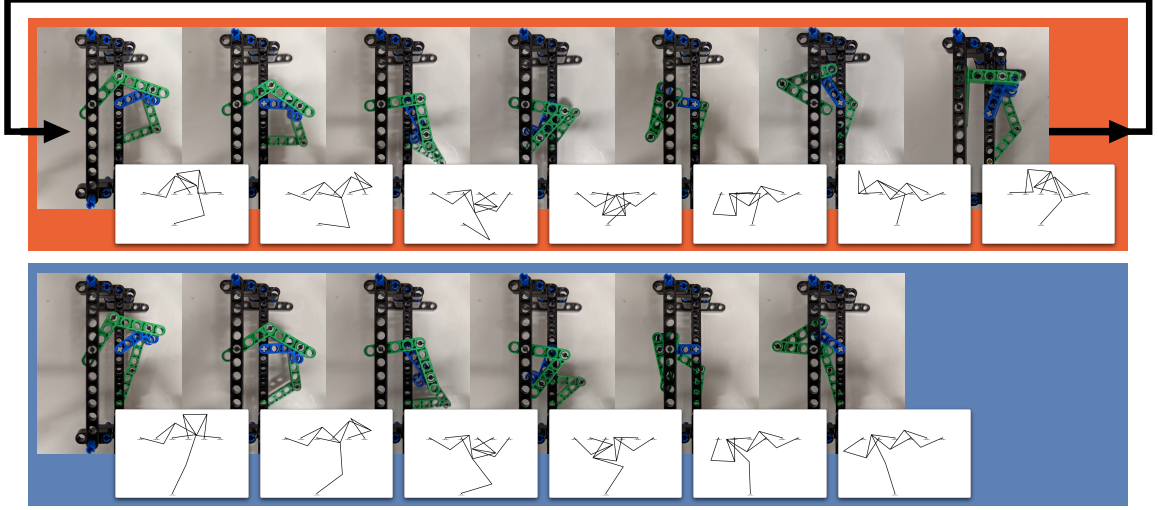


Figure 3.7. Top row (red): Ungated device made from LEGOs with the corresponding simulation. This device can continue rotating and return back to its initial position, as indicated by the arrow. Bottom row (blue): Gated device made from LEGOs with the corresponding simulation. This device gets stuck in the configuration shown in the last frame and is forced to reverse direction in order to continue moving.

3.4 Discussion

In this paper, we have described a procedure to design the topology of the configuration space of mechanical linkage. The idea rests on the ability to identify critical points and, especially, branch points – singular configurations of a linkage in which several pathways meet. By analyzing the shape of the configuration space near these branch points, we are able to design perturbations to the lengths and positions of a fixed set of vertices that change the shape of the topology of the configuration space in well-defined ways. As a demonstration, we used our techniques to design a gate for the propagation of the spinner soliton in a Kane-Lubensky chain. While we applied our approach to linkages with fixed edge length, there is no reason they would not also apply more generally to other systems with holonomic constraints.

Because the design procedure works by controlling configuration space topology, the resulting mechanisms should be quite robust to fabrication errors and the toler-

ance of the joints, so long as one chooses lengths L_α sufficiently far from the critical value set.

It would be interesting to extend this work in a few further directions. First, when bars are no longer rigid but elastic, there arises the possibility of a snap through transition between the different hyperbolas on either side of a branch point. Indeed, tuning various branches close to or farther from a branch point could be used to tune the ease of initiating a snap through transition. This could potentially lead to mechanical structures and mechanical metamaterials whose mechanical response can be reprogrammed *in situ*.

A second interesting extension would be to consider mechanisms built from responsive materials that are sensitive to external stimuli. In that case, the dynamic increase or decrease in the lengths of bars could be used to drive the pathway of a mechanism in an environmentally dependent manner. This could also be affected if the positions of certain pinned vertices could be made to depend on the external environment or the state of a second input mechanism. This would enable the realization of simple mechanical logic that is robust to some damage because it relies only on the topology of a configuration space [110, 111].

Finally, we note that our design principle exploits the fact that the configuration space topology can only change at critical points – configurations where the Jacobian of the constraints fails to be full rank. Our approach is somewhat reminiscent of Morse theory, in which the extrema of a scalar function can be related to the topology of the space on which that function is defined [112]. Morse theory has been used to study the configuration spaces of spherical (and other) linkages [96, 97], but we leave it to future work to make this connection more precise.

CHAPTER 4

CONCLUSIONS AND OUTLOOK

In this dissertation we have provided additional tools for our self-actuating, shape-changing, programmable materials design toolbox. We focused on origami-inspired materials in the first two chapters before moving on to linkages in the last chapter. The tools we presented are only steps in the realization of these materials but can also give us an understanding of why problems arise in design and more generally of the underlying physics across types of materials.

In Chapter 1, we extended existing models of origami-like materials to encompass more than origami with rigid faces and folds that start from a flat sheet of material. We started with adding Gaussian curvature to single vertices and saw that this results in the highly branched configurations spaces characteristic of origami either splitting or merging at the flat state for positive and negative Gaussian curvature, respectively. Next, we expanded the spring model for origami to include elastic components, in the form of both stretching and bending of edges and faces. We saw that the energy landscapes for these origami then consist of deep trenches in place of the branches and observed how changing the programmed angles between branches creates and annihilates minima near the flat state. Since experimental realizations have physical restrictions on how they can be designed and programmed, these created minima play an important role in both understanding why misfolding occurs and how to prevent it.

After simulating elastic origami folding, we were able to show that the affine deformations actually help origami fold correctly, rather than preventing folding to

the target configuration. This can be understood as the origami being able to stretch or bend to pull itself out of one of the deep branch trenches and into another. We had already discussed that the splitting of branches by positive Gaussian curvature could be utilized as a tool to prevent misfolding, but this additional perspective on elastic origami suggests that the joining of branches via negative Gaussian curvature may also be beneficial. By further joining the branches it may be easier for the origami to travel between branches in order to find the global minima. Considering that Gaussian curvature can be induced at an origami vertex by either lengthening or shortening the equilibrium lengths of the vertex's adjoining folds and that this is itself an affine deformation, a more careful study of the interplay of these two phenomena would be a direction for future work.

