

DISS. ETH NO. 26514

**A Computational Method for the Synthesis of Rigid Origami Crease Patterns**

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZÜRICH

(Dr. sc. ETH Zürich)

Presented by

LUCA ZIMMERMANN

Master of Science ETH, ETH Zürich

Bachelor of Science ETH, ETH Zürich

Born on 04.02.1991

Citizen of Zürich, Switzerland

Accepted on the recommendation of

Prof. Dr. Kristina Shea

Dr. Tino Stanković

Prof. Dr. Evgueni Filipov

2020



## Abstract

While origami is an ancient art form, its application in engineering science has only been popularized in recent decades when the scientific community recognized its numerous benefits. The benefits and the widespread applicability of origami are accompanied by a set of geometric and kinematic challenges involving rigid foldability, an exponential number of Rigid Body Modes (RBMs), as well as complex relations between the kinematic determinacy, the number of Degrees-Of-Freedom (DOF), and symmetry. These challenges complicate the adoption of origami principles for scientific purposes, which led to the development of various computational methods in related works that support the application of origami in engineering design tasks. However, most of these methods isolate and address specific challenges and focus on the adaptation of existing crease patterns rather than the design of novel crease patterns, leading to today's design process that is tedious, time-consuming, and limited to a handful of experienced scientists. This gap motivates the present thesis and defines its objective as the development of a computational method for the synthesis of rigidly foldable crease patterns to support the application of origami in engineering design tasks. The first approach to the computational method is a numerical approach that introduces a new kinematic simulation method with which a manually adapted flasher pattern is analyzed. This analysis contributes by visualizing the search space and by revealing the existence of rigidly foldable regions in the search space of rigid foldability, based on which the flasher pattern is optimized using a stochastic search method. While successful for a single crease pattern topology, the numerical approach is too time-intensive to be scaled, leading to a deeper investigation into analytical kinematics. This investigation yields the Principle of Three Units (PTU) stating that the kinematic behavior of a single vertex is only dependent on its vertex triangle. By applying the triangle inequality, the PTU results in the conditions for the rigid and flat foldability of single degree- $n$  vertices. The corresponding kinematic model enables the assessment of different RBMs and offers an active selection of RBMs to be modeled. In addition, the PTU leads to two guidelines for the generation of kinematically determinate and acyclic crease pattern graphs. The guidelines and conditions arising from the PTU are then embedded within a graph grammar whose rule set consists of two rules. The rule application is automated, leading to a new approach for the computational method that enables the enumeration of the vast search space of origami, the synthesis of novel crease patterns including grippers and robotic arms, and yields the potential to apply the origami principle to yet uncharted territories in engineering design.

## Zusammenfassung

Origami, das Falten eines quadratischen Blatt Papiers, ist eine alte Kunstform deren Anwendung in Ingenieurwissenschaften erst in den letzten Jahrzehnten einen starken Zuwachs an Forschung erlebte nachdem man ihre unzähligen Vorteile erkannte. Diese Vorteile sind von kinematischen Herausforderungen begleitet wie die starre Faltbarkeit, eine exponentielle Anzahl von Starrkörpermodi, und komplexe Relationen zwischen der kinematischen Bestimmtheit, der Anzahl an Freiheitsgraden, und der Symmetrie. Diese Herausforderungen erschweren die Anwendung des Prinzips Origami, was zur Entwicklung von verschiedensten Computermethoden und Algorithmen führte, die die geometrischen Herausforderungen aber nur isoliert betrachten und sich vor allem auf existierende Faltmuster beschränken. Deswegen ist der heutige Designprozess mühsam, langwierig, und wird nur von erfahrenen Wissenschaftlern ausgeführt. Diese Lücke motiviert die vorliegende Arbeit und definiert ihr Ziel als die Entwicklung einer Computermethode für die Synthese von starr faltenden Faltmustern für Ingenieurzwecke. Der erste Anlauf zur Erreichung des Ziels basiert auf einer numerischen Simulation mit der ein manuell adaptierter Flasher zunächst visuell analysiert wird, was dreidimensionale Regionen starrer Faltbarkeit offenbart. Dann wird der Flasher mit einer stochastischen Methode optimiert, was zwar erfolgreich aber nicht effektiv genug ist um auf viele verschiedene Faltmuster angewendet zu werden. Dies führt zu einer grundlegenden Betrachtung der analytischen Kinematik von Origami und zum Prinzip der Drei Einheiten (PDE), das besagt, dass sich ein einzelner Faltpunkt auf dem starren Papier genauso verhält wie ein einziges zugrundeliegendes sphärisches Dreieck. Durch die Anwendung der Dreieckungleichung ist es dem PDE möglich, die Bedingung für die starre und die flache Faltbarkeit eines Faltpunkts mit  $n$  angrenzenden Faltlinien auszudrücken. Das dazugehörige kinematische Modell ermöglicht die Betrachtung verschiedener Starrkörpermodi und bietet deren aktive Modellierung. Weiterhin führt das PDE zu zwei Richtlinien zur Generierung kinematisch bestimmter und azyklischen Faltmustern. Diese Richtlinien werden dann zusammen mit der Bedingung für die starre Faltbarkeit in einer Graph-grammatik eingebettet, die ein Regelsystem mit zwei Regeln enthält. Die Anwendung dieser Regeln ist automatisiert, was es der Computermethode ermöglicht den Suchraum verschiedenster Faltmuster zu durchsuchen und gleichzeitig das Optimierungsmodell zu erstellen und die Optimierung durchzuführen. Dies führt zum Erreichen des Ziels dieser Arbeit, die ihren Nutzen in der Synthese von neuartigen Faltmustern wie Origami Greifern und Roboterarmen beweist und auf deren Grundlage das Prinzip Origami auf noch unerforschte Wissenschaftsbereiche ausgedehnt werden kann.

## Acknowledgements

Writing a doctoral thesis is a journey that took me four years. In many aspects, these last four years symbolize my personal development that is also paralleled in the thesis. In short, this development involves Search (Section 3), Discovery (Section 4), and Implementation (Section 5), the repetition of which I perceive as the natural cycle of Learning. Today, I consider Learning as the most important pursuit in (my) life, with the only exception of Love, and Love I have received in abundance from the people that accompanied me on this wonderful journey. Here is a good place to return some of it.

Family, you reside at the deepest bottom of my heart, at depths I can hardly express in words without writing another thesis, but let me try: Nina, Mischa, you have always been the most important co-adventurers in uncovering the secrets of the universe, you have kept me in check when I got sidetracked, and you have anchored me when the sea was in storm. Thank you for the memories. Mami, you inspired me with your enthusiasm for invention and instilled me with your analytical thinking, but perhaps most of all, you taught me the value of integrity. Pöpä, you inspired me with your unrelenting support and equipped me with your lighthearted manner, but perhaps most of all, you taught me the value of fairness. I thank you all deeply.

Sarah, my wind, you have swept into my life leaving a swath of unpredictability (the good kind) and beautiful Love (the romantic one). Thank you for being patient whenever I worked until deep in the night and for being my most wonderful distraction whenever I was able to spend a day without working.

Fortunately, I can call many people my friends; unfortunately, that does not make for a good acknowledgment since I cannot name all of you personally. Let me express my gratitude in this poor manner and let me make up for it twice in real life, which is made rich by all of you. However, special thanks go to my lab friends with whom I shared the same dreams, anxieties, and (most importantly) the funnest times. Thanks, peeps.

Evgueni, thank you for supporting me at all my conferences, for inviting me to your student dinners, and for traveling a long way to my defense: I enjoyed your company at all of these happenings.

Kristi, you suggested origami to me, a topic that has brought me immense pleasure and a much deeper understanding of processes otherwise hidden in our natural environment. Thank you for teaching me to apply structure and purpose to my work and for letting me develop my own ideas.

Tino, you contributed most to this thesis and earn my biggest gratitude. We share the same excitement for fundamental research, and I always knew that an idea is good as soon as (and only when) I was able to excite you too. In all these years, I learned many things from you and still never felt a subordinate, but a friend. You are a great mentor!

## Table of Contents

Abstract.....	i
Zusammenfassung.....	ii
Acknowledgements.....	iii
List of Symbols.....	vi
1 Introduction.....	1
1.1 Scope of the Thesis.....	4
1.2 Approach.....	5
1.3 Expected Contributions.....	7
1.4 Structure of the Thesis.....	8
2 Background.....	11
2.1 Rigid and Flat Foldability.....	11
2.2 Mountain-Valley Assignment and Rigid Body Modes.....	12
2.3 Simulation of the Folding Process.....	14
2.4 Analysis of Crease Patterns.....	15
2.5 Generation of Crease Patterns.....	15
2.6 Discussion.....	16
3 Numerical Approach to the Computational Method.....	19
3.1 Representation.....	20
3.2 Evaluation.....	20
3.2.1 Input to the Simulation.....	21
3.2.2 Constraints and Solving Procedure.....	21
3.2.3 Rigidity Error.....	22
3.2.4 Self-Intersection.....	22
3.2.5 Output of the Simulation.....	23
3.3 Generation: Manual Adaptation of a Flasher Pattern.....	23
3.4 Design Task.....	24
3.5 Sensitivity.....	25
3.5.1 Sector Angles around $v_1$ .....	26
3.5.2 Influence of $\mathbf{x}_3^{(0)}$ on the Behavior of the First Layer.....	27
3.5.3 Influence of $\mathbf{x}_4^{(0)}$ on the Behavior of the Entire Pattern.....	29
3.6 Guidance.....	32
3.7 Results.....	34
3.8 Discussion.....	35
4 Analytical Kinematics of Origami.....	39
4.1 Degree-4 Vertex.....	39
4.1.1 Preliminaries.....	40
4.1.2 Derivation of the Necessary and Sufficient Condition.....	43

4.1.3 Rigid and Flat Foldability in Degree-Four Vertices .....	44
4.2 Degree- $n$ Vertex .....	49
4.2.1 The Principle of Three Units.....	49
4.2.2 The Principle of Three Units in Relation to Origami Phenomena.....	50
4.2.3 Kinematic Model of the Principle of Three Units .....	52
4.2.4 Implications for Single Vertices.....	56
4.2.5 Implications for Crease Patterns .....	60
4.3 Manual Application and Results .....	63
4.3.1 Flasher Revisited .....	63
4.3.2 Rigid Body Modes.....	65
4.3.3 Target Shape and Symmetry .....	67
4.4 Discussion .....	69
5 Analytical Approach to the Computational Method .....	73
5.1 Representation.....	74
5.2 Generation.....	75
5.2.1 Rule $r_1$ : Extend Vertex .....	76
5.2.2 Rule $r_2$ : Combine Vertices.....	85
5.2.3 Automated Graph Generation and Filtering.....	87
5.3 Guidance .....	89
5.4 Evaluation.....	90
5.4.1 Optimization.....	90
5.4.2 Intersection .....	91
5.5 Design Tasks and Application of the Method.....	91
5.5.1 Input.....	92
5.5.2 Generation: Additional Filters.....	93
5.5.3 Optimization.....	94
5.6 Results .....	95
5.7 Discussion .....	98
6 Discussion .....	103
6.1 Objective.....	103
6.2 Research Question 1, Part 1 .....	106
6.3 Research Question 1, Part 2 .....	106
6.4 Research Question 2 .....	109
6.5 Research Question 3 .....	110
6.6 Research Question 4 .....	114
6.7 Limitations and Future Work.....	115
7 Summary and Conclusion.....	117
References .....	121

## List of Symbols

$A$	Projected area
<b>A-E</b>	Two-dimensional regions
$\alpha$	Sector angle
$\beta$	Start angle
$c$	Crease line
<b>c</b>	Crease line vector
$\Gamma$	Rigidity error
$\gamma$	Side of spherical triangle
<b>d</b>	Normalized direction vector
$\delta$	Distortion, or end angle
$E$	Set of edges
$e$	Edge
$\varepsilon$	Error
$F_L, F_R$	Facet label left and right, respectively
<b>F</b>	Folded state
$f$	Facet
$f_R$	Function expressing local rotation matrices
$G$	Graph
$\mathcal{G}$	Set of graphs
$GG$	Graph grammar
$\theta$	Measure for self-intersection
$\theta$	Polar angle, or internal angle
$\vartheta$	Azimuth angle, or target criteria
$i$	Enumeration
$J$	Number of iterations
$j$	Current iteration, or enumeration
$L_i$	Vertex label
$L_{i,j}$	Edge label
$L_V, L_E$	Set of vertex and edge labels, respectively
$l$	Crease line length
$\mathcal{L}$	Language
$LHS$	Left-hand side
$M$	Rigid body mode
<b>M</b>	Mountain crease
$M_\uparrow, M_\downarrow$	Rigid body mode up and down, respectively
$\mathcal{M}$	Match
$N$	Number of internal vertices
$n$	Degree of a vertex



$n_m, n_v$	Number of mountain and valley creases, respectively
$O$	Origin
$P$	Point location, or predecessor of a vertex
$q$	Number of constraints
$R$	Three-dimensional region, or set of global rotation matrices
$\mathbf{R}$	Rotation matrix
$\mathcal{R}$	Set of rules
$r$	Rule
$RHS$	Right-hand side
$\rho$	Dihedral angle
$S$	Set
$\Sigma$	Map
$T$	Type of a vertex
$t$	Time, or simulation step
$U$	Unit angle
$\mathbf{U}$	Unfolded state
$u$	Unit
$V$	Set of vertices
$\mathbf{V}$	Valley crease
$v$	Vertex
$\varphi$	Unknown dihedral angle
$w$	Weight
$\Phi$	Set of optimization variables
$X$	Set of three-dimensional coordinates
$\mathbf{X}$	Extended in-plane coordinates
$x$	Set of in-plane coordinates
$\mathbf{x}$	Vertex coordinates
$\chi$	Symbol for extendable vertex
$x, y, z$	Euclidean coordinates
$\psi$	Set of optimization constraints
$\Omega$	Spatial feasibility, or objective function
$\uplus$	Disjoint union of sets
$\emptyset$	Empty symbol, or terminal symbol



## 1 Introduction

Folding is a universal principle observed throughout nature from micro to macro scale. Proteins self-assemble into three-dimensional folded structures to interact with their biological environment [1], insect wings fold into carapaces for stowage and deploy for flying [2, 3], the human cortex convolutes during the rapid development within the skull [4], plants fold in a circadian rhythm [5], and some theories hint at the folded structure of the universe itself [6].

The human adoption of this natural principle is closely related to the introduction and availability of paper and dates back centuries to millennia [7]. The first man-made paper folding was targeted at packaging, art, and ceremonial artifacts [8], but all early paper models exhibited low complexity [7]. These simplistic models remained the only artificially folded structures until the early twentieth century when Akira Yoshizawa introduced a plethora of new fold patterns together with a visual representation that documented the individual folding steps [9]. These advances sparked a renewed interest in paper folding and coined the term *origami*.

Origami in its original art form connotes the folding of a quadratic, flat piece of paper without cutting or gluing. An origami (Fig. 1a) is usually represented as a *crease pattern* (Fig. 1b) that depicts the location and distribution of the *crease lines* in the unfolded state. A *vertex* is a point at which a minimum of two crease lines intersect, and the angle between two intersecting crease lines is called a *sector angle*. The polygons between the crease lines are called *facets*, and the rotational angle between two facets is called a *dihedral angle* (Fig. 1c) that is zero radians in the unfolded, and  $\pm\pi$  radians in the fully folded state.

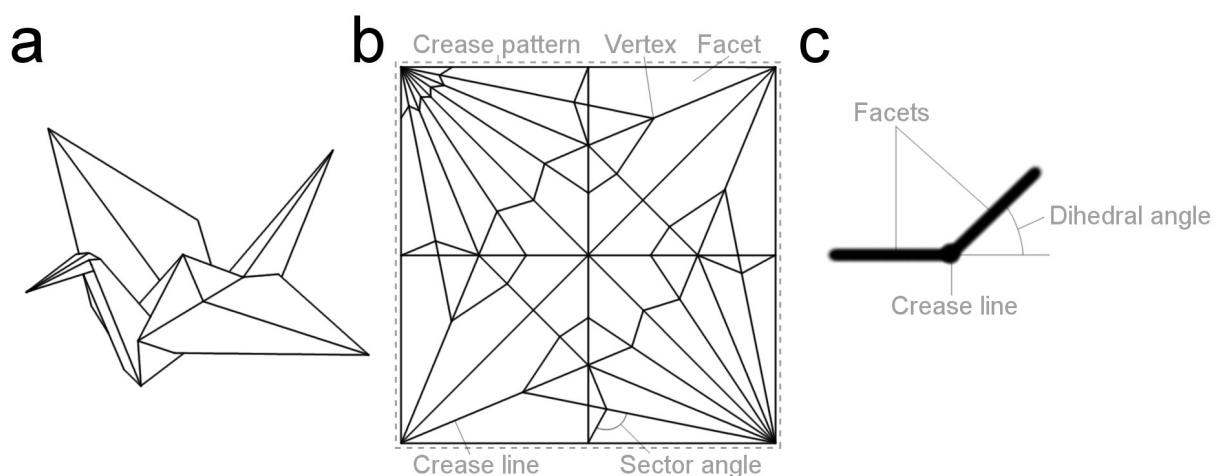


Fig. 1: (a) Origami crane, (b) crease pattern of the origami crane with vertices, crease lines, facets, and sector angles, and (c) the dihedral angle of a crease line between two facets

In addition to popularizing the term origami, which has become the umbrella term for any type of folding independent of shape, material, or scientific field, Yoshizawa's work led to a renaissance of folding when mathematicians, engineers, and artists recognized the benefits of origami. These benefits are numerous: origami is scale-independent [10] since the kinematics of folding depend on the sector angle relations that are preserved when scaled. The kinematics further allow for a compact or flat stowage in the unfolded state and complex three-dimensional motion during deployment [11]. The deployment can be achieved by a low number of Degrees-Of-Freedom (DOF), which minimizes the amount of resources required to actuate the folding motion and enables a reliable control [12]. The facilitated actuation and the complex motion then enable programmable structures that can change mechanical properties, shape, and function on demand [13, 14]. In addition, origami can be produced in the flat state and by additive manufacturing techniques [15], which enhances realization possibilities and simultaneously reduces the manufacturing cost and the assembly time.

Due to its benefits, the principle of origami offers potential for various scientific fields that pose different engineering design tasks. The transformation of origami into engineering applications includes an entire spectrum of more abstract to more direct implementations. The abstract end of the spectrum corresponds to so-called *origami-inspired* products [16] that fold and exhibit origami-like geometry but otherwise show little resemblance to origami. More closely related to traditional origami are *origami-adapted* mechanisms [17] that are based on origami crease patterns but use non-paperlike materials and accommodate for finite thickness. Finally, *origami-applied* systems [16] use paperlike material and exhibit little to no alteration of the underlying crease pattern. Together, these categories constitute the entire spectrum of *origami-based* design [18], which finds applications in mathematics [19], material science [20, 21], DNA [22] and biomedical research [23], mechanical engineering [24], robotics [14, 25-28], consumer goods [29], architecture [30], and space [18, 31-33].

In recent decades, the widespread applicability of origami-based design has caused a wealth of research: while the year 2000 recorded about five hundred new scientific papers containing the word "origami", the number of publications per year has since increased by more than tenfold, resulting in a current total of almost one hundred thousand scientific works related to origami [34]. This substantial increase was accompanied by the adaptation and introduction of various computational methods and tools to facilitate the application of origami in engineering design tasks. A number of methods simulate the folding motion through various approaches including bar-and-hinge models [35-37], finite element

methods [38, 39], and simulators that assess the closure constraint of the surface [40]. Other computational algorithms target the generation of crease patterns [41-46] or adjust the geometry of existing ones [47, 48], plan the folding motion [49], analyze the mobility [50-52], and approximate surfaces [46, 53].

Despite the trend toward computer-aided design processes, today most crease patterns employed in technical applications are still designed manually or selected from a handful of well-known origami principles [11]. The most prominent of these principles is the Miura-ori pattern [54] that finds application in the packaging of large membrane structures [55], the design of tubes [56] and cylinders [57], surface approximation [53], and metamaterials [10, 20, 21, 58]. Other famous patterns include the Yoshimura [59] and Resch [48] tessellations, the Huffman grid [60], and many more [61]. Since all of these crease patterns have the potential to serve numerous engineering purposes, a large part of origami research focuses on the adaptation of these existing principles instead of searching for novel crease patterns. Computational algorithms that are capable of generating novel crease patterns are either targeted at artistic origami [44], limited to quadrilateral creased paper [45], result in patterns that are not rigidly foldable [46], or utilize ground structures that only allow for regular sector angle configurations and thus offer limited design freedom [41-43].

The lack of computational methods for the generative design of novel crease patterns targeted at engineering applications arises from a multitude of geometric complexities intrinsic to origami. First, many engineering applications require an origami to *fold rigidly* for the incorporation of rigid materials or electronics. Rigid foldability is the property of an origami that folds continuously from an unfolded to a folded state without deformation in its facets, which results in complicated geometric conditions for the sector and dihedral angles. Although extensively researched, there exist no such generic conditions for rigid foldability even for the simplest origami crease patterns. Second, each internal vertex in a crease pattern exhibits two Rigid Body Modes (RBMs) that allow for both “upward” and “downward” motion [62, 63], which results in an exponential number of possible motions with respect to the number of vertices involved. The RBMs are closely related to the assignment of *mountain* and *valley* crease lines determining the signs of all dihedral angles within a crease pattern, which has been proven to be NP-hard [64]. Third, because of the flat initial state and the multiplicity of closed kinematic chains, origami crease patterns represent complex multibody systems whose mobility cannot be assessed accurately by most conventional methods [65]. The mobility of an origami is influenced by both the *topology* and

the *geometry* of a crease pattern, which denote the number and distribution of crease lines between vertices and the location of these vertices, respectively. In particular, the topology and the geometry define the relations between the kinematic determinacy, the number of DOF, and the symmetry of a crease pattern, but the exact nature of these relations is still largely unexplored. Fourth, real-world applications do not allow for self-intersection, for which there still exist no intrinsic conditions. Finally, converting infinitely thin crease patterns into realizations with finitely thick materials imposes problems on both crease pattern and hinge design [66].

These complexities explain the challenges involved in the generation of novel crease patterns: while many computational support tools isolate and tackle specific geometric problems, the synthesis of novel crease patterns encompasses all of the above complexities and thus represents the pinnacle of origami design. This pinnacle has not been reached by any related work, and today the design of novel origami crease patterns for engineering applications is tedious, time-consuming, and limited to a handful of experienced scientists [11]. This gap stipulates the need for a computational method capable of generating origami crease patterns to capitalize on the numerous benefits, the vast design space [67], and the widespread applicability of scientific origami.

## 1.1 Scope of the Thesis

**Objective:** The objective of this thesis is to develop a computational method for the synthesis of rigidly foldable crease patterns to support the application of origami in engineering design tasks.

The broad formulation of the objective requires two definitions to delineate the scope of this work, the first of which corresponds to the exact nature of the crease patterns analyzed and generated. This thesis targets the synthesis of origami crease patterns that lie between origami-applied and origami-adapted structures [11]. The focus lies on unaltered crease patterns with zero thickness, pertaining to origami-applied mechanisms. Conversely, all crease patterns are required to fold rigidly, corresponding to non-paperlike materials and thus to origami-adapted structures. The rigidly foldable crease patterns are not restricted to conform to a quadratic shape in the unfolded state and neither cutting nor gluing is considered. If possible, the crease patterns in focus are *developable*, meaning that they can be folded from a flat surface [47]. In addition, the focus lies on crease patterns that exhibit a low number of DOF to facilitate a realization with the least possible amount of resources. A low number of actuators further enables a reliable control [12] that is beneficial especially for remote or hardly accessible environments for which origami engineering applications are often designed [42].

The second definition explains the nature of the engineering design tasks considered in this thesis. Since origami offers the realization of technical solutions for various scientific fields, the range of engineering design tasks to which the origami principle can be applied is broad, and can involve, e.g., forces [68], electromagnetic properties [69], or material characteristics [70]. Independent of their purpose, however, all design tasks realized by origami require crease patterns that exhibit an appropriate kinematic behavior. Hence, this thesis focuses on purely kinematic aspects of the engineering design tasks to explore the geometric capabilities of origami as a basis for the synthesis of new technical solutions.

To achieve the objective, this work addresses the following four research questions (RQs) that are closely related to the geometric challenges of origami:

**RQ1:** What are the characteristics of the search space of rigid foldability and what are the conditions for an origami to fold rigidly?

**RQ2:** How can the exponential number of RBMs be modeled kinematically?

**RQ3:** What are the relations between the kinematic determinacy, the number of DOF, and the symmetry of an origami crease pattern?

**RQ4:** How can the answers to RQ1-RQ3 be embedded within an automated, generative method to synthesize origami crease patterns?

## 1.2 Approach

To answer the RQs and achieve the objective, the approach taken in this work to developing a computational method targets the integration of six parts that constitute a Computational Design Synthesis (CDS) method, which is defined as the “algorithmic creation of designs; the organized, methodological modeling, implementation and execution of design creation on a computer” [71]. These six parts involve an *input*, the *representation*, *evaluation*, and *generation* of designs, the *guidance* through the design space, as well as an *output*, all of which are detailed in the following paragraphs.

**Input:** The input involves the definition of engineering design tasks that focus on the kinematic capabilities of origami. Such design tasks are usually expressed by a folded state for stowage and a deployed state for a specific purpose, or vice versa, and by the transition between these two states, i.e. the actuation of the folding motion. The input considered in this work should thus incorporate an initial

crease pattern together with its actuation, a target shape or function, as well as possible spatial constraints, so that crease patterns can be tailored to their environment and the available resources. Throughout this thesis, the exact definition of the input changes according to the current approach and will be explained in more detail in Sections 3, 4, and 5.

**Representation:** In general, the representation of designs within a CDS method defines the focus of the computational search and determines the exact approach taken by the method [71]. As such, the representation defines the means through which the designs are generated and evaluated, which is why choosing a suitable representation is a core task in the development of a CDS method. The representation of an origami should include both the topology and the geometry of the underlying crease pattern, for which there exist various ways in literature such as connectivity matrices [50, 51], diagrams [72], as well as undirected [73] and directed graphs [74]. The representation also involves the choice of suitable coordinates that define the mathematical framework. Related works include natural and relative coordinates [75] modeled by vertex locations as well as sector and dihedral angles, respectively, both of which are utilized in this thesis depending on the specific purpose.

**Generation:** Automatically generating design alternatives through computational methods is instrumental for the creation of novel designs. In complex design tasks with large search spaces, human designers may suffer from confirmation bias [76] and design fixation [77], the latter of which is expressed in origami research by the focus on existing crease patterns as explained above. The generation of origami crease patterns is mainly engaged with the challenges involved in the creation of crease pattern topologies, and related existing methods are presented in Section 2.6.

**Evaluation:** The evaluation of generated design alternatives usually represents the most time-intensive step in the conceptual design for engineering tasks [71]. Computationally automating the evaluation has the potential to speed up the design process and reduce the problems associated with the human-oriented selection process in growing search spaces [78]. Since origami exhibits vast search spaces [67] and complex three-dimensional motion, an efficient computational evaluation of the folding process is paramount. Due to the nature of the considered design tasks, in this thesis the evaluation of crease patterns focuses on the assessment of the kinematic behavior that involves rigid foldability, spatial configurations of folded and unfolded states, as well as intersection. A description of the related background is given in Sections 2.4 and 2.5.



**Guidance:** To replace the human selection process [79] and fully capitalize on the advantages offered by the computational generation and evaluation of design alternatives, a CDS method requires a suitable strategy to guide the exploration of the search space. In general, the guidance strategy makes decisions based on the outcome of previous evaluations and determines the following steps taken to adjust designs or generate new ones. Due to the few implementations of computational methods for the synthesis of crease patterns, guidance strategies in origami are understudied and often replaced by direct computational algorithms that implicitly incorporate the design of crease patterns, such as in [44]. The guidance strategies employed in this thesis are detailed in Sections 3 and 5.

**Output:** As soon as the termination criteria are met, a CDS method outputs the feasible design alternatives that in this thesis equate to rigid origami concept(s) that satisfy the prescribed engineering tasks. The term “concept” is used here to clarify that this work addresses all of the above-mentioned geometric complexities, i.e. rigid foldability, RBMs, mobility, and intersection, except the problem of finite thickness [66]. Although certain measures can be taken to conform crease patterns to the challenges of finite thickness, the origami concepts developed in this work still require adaptation to finite thickness after the application of the proposed methods.

### 1.3 Expected Contributions

The expected contributions are tied to the answers of the RQs and the achievement of the objective of this thesis:

**RQ1:** Since this work focuses on purely kinematic design tasks, large parts of the thesis are targeted at uncovering the mathematical underpinnings of rigid foldability. Although many related works are directed toward rigid foldability, a generic intrinsic condition for rigid foldability is missing even for the simplest origami crease patterns. Discovering such a condition and exploring the characteristics of the corresponding search space is expected to influence origami research on a fundamental level.

**RQ2:** The exponential number of RBMs exhibited by crease patterns plays an important role in the complexity of origami and the size of the vast design space. As further detailed in Section 2, kinematically modeling different RBMs through existing approaches is a cumbersome process, which is why the development of a reliable kinematic model to both discover RBMs and assess the corresponding kinematic behavior is a significant contribution to origami research.

**RQ3:** The relations between kinematic determinacy, the number of DOF, and the symmetry of crease patterns determine the mobility of origami mechanisms. Understanding these relations is expected to

contribute by offering an approach to generating kinematically determinate, rigidly foldable crease patterns whose folding motion is driven by the prescribed number of DOF.

**RQ4:** Addressing RQ1-RQ3 should provide solutions to the geometric challenges of origami. However, the discoveries made still have to be embedded within an automated method whose integration then contributes by combining the knowledge of origami design into one concise resource and by enabling the objective of this work.

**Objective:** In comparison to today's manual-oriented design of origami crease patterns performed by specialists, automating the synthesis of origami crease patterns offers an improved approach to incorporating origami principles as a base for the development of novel technical solutions. Such a computational method contributes by generating and evaluating many more crease patterns than possibly assessed by human designers, by reducing the time-consumption and the tedium of human designers, and by creating novel origami crease patterns that expand the range of engineering design tasks realized by the application of origami.

#### 1.4 Structure of the Thesis

The thesis is structured according to Fig. 2. **Section 2** elaborates the important related works and presents the state-of-the-art in origami research, which highlights the gaps and the implications for this work. **Section 3** presents a numerical approach to the computational method that specifically targets and partially answers RQ1 and RQ2 by introducing a kinematic simulation method that enables both the analysis of the search space of rigid foldability and the assessment of RBMs. The findings and shortcomings of the numerical approach then direct the work performed in **Section 4**, which completes the answers to RQ1 and RQ2 and provides an answer to RQ3 by investigating the analytical kinematics of origami. This investigation reveals deep insight into the mathematics of folding and builds a strong basis for an analytical approach to the computational method. This analytical approach is presented in **Section 5** that combines the knowledge gathered by embedding the answers to RQ1-RQ3 within an automated computational synthesis (RQ4) to achieve the objective of this thesis. **Section 6** discusses the findings and contributions with respect to the objective and the RQs and outlines the limitations and respective future paths. Finally, **Section 7** summarizes the thesis and lists the major contributions of this work.

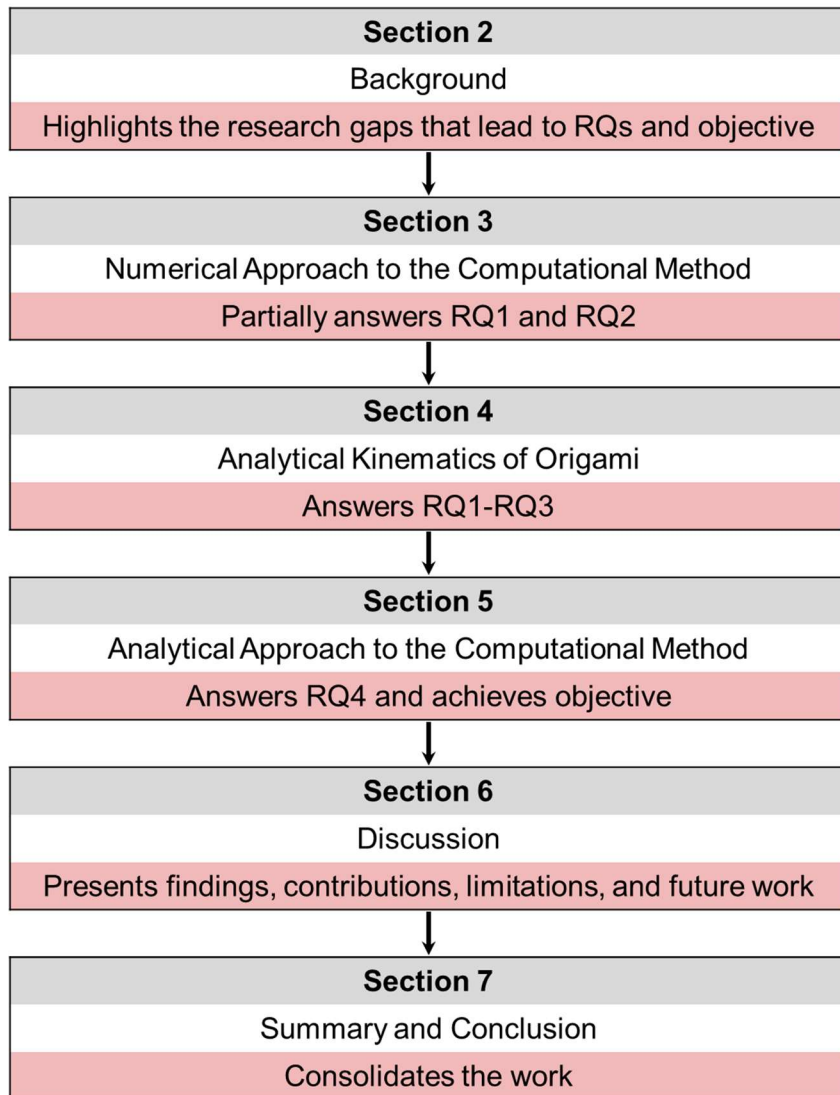


Fig. 2: Structure of the thesis



## 2 Background

This section covers the related background in rigid and flat foldability, mountain-valley assignments and RBMs, the simulation of the folding process, as well as the analysis and the generation of crease patterns. The section concludes by reiterating the research gaps that lead to the objective and the research questions of this thesis.

### 2.1 Rigid and Flat Foldability

Rigid foldability is the notion of a continuous folding process throughout which the facets of an origami stay undeformed, which requires all sector angles to remain constant. The entire folding motion then arises solely from rotations around the crease lines whose folded state is described by the dihedral angles. When an origami is rigidly foldable, its complex folding motion can be realized using exclusively stiff materials whose implementation is advantageous for, e.g., the embedding of electronics [80, 81] or the protection against the environment [33, 82, 83]. Although an onset of compliant origami mechanisms [84] has been taking place in recent years, rigid foldability still represents an important prerequisite for many technical applications, explaining the wealth of research directed toward rigid foldability and making it one of the most discussed topics in origami science.

Part of the research on rigid foldability focuses on the analysis of single vertices. The kinematics of a single vertex depend on the *degree* of the vertex, which denotes the number of incident crease lines. Miura proved that degree-1, degree-2, and degree-3 vertices cannot fold rigidly without collinear crease lines [85], and collinear crease lines are infeasible in rigid origami since all sector angles  $\alpha$  are constrained to be non-zero and strictly smaller than  $\pi$ ,  $\alpha \in (0, \pi)$ . Degree-4 vertices are the simplest rigidly foldable vertices and exhibit a single DOF [21, 74], meaning that the input of one dihedral angle determines the remaining dihedral angles and thus the complete folded state of the vertex [86]. Huffman [60] was one of the first to research single vertices of degree four to derive relations between sector and dihedral angles using the concept of Gaussian curvature. The same concept was used by Lang et al. [87] to prove compatibility conditions between opposite pairs of dihedral angles. In addition, the kinematic behavior of all possible sector angle configurations of degree-four vertices are catalogued by Waitukaitis and van Hecke [88]. Since each additional crease line added to a vertex brings an additional DOF that increases the level of geometric complexity, much less is known about the rigid foldability of higher order vertices. Although there exists an approach to determining symbolic equations for the unknown dihedral angles [89], known conditions for rigid foldability usually apply only to specific types

of crease patterns [59, 90-94], and there is no generic approach capable of determining analytical conditions for the rigid foldability of degree- $n$  vertices. The limits of the rigid motion of generic degree- $n$  vertices with  $n \geq 4$  can only be assessed using a diagram method [72] or approaches that numerically assess the closure constraint of a vertex, which is satisfied if the multiplication of all rotation matrices on a closed path around a vertex results in the identity matrix [74].

The closure constraint approach also enables the assessment of multi-vertex crease patterns when all the closed paths around the vertices within a crease pattern are examined [95]. Other approaches to judging the rigid foldability of multi-vertex crease patterns are the afore-mentioned diagram method [72], adapted mobility rules for patterns exhibiting symmetry [50, 51], and the fold angle multipliers [91] that are based on findings for *flat foldability*.

Flat foldability determines whether a rigid origami can fold from the initial flat state into a final folded state that is also completely flat [96]. This second flat state enables efficient packaging desired in many engineering applications [54], which renders flat foldability the second most discussed mathematical notion in origami after rigid foldability.

The most prominent condition for flat foldability is the Kawasaki-Justin condition [97, 98] that states that the sum of all even sector angles is equal to the sum of all odd sector angles:

$$\alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 + \dots - \alpha_{2n} = 0 \quad (1)$$

This condition is derived from the two-colorability of a single vertex and is thus only valid for vertices of even degree, whereas an equivalent condition for vertices of odd degree does not exist.

A special set of conditions applies to flat foldable degree-4 vertices, for which Tachi [30, 99] identified that the ratio of half-angle tangents between all fold pairs in a developable and flat foldable quadrilateral mesh is constant, independent of the folded state. Based on these findings, Evans et al. [91] presented the above-mentioned fold angle multipliers to analyze rigidly foldable origami twists. Since the fold angle multipliers are based on relations for rigid foldability, they can be used as conditions for the rigid foldability of quadrilateral creased paper [99]. In addition, they are the only evidence that the conditions for rigid and flat foldability could be related.

## 2.2 Mountain-Valley Assignment and Rigid Body Modes

The initial flat state of an origami crease pattern is a singular state [100, 101] that allows for many different branches of motion arising from both positive and negative values of dihedral angles. A

mountain-valley (MV) assignment of a crease pattern determines for each crease line if it folds down (mountain) or up (valley). Bern and Hayes proved that such an MV assignment for an arbitrary crease pattern is NP-hard [64]. With respect to flat foldable and even-degree vertices, Maekawa's condition states that the number of mountain creases  $n_m$  and the number of valley creases  $n_v$  differ by two,  $n_m - n_v = 2$ , or vice versa. The extension of Maekawa's condition to degree- $n$  vertices stems from Abel et al. [62] who proved that any rigidly foldable vertex requires either a bird's foot or a cross, both of which are specific MV assignments that prescribe the parity of certain subsets of crease lines.

Closely related to the MV assignment are the RBMs of a vertex that determine if a vertex on the interior of a crease pattern [62] folds up or downward. Fig. 3 [102] shows the example of a crease pattern that corresponds to one side of a six-sided flasher pattern [17]. The crease pattern consists of  $N = 2$  internal vertices and thus results in  $2^N$  [63], or 4 RBMs shown from left to right.

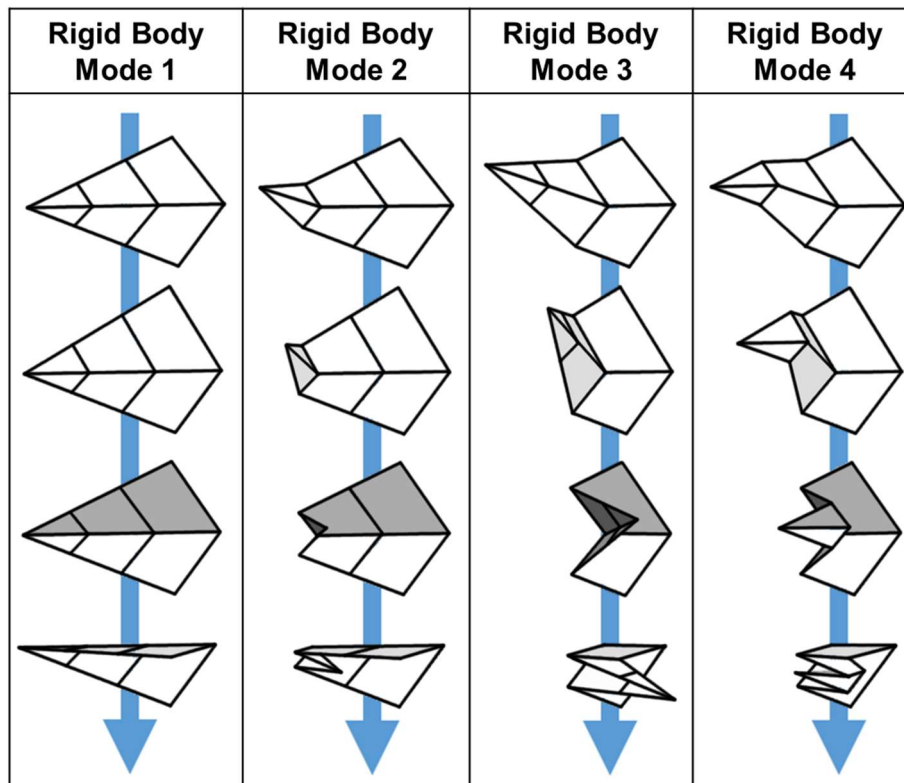


Fig. 3: Four RBMs from left to right of one side of a six-sided flasher pattern folding from top to bottom

All four RBMs exhibit different motions and final folded states, all of which might satisfy or dissatisfy spatial requirements or constraints pertaining to a given engineering design task. To determine the fitness of a crease pattern with respect to a design task, this considerable difference in kinematic behavior exhibited by all crease patterns requires special attention within a computational method that must incorporate the assessment of different RBMs.