In Chapter 2 we moved on to two seemingly diverging problems: origami that is forced via torques on the faces rather than the folds and minimal forcing sets. For the face-torqued origami, we developed a matrix equation for converting the torques to forces on the vertices and analyzed this matrix and how to use it to design these origami. Along the way, however, we saw a new way to account for isometries in non-triangular origami faces and the possibility of bending them, which can be tied in to the face bending allowed in the elastic origami of Chapter 1. For minimal forcing sets, we presented an algorithm for finding an upper bound on the minimal forcing set for arbitrary origami which we term a maximal minimal forcing set. By finding a condition for an arbitrary vertex, that all but two folds need to be assigned for any vertex, and converting the origami to a graph, we were able to reduce the problem to that of finding a 2-subgraph across the origami. When conducting simulations to verify that our generated forcing sets truly forced the origami, we realized that this problem can also be connected to the conclusion we reached for elastic origami. If we know a forcing set, then assigning only those folds and setting the remaining folds to be weak allows the weak folds to do whatever is required to reach the target fold

configuration. Our simulation results for minimally programmed origami suggest that this may be another tool to prevent misfolding.

Taking Chapters 1 and 2 altogether, we have a full toolbox for designing a given origami structure. The exact order in which to apply the various tools depends to some extent on the physical realization of the origami. Almost every implementation has some form of limitation, whether it's the precision with which target angles can be programmed to not being able to set a fold to zero to not having independent methods for setting the target fold angle and fold stiffnesses. We will try to address some of these limitations while outlining a method for designing a robust origami. To start, the repeated energy minimization method used in both chapters gives us a way to check the number of possible minima and their relative basin of attraction sizes for a set of target angles, which in turn gives a way to estimate the robustness of the system. If either many minima are seen or the basin of attraction appears small for the desired configuration target, then further engineering is required. If stiffening or weakening faces is possible, we can then repeat this exploration of possible minima with attainable values of face fold stiffnesses, compare the relative number of minima and their basin sizes, and choose the version with the fewest number of minima and largest basin size. If the system can have zero fold angle folds, then we can find the maximal minimal forcing set and test if programming only these folds results increases robustness. If the system can't be programmed directly to the desired target angles then we can probe how different possible cases near the target angles affect the robustness. Regardless, using some to all of these cases can guide us to the best choice of parameters for a given origami.

Turning back to linkages, in Chapter 3 explored how to design a linkage's global, qualitative motions rather than their exact motions. We presented a new formalism for understanding and visualizing constraints by focusing instead on the tangents to these constraints. This allowed us to calculate the entire critical value surfaces,

or those points where the qualitative motion of a linkage changes, i.e. from a rigid structure to one with critical points. This new topological perspective on critical points opens up an entirely new set of techniques to understand how changes in bar lengths change the motion. While we stuck to linkages for the scope of this thesis, the tangent formalism can be generalized to any constraint equations. The critical value surfaces, then, can be understood as a topological description of bifurcations. Much, much more work can (and needs to) be done to fully understand this interplay.

APPENDIX

MATHEMATICA CODE AND DOCUMENTATION

This appendix contains the code and light documentation for the Mathematica functions developed as tools for the projects within this thesis. All of the functions utilize the `mechanisms` package, which can (hopefully, still) be found on GitHub at <https://github.com/cdsantan/mechanisms>. The version used for each set of functions is noted at the start of each appendix section. The package introduces functions for generating mechanisms and origami, which are treated by Mathematica as a collection of `Cell` objects. We will refer to the entire set of `Cell` objects that describe a mechanism as a mechanism object throughout this appendix.

A.1 Code for Chapter 2

The Mathematica code in this section was written using version 0.92 of the `mechanisms` package.

A.1.1 Code for Face-Forced Origami

This section contains the functions for finding the face-vertex connectivity matrix, which relates the torques on the faces of an origami in the flat state to vertical forces on the vertices, and for identifying forces on the vertices that would bend non-triangular faces.

A.1.1.1 `faceCoord[]`

First, we need to be able to find the coordinate system where each face has a coordinate system with its origin at the face's center of mass, found using the built-in

Mathematica function `RegionCentroid[]`. Figs. A.1 and A.2 show two version of the code. Respectively, they take an origami object and a list of faces or just the origami object and return the coordinate system in the form $((x_{1,1}, y_{1,1}), (x_{1,2}, y_{1,2}), \dots), \dots)$ where $(x_{i,j}, y_{i,j})$ are the $x - y$ coordinates of the j th vertex from the center of the i th face. We leave the form example intentionally vague to show that the faces can have any number of edges and the first and second face are not necessarily connected at a vertex.

```
faceCoord[m_, faces_] := Module[{
  pos = MechanismPositions[m],
  centroids
},
  (*Find all of the centroids*)
  centroids = RegionCentroid[Polygon[pos[[#]]]] & /@ faces;
  (*Return the face coordinates in the order of faces given*)
  Table[pos[[faces[[i]]]] - Array[centroids[[i]] # / # &, Length[faces[[i]]], {i, 1, Length[faces]}]]
]
```

Figure A.1. Mathematica function for finding the face coordinates given a mechanism and a set of faces.

```
faceCoord[m_] := Module[{
  pos = MechanismPositions[m],
  faces = MechanismCellData[m][[2, 1, 1]],
  centroids
},
  (*Find all of the centroids*)
  centroids = RegionCentroid[Polygon[pos[[#]]]] & /@ faces;
  (*Return the face coordinates in the order of faces given by MechanismCellData[]*)
  Table[pos[[faces[[i]]]] - Array[centroids[[i]] # / # &, Length[faces[[i]]], {i, 1, Length[faces]}]]
]
```

Figure A.2. Mathematica function for finding the face coordinates given only a mechanism.

A.1.1.2 connectivityMatrix[]

Now, we can use `faceCoord[]` to find the face-vertex connectivity matrix \mathbf{C} , which can be used to relate the torques on the face of an origami, $\vec{\tau}$, to the forces on the edges, \vec{f} , with the relationship $\vec{\tau} = \mathbf{C}\vec{f}$, as explained in more depth in Sec 2.1.1. To do so, we use the geometry of the origami embedded in the face data for an origami mechanism to construct the matrix. The function to do so can be seen in Fig. A.3. It takes a mechanism object and returns the matrix. From here, the built-in Mathematica function `PseudoInverse[]` will return the Moore-Penrose inverse and can be used for further calculations for the origami.

```
connectivityMatrix[m_]:=Module[{
  faces=MechanismCellData[m][[2,1,1]],
  vertices=MechanismCellData[m][[1,1,1]],
  coord=faceCoord[m],vector},
  vector=ConstantArray[0,Length[vertices]];
  (*Assemble the face-vertex connectivity matrix*)
  Flatten[Table[{ReplacePart[vector,AssociationThread[faces[[i]],coord[[i,All,2]]],
    ReplacePart[vector,AssociationThread[faces[[i]],-coord[[i,All,1]]]},
    {i,1,Length[faces]}],1]
]
```

Figure A.3. Mathematica function for finding the face-vertex connectivity matrix of Sec. 2.1.1