### 2.3 Simulation of the Folding Process

Approaches to simulating the origami folding process involve bar-and-hinge models, Finite Element (FE) analyses, and the *Rigid Origami Simulator* by Tachi [40]. The latter tries to satisfy the closure constraint of the surface on all paths around vertices by solving the pseudo-inverse of the Jacobian matrix to determine the unknown dihedral angles at each time step. This approach requires a valid MV assignment and thus some prior knowledge about the underlying crease pattern. However, the generation of novel crease patterns does not provide any information about valid MV assignments, rendering the Rigid Origami Simulator inapplicable to the automated evaluation of generated crease patterns.

FE analyses [38, 39] discretize the surface of the origami and make use of mathematics derived from plate structures. These works focus more on the practical implementation of origami rather than the strictly kinematic behavior since they model the elastic behavior of materials and allow for bending and stretching.

Filipov et al. [35] present a combination of FE analyses and bar-and-hinge models. Instead of discretizing the surface into small elements, this approach models quadrilateral facets by introducing fewer nodes per facet and by only then applying FE principles, which enables a more efficient structural analysis of origami mechanisms compared to conventional FE methods. However, generalizing the approach for facets with more than four sides might involve considerable future work since the results are dependent on the exact amount and placement of the introduced nodes.

Purely kinematic bar-and-hinge models are presented by Cai [37] et al. and Diaz [36]. In both works, the sides and diagonals of all facets are replaced by bars that correspond to a length constraint between two vertices  $v_i$  and  $v_j$  in the form

$$(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 - l_{ij}^2 = 0 \quad (2)$$

where  $x$ ,  $y$ , and  $z$  are the coordinates in the Euclidean coordinate system and  $l$  is the initial distance between the two vertices. Diaz [36] then minimizes the difference in length of all bars to determine the vertex locations, while Cai et al. [37] solve the Jacobian matrix of the constraint system at each time step of the folding process. In relation to an automated approach for origami design, these bar-and-hinge models are promising since they offer a simple, scalable, and time-efficient way to simulate the kinematic behavior of an origami crease pattern, but their implementation in related works lack the search for different RBMs.



Preliminary work [102] by the author introduced a bar-and-hinge simulator that detects RBMs by perturbing all vertices slightly in different directions. However, the Jacobian of the constraint system is solved by using the Newton-Raphson scheme [75], which is prone to divergence if the underlying crease pattern is not rigidly foldable. Since arbitrarily generated crease patterns are not generally rigidly foldable, the approach provides only a binary result for rigid foldability without any feedback to determine the adjustment of the geometry required for rigid foldability.

## 2.4 Analysis of Crease Patterns

The analysis of a crease pattern is mainly concerned with the assessment of the crease pattern topology to determine if the underlying graph is kinematically determinate. Kinematic determinacy expresses the notion that the number of existing mechanisms within a structure is equal to the respective number of DOF [103], which is the prerequisite for a well-defined motion.

The kinematic determinacy of an origami can in general be analyzed by calculating the rank of the Jacobian matrix and by comparing the result to the number of unknowns [37]. If these numbers match, a crease pattern can fold rigidly with the respective actuation. If the matrix rank is smaller than the number of unknowns, the system is underconstrained and requires a more rigorous control to fold in a determinate way. In contrast, overconstrained structures play a significant role in origami [99], a phenomenon well studied for quadrilateral creased paper [93] by the introduction of the fold angle multipliers [91]. For crease patterns including higher order vertices, overconstrained but rigidly foldable mechanisms are not reported in origami literature but could in general be analyzed by adapted mobility rules [50, 51] or other related methods [65].

## 2.5 Generation of Crease Patterns

A long-standing tradition in the generation of origami crease patterns is the adaptation of fold patterns that appear in nature. Examples are the Miura-ori pattern that copies corrugated metal tubes and flower leaf surfaces [104], patterns found on insect wings [3], pinecones, and many more [105].

Lang's Treemaker [44] represents the state-of-the-art for the generation of artistic crease patterns. This computational algorithm optimizes the distribution of circles within the quadratic piece of paper and creates the required number of flaps that can then be folded into intricate origami models by experienced origami design practitioners. Another generative algorithm for artistic origami is the Origamizer [46] that approximates polyhedral surfaces by creating detailed crease patterns involving advanced crimp folds. However, neither of these approaches guarantees rigid foldability.

An approach that does guarantee rigid foldability is based on the fold angle multipliers that can also be introduced as design features instead of checks for rigid foldability. This reversed usage of fold angle multipliers enables the generation of quadrilateral creased paper [45] by designing all vertices so that they satisfy a set of conditions for flat foldability. The fully determinate algorithm then generates quadrilateral, flat foldable, and overconstrained crease patterns that fold rigidly. The approach, however, is limited to fold angle multipliers and thus to degree-4 vertices, and no related condition has been reported for higher order vertices.

The last class of generative approaches [41-43] relates to structural topology optimization and is based on a discretized ground structure. By representing this ground structure as a genotype, each element can be switched on or off and the resulting patterns are optimized using Genetic Algorithms. This procedure requires vast amounts of objective function evaluations, which demands a coarse ground structure in which all sector angles are  $\frac{\pi}{4}$  radians or multiples thereof. Such regular sector angle configurations restrict the design freedom and lead to crease patterns with limited functionality.

## 2.6 Discussion

Rigid foldability is well researched with respect to degree-4 vertices, but angle relations for higher order vertices are understudied or completely missing. No method is currently able to determine the analytical conditions for the rigid foldability of degree- $n$  vertices, and the limits of the rigid motion of these vertices can only be assessed using numerical approaches. In addition, analytical conditions for flat foldability only exist for even-degree vertices. These findings drive RQ1.

The available numerical simulators are resourceful, but they either require valid MV assignments, include forces and material properties, or do not focus on the discovery of RBMs. Since each crease pattern exhibits an exponential number of RBMs, being able to kinematically model different RBMs is paramount for an efficient search through the vast design space of origami, as targeted by RQ2.

Methods for assessing the mobility of origami crease patterns exist, but they either require elaborate Jacobian matrix rank calculations or focus on the analysis of the kinematic behavior rather than the generation of crease patterns. To enable a reliable method for the generative design of origami crease patterns, the mobility of origami mechanisms must be better understood, as expressed by RQ3.

The existing generative methods either focus on the manual adaptation of natural crease patterns, generate artistic origami without guaranteeing rigid foldability, are limited to quadrilateral creased paper, or rely on ground structures that limit the design freedom. These methods do not sufficiently leverage the benefits offered by origami for the realization of novel technical solutions, which is why the objective of this work is the development of a new generative method for the synthesis of crease patterns to support the application of origami in engineering design tasks.



### 3 Numerical Approach to the Computational Method

The content of Section 3 has been published as: [106] Zimmermann, L., Shea, K., and Stanković, T., "Origami Sensitivity – On the Influence of Vertex Geometry," in Proceedings of Origami7: Seventh International Meeting of Origami Science, Mathematics and Education, 2018, p. 1087-1102.

This section presents the first approach (Fig. 4) to achieving the objective of the thesis. As described in the last section, establishing such an approach first requires more knowledge about rigid foldability (RQ1) as well as the evaluation of the folding motion and the RBMs (RQ2). The approach presented here simultaneously addresses RQ1 and RQ2 by introducing a new kinematic simulation method that enables the examination of the search space of rigid foldability and the assessment of RBMs. At this point, advanced analytical conditions for rigid foldability are missing, which is why the new kinematic simulation and thus the overall approach are realized through purely numerical means. Due to the absence of generative methods that satisfy the given requirements, the automated generation step of the planned CDS method is substituted in this section by the manual adaptation (Fig. 4) of an existing crease pattern, similar to the generation approach in related works.

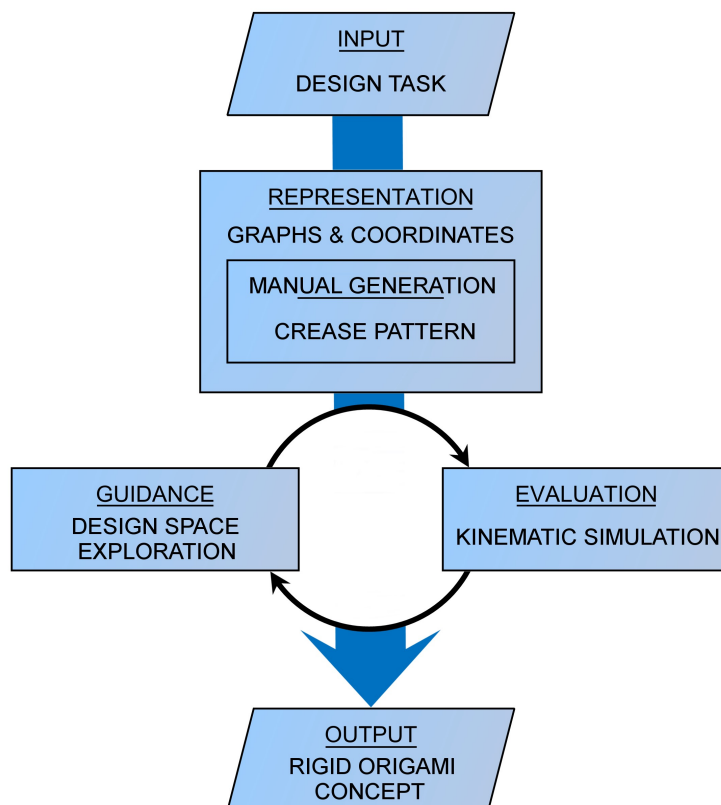


Fig. 4: Workflow of the numerical approach to the computational method

The section first presents the representation of crease patterns and introduces the kinematic simulation. Then, a flasher pattern is manually adapted to fold rigidly and, based on this pattern, an engineering design task is presented. Subsequently, a sensitivity analysis is performed to investigate

the search space of rigid foldability (RQ1) and to assess the kinematic behavior with respect to the RBMs (RQ2). Following the sensitivity analysis, a suitable method is selected to guide the optimization of the flasher pattern within an automated loop as shown in Fig. 4. Finally, the section presents the resulting origami concept and discusses the findings.

### 3.1 Representation

Crease patterns are implicitly represented as undirected graphs  $G = (V, E)$  where  $V$  is a non-empty set of vertices  $v$  and  $E$  is a non-empty set of edges  $e$ . Explicitly, crease patterns are represented as a bar-and-hinge model since such models are scalable, simple to implement, and lead to sparse constraint matrices that can be solved by fast algorithms. The constraints are formulated according to Eq. (2) using vertex coordinates  $\mathbf{x}$  represented by both natural and relative coordinates [75] in both Cartesian and spherical coordinates depending on the task at hand. The geometric distribution and locations of all vertices within a crease pattern will be collectively called the *vertex geometry*.

### 3.2 Evaluation

Since arbitrary generated vertex geometries do not generally lead to rigid foldability, the simulation method needs to be able to evaluate both rigidly and non-rigidly foldable crease patterns in order to be applicable within a computational synthesis of crease patterns.

By representing an origami through the locations of vertices, rigid foldability equates to constant Euclidean distances between all vertices that belong to the same facet. In contrast, facets in non-rigidly foldable patterns experience bending and stretching, and the Euclidean distances between vertices of the same facet change. In bar-and-hinge models, the folding motion of a crease pattern is usually driven by a set of parametric vertex trajectories. Fig. 5 depicts the schematic example of a single bar between two vertices  $v_1$  and  $v_2$  that is “folded” from an initial vertical into a horizontal position by three different trajectories from left to right in  $J = 2$  total iterations. The trajectory in the middle (black, dashed) of Fig. 5 is a perfect quarter circle, which is why the initial Euclidean distance, called the *target length*  $l_t$ , between the initial vertex locations  $\mathbf{x}_1^{(0)}$  and  $\mathbf{x}_2^{(0)}$  is equal to the actual length  $l_a^{(j)}$  in all iterations  $j$ . If all Euclidean distances in a crease pattern behave in the same way, the crease pattern is rigidly foldable.

However, the trajectories on the left and right-hand side of Fig. 5 shorten and elongate the bar, respectively, and the actual lengths  $l_a^{(1)}$  and  $l_a^{(2)}$  are not equal to the target length  $l_t$ . This length change leads to a motion that is not rigidly foldable, as characterized by the errors  $\varepsilon^{(1)}$  and  $\varepsilon^{(2)}$ .

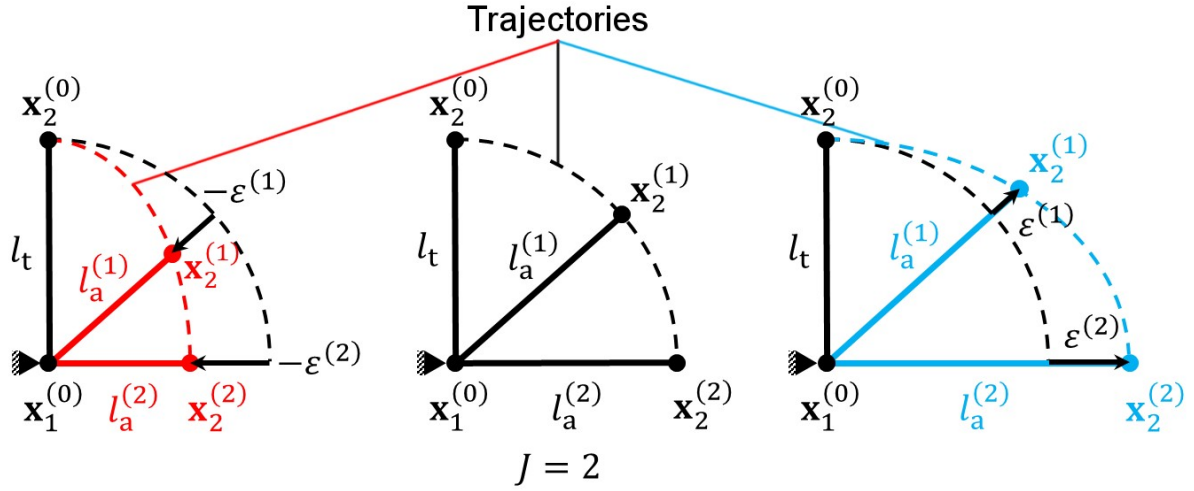


Fig. 5: The same vertical bar is “folded” from an initial vertical into a horizontal position by three different trajectories. While the bar folds rigidly for the case in the middle, it shortens and elongates on the left and the right side, respectively.

The basic idea of the simulation method is to allow these errors but to minimize the difference between the actual and the target Euclidean distance between vertices to obtain a measure for the total distortion of the crease pattern. Instead of imposing explicit conditions for rigid foldability, the distortion is presented as an error that can then be guided towards zero by optimizing the vertex geometry.

### 3.2.1 Input to the Simulation

The input to the simulation is a crease pattern with vertices and edges, the initial locations of the vertices, boundary conditions such as fixed vertices or facets, and one or multiple vertex trajectories that drive the motion of the origami in a desired number of iterations  $J$ .

### 3.2.2 Constraints and Solving Procedure

First, the simulation method obtains all target lengths  $l_t$  from the initial crease pattern provided by the input. The number of target lengths  $q$  corresponds to the Euclidean distances of the sides and diagonals of all facets in the crease pattern according to Diaz [36]. Because the actual lengths  $l_a^{(j)}$  may shorten or elongate in each iteration  $j$ , the constraint system is composed of the absolute of the difference between  $l_a^{(j)}$  and  $l_t$ , which is kept lower or equal to small and strictly positive values in the error vector  $\epsilon^{(j)}$ . This is formulated as two sets of constraints bounded by  $-\epsilon^{(j)}$  and  $\epsilon^{(j)}$  from below and above, respectively. The errors are then summed over all  $q$  constraints and presented as the distortion  $\delta^{(j)}$  that is minimized in each iteration  $j$  with respect to the errors  $\epsilon$  and the vertex locations  $x$ . This procedure is stated as a constrained nonlinear optimization problem in Eq. (3) that is solved using the function *FindMinimum* in *Mathematica 10* with default settings. *FindMinimum* uses the Interior Point

method [107] that requires a starting point for the optimization, here defined as the vertex locations in the previous iteration  $j - 1$  for the search of vertex locations in the current iteration  $j$ .

$$\min_{\varepsilon, \mathbf{x}} \left\{ \delta^{(j)} = \sum_{i=1}^q \varepsilon_i^{(j)} \mid -\varepsilon_i^{(j)} \leq \mathbf{l}_a^{(j)} - \mathbf{l}_t^{(j)} \leq \varepsilon_i^{(j)} \right\} \quad (3)$$

### 3.2.3 Rigidity Error

If the distortion satisfies  $\delta^{(j)} \leq 10^{-4}$  in each iteration  $j$  (Section 3.5.2 will show how this boundary is defined), all folding steps exhibit numerical rigid foldability and the crease pattern is globally rigidly foldable for the given actuation with a reasonably high number of iterations  $J$ . To obtain a single measure for each simulation, a rigidity error  $\Gamma$  is defined as the averaged sum of the distortions  $\delta^{(j)}$  over all iterations  $J$ :

$$\Gamma = \frac{1}{J} \sum_{j=1}^J \delta^{(j)} \quad (4)$$

### 3.2.4 Self-Intersection

All facets are decomposed into triangles and subjected to a triangle-triangle intersection check by Möller [108] in each iteration  $j$ . This check, however, does not necessarily detect dihedral angles that exceed the allowed range of  $\pm\pi$  and may thus neglect the intersection of two adjacent facets. Hence, the intersection check is extended by calculating two vectors for all pairs of adjacent facets, each vector lying in the respective facet plane and pointing away perpendicularly from the common crease line. Fig. 6 shows these vectors in red for two adjacent facets  $f_1$  and  $f_2$  angled at a dihedral angle  $\rho$  that is smaller than  $\pi$  on the left and greater than  $\pi$  on the right.

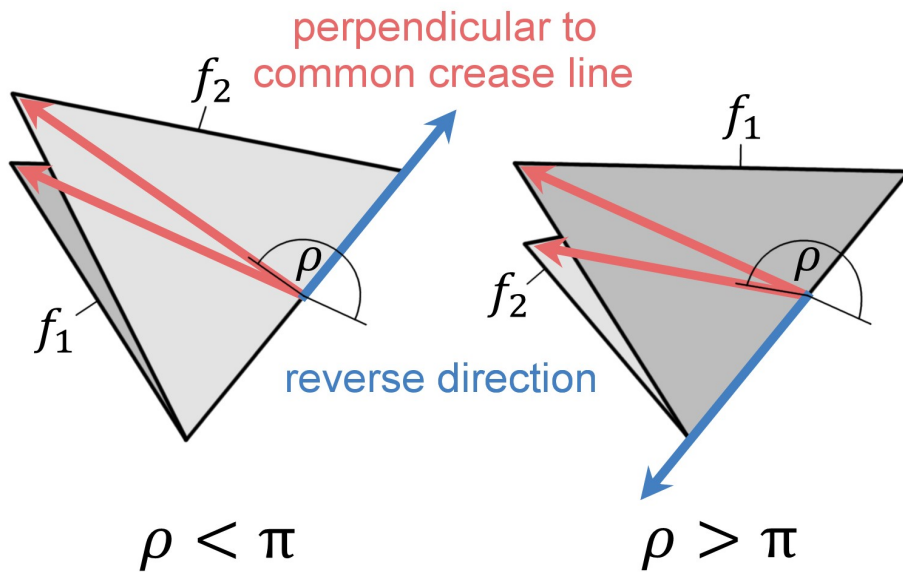


Fig. 6: Two adjacent facets angled at dihedral angles  $\rho$  that are smaller (left) and greater (right) than  $\pi$ . The vectors perpendicular to the common crease line are shown in red, and the cross product vector that reverses its direction is shown in blue.



Once  $\rho$  approaches the boundary of its allowed range, the cross product of the vectors perpendicular to the common crease line is compared for successive iterations, and an intersection is detected if the resulting vector reverses its direction, as illustrated in Fig. 6 by the blue vectors.

The measure for self-intersection  $\theta$  provided by the simulation method is defined as the number of iterations in which self-intersection occurs, averaged over all iterations  $J$ .

### 3.2.5 Output of the Simulation

The output for each simulation of a folding motion is the rigidity error  $\Gamma$  and all distortions  $\delta^{(j)}$ , the measure for self-intersection  $\theta$ , as well as the locations of all vertices in the last iteration  $j = J$  to assess the shape of the folded origami.

### 3.3 Generation: Manual Adaptation of a Flasher Pattern

To address RQ1 and RQ2, the simulation method needs to be applied to a crease pattern that offers a suitable topology for the investigation of the kinematic search space. An origami flasher (Fig. 7 left) is a pattern first introduced by Palmer and Shafer [109] and represents the principle of wrapping a piece of paper around its vertical axis. In technical applications, flashers are employed, e.g., for solar panels [17] since the pattern can be folded into an enclosed space of a launch vehicle and, once deployed, exhibits a large surface area that enables the collection of solar energy. Flashers exhibit a low number of DOF (one per side), which facilitates the actuation of the folding process through a low number of actuators.

Currently, existing flasher patterns do not fold rigidly because they are overconstrained, and the only way flashers have been shown to fold rigidly is by introducing cuts [87]. However, Zirbel et al. [17] hint at the possibility of rigidly foldable flashers if more crease lines were introduced along the diagonals within some of the quadrilateral facets. Indeed, folding a paper model of a four-sided flasher reveals these crease lines around which the quadrilateral facets  $f_4$  and  $f_5$  bend (Fig. 7).

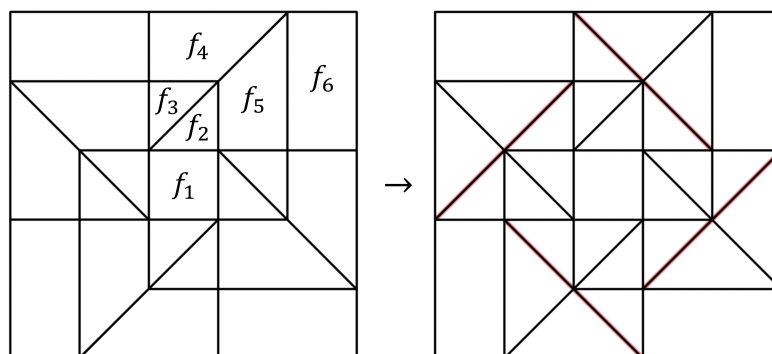


Fig. 7: Existing four-sided flasher pattern (left) and adapted flasher pattern (right) that folds rigidly because of the additional crease lines

Introducing these crease lines leads to a novel flasher pattern (Fig. 7 right), here called the *adapted flasher*, whose kinematic determinacy can be analyzed by subjecting it to various methods listed in Section 2.5. Here, the matrix rank method by Cai et al. [37] is applied, which yields a Jacobian matrix rank of fifteen for fifteen unknowns (five vertices in motion, each with three coordinates  $x$ ,  $y$ , and  $z$ ), confirming that the pattern is kinematically determinate. In addition, the rigid foldability is assessed by simulating the adapted flasher with the simulation method introduced in Section 3.2. To fold the adapted flasher, the central facet  $f_1$  in Fig. 7 (left) is fixed in space and  $f_2$  as well as its symmetric equivalents are folded around the shared crease lines by  $-\frac{\pi}{2}$  radians. Fig. 8 depicts the folding motion of the adapted flasher from left to right, resulting in a rigidity error  $\Gamma = 1.6 * 10^{-6}$  and distortions  $\delta^{(j)} \leq 10^{-4} \forall j$ , which numerically demonstrates that the adapted flasher is indeed rigidly foldable.

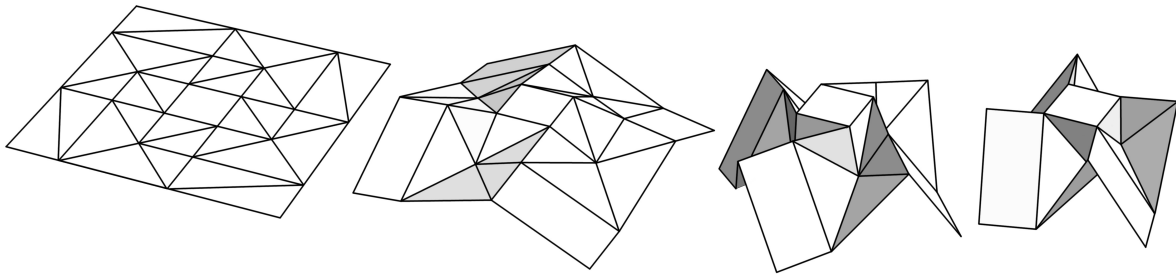


Fig. 8: Adapted flasher pattern folding rigidly from left to right

While the adapted flasher is rigidly foldable, a problem arises with respect to its folded shape: flashers are supposed to fold completely into a vertical cuboid shape, similar to a downward extrusion of the central facet. As depicted in Fig. 8 on the right, however, the adapted flasher is still partly unfolded after the above-described actuation, and an additional actuation of the outer layers would have to be introduced to completely fold the pattern into a cuboid. This sequential folding demands more actuators and complicates the control of the adapted flasher, which renders the adapted flasher less attractive for practical applications and leads to the formulation of the design task.

### 3.4 Design Task

To mimic the behavior of non-rigidly foldable flashers in a solar panel scenario, the goal of the design task for the remainder of Section 3 is to optimize the vertex geometry of the adapted flasher such that it folds into the red cuboid illustrated in Fig. 9b while maintaining a maximally large surface area for the collection of solar energy. The crease pattern in focus is the truncated flasher pattern depicted in Fig. 9a, which shows the allocation of vertices for one side of the four-sided pattern. The input to the kinematic simulation is a total number of iterations  $J = 30$  and the crease pattern in Fig. 9a with vertices

and edges. The center of facet  $f_1$  is fixed at the origin  $O$  and no rotation is allowed. The vertex  $v_1$  is located at  $\mathbf{x}_1^{(0)} = (1, 1, 0)$ , which defines the enclosing volume as a cuboid (Fig. 9b) of the dimensions  $-1 \leq x \leq 1, -1 \leq y \leq 1, -4 \leq z < 0$ . Note that the strict inequality for the upper boundary of  $z$  prevents self-intersection with the facet  $f_1$ . The locations of the remaining vertices  $v_{2-5}$  are not determined yet and have to be adjusted to satisfy the design task.

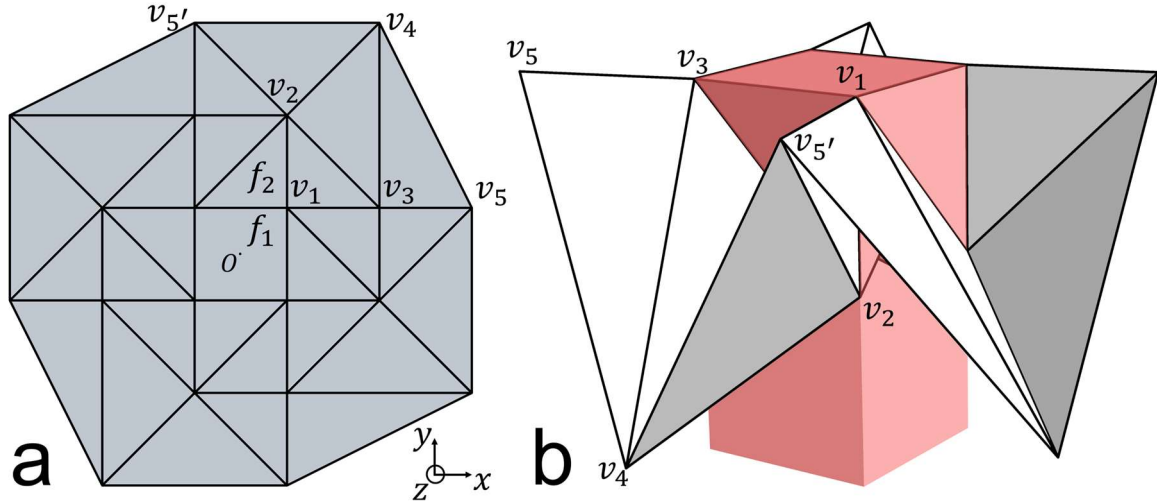


Fig. 9: (a) Rigidly foldable four-sided flasher pattern whose folded state should eventually fit into the depicted cuboid (b)

### 3.5 Sensitivity

In this section, the adapted flasher pattern is subjected to a sensitivity analysis to examine the influence of the vertex geometry on the kinematic behavior of the pattern. The focus of the analysis lies on the visualization of the kinematic search space characteristics to precipitate a better understanding of the kinematic behavior, in particular of the rigid foldability (RQ1) and the RBMs (RQ2).

The pattern in Fig. 9a can be perceived as being composed of ring-wise *layers*. The  $0^{th}$  layer consists of the central facet  $f_1$ , the first layer is bounded by vertices  $v_2, v_3$  and their rotations, and the second layer is bounded by vertices  $v_4, v_5$  and their rotations (such as  $v_5'$ ). The sensitivity analysis of the adapted flasher is structured layer-wise from inside to outside. First, the sector angles of  $v_1$  are examined in relation to rigid foldability in order to fix a starting location for  $v_2$  that then remains constant throughout the sensitivity analysis. This step also determines the trajectory of  $v_2$  and thus the actuation that is required as an input to the simulation. Subsequently, the effects of the starting location of  $v_3$  on the kinematic behavior of the first layer are analyzed, after which the influence of the starting location of  $v_4$  on the behavior of the entire crease pattern is demonstrated.

### 3.5.1 Sector Angles around $v_1$

Vertex  $v_1$  of the adapted flasher is a degree-5 vertex and thus perfectly suited for an analysis of rigid foldability since there exists little data on the rigid foldability of degree- $n$  vertices for  $n > 4$ . To do so, the outermost layer of the adapted flasher is ignored for now and the kinematic properties of  $v_1$  are visualized by the rigidity error  $\Gamma$  over two sector angles.

To parametrize  $v_1$  with two sector angles, three of its five sector angles need to be eliminated. Vertex  $v_1$  in Fig. 10a is developable, which eliminates  $\alpha_5 = 2\pi - \sum_{i=1}^4 \alpha_i$ . As defined by the starting location of  $v_1$ , the facet  $f_1$  is a square, and  $\alpha_2 = \frac{\pi}{2}$ . To conform the motion of the adapted flasher to the conventional flasher design [87], the  $y$ -axis component of the starting location of  $v_2$  is constrained to  $y_2^{(0)} = 1 + l_c = 3$ , where  $l_c$  is the side length of the central facet  $f_1$  (Fig. 10a). This constraint then leads to  $\alpha_3 = \frac{\tan \alpha_1}{\tan \alpha_1 - 1}$ , which leaves the two remaining parameters  $\alpha_1$  and  $\alpha_4$ . To decouple these parameters, instead of  $\alpha_4$  the sensitivity is performed with the sector angle  $\alpha'_4$  (Fig. 10a) that starts from the stationary axis  $y = 1$ .

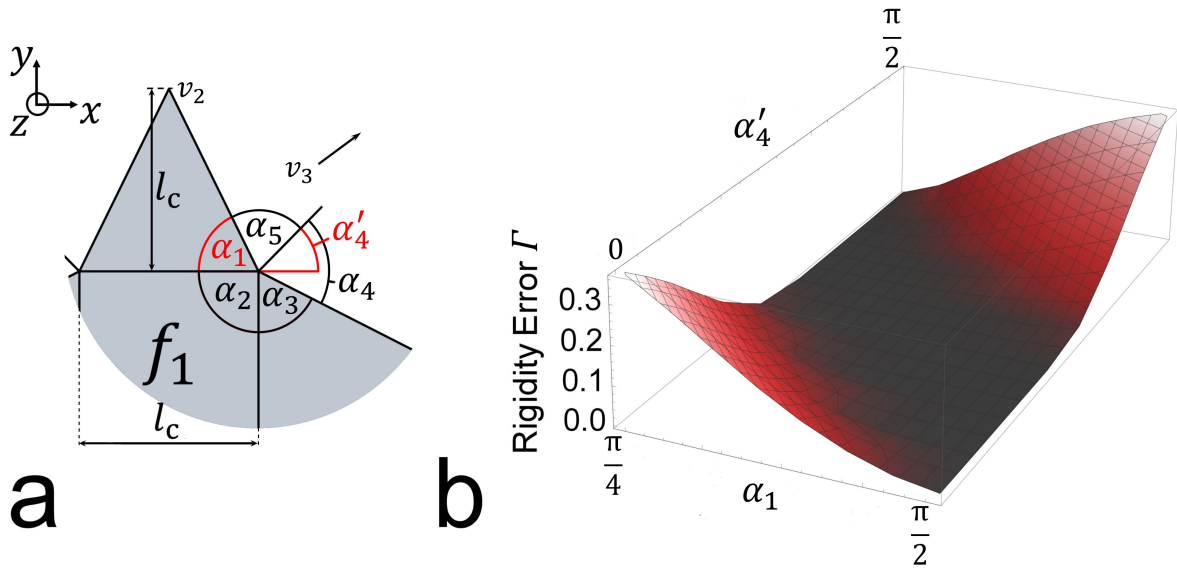


Fig. 10: Parametrization of sector angles around  $v_1$  (a) and the rigidity error  $\Gamma$  for  $\frac{\pi}{4} \leq \alpha_1 \leq \frac{\pi}{2}$  and  $0 < \alpha'_4 < \frac{\pi}{2}$ , revealing a “rigid foldability valley” (b).

Fig. 10b shows the result of the rigidity error  $\Gamma$  for  $\frac{\pi}{4} \leq \alpha_1 \leq \frac{\pi}{2}$  and  $0 < \alpha'_4 < \frac{\pi}{2}$  and reveals a smooth surface that exhibits the shape of a valley. The valley is constituted by both a flat bottom of constant width that runs linearly across the search space and by slopes that rise monotonically on either side. The sector angle configurations in the valley correspond to rigidly foldable vertex geometries satisfying  $\delta^{(j)} \leq 10^{-4} \forall j$ , whereas the configurations on the slope of the valley are not rigidly foldable.

This broad valley underlying the rigid foldability of a single vertex is an interesting result considering that most symbolic sector angle relations seem to indicate that rigid foldability occurs in specific sector angle configurations rather than wide regions. Even more striking is that such a smooth behavior is exhibited by a degree-5 vertex, for which there exist no sector angle relations due to the high level in geometric complexity.

For the purpose of the sensitivity analysis, the broad flat bottom of the valley signifies that there is ample choice of rigidly foldable starting locations for  $v_2$ . For aesthetic reasons, i.e. rotational symmetry as well as mirror symmetry,  $\alpha_1$  is set to  $\alpha_1 = \tan(l_c)$ . This definition results in the facet geometry depicted in Fig. 10a (grey facets), the starting location  $\mathbf{x}_2^{(0)} = (0, 3, 0)$ , and the trajectory  $\mathbf{x}_2(t) = (0, 1 + 2 \cos t, -2 \sin t)$  that actuates the folding process in  $-\frac{\pi}{2} \leq t \leq 0$ . For the remainder of Section 3, the starting location of  $v_2$  remains unchanged.

### 3.5.2 Influence of $\mathbf{x}_3^{(0)}$ on the Behavior of the First Layer

Since the starting locations of  $v_1$  and  $v_2$  are determined,  $v_3$  is the only remaining vertex to be analyzed in the first layer. Now, the sensitivity analysis is expanded from the pure assessment of rigid foldability to the full output of the simulation method that also includes the end locations of the vertices and the measure for self-intersection. In addition,  $v_3$  is allowed to deviate from the flat plane in order to investigate the behavior of a non-developable crease pattern. For this investigation, the starting location of  $v_3$  is parametrized with respect to the Cartesian coordinate system and adjusted incrementally within the range  $1.1 \leq x_3^{(0)} \leq 3, 1.1 \leq y_3^{(0)} \leq 3, -1 \leq z_3^{(0)} \leq 1$  in discrete steps of 0.1.

Fig. 11 shows the influence of the starting location of  $v_3$  on the behavior of the first layer of the adapted flasher. The starting configurations  $x_3^{(0)}, y_3^{(0)}$ , and  $z_3^{(0)}$  constitute the axes of all plots (a-f) while the legends in Fig. 11 use color schemes to represent the results of the end locations of  $v_3$  (a-c), the rigidity error  $\Gamma$  (d), and the self-intersection measure  $\theta$  (e). The feasible solutions (f) are found by superimposing the plots in Fig. 11a-e. Due to the mirror symmetry of the first layer, the plots are sliced into half-spaces to provide a better view of their insides. The half-spaces shown in (c-f) can be completed by mirroring the results on the plane  $x_3^{(0)} = y_3^{(0)}$ , whereas the mirrored image of (a) would complete (b) and vice versa.

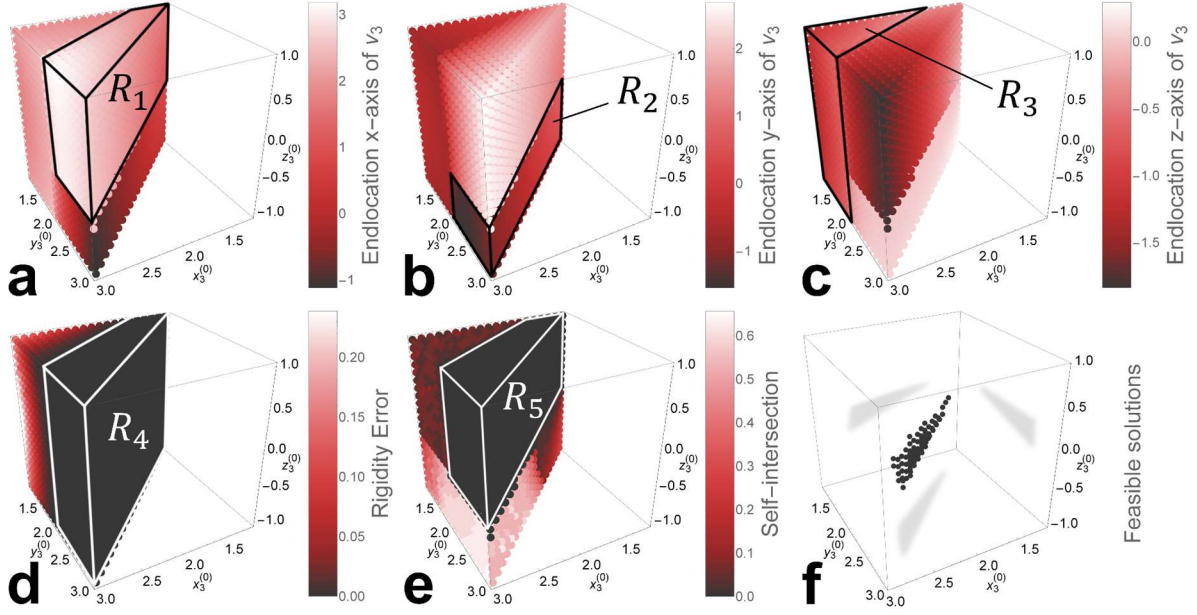


Fig. 11: Influence of the starting location of  $v_3$  on the first layer of the adapted flasher: the absolute end locations of  $v_3$  on the  $x$ ,  $y$ , and  $z$ -axes (a-c), the rigidity error (d), the self-intersection measure (e), the feasible solutions (f) found by superposition of (a-e), and the regions  $R_1$  to  $R_5$ .

The most apparent feature of the results is the formation of the three-dimensional *regions*  $R_1$  to  $R_5$ . Regions are bounded subspaces that signify distinct differences of the kinematic behavior in contrast to their neighboring subspaces. Within the highlighted regions, the kinematic behavior is continuous or constant but changes abruptly if a mutual boundary is crossed by adjusting the vertex geometry.

In comparison to the boundaries of the regions  $R_{1-3}$  that arise from the abrupt change of the vertex end locations, the transition from true rigid foldability to the onset of distortion in Fig. 11d is more continuous and thus harder to visually determine. However, the boundary of  $R_4$  in Fig. 11d can be determined by transferring the vertical boundary between region  $R_3$  and the regions  $R_1$  and  $R_2$  (Fig. 11a-c). This boundary can then be used to identify the numerical difference between rigidly foldable and distorting crease patterns, resulting in the condition that the distortions in each iteration  $j$  have to satisfy  $\delta^{(j)} \leq 10^{-4}$  as mentioned in Section 3.2.3. The result of the rigidity error in Fig. 11d further reconfirms the rigid foldability valley that already appeared in Fig. 10b, this time in three dimensions and independent of  $z_3^{(0)}$ , which demonstrates that the rigid foldability of vertex  $v_1$  is only dependent on the sector angle  $\alpha'_4$  once  $\alpha_1$  is constant (Fig. 10a).