A.1.1.3 bendingForces[]

Finally, we can also use `faceCoord[]` to find forces that would bend non-triangular faces as explained in detail in 2.1.2. We can also use these forces to compute a projection operator, which acting on a force vector will project it to one that does not bend any faces. This is done using $\mathbf{P}_B = \mathbf{I}_V - M_B(M_B^T M_B)^{-1} M_B^T$, although note that in the Mathematica function the transposes are swapped because Mathematica treats vectors as rows rather than columns. Fig. A.4 shows this function, which takes

a mechanism object and returns the forces as vectors and the projection matrix as a matrix.

```

bendingForces[m_]:=Module[{
  (*Select only non-triangulated faces*)
  faces=Select[MechanismCellData[m][2,1,1],Length[#3]&],
  vertices=MechanismCellData[m][1,1,1],
  coord,vector,allowedMotions,bendingMotions
},
  coord=faceCoord[m,faces];
  vector=ConstantArray[0,Length[vertices]]; (*sets up the seed vector so every vector will be in vertex space*)
  (*Create a list of the allowed motion vectors for each face. Notes: turn this into a separate function later?*)
  allowedMotions=Table[{
    (*translation vector*)
    ReplacePart[vector,{faces[[i,1]],{faces[[i,2]],{faces[[i,3]],{faces[[i,4]]}}->1},
    (*x-rotation vector*)
    ReplacePart[vector,AssociationThread[faces[[i]]->DeleteCases[DeleteCases[Flatten[Cross[{1,0,0},#]&/@coord[[i]],0,0.]]],
    (*y-rotation vector*)
    ReplacePart[vector,AssociationThread[faces[[i]]->DeleteCases[DeleteCases[Flatten[Cross[{0,1,0},#]&/@coord[[i]],0,0.]]],{i,1,Length[faces]]];
  (*Find the null space for each set of allowed motions for each face, giving us the force vectors that would fold the faces*)
  bendingMotions=Flatten[Table[
    Join[allowedMotions[[i]],ReplacePart[Array[0#&,Length[vertices]],#->1]&/@Complement[MechanismCellData[m][1,1,1],faces[[i]]]]//NullSpace//Chop,
    {i,1,Length[faces]}],1];
  (*And finally construct the projection matrix, outputting the matrix and forces*)
  {IdentityMatrix[Length[vertices]]-(Transpose[bendingMotions].Inverse[bendingMotions.Transpose[bendingMotions]].bendingMotions),bendingMotions}
]

```

Figure A.4. Mathematica function for finding vertical forces that result in non-triangular faces bending.

A.1.2 Code for Finding Maximal Minimum Forcing Sets and Simulations

This section contains the functions used to find the maximal minimal forcing sets and run the simulations in Sec 2.2.

A.1.2.1 maxMinForcingSet[]

This subsection covers the function for finding the maximal minimum forcing set. The function takes an origami object and returns the set of unassigned folds. We start by removing the original boundary vertices of the origami, separating the edges that originally attached to these vertices and giving these edges new ending vertices. The code for this can be seen in Fig. A.5. Next, we expand the internal vertices by removing the original vertices and attaching new vertices to the ends of the edges that originally connected at the old vertex. We then add $d - 2$ additional vertices, where d is the original degree of the vertex. Then we add in the edges to create a bipartite graph with the added vertices and original edges. We then use the

```

maxMinForcingSet[m_] := Module[{
  vertexDegrees = (Length[Cases[MechanismEdges[m], {_, #} | {#, _}]] & /@ InteriorVertices[m]),
  oldOuterInnerEdges = Cases[InteriorEdges[m], Alternatives @@ Flatten[{{#, _}, {_, #}}] & /@ Flatten[BoundaryVertices[m]], 1]],
  innerInnerEdges, newOuterVertices, outerDegrees, outerInnerEdges, intermediateGraph, innerVertexPositions, newInnerVertices,
  finalGraph, maxMatching, maxMatchingOldLabels
],
  (*Start by removing the boundary edges and separating the boundary vertices*)
  innerInnerEdges = DeleteCases[MechanismEdges[m], Alternatives @@ Flatten[{{#, _}, {_, #}}] & /@ Flatten[BoundaryVertices[m]], 1];
  outerDegrees = vertexDegrees - (Length[Cases[innerInnerEdges, Alternatives @@ {{#, _}, {_, #}}]] & /@ InteriorVertices[m]);
  newOuterVertices = Range[Max[MechanismVertices[m]] + 1, Max[MechanismVertices[m]] + Total[outerDegrees]];
  outerInnerEdges =
    Partition[Riffle[Table[Table[InteriorVertices[m][[j]], {i, 1, outerDegrees[j]}], {j, 1, Length[InteriorVertices[m]]}] // Flatten,
      newOuterVertices], 2];
  intermediateGraph = Join[innerInnerEdges, outerInnerEdges];

```

Figure A.5. Part of the Mathematica function for finding the maximal minimal forcing set. This portion

built-in Mathematica function `FindIndependentEdgeSet[]` which finds a maximum matching on a given graph using a blossom algorithm. These steps can be seen in Fig. A.6. For the final step, we need to return the found matching to the original vertex

```

(*Next expand the interior vertices into bipartite graphs. To do so find all instances of the interior vertices and
replace them with new nodes.*)
innerVertexPositions = Flatten[Position[Flatten[intermediateGraph], #]] & /@ InteriorVertices[m];
newInnerVertices =
  TableList[Range[Max[Flatten[intermediateGraph]] + 1,
    Max[Flatten[intermediateGraph]] + Total[Join[vertexDegrees, vertexDegrees - ConstantArray[2, Length[vertexDegrees]]]]],
    Join[vertexDegrees, vertexDegrees - ConstantArray[2, Length[vertexDegrees]]]];
finalGraph =
  Join[Partition[ReplacePart[Flatten[intermediateGraph],
    Table[Thread[innerVertexPositions[[i]] → newInnerVertices[[i]], {i, 1, Length[innerVertexPositions]}] // Flatten], 2],
    Flatten[
      Table[Tuples[{Partition[newInnerVertices, Length[InteriorVertices[m]]][[1, i]],
        Partition[newInnerVertices, Length[InteriorVertices[m]]][[2, i]]], {i, 1, Length[InteriorVertices[m]]}], 1];
    (*And find the maximum matching for the resulting graph*)
    maxMatching = FindIndependentEdgeSet[finalGraph] /. UndirectedEdge[x_, y_] → {x, y};

```

Figure A.6. Part of the Mathematica function for finding the maximal minimal forcing set. This portion

numbering of the origami. We do this by reversing the original steps and deleting the duplicate edges generated. The code for this process can be seen in Fig. A.7