The regions  $R_1$  and  $R_2$  are separated horizontally by a plane with a slightly negative inclination toward increasing  $x_3^{(0)}$  and  $y_3^{(0)}$ , as observable in Fig. 11c. The reason for this boundary is the existence

of two RBMs. The RBM into which the origami falls is determined by  $z_3^{(0)}$ : for slightly negative values of  $z_3^{(0)}$  the crease line between  $v_1$  and  $v_3$  becomes a mountain (**M** in Fig. 12), and further decreasing  $z_3^{(0)}$  leads to a respective valley crease (**V** in Fig. 12).

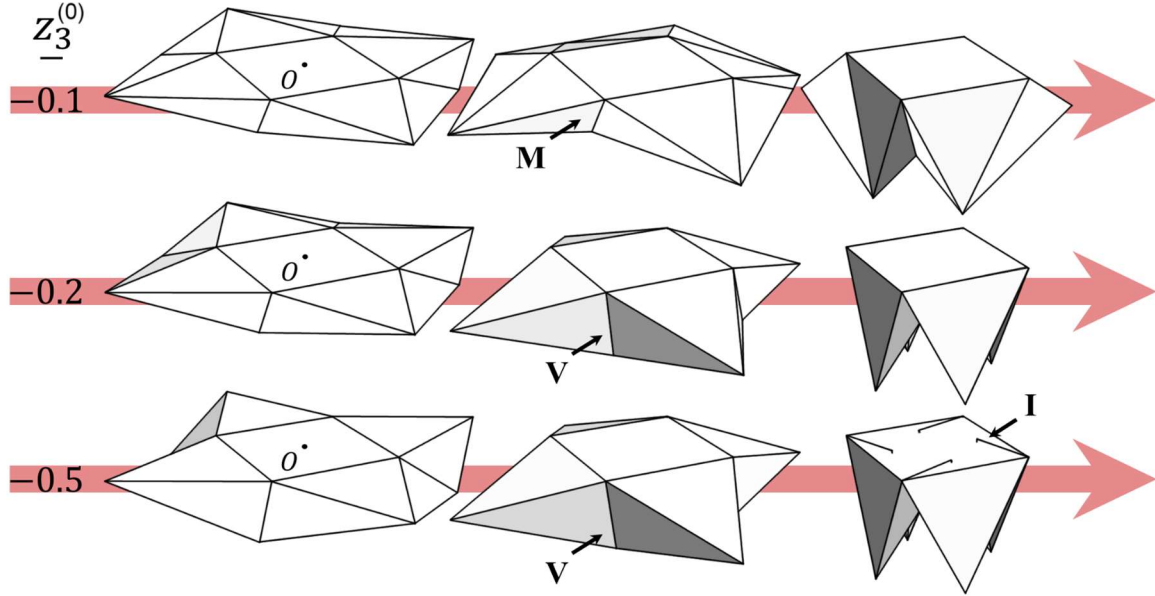


Fig. 12: Influence of  $z_3^{(0)}$ : a slightly positive value leads to a mountain crease **M**, and slightly negative values lead to valleys **V** and eventually to self-intersection **I**

Fig. 12 shows starting configurations of  $x_3^{(0)} = 1.5$ ,  $y_3^{(0)} = 1.7$ , and decreasing values of  $z_3^{(0)}$  from top to bottom. If  $z_3^{(0)} = -0.1$ , the vertex geometry lies within region  $R_1$  (Fig. 11a) and  $v_3$  folds on the “outer” side of the origami and away from the origin  $O$ . For  $z_3^{(0)} = -0.2$ , which relocates the starting configuration into region  $R_2$  (Fig. 11b),  $v_3$  folds on the “inside” toward the origin  $O$  and into the cuboid. If  $z_3 = -0.5$ , the starting configuration is still within region  $R_2$  and thus exhibits the same RBM as  $z_3^{(0)} = -0.2$ , but the facets adjacent to  $v_3$  intersect with the central facet  $f_1$  (as denoted with **I** in Fig. 12). Hence, the boundaries between the end locations and self-intersection are coupled but not identical: although regions  $R_1$  and  $R_5$  largely overlap, the inclinations of the separating planes differ, and region  $R_5$  transcends the boundary between regions  $R_1$  and  $R_2$  (Fig. 11). This intersection produces a wedge of feasible solutions that fold rigidly into the cuboid without self-intersecting (Fig. 11f).

### 3.5.3 Influence of $x_4^{(0)}$ on the Behavior of the Entire Pattern

While the sensitivity of a single vertex is investigated in the previous section, this section is interested in the generality of the findings, in particular of the region-specific behavior. For this purpose, the sensitivity analysis is extended to include the outermost layer of the adapted flasher, and the starting

locations of  $v_3$  and  $v_5$  are fixed to examine only the effect of  $\mathbf{x}_4^{(0)}$ . Note that the choice of fixing the starting location  $\mathbf{x}_5^{(0)}$  and adjusting  $\mathbf{x}_4^{(0)}$  is interchangeable since both vertices belong to the second layer and are connected to the same vertices in the first layer.

The starting location of  $v_3$  is fixed in its feasible solution space,  $\mathbf{x}_3^{(0)} = (1.2, 1.3, 0)$  so that the first layer satisfies the design task. The starting location of  $v_5$  is arbitrarily set to  $\mathbf{x}_5^{(0)} = (3.1, 2, -0.2)$  and the range of the starting locations of  $v_4$  is  $-1 \leq x_4^{(0)} \leq 1, 3 \leq y_4^{(0)} \leq 4, -0.5 \leq z_4^{(0)} \leq 0.5$ . Fig. 13 shows the results for the end locations of all vertices  $v_3$  to  $v_5$  (a-i), the rigidity error (j), the self-intersection (k), and the feasible solutions that do not exist in the space analyzed.

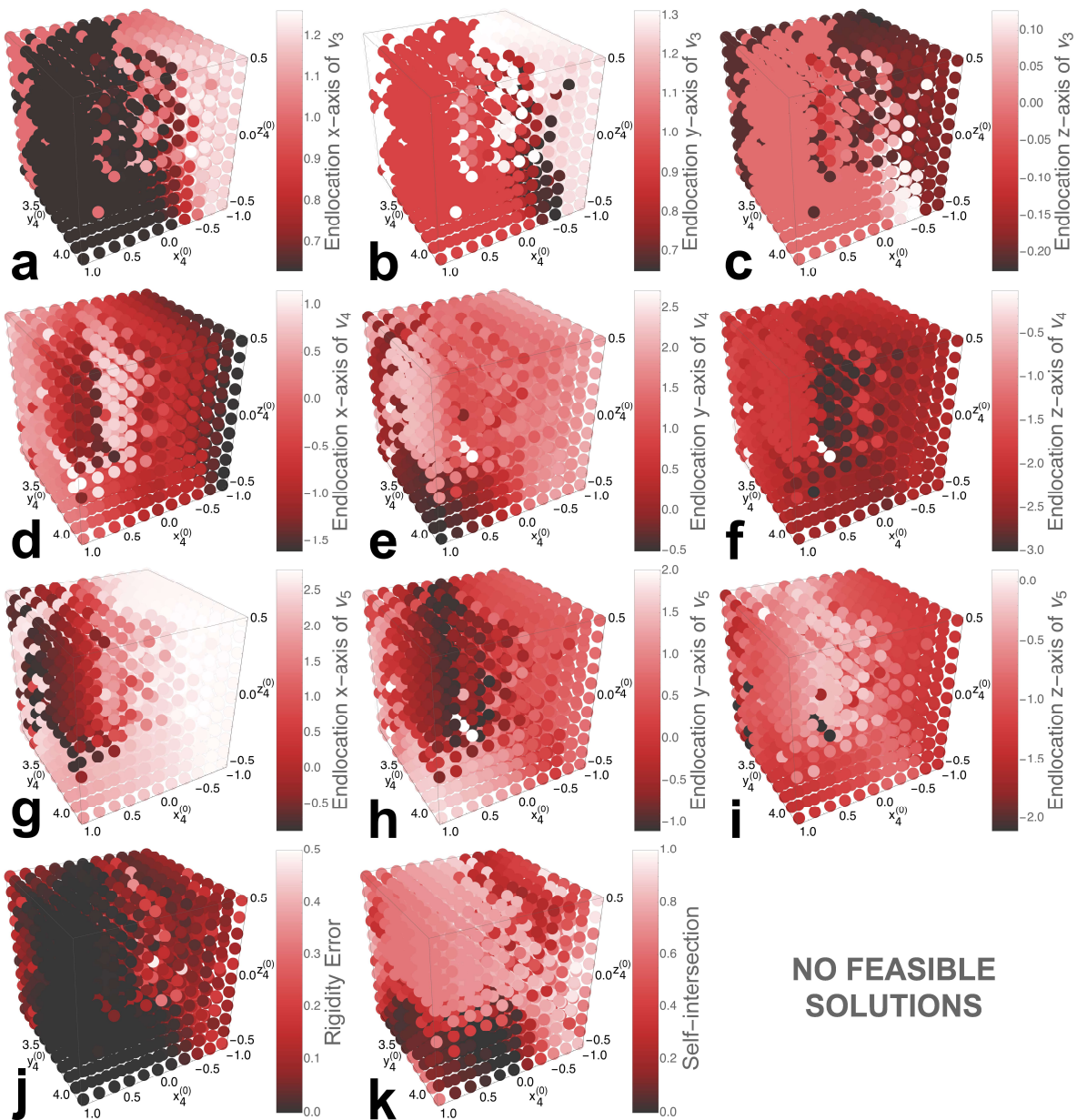




Fig. 13: Influence of the starting location of  $v_4$  on the behavior of the entire adapted flasher pattern: the absolute end locations of vertices  $v_3$ ,  $v_4$ , and  $v_5$  on the  $x$ ,  $y$ , and  $z$ -axes (a-i), the rigidity error (j), and the measure for self-intersection (k).

In comparison to Fig. 12, the striking feature in all plots of Fig. 13 is the noisy kinematic behavior. Although there seem to exist regions with smooth behavior, the search space is interspersed with irregular configurations that behave completely different from their surrounding ones. This chaotic behavior cannot be properly explained here since the reason could be both the origami search space or the chaotic behavior of the underlying optimizer [110]. However, a much smoother behavior is revealed when the non-rigid configurations are eliminated, which is illustrated in Fig. 14 for the end locations of all vertices  $v_3$ ,  $v_4$ , and  $v_5$ .

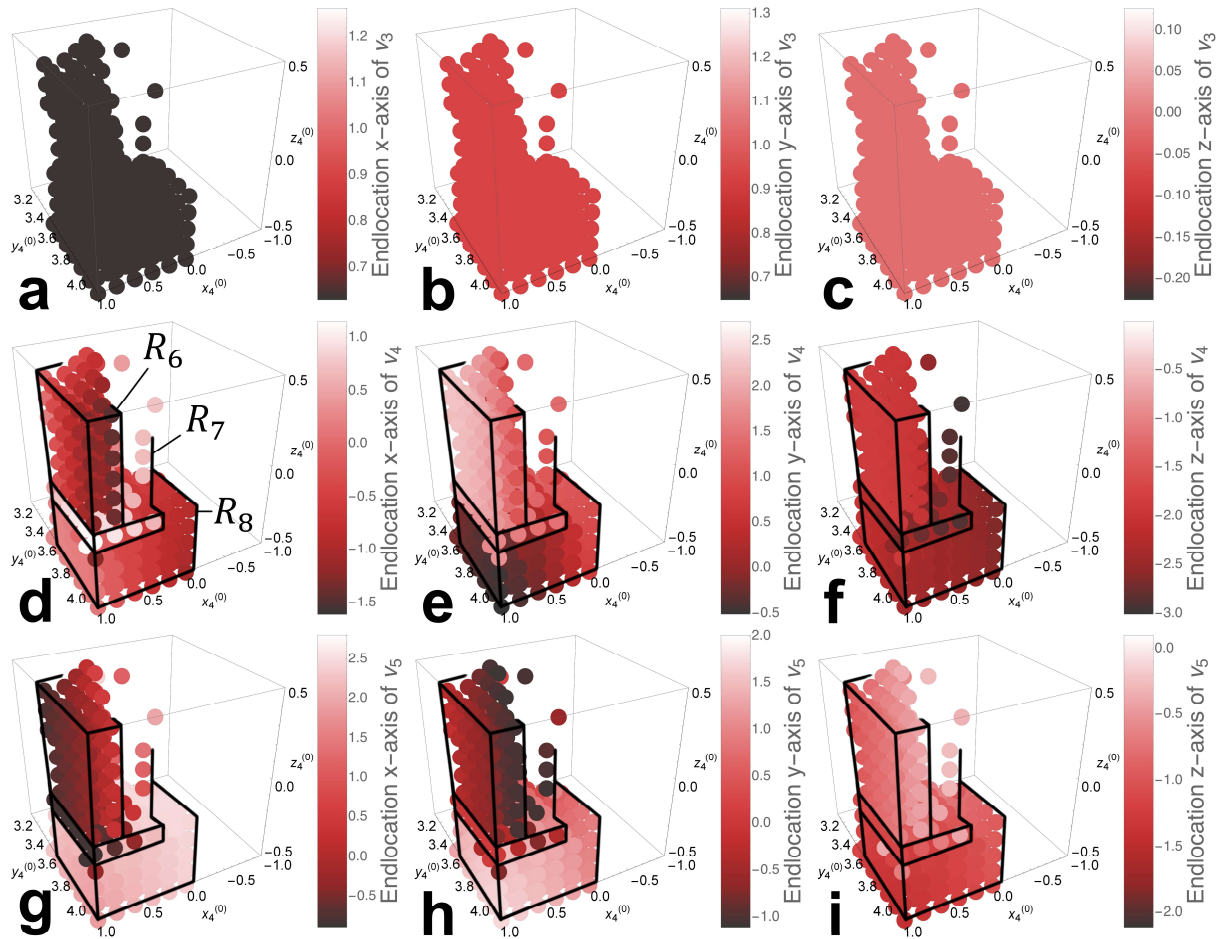


Fig. 14: End locations of all vertices  $v_3$ ,  $v_4$ , and  $v_5$  only for rigidly foldable configurations. Again, there emerge distinct regions within which the kinematic behavior is constant, or smooth and monotonic.

The end locations of  $v_3$  for purely rigidly foldable vertex geometries are constant, which means that  $v_3$  is always folded to the exact same location. This result hints at the hierarchy of layers: if the outer layer of the adapted flasher folds rigidly, the kinematic behavior of the inner layer is not influenced. If in

contrast the outer layer does not fold rigidly, it can distort the entire pattern, which follows from the sum of the errors in Eq. (3).

Again, the end locations of  $v_4$  and  $v_5$  in Fig. 14 result in the formation of regions  $R_6$ ,  $R_7$ , and  $R_8$  in Fig. 14d, although the boundaries of the regions are less regular and thus harder to delineate. Still, the kinematic behavior within these regions is smooth and monotonic, indicating the special characteristics of the search space that corresponds to rigid foldability. Moreover, the abrupt change in kinematic behavior between the regions in the vertical direction are again caused by different RBMs as depicted in Fig. 15 where the plot from Fig. 14d serves as a reference for the locations of three RBMs.

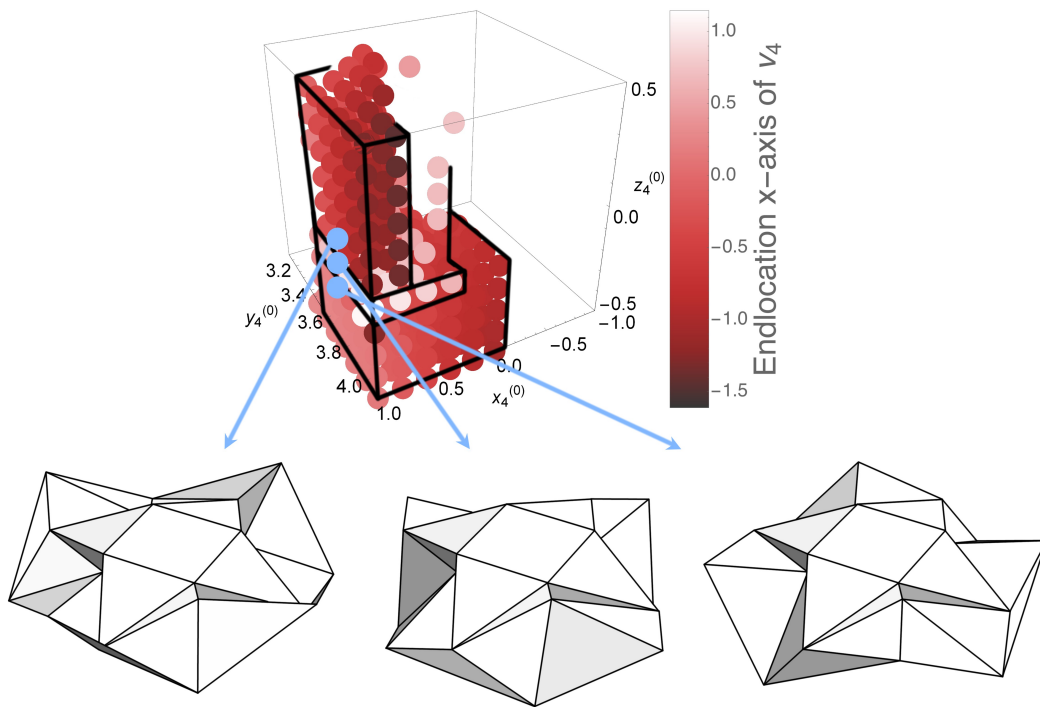


Fig. 15: Three RBMs from left to right located at  $x_4^{(0)} = 1$ ,  $y_4^{(0)} = 3.6$ , and  $z_4^{(0)} = 0.2$ ,  $z_4^{(0)} = 0.0$ , as well as  $z_4^{(0)} = -0.2$ , respectively

### 3.6 Guidance

As shown in Fig. 4, the numerical approach does not incorporate the computational generation of crease patterns, which is why the guidance in this section simply equates to the optimization of the adapted flasher pattern. The sensitivity analysis in the last section leads to a handful of findings that influence the choice of optimization strategy with which the design space of an origami is searched for optimized designs that satisfy the given engineering design task. The kinematic behavior of a single vertex is smooth and monotonic within rigidly foldable regions whose intermediate boundaries are influenced by the existence of different RBM. In contrast, the assessment of multiple coupled vertices

yields noisy results that exhibit smooth behavior only within regions that are rigidly foldable. Although the simulation method enables the visualization of the search space, it does not yield a method to determine the boundaries of the rigidly foldable regions without a prior sensitivity analysis. In addition, the guidance within a computational method needs to be able to guide the search independent of the underlying crease pattern topology. Considering the expansion of the method to arbitrarily complicated crease pattern topologies, the optimization strategy needs to be stochastic, handle large sets of variables, and deal with a noisy kinematic behavior, which is why the vertex geometry of the adapted flasher is optimized by using the Simulated Annealing algorithm [111].

As stated in Section 3.4, the goal of the design task is to find a vertex geometry that maximizes the surface area of the adapted flasher while ensuring that the crease pattern folds rigidly into the given cuboid (Fig. 9b) without self-intersection. The objective function is formulated as  $-\frac{A}{w} + (\Gamma + \theta + \Omega)$ , where  $\frac{A}{w}$  is the projected surface area  $A$  divided by a weight  $w$  and  $(\Gamma + \theta + \Omega)$  is a penalty term that includes the rigidity error  $\Gamma$ , the self-intersection measure  $\theta$ , and a function  $\Omega$  for the spatial feasibility that penalizes vertices whose end locations lie outside of the cuboid. The weight is determined heuristically to be  $w = 50$  so that the projected area  $A$  contributes to the objective function in the approximate magnitude of the penalty term in the final stages of the optimization. The function  $\Omega$  for the spatial feasibility of each vertex is expressed by the sum of the distances between the vertex end locations and the cuboid in separate directions  $x, y$ , and  $z$ . Thus, the penalty term is zero for a rigidly foldable configuration that folds into the cuboid without self-intersecting, and the only influencing factor within the spatially feasible region is the size of the projected surface area  $A$ .

The variables to be optimized are the starting locations  $\mathbf{x}_3^{(0)}$ ,  $\mathbf{x}_4^{(0)}$ , and  $\mathbf{x}_5^{(0)}$ . In comparison to Cartesian coordinates that facilitate the visual representation of the search space (Fig. 11, Fig. 13, and Fig. 14), spherical coordinates are better suited for the optimization of crease patterns that represent spherical mechanisms [112]. Hence, the variables are expressed by crease line lengths  $l$  as well as polar and azimuth angles  $\theta$  and  $\vartheta$ , respectively. To decouple the variables, the coordinates of  $v_3$  are related to  $v_1$ , and the vertices  $v_4$  and  $v_5$  are related to  $v_2$  and its rotations (Fig. 9a), respectively.

The optimization scheme is then formulated as:

$$\begin{aligned} & \min_{\mathbf{r}^{(0)}, \theta^{(0)}, \vartheta^{(0)}} -\frac{A}{50} + (\Gamma + \theta + \Omega) \\ & 0.4 \leq l_3^{(0)} \leq 1.5, -45^\circ \leq \theta_3^{(0)} \leq 0^\circ, 20^\circ \leq \vartheta_3^{(0)} \leq 70^\circ \\ & 0.5 \leq l_4^{(0)} \leq 3.0, -30^\circ \leq \theta_4^{(0)} \leq 30^\circ, 45^\circ \leq \vartheta_4^{(0)} \leq 135^\circ \\ & 0.5 \leq l_5^{(0)} \leq 3.0, -30^\circ \leq \theta_5^{(0)} \leq 30^\circ, 45^\circ \leq \vartheta_5^{(0)} \leq 135^\circ \end{aligned} \quad (5)$$

The parameters of the Simulated Annealing algorithm are tuned according to Downsland and Thompson [113]. The starting temperature is defined as 200, which is the approximate maximum change of the objective function in early iterations, and the temperature schedule is logarithmic with a reduction factor of 0.89. Both the number of inner loops and the number of outer loops is 50, resulting in 2'500 objective function evaluations. The size of the neighborhood for the variables is one tenth of their range, as listed in Eq. (5).

### 3.7 Results

A single kinematic simulation of the flasher pattern with  $J = 30$  iterations takes about 10 seconds on an Intel i7 processor with 16GB RAM. The total runtime of the optimization with 2'500 such simulations (and thus objective function evaluations) is approximately 8.5 hours.

The optimization is performed 20 times to determine the convergence characteristics of the method, and of these runs the worst and the best objective values are  $-0.26$  and  $-0.88$ , respectively, with a mean of  $-0.60$  and a standard deviation of  $-0.22$ . A typical run is depicted in Fig. 16, which plots the objective values over all objective function evaluations.

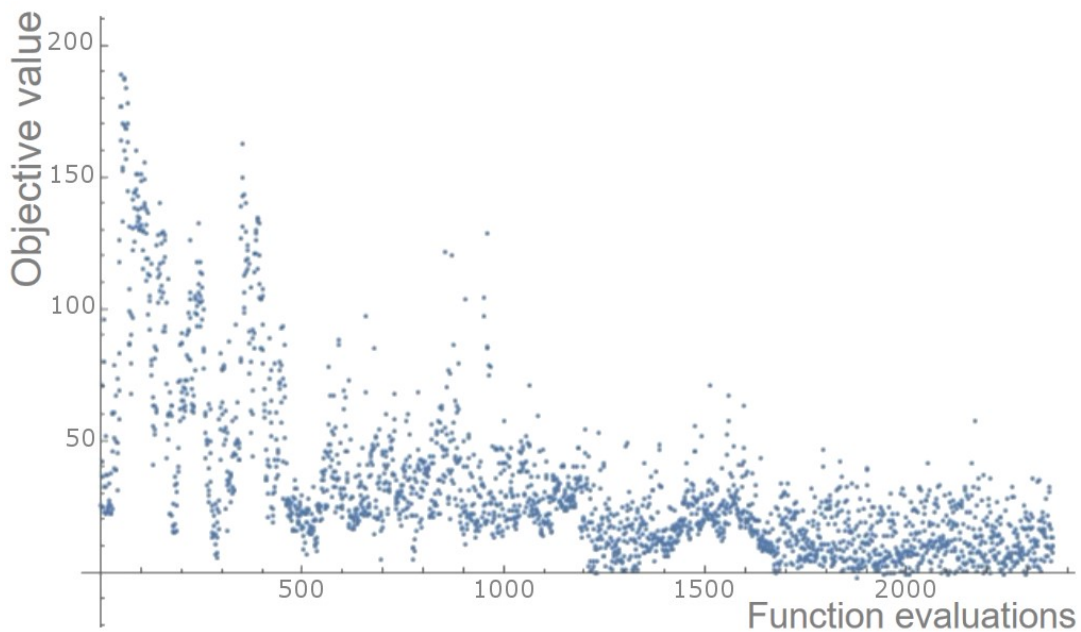


Fig. 16: Objective values over function evaluations in a typical optimization run for the adapted flasher pattern

The best solution found is illustrated in Fig. 17a together with the original adapted flasher pattern. The optimized vertex geometry exhibits an objective value of  $-0.88$  that corresponds to the starting locations  $\mathbf{x}_3^{(0)} = (1.50, 1.39, -0.10)$ ,  $\mathbf{x}_4^{(0)} = (0.53, 5.28, 0.03)$ , and  $\mathbf{x}_5^{(0)} = (2.15, 1.78, -0.43)$ , a projected surface area of 44.0 (in comparison to the original surface area of 84.0), a rigidity error  $\Gamma =$

$1.6 * 10^{-6}$  with  $\delta^{(j)} \leq 10^{-4} \forall j$ , an intersection measure  $\theta = 0$ , and a spatial feasibility  $\Omega = 0$  (in comparison to the original spatial feasibility of 4.39). Fig. 17b shows the folding motion of the best solution from left to right with the cuboid superimposed in the final folded state.

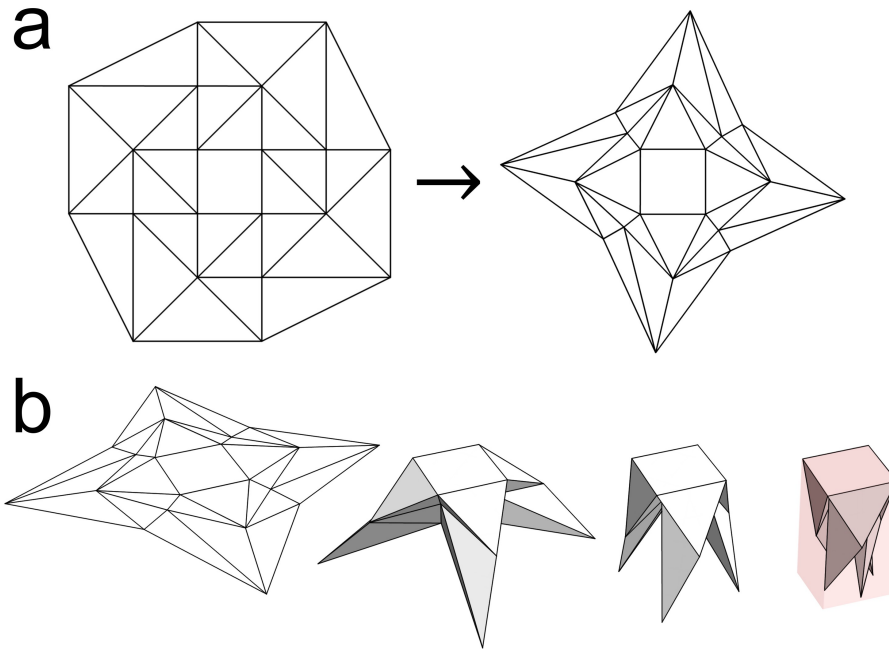


Fig. 17: The optimized adapted flasher pattern folding from left to right into the prescribed cuboid

### 3.8 Discussion

The formulation of Eq. (3) as an optimization problem enables the presented simulation method to compute the folding motion of rigidly and non-rigidly vertex geometries independent of the extent of distortion. The simulation method further models different RBMs and provides a feedback for the state of rigid foldability by returning the rigidity error and the distortions in each iteration. These distortions can then be used to classify patterns into rigidly and non-rigidly foldable configurations with a limit of  $\delta^{(j)} \leq 10^{-4} \forall j$ .

The manual adaptation of an existing flasher pattern leads to a novel flasher pattern whose kinematic determinacy and rigid foldability are confirmed by a matrix rank assessment and the simulation through the presented simulation method, respectively. The adapted flasher offers a suitable basis for the analysis of the vertex geometry by incorporating degree-5 vertices and by offering layers that can be assessed in a stepwise manner.

With respect to RQ1, the sensitivity analysis shows the existence of regions within the kinematic search space of the vertex geometry. In the case of a single vertex, these regions show well-defined boundaries within which the kinematic behavior is continuous. Fig. 10b shows an unexpectedly wide

rigid foldability valley, which is a counterintuitive result considering that the compatibility conditions known for degree-4 vertices [87] seem to imply isolated rigidly foldable configurations. The region-specific behavior is also present in the kinematic search space of a non-developable vertex in Fig. 11 that offers an in-depth analysis of the connection between vertex end locations, rigid foldability, and self-intersection. However, determining the exact location and orientation of the region boundaries becomes more complex for multiple connected vertices. Fig. 13 illustrates the emergence of noisy regions whose cause cannot be identified by the sensitivity analysis. What remedies the situation is that these noisy regions do not coincide with rigidly foldable regions, which is why they can be eliminated by considering only rigid configurations. Then, the behavior within the rigidly foldable regions is again smooth and monotonic and changes only when the respective boundaries are crossed. While the discovery of these rigidly foldable regions implies that the mathematics of folding follow a much more regular behavior than anticipated considering the geometric complexity, the numerical simulation neither provides a reason for the regions nor a mathematical model to determine the locations of their boundaries in the general case.

The boundaries between two rigidly foldable regions are caused by different RBMs addressed in RQ2. Fig. 11 demonstrates the existence of two different RBMs exhibited by a single vertex, which is expected, as explained in Section 2. The results in Fig. 15, however, show only three different RBMs although the adapted flasher pattern includes  $N = 2$  internal vertices, which would suggest  $2^N = 4$  RBMs. Since Fig. 14 and Fig. 15 only depict part of the search space, the fourth RBM must be located in a different part that has not been analyzed. However, the sensitivity analysis does not reveal where the missing RBM is located, and discovering all RBMs in the general case would require an analysis of the entire search space. In addition, the discovery of RBMs is enabled mostly by allowing non-developable configurations of the adapted flasher pattern, which can be observed in, e.g., Fig. 12 where the boundary between different RBMs is located between  $z_3^{(0)} = -0.1$  and  $z_3^{(0)} = -0.2$ . However, according to related works [62], the exponential number of different RBMs are exhibited by completely flat, developable patterns. Thus, although the numerical simulation method is able to discover RBMs, it can only do so if the vertices deviate from the flat initial state.

The stochastic optimization with a total runtime of approximately 8.5 hours for a single crease pattern topology is not satisfactory considering that the planned generative method should search a vast design space of different crease pattern topologies. The time-efficiency for the optimization procedure

can be increased by detecting the location of region boundaries to avoid noisy regions and by adjusting the vertex geometry to lie within rigidly foldable regions so that gradient based local optimization techniques can be applied to capitalize on the smoothness of the search space within these regions. Based on this idea, the author published a journal article [114] presenting an algorithm that relocates the locations of vertices into rigidly foldable regions relying on heuristics that utilize the distortions occurring during the folding motion. Although successful, the algorithm does not offer a choice for the specific region toward which the vertex geometry is adjusted. Unfortunately, there exist numerous rigidly foldable regions as demonstrated by Fig. 14 and Fig. 15, and determining which of these regions lead to feasible solutions in arbitrarily generated crease pattern topologies would require the introduction of additional heuristics.

While the numerical approach is able to successfully tailor the geometry of a single crease pattern topology to a given design task, expanding the approach to the assessment of multiple topologies is too time-consuming for an automated CDS method. Neither the stochastic optimization nor its combination with a heuristic algorithm are guaranteed to yield any useful results in the context of design automation, and the optimization of the vertex geometry resembles a passive search for suitable solutions rather than the direct, purposeful design of origami crease patterns. A more direct design approach thus requires deeper knowledge about the kinematics of origami.

Possible starting points for a deeper investigation are indicated by three key findings, the first of which is the discovery of the wide rigid foldability valley in Fig. 10b. In contrast to the noisy regions exhibited by multiple vertices, this valley stems from the analysis of a single vertex, implying that a closer inspection of single vertices could reveal more about the mathematical underpinnings of rigid foldability.

The reason for the boundaries of the rigidly foldable regions and the locations of these boundaries are yet undetermined. Including the vertex end locations in the analysis performed in relation to Fig. 10b results in the second finding that motivates a more thorough examination of origami kinematics: as highlighted in Fig. 18, the sector angle configurations coinciding with the boundary between rigid and non-rigid foldability are all flat foldable.

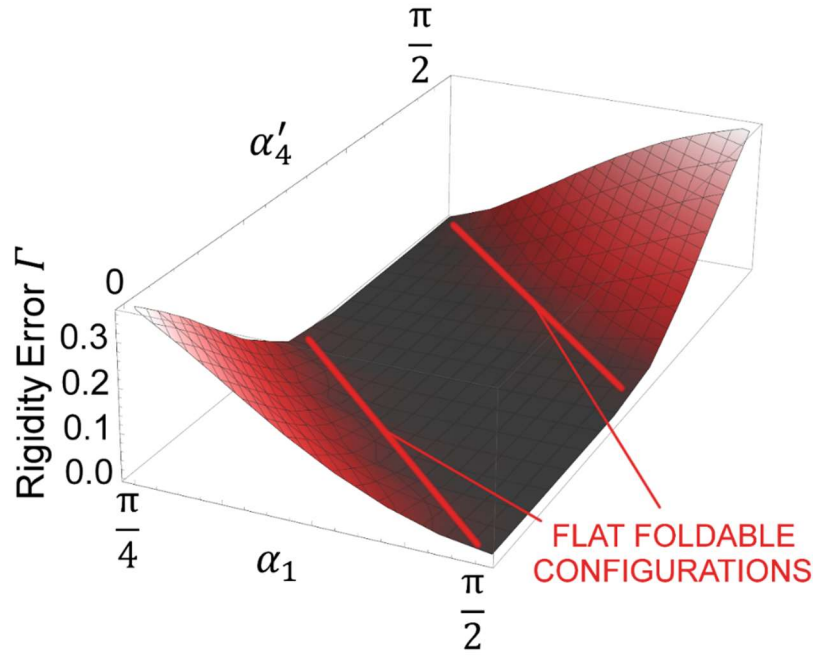


Fig. 18: Rigid foldability valley corresponding to Fig. 10b but with highlighted flat foldable configurations that emerge on the boundary between rigidly and non-rigidly foldable vertex geometries

A connection between rigid and flat foldability has been noted only in the formulation of the fold angle multipliers [91], but the exact nature of this connection has not been determined. Moreover, both works focus on and are limited to degree-4 vertices, and analogous relations for degree-5 vertices and higher order vertices have not been established. The result in Fig. 18 indicates that more knowledge could be gained by examining the connection between rigid and flat foldability, the two most important mathematical notions in origami.

The third key finding is the hierarchy of layers in Fig. 14, where rigidly foldable outer layers do not impact rigidly foldable inner layers. This phenomenon suggests that vertices closer to the actuation play a more important role for rigid foldability because they influence outer layers without experiencing the reciprocal effect, which hints at the possibility to generate crease pattern by adding single vertices to existing rigidly foldable layers. This motivates an investigation of single vertices as well as their relation to other single vertices within a crease pattern, which might further yield results for the kinematic relations in RQ3.

These three key findings hint at the existence of yet uncovered fundamental mathematical underpinnings whose application within a computational method offers the potential to surpass the performance achievable by any numerical approach.



## 4 Analytical Kinematics of Origami

Analytical kinematic relations and corresponding conditions between sector and dihedral angles, as described in Section 2, are scarce for the simplest single vertices of degree four and missing altogether for degree- $n$  vertices when  $n$  goes from 5 to infinity. Most existing crease patterns correspond to omnipresent degree-4 vertices [30, 56, 57, 99, 115], and a condition for the rigid foldability of degree- $n$  vertices would allow for the embedding of higher order vertices into crease patterns to provide more flexibility in the shapes that can be achieved by origami. In addition, the Kawasaki-Justin condition for flat foldability [97, 98] includes only single vertices of even degree, and a general connection between rigid and flat foldability is missing. Discovering this underlying connection between the two most important mathematical notions in origami might stimulate a better understanding of the kinematics of origami and facilitate the design of efficiently packed, rigidly and flat foldable crease patterns.

This section thus delves deeper into the mathematical underpinnings of folding and investigates the fundamental analytical kinematics of origami mechanisms. Motivated by Section 3, the analysis reverts to single vertices. To answer RQ1, the first focus lies on the conditions for rigid and flat foldability of degree-4 vertices, which are then generalized for single degree- $n$  vertices and entire crease patterns. In the generalization for degree- $n$  vertices, the RBMs are incorporated into the kinematic model as targeted by RQ2. Subsequently, the implications for the generation of crease patterns are presented, which provides insight into the relations between the kinematic determinacy, the number of DOF, and symmetry, as prompted by RQ3. The findings are then manually applied to readjust the flasher pattern and to generate new crease patterns, once for the exploration of RBMs and then for the design of a chair. Finally, the section is discussed and concluded.

### 4.1 Degree-4 Vertex

The content of Section 4.1 has been published as: [116] Zimmermann, L., and Stanković, T., "Rigid and Flat Foldability of a Degree-Four Vertex in Origami," *Journal of Mechanisms and Robotics*, 12(1), 2020.

The following subsection formally introduces the degree-four vertex and examines some properties of spherical four-bar mechanisms that are useful for the subsequent derivation of the necessary and sufficient condition for the rigid foldability of degree-4 vertices. Then, the purely sufficient condition is established and reduced to the Kawasaki-Justin condition for flat foldability.

#### 4.1.1 Preliminaries

Fig. 19 depicts a generic degree-four vertex with sector angles  $\alpha_i$  that are assigned in the counterclockwise direction with increasing  $i = 1, 2, 3, 4$ . The starting sector can be chosen arbitrarily. For convenience,  $\alpha_i$  is used to denote both the sector angles and the sectors interchangeably.

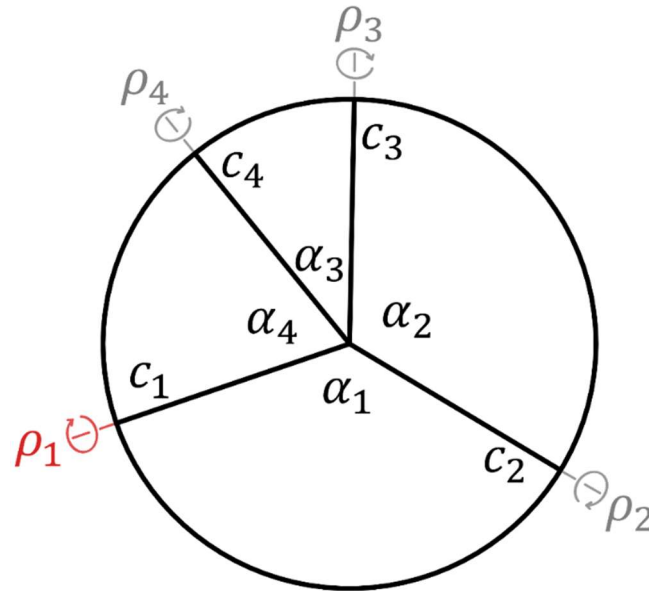


Fig. 19: Developable degree-four vertex with sector angles  $\alpha_{1-4}$ , crease lines  $c_{1-4}$ , and dihedral angles  $\rho_{1-4}$ , ordered in the counterclockwise direction. The default driving angle  $\rho_1$  is located between sectors  $\alpha_1$  and  $\alpha_4$ .

The degree-four vertex analyzed here is developable, which means that its unfolded state is flat and the sum of its sector angles  $\alpha_i$  is equal to a full rotation,  $\sum_i \alpha_i = 2\pi$ . A crease line is labelled  $c_i$  if it precedes the sector  $\alpha_i$  when the vertex is traversed in the counterclockwise direction. The single DOF required to drive the folding motion of the vertex [21, 74] is determined by prescribing one dihedral angle that will be called the *driving angle*. Unless stated otherwise,  $\rho_1$  is defined as the default driving angle for the remainder of Section 4.1, as illustrated in red in Fig. 19.

To explain the ideas that lead to the derivation of the necessary and sufficient condition, an analogy is introduced between the degree-four vertex and the century-old problem of a spherical four-bar linkage with four revolute (4R) joints (Fig. 20). This analogy is both valid and useful because the degree-four vertex has been shown to belong to the class of spherical four-bar mechanisms [100, 112, 117, 118] that have been extensively studied in the past [119]. In such mechanisms, an input crank ( $\alpha_4$  in Fig. 20a,  $\alpha_2$  in Fig. 20b) is rotated around a fixed frame ( $\alpha_1$ ), and its motion is transferred through a coupler ( $\alpha_3$ ) to the output crank ( $\alpha_2$  in Fig. 20a,  $\alpha_4$  in Fig. 20b). The possible range of motion for both input and output cranks is illustrated using a dashed line.