A.1.2.2 programmableAngles[]

Before we can move on to the simulation functions, we need to be able to identify target angles on a branch. To do so, we will generate a set of highly perturbed vertex positions with a distribution width of the smallest fold. After minimizing the energy

```

(*And finally poke it back into the original vertex names and remove the added bipartite edges*)
maxMatchingOldLabels =
maxMatching /.
  Flatten[
    Table[Thread[Join[Partition[newInnerVertices, Length[InteriorVertices[m]]][1, i],
      Partition[newInnerVertices, Length[InteriorVertices[m]][2, i]] → InteriorVertices[m][[i]], {i, 1, Length[InteriorVertices[m]]}]] /.
    Flatten[Thread[#[[1] → #[[2]]] & /@
      Partition[Riffle[TakeList[newOuterVertices, outerDegrees],
        DeleteCases[Flatten[Cases[oldOuterInnerEdges, {#, _} | {_, #}], #] & /@ InteriorVertices[m]], 2]]];
    DeleteCases[maxMatchingOldLabels, Alternatives@@ ({#, #} & /@ InteriorVertices[m])]
  ]

```

Figure A.7. Part of the Mathematica function for finding the maximal minimal forcing set. This portion

of the origami with initial positions from this distribution, we filter the options for those that have at least one angle above a chosen threshold value. The code for this function can be seen in Fig. A.8.

```

programmableAngles[m_, threshold_, n_] := Module[{
  minLength = MechanismCellData[m][[3, 2, 2]] // Min,
  displList, posList, angleList
},
  (*generate the distribution to use as seeds,
  using the minimum bar length as the distribution width*)
  displList = RandomPositions[m, n, "distribution" → UniformDistribution[{-minLength, minLength}]];
  (*feed the distribution to the energy minimizer and extract only complete minimizations*)
  posList = Cases[(MinimizeMechanismEnergy[m, InitialPositions → #, MaxIterations → 10^6] & /@ displList),
    {_Real, _}];
  (*convert the generated minimizations to fold angles*)
  angleList = TorsionalFoldAngle[m, #, InteriorEdges[m]] & /@ posList[;;, 2];
  (*select only cases with large enough angles*)
  Select[angleList, Or @@ (GreaterThan[threshold] /@ Abs[Flatten[#]]) &]
]

```

Figure A.8. Mathematica function to find possible programmable angles for an origami.

A.1.2.3 minProgrammedOri[]

This function takes an origami, a set of angles, and fold stiffnesses for the assigned and unassigned folds and returns an origami object that has the folds from the maximal minimal forcing set programmed to the relevant angles from the set of angles and the rest of the folds programmed to fold to a zero angle. The code for this function can be seen in Fig. A.9.

```

minProgrammedOri[m_, angles_, kassigned_, kunassigned_] := Module[{
  unassignedE = maxMinForcingSet[m],
  inE = InteriorEdges[m],
  assignedE, angleData, positions
},
  (*maxMinForcingSet finds the unassigned angles. Use these to find the assigned angles.*)
  assignedE = Delete[inE, Flatten[Position[inE, # | Reverse[#]] & /@ unassignedE, 1]];
  positions = Position[inE, # | Reverse[#]] & /@ assignedE // Flatten;
  (*create the thread to map angle values to fold assignments*)
  angleData = Flatten[
    {Thread[TorsionalFold[assignedE, {"TorsionalStiffness" → kassigned, "Angle" → #} & /@
      (angles[[#]] & /@ positions)]],
    Thread[TorsionalFold[unassignedE,
      ConstantArray[{"TorsionalStiffness" → kunassigned, "Angle" → 0},
        Length[unassignedE]]]}];
  (*create the origami object*)
  AddCells[m, angleData]
]

```

Figure A.9. Mathematica function to generate origami programmed only with the maximal minimum forcing set.

A.1.2.4 repeatMinimizeOrigami[]

To check the validity of our minimal forcing sets, we then want to check for other minima in the energy of the origami generated by the last function. To do so, we write a function that takes an origami object and a number of minimizations and returns the minimized energies of that many origami. The function generates a distribution of origami perturbed out of the flat state with a width of 20% of the smallest bar length, and uses that distribution as the initial states for the energy minimizations. This function can be seen in Fig. A.10. This version of a repeated minimization is similar to that written for Chapter 1, but uses the conjugate gradient method. Since we are using only one degree of freedom origami for these simulations, this method is sufficient and not overly computationally taxing. Additionally, it better mimics how an origami would fold from the flat state than other methods, such as quasi-Newton methods which can move more freely between origami branches.

```

repeatMinimizeOrigami[m_, n_] := Module[{
  minBar = MechanismCellData[m][[4, 2, 2]] // Min,
  dispList
},
(*generate the distribution to use as initial values,
using a distribution width of 20% of the smallest bar length*)
dispList = RandomPositions[m, n, "distribution" → UniformDistribution[{-0.2*minBar, 0.2*minBar}]];
(*and minimize*)
Cases[
  (MinimizeMechanismEnergy[m, InitialPositions → #, MaxIterations → 10^6,
    Method → "ConjugateGradient"] & /@ dispList), {_Real, _}]
]

```

Figure A.10. Mathematica function to repeatedly minimize an origami structure based on the conjugate gradient method.

A.1.2.5 foldDist[]

To compare the folded structures to the target angles, we write a function that averages over the differences between the two angles. This code can be seen in Fig. A.11

```

foldDist[m_, pos_, ta_] := Mean[(TorsionalFoldAngle[m, pos, InteriorEdges[m]] - ta) ^ 2]

```

Figure A.11. Mathematica function to find the average difference between the fold angles of an origami and a target set of folds angles.

BIBLIOGRAPHY

- [1] M. Berry, M. E. Lee-Trimble, and C. D. Santangelo. Topological transitions in the configuration space of non-Euclidean origami. *Physical Review E*, 101(4): 043003, April 2020. doi: 10.1103/PhysRevE.101.043003. URL <https://link.aps.org/doi/10.1103/PhysRevE.101.043003>. Publisher: American Physical Society.
- [2] M. E. Lee-Trimble, Ji-Hwan Kang, Ryan C. Hayward, and Christian D. Santangelo. Robust folding of elastic origami. *Soft Matter*, 18:6384–6391, 2022. doi: 10.1039/D2SM00369D. URL <http://dx.doi.org/10.1039/D2SM00369D>.
- [3] Stuart C Burgess. A review of linkage mechanisms in animal joints and related bio-inspired designs. *Bioinspiration & Biomimetics*, 2021.
- [4] Gert Roos, Heleen Leysen, Sam Van Wassenbergh, Anthony Herrel, Patric Jacobs, Manuel Dierick, Peter Aerts, and Dominique Adriaens. Linking morphology and motion: a test of a four-bar mechanism in seahorses. *Physiological and Biochemical Zoology*, 82(1):7–19, 2009.
- [5] Bo Liu, Wenjie Ge, Dianbiao Dong, Lei Zheng, and Guoxiong Zhang. Kinematic analysis and optimization of a kangaroo geared five-bar knee joint mechanism. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1557–1563. IEEE, 2017.
- [6] Martial Mazars. Statistical physics of the freely jointed chain. *Physical Review E*, 53(6):6297, 1996.
- [7] Adnan Sljoka. *Algorithms in rigidity theory with applications to protein flexibility and mechanical linkages*. York University Toronto, Ontario, Canada, 2012.
- [8] Adnan Sljoka and Alexandr Bezginov. Predicting protein hinge motions and allostery using rigidity theory. In *AIP Conference Proceedings*, volume 1368, pages 167–170. American Institute of Physics, 2011.
- [9] BM Hespenheide, DJ Jacobs, and MF Thorpe. Structural rigidity in the capsid assembly of cowpea chlorotic mottle virus. *Journal of Physics: Condensed Matter*, 16(44):S5055, 2004.
- [10] Donald J Jacobs, Andrew J Rader, Leslie A Kuhn, and Michael F Thorpe. Protein flexibility predictions using graph theory. *Proteins: Structure, Function, and Bioinformatics*, 44(2):150–165, 2001.

- [11] Shawna Thomas, Xinyu Tang, Lydia Tapia, and Nancy M Amato. Simulating protein motions with rigidity analysis. *Journal of Computational Biology*, 14(6):839–855, 2007.
- [12] Alexander E Marras, Lifeng Zhou, Hai-Jun Su, and Carlos E Castro. Programmable motion of dna origami mechanisms. *Proceedings of the National Academy of Sciences*, 112(3):713–718, 2015.
- [13] Dongsheng Lei, Alexander E Marras, Jianfang Liu, Chao-Min Huang, Lifeng Zhou, Carlos E Castro, Hai-Jun Su, and Gang Ren. Three-dimensional structural dynamics of dna origami bennett linkages using individual-particle electron tomography. *Nature communications*, 9(1):1–8, 2018.
- [14] Bin Liu, Jesse L Silverberg, Arthur A Evans, Christian D Santangelo, Robert J Lang, Thomas C Hull, and Itai Cohen. Topological kinematics of origami metamaterials. *Nature Physics*, 14(8):811–815, 2018.
- [15] Tomohiro Tachi. Simulation of rigid origami. *Origami*, 4(08):175–187, 2009.
- [16] K Liu and GH Paulino. Nonlinear mechanics of non-rigid origami: an efficient computational approach. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2206):20170348, 2017.
- [17] S Sirono. Dilatancy of two-dimensional disk packing. *EPL (Europhysics Letters)*, 96(3):30003, 2011.
- [18] Robert Connelly, Jeffrey D Shen, and Alexander D Smith. Ball packings with periodic constraints. *Discrete & Computational Geometry*, 52(4):754–779, 2014.
- [19] Ojan Khatib Damavandi, Varda F Hagh, Christian D Santangelo, and M Lisa Manning. Energetic rigidity. i. a unifying theory of mechanical stability. *Physical Review E*, 105(2):025003, 2022.
- [20] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin Van Hecke. Flexible mechanical metamaterials. *Nature Reviews Materials*, 2(11):1–11, 2017.
- [21] CL Kane and TC Lubensky. Topological boundary modes in isostatic lattices. *Nature Physics*, 10(1):39–45, 2014.
- [22] Matthias Merkel, Karsten Baumgarten, Brian P Tighe, and M Lisa Manning. A minimal-length approach unifies rigidity in underconstrained materials. *Proceedings of the National Academy of Sciences*, 116(14):6560–6568, 2019.
- [23] Alfred B Kempe. On a general method of describing plane curves of the nth degree by linkwork. *Proceedings of the London Mathematical Society*, 1(1):213–216, 1875.
- [24] Zijia Li, Josef Schicho, and Hans-Peter Schröcker. Kempe’s universality theorem for rational space curves. *Foundations of Computational Mathematics*, 18(2):509–536, 2018.

- [25] Matteo Gallet, Christoph Koutschan, Zijia Li, Georg Regensburger, Josef Schicho, and Nelly Villamizar. Planar linkages following a prescribed motion. *Mathematics of Computation*, 86(303):473–506, 2017.
- [26] Kazuko Fuchi, Alejandro R Diaz, Edward J Rothwell, Raoul O Ouedraogo, and Junyan Tang. An origami tunable metamaterial. *Journal of Applied Physics*, 111(8):084905, 2012.
- [27] Mark Schenk and Simon D. Guest. Geometry of Miura-folded metamaterials. *Proceedings of the National Academy of Sciences*, 110(9):3276–3281, February 2013. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1217998110. URL <http://www.pnas.org/lookup/doi/10.1073/pnas.1217998110>.
- [28] Z. Y. Wei, Z. V. Guo, L. Dudte, H. Y. Liang, and L. Mahadevan. Geometric Mechanics of Periodic Pleated Origami. *Physical Review Letters*, 110(21):215501, May 2013. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.110.215501. URL <https://link.aps.org/doi/10.1103/PhysRevLett.110.215501>.
- [29] Jesse L. Silverberg, Arthur A. Evans, Lauren McLeod, Ryan C. Hayward, Thomas Hull, Christian D. Santangelo, and Itai Cohen. Using origami design principles to fold reprogrammable mechanical metamaterials. *Science*, 345(6197):647–650, August 2014. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1252876. URL <https://science-sciencemag-org.silk.library.umass.edu/content/345/6197/647>. Publisher: American Association for the Advancement of Science Section: Report.
- [30] Toen Castle, Yigil Cho, Xingting Gong, Euiyeon Jung, Daniel M Sussman, Shu Yang, and Randall D Kamien. Making the cut: Lattice kirigami rules. *Physical review letters*, 113(24):245502, 2014.
- [31] Ingo Burgert and Peter Fratzl. Actuation systems in plants as prototypes for bioinspired devices. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1893):1541–1557, April 2009. doi: 10.1098/rsta.2009.0003. URL <https://royalsocietypublishing-org.silk.library.umass.edu/doi/full/10.1098/rsta.2009.0003>. Publisher: Royal Society.
- [32] Kazuya Saito, Shuhei Nomura, Shuhei Yamamoto, Ryuma Niiyama, and Yoji Okabe. Investigation of hindwing folding in ladybird beetles by artificial elytron transplantation and microcomputed tomography. *Proceedings of the National Academy of Sciences*, 114(22):5624–5628, May 2017. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1620612114. URL <https://www.pnas.org/content/114/22/5624>. Publisher: National Academy of Sciences Section: Biological Sciences.