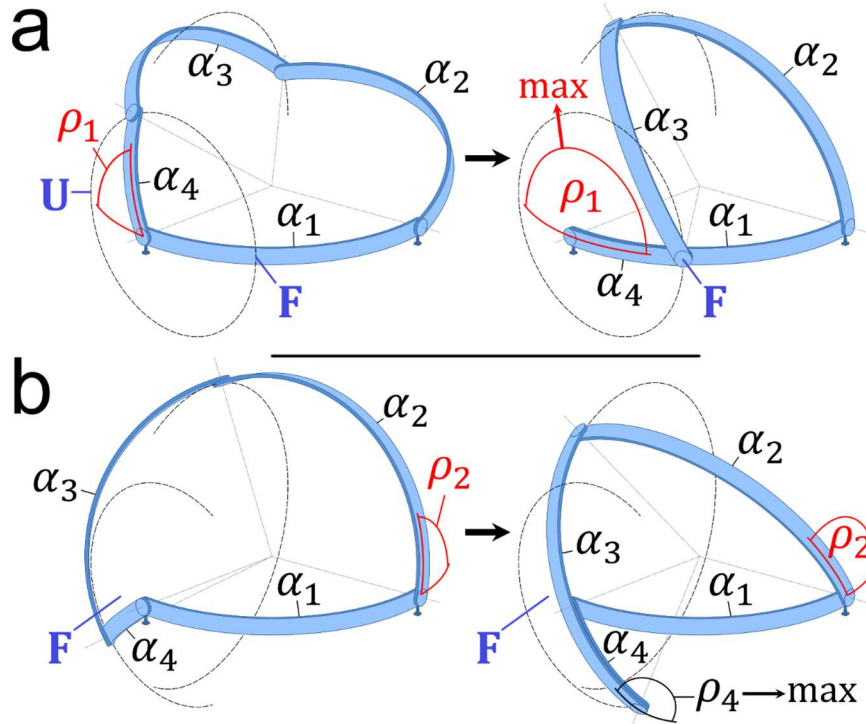


Fig. 20: Spherical four-bar linkage with highlighted driving angles  $\rho_1$  (a) and  $\rho_2$  (b), corresponding input cranks  $\alpha_4$  (a) and  $\alpha_2$  (b), fixed frame  $\alpha_1$ , coupler  $\alpha_3$ , as well as output cranks  $\alpha_2$  (a) and  $\alpha_4$  (b). The range of motion for both input and output cranks is illustrated using a dashed line for the corresponding trajectories. The unfolded state **U** and the fully folded states **F** are theoretical extreme states. The maximum (max) dihedral angle  $\rho_4$  is shown on the right-hand side of (b).

In a spherical four-bar linkage, the input crank is able to perform a full rotation only if it can rotate through both states **U** and **F** (Fig. 20a, left) [120] because these states represent the global extremes of the driving angle at  $\rho_1 = 0$  (**U**) and  $\rho_1 = \pm\pi$  (**F**). However, a developable origami is defined in its unfolded state (**U**) with all dihedral angles  $\rho_i = 0$ , which means that the unfolded state **U** is rigidly foldable in all cases. In an origami, the fully folded state **F** is thus the only critical state for an unrestricted range of the dihedral angle  $\rho_1$ . If  $\rho_1$  is able to reach **F** (Fig. 20a, right), then the degree-four vertex will be rigidly foldable independent of the value of  $\rho_1$  and the result will depend on the sector angles only.

In Fig. 20b, the motion of the identical linkage is driven instead by the link  $\alpha_2$  that now serves as an input crank, which drastically impacts the possible range of motion (as can be observed by comparing the trajectories of  $\alpha_4$  in Fig. 20a and Fig. 20b). This demonstrates the significance of the choice of driving angle on the range of motion: a degree-four vertex analogous to the spherical four-bar linkage in Fig. 20 would be guaranteed to fold rigidly when driven by  $\rho_1$  while it would only fold rigidly within certain limits when driven by  $\rho_2$ . This impacts the analysis of the rigid foldability of a degree-four vertex, for which it is paramount to consider the perspective of a driving angle (with the exception of certain cases that will be addressed in Section 4.1.3).

The last point to address in this analogy to spherical four-bar linkages is the reason for which a degree-four vertex is not always rigidly foldable in the full range of a given driving angle. Since it has been determined that the unfolded state  $\mathbf{U}$  is not critical for developable vertices, the only stopping condition to assess for the driving angle  $\rho_2$  is the one toward its fully folded state  $\mathbf{F}$  (right-hand side of Fig. 20b). Once the opposite dihedral angle  $\rho_4$  is at its own maximum,  $\rho_2$  cannot be further increased and the motion comes to a halt. Stated more generally, the limiting factor for the range of motion is the dihedral angle opposite of the driving angle. Furthermore, this is the only limiting factor when self-intersection is not considered as a stopping condition, and self-intersection can be neglected without loss of generality since Streinu and Whiteley [117] proved that any vertex can unfold without self-intersection, as illustrated in Fig. 21.

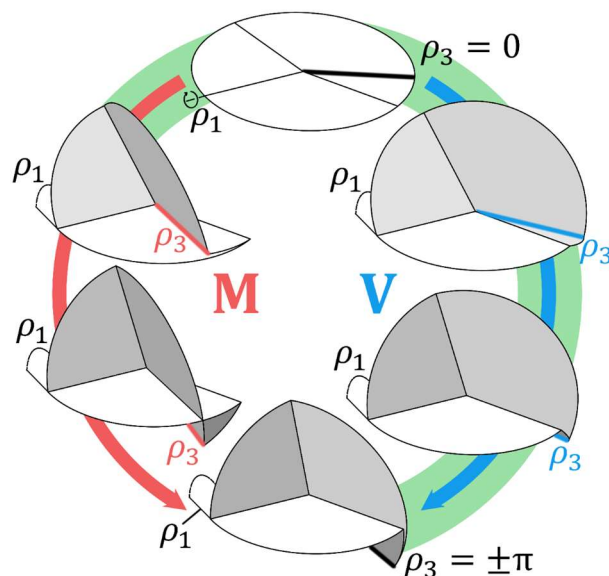


Fig. 21: A generic degree-four vertex in different states of a folding process with driving angle  $\rho_1$  and its opposite dihedral angle  $\rho_3$ . No self-intersection occurs when  $\rho_3$  has the same parity as the driving angle  $\rho_1$  (valley  $\mathbf{V}$ , right branch) and vice-versa when  $\rho_3$  has the opposite parity (mountain  $\mathbf{M}$ , left branch). The states connected by the broad arc are foldable in a physical model. When self-intersection as a stopping condition is not considered, the extreme state is reached in both RBMs when  $\rho_3 = \pm\pi$ .

Fig. 21 shows the folding motion of a degree-four vertex with driving angle  $\rho_1$  and different parities of its opposite dihedral angle  $\rho_3$ , corresponding to different RBMs. If  $\rho_1$  and  $\rho_3$  have the same parity (valley  $\mathbf{V}$ , right branch), the vertex folds “from the outside to the inside” and evokes no self-intersection. In this case, there is no difference in the stopping condition between physical paper and the mathematical model. When  $\rho_1$  and  $\rho_3$  have the opposite parity (mountain  $\mathbf{M}$ , left branch), the vertex folds “from the inside to the outside”, and self-intersection occurs. In this case, a paper model would only fold until it reaches the first state depicted in the left RBM (connected to the broad arc in Fig. 21). When paper is not modelled as an impenetrable material, no angle constraint is violated and the folding

motion can be continued until  $\rho_3 = \pm\pi$  is reached. Hence, the extreme state is reached in both RBMs once the dihedral angle opposite the driving angle is at its extreme. The stopping condition with self-intersection will be discussed in Section 4.1.3.

#### 4.1.2 Derivation of the Necessary and Sufficient Condition

Of the numerous ways to approach the problem of rigid foldability for a degree-four vertex, the following formulation is chosen: if a single dihedral angle ( $\rho_1$ ) within a degree-four vertex is prescribed, what condition applies to the sector angles  $\alpha_{1-4}$  so that the vertex folds rigidly?

The incomplete vertex on the left-hand side of Fig. 22 only consists of two sectors  $\alpha_1$  and  $\alpha_4$  connected by the crease line  $c_1$  that is driven with  $\rho_1$ . Let  $\mathbf{c}_1 = (1,0,0)$  be the unit vector on crease line  $c_1$  located on the  $x$ -axis of the coordinate system shown in Fig. 22 (left). Furthermore, let two vectors  $\mathbf{c}_2$  and  $\mathbf{c}_4$  correspond to the unit vectors of the crease lines  $c_2$  and  $c_4$ , respectively, and let  $c_2$  lie in the  $xy$ -plane. The vectors  $\mathbf{c}_2$  and  $\mathbf{c}_4$  can then be expressed as:

$$\mathbf{c}_2 = \mathbf{R}_z(\alpha_1)\mathbf{c}_1 \text{ and } \mathbf{c}_4 = \mathbf{R}_x(-\rho_1)\mathbf{R}_z(-\alpha_4)\mathbf{c}_1 \quad (6)$$

where  $\mathbf{R}_x$  and  $\mathbf{R}_z$  are the rotation matrices that rotate around the  $x$ - and the  $z$ -axis, respectively, by the rotational angle in the parentheses.

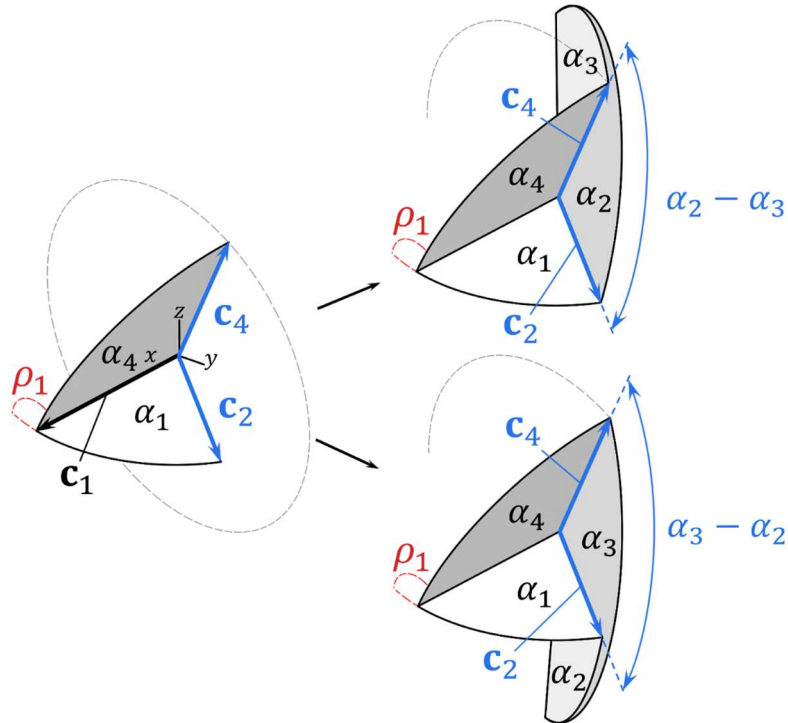


Fig. 22: Left: Incomplete vertex with only two sectors  $\alpha_1$  and  $\alpha_4$ , driving angle  $\rho_1$ , and three vectors  $\mathbf{c}_1$ ,  $\mathbf{c}_2$ , and  $\mathbf{c}_4$  that pertain to the crease lines  $c_1$ ,  $c_2$ , and  $c_4$ , respectively. Crease line  $c_1$  corresponds to the  $x$ -axis of the depicted coordinate system, and  $c_2$  lies in the  $xy$ -plane. Right: Complete degree-four vertex at two possible extreme states at which the angle between vectors  $\mathbf{c}_2$  and  $\mathbf{c}_4$  is equal to  $\alpha_2 - \alpha_3$  (top) and  $\alpha_3 - \alpha_2$  (bottom), respectively.

Following directly from the evenness of the cosine function  $\cos(\rho_1)$  for  $\rho_1 \in [-\pi, \pi]$ , the angle between  $\mathbf{c}_2$  and  $\mathbf{c}_4$ ,  $\angle(\mathbf{c}_2, \mathbf{c}_4)$ , is at its maximum in the unfolded state  $\rho_1 = 0$  and monotonically decreases whenever  $\rho_1$  deviates from zero to either side. When only two sectors are present,  $\rho_1$  can reach its fully folded state,  $\rho_1 = \pm \pi$ , and its range of motion is unrestricted (grey trajectory in Fig. 22 left).

As soon as the missing sectors  $\alpha_2$  and  $\alpha_3$  are introduced (Fig. 22 right), a closure constraint is added to the system, and the range of motion of the driving angle  $\rho_1$  is coupled to its opposite sectors  $\alpha_2$  and  $\alpha_3$ . Because the unfolded state is still guaranteed to be rigidly foldable ( $\angle(\mathbf{c}_2, \mathbf{c}_4)$  is at its maximum), the stopping condition needs to be connected to a minimum possible  $\angle(\mathbf{c}_2, \mathbf{c}_4)$ . This minimum is reached when either  $\rho_1$  reaches  $\pm \pi$ , in which case the vertex is guaranteed to be rigidly foldable, or when sectors  $\alpha_2$  and  $\alpha_3$  are coplanar, as shown in Fig. 22 (right). At this state, the degree-four vertex collapses into a spherical triangle whose sides are constituted by  $\alpha_1$ ,  $\alpha_4$ , and the cone spanned by the difference between the opposite sectors  $\alpha_2$  and  $\alpha_3$ . Hence, the condition for the rigid foldability of a degree-four vertex states that  $\angle(\mathbf{c}_2, \mathbf{c}_4)$  has to be greater than or equal to the absolute value of the difference between sectors  $\alpha_2$  and  $\alpha_3$  at any state of the folding process:

$$\angle(\mathbf{c}_2, \mathbf{c}_4) \geq |\alpha_2 - \alpha_3| \text{ (or equivalently } \angle(\mathbf{c}_2, \mathbf{c}_4) \geq |\alpha_3 - \alpha_2|). \quad (7)$$

With the magnitudes  $\|\mathbf{c}_2\| = \|\mathbf{c}_4\| = 1$ , the angle between  $\mathbf{c}_2$  and  $\mathbf{c}_4$  is

$$\angle(\mathbf{c}_2, \mathbf{c}_4) = \cos^{-1}\left(\frac{\mathbf{c}_2 \cdot \mathbf{c}_4}{\|\mathbf{c}_2\| \|\mathbf{c}_4\|}\right) = \cos^{-1}(\mathbf{c}_2 \cdot \mathbf{c}_4) = \cos^{-1}(\cos(\alpha_1)\cos(\alpha_4) - \cos(\rho_1)\sin(\alpha_1)\sin(\alpha_4)) \quad (8)$$

Hence, the necessary and sufficient condition for the rigid foldability of a degree-four vertex states that:

*A developable degree-four vertex with driving angle  $\rho_1 \in [-\pi, \pi]$  and sector angles  $\alpha_{1-4}$  that satisfy  $\sum_i \alpha_i = 2\pi$  and  $\alpha_i \in (0, \pi)$ , is rigidly foldable if and only if*

$$\cos^{-1}(\cos(\alpha_1)\cos(\alpha_4) - \cos(\rho_1)\sin(\alpha_1)\sin(\alpha_4)) \geq |\alpha_2 - \alpha_3|. \quad (9)$$

#### 4.1.3 Rigid and Flat Foldability in Degree-Four Vertices

The key to the derivation of Eq. (9) is the collapse of the degree-four vertex into a spherical triangle at its extreme state (Fig. 22 right, and equivalently Fig. 20b right). At this state, the third side of the triangle is equal to the difference between the sectors opposite of the driving angle and independent of all dihedral angles that are not driven. This reduces the complexity of the geometric relations between dihedral and sector angles, and Eq. (9) can be expressed exclusively in dependence of the driving angle.

Eq. (9) includes all possible folded states: both RBMs are discussed in Section 4.1.1, and Fig. 22 includes both possible MV assignments of the dihedral angles opposite the driving angle. In addition, Eq. (9) contains the driving angle within the even cosine function, showing that all rigidly foldable states are symmetric, which explains why both Fig. 21 and Fig. 22 only depict positive driving angles (valleys).

Before Eq. (9) is discussed in more detail, the purely sufficient condition is derived first, which states the sector angle configurations that are guaranteed to fold rigidly independent of the value of  $\rho_1$ . This condition is equivalent to a full rotation of the input crank in Fig. 20a right, which is why Eq. (9) is analyzed for the fully folded states  $\rho_1 = \pm \pi$ . The left-hand side of Eq. (9) then results in the expression:

$$\cos^{-1}(\cos(\alpha_1) \cos(\alpha_4) + \sin(\alpha_1) \sin(\alpha_4)) = \cos^{-1}(\cos(\alpha_1 - \alpha_4)) \quad (10)$$

The right-hand side of Eq. (10) requires special attention since the cosine function is not generally invertible throughout its entire range. The simplification  $\cos^{-1}(\cos(\alpha_1 - \alpha_4)) = \alpha_1 - \alpha_4$  is valid only in the range  $\alpha_4 \in [\alpha_1 - \pi, \alpha_1]$ , and with  $\alpha_1 < \pi$  and  $0 < \alpha_4$ , this range is equal to  $0 < \alpha_4 \leq \alpha_1 < \pi$ . For the opposite case where  $0 < \alpha_1 \leq \alpha_4 < \pi$ , this results in  $\cos^{-1}(\cos(\alpha_1 - \alpha_4)) = \alpha_4 - \alpha_1$ . Hence,

$$\cos^{-1}(\cos(\alpha_1 - \alpha_4)) = \begin{cases} \alpha_1 - \alpha_4 & \text{for } \alpha_4 \leq \alpha_1 \\ \alpha_4 - \alpha_1 & \text{for } \alpha_4 > \alpha_1 \end{cases} \quad (11)$$

The absolute in Eq. (9) can be replaced by setting the right-hand side to  $\alpha_2 - \alpha_3$  whenever  $\alpha_2 > \alpha_3$ , and vice-versa, which leads to four distinct cases ( $\alpha_1 = \alpha_4$  and  $\alpha_2 = \alpha_3$  will be discussed separately):

$$\text{For } \begin{cases} \alpha_4 < \alpha_1 \\ \alpha_4 > \alpha_1 \end{cases} \begin{cases} \alpha_2 > \alpha_3 \\ \alpha_2 < \alpha_3 \end{cases} \rightarrow \begin{cases} \alpha_1 + \alpha_3 \geq \alpha_2 + \alpha_4 \\ \alpha_1 + \alpha_3 \leq \alpha_2 + \alpha_4 \\ \alpha_1 + \alpha_2 \leq \alpha_3 + \alpha_4 \\ \alpha_1 + \alpha_2 \geq \alpha_3 + \alpha_4 \end{cases} \quad (12)$$

Eq. (12) presents the sufficient conditions for the rigid foldability of a degree-four vertex whose motion is driven by the dihedral angle  $\rho_1$ . If  $\alpha_1 = \alpha_4$  and  $\rho_1 = \pm \pi$ , the left-hand side of Eq. (9) becomes zero, resulting in  $\alpha_2 = \alpha_3$ , in which case the degree-four vertex is symmetric and trivially flat foldable. Conversely, if  $\alpha_2 = \alpha_3$  and  $\rho_1 = \pm \pi$ , the right-hand side of Eq. (9) becomes zero. Since the left-hand side of Eq. (9) involves the standard formulation of an angle between two vectors that by definition cannot be smaller than zero, the case  $\alpha_2 = \alpha_3$  is guaranteed to be rigidly foldable for all  $\rho_1, \alpha_1$ , and  $\alpha_4$ . This will be explained by a visual example (Fig. 23), for which a simpler and more concise form of the sufficient conditions in Eq. (12) is derived first. The distinction in Eq. (11) is equivalent to an absolute, which reduces Eq. (9) to:

$$|\alpha_1 - \alpha_4| \geq |\alpha_2 - \alpha_3| \quad (13)$$

Equation (13) is the most concise form of the sufficient condition for the rigid foldability of a degree-four vertex and is equivalent to the condition given by Murray and Larochelle [120] for spherical 4R mechanisms. Eq. (13) is independent of the driving angle, but is still only valid when the folding motion is driven by  $\rho_1$ . To indicate this, the sufficient condition can be formulated as:

*A degree-four vertex is guaranteed to fold rigidly if the absolute value of the difference between the sectors adjacent to the driving angle is greater than or equal to the absolute value of the difference between the opposite sectors.*

This result further demonstrates that if a degree-four vertex is not rigidly foldable when driven by a certain driving angle, it is guaranteed to fold rigidly when the respective opposite angle is driven. This relates back to Fig. 22, where, if the dihedral angle between sectors  $\alpha_2$  and  $\alpha_3$  were driven, the vertex would be guaranteed to fold rigidly.

Equation (13) can be explained through a more practical approach that is also beneficial for the readers interested in the actual folding of a rigidly foldable degree-four vertex. Fig. 23 (top) depicts a degree-four vertex with a driving angle  $\rho_1$  and its two adjacent sectors  $\alpha_1$  and  $\alpha_4$  of fixed size.

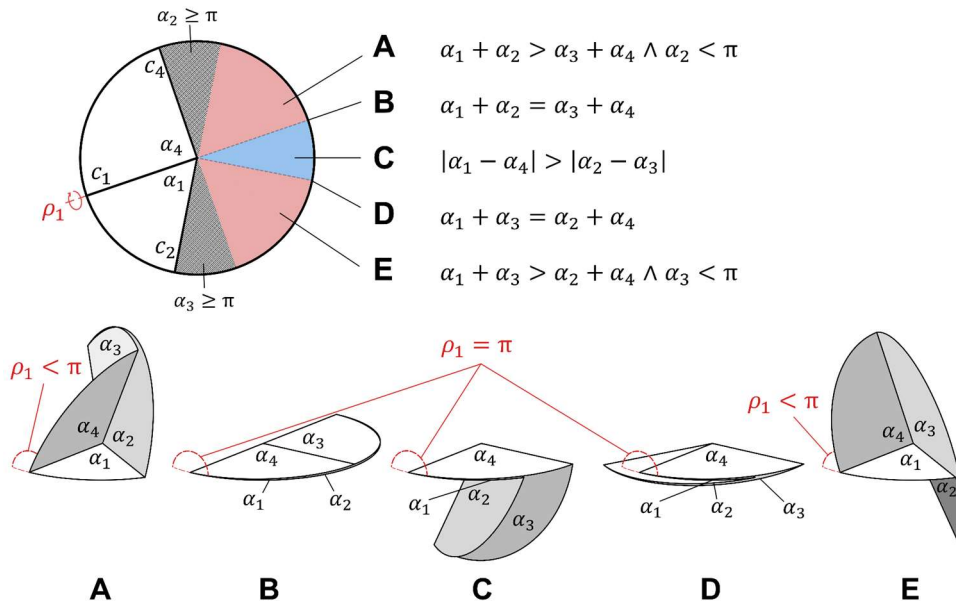


Fig. 23: Top: Degree-four vertex with driving angle  $\rho_1$ , fixed sectors  $\alpha_1$  and  $\alpha_4$ , and different cases **A-E** with their respective equations. Bottom: Extreme folded states reached at  $\rho_1 < \pi$  for **A** and **E**, and  $\rho_1 = \pi$  for **B-D**, respectively.

To illustrate that the sizes of the sector angles  $\alpha_2$  and  $\alpha_3$  are variable, crease line  $c_3$  is missing and can be placed in either of three regions **A**, **C**, and **E** or on their separating lines **B** and **D** (the grey regions are infeasible since  $\alpha_i < \pi$ ). Each case **A-E** exhibits a different behavior with respect to its extreme folded state as shown in Fig. 23 (bottom). When  $c_3$  is located in region **A**,  $\rho_1$  cannot reach  $\pi$



because its opposite angle reaches its own extreme before  $\rho_1$ . When  $c_3$  is moved from region **A** in the clockwise direction, the vertex simultaneously becomes rigidly and flat foldable on the line **B**. Moving further through region **C**, the vertex stays rigidly foldable, but sectors  $\alpha_2$  and  $\alpha_3$  rise out of the plane and form a spherical triangle together with the flat plane. When  $c_3$  is located on the line **D**, the vertex adopts its second flat foldable state. Region **E** is analogous to **A**, both of which appear in similar form in Fig. 22 (right). Fig. 23 thus perfectly corresponds to the rigidly foldable regions discovered in Section 3.

Relating back to Eq. (13), the angle that spans the rigidly foldable region in Fig. 23 equates to the (absolute) difference of the fixed sectors  $\alpha_1$  and  $\alpha_4$ . In addition, the rigidly foldable region is located adjacent to the line collinear to crease line  $c_1$  (line **B**) that separates the degree-four vertex into two half-circles, and the region **C** lies within the half-circle that contains the smaller of the sectors  $\alpha_1$  or  $\alpha_4$ . If the crease line  $c_2$  rotates in the counterclockwise direction, regions **E** and line **D** simultaneously rotate with  $c_2$ . Once  $\alpha_1$  and  $\alpha_4$  are the same size, the rigidly foldable region collapses into a single line and then increases again on the other side of the half-space. This results in a symmetric arrangement of regions with respect to the angle bisector generated by sectors  $\alpha_1$  and  $\alpha_4$ , which finally explains why  $\alpha_2 = \alpha_3$  is always rigidly foldable.

The connection between rigid and flat foldability of single degree-four vertices can now be addressed in more detail. In the binding case, the sufficiency conditions in Eq. (12) become equalities and reduce to the Kawasaki-Justin conditions for flat foldability. In this sense, flat foldability in a degree-four vertex is just a special case of rigid foldability. However, more can be revealed by considering the driving angle at its extreme,  $\rho_1^*$ , achieved by changing Eq. (9) into an equality and solving for  $\rho_1$ :

$$\rho_1^* = \pm \cos^{-1} \left( \frac{\cos(\alpha_1) \cos(\alpha_4) - \cos(\alpha_2 - \alpha_3)}{\sin(\alpha_1) \sin(\alpha_4)} \right) \quad (14)$$

In Eq. (14), the absolute of the former right-hand side has been removed because of the evenness of the cosine function. In the regions **A** and **E** as well as the lines **B** and **D**, the result for  $\rho_1^*$  is between or equal to  $\pm\pi$ , respectively. In region **C**, the result for  $\rho_1^*$  equates to  $\pm\pi$  plus an imaginary term since the argument within the arccosine in Eq. (14) lies outside of the arccosine range. When either of the Kawasaki-Justin conditions are inserted, e.g.  $\alpha_2 - \alpha_3 = \alpha_1 - \alpha_4$ , the term within the arccosine in Eq. (14) reduces to  $-1$ :

$$\frac{\cos(\alpha_1) \cos(\alpha_4) - \cos(\alpha_1 - \alpha_4)}{\sin(\alpha_1) \sin(\alpha_4)} = \frac{\cos(\alpha_1) \cos(\alpha_4) - (\cos(\alpha_1) \cos(\alpha_4) + \sin(\alpha_1) \sin(\alpha_4))}{\sin(\alpha_1) \sin(\alpha_4)} = -1 \quad (15)$$

Eq. (15) demonstrates that  $\rho_1^*$  is guaranteed to be  $\pm\pi$  and that any flat foldable configuration in the degree-four vertex is always rigidly foldable independent of the size of the sector angles, and thus independent of the choice of driving angle.

Conversely, Eq. (16) is obtained by setting the term within the arccosine in Eq. (14) equal to  $-1$ , by substituting  $\alpha_3$  with  $\alpha_3 = 2\pi - (\alpha_1 + \alpha_2 + \alpha_4)$ , and by then solving for  $\alpha_2$ :

$$\alpha_2 = \frac{1}{2}(-\alpha_1 - \alpha_4 \pm \cos^{-1}(\cos(\alpha_1 - \alpha_4))) \quad (16)$$

Together with Eq. (11), Eq. (16) leads to  $\alpha_2 = \alpha_1$  or  $\alpha_2 = \alpha_4$ , which are again flat foldable configurations. Hence, in addition to always folding rigidly, flat foldable configurations are the only ones to do so for any choice of driving angle. This connection between flat and rigid foldability can be stated for degree-four vertices as an interesting dualism: while flat foldability acts as a boundary to the region that is guaranteed to be rigidly foldable (**C** in Fig. 23), flat foldable configurations are the only ones guaranteed to always fold rigidly, independent of the choice of driving angle.

A last special property of flat foldable degree-four vertices concerns self-intersection and relates back to the stopping condition that applies when paper is modeled as an impenetrable material (Fig. 21, left branch). As can be observed in the first depicted state of the left RBM in Fig. 21, the degree-four vertex collapses into a spherical triangle similar to the ones shown in Fig. 22 but with different sides. Note that the vertex can collapse to either side, which results in a triangle with the sides  $\alpha_1$ ,  $\alpha_2$ , and  $|\alpha_3 - \alpha_4|$  or in a triangle with the sides  $|\alpha_1 - \alpha_2|$ ,  $\alpha_3$ , and  $\alpha_4$ . In both cases, the maximum dihedral angle can be determined directly by applying the same procedure described in Section 4.1.2. The resulting maximum dihedral angle is in general smaller than the one provided in Eq. (14) with the exception of flat foldable configurations for which the driving and its opposite dihedral angle reach  $\pm\pi$  at the same time. Then, both branches shown in Fig. 21 are free of self-intersection, which demonstrates that flat foldable configurations are the only ones that are guaranteed to fold without self-intersection. This special property of flat foldable degree-four vertices has also been demonstrated by Balkcom [74].

These findings conclude the analysis of the degree-4 vertex, and the next section generalizes the findings for degree- $n$  vertices. The crucial idea for this generalization is sparked by Fig. 21, Fig. 22, and Fig. 23, in which all the last rigidly foldable states have one thing in common: they all exhibit a single spherical triangle.

## 4.2 Degree- $n$ Vertex

The content of Section 4.2 has been accepted for publication as: [121] Zimmermann, L., Shea, K., and Stanković, T., “Conditions for Rigid and Flat Foldability of Degree- $n$  Vertices in Origami,” *Journal of Mechanisms and Robotics*, 12(1), 2020.

This section first introduces the Principle of Three Units (PTU) that derives from the appearance of the spherical triangles, after which the PTU is related to known origami phenomena in order to embed it within current origami research. Then, a kinematic model and the implications for single vertices and crease patterns are presented.

### 4.2.1 The Principle of Three Units

The PTU predicates on the fact that every single degree- $n$  vertex requires  $n - 3$  inputs and 3 outputs to fold in a kinematically determinate manner [74]. The inputs are the driving angles that need to be prescribed, and the outputs are the three remaining dihedral angles, here called the *unknown dihedral angles*, whose behavior is determined by the driving angles and the size of the sector angles around the vertex. As an example, Fig. 24a depicts a degree-8 vertex with five driving angles  $\rho_{1-4}$  and  $\rho_6$  (in black) assigned to an arbitrary set of crease lines together with the corresponding unknown dihedral angles  $\rho_5$ ,  $\rho_7$ , and  $\rho_8$  (in grey).

Independent of both the chosen set of driving angles and the vertex degree (for  $n \geq 4$ ), virtually cutting the vertex at the locations of the unknown dihedral angles reveals three parts [74], here called *units*  $u_1$ ,  $u_2$ , and  $u_3$  (Fig. 24b). This virtual cut eliminates the unknown dihedral angles, and what remains within the units are the sector and driving angles determined by the user, which is why the entire kinematic behavior of each individual unit can be determined analytically. Related works [74, 95] then proceed to calculate the three corresponding rotation matrices to solve for the unknown dihedral angles by satisfying the closure constraint of the continuous surface around the vertex.

The slight shift of the PTU is the realization that the three units can be capitalized on in a different way. As shown in Fig. 24b and Fig. 24c, each unit  $u_i$  spans an angle between its first and its last crease line called a *unit angle*  $U_i$  that can be expressed analytically by alternately multiplying the rotation matrices of sector and driving angles and calculating the angle between the first and the last vector obtained. If a unit consists of a single sector, the corresponding unit angle is constant in size,  $U_3 = \alpha_8$  in Fig. 24b and Fig. 24c, while the unit angles that pertain to units with multiple sectors change their size depending on the state of their driving angles, such as  $U_1(0), U_2(0)$  at the initial state and  $U_1(t_1), U_2(t_1)$  at an arbitrary state  $t_1$  in Fig. 24b and Fig. 24c. These unit angles are independent of each other in a

virtually cut scenario (Fig. 24c), but once the vertex is reunited they are again subject to the original closure constraint of the surface. However, this closure constraint now applies to a single spherical triangle, here called a *vertex triangle* (Fig. 24d). The PTU states that the kinematics of any rigidly foldable degree- $n$  vertex are determined by this single spherical vertex triangle.

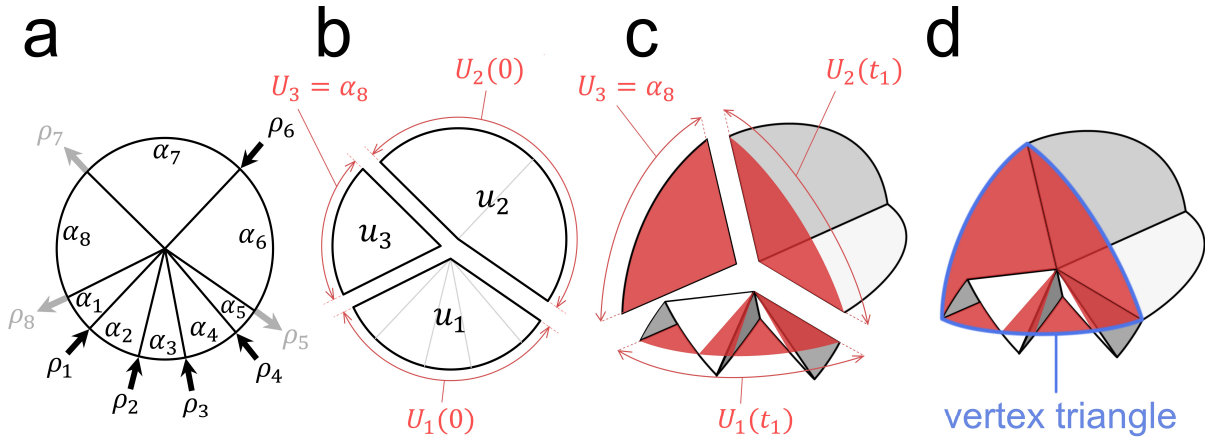


Fig. 24: (a) Single vertex with sector angles  $\alpha_{1-8}$ , driving angles  $\rho_{1-4}$  and  $\rho_6$ , and unknown dihedral angles  $\rho_5$ ,  $\rho_7$ , and  $\rho_8$ . (b) The vertex from a) is virtually cut at the crease lines with unknown dihedral angles to reveal three unit angles  $U_1(0)$ ,  $U_2(0)$ , and  $U_3 = \alpha_8$  in the unfolded state. (c) The virtually cut vertex at an arbitrary folded state  $t_1$  with unit angles  $U_1(t_1)$ ,  $U_2(t_1)$ , and  $U_3 = \alpha_8$ . (d) The reunited vertex and its vertex triangle.

#### 4.2.2 The Principle of Three Units in Relation to Origami Phenomena

The representation of a vertex as a single spherical triangle provides some intuitive explanations for a number of origami phenomena, some of which will be recounted here in order to place the PTU in the context of current origami research.

First, the PTU offers a reason as to why degree-four vertices are the smallest degree vertices to fold non-trivially. Miura proved that any single vertex up to degree-3 is not able to fold unless two of its crease lines are collinear [85]. In the PTU, a degree-3 vertex corresponds to a spherical triangle with three stiff sides, similar to the units shown in Fig. 24b. It has long been proven that three given sides determine the shape of a triangle unambiguously, which is why no motion is possible with three fixed sides. However, if one of the sector angles is equal to  $\pi$ , trivial motion is possible along the longest triangle side, analogous to the case of two collinear crease lines as presented by Miura.

Second, the PTU illustrates why every single vertex has two RBMs when all necessary driving angles are prescribed. Fig. 25a shows the two RBMs of a degree-6 vertex that consists of three symmetric units. The shape of a triangle is determined when three sides are given, but the set of constraints present in origami allows for two possibilities: when the side  $\overline{AB}$  is virtually fixed in space, the two remaining sides can be joined either in point  $C_1$  or in  $C_2$ , which leads to the two vertex triangles

$\Delta ABC_1$  and  $\Delta ABC_2$ , respectively. These triangles themselves do not differ in shape because the former is just a rotated version of the latter, and vice versa. However, they do differ in orientation with respect to the folded surface, and the underlying vertex can assume one of two states. The kinematic discrepancy between the states lies in the sign of the internal angles of the vertex triangle with respect to the reference frame, and the two distinct cases need to be integrated into the kinematic model presented in the next section to address RQ2.

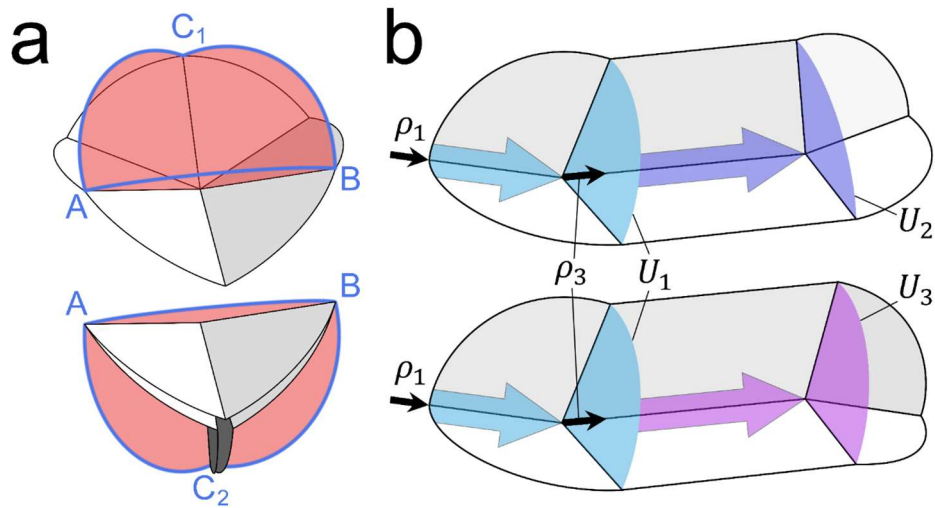


Fig. 25: a) Symmetrical degree-6 vertex with two RBMs that result from the two possible vertex triangles  $\Delta ABC_1$  and  $\Delta ABC_2$ . b) Crease patterns consisting of two connected vertices each. The driving angle  $\rho_1$ , the input  $U_1$  to the first vertex, and the outgoing dihedral angle  $\rho_3$  are equivalent in both cases, but the inputs  $U_2$  and  $U_3$  to the second vertex differ because of the respective sector angle configuration.

Third, the PTU consolidates seemingly conflicting discoveries made by a number of works with respect to the connection between origami and complex networks. Any foldable single vertex is a node that takes  $n - 3$  inputs and transforms these into three outputs [74]. If any two of these nodes are connected, the output of the first is an input to the second, which implies that the dihedral angles are the information transmitted between the nodes. This phenomenon can be observed in open chain linkages such as slender origami [122] and thin origami strings [123], where the input of a dihedral angle at the beginning of a string is propagated throughout a series of single vertices toward its end. The same flow of dihedral angles has been capitalized on in closed chains by Lang et al. [45] to efficiently design quadrilateral creased paper. Huffman makes a similar discovery in his seminal work [60], but presents an analogy between origami and electrical networks in which he postulates that the electrical current is equivalent to the curvature components of the surface. Kirchhoff's first law relies upon the fact that the net charge of an electrical node is conserved, which means that the outward flow is equal to the inward flow. However, symbolic equations for origami dihedral angles [89] demonstrate that the net sum of dihedral angles at a vertex is neither zero nor constant. This fact seems to contradict the above

observation that the dihedral angles are the sole transmitted information between the nodes. If rigid origami does behave like an electrical network, the only known candidate for the conserved entity is the surface area. The sum of the sector angles around a vertex is by definition constant, and without deformation in the sectors, the surface area remains unchanged throughout any rigid motion. This is true for all degree- $n$  vertices, also when the initial state of the surface is not flat. Indeed, this is also what the PTU implies: by representing the surface of a vertex with a vertex triangle, the kinematics are determined by the unit angles  $U$  that represent spanned surfaces resulting from a combination of sector and dihedral angles.

Fig. 25b top and bottom show crease patterns consisting of two connected vertices each, whose motion is driven by the driving angle  $\rho_1$ . Together with its adjacent sector angles,  $\rho_1$  spans the unit angle  $U_1$  that represents the input to both crease patterns. Since the flow between the connected vertices is directed, the dihedral angle  $\rho_3$  is independent of the geometry of the second vertex and thus equal in both cases. However, the sector angles of the second vertex in both crease patterns vary slightly, which is why the inputs  $U_2$  and  $U_3$  to the second vertex differ. Consequently, both connected vertices exhibit different folding motions. Hence, the PTU strengthens Huffman's view that the transmitted information are curvature components, and it contributes by determining that these components are the unit angles  $U$ . The transmission of the unit angles will be explained in more detail in Section 4.2.5.

In addition to explaining these phenomena, the PTU offers a kinematic model that efficiently determines the unknown dihedral angles of degree- $n$  vertices, which is developed next.