- [33] Kaori Kuribayashi-Shigetomi, Hiroaki Onoe, and Shoji Takeuchi. Cell Origami: Self-Folding of Three-Dimensional Cell-Laden Microstructures Driven by Cell Traction Force. *PLoS ONE*, 7(12), December 2012. ISSN 1932-6203. doi: 10.1371/journal.pone.0051085. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3521028/>.
- [34] Kaori Kuribayashi, Koichi Tsuchiya, Zhong You, Dacian Tomus, Minoru Umemoto, Takahiro Ito, and Masahiro Sasaki. Self-deployable origami stent grafts as a biomedical application of Ni-rich TiNi shape memory alloy foil. *Materials Science and Engineering: A*, 419(1):131–137, March 2006. ISSN 0921-5093. doi: 10.1016/j.msea.2005.12.016. URL <http://www.sciencedirect.com/science/article/pii/S0921509305014930>.
- [35] Koryo Miura. Method of Packaging and Deployment of Large Membranes in Space. *The Institute of Space and Astronautical Science report*, 618:1–9, December 1985. ISSN 02856808. URL <https://ci.nii.ac.jp/naid/110000029131/>. Publisher: Japan Aerospace Exploration Agency.
- [36] Arda Kotikian, Connor McMahan, Emily C. Davidson, Jalilah M. Muhammad, Robert D. Weeks, Chiara Daraio, and Jennifer A. Lewis. Untethered soft robotic matter with passive control of shape morphing and propulsion. *Science Robotics*, 4(33), August 2019. ISSN 2470-9476. doi: 10.1126/scirobotics.aax7044. URL <https://robotics.sciencemag.org/content/4/33/eaax7044>. Publisher: Science Robotics Section: Research Article.
- [37] Cagdas D. Onal, Robert J. Wood, and Daniela Rus. An Origami-Inspired Approach to Worm Robots. *IEEE/ASME Transactions on Mechatronics*, 18(2):430–438, April 2013. ISSN 1941-014X. doi: 10.1109/TMECH.2012.2210239. Conference Name: IEEE/ASME Transactions on Mechatronics.
- [38] Origami simulator. <https://origamisimulator.org>. Accessed: 2021-06-29.
- [39] Tomohiro Tachi and Thomas C. Hull. Self-Foldability of Rigid Origami. *Journal of Mechanisms and Robotics*, 9(2), April 2017. ISSN 1942-4302. doi: 10.1115/1.4035558. URL <https://asmedigitalcollection.asme.org/mechanismsrobotics/article/9/2/021008/473062/Self-Foldability-of-Rigid-Origami>. Publisher: American Society of Mechanical Engineers Digital Collection.
- [40] Tomohiro Tachi. Origamizer. URL <http://origami.cu-tokyo.ac.jp/~tachi/softw are>, 2008.
- [41] Bryan Gin-ghe Chen and Christian D. Santangelo. Branches of Triangulated Origami Near the Unfolded State. *Physical Review X*, 8(1):011034, February 2018. doi: 10.1103/PhysRevX.8.011034. URL <https://link.aps.org/doi/10.1103/PhysRevX.8.011034>. Publisher: American Physical Society.

- [42] Menachem Stern, Matthew B. Pinson, and Arvind Murugan. The Complexity of Folding Self-Folding Origami. *Physical Review X*, 7(4):041070, December 2017. doi: 10.1103/PhysRevX.7.041070. URL <https://link.aps.org/doi/10.1103/PhysRevX.7.041070>. Publisher: American Physical Society.
- [43] Mark Schenk and Simon D Guest. Origami folding: A structural engineering approach. *Origami*, 5:291–304, 2011.
- [44] Levi H. Dudte, Etienne Vouga, Tomohiro Tachi, and L. Mahadevan. Programming curvature using origami tessellations. *Nature Materials*, 15(5):583–588, May 2016. ISSN 1476-4660. doi: 10.1038/nmat4540. URL <http://www.nature.com/articles/nmat4540>. Number: 5 Publisher: Nature Publishing Group.
- [45] E. T. Filipov, K. Liu, T. Tachi, M. Schenk, and G. H. Paulino. Bar and hinge models for scalable analysis of origami. *International Journal of Solids and Structures*, 124:26–45, October 2017. ISSN 0020-7683. doi: 10.1016/j.ijsolstr.2017.05.028. URL <http://www.sciencedirect.com/science/article/pii/S0020768317302408>.
- [46] Matthew B. Pinson, Menachem Stern, Alexandra Carruthers Ferrer, Thomas A. Witten, Elizabeth Chen, and Arvind Murugan. Self-folding origami at any energy scale. *Nature communications*, 8(15477), 2017.
- [47] Jun-Hee Na, Arthur A. Evans, Jinhye Bae, Maria C. Chiappelli, Christian D. Santangelo, Robert J. Lang, Thomas C. Hull, and Ryan C. Hayward. Programming Reversibly Self-Folding Origami with Micropatterned Photo-Crosslinkable Polymer Trilayers. *Advanced Materials*, 27(1):79–85, 2015. ISSN 1521-4095. doi: 10.1002/adma.201403510. URL <http://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201403510>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201403510>.
- [48] Brandon H. Hanna, Jason M. Lund, Robert J. Lang, Spencer P. Magleby, and Larry L. Howell. Waterbomb base: a symmetric single-vertex bistable origami mechanism. *Smart Materials and Structures*, 23(9):094009, August 2014. ISSN 0964-1726. doi: 10.1088/0964-1726/23/9/094009. URL <https://doi.org/10.1088/0964-1726/23/9/094009>. Publisher: IOP Publishing.
- [49] Scott Waitukaitis. Origami Multistability: From Single Vertices to Metasheets. *Physical Review Letters*, 114(5), 2015. doi: 10.1103/PhysRevLett.114.055503.
- [50] Jesse L. Silverberg, Jun-Hee Na, Arthur A. Evans, Bin Liu, Thomas C. Hull, Christian D. Santangelo, Robert J. Lang, Ryan C. Hayward, and Itai Cohen. Origami structures with a critical transition to bistability arising from hidden degrees of freedom. *Nature Materials*, 14(4):389–393, April 2015. ISSN 1476-4660. doi: 10.1038/nmat4232. URL <https://www.nature.com/articles/nmat4232/>. Number: 4 Publisher: Nature Publishing Group.