#### 4.2.3 Kinematic Model of the Principle of Three Units

The goal of this section is to model the kinematics of degree- $n$  vertices, and thus to determine the three unknown dihedral angles present in a vertex. Since the three unit angles  $U$  can be expressed by known sector and driving angles, the internal angles of the vertex triangle can be found through the spherical law of cosines. Unfortunately, the unknown dihedral angles are not solely dependent on the internal angles of the vertex triangle. Except for units consisting of single sectors, any unknown dihedral angle first consists of a rotation from the last sector of the previous unit to the vertex triangle, then a rotation by the internal angle, and finally a rotation from the vertex triangle to the first sector of the following unit. Accordingly, these angles will be called the *end angles*  $\delta$ , the *internal angles*  $\theta$ , and the *start angles*  $\beta$ , respectively (Fig. 26). These angles are first calculated for each unit individually and then assembled in the end of the section to express the unknown dihedral angles.

The frame of reference is important for the following calculations, so it is assumed that at least one sector lies in the  $xy$ -plane and that the default viewpoint lies above this plane on the positive  $z$ -axis. Then, the sector angles are numbered in counter-clockwise order, and because each unit is modeled individually, all sector as well as driving angles are denoted with a subscript  $j = 1,2,3$  to allocate them unambiguously to a unit  $u_j$ . Using  $\rho_j = \rho_j(t)$  leads to the following definition of a unit:

*Definition: A unit  $u_j$  consists of an open chain of  $m$  sectors  $\alpha_{u_j} = (\alpha_{j,1}, \dots, \alpha_{j,m})$ , in which each dihedral angle  $\rho_{u_j} = (\rho_{j,1}, \dots, \rho_{j,m-1})$  between two consecutive sectors is a driving angle. If neither of the crease lines on either side of a sector are driven, this sector represents a unit on its own.*

Fig. 26a schematically shows a generic unit that, for illustrative purposes, is equivalent to the folded state of unit  $u_1$  in Fig. 24c. Since single vertices belong to the class of spherical mechanisms [100, 112, 117], the following calculations are performed using spherical geometry. In addition, all the point coordinates calculated in the kinematic model are assumed to lie on a sphere of unit radius to omit normalization. The first sector  $\alpha_{j,1}$  of a unit is defined to lie in the  $xy$ -plane as illustrated in Fig. 26a on the left-hand side by the Cartesian coordinate system whose origin lies at  $(0,0,0)$ . Then, the first point  $P_{j,0}$  is located at  $P_{j,0} = (1,0,0)$ .

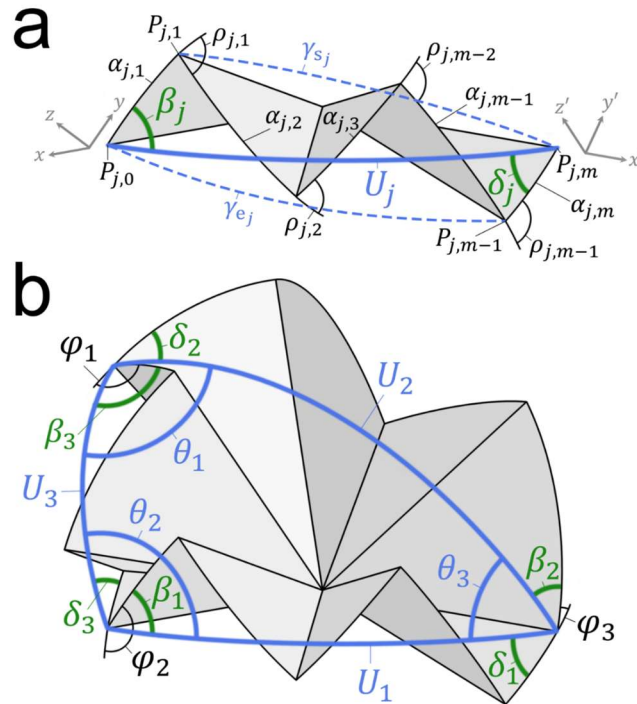


Fig. 26: a) A unit  $u_j$  with sector angles  $\alpha_{j,1}, \dots, \alpha_{j,m}$ , driving angles  $\rho_{j,1}, \dots, \rho_{j,m-1}$ , unit angle  $U_j$ , start angle  $\beta_j$ , and end angle  $\delta_j$ . Spanned angles  $\gamma_{s_j}$  and  $\gamma_{e_j}$  are used to calculate start and end angle, respectively. b) The same unit (now  $u_1$ ) constitutes an arbitrary single vertex together with two other generic units  $u_2$  and  $u_3$ , where the vertex triangle with sides  $U_1, U_2$ , and  $U_3$  is shown in thick lines. The combination of start angles  $\beta_{1-3}$ , end angles  $\delta_{1-3}$ , and internal angles  $\theta_{1-3}$  leads to the unknown dihedral angles  $\phi_{1-3}$ .

The unit angle  $U_j$  lies between the first and the last crease line of a unit (Fig. 26a), and the location of the last point  $P_{j,m}$  is calculated with the following expression:

$$P_{j,m} = \left( \prod_{i=1}^{m-1} \mathbf{R}_z(\alpha_{j,i}) \mathbf{R}_x(\rho_{j,i}) \right) \mathbf{R}_z(\alpha_{j,m}) P_{j,0} \quad (17)$$

where  $\mathbf{R}_x$  and  $\mathbf{R}_z$  are the standard rotation matrices that rotate around the  $x$ -axis and the  $z$ -axis, respectively, by the rotational angle within the parentheses. Then, the unit angle  $U_j$  is expressed by the standard formulation of an angle between two vectors:

$$U_j = \cos^{-1}(P_{j,0} \cdot P_{j,m}) \quad (18)$$

If a unit consists of a single sector, the unit angle  $U_j$  is equal to the sector angle  $\alpha_{j,1}$  and the calculations in Eqs. (17) and (18) can be omitted.

Fig. 26b depicts the same unit as Fig. 26a (now  $u_1$ ) together with two other arbitrary units that constitute a single vertex. Since the unit angles  $U_1$ ,  $U_2$ , and  $U_3$  are equivalent to the sides of the vertex triangle, the internal angles of the vertex triangle are simply calculated using the spherical law of cosines:

$$\begin{aligned} \theta_1 &= \cos^{-1} \frac{\cos U_1 - \cos U_2 \cos U_3}{\sin U_2 \sin U_3} \\ \theta_2 &= \cos^{-1} \frac{\cos U_2 - \cos U_1 \cos U_3}{\sin U_1 \sin U_3} \\ \theta_3 &= \cos^{-1} \frac{\cos U_3 - \cos U_1 \cos U_2}{\sin U_1 \sin U_2} \end{aligned} \quad (19)$$

Since these internal angles are by definition positive, their signs will be adapted to the respective RBM once the unknown dihedral angles are assembled at the end of this section.

The start angle  $\beta_j$  is an internal angle of the spherical triangle whose sides are constituted by  $\alpha_{j,1}$ ,  $U_j$ , and  $\gamma_{s_j}$  (Fig. 26a). Of these, the only missing side is  $\gamma_{s_j}$ , which can be expressed analogous to Eq. (18) as  $\gamma_{s_j} = \cos^{-1}(P_{j,1} \cdot P_{j,m})$  with the point  $P_{j,1} = \mathbf{R}_z(\alpha_{j,1})P_{j,0}$ . Before the magnitude of  $\beta_j$  is calculated by using the spherical law of cosines, a sign for the start angle needs to be introduced since the direction of rotation matters once the vertex is assembled. This rotation is applied in the counter-clockwise direction when the first sector of a unit lies within the vertex triangle and in the clockwise direction when the first sector lies on the outside ( $\beta_1$  and  $\beta_3$ , and  $\beta_2$  in Fig. 26b, respectively). Hence,  $\beta_j$  is defined to be positive when the point  $P_{j,m}$  in Fig. 26a lies beneath the  $xy$ -plane and negative in the opposite case. In mathematical terms, this results in the negative sign of the  $z$ -coordinate of  $P_{j,m}$ ,  $-\text{sgn}(P_{j,m}^z)$ , which leads to the following expression for the start angle  $\beta_j$ :



$$\beta_j = -\text{sgn}(P_{j,m}^z) \cos^{-1} \frac{\cos \gamma_{s_j} - \cos \alpha_{j,1} \cos U_j}{\sin \alpha_{j,1} \sin U_j} \quad (20)$$

While the magnitude of the end angle  $\delta_j$  can be equivalently calculated by applying the spherical law of cosines to the spherical triangle with the sides  $\alpha_{j,m}$ ,  $U_j$ , and  $\gamma_{e_j}$  (Fig. 26a), determining its sign is not as straightforward as for the start angle. The difficulty is that the location of  $P_{j,0}$  with respect to the last sector  $\alpha_{j,m}$  cannot be determined in the current coordinate system  $(x, y, z)$  without a solving procedure that involves inverse matrix calculations. To avoid these, an auxiliary coordinate system  $(x', y', z')$  is introduced, whose  $x'y'$ -plane is coplanar to the sector  $\alpha_{j,m}$  (Fig. 26a, right-hand side). Then, the above procedure is repeated with negative sector and dihedral angles in reverse order to achieve an analogous sign of the  $z'$ -coordinate of  $P_{j,0}$ . In practice, the following calculations for the end angle are decoupled from the previous ones, and the same rotation matrices  $\mathbf{R}_{x'} = \mathbf{R}_x$  and  $\mathbf{R}_{z'} = \mathbf{R}_z$  can be applied. Using the same notation for the points as in Fig. 26a, the point  $P_{j,m}$  is then located at  $P'_{j,m} = (1,0,0)$ , and the point  $P'_{j,0}$  can be expressed as:

$$P'_{j,0} = \mathbf{R}_z(-\alpha_m) \left( \prod_{i=1}^{m-1} \mathbf{R}_z(-\rho_{m-i}) \mathbf{R}_x(-\alpha_{m-i}) \right) P'_{j,m} \quad (21)$$

With the point  $P'_{j,m-1} = \mathbf{R}_z(-\alpha_m)$  and with the spanned angle  $\gamma_{e_j} = \cos^{-1}(P'_{j,m-1} \cdot P'_{j,0})$ , this leads to the following expression for the end angle  $\delta_j$ :

$$\delta_j = -\text{sgn}(P'_{j,0}) \cos^{-1} \frac{\cos \gamma_{e_j} - \cos \alpha_{j,m} \cos U_j}{\sin \alpha_{j,m} \sin U_j} \quad (22)$$

If a unit consists of a single sector, both its start and end angle are zero since the sector already represents the side of the vertex triangle, and the calculations in Eqs. (20) - (22) can be omitted.

Finally, to assemble the unknown dihedral angles  $\boldsymbol{\varphi}$  that emerge at the corners of the vertex triangle (Fig. 26b), the kinematic model needs to account for the RBMs of the vertex denoted as  $M_\uparrow$  and  $M_\downarrow$ . By associating  $M_\uparrow$  with the “upward” and  $M_\downarrow$  with the “downward” motion equivalent to Fig. 25a, the rotations of the internal angles are applied in the counter-clockwise and the clockwise direction, respectively. The unknown dihedral angles  $\boldsymbol{\varphi}$  can then be calculated with:

$$\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \varphi_3) = \begin{cases} (\beta_3, \beta_1, \beta_2) + (\pi - \theta_1, \pi - \theta_2, \pi - \theta_3) + (\delta_2, \delta_3, \delta_1), & \text{if } M_\uparrow \\ (\beta_3, \beta_1, \beta_2) + (\theta_1 - \pi, \theta_2 - \pi, \theta_3 - \pi) + (\delta_2, \delta_3, \delta_1), & \text{if } M_\downarrow \end{cases} \quad (23)$$

#### 4.2.4 Implications for Single Vertices

Having completed the kinematic model, this section first focuses on the implications and advantages of the PTU with respect to single vertices. To describe a vertex in a concise form, the following notation is introduced, defining a vertex  $v$  unambiguously by its mode and units

$$v = v(\text{mode}, u_1, u_2, u_3) = v\left(M, (\alpha_{u_1}, \rho_{u_1}), (\alpha_{u_2}, \rho_{u_2}), (\alpha_{u_3}, \rho_{u_3})\right) \quad (24)$$

where  $M$  is the RBM and  $\alpha_{u_j}$  and  $\rho_{u_j}$  are the sector and driving angle vectors corresponding to unit  $u_j$ . Note that not all units contain a driving angle, in which case the driving angles  $\rho_{u_j}$  are omitted in Eq. (24) and the sector angles  $\alpha_{u_j}$  contain only a single sector angle,  $\alpha_{u_j} = \alpha_{j,1}$ . For simplified readability, in this notation the sector angles are given in degrees instead of radians. In addition,  $U_j = U_j(t)$  is used to denote a general unit angle, and the value for the argument  $t$  is provided only when specific folding states are addressed, such as for the initial state  $U_j(0)$  or the fully folded state  $U_j(\pi)$ .

The following subsections first cover computational aspects of the kinematic model, after which the intrinsic conditions for the rigid and flat foldability of degree- $n$  vertices are presented.

##### 4.2.4.1 Computation

The kinematic model of the PTU involves only forward matrix multiplications and the law of spherical cosines and thus enables the analytical calculation of the unknown dihedral angles  $\phi$  in real time. To demonstrate the benefits of the kinematic model by an example, Fig. 27 depicts three vertices with increasing degree  $n = 12, 36,$  and  $360$  from top to bottom, where the folded states go from the initial state on the left to the fully folded state on the right.

All of these vertices are shown in RBM  $M_{\uparrow}$  and are rotationally symmetric with three identical units. Each unit contains  $\frac{n}{3}$  sector angles  $\alpha_{\text{sym}} = \alpha_{u_j} = \left(\frac{360}{n}, \dots, \frac{360}{n}\right)^T$  and  $\frac{n-3}{3}$  linear driving angles with alternating signs,  $\rho_{\text{sym}} = \rho_{u_j} = (t, -t, \dots, -t, t)^T$ . Following Eq. (24), the units are  $u_{\text{sym}} = (\alpha_{\text{sym}}, \rho_{\text{sym}})$  and the vertices in Fig. 27 can be denoted with  $v(M_{\uparrow}, u_{\text{sym}}, u_{\text{sym}}, u_{\text{sym}})$ . The folding motion of these vertices shows that with increasing  $n$  the shape of the surface around the vertex progressively converges toward the shape of the underlying vertex triangle: while the degree-12 vertex is still relatively coarse, the shapes of the surface and the vertex triangle are almost identical for  $n = 360$ . To the best knowledge of the author, this degree-360 vertex is the highest degree vertex presented in origami literature, and while this is not the upper possible limit of the PTU, an even higher degree does not seem to provide any practical advantage, at least for rigid origami.

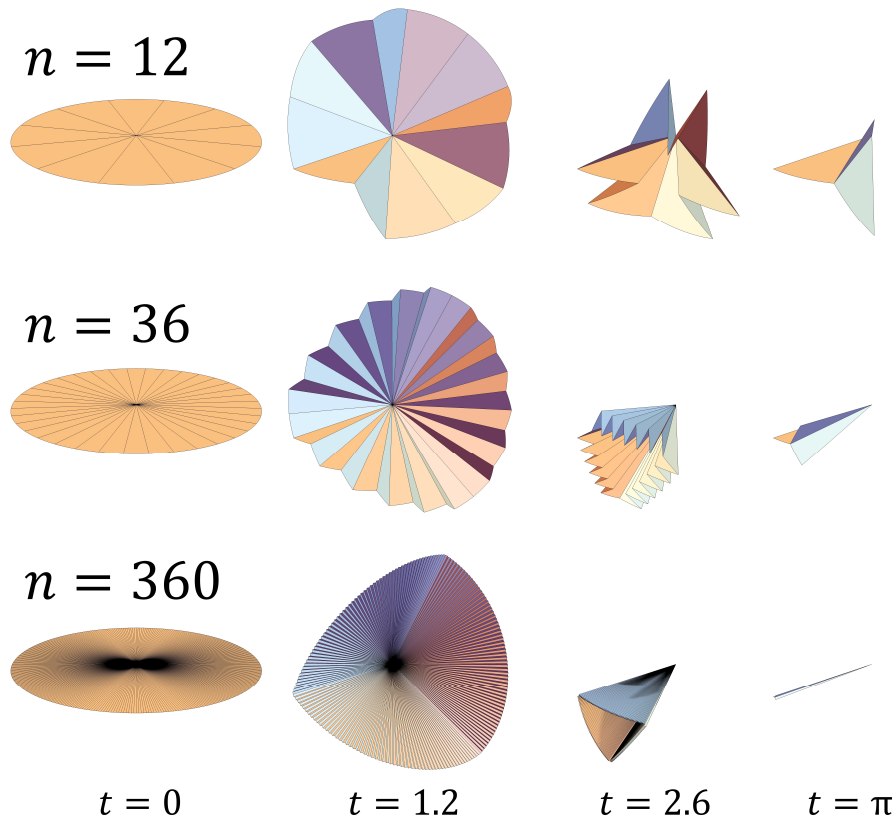


Fig. 27: Symmetric vertices of degree  $n = 12, 36, 360$  from top to bottom that fold from left to right with  $t = 0, 1.2, 2.6, \pi$ , respectively

In a more mathematical sense, Fig. 27 further implies that a symmetric vertex with  $n \rightarrow \infty$  converges into a single line in its fully folded state, which demonstrates the versatility of origami mechanisms that can theoretically fold from 2D objects into 3D shapes and back into 1D primitives.

#### 4.2.4.2 Rigid Foldability

While the kinematic model of the PTU enables the calculation of the unknown dihedral angles, the boundaries to which these angles are well defined have to be determined in order to answer RQ1. Again, the PTU provides an intuitive approach to determining these boundaries. The motion of a vertex is subject to a closure constraint that dictates the continuity of the surface around the vertex. Although this surface is virtually replaced by the vertex triangle in the PTU, the original closure constraint remains intact because the vertex triangle is just a kinematic representation of the original surface. Thus, a vertex can only fold rigidly if its unit angles  $U_j$  constitute a vertex triangle, or in other words, if such a vertex triangle exists.

The condition for the existence of a triangle is a fundamental notion in geometry and has long been proven for planar as well as spherical triangles. The condition is termed the triangle inequality [124] that states that the longest side of a triangle has to be smaller than or equal to the sum of the two

shorter sides. Fig. 28 depicts a state in which the triangle inequality is not satisfied because the sum of the two shorter sides  $U_1$  and  $U_3$  is smaller than the longest side  $U_2$ . Consequently, these sides cannot be connected to a triangle as illustrated in Fig. 28 using dashed lines, which results in a state that is not rigidly foldable.

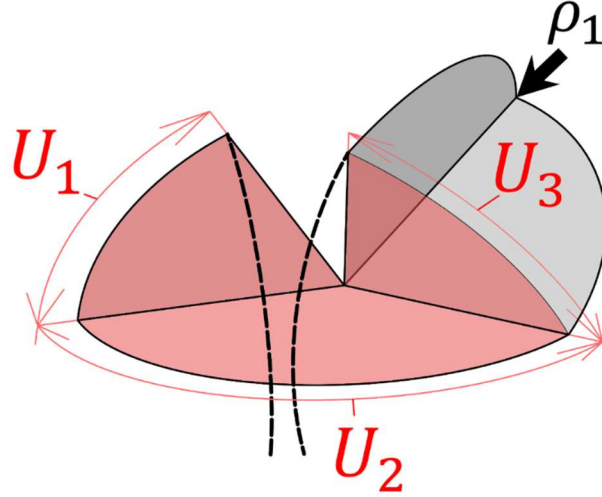


Fig. 28: Non-rigidly foldable state of the vertex  $v(M_1, (60), (120), ((90,90)^T, (t)))$ . As illustrated by the dashed lines, the unit angles  $U_{1-3}$  cannot be joined to a vertex triangle because  $U_2 > U_1 + U_3$ .

At the depicted state, the driving angle  $\rho_1$  has surpassed its limits and the unit angle  $U_3$  is too small to satisfy the triangle inequality. Hence, with the set  $S_R = \{U_1, U_2, U_3\}$ , and with the longest side  $U_{\max} = \max(S_R)$ , the medium side  $U_{\text{med}} = \text{median}(S_R)$ , as well as the shortest side  $U_{\min} = \min(S_R)$ , the condition for the rigid foldability of any single degree- $n$  vertex can be expressed as:

$$U_{\max} \leq U_{\text{med}} + U_{\min} \quad (25)$$

Note that the assignment of  $U_{\max}$  can switch during the folding process because the size of driven unit angles changes over time. As an example, folding the vertex  $v(M_1, (60), (120), ((90,90)^T, (t)))$  in Fig. 28 results in a step-wise process where  $U_{\max} = U_3$  initially represents the biggest unit angle that is equal to the sum of the other unit angles,  $U_3(0) = U_1(0) + U_2(0)$ . Then,  $U_3$  monotonically decreases but remains  $U_{\max}$  in the range  $t \in [0, \frac{\pi}{3})$ . For  $t > \frac{\pi}{3}$ ,  $U_3$  becomes smaller than  $U_2$  and the assignment of  $U_{\max}$  in Eq. (25) changes. Now, the condition reads  $U_2 \leq U_1 + U_3$ , which translates to  $\frac{2\pi}{3} \leq \frac{\pi}{3} + \cos^{-1}(-\cos(t))$  and finally to  $t \leq \frac{2\pi}{3}$ . Hence, the degree-4 vertex in Fig. 28 folds rigidly until  $t = \frac{2\pi}{3}$ . The same simple procedure can be applied to vertices with  $n > 4$  where the only increase in complexity is the computation of the unit angles  $U_j$  that can be calculated straightforward using Eq. (18).

#### 4.2.4.3 Flat Foldability

While rigid foldability corresponds to the entire folding motion, flat foldability only corresponds to the fully folded state in which all dihedral angles are required to be equal to zero or  $\pm\pi$ . Since the unknown dihedral angles  $\varphi$  in Eq. (23) result from the sum of start, internal, and end angles, the combination of these angles needs to be equal to zero or  $\pm\pi$ . By requiring that all driving angles go to  $\pm\pi$ , each unit itself folds flat, in which case the start and end angles are zero because the unit aligns with the plane of the unit angle  $U_j$ . As a consequence, the only remaining factors are the internal angles  $\theta$  that consequently have to be zero or  $\pm\pi$  themselves. Hence, in a vertex that folds flat, the vertex triangle also folds flat. Disregarding the unfolded state, a vertex triangle can only fold flat if the sum of its two shorter sides is equal to the longest side, equivalent to the triangle inequality that is transformed into an equality. Thus, with the set  $S_F = \{U_1(\pi), U_2(\pi), U_3(\pi)\}$  that contains the fully folded unit angles ( $t = \pi$ ), and with the maximum, the medium, and the smallest elements  $U(\pi)_{\max} = \max(S_F)$ ,  $U(\pi)_{\text{med}} = \text{median}(S_F)$ , and  $U(\pi)_{\min} = \min(S_F)$ , respectively, the condition for the flat foldability of a degree- $n$  vertex can be expressed as:

$$U(\pi)_{\max} = U(\pi)_{\text{med}} + U(\pi)_{\min} \quad (26)$$

The condition in Eq. (26) is readily transformed into a form that solely contains sector angles since the fully folded unit angles  $U_j(\pi)$  can be calculated by hand. Every driving angle is fully folded, which means that consecutive sector angles within a unit align in opposite directions, and a fully folded unit angle thus results in  $U_j(\pi) = |\alpha_{j,1} - \alpha_{j,2} + \alpha_{j,3} - \dots - \alpha_{j,m}|$ . Note that the unit angles are by definition positive, which is ensured here by the absolute. To illustrate the applicability of Eq. (26), the example of a degree-7 vertex  $v(M_7, ((80,40,40)^T, (t, -t)^T), ((90,40,50)^T, (t, -t)^T), (20))$  is considered, for which the fully folded unit angles result in  $U_1(\pi) = |80 - 40 + 40| = 80$ ,  $U_2(\pi) = |90 - 40 + 50| = 100$ , and  $U_3(\pi) = 20$ . Consequently, the vertex corresponds to the case  $U_2(\pi) = U_1(\pi) + U_3(\pi)$  and folds flat. Fig. 29 demonstrates that this is indeed the case and that the folding motion is non-trivial, i.e. that no dihedral angle remains zero during the folding motion.

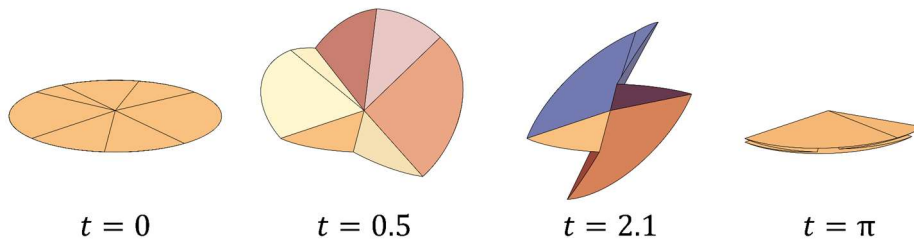


Fig. 29: Flat foldable degree-7 vertex  $v(M_7, ((80,40,40)^T, (t, -t)^T), ((90,40,50)^T, (t, -t)^T), (20))$  folding from left to right illustrated at the states  $t = 0, 0.5, 2.1, \pi$

A special situation arises when all fully folded unit angles  $U_j(\pi)$  are equal to zero, in which case the vertex triangle degenerates into a single point. Then, new DOF are introduced in the final state and the folding motion needs to be continued by a new set of driving angles to fold the vertex completely flat. Examples for this special case are symmetric vertices as depicted in Fig. 27.

#### 4.2.5 Implications for Crease Patterns

Since the PTU is able to model the kinematics of any single degree- $n$  vertex and further provides conditions for the rigid and flat foldability, the question arises whether the PTU can also assess the global rigid foldability of an origami. To answer this question, the PTU is analyzed here with respect to crease patterns. The next subsections first demonstrate the implications of the PTU on the generation of crease patterns, present the subset of crease patterns whose global rigid foldability can be assessed, and then show the limitations of the PTU by explaining the cases whose kinematics cannot be globally modeled. In doing so, the following subsections also present findings that correspond to RQ3.

##### 4.2.5.1 Crease Pattern Generation

As mentioned in Section 4.2.2, the information transmitted between vertices are the unit angles  $U$ . While the unit angles are quantitatively assessed with respect to single vertices in previous sections, here the focus lies on their qualitative propagation throughout a crease pattern.

The PTU predicated on the fact that a single vertex of degree  $n$  transforms  $n - 3$  inputs into three outputs. These inputs and outputs result in a directed flow and can be modeled in a planar directed graph  $G = (V, E)$  where  $V$  is the set of vertices  $v_i$  and  $E$  the set of directed edges  $e_{i,j}$  going from  $v_i$  to  $v_j$  such that  $i \neq j$ . For each vertex, the respective driven crease lines can then be represented as incoming edges and the crease lines with unknown dihedral angles as outgoing edges [74].

In the PTU, the kinematics of single vertices are modeled individually, even when these vertices are embedded within a crease pattern. For an internal vertex  $v_i$  of degree  $n_i$ , its indegree  $\deg^-(v_i) = n_i - 3$  linearly scales with  $n_i$  and is unrestricted as long as  $n_i \geq 4$ . However, its outdegree  $\deg^+(v_i)$  must always satisfy the condition  $\deg^+(v_i) = 3$ . In other words, any number of incoming edges can be added to a vertex without compromising its kinematic determinacy as long as the vertex has three outgoing edges. Hence, if all vertices within a crease pattern have three outgoing edges, the entire crease pattern itself is kinematically determinate. With respect to RQ3, the PTU thus provides a simple rule to generate kinematically determinate crease patterns by requiring that each vertex within the crease pattern has three outgoing edges.

Fig. 30 illustrates an example of an arbitrary origami crease pattern generated by the above rule and shows the assignment of the units  $u_{1_i}$ ,  $u_{2_i}$ , and  $u_{3_i}$  for each individual vertex  $v_i$ , which can eventually be used to assess the kinematic behavior of the entire crease pattern.

Fig. 30a starts with a single degree-4 vertex  $v_2$  that is driven by the edge  $e_{1,2}$  to which an arbitrary driving angle  $\rho_d$  is assigned in order to fold the crease pattern whose boundary is illustrated by dotted lines. The first unit of  $v_2$ ,  $u_{1,2}$ , contains the sectors adjacent to the edge  $e_{1,2}$  and is driven by  $\rho_d$ , whereas the remaining units  $u_{2,2}$  and  $u_{3,2}$  are constant. This information allows for both the expression of the unit angles  $U_{1,2}$ ,  $U_{2,2}$ , and  $U_{3,2}$  (not illustrated in Fig. 30) with Eq. (18) and the design of the sector angles around  $v_2$  in a way such that  $v_2$  folds rigidly for the given  $\rho_d$ .

As a next step, there is a choice of which vertices to extend with three outgoing edges, which is performed in Fig. 30b where the vertices  $v_3$  and  $v_5$  are extended. While the units of  $v_3$  and  $v_5$  are assigned analogously to the ones of  $v_2$ , they are driven by edges incident to  $v_2$  whose unknown dihedral angles can be expressed using the kinematic model presented in Section 4.2.3. This completes the definition of the respective incoming units  $u_{1,3}$  and  $u_{1,5}$  and thus determines the information transmitted between  $v_2$  and the vertices  $v_3$  and  $v_5$ .

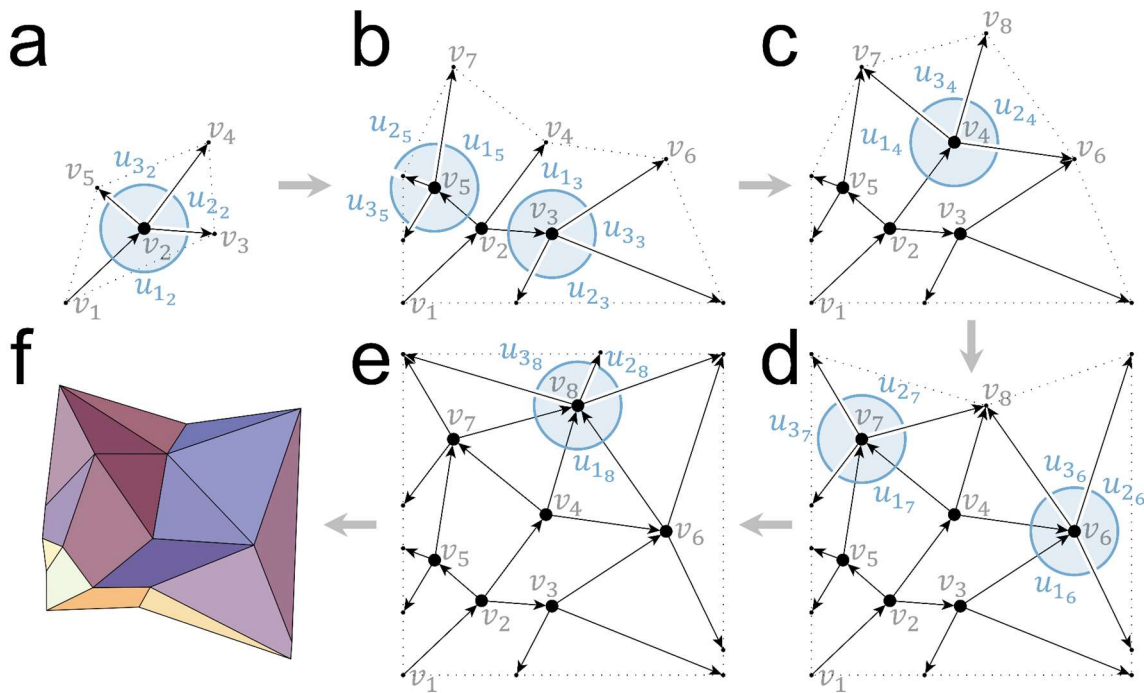


Fig. 30: (a-e) The step-wise generation of an arbitrary crease pattern represented as directed graph that complies with the rule that every vertex requires three outgoing edges to be kinematically determinate. In addition, the assignment of units that enables the assessment of the kinematic behavior of the entire crease pattern is illustrated for each individual vertex. (f) Arbitrary folded state of the resulting crease pattern.

In Fig. 30c, the same procedure is applied to vertex  $v_4$ , but this time two of the outgoing edges are connected to the already existing vertices  $v_6$  and  $v_7$ . Since there is no restriction on the number of incoming edges, vertices  $v_6$  and  $v_7$  can be equally extended and result in degree-5 vertices in Fig. 30d. Again, all the incoming dihedral angles are known for both vertices, so the units can be modeled and the sector angles can be designed such that the unit angles around  $v_6$  and  $v_7$  satisfy Eq. (25).

Two of the outgoing edges of  $v_6$  and  $v_7$  are connected to an existing vertex  $v_8$  that then becomes a degree-6 vertex after its extension in Fig. 30e. The PTU can be identically applied to  $v_8$ , with the sole difference that the unit  $u_{1_8}$  comprises four sectors and three driving angles. Note that the vertices on the boundary of the paper do not require three outgoing edges because they are not subject to the closure constraint of the surface.

At any state of the generation process in Fig. 30a-e, the graphs lead to kinematically determinate crease patterns. The kinematics of each generated vertex can be modeled as soon as it has been extended with three outgoing edges since at this point all information necessary to describe the unit angles is known. In this way, the PTU determines the propagation of unit angles throughout the crease pattern. In addition, the condition for rigid foldability given in Eq. (25) can be directly embedded in the design of the sector angles around the vertex so that each generated vertex and thus the entire crease pattern folds rigidly. This is illustrated in Fig. 30f that depicts an arbitrary rigidly foldable state of the final crease pattern.

However, there is an implicit assumption within the kinematic model of the PTU that restricts its application with respect to crease patterns. The kinematics of each individual vertex can only be assessed if all unit angles and thus all incoming  $n - 3$  dihedral angles are known. This leads to the condition that the underlying directed graph must be acyclic. Within such an acyclic graph, the information propagated to a vertex  $v$  by its predecessors is independent of the outgoing information of  $v$ . Consequently, all vertices can be kinematically modeled in the order in which they were generated, and the PTU provides the global rigid foldability of the pattern.

#### 4.2.5.2 Limitations: Acyclic Graphs

Fig. 31a depicts an acyclic graph with three vertices that exhibit a clear order in which they can be modeled. Vertex  $v_1$  is driven by the dashed edge and passes information to  $v_2$  that propagates its information to  $v_3$ . In Fig. 31b, one of the outgoing edges of  $v_3$  is incident to the driven vertex  $v_1$ . The



information propagated by this edge, however, is dependent on the kinematic behavior of  $v_1$  itself, in which case a system of constraints would have to be solved to gain the unknown dihedral angles. However, such a system is difficult to solve due to the high non-linearity of the unknown dihedral angles even for simple patterns such as in Fig. 31b, which essentially limits the PTU to acyclic graphs. Note that the same graph connectivity as in Fig. 31b results in a different graph when  $v_2$  is driven instead of  $v_1$  (Fig. 31c). In this case, the kinematics of vertices  $v_2$ ,  $v_3$ , and  $v_1$  can be modeled in the given order, and the PTU is again able to determine the global rigid foldability of the pattern. Unfortunately, the boundaries of the global rigid foldability in Fig. 31c cannot be transferred to the crease pattern in Fig. 31b because the range of motion itself depends on the chosen set of driving angles [119].

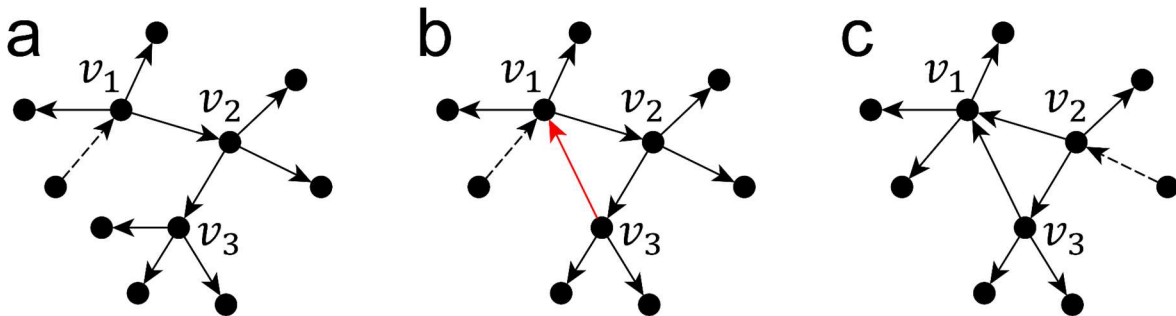


Fig. 31: Directed graphs with three vertices  $v_{1-3}$  and dashed edges that represent the driving information. (a) Acyclic graph with clear modeling order  $v_1, v_2, v_3$ . (b) Cyclic graph whose kinematics cannot be modeled by the PTU. (c) The same graph connectivity as in (b) leads to an acyclic graph when  $v_2$  is driven, in which case the modeling order is  $v_2, v_3, v_1$ .

#### 4.3 Manual Application and Results

The PTU offers deep insight into the mathematics of folding and presents vast possibilities for the design of origami crease patterns. To highlight these possibilities, this section shows the manual application of the PTU to generate an improved version of the adapted flasher (Fig. 8), to investigate the ability of the PTU to model different RBMs, and to design a target shape in the form of a chair in order to explain the influence of symmetry in origami as part of the answer to RQ3.

##### 4.3.1 Flasher Revisited

While the optimized geometry of the adapted flasher crease pattern in Fig. 17 does fold into the cuboid prescribed in Section 3.4, adding more layers to the outside of the pattern requires additional adjustments to the crease pattern geometry since the adapted flasher does not fold into a regular shape. Here, the focus lies on adjustments of the original pattern that preserve the regular folded shape and the actuation with one DOF on all four sides.

The approach taken in Section 3.3 to adapting the pattern is based purely on empirical evidence observed as crease lines appearing in the folded paper model, as suggested by Zirbel et al. [17]. Their suggestion involved the introduction of diagonal crease lines in the quadrilateral facets, which was complied with by adding crease lines to  $f_4$  and  $f_5$  in Fig. 7, resulting in the adapted flasher pattern in Fig. 32a. However, since there are two quadrilateral facets with two diagonals each, there arise four possible ways to adjust the original flasher pattern as illustrated in Fig. 32a-d.

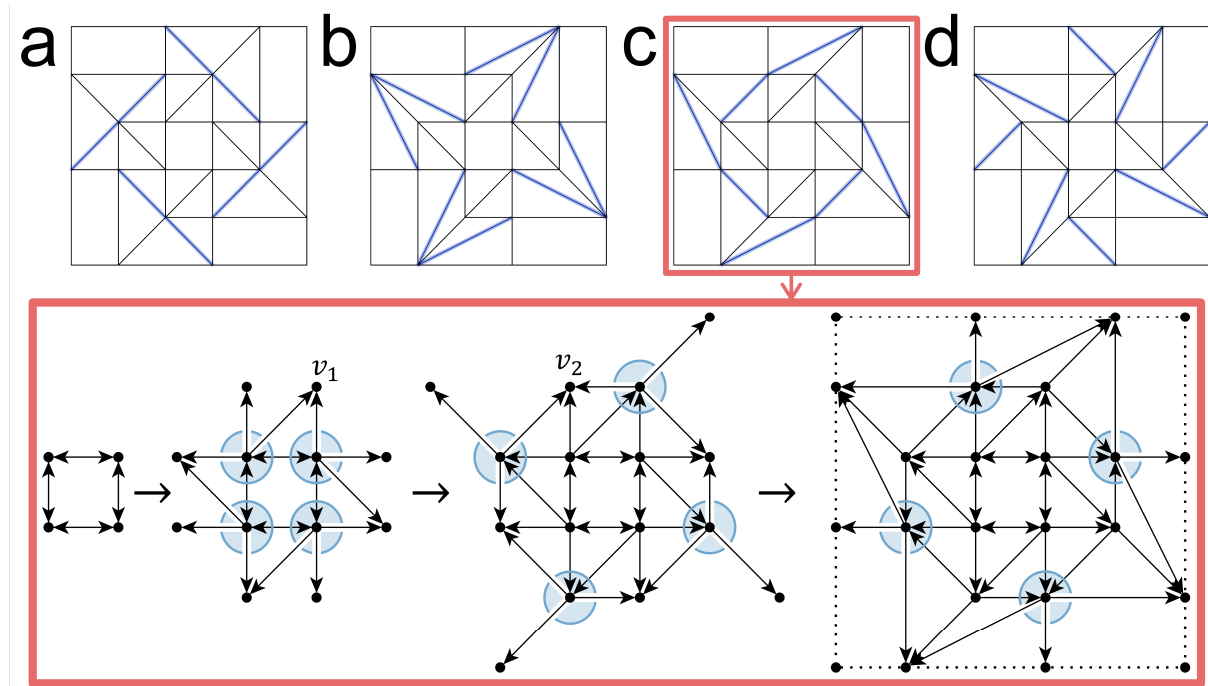


Fig. 32: Top: All different possibilities to adjust the original flasher pattern, where (a) corresponds to the adapted flasher while (b-d) have not yet been assessed. Bottom: Manual generation process from left to right of solution (c), the only crease pattern of (a-d) that can be modeled using the PTU.

The findings of the previous sections signify that the PTU can only model the pattern in Fig. 32c since the patterns in Fig. 32a, b, and d, although kinematically determinate, lead to cyclic graphs.

The manual generation process of the pattern in Fig. 32c is depicted at the bottom of Fig. 32 from left to right. The far left starts with the central facet whose crease lines are all driven by the same driving angles going from 0 to  $\frac{\pi}{2}$ . Represented as a directed graph, these driven crease lines result in double-sided arrows because the driving angles impact the incident vertices on both ends of the actuated edges, which leads to an indegree of two for each vertex of the central facet. Extending all of these vertices by three outgoing crease lines and successively connecting two of the new crease lines to the same vertex,  $v_1$ , on each of the four sides results in a first layer that is identical to the first layer of the adapted flasher pattern in Fig. 9a. Extending  $v_1$  and then  $v_2$  together with their rotations yields the pattern at the bottom far right of Fig. 32 that corresponds to the pattern in Fig. 32c.

Using the kinematic model of the PTU presented in Section 4.2.3, the folding motion of the novel pattern generated can now be assessed. As shown in Fig. 33, this new version of the flasher, here called the *rigid flasher*, folds rigidly with a single DOF (on all four sides) as depicted by the actuation on the bottom left of Fig. 32. In comparison to the adapted flasher (Fig. 9a), however, the rigid flasher folds perfectly onto the prescribed cuboid surface, which complies with the original purpose of the non-rigid flasher and allows for the extension of the rigid flasher to additional layers by analogously applying the PTU. The rigid flasher is thus an improved version of the adapted flasher and can serve as a basis for space applications, as presented by [17].

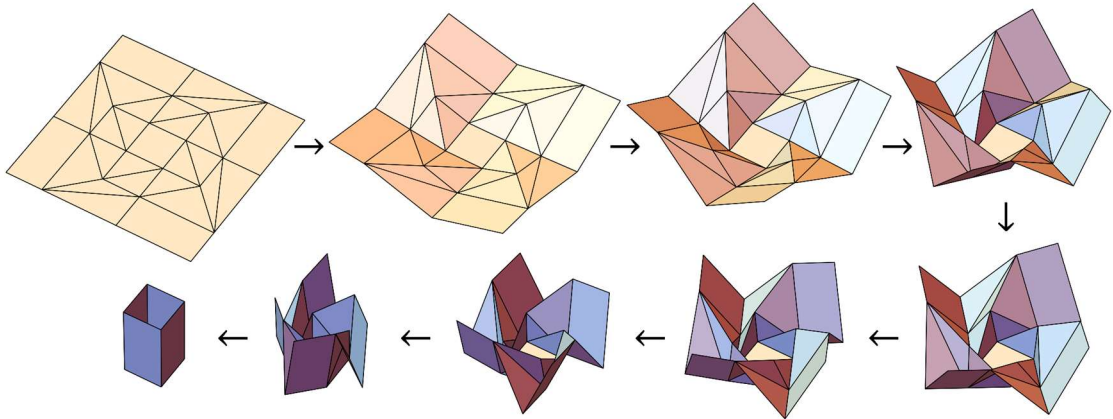


Fig. 33: Motion of the rigid flasher folding from the initial flat state into a perfect cuboid shape

4.3.2 Rigid Body Modes

As described in Section 4.2.3, the kinematic model of the PTU handles RBMs as an input, which enables the enumeration of different RBMs according to the needs of a design practitioner. To highlight this ability, Fig. 34 shows the generation process of a symmetric pattern here called the *star pattern* that resembles the flasher pattern but exhibits fewer internal vertices to decrease the number of RBMs.

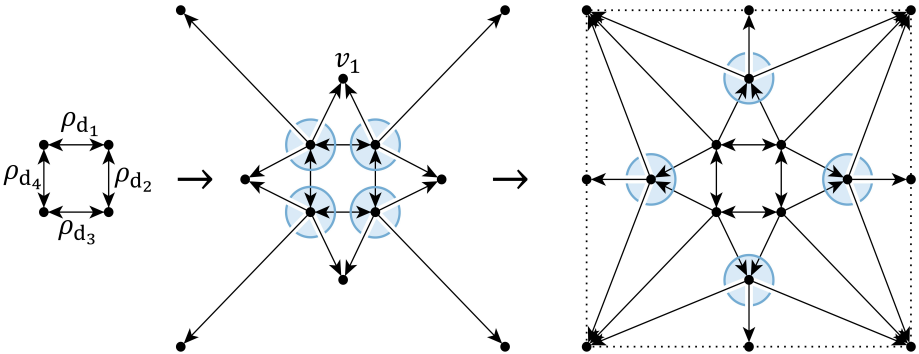


Fig. 34: Manual generation process of the star pattern from left to right

The same crease pattern used in Fig. 32 is initialized on the far left of Fig. 34, after which all four vertices of the central facet are extended. The topology of the resulting graph in the middle of Fig.

34 is identical to the first layer of the rigid flasher, but the diagonal crease lines are elongated and  $v_1$  and its rotations are located mirror-symmetrically. On the far right of Fig. 34,  $v_1$  and its rotations are extended with three outgoing edges, and two of these edges are incident to the vertices in the corner of the crease pattern.

The star pattern exhibits eight internal vertices, all of which can adopt two different RBMs, resulting in  $2^8 = 256$  different folded shapes. As expressed in Eq. (23), RBMs influence the unknown dihedral angles, but they do so symmetrically with respect to the flat plane, which is why the condition for the rigid foldability of a crease pattern in Eq. (25) does not depend on the RBMs. What does affect the rigid foldability, however, are the driving angles: for, e.g., a positive driving angle with a RBM  $M_1$ , the symmetric counterpart in motion is a negative driving angle with a RBM  $M_l$ , and vice versa. Instead of actuating the star pattern with identical driving angles  $\rho_{d_{1-4}}$  (Fig. 34) as done for the rigid flasher, all of these driving angles can be chosen independently of each other as long as the condition for rigid foldability is satisfied. Thus, the star pattern exhibits 256 different RBMs for each individual actuation, resulting in a much larger number of different possible motions, which again demonstrates the vast search space of origami. Hence, enumerating the whole search space of different motions for a simple pattern such as the star pattern is virtually impossible, which is why the benefit of the PTU is highlighted by copying a handful of RBMs and different actuations of an additively manufactured version of the star pattern in Fig. 35.

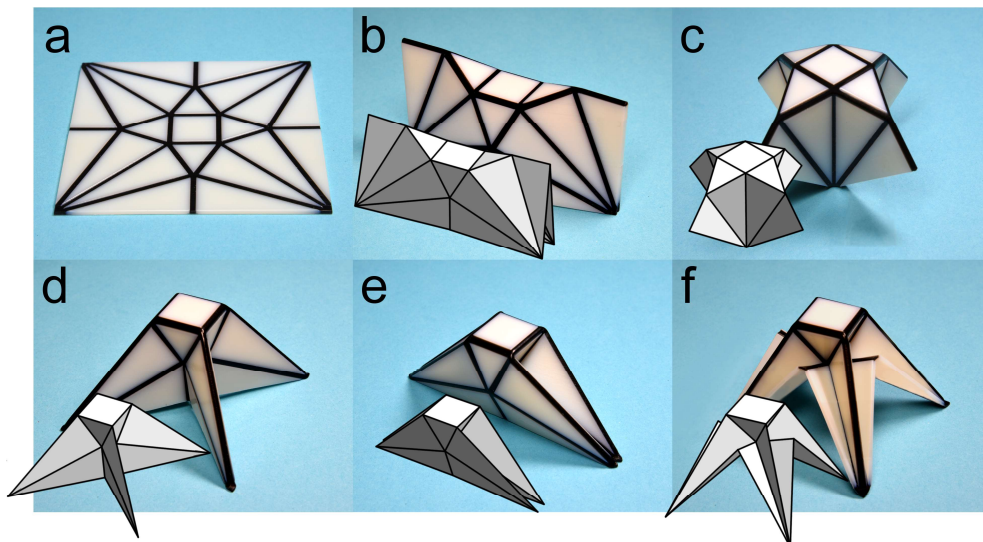


Fig. 35: Flat and folded states of the star pattern in comparison to an additively manufactured version showing different RBMs as well as different actuations

Fig. 35a shows the star pattern in its initial flat state, and Fig. 35c, Fig. 35d, and Fig. 35f depict folded states achieved by folding the pattern with identical driving angles  $\rho_{d_{1-4}}$ . Fig. 35b and Fig. 35e

are folded by varying the driving angles, where  $\rho_{d_2}$  and  $\rho_{d_4}$  are positive in Fig. 35b, and slightly more negative ( $\rho_{d_2} = \rho_{d_4} = -t$ ) in comparison to their symmetric counterparts ( $\rho_{d_1} = \rho_{d_3} = -\frac{t}{1.2}$ ) in Fig. 35e.

#### 4.3.3 Target Shape and Symmetry

In this section, the PTU is manually applied to design a crease pattern that mimics the target shape of a chair. While the crease patterns in the previous sections were generated in the flat state, the design process can be reversed to generate the crease pattern in the folded state. The only difference to the conventional approach is that the developability of the surface needs to be guaranteed in addition to the condition for rigid foldability. This requirement adds a simple set of constraints stating that the sum of the sector angles around each vertex needs to be equal to  $2\pi$ . Then, with some experience an origami design practitioner can utilize the PTU to plan and design a target shape, a process illustrated in Fig. 36.

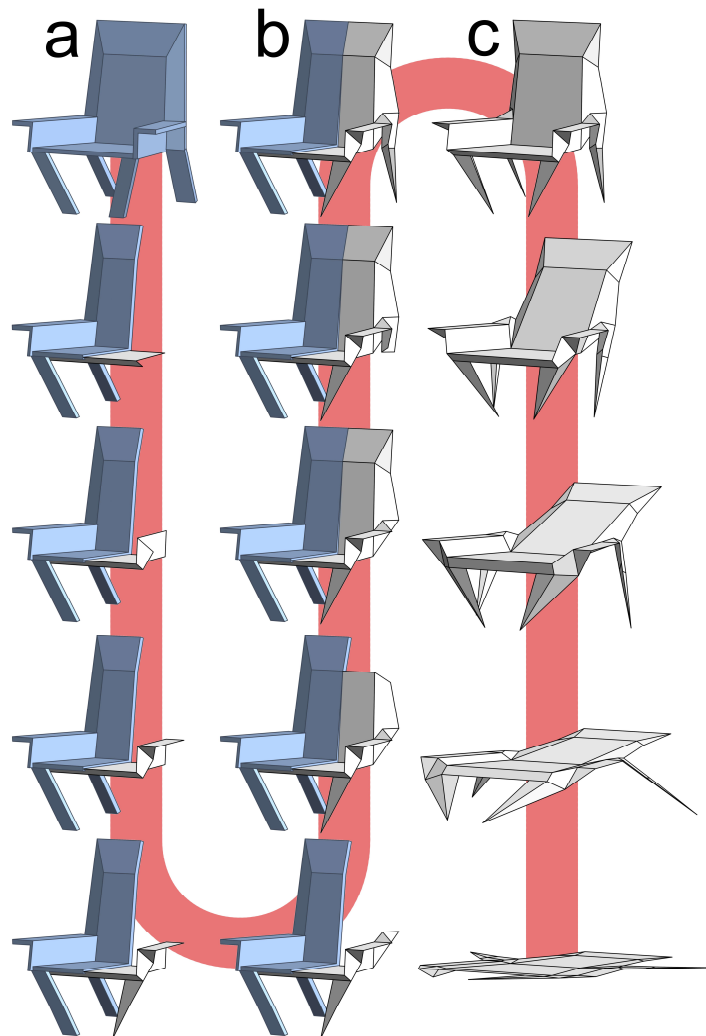


Fig. 36: The step-wise generation of an origami chair achieved by manually applying the PTU. The entire chair is depicted in (a), while one half of the model is eliminated in all steps until (b) to display the development of the origami. From (c) onwards, the origami chair folds from its deployed to its flat state with a single DOF.

Fig. 36 shows the stepwise generation of an origami chair from the original chair in Fig. 36a to the completed model shown in Fig. 36b, which, together with the intermediate steps, depicts a sliced version of the chair and its origami complement. Fig. 36c shows the chair in its full origami form that unfolds to the flat state at the bottom right of Fig. 36 with a single DOF.

The same procedure is illustrated in Fig. 37 where the generation steps are shown for the crease pattern in the unfolded state.

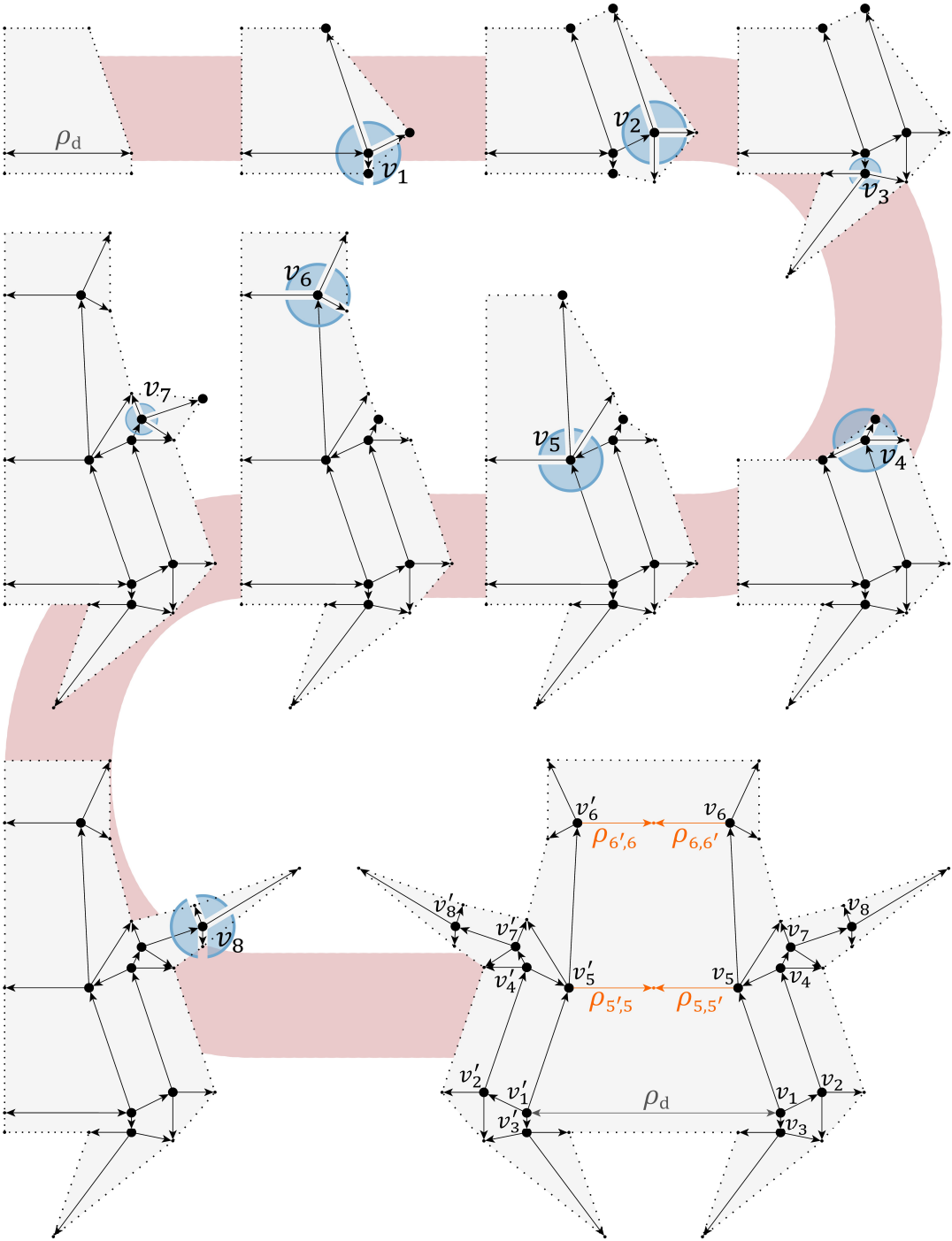


Fig. 37: The stepwise development of an origami chair pattern with vertices  $v_{1-8}$  and their symmetric counterparts  $v'_{1-8}$ , driving angle  $\rho_d$ , and identical sets of dihedral angles  $\rho_{5,5'} = \rho_{5',5}$  and  $\rho_{6,6'} = \rho_{6',6}$

The initial crease pattern at the top left of Fig. 37 corresponds to the first step after Fig. 36a and involves two facets sharing a crease line that is folded by the driving angle  $\rho_d$ . Each subsequent step signifies the addition of one vertex from  $v_1$  to  $v_8$ , first detailing the seating surface, then the armrest, the front leg, the backrest, and the hind leg. Finally, the one-sided crease pattern is mirrored to result in the final chair crease pattern at the bottom right of Fig. 37 that corresponds to Fig. 36c. The process complies with the guidelines of the PTU for the generation of crease patterns and satisfies the condition for rigid foldability. The sector angles are determined by adjusting their sizes to approximate the chair model in Fig. 36a in a trial-and-error process.

Special about the origami chair pattern at the bottom right of Fig. 37 is that it exhibits a mirror symmetry with respect to the vertical axis. In theory, the crease pattern is overconstrained around both vertices  $v_5$  and  $v_6$ . Due to the symmetry, however, the dihedral angles  $\rho_{5,5'}$  and  $\rho_{6,6'}$  are identical to their counterparts  $\rho_{5',5}$  and  $\rho_{6',6}$ , respectively, and the origami chair still folds rigidly. If any of the vertices  $v_{1-8}$ , their symmetric counterparts, or the vertices on the border of the pattern were to change location, any deviation of the driving angle  $\rho_d$  from its initial state would lead to a non-rigid motion. With respect to RQ3, this means that symmetry (both mirror symmetry as shown for the chair and rotational symmetry such as in the Miura-ori, as well as patterns emerging from fold angle multipliers) renders some constraints redundant and allows for seemingly overconstrained origami crease patterns to fold rigidly.

#### 4.4 Discussion

This section presents an investigation of the analytical kinematics of origami by first examining degree-4 vertices and then generalizing the findings for degree- $n$  vertices. This generalization encapsulates the findings for degree-4 vertices, so only the PTU and its implications are discussed here.

The PTU models the kinematics of single degree- $n$  vertices analytically and in real-time. Its efficiency is based on the existence of the vertex triangle at the core of a vertex. The sides of this triangle are well defined by the inputs necessary to fold the vertex, which are the sector angles, the  $n - 3$  driving angles, and a RBM  $M_1$  or  $M_1$ . The MV assignment implicitly depends on the RBM, and the three unknown dihedral angles can always satisfy the conditions presented by Abel et al. [62].

In addition, the kinematic model does not presuppose a developable surface and is thus applicable to generic spherical linkages with rotational joints. This ability vastly extends the possible applicability of the PTU and renders it a versatile method for the analysis and design of origami vertices and spherical mechanisms.

The only limitation of the kinematic model known to the author corresponds to the sign functions in Eqs. (20) and (22). When a start angle  $\beta_j$  or an end angle  $\delta_j$  converges toward  $\pi$  and surpasses that value, the value of the sign function switches, and the corresponding start or end angle jumps from  $+\pi$  to  $-\pi$ , or vice versa. Since this jump is equal to  $2\pi$  and thus a full rotation, the surface of the origami remains continuous, but the function of the corresponding unknown dihedral angle is discontinuous. This discontinuity might complicate future arithmetic operations although it is not certain if such operations are even applicable to the highly nonlinear dihedral angles. In addition, this situation is uncommon and leads to units that wrap around themselves, which usually results in self-intersecting vertices. Furthermore, the conditions for rigid and flat foldability are not affected since they depend only on the unit angles.

Next to its kinematic model, the PTU provides a set of intuitive answers to previously unsolved problems such as the conditions for the rigid and flat foldability of degree- $n$  vertices. While the former is based on the triangle inequality, the latter corresponds to the same equation converted into an equality. Note that the triangle inequality for spherical triangles is valid only for Euler triangles in which no sides are greater than  $\pi$  [124]. This condition is satisfied in origami with the range of the sector angles,  $0 < \alpha < \pi$ , and the unit angles  $U_j$  that are by definition smaller than  $\pi$ . The observation that both conditions for rigid and flat foldability are based on the same triangle provides a link between the two biggest notions in origami. At the last rigidly foldable state of a vertex, the vertex triangle degenerates into a flat triangle, and this state simply coincides in flat foldable vertices with the driving angles being at their extreme at  $\pm\pi$ . Hence, flat foldable vertices are the most robust vertices in terms of their range of motion.

However, the condition for flat foldability in Eq. (26) considers only the final folded state and does not guarantee that the vertex folds rigidly up to this state. A vertex that folds flat but not rigidly can occur in a developable vertex if the sum of the sectors of a unit is greater than  $\pi$  and if any two driving angles of that unit exhibit different signs. Then, the unit angle of that unit will initially increase when the driving angles deviate from zero and the condition for rigid foldability is not satisfied because the other two units already span the maximum distance in the unfolded state. It can happen, however, that the increased unit angle subsequently decreases and becomes smaller than the sum of the other units. The vertex then starts to fold rigidly midway through the folding motion, and thus exhibits a disconnected kinematic topology [20]. Such a vertex is able to fold flat although it does not fold rigidly during the entire motion. Fig. 38 shows the example of a developable vertex  $v(M_{\downarrow}, ((40,100,70)^T, (t, -t)^T), (70), (80))$  whose first unit initially spans sector angles that sum to a value greater than  $\pi$ ,  $40^\circ + 100^\circ + 70^\circ > 180^\circ$ ,



and whose driving angles are of opposite sign ( $t, -t$ ). The maximum unit angle  $U_{\max}$  thus increases initially, leading to a non-rigid motion for  $0 < t < \sim 1.36$ , so that the vertex folds flat but rigidly only from the dashed line onward.

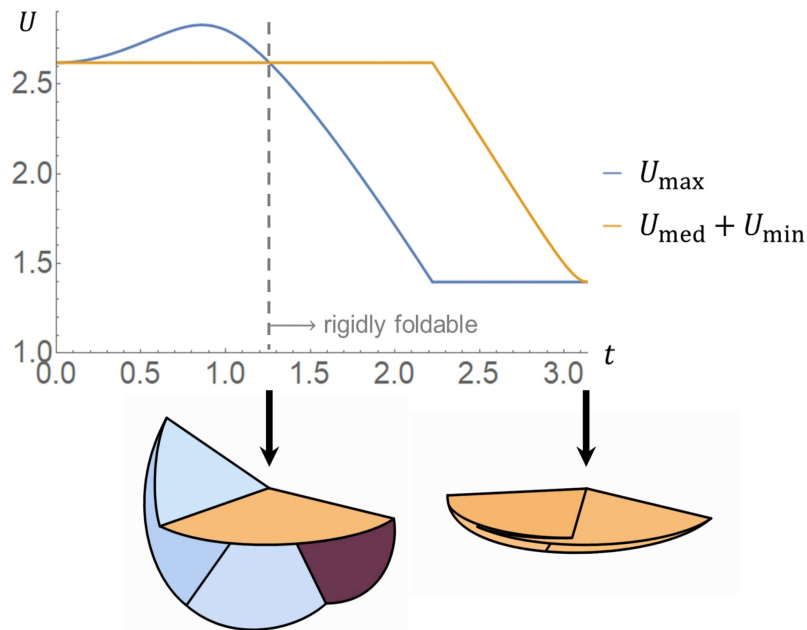


Fig. 38: A developable vertex that folds flat but only rigidly from the dashed line onward

Consequently, the condition for flat foldability only guarantees rigid foldability for vertices that involve units whose sector angles sum to less than  $\pi$ . Such vertices are generally more robust, especially within crease patterns where the sign of the propagated dihedral angles is not obvious without prior assessment.

With respect to crease patterns, the PTU contributes by specifying that the propagated information within a crease pattern are the unit angles  $U_j$  and by providing a global kinematic model for crease patterns that arise from acyclic graphs. Hence, although the PTU is a powerful tool to assess the motion and global rigid foldability of a wide range of crease patterns, it does not suffice for determining a generic condition for global rigid foldability. In addition, the PTU offers a simple rule for the generation of crease patterns, which states that an origami is kinematically determinate if its vertices exhibit three outgoing edges. Note that this rule is a sufficient rather than a necessary condition because crease pattern topologies with symmetry can be seemingly overconstrained and still fold rigidly, as shown in the example of the origami chair. In these cases, the multitude of incoming dihedral angles are congruent and thus redundant.

With respect to the second part of RQ1, the rigid foldability of an origami crease pattern can be assessed by focusing on the single vertices that constitute the building blocks of any crease pattern. To

fold rigidly, a single vertex has to comply with  $U_{\max} \leq U_{\text{med}} + U_{\min}$ , and when this condition is satisfied within an acyclic crease pattern graph, the entire origami folds rigidly. This answer to RQ1 is a major contribution to origami since rigid foldability is an important requirement for many engineering design tasks.

The PTU also answers RQ2 by offering a kinematic model for the different RBMs. In comparison to related methods that can simulate the RBMs only by including prior knowledge about the MV assignment or by perturbing the vertices, the PTU recognizes that the different RBMs arise from the two possible formations of the vertex triangle and thus enables the modeling of RBMs as a design input. This finding is a significant contribution to the design of origami crease patterns because the RBMs can be deliberately chosen based on the design task instead of posing an insurmountable geometric complexity.

With respect to RQ3, the PTU states that a crease pattern is kinematically determinate if all of its individual vertices have three outgoing crease lines. This rule can only be disobeyed when either mirror or rotational symmetry is present in the geometry of the crease pattern graph, or if the geometry obeys the fold angle multipliers in which case there emerges a symmetry of information. Symmetry reduces the number of constraints by rendering some constraints redundant because some of the incoming dihedral angles are congruent to the actuation of the crease pattern. The rule for the kinematic determinacy and the possible reduction of constraints through symmetry then result in the appropriate number of DOF. The number of DOF, which before was the subject of complex analyses for the mobility of origami mechanisms is now a simple input to the generation of a crease pattern as demonstrated by the examples in Sections 4.3.1-4.3.3. The explanation of these relations thus enables the straightforward generation of origami crease patterns, which is an important contribution to the science in origami.

In Section 4.4, the PTU is manually applied to design different crease patterns including the rigid flasher, the star pattern and an origami chair. However, the design space offered by origami is vast with respect to both the topology and geometry of crease pattern graphs as well as the RBMs. Manually applying the PTU leads to results only because the author either designed a crease pattern after a template (rigid flasher), generated a very simple pattern (star pattern), or already has experience in designing crease patterns (origami chair). The PTU thus still represents a tool for advanced origami design practitioners, and the full range of capabilities offered by origami can only be exploited by automating the design process. The embedding of the PTU and the answers to RQ1-RQ3 into an automated process, as prompted by RQ4, is described in the next section.

## 5 Analytical Approach to the Computational Method

In comparison to the numerical approach in Section 3, the PTU enables a more direct approach to the computational method for the synthesis of origami crease patterns. This section describes an automated graph grammar based approach that integrates the analytical condition for rigid foldability (RQ1), the new analytical kinematic model including the RBMs (RQ2), and the guidelines to generate kinematically determinate crease patterns so that the folding motion of the origami can be driven by the desired number of DOF (RQ3). The workflow of the automated method is illustrated in Fig. 39 where the six parts constituting a CDS method are highlighted in red.

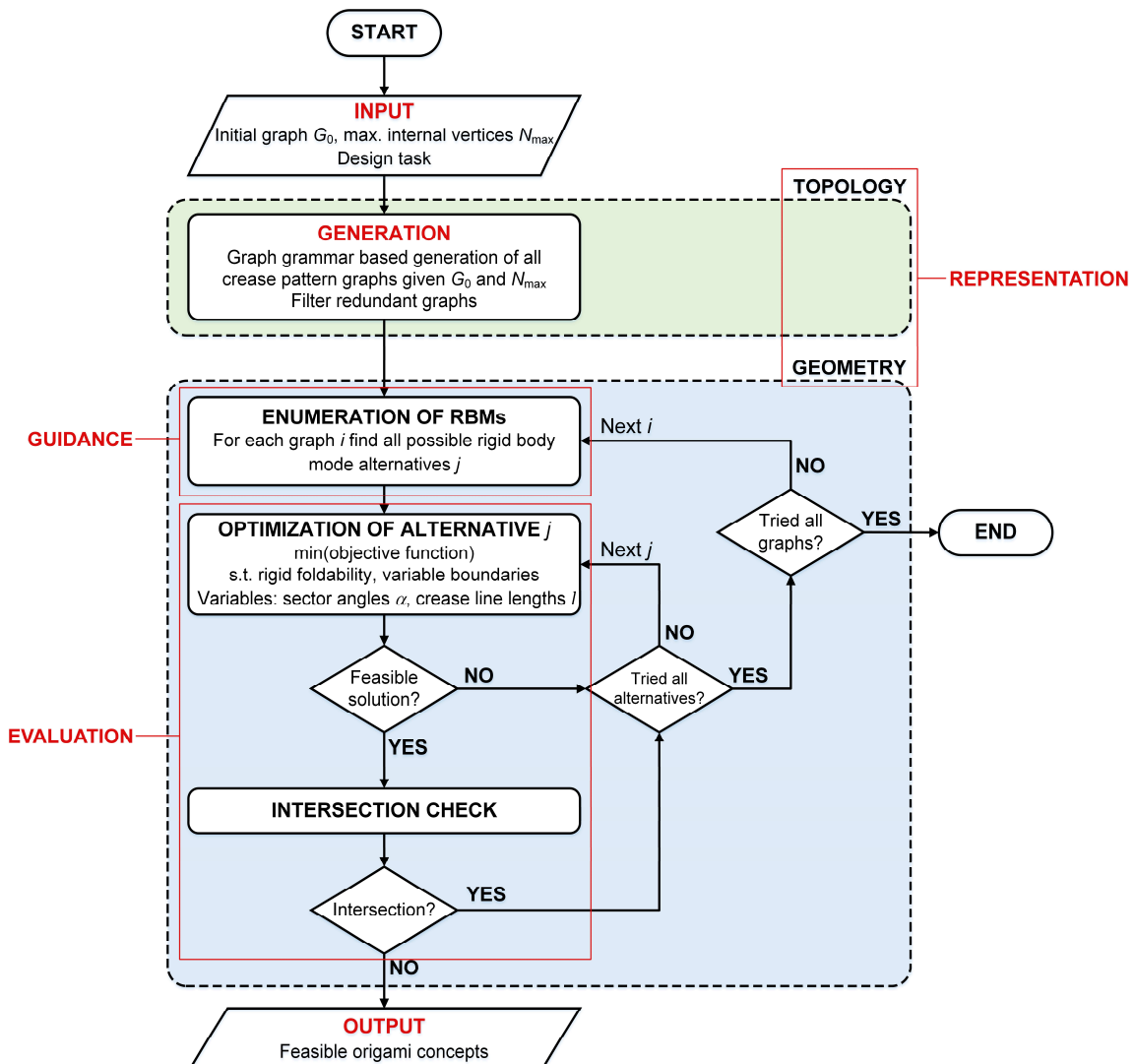


Fig. 39: Workflow of the method containing the input, the representation of the topology and the geometry, the generation, the guidance that enumerates all RBMs, the evaluation that incorporates both the optimization of design alternatives and the intersection check, as well as the output of a collection of feasible origami concepts that satisfy the given design task.

The user-defined input involves the definition of an initial graph  $G_0$  that serves as a starting point for the graph grammar generation of origami concepts, a maximum number of internal vertices  $N_{max}$

contained within these concepts, and the formulation of a design task that involves an objective function and other optimization-related inputs.

The computational method represents crease patterns as graphs and generates all possible graph topologies (“Topology” in Fig. 39) during the generation step, in which the method employs a rule-based graph grammar system to generate all possible crease pattern topologies that arise from the initial graph  $G_0$  and the maximum number of internal vertices  $N_{\max}$ . Simultaneously, the graph grammar automatically introduces the variables and constraints of the generated graph topologies for the optimization in subsequent steps. The generated topologies are then checked for redundancy and eliminated based on isomorphism as well as further elimination criteria that can be supplied manually if required.

Subsequently, the method steps into a loop in which a graph is first embedded within three dimensions by associating it with the vertex coordinates and the relevant variables to represent its geometry (“Geometry” in Fig. 39). Then, each graph is associated with all different RBM assignments to its internal vertices, corresponding to the enumeration of all *design alternatives*. To evaluate the best possible geometry of a design alternative, the evaluation step first optimizes each alternative and, if an optimized alternative meets the design criteria supplied with the design task, subjects it to an intersection check. A design alternative that fails this intersection check is discarded, otherwise it is added to the list of feasible origami concepts.

This section is structured according to a CDS method including the representation, the generation, the guidance, and the evaluation. Then, the proposed method for the synthesis of novel origami concepts is applied to two engineering tasks that involve the design of rigid origami grippers and robotic arms. Subsequently, the section presents the results and discusses the method.

## 5.1 Representation

The topology of a crease pattern is represented as a directed, labeled graph  $G = (V, E, L_V, L_E)$ , where  $V$  is a non-empty set of vertices  $v_i$  and  $E$  is a non-empty set of directed edges  $e_{i,j}$  from vertices  $v_i$  to  $v_j$  such that  $i \neq j$ . Vertices  $v_i$  are called the *predecessors* of  $v_j$  and  $v_j$  are called the *successors* of  $v_i$ .  $L_V$  is a non-empty set of vertex labels and  $L_E$  is a non-empty set of edge labels. Let  $\Sigma_v$  be a map  $\Sigma_v: V \rightarrow L_V$  that labels each vertex  $v_i$  to its ordered predecessors  $P_i$  and a type  $T_i$ , so that each vertex label  $L_i$  in  $L_V$  is an ordered pair defined as  $L_i = (P_i, T_i)$ . If a vertex  $v_i$  has no predecessors,  $P_i = (\emptyset)$ , then  $v_i$  is a *source* vertex. Otherwise,  $P_i$  is populated with all predecessors of  $v_i$  ordered in the counter-

clockwise direction with respect to the crease pattern graph surrounding  $v_i$ . The type  $T_i$  of vertex  $v_i$  adopts symbol  $\chi$  when the vertex can be extended, which means that new outgoing edges can be generated, or it adopts the symbol  $T_i = \emptyset$  when the vertex is either a source or has already been extended. Let  $\Sigma_E$  be a map  $\Sigma_E: E \rightarrow L_E$  that assigns an edge label  $L_{i,j}$  in  $L_E$  to each directed edge  $e_{i,j}$ , where  $L_{i,j} = (F_{L_{i,j}}, F_{R_{i,j}})$  is an ordered pair of the facets  $f$  located on the left and on the right side of  $e_{i,j}$ , respectively.

To clarify the notation, Fig. 40 shows an example graph that also represents the simplest possible initial graph. This graph consists of just two vertices  $v_1$  and  $v_2$  connected by a directed edge  $e_{1,2}$ . The vertex label  $L_1 = ((\emptyset), \emptyset)$  denotes that the vertex  $v_1$  is a source and cannot be extended, respectively. The vertex label  $L_2 = ((v_1), \chi)$  denotes that  $v_1$  is the single predecessor of  $v_2$  and that  $v_2$  can be extended, respectively. Finally, the two components  $F_{L_{1,2}} = f_1$  and  $F_{R_{1,2}} = f_2$  of the edge label  $L_{1,2}$  signify that the edge  $e_{1,2}$  is adjacent to the facet  $f_1$  on its left and  $f_2$  on its right side.

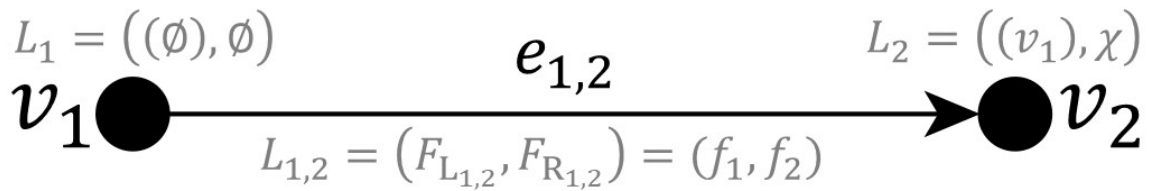


Fig. 40: Example of an initial graph  $G_0$  comprised of two vertices  $v_1$  and  $v_2$  connected by a directed edge  $e_{1,2}$ , vertex labels  $L_1$  and  $L_2$ , as well as the edge label  $L_{1,2}$

In addition to the initialization associated with the representation of the topology, the user has to relate the initial graph to its engineering purpose and thus to its geometry. Hence, the user needs to allocate in-plane coordinates  $\mathbf{x}_i$  to all vertices  $v_i$  and define a driving angle  $\rho_{i,j}$  for every incoming edge incident to a vertex  $v_j$  that can be extended. As an example, for the graph in Fig. 40 this could be defined as  $\mathbf{x}_1 = (0,0)$ ,  $\mathbf{x}_2 = (1,0)$ , and  $\rho_{1,2} = t$  where  $t$  represents a linear driving angle that goes from zero to some value  $t_{\max} \in [-\pi, \pi]$ .

## 5.2 Generation

The origami graph grammar  $GG$  is defined by the triple  $GG = (G_0, \mathcal{R}, \emptyset)$  where  $G_0$  is the initial graph,  $\mathcal{R} = (r_1, r_2)$  is the set of rules containing  $r_1$  and  $r_2$ , and  $\emptyset$  is the terminal symbol with respect to the type  $T_i$  of a vertex  $v_i$  that prevents rule  $r_1$  from being applied to a vertex with  $T_i = \emptyset$ .

The definition of the initial graph  $G_0$  is an input to the method and can involve many different topologies. This definition, however, is restricted by the two guidelines for the generation of kinematically

determinate crease patterns given in Section 4: first, all initial graphs  $G_0$  must be acyclic. Second, the outdegree of all vertices in  $G_0$  must be smaller than 3; the method models the kinematics of a vertex while extending that vertex, and it does not extend vertices that already satisfy  $\deg^+(v_i) = 3$ . In fact, it makes sense to initialize a graph as simply as possible to minimize the number of DOF and the tedium of a human design practitioner, which is why the initial graph  $G_0$  in Fig. 40 will be used for both design tasks in Section 5.5.

The rule set  $\mathcal{R} = (r_1, r_2)$  is designed to conform to the two guidelines presented in Section 4. The first rule  $r_1$  extends a vertex by new outgoing edges and incident vertices and ensures that each extended vertex has three outgoing edges after the application. The second rule  $r_2$  combines two vertices and their incoming edges to enable the generation of higher order vertices while guaranteeing that the generated graphs stay acyclic. In addition to the graph transformations, the rules  $r_1$  and  $r_2$  embed a graph  $G$  in the plane, model extended vertices kinematically, and build or adjust the sets of the optimization variables and constraints. To automate this process, the method initializes empty sets for the in-plane coordinates  $x$ , the units  $u$ , the dihedral angles  $\rho$ , the global rotation matrices  $R$ , the optimization variables  $\Phi$ , and the optimization constraints  $\psi$ . Moreover, the method initializes a set of three-dimensional vertex coordinates  $X = \{\mathbf{X}_i\}$  where  $\mathbf{X}_i$  are the in-plane coordinates of all vertices  $v_i$  contained within the initial graph  $G_0$ , which are appended with a zero that represents the  $z$ -coordinate. Then, these sets are automatically filled and adjusted when the rules  $r_1$  and  $r_2$  are applied.

### 5.2.1 Rule $r_1$ : Extend Vertex

Rule  $r_1$  is a production of the type  $r_1: LHS_1 \rightarrow RHS_1$ , where the  $LHS_1$  is a single vertex  $v_a$  to which the rule is applied and where the  $RHS_1$  is the same vertex  $v_a$  with a number of new outgoing edges and incident vertices. Depending on the initial graph  $G_0$ , the outdegree  $\deg^+(v_a)$  on the  $LHS_1$  can be 0, 1, or 2, and  $r_1$  should accordingly generate  $3 - \deg^+(v_a)$  new edges and vertices. Thus, rule  $r_1$  is parametric with respect to the number of successors of  $v_a$  on the  $LHS_1$ , which results in six possible rule application scenarios  $r_{1,1} - r_{1,6}$ . For clarity, only the scenario  $r_{1,1}$  (Fig. 41) is described in detail, whereas all other rule application scenarios  $r_{1,2} - r_{1,6}$  are listed in a more concise form in Section 5.2.1.6. As illustrated in Fig. 41,  $r_{1,1}$  corresponds to the case where  $v_a$  has no successors on the  $LHS_1$ .

Fig. 41 depicts the  $LHS_{1,1}$  and the  $RHS_{1,1}$  in red and the graph surrounding  $v_a$  in grey as a reference. This surrounding graph consists of any number of predecessors  $v_{p_m}$  for  $1 \dots m$  ordered in the

counter-clockwise order, where the respective edges are divided by the sector angles  $\alpha_{p_1}$  to  $\alpha_{p_{m-1}}$ . In addition, the top of Fig. 41 shows a schematic graph transformation (framed).