- [51] Ji-Hwan Kang, Hyunki Kim, Christian D. Santangelo, and Ryan C. Hayward. Enabling Robust Self-Folding Origami by Pre-Biasing Vertex Buckling Direction. *Advanced Materials*, 31(39):0193006, 2019. ISSN 1521-4095. doi: 10.1002/adma.201903006. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201903006>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201903006>.
- [52] Menachem Stern, Viraj Jayaram, and Arvind Murugan. Shaping the topology of folding pathways in mechanical systems. *Nature Communications*, 9(1):4303, October 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-06720-1. URL <https://www.nature.com/articles/s41467-018-06720-1>. Number: 1 Publisher: Nature Publishing Group.
- [53] Paul Plucinsky, Benjamin A Kowalski, Timothy J White, and Kaushik Bhattacharya. Patterning nonisometric origami in nematic elastomer sheets. *Soft matter*, 14(16):3127–3134, 2018.
- [54] Milton R Garza, Edwin A Peraza-Hernandez, and Darren J Hartl. Self-folding origami surfaces of non-zero gaussian curvature. In *Behavior and Mechanics of Multifunctional Materials XIII*, volume 10968, page 109680R. International Society for Optics and Photonics, 2019.
- [55] Roger C Alperin, Barry Hayes, and Robert J Lang. Folding the hyperbolic crane. *The Mathematical Intelligencer*, 34(2):38–49, 2012.
- [56] Nakul Prabhakar Bende, Arthur A Evans, Sarah Innes-Gold, Luis A Marin, Itai Cohen, Ryan C Hayward, and Christian D Santangelo. Geometrically controlled snapping transitions in shells with curved creases. *Proceedings of the National Academy of Sciences*, 112(36):11175–11180, 2015.
- [57] HS Seung and David R Nelson. Defects in flexible membranes with crystalline order. *Physical Review A*, 38(2):1005, 1988.
- [58] Hugo Akitaya, Erik D. Demaine, Takashi Horiyama, Thomas C. Hull, Jason S. Ku, and Tomohiro Tachi. Rigid Foldability is NP-Hard. *arXiv:1812.01160 [cs, math]*, December 2018. URL <http://arxiv.org/abs/1812.01160>. arXiv: 1812.01160.
- [59] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.
- [60] Robert Connelly. The rigidity of certain cabled frameworks and the second order rigidity of arbitrarily triangulated convex surfaces. *Advances in Mathematics*, 37:272–299, 1980.
- [61] TC Lubensky, CL Kane, Xiaoming Mao, Anton Souslov, and Kai Sun. Phonons and elasticity in critically coordinated lattices. *Reports on Progress in Physics*, 78(7):073901, 2015.

- [62] Michael Kapovich and John J Millson. Hodge theory and the art of paper folding. *Publications of the Research Institute for Mathematical Sciences*, 33 (1):1–31, 1997.
- [63] Ileana Streinu and Walter Whiteley. Single-vertex origami and spherical expansive motions. In *Japanese Conference on Discrete and Computational Geometry*, pages 161–173. Springer, 2004.
- [64] Scott Waitukaitis and Martin van Hecke. Origami building blocks: Generic and special four-vertices. *Physical Review E*, 93(2):023003, 2016.
- [65] Scott Waitukaitis, Peter Dieleman, and Martin van Hecke. Non-Euclidean Origami. pages 1–8, 2019. URL <http://arxiv.org/abs/1909.13674>.
- [66] Robert J Lang. Tessellatica, 2019. URL <https://langorigami.com/article/tessellatica/>.
- [67] Samuel Randlett. *The Flapping Bird*. Magic, INC, January 1969. ISBN B000LXQVWC.
- [68] Amanda Ghassaei, Erik D Demaine, and Neil Gershenfeld. Fast, interactive origami simulation using gpu computation. *Origami*, 7:1151–1166, 2018.
- [69] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.
- [70] Addison Liu, Mykell Johnson, and Cynthia Sung. Increasing Reliability of Self-Folding of the Origami HyPar. *Journal of Mechanisms and Robotics*, 14(6), 04 2022. ISSN 1942-4302. doi: 10.1115/1.4054310. URL <https://doi.org/10.1115/1.4054310>. 060904.
- [71] Charlotte Py, Paul Reverdy, Lionel Doppler, José Bico, Benoît Roman, and Charles N. Baroud. Capillary origami: Spontaneous wrapping of a droplet with an elastic sheet. *Phys. Rev. Lett.*, 98:156103, Apr 2007. doi: 10.1103/PhysRevLett.98.156103. URL <https://link.aps.org/doi/10.1103/PhysRevLett.98.156103>.
- [72] Elliot Hawkes, B An, Nadia M Benbernou, H Tanaka, Sangbae Kim, Erik D Demaine, D Rus, and Robert J Wood. Programmable matter by folding. *Proceedings of the National Academy of Sciences*, 107(28):12441–12445, 2010.
- [73] Michael F. Reynolds, Alejandro J. Cortese, Qingkun Liu, Zhangqi Zheng, Wei Wang, Samantha L. Norris, Sunwoo Lee, Marc Z. Miskin, Alyosha C. Molnar, Itai Cohen, and Paul L. McEuen. Microscopic robots with onboard digital control. *Science Robotics*, 7(70):eabq2296, 2022. doi: 10.1126/scirobotics.abq2296. URL <https://www.science.org/doi/abs/10.1126/scirobotics.abq2296>.