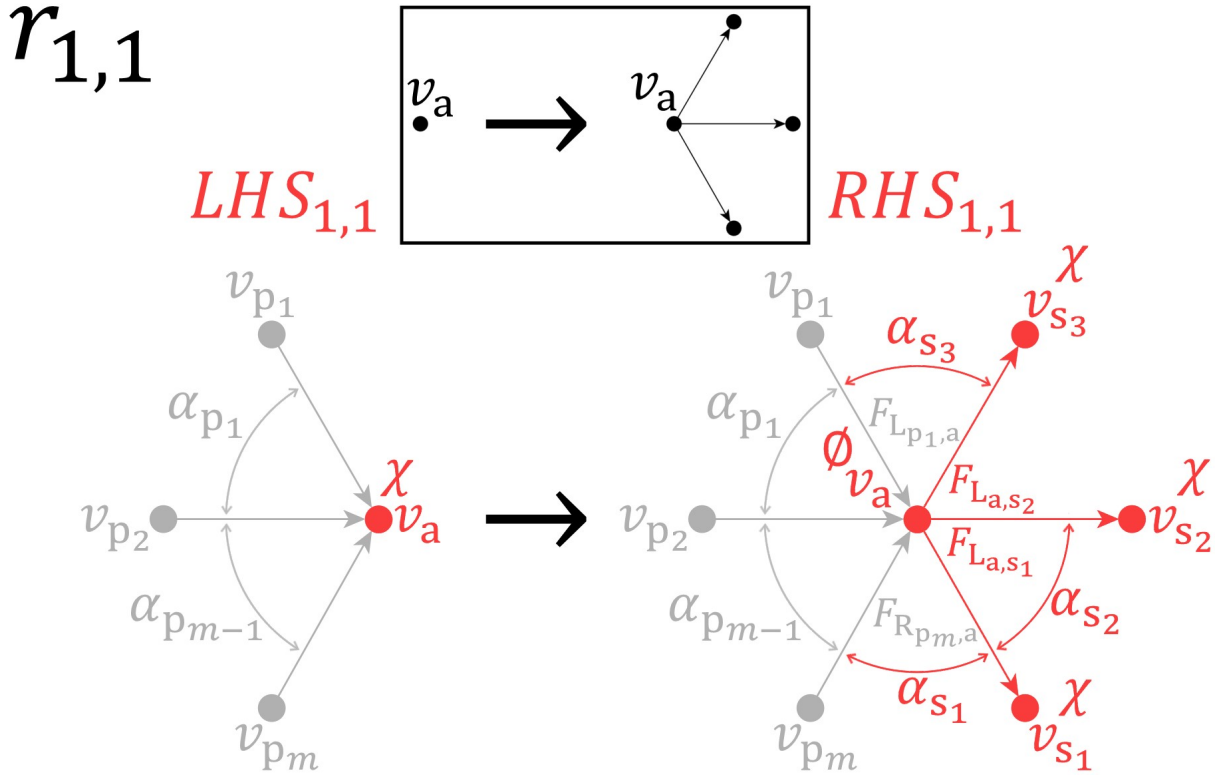


Fig. 41: Rule  $r_{1,1}$  for the specific scenario in which there are no successors of  $v_a$  on the  $LHS_1$ . Top: Schematic graph transformation (framed). Bottom:  $LHS_{1,1}$  and  $RHS_{1,1}$  (red) and surrounding graph (grey).

### 5.2.1.1 LHS Matching of $r_{1,1}$

A match  $\mathcal{M}_{1,1}$  of the  $LHS_{1,1}$  is found if  $v_a$  is extendable as defined by its type  $T_a = \chi$  and if  $\deg^+(v_a) = 0$  (which corresponds to this specific scenario).

### 5.2.1.2 Graph Transformation of $r_{1,1}$

On the  $RHS_{1,1}$ , rule  $r_{1,1}$  generates three new successor vertices  $v_{s_{1-3}}$  numbered in counter-clockwise order that are connected to  $v_a$  with directed edges  $e_{a,s_{1-3}}$ . All generated vertices have identical vertex labels  $L_{s_{1-3}} = ((v_a), \chi)$ , making them applicable for a successive rule application of  $r_1$ . While the edge labels adjacent to existing facets adopt the edge labels of the corresponding edges ( $F_{L_{p_1,a}}$  and  $F_{R_{p_m,a}}$  in Fig. 41 right), two new facets are generated and denoted with  $F_{L_{a,s_1}}$  and  $F_{L_{a,s_2}}$ . Hence, the edge labels result in  $L_{a,s_1} = (F_{L_{a,s_1}}, F_{R_{p_m,a}})$ ,  $L_{a,s_2} = (F_{L_{a,s_2}}, F_{L_{a,s_1}})$ , and  $L_{a,s_3} = (F_{L_{p_1,a}}, F_{L_{a,s_2}})$ . All newly generated indices  $s_{1-3}$  are in accordance with the enumeration of the nodes in  $G$  prior to the rule application.

The application of  $r_1$  to a graph  $G$ ,  $G \xrightarrow{r_1} G'$ , results in the graph  $G' = (V' = V \cup \{v_{s_{1-3}}\}, E' = E \cup \{e_{a,s_{1-3}}\}, L'_V = L_V \cup \{L_{s_{1-3}}\}, L'_E = L_E \cup \{L_{a,s_{1-3}}\})$  where  $\cup$  represents the disjoint union of sets [125]. In addition, the type of the vertex  $v_a$  is set to  $T'_a = \emptyset$  to prevent any further extension of  $v_a$ .

### 5.2.1.3 Coordinates and Optimization Variables of $r_{1,1}$

Having completed the graph transformation of  $r_{1,1}$  that affects the graph topology, more information is required about the graph geometry to apply the second rule  $r_2$  and perform the subsequent optimization. To embed the graph  $G'$  within the plane, a set of coordinates is required to describe the locations of all generated successors. Hence, all new edges  $e_{a,s_{1-3}}$  are located with their respective sector angles  $\alpha_{s_{1-3}}$  (Fig. 41 right) and assigned with an edge length  $l_{s_{1-3}}$ . Using the normalized direction vector  $\mathbf{d}_{i,j}$  between vertex locations  $\mathbf{x}_i$  and  $\mathbf{x}_j$  of vertices  $v_i$  and  $v_j$ :

$$\mathbf{d}_{i,j} = \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \quad (27)$$

the in-plane locations of the generated successors with respect to the position  $\mathbf{x}_a$  of  $v_a$  and the predecessors  $P_a$  can be expressed as:

$$\begin{aligned} \mathbf{x}_{s_1} &= \mathbf{x}_a + l_{a,s_1} \mathbf{R}(\alpha_{s_1}) \mathbf{d}_{a,p_m} \\ \mathbf{x}_{s_2} &= \mathbf{x}_a + l_{a,s_2} \mathbf{R}(\alpha_{s_1} + \alpha_{s_2}) \mathbf{d}_{a,p_m} \\ \mathbf{x}_{s_3} &= \mathbf{x}_a + l_{a,s_3} \mathbf{R}(-\alpha_{s_3}) \mathbf{d}_{a,p_1} \end{aligned} \quad (28)$$

where  $\mathbf{R}$  is the two-dimensional rotation matrix. The locations of the newly generated vertices are then added to the set of in-plane coordinates,  $x' = x \cup \{\mathbf{x}_{s_{1-3}}\}$ .

In addition to the sector angles and edge lengths, all scenarios of  $r_1$  introduce an optimization variable  $M_a$  that stands for the RBM of the vertex  $v_a$ . The set of optimization variables is then expressed as  $\Phi' = \Phi \cup \{\alpha_{s_{1-3}}, l_{a,s_{1-3}}, M_a\}$ .

### 5.2.1.4 Kinematic Modeling of $r_{1,1}$

When  $r_1$  is applied,  $v_a$  becomes an internal vertex of the crease pattern and can be kinematically modeled using the PTU. According to the scenario of  $r_{1,1}$  in Fig. 41, the units  $u_{i_a}$  of  $v_a$  can be expressed in terms of their respective sector and dihedral angles as:

$$\begin{aligned} u_{1_a} &= \left( (\alpha_{s_3}, \alpha_{p_1}, \dots, \alpha_{p_{m-1}}, \alpha_{s_1}), (\rho_{p_{1,a}}, \dots, \rho_{p_{m,a}}) \right) \\ u_{2_a} &= (\alpha_{s_2}) \\ u_{3_a} &= \left( 2\pi - \sum_{i=1}^3 \alpha_{s_i} - \sum_{j=1}^m \alpha_{p_j} \right) \end{aligned} \quad (29)$$

If  $v_a$  exhibits only a single predecessor, the sum over  $\alpha_{p_j}$  in  $u_{3_a}$  is neglected. Then, the units in Eq. (29) are added to the set of units,  $u' = u \cup \{u_{1a-3a}\}$ . By setting these units and the RBM  $M_a$  into Eq.



(23), the method calculates the unknown dihedral angles  $\varphi_{1a-3a}$  of  $v_a$  and associates them with the generated edges in the correct order shown in Fig. 26,  $\rho_{a,s_1} = \varphi_{3a}$ ,  $\rho_{a,s_2} = \varphi_{1a}$ , and  $\rho_{a,s_3} = \varphi_{2a}$ . These angles are then added to the set of dihedral angles,  $\rho' = \rho \cup \{\rho_{a,s_{1-3}}\}$ .

The expression of the units and the dihedral angles enables the method to kinematically model each extended vertex individually. However, the locations of the vertices need to be expressed in the three-dimensional space to represent the actual origami, which requires the method to transfer the individual (local) kinematics to a global kinematic behavior. To do so, the method uses the kinematic model presented in Section 4.2.3 to express the function

$$f_{\mathbf{R}}(M_a, u_{1a}, u_{2a}, u_{3a}) = (\mathbf{R}_{3a}, \mathbf{R}_{1a}, \mathbf{R}_{2a}) \quad (30)$$

that takes the RBM  $M_a$  and the three units  $u_{1a-3a}$  of  $v_a$  and returns three rotation matrices, here called the *local* rotation matrices  $\mathbf{R}_{1a-3a}$ , that correspond to the rotations at the crease lines of the unknown dihedral angles, as shown in Fig. 42 for an arbitrary single vertex. This calculation always starts from the first crease line of the first unit that serve as references for the  $x$ -axis and the  $xy$ -plane, respectively. In Eq. (30), the local rotation matrices are expressed in counter-clockwise order starting from the first unit to conform to the numbering of the newly generated edges in Fig. 41.

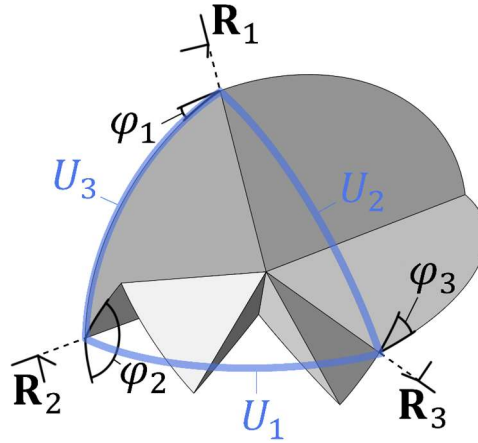


Fig. 42: Arbitrary single vertex with unit angles  $U_{1-3}$ , unknown dihedral angles  $\varphi_{1-3}$ , and local rotation matrices  $\mathbf{R}_{1-3}$

By applying  $f_{\mathbf{R}}$ , the method determines the local rotation matrices and turns them into global rotation matrices with  $\mathbf{R}_{a,s_1} = \mathbf{R}_{p_{1,a}} \mathbf{R}_z(\pi) \mathbf{R}_{3a}$ ,  $\mathbf{R}_{a,s_2} = \mathbf{R}_{p_{1,a}} \mathbf{R}_z(\pi) \mathbf{R}_{1a}$ , and  $\mathbf{R}_{a,s_3} = \mathbf{R}_{p_{1,a}} \mathbf{R}_z(\pi) \mathbf{R}_{2a}$ , where  $\mathbf{R}_{p_{1,a}}$  is the global rotation matrix of the first crease line of the first unit. A special case arises when  $v_a$  belongs to the initial graph since the global rotation matrix  $\mathbf{R}_{p_{1,a}}$  has not been determined by an application of  $r_1$ . In this case, the method sets  $\mathbf{R}_{p_{1,a}} = \mathbf{R}_z(\alpha_p)$  where  $\alpha_p$  is the angle between the edge

$e_{p_{1,a}}$  and the  $x$ -axis. The global rotation matrices are then added to the set of global rotation matrices  $R' = R \cup \{\mathbf{R}_{a,s_{1-3}}\}$ . Since the first crease line of a vertex is locally associated with the  $x$ -axis, the three-dimensional coordinates  $\mathbf{X}_{s_i}$  can be expressed as  $\mathbf{X}_{s_i} = \mathbf{X}_a + l_{a,s_i} * \mathbf{R}_{a,s_i} \cdot (1, 0, 0)^T$ . These coordinates are subsequently added to the set of three-dimensional vertex coordinates,  $X' = X \cup \{\mathbf{X}_{s_{1-3}}\}$ .

#### 5.2.1.5 Optimization Constraints of $r_{1,1}$

Independent of the scenario, two types of optimization constraints are produced by  $r_1$ , of which one defines the boundaries of the optimization variables while the other pertains to the rigid foldability.

The method introduces finite lower and upper bounds for the sector angles that are greater than zero and smaller than  $\pi$ , respectively. As a result of the subsequent optimization, the shapes of the facets then become less sheared and less thin than their counterparts without these constraints, which facilitates the practical realization of crease patterns with finitely thick materials in the future. Eq. (31) lists the optimization constraints applied to the sector angles as:

$$\begin{aligned} 0 < \alpha_{\min} \leq \alpha_{s_{1-3}} \leq \alpha_{\max} < \pi \\ \sum_{i=1}^3 \alpha_{s_i} + \sum_{j=1}^m \alpha_{p_j} \leq 2\pi - \alpha_{\min} \end{aligned} \quad (31)$$

where the second row pertains to the sector angle of the sector corresponding to  $F_{L_{a,s_2}}$  in Fig. 41. Again, if  $v_a$  exhibits only a single predecessor, the sum over  $\alpha_{p_j}$  in Eq. (31) is neglected. The same procedure is applied to the lengths of the generated edges:

$$l_{\min} \leq l_{s_{1-3}} \leq l_{\max} \quad (32)$$

The second type of constraint introduced by  $r_1$  corresponds to the condition for rigid foldability in Eq. (25) that states  $U_{\max}(t) \leq U_{\text{med}}(t) + U_{\min}(t)$ . The difficulty for the application of this condition arises from the fact that the unit angles change their size over time depending on the parameter  $t$ , which means that Eq. (25) has to be satisfied for the entire folding motion.

Analogous to Fig. 41, the rule system presented here always results in only one time-dependent unit  $u_1$ , whereas  $u_2$  and  $u_3$  are constant. A set of optimization constraints that guarantees rigid foldability for the entire folding procedure thus involves the application of Eq. (25) with both  $\min(U_1)$  and  $\max(U_1)$ . When the cases portrayed in Fig. 38 are discarded, however, one can reasonably assume that  $\max(U_1)$  occurs at  $t = 0$ , in which case Eq. (25) is satisfied in the initial flat state. Then, the unit angle  $U_1$

decreases monotonically once  $t$  deviates from zero toward  $t = t_{\max}$ , which is why the proposed method only examines the case  $\min(U_1)$  that occurs at  $t = t_{\max}$ :

$$U_{\max}(t_{\max}) \leq U_{\text{med}}(t_{\max}) + U_{\min}(t_{\max}) \quad (33)$$

The constraints in Eq. (31), (32), and (33) are then added to the set of optimization constraints  $\psi'$  after each application of  $r_1$ .

### 5.2.1.6 Other scenarios for $r_1$

The remaining scenarios for  $r_1$ ,  $r_{1,2} - r_{1,6}$ , are implemented analogously to  $r_{1,1}$ , but the description of these scenarios are given in short in Table 1 and Table 2 that correspond to Fig. 43 and Fig. 44, respectively. Fig. 43 illustrates the scenarios  $r_{1,2}$  and  $r_{1,3}$  that differ from  $r_{1,1}$  in that the outdegree of  $v_a$  on the  $LHS_{1,2}$  and  $LHS_{1,3}$  is  $\deg^+(v_a) = 1$ , in which case there exists a predefined facet on either side of  $v_a$  in the initial graph  $G_0$  as illustrated in grey in the schematic graph transformations (framed).

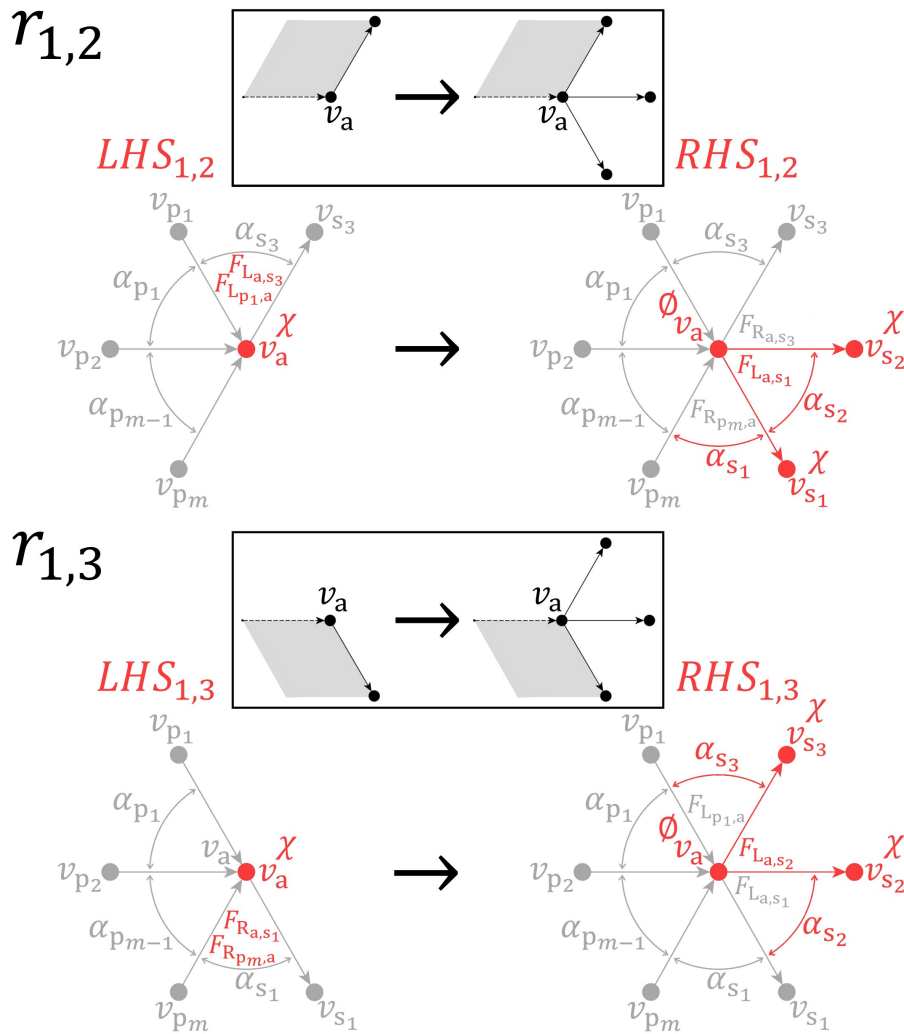


Fig. 43: Scenarios for  $r_{1,2}$  and  $r_{1,3}$  in which  $\deg^+(v_a) = 1$  on the  $LHS_{1,2}$  and  $LHS_{1,3}$  (red) together with the surrounding graph (grey), and schematic graph transformations (framed) where the grey facets are given in the initial graph  $G_0$

All the necessary information for the rule scenarios  $r_{1,2}$  and  $r_{1,3}$  is listed individually in Table 1.

The LHS matching and the graph transformations that correspond to the graph topology are highlighted in red. The kinematic modeling, the optimization variables, the optimization constraints, as well as the in-plane coordinates all correspond to the crease pattern geometry or the optimization model. If  $v_a$  has only one predecessor, all terms containing  $\alpha_p$  are neglected.

Table 1: Information of the rule scenarios  $r_{1,2}$  and  $r_{1,3}$

$r_{1,2}$		
LHS Match	RHS Transformation	Kinematic Modeling
$T_a = \chi$ $\deg^+(v_a) = 1$ $F_{L_{a,s_3}} = F_{L_{p_1,a}}$	$T'_a = \emptyset$ $V' = V \cup \{v_{s_1}, v_{s_2}\}$ $E' = E \cup \{e_{a,s_1}, e_{a,s_2}\}$ $L'_V = L_V \cup \{(v_a, \chi), ((v_a), \chi)\}$ $L'_E = L_E \cup \{(F_{L_{a,s_1}}, F_{R_{p_m,a}}), (F_{R_{a,s_3}}, F_{L_{a,s_1}})\}$	$u_1 = ((\alpha_{s_3}, \alpha_p, \alpha_{s_1}), (\rho_p))$ $u_2 = (\alpha_{s_2})$ $u_3 = \left(2\pi - \sum_{i=1}^3 \alpha_{s_i} - \sum_{j=1}^{m-1} \alpha_{p_j}\right)$
<b>Optimization Constraints</b>		<b>Optimization Variables</b>
$U_{\max} \leq U_{\text{med}} + U_{\min}$ $\alpha_{\min} \leq \alpha_{s_1} \leq \alpha_{\max}, \alpha_{\min} \leq \alpha_{s_2} \leq \alpha_{\max}$ $\sum_{i=1}^3 \alpha_{s_i} + \sum_{j=1}^{m-1} \alpha_{p_j} \leq 2\pi - \alpha_{\min}$ $l_{\min} \leq l_{s_1} \leq l_{\min}, l_{\min} \leq l_{s_2} \leq l_{\min}$		$M_a, \alpha_{s_1}, \alpha_{s_2}, l_{a,s_1}, l_{a,s_2}$
		<b>In-plane Coordinates</b>
		$\mathbf{x}_{s_1} = \mathbf{x}_a + l_{a,s_1} \mathbf{R}(\alpha_{s_1}) \mathbf{d}_{a,p_m}$ $\mathbf{x}_{s_2} = \mathbf{x}_a + l_{a,s_2} \mathbf{R}(\alpha_{s_1} + \alpha_{s_2}) \mathbf{d}_{a,p_m}$
$r_{1,3}$		
LHS Match	RHS Transformation	Kinematic Modeling
$T_a = \chi$ $\deg^+(v_a) = 1$ $F_{R_{a,s_1}} = F_{R_{p_m,a}}$	$T'_a = \emptyset$ $V' = V \cup \{v_{s_2}, v_{s_3}\}$ $E' = E \cup \{e_{a,s_2}, e_{a,s_3}\}$ $L'_V = L_V \cup \{(v_a, \chi), ((v_a), \chi)\}$ $L'_E = L_E \cup \{(F_{L_{a,s_2}}, F_{L_{a,s_1}}), (F_{L_{p_1,a}}, F_{L_{a,s_2}})\}$	$u_1 = ((\alpha_{s_3}, \alpha_p, \alpha_{s_1}), (\rho_p))$ $u_2 = (\alpha_{s_2})$ $u_3 = \left(2\pi - \sum_{i=1}^3 \alpha_{s_i} - \sum_{j=1}^{m-1} \alpha_{p_j}\right)$
<b>Optimization Constraints</b>		<b>Optimization Variables</b>
$U_{\max} \leq U_{\text{med}} + U_{\min}$ $\alpha_{\min} \leq \alpha_{s_2} \leq \alpha_{\max}, \alpha_{\min} \leq \alpha_{s_3} \leq \alpha_{\max}$ $\sum_{i=1}^3 \alpha_{s_i} + \sum_{j=1}^{m-1} \alpha_{p_j} \leq 2\pi - \alpha_{\min}$ $l_{\min} \leq l_{s_2} \leq l_{\min}, l_{\min} \leq l_{s_3} \leq l_{\min}$		$M_a, \alpha_{s_2}, \alpha_{s_3}, l_{a,s_2}, l_{a,s_3}$
		<b>In-plane Coordinates</b>
		$\mathbf{x}_{s_2} = \mathbf{x}_a + l_{a,s_2} \mathbf{R}(\alpha_{s_2}) \mathbf{d}_{a,s_1}$ $\mathbf{x}_{s_3} = \mathbf{x}_a + l_{a,s_3} \mathbf{R}(\alpha_{s_3}) \mathbf{d}_{a,p_1}$

Fig. 44 illustrates the scenarios  $r_{1,4}$ ,  $r_{1,5}$ , and  $r_{1,6}$  that differ from  $r_{1,1}$  in that the outdegree of  $v_a$  on the LHS is  $\deg^+(v_a) = 2$ , in which case there exist two predefined facets in the initial graph  $G_0$  as illustrated in grey in the schematic graph transformations (framed) and only one new vertex is generated. As in Fig. 41 and Fig. 43, all LHSs and RHSs are shown in red while the surrounding graph is depicted in grey.

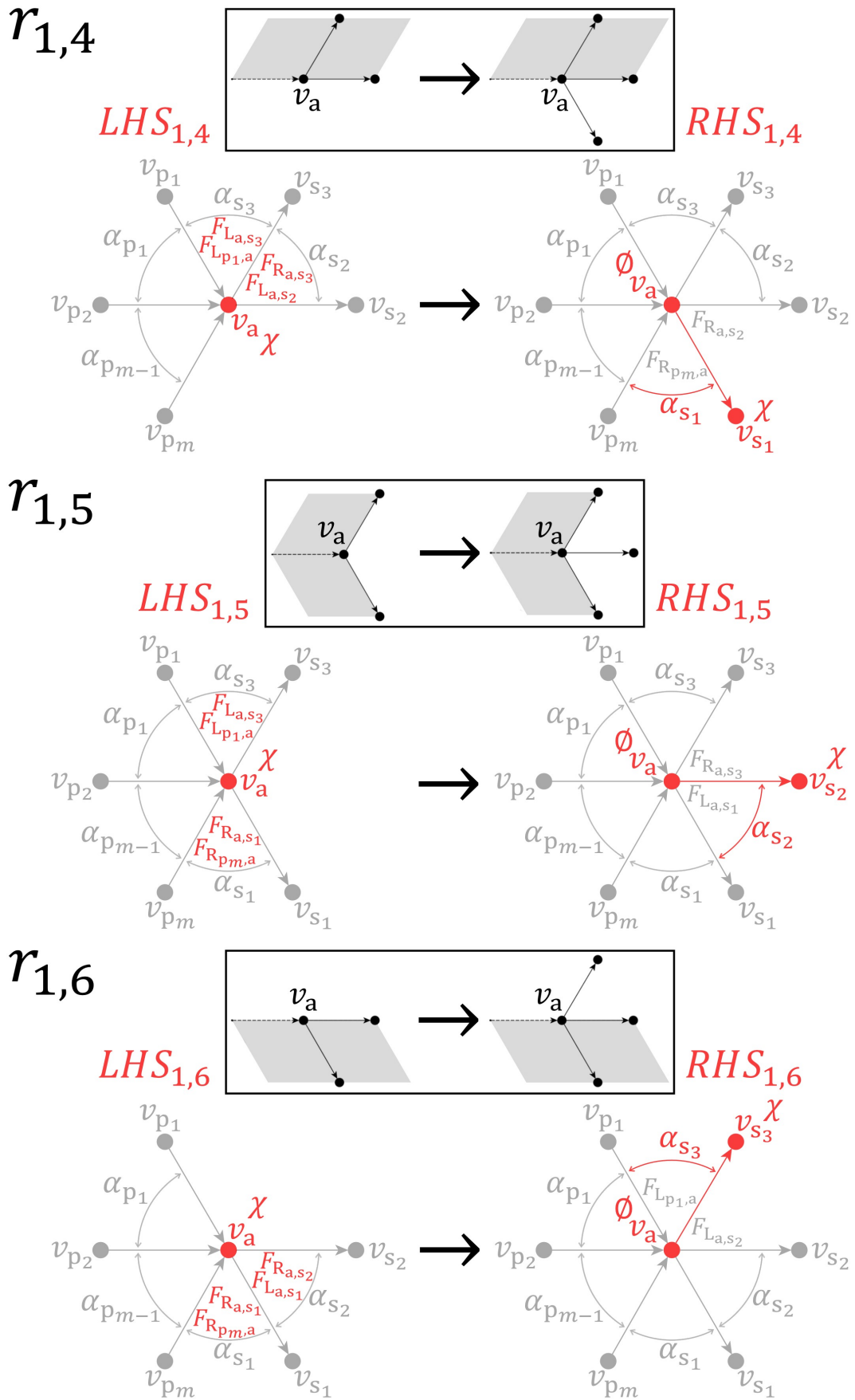


Fig. 44: Scenarios for  $r_{1,4}$ ,  $r_{1,5}$ , and  $r_{1,6}$  in which  $\deg^+(v_a) = 2$  on the  $LHS_{1,4}$ ,  $LHS_{1,5}$ , and  $LHS_{1,6}$  (red) together with the surrounding graph (grey), and schematic graph transformations (framed) where the grey facets are defined in the initial graph  $G_0$

All the necessary information for the rule scenarios  $r_{1,4}$ ,  $r_{1,5}$ , and  $r_{1,6}$  are listed individually in Table 2 that includes the LHS matching and the graph transformations that correspond to the graph topology (highlighted in red) as well as the kinematic modeling, the variables, the constraints, and the coordinates, all of which correspond to the crease pattern geometry and the optimization model. If  $v_a$  has only one predecessor, all terms containing  $\alpha_p$  are neglected.

Table 2: Information of the rule scenarios  $r_{1,4}$ ,  $r_{1,5}$ , and  $r_{1,6}$

$r_{1,4}$		
LHS Match	RHS Transformation	Kinematic Modeling
$T_a = \chi$ $\deg^+(v_a) = 2$ $F_{L_{a,s_3}} = F_{L_{p_1,a}}$ $F_{R_{a,s_3}} = F_{L_{a,s_2}}$	$T'_a = \emptyset$ $V' = V \cup \{v_{s_1}\}$ $E' = E \cup \{e_{a,s_1}\}$ $L'_V = L_V \cup \{(v_a, \chi)\}$ $L'_E = L_E \cup \{(F_{R_{a,s_2}}, F_{R_{p_m,a}})\}$	$u_1 = ((\alpha_{s_3}, \alpha_p, \alpha_{s_1}), (\rho_p))$ $u_2 = \left(2\pi - \sum_{i=1}^3 \alpha_{s_i} - \sum_{j=1}^{m-1} \alpha_{p_j}\right)$ $u_3 = (\alpha_{s_2})$
<b>Optimization Constraints</b>		<b>Optimization Variables</b>
$U_{\max} \leq U_{\text{med}} + U_{\min}$ $\alpha_{\min} \leq \alpha_{s_1} \leq \alpha_{\max}$ $\sum_{i=1}^3 \alpha_{s_i} + \sum_{j=1}^{m-1} \alpha_{p_j} \leq 2\pi - \alpha_{\min}$ $l_{\min} \leq l_{s_1} \leq l_{\min}$		$M_a, \alpha_{s_1}, l_{a,s_1}$ <b>In-plane Coordinates</b> $\mathbf{x}_{s_1} = \mathbf{x}_a + l_{a,s_1} \mathbf{R}(\alpha_{s_1}) \mathbf{d}_{a,p_m}$
$r_{1,5}$		
LHS Match	RHS Transformation	Kinematic Modeling
$T_a = \chi$ $\deg^+(v_a) = 2$ $F_{R_{a,s_1}} = F_{R_{p_m,a}}$ $F_{L_{a,s_3}} = F_{L_{p_1,a}}$	$T'_a = \emptyset$ $V' = V \cup \{v_{s_2}\}$ $E' = E \cup \{e_{a,s_2}\}$ $L'_V = L_V \cup \{(v_a, \chi)\}$ $L'_E = L_E \cup \{(F_{R_{a,s_3}}, F_{L_{a,s_1}})\}$	$u_1 = ((\alpha_{s_3}, \alpha_p, \alpha_{s_1}), (\rho_p))$ $u_2 = (\alpha_{s_2})$ $u_3 = \left(2\pi - \sum_{i=1}^3 \alpha_{s_i} - \sum_{j=1}^{m-1} \alpha_{p_j}\right)$
<b>Optimization Constraints</b>		<b>Optimization Variables</b>
$U_{\max} \leq U_{\text{med}} + U_{\min}$ $\alpha_{\min} \leq \alpha_{s_2} \leq \alpha_{\max}$ $\sum_{i=1}^3 \alpha_{s_i} + \sum_{j=1}^{m-1} \alpha_{p_j} \leq 2\pi - \alpha_{\min}$ $l_{\min} \leq l_{s_2} \leq l_{\min}$		$M_a, \alpha_{s_2}, l_{a,s_2}$ <b>Coordinates</b> $\mathbf{x}_{s_2} = \mathbf{x}_a + l_{a,s_2} \mathbf{R}(\alpha_{s_2}) \mathbf{d}_{a,s_1}$
$r_{1,6}$		
LHS Match	RHS Transformation	Kinematic Modeling
$T_a = \chi$ $\deg^+(v_a) = 2$ $F_{R_{a,s_1}} = F_{R_{p_m,a}}$ $F_{L_{a,s_1}} = F_{R_{a,s_2}}$	$T'_a = \emptyset$ $V' = V \cup \{v_{s_3}\}$ $E' = E \cup \{e_{a,s_3}\}$ $L'_V = L_V \cup \{(v_a, \chi)\}$ $L'_E = L_E \cup \{(F_{L_{p_1,a}}, F_{L_{a,s_2}})\}$	$u_1 = ((\alpha_{s_3}, \alpha_p, \alpha_{s_1}), (\rho_p))$ $u_2 = (\alpha_{s_2})$ $u_3 = \left(2\pi - \sum_{i=1}^3 \alpha_{s_i} - \sum_{j=1}^{m-1} \alpha_{p_j}\right)$
<b>Optimization Constraints</b>		<b>Optimization Variables</b>
$U_{\max} \leq U_{\text{med}} + U_{\min}$ $\alpha_{\min} \leq \alpha_{s_3} \leq \alpha_{\max}$ $\sum_{i=1}^3 \alpha_{s_i} + \sum_{j=1}^{m-1} \alpha_{p_j} \leq 2\pi - \alpha_{\min}$ $l_{\min} \leq l_{s_3} \leq l_{\min}$		$M_a, \alpha_{s_3}, l_{a,s_3}$ <b>In-plane Coordinates</b> $\mathbf{x}_{s_3} = \mathbf{x}_a + l_{a,s_3} \mathbf{R}(-\alpha_{s_3}) \mathbf{d}_{a,p_1}$

### 5.2.2 Rule $r_2$ : Combine Vertices

The second rule  $r_2$  is defined as a production of the type  $r_2: LHS_2 \rightarrow RHS_2$ , where  $LHS_2$  and  $RHS_2$  are both colored in red and the surrounding graph is shown in grey in Fig. 45 that also includes a framed, schematic graph transformation.

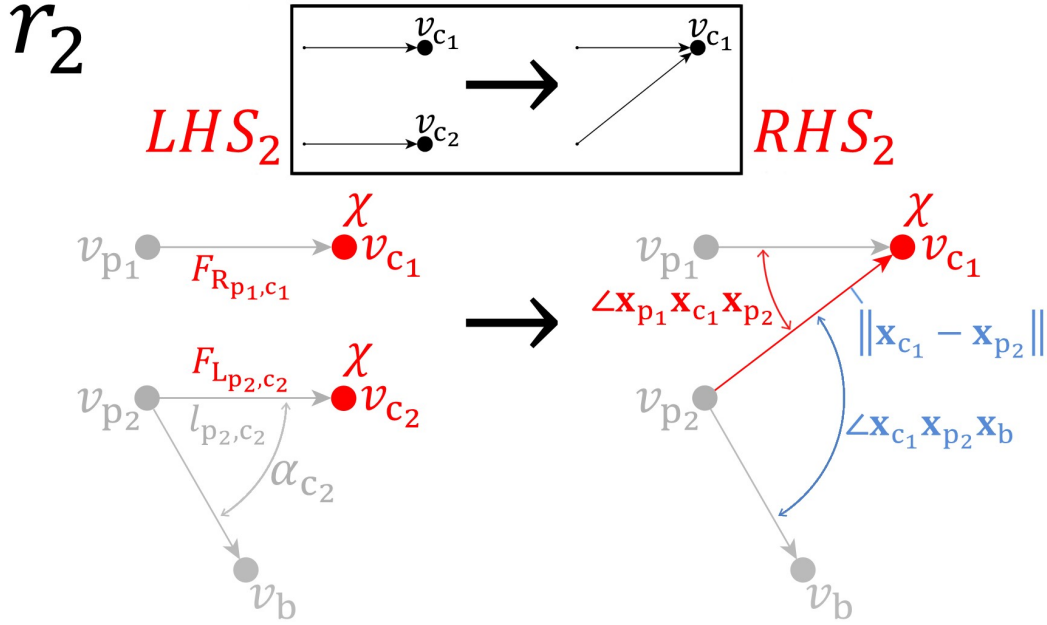


Fig. 45: Rule  $r_2$  combines two vertices on its  $LHS_2$  into a single vertex on its  $RHS_2$ . Top: Schematic graph transformation (framed). Bottom: Parts shown in red belong to the  $LHS_2$  and the  $RHS_2$ , the surrounding graph is depicted in grey, and adjustments to the constraint system are shown in blue.

Rule  $r_2$  accepts two vertices  $v_{c1}$  and  $v_{c2}$  that share an edge label on the  $LHS_2$  and combines them into a single vertex  $v_{c1}$  on the  $RHS_2$ . The incoming edges formerly incident to  $v_{c2}$  are reassigned to  $v_{c1}$  and a new constraint is introduced between the reassigned edges and the edges incident to  $v_{c1}$ . Then,  $r_2$  adjusts the set of optimization variables and the set of optimization constraints by replacing all instances of the lengths and sector angles corresponding to the reassigned edges, such as  $l_{p2,c2}$  and  $\alpha_{c2}$  in Fig. 45 left, respectively, with the parts colored in blue in Fig. 45 right. The vertex  $v_b$  signifies any vertex that was coupled to the sector angles ( $\alpha_{c2}$ ) of the reassigned edge.

#### 5.2.2.1 LHS Matching of $r_2$

A match  $\mathcal{M}_2$  of the  $LHS_2$  in  $G$  involves four conditions. Both vertices  $v_{c1}$  and  $v_{c2}$  have to be extendable,  $T_{c1} = T_{c2} = \chi$ . This condition guarantees the generation of acyclic graphs since no vertex can be combined with any of its predecessors.

The second condition requires that both vertices  $v_{c1}$  and  $v_{c2}$  belong to the same facet. This can be checked by comparing the edge labels of the incoming edges incident to  $v_{c1}$  and  $v_{c2}$ , which is shown

in Fig. 45 on the  $LHS_2$  where  $F_{R_{p_1,c_1}} = F_{L_{p_2,c_2}}$  has to be satisfied. Note that while Fig. 45 illustrates the case in which both vertices  $v_{c_1}$  and  $v_{c_2}$  each exhibit one predecessor, multiple predecessors to both vertices are possible. In this case, the method compares the edge labels of the first and last predecessors of both vertices. Identifying these first and last predecessors is straightforward since the predecessors of a vertex are always ordered in the counter-clockwise direction as explained for rule  $r_1$ .

To prevent the combination of vertices that share the same predecessor, such as  $v_{s_{1-3}}$  in Fig. 41, the third condition states that  $v_{c_1}$  and  $v_{c_2}$  cannot exhibit the same predecessors,  $P_{c_1} \cap P_{c_2} = \{\emptyset\}$ .

Since  $r_2$  deletes a vertex, the fourth condition states that  $v_{c_1}$  and  $v_{c_2}$  cannot both be contained in the initial graph, which can be assessed by checking if their in-plane coordinates are dependent on any optimization variables. If one vertex is contained within the initial graph, it becomes the “dominant” vertex that is not deleted, and if neither belongs to the initial graph then either vertex can be eliminated. The following graph transformation describes the generic case in which  $v_{c_1}$  is dominant.

### 5.2.2.2 Graph Transformation of $r_2$

Once  $r_2$  is applied,  $G \xrightarrow{r_2} G''$ , the  $RHS_2$  includes  $v_{c_1}$  while  $v_{c_2}$  is subtracted from the set of vertices,  $V'' = V \setminus \{v_{c_2}\}$ . The edge formerly incident to  $v_{c_2}$  is subtracted from the set of edges and a new edge is introduced from the predecessors of  $v_{c_2}$  to  $v_{c_1}$ ,  $E'' = (E \setminus \{e_{p_2,c_2}\}) \cup \{e_{p_2,c_1}\}$ . The predecessors of both vertices have to be united in the correct order such that the predecessors of the vertex contributing the right side of the edge label ( $F_{R_{p_1,c_1}}$  in Fig. 45) go first. In addition, vertex  $v_{c_1}$  on the  $RHS_2$  can still be extended, resulting in the vertex label  $L''_{c_1} = ((v_{p_1}, v_{p_2}), \chi)$ . The edge label  $L_{c_2}$  is then subtracted from the set of vertex labels,  $L''_V = L_V \setminus \{L_{c_2}\}$ . The edge label formerly corresponding to the edge incident to  $v_{c_2}$  is transferred identically to the reassigned edge,  $L''_{p_2,c_1} = L_{p_2,c_2}$ . Finally, the edge label of the edge incident to  $v_{c_2}$  is subtracted from the set of edge labels,  $L''_E = L_E \setminus \{L_{p_2,c_2}\}$ . If  $v_{c_2}$  is incident to multiple incoming edges on the  $LHS_2$ , the above procedure is applied to all edges and edge labels.