- [74] Hanyu Alice Zhang, Chrisy Xiyu Du, Tanner Pearson, Conrad Smart, Zexi Liang, Michael P Brenner, Paul L McEuen, and Itai Cohen. Magnetic handshake polymers with designable properties. APS March Meeting, 2022.
- [75] Adi Ben-Isreal and Thomas N. E. Greville. *Generalized inverses: theory and applications*. John Wiley and Sons, 1974.
- [76] Brad Ballinger, Mirela Damian, David Eppstein, Robin Y. Flatland, Jessica Ginepro, and Thomas C. Hull. Minimum forcing sets for miura folding patterns. *CoRR*, abs/1410.2231, 2014. URL <http://arxiv.org/abs/1410.2231>.
- [77] Mirela Damian, Erik D. Demaine, Muriel Dulieu, Robin Y. Flatland, Hella Hoffman, Thomas C. Hull, Jayson Lynch, and Suneeta Ramaswami. Minimal forcing sets for 1d origami. *CoRR*, abs/1703.06373, 2017. URL <http://arxiv.org/abs/1703.06373>.
- [78] Zachary Abel, Jason Cantarella, Erik D. Demaine, David Eppstein, Thomas C. Hull, Jason S. Ku, Robert J. Lang, and Tomohiro Tachi. Rigid origami vertices: Conditions and forcing sets. *JoCG*, 7(1):171–184, 2016. URL https://erikdemaine.org/papers/RigidOrigami_JoCG/paper.pdf.
- [79] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, September 2000. ISBN 0130144002.
- [80] David Hartvigsen. Finding maximum square-free 2-matchings in bipartite graphs. *Journal of Combinatorial Theory, Series B*, 96(5):693–705, 2006. ISSN 0095-8956. doi: <https://doi.org/10.1016/j.jctb.2006.01.004>. URL <https://www.sciencedirect.com/science/article/pii/S0095895606000050>.
- [81] W. T. Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics*, 6:347–352, 1954. doi: 10.4153/CJM-1954-033-3.
- [82] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi: 10.4153/CJM-1965-045-4.
- [83] M. Berry, David Limberg, M. E. Lee-Trimble, Ryan Hayward, and C. D. Santangelo. Controlling the configuration space topology of mechanical structures. *Phys. Rev. E*, 106:055002, Nov 2022. doi: 10.1103/PhysRevE.106.055002. URL <https://link.aps.org/doi/10.1103/PhysRevE.106.055002>.
- [84] P Kumar and S Pellegrino. Computation of kinematic paths and bifurcation points. *International Journal of Solids and Structures*, 37(46-47):7003–7027, 2000.
- [85] Igor Fernández de Bustos, Josu Aguirrebeitia, Rafael Avilés, and Rubén Ansolá. Second order mobility analysis of mechanisms using closure equations. *Meccanica*, 47(7):1695–1704, 2012.

- [86] Andreas Müller and Dimitar Zlatanov. *Singular Configurations of Mechanisms and Manipulators*. Springer, 2019.
- [87] Robert Connelly and Herman Servatius. Higher-order rigidity—what is the proper definition? *Discrete & Computational Geometry*, 11(2):193–200, 1994.
- [88] Robert Connelly and Walter Whiteley. Second-order rigidity and prestress stability for tensegrity frameworks. *SIAM Journal on Discrete Mathematics*, 9(3):453–491, 1996.
- [89] Andreas Müller. Higher-order analysis of kinematic singularities of lower pair linkages and serial manipulators. *Journal of Mechanisms and Robotics*, 10(1), 2018.
- [90] Ralph C Merkle, Robert A Freitas, Tad Hogg, Thomas E Moore, Matthew S Moses, and James Ryley. Mechanical computing systems using only links and rotary joints. *Journal of Mechanisms and Robotics*, 10(6):061006, 2018.
- [91] Karl Wohlhart. Kinematotropic linkages. In *Recent advances in robot kinematics*, pages 359–368. Springer, 1996.
- [92] Carlo Galletti and Pietro Fanghella. Single-loop kinematotropic mechanisms. *Mechanism and Machine Theory*, 36(6):743–761, 2001.
- [93] Bryan Gin-gé Chen, Nitin Upadhyaya, and Vincenzo Vitelli. Nonlinear conduction via solitons in a topological mechanical insulator. *Proceedings of the National Academy of Sciences*, 111(36):13004–13009, 2014.
- [94] Po-Wei Lo, Christian D Santangelo, Bryan Gin-gé Chen, Chao-Ming Jian, Krishanu Roychowdhury, and Michael J Lawler. Topology in nonlinear mechanical systems. *Physical Review Letters*, 127(7):076802, 2021.
- [95] Josef Schicho. And yet it moves: Paradoxically moving linkages in kinematics. *Bulletin of the American Mathematical Society*, 59(1):59–95, 2022.
- [96] Michael Kapovich and John Millson. On the moduli space of polygons in the euclidean plane. *Journal of Differential Geometry*, 42(1):133–164, 1995.
- [97] Michael Kapovich and John J Millson. On the moduli space of a spherical polygonal linkage. *Canadian Mathematical Bulletin*, 42(3):307–320, 1999.
- [98] Andreas Müller. Kinematic tangent cone—a useful concept for the local mobility and singularity analysis. In *IFTToMM World Congress on Mechanism and Machine Science*, pages 337–346. Springer, 2019.
- [99] Miranda Holmes-Cerfon, Louis Theran, and Steven J Gortler. Almost-rigidity of frameworks. *Communications on Pure and Applied Mathematics*, 74(10):2185–2247, 2021.

- [100] Krzysztof Tchoń. Singularity avoidance in robotic manipulators: A differential form approach. *Robotica*, 13(6):599–606, 1995.
- [101] Krzysztof Tchoń. Singularity avoidance in robotic manipulators: A differential form approach. *Systems & control letters*, 30(4):165–176, 1997.
- [102] H. Lipkin and E. Pohl. Enumeration of singular configurations for robotic manipulators. *Journal of Mechanical Design*, 113(3):272–279, 1991.
- [103] D Zlatanov, RG Fenton, and B Benhabib. A unifying framework for classification and interpretation of mechanism singularities. *Journal of mechanical design*, 117(4):566–572, 1995.
- [104] Peter S Donelan. Singularity-theoretic methods in robot kinematics. *Robotica*, 25(6):641–659, 2007.
- [105] Jason Z Kim, Zhixin Lu, Ann S Blevins, and Dani S Bassett. Nonlinear dynamics and chaos in conformational changes of mechanical metamaterials. *Physical Review X*, 12(1):011042, 2022.
- [106] Jason Z Kim, Zhixin Lu, Steven H Strogatz, and Danielle S Bassett. Conformational control of mechanical networks. *Nature Physics*, 15(7):714–720, 2019.
- [107] J. Michael McCarthy and Gim Song Soh. *Geometric Design of Linkages*. Interdisciplinary Applied Mathematics. Springer New York, 2010. ISBN 9781441978929.
- [108] Clark R Barker. A complete classification of planar four-bar linkages. *Mechanism and Machine Theory*, 20(6):535–554, 1985.
- [109] Mees Muller. A novel classification of planar four-bar linkages and its application to the mechanical analysis of animal systems. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 351 1340:689–720, 1996.
- [110] Yuanping Song, Robert M Panas, Samira Chizari, Lucas A Shaw, Julie A Jackson, Jonathan B Hopkins, and Andrew J Pascall. Additively manufacturable micro-mechanical logic gates. *Nature communications*, 10(1):1–6, 2019.
- [111] Zhiqiang Meng, Weitong Chen, Tie Mei, Yuchen Lai, Yixiao Li, and CQ Chen. Bistability-based foldable origami mechanical logic gates. *Extreme Mechanics Letters*, 43:101180, 2021.
- [112] Liviu I Nicolaescu et al. *An invitation to Morse theory*. Springer, 2007.