### 5.2.2.3 Coordinates and Optimization Variables of $r_2$

The in-plane location and thus the coordinates  $\mathbf{x}_j$  of a vertex  $v_j$  are always determined by a single sector angle  $\alpha_j$  and a single edge length  $l_{p_i,j}$ . Since  $r_2$  deletes the vertex  $v_{c_2}$ , the corresponding sector angle and edge lengths of  $v_{c_2}$  are eliminated from the set of optimization variables,  $\Phi'' = \Phi \setminus \{\alpha_{c_2}, l_{p_2,c_2}\}$ .



#### 5.2.2.4 Optimization Constraints of $r_2$

Independent of the number of predecessors in  $P_{c_1} \cup P_{c_2}$ ,  $r_2$  introduces one new constraint (shown in Fig. 45 in red) that corresponds to the sector angle between the edges incident to  $v_{p_1}$  and  $v_{p_2}$ . Since this sector angle is dependent on the locations of the vertices involved, it needs to be expressed as an angle with respect to the in-plane coordinates,  $\angle \mathbf{x}_{p_1} \mathbf{x}_{c_1} \mathbf{x}_{p_2}$ . The constraint introduced by  $r_2$  applies to the upper and lower bounds equivalent to Eq. (31):

$$\alpha_{\min} \leq \angle \mathbf{x}_{p_1} \mathbf{x}_{c_1} \mathbf{x}_{p_2} \leq \alpha_{\max} \quad (34)$$

Eq. (34) is then added to the set of optimization constraints  $\psi$ .

Rule  $r_2$  eliminates the sector angle and the edge lengths that formerly determined the coordinates of  $v_{c_2}$ , but the constraints that apply to these variables are still valid and present in the set of constraints  $\psi$ . Thus,  $r_2$  expresses all instances of these variables in terms of their in-plane coordinates (Fig. 45, blue) within  $\psi$ . Variable  $\alpha_{c_2}$  is replaced with  $\angle \mathbf{x}_{c_1} \mathbf{x}_{p_2} \mathbf{x}_b$  and  $l_{p_2, c_2}$  is replaced with the Euclidean distance  $\|\mathbf{x}_{c_1} - \mathbf{x}_{p_2}\|$  between  $v_{p_2}$  and  $v_{c_1}$ . Hence,

$$\psi'' = \psi(\alpha_{c_2} \leftarrow \angle \mathbf{x}_{c_1} \mathbf{x}_{p_2} \mathbf{x}_b, l_{p_2, c_2} \leftarrow \|\mathbf{x}_{c_1} - \mathbf{x}_{p_2}\|) \quad (35)$$

When  $v_{c_2}$  has multiple predecessors on the  $LHS_2$ , this replacement procedure needs to be performed analogously for all reassigned edges. This procedure also applies when the sector angles and edge lengths are already expressed in terms of the in-plane coordinates, i.e. when  $r_2$  has already been applied to  $v_{c_2}$ .

#### 5.2.3 Automated Graph Generation and Filtering

The rule set  $\mathcal{R} = (r_1, r_2)$  is designed to comply with the guidelines for the generation of kinematically determinate and acyclic crease pattern graphs presented in Section 4, where the rules are implicitly applied in manual generation processes. Together, the rules  $r_1$  and  $r_2$  are thus able to generate the rigid flasher (Fig. 32), the star pattern (Fig. 34), and the origami chair pattern (Fig. 37b), and can further be utilized to generate other existing crease patterns such as slender origami [122], origami strings [123], or the gripper in Ref. [3]. However, the manual design process in Section 4 is a trial-and-error procedure, which is why the application of the graph grammar  $GG$  needs to be automated.

To automate the generation step (Fig. 39), the method applies the two rules whenever a respective LHS match  $\mathcal{M}_1$  or  $\mathcal{M}_2$  is found and enumerates all possible rule application sequences that arise from the initial graph  $G_0$  and the maximum number of internal vertices  $N_{\max}$ .

However, distinct rule application sequences do not guarantee distinct graphs [126], which is why the method checks the generated crease pattern graphs for redundancy and filters them before they are subjected to the optimization. For generic design tasks, the method filters graphs based on isomorphism while filters specific to a design task have to be introduced manually if required. The filtering of isomorphic graphs is described here, while two additional filters specific to the design tasks of the gripper and the robotic arm are explained in Section 5.5.2.

For the rule system presented, two graphs are isomorphic if they exhibit an edge-preserving vertex bijection. If this condition is satisfied, the set of vertex labels  $L_V$  is guaranteed to coincide, whereas the set of edge labels  $L_E$  does not play a role in terms of graph morphology (i.e. there is no difference in kinematics when facets are numbered differently). Isomorphism checks are implemented in many standard programs, and the method uses the function *IsomorphicGraphQ* integrated in *Mathematica 11*.

Both the automated generation of crease pattern graphs and the isomorphism check are listed in the pseudocode in Table 3. The input to the generation step are the rule set  $\mathcal{R} = (r_1, r_2)$  as well as the user-defined initial graph  $G_0$  and the maximum number of internal vertices  $N_{\max}$ . The output comprises two parts, the first of which corresponds to the *language*  $\mathcal{L}$  of the graph grammar  $GG$ . The language  $\mathcal{L}(GG)$  is comprised of the crease pattern graphs  $G$  in the set of all possible crease pattern graphs  $\mathcal{G}$  labeled over  $\emptyset$  that can be generated by applying the rule set  $\mathcal{R}$  to the initial graph  $G_0$ ,  $\mathcal{L}(GG) = \{G \in \mathcal{G}_\emptyset | G_0 \xrightarrow{\mathcal{R}} G\}$ . The second part of the output corresponds to the optimization variables  $\Phi$  and constraints  $\psi$  for all generated graphs  $G \in \mathcal{G}_\emptyset$ , the collection of which enables the automated optimization of the design alternatives in the subsequent evaluation step of the computational method.

To achieve this output, the method first initializes a set of graphs to be assessed  $\mathcal{G} = \{G_0\}$  and an empty set for all possible crease pattern graphs  $\mathcal{G}_\emptyset = \{\emptyset\}$  in Step 1 of Table 3. Then, the method steps into a loop that is terminated only when all graphs have been assessed,  $\mathcal{G} = \{\emptyset\}$  (Step 2). In Step 3, the intermediate set of graphs  $\mathcal{G}_1$  is introduced,  $\mathcal{G}_1 = \mathcal{G}$ , and the graphs to assess are emptied,  $\mathcal{G} = \{\emptyset\}$ , after which the method iterates through all graphs  $G$  in  $\mathcal{G}_1$  (Step 4). In Step 5, a graph  $G$  is subjected to the isomorphism check and the following procedure is only applied if  $G \notin \mathcal{G}_\emptyset$ . By applying the conditions for the LHS matches to all vertices within  $G$ , the method then finds all matches  $\mathcal{M}_1$  and  $\mathcal{M}_2$  (Step 6). For each match  $\mathcal{M}_1$  (Step 7), rule  $r_1$  is applied only if the number of internal vertices  $N$  of  $G$  is smaller than the maximum number of internal vertices,  $N(G) < N_{\max}$  (Step 8). If this check is successful,

Step 9 applies  $r_1$ ,  $G \xRightarrow{r_1} G'$ , resulting in the graph transformations and the kinematic modeling of the respective extended vertex  $v_a$  as well as in the updating of the optimization variables  $\Phi(G')$  and constraints  $\psi(G')$ . These variables and constraints are then collected in Step 10 for the subsequent optimization, and the newly generated graph  $G'$  is added to the set of graphs to be assessed  $\mathcal{G}$  in Step 11. For each match  $\mathcal{M}_2$  found in Step 6 (Step 12), rule  $r_2$  is applied in Step 13 to result in a graph  $G''$ ,  $G \xRightarrow{r_2} G''$ . Again, the rule application automatically performs the graph transformation and the adjustments to the kinematic model, and Step 14 collects the optimization variables  $\Phi(G'')$  and constraints  $\psi(G'')$  for the subsequent optimization of  $G''$ . In Step 15, the newly generated graph  $G''$  is added to the set of graphs  $\mathcal{G}$  that are assessed in the next iteration of the pseudocode, and Step 16 finally adds the graph  $G$  to the set of all possible graphs  $\mathcal{G}_\emptyset$ .

Table 3: Pseudo-code of the automated generation of crease pattern graphs and the collection of optimization variables  $\Phi$  and constraints  $\psi$  for the subsequent optimization

<b>Input:</b>	Initial graph $G_0$ , rules $\mathcal{R} = (r_1, r_2)$ , max. number of internal vertices $N_{\max}$
<b>Output:</b>	The entire language $\mathcal{L}(GG) = \{G \in \mathcal{G}_\emptyset   G_0 \xRightarrow{\mathcal{R}} G\}$ Sets of optimization variables $\Phi$ and constraints $\psi$ for all $G \in \mathcal{G}_\emptyset$
<b>1:</b>	<b>Initialize</b> $\mathcal{G} = \{G_0\}; \mathcal{G}_\emptyset = \{\emptyset\};$
<b>2:</b>	<b>While</b> $\mathcal{G} \neq \{\emptyset\}$
<b>3:</b>	<b>Set</b> $\mathcal{G}_1 = \mathcal{G}$ and $\mathcal{G} = \{\emptyset\};$
<b>4:</b>	<b>For each</b> $G \in \mathcal{G}_1;$
<b>5:</b>	<b>If</b> $G \notin \mathcal{G}_\emptyset$ <span style="float: right; color: #00AEEF;">Isomorphism check</span>
<b>6:</b>	<b>Find</b> all matches $\mathcal{M}_1: LHS_1 \rightarrow G$ and $\mathcal{M}_2: LHS_2 \rightarrow G$
<b>7:</b>	<b>For each</b> match $\mathcal{M}_1$
<b>8:</b>	<b>If</b> $N(G) < N_{\max}$ <span style="float: right; color: #00AEEF;">Termination criterion</span>
<b>9:</b>	<b>Apply</b> $r_1, G \xRightarrow{r_1} G';$ <span style="float: right; color: #00AEEF;">Graph transformation and kin. modeling</span>
<b>10:</b>	<b>Collect</b> sets of optimization variables $\Phi(G')$ and constraints $\psi(G')$
<b>11:</b>	<b>Add</b> $G'$ to $\mathcal{G};$
<b>12:</b>	<b>For each</b> match $\mathcal{M}_2$
<b>13:</b>	<b>Apply</b> $r_2, G \xRightarrow{r_2} G'';$ <span style="float: right; color: #00AEEF;">Graph transformation and kin. modeling</span>
<b>14:</b>	<b>Collect</b> sets of optimization variables $\Phi(G'')$ and constraints $\psi(G'')$
<b>15:</b>	<b>Append</b> $G''$ to $\mathcal{G};$
<b>16:</b>	<b>Add</b> $G$ to $\mathcal{G}_\emptyset;$

Hence, the method automatically generates the entire language of all possible crease pattern graphs  $\mathcal{L}(GG) = \{G \in \mathcal{G}_\emptyset | G_0 \xRightarrow{\mathcal{R}} G\}$  that only contains nonisomorphic graphs  $G$  as well as the optimization variables  $\Phi$  and constraints  $\psi$  for all  $G \in \mathcal{G}_\emptyset$ . As shown in Fig. 39, after the generation step all graphs  $G$  are forwarded to the guidance step.

### 5.3 Guidance

The RBMs  $M$  contained in the set of the optimization variables  $\Phi$  play a special role since every  $M$  only adopts the discrete states “up” and “down”, respectively. Each internal (or extended) vertex in a

graph  $G$  contributes one RBM, which results in  $2^N$  different design alternatives for a crease pattern with  $N$  internal vertices. For the lack of more knowledge about the search space associated with the RBM, the method enumerates all possible RBM assignments and subjects each alternative to the evaluation.

## 5.4 Evaluation

As illustrated in Fig. 39, the evaluation includes both the optimization of a crease pattern to assess the kinematic behavior optimized with respect to the design task as well as an intersection check. While all design alternatives are optimized, only the alternatives that meet the design criteria are subjected to the intersection check.

### 5.4.1 Optimization

Much of what is required for the optimization is intrinsic to the rule system. By applying the rules, the method automatically generates the set of optimization variables  $\Phi$  and constraints  $\psi$  of each graph  $G$ , both of which are expressed symbolically. What remains to be defined by the user to run the optimization is a design task provided in the input to the computational method involving an objective function  $\Omega$  and the numerical values for the variable boundaries  $l_{\min}$ ,  $l_{\max}$ ,  $\alpha_{\min}$ , and  $\alpha_{\max}$ . The objective function can include and should be dependent on the dihedral angles  $\rho_{i,j}$  or the three-dimensional vertex coordinates  $\mathbf{X}_i$ . In practice, a user defines one or multiple states  $t = t_{\text{opt}} \in [-\pi, \pi]$  at which the dihedral angles  $\rho_{i,j}(t_{\text{opt}})$  or the vertex coordinates  $\mathbf{X}_i(t_{\text{opt}})$  should equal certain values or locations, respectively. The definition of the objective function for the design tasks is given in Section 5.5.3.

Depending on the objective function  $\Omega$ , the proposed method adjusts the set of optimization variables as a last step before the optimization. For an origami crease pattern, the lengths of the edges that are incident to degree-1 vertices on the border of the paper do not influence the kinematics of the crease pattern. Thus, all variables corresponding to the edge lengths  $l_{i,j}$  that are present neither in the objective function nor in the set of constraints  $\psi$  are discarded from the set of optimization variables  $\Phi$ , resulting in an adjusted set  $\Phi_{\text{opt}}$ . For the subsequent representation of the origami, the discarded edge lengths in the set of three-dimensional vertex coordinates are substituted with the value of the minimum edge length,  $X \leftarrow X(l_{i,j} \leftarrow l_{\min})$ .

This procedure results in the following optimization scheme to optimize the geometry of all design alternatives:

$$\begin{aligned} & \min_{\Phi_{\text{opt}}} \Omega \\ & \text{s. t. } \psi \end{aligned} \tag{36}$$

Equation (36) is then solved by the function *NMinimize* integrated into *Mathematica 11*, which employs global numerical solvers to find a set of variables  $\Phi_{\text{opt}}^*$  that optimizes the objective function such that  $\Omega(\Phi_{\text{opt}}^*) = \Omega^*$ . *NMinimize* is applied with default settings in which the nonlinear optimization problem is reformulated as an unconstrained problem using penalty functions and then solved with Differential Evolution [127].

#### 5.4.2 Intersection

If an optimized design alternative satisfies  $\Omega^* \leq \vartheta$ , where  $\vartheta$  is the target criteria provided by the user in the design task, the design alternative is subjected to an intersection check that requires the method to first transform the set of optimized variables  $\Phi_{\text{opt}}^*$  into an origami with facets. To do so, the optimized variables are first substituted into the set of three-dimensional vertex coordinates  $X^* = X(\Phi_{\text{opt}} \leftarrow \Phi_{\text{opt}}^*)$  that is then only dependent on  $t$ . To find the vertices that constitute a facet, the method iterates through all edges and for each edge assigns both incident vertices to the facets contained in the respective edge label. This procedure results in sets of vertices that represent the facets, and the method assigns to each of these sets a polygon. By associating each vertex  $v_i$  with its three-dimensional coordinates  $X_i^*$ , a design alternative is represented as a set of polygons that move with respect to  $t$ .

Subsequently, the design alternative is subjected to the same intersection check described in Section 3.2.4. Since the intersection check is purely numerical, the method performs the check 16 times throughout the folding motion from  $t = 0$  to  $t = t_{\text{max}}$  in even intervals. The specific number of checks is determined empirically to achieve an appropriate trade-off between a low time-consumption and a reliable intersection check but can be adapted based on the user's needs or the design task.

#### 5.5 Design Tasks and Application of the Method

In this section, the proposed method is applied to two design tasks that include a gripper and a robotic arm. The principle of rigid origami has been employed for gripping tasks before, such as in the Oriceps [25] or in Ref. [3]. However, both of these existing origami grippers were designed by hand and for a design task that only involves gripping; to display the usefulness of the proposed method, this work additionally introduces an obstacle that lies between the crease pattern and the point object that the final origami should be able to grip (Fig. 46a). In the robotic arm task, not one but 11 point objects are approximated so that the tip of the arm follows a given trajectory (Fig. 46b), equivalent to the path generation synthesis of spatial mechanisms [128]. To the knowledge of the author, origami has not been applied to the approximation of trajectories, opening up a new domain for the application of origami.

The section is structured according to the workflow in Fig. 39: it first describes the input, then the two additional filters to check for redundant graphs within the generation step, after which it presents the optimization and shows the results produced by the method. Since both design tasks of the gripper and the robotic arm are closely related, the following sections are only partitioned when necessary.

### 5.5.1 Input

The design tasks of the gripper and the robotic arm are illustrated in Fig. 46a and Fig. 46b, respectively. The initial graph  $G_0$  (identical for both tasks) with vertices  $v_1$  and  $v_2$  and edge  $e_{1,2}$  is shown in black, and corresponds to the graph described in Fig. 40 with the same vertex labels  $L_1 = ((\emptyset), \emptyset)$  and  $L_2 = ((v_1), \chi)$ , edge label  $L_{1,2} = (f_1, f_2)$ , as well as in-plane coordinates  $\mathbf{x}_1 = (0, 0)$  and  $\mathbf{x}_2 = (1, 0)$ . The driving angle of the edge  $e_{1,2}$  in both cases is linear,  $\rho_{1,2} = t$ , where  $t$  goes from zero to  $t_{\max} = \frac{\pi}{2}$ , applied mirror-symmetrically with respect to the  $xz$ -plane. The actuation thus corresponds to a single DOF, and the maximum number of internal vertices for both design tasks is set to  $N_{\max} = 3$ .

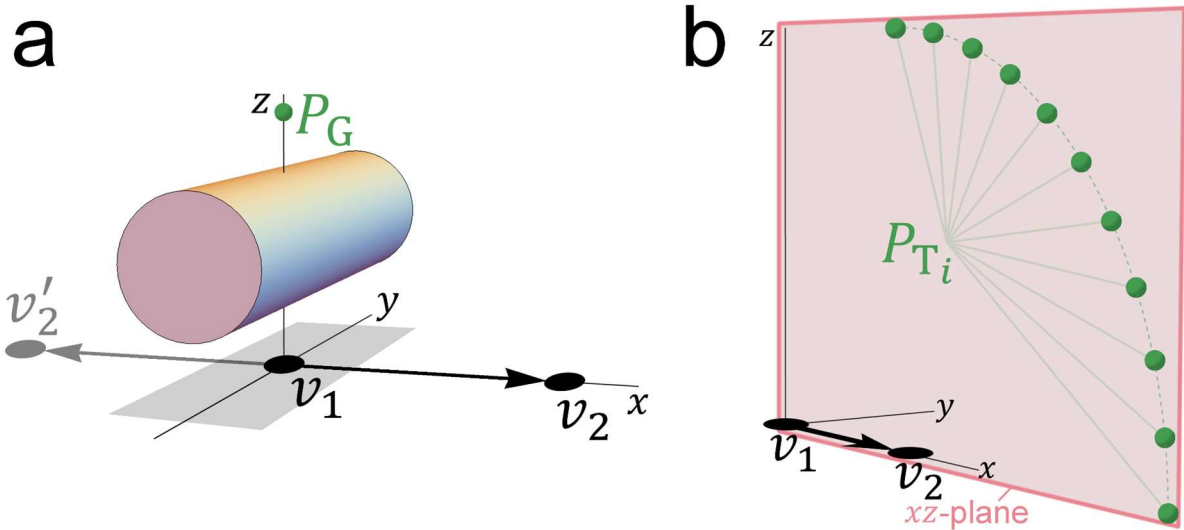


Fig. 46: (a) The gripper design task with the point to grip  $P_G = (0, 0, 1)$  and the cylinder obstacle with radius 0.25 whose axis is coincident with the line segment going from  $(0, -1, 0.5)$  to  $(0, 1, 0.5)$ . The vertex  $v'_2$  shown in grey illustrates that the crease pattern graph generated with the initial graph is rotated after the optimization to result in a rotationally symmetric gripper. (b) The robotic arm design task with all points  $P_{T_i}$  evenly distributed on the given trajectory that lies on the  $xz$ -plane. Both gripper (a) and robotic arm (b) start with the same initial graph  $G_0$  shown in black corresponding to the graph described in Fig. 40 with  $v_1$  and  $v_2$  located at  $\mathbf{x}_1 = (0, 0)$  and  $\mathbf{x}_2 = (1, 0)$ .

The point object  $P_G$  to grip in Fig. 46a is located at  $P_G = (0, 0, 1)$  and the obstacle is represented by a cylinder with radius 0.25 whose axis is coincident with the line segment going from  $(0, -1, 0.5)$  to  $(0, 1, 0.5)$ . This setup allows for the generation of only one side of the gripper that emerges from the initial graph shown in black, after which the arm is rotated, once optimized, by  $180^\circ$  around the  $z$ -axis, which is illustrated in Fig. 46a by the graph in grey corresponding to  $v'_2$ .

Fig. 46b shows the trajectory to approximate and the 11 points  $P_{T_i}$  located at  $(1,0,0) + (2 \cos t_i, 0, 2 \sin t_i)$  where  $t_i$  goes from 0 to  $\frac{\pi}{2}$  in steps of  $\frac{\pi}{20}$ .

### 5.5.2 Generation: Additional Filters

In addition to the filter based on isomorphism, two additional filters are implemented in Step 5 of Table 3 for both design tasks to further reduce the number of distinct crease pattern graphs generated by the method.

To facilitate the description of a crease pattern graph, here the following notation is introduced for the rule application sequences: when a vertex  $v_i$  is extended by  $r_1$ ,  $i$  is added to the sequence, and when vertices  $v_i$  and  $v_j$  are combined by  $r_2$ ,  $(i, j)$  is added to the sequence. As an example, extending vertices  $v_2$  and  $v_3$  in the given order and then combining vertices  $v_4$  and  $v_8$  yields a sequence denoted as 2, 3, (4, 8).

Since both design tasks illustrated in Fig. 46 are mirror symmetric with respect to the  $xz$ -plane, a graph can be filtered if it is mirror symmetric to another graph with respect to the  $x$ -axis. An example for the two symmetric graphs 2, 3 and 2, 5 is given in Fig. 47a. These two graphs will result in the same (symmetric) optimized configuration, which renders one of the graphs redundant. In addition, any rule applied to, e.g., 2, 5 will generate a graph that is symmetric to a graph that can be generated by applying the corresponding rule to 2, 3. Thus, in both design tasks the method filters all symmetric graphs and their *descendants* that result from applying more rules to symmetric graphs.

The second additional filter for both problems results from the task that the origami concept must perform. One of the generated vertices contained in the graphs has to approximate the point to grip  $P_G$  in the gripper task or the trajectory points  $P_{T_i}$  in the robotic arm task. For simplicity, such a vertex is called the *gripping vertex* for both tasks. Extending a vertex by  $r_1$  only adds value to the resulting graph, not its descendants, if one of its successors becomes the gripping vertex, which is why only the successors of the vertex extended last can perform the gripping task. Fig. 47b shows the example of a crease pattern graph 2, 4, 3 that contains unnecessary vertices and edges, here called a *semantically invalid* graph [129]. Since  $v_3$  is the vertex extended last,  $v_9$ ,  $v_{10}$ , or  $v_{11}$  are the gripping vertex candidates. In this case, extending  $v_4$  was semantically invalid since all of its successors  $v_6$ ,  $v_7$ , and  $v_8$  can be cut from the graph without changing the performance of the origami gripper. In contrast to symmetric graphs, however, the descendants of semantically invalid graphs can lead to useful crease pattern graphs, which is why their descendants are not filtered.

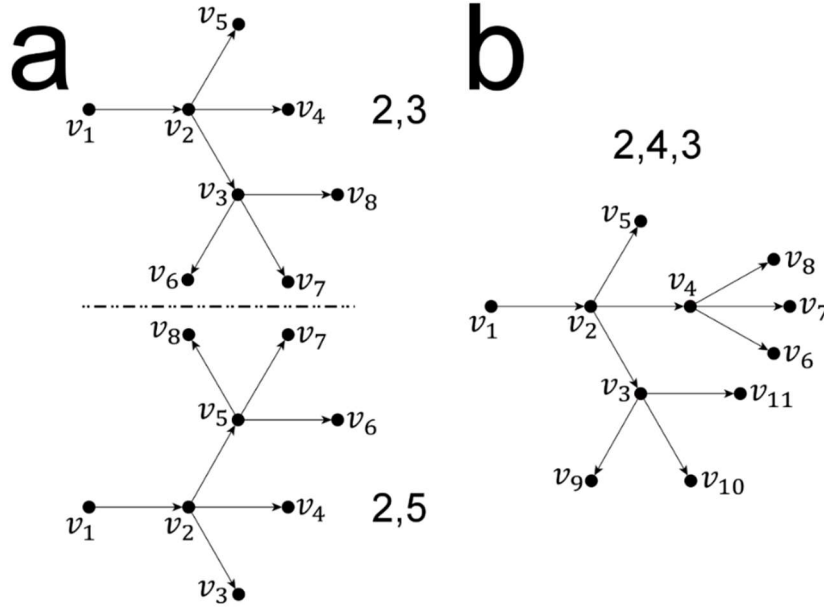


Fig. 47: (a) Two crease pattern graphs 2,3 and 2,5 that are symmetric to each other with respect to the  $x$ -axis. Both graphs and their descendants will result in the same optimized configuration, which is why either of the graphs and its respective descendants can be eliminated. (b) A crease pattern graph 2,4,3 that is semantically invalid.

### 5.5.3 Optimization

As explained in the last section, all successors of the vertex extended last are candidates to grip  $P_G$  or approximate  $P_{T_i}$ , which is why each design alternative is optimized three successive times with a different gripping vertex. These successors of the vertex extended last are denoted as  $v_{l,j}$  and the respective three-dimensional coordinates are denoted as  $\mathbf{X}_{l,j}(t)$  with  $j = 1, 2, 3$ .

Then, the objective function for each optimization in the gripper design task is the Euclidean distance between  $P_G$  and the location of the gripping vertex at  $t_{\text{opt}} = t_{\text{max}} = \frac{\pi}{2}$ :

$$\Omega = \left\| P_G - \mathbf{X}_{l,j} \left( \frac{\pi}{2} \right) \right\| \quad (37)$$

An optimization of the gripper is considered successful if the objective value  $\Omega^*$  of a design alternative is smaller than or equal to  $\vartheta = 10^{-5}$ .

In the robotic arm task, each optimization minimizes the sum of the Euclidean distance between all trajectory points  $P_{T_i}$  and the respective locations of the gripping vertex at  $t_{\text{opt}} = t_i = (0, \frac{\pi}{20}, \dots, \frac{9\pi}{20}, \frac{\pi}{2})$ :

$$\Omega = \sum_{i=1}^{11} \left\| P_{T_i} - \mathbf{X}_{l,j}(t_i) \right\| \quad (38)$$

An optimization of the robotic arm is considered successful if the objective value  $\Omega^*$  of a design alternative is smaller than or equal to  $\vartheta = 3 * 10^{-1}$ .

The variable boundaries for both tasks are  $l_{\text{min}} = 0.1$ ,  $l_{\text{max}} = 1.5$ ,  $\alpha_{\text{min}} = \frac{\pi}{18}$ , and  $\alpha_{\text{max}} = \frac{17\pi}{18}$ .



## 5.6 Results

With the initial graph  $G_0$  in Fig. 46,  $N_{\max} = 3$ , and rules  $r_1$  and  $r_2$ , the graph grammar  $GG$  generates a total of 291 crease pattern graphs for both design tasks. After filtering isomorphic, symmetric, and semantically invalid graphs, 52 of the original 291 graphs remain. The design space of all possible crease pattern topologies and rule application sequences for both design tasks is visualized in the search tree in Fig. 48. The nodes colored in yellow illustrate the 52 meaningful graphs generated, whereas both the symmetric (sym.) as well as the semantically invalid (inv.) graphs are illustrated with white nodes. Each level of the search tree contains the graphs that exhibit the same number of applied rules starting from one application at the top to seven applications at the bottom.

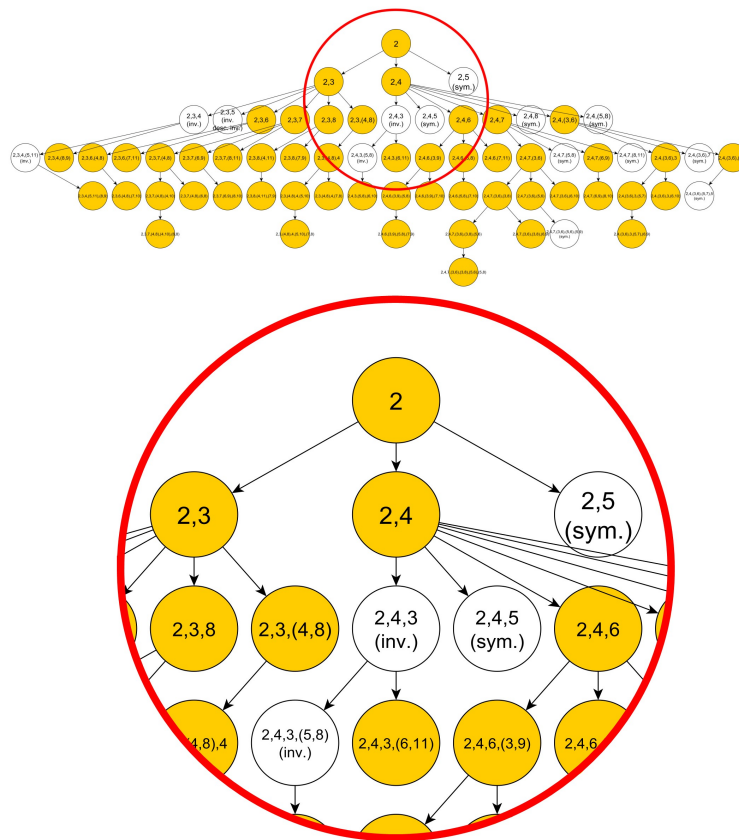


Fig. 48: Top: Search tree with 52 meaningful graphs shown as yellow nodes as well as symmetric (sym.) and semantically invalid (inv.) graphs shown as white nodes, where each level of the search tree corresponds to the same number of applied rules. Bottom: A scaled up section of the search tree shows the effect of the filtering after which the descendants of symmetric graphs are completely removed, while the descendants of the semantically invalid graphs can result in meaningful graphs. This search tree corresponds to both the gripper and the robotic arm design task.

The bottom of Fig. 48 shows a scaled up section of the search tree that contains the graphs generated by the application sequences 2, 3, 2, 5, and 2, 4, 3 shown in Fig. 47. As explained before, the descendants of symmetric graphs such as 2, 5 are filtered, while the descendants of semantically invalid graphs can be meaningful, such as 2, 4, 3, (6, 11).

From the 52 distinct graphs there emerge 1170 distinct design alternatives (and as many objective function evaluations) that are comprised of the different assignments of RBMs and gripping vertices. The optimization of a single design and the intersection check of a successful design takes from a few seconds to minutes depending on the complexity of the graph. The enumeration of the whole design space takes about 35 hours for the gripper task and 40 hours for the robotic arm task on an Intel i7 processor with 8GB RAM.

With respect to the gripper, 836 of the 1170 design alternatives are able to grip the target point  $P_G$ , and thereof, 148 alternatives do not self-intersect. When the cylinder is present, the number of feasible origami concepts reduces to 36. The gripping motion of three such feasible origami concepts is depicted in Fig. 49 at discrete folding states from left to right. Fig. 49a shows a gripper concept with a thin stem and big, arrow-like arms whose motion runs along the axis of the cylinder. The gripper in Fig. 49b occupies little space because most of its surface in the flat state lies beneath the cylinder, and its gripping motion is angled at approximately  $45^\circ$  to the axis of the cylinder. The gripper in Fig. 49c is larger than the two other concepts, grips perpendicularly to the cylinder, and contains a degree-5 vertex.

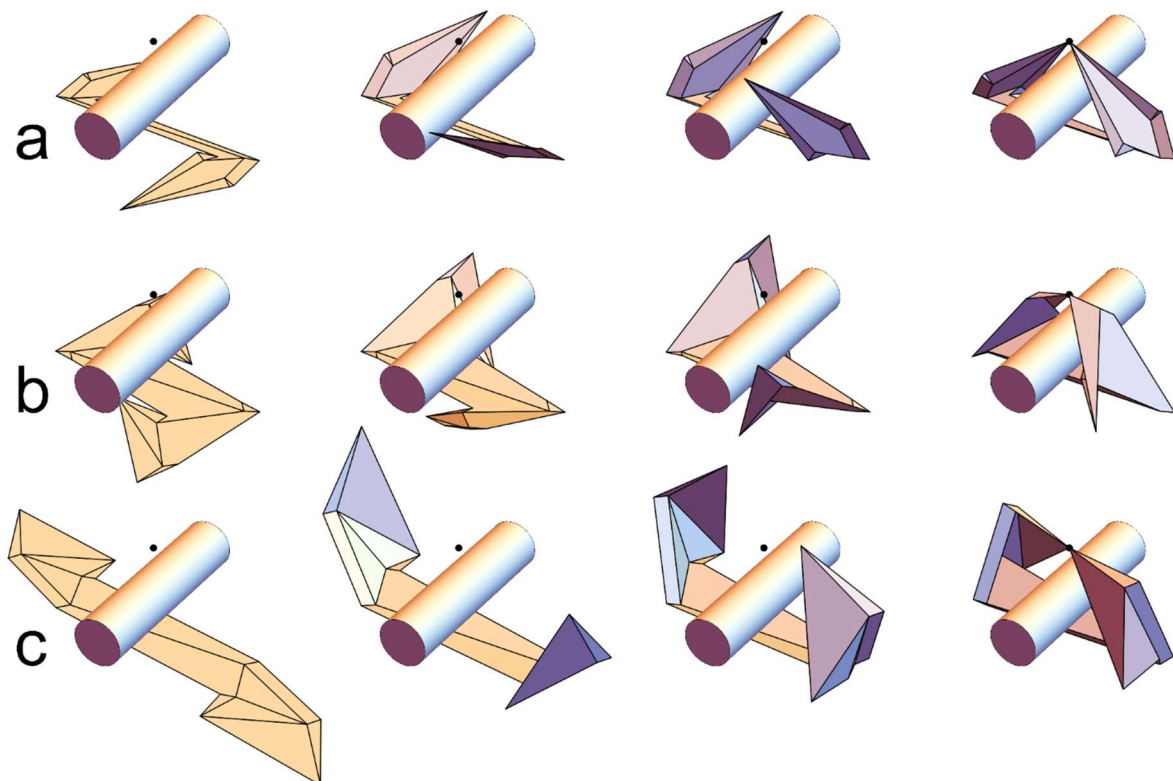


Fig. 49: Three different origami gripper concepts. (a) Gripper with rule application sequence 2, 3, 4, (8, 9) and gripping motion along the cylinder axis. (b) Gripper with rule application sequence 2, 3, 8, (7, 9) that is tightly packed beneath the cylinder in its flat state. (c) Gripper with rule application sequence 2, 3, (4, 8), 4, (7, 8) and gripping motion perpendicular to the cylinder axis.

With respect to the robotic arm task, 56 of the 1170 design alternatives are able to successfully approximate the trajectory points  $P_{T_1}$ , and thereof, 29 alternatives do not self-intersect. The folding motion of three design alternatives is depicted in Fig. 50 from left to right together with the trajectory points  $P_{T_1}$ . The three design alternatives exhibit an objective value of  $\Omega^* = 2 * 10^{-2}$  (a),  $\Omega^* = 3 * 10^{-2}$  (b), and  $\Omega^* = 4 * 10^{-2}$  (c).

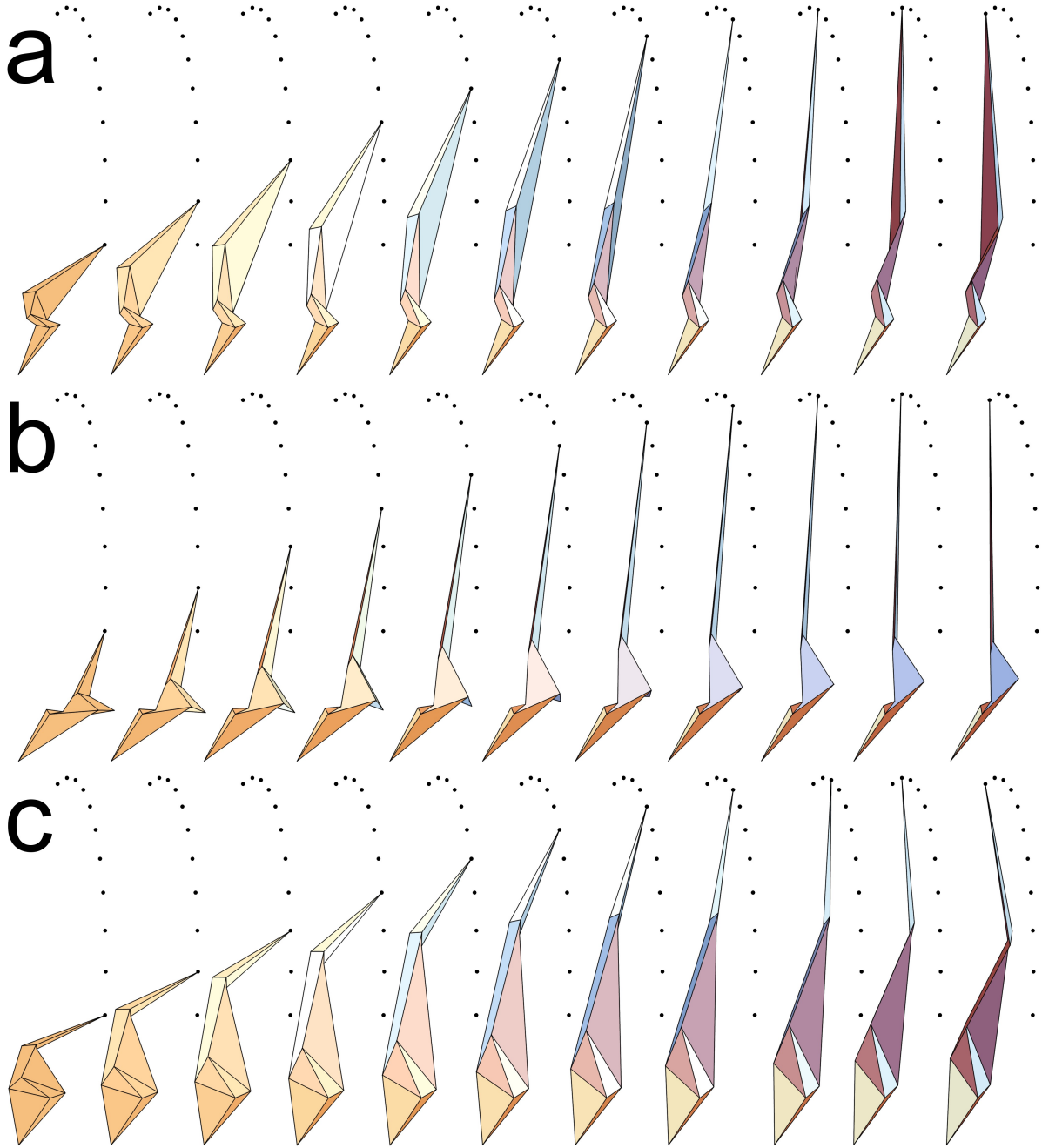


Fig. 50: (a-c) Folding motion from left to right of three design alternatives for the robotic arm. The concepts are generated by the rule application sequences 2, 4, 7, (6,9), 2, 3, 8, and 2, 4, 7, (3,6), (5,6) in (a-c), respectively.

## 5.7 Discussion

The crease pattern graphs generated purely by applications of  $r_1$  are the fastest to optimize, and the time consumption increases with each application of  $r_2$  because the constraint system  $\psi$  becomes more complex when the constraints depend on the in-plane coordinates  $x$ . The complexity lies in the symbolic representation of all dihedral angles, global rotation matrices, three-dimensional vertex coordinates, constraints, and the objective function. However, in comparison to the numerical approach in Section 3 that takes multiple hours for the optimization of a single design alternative, the symbolic representation enables the optimization of a design alternative in seconds to minutes. This speed up is achieved by the use of function derivatives within the solver, which enables a fast global optimization of each design alternative and thus the enumeration of the design space.

Applied to both the gripper and the robotic arm design task, the enumeration of the design space yields 52 distinct crease pattern graph topologies depicted in Fig. 48. While the search tree is narrow at the top where  $r_1$  predominates, the midsection of the search tree becomes broader since more extended vertices offer more possibilities for the application of  $r_2$ . The number of possible rule applications then progressively decreases with each line because of the maximum number of extended vertices  $N_{\max} = 3$ , which is why the search tree narrows again at the bottom. The overall shape of the search tree, however, is typical and generally independent of  $N_{\max}$ .

Fig. 51a plots the number of design alternatives over the number of rule applications for the gripper design task so that the  $x$ -axis corresponds to the vertical axis of the search tree shown in Fig. 48. The number of all 1170 designs in Fig. 51a is distributed according to the overall shape of the search tree. The distribution of the 836 designs that are able to grip  $P_G$  behaves similarly, although the curve peaks at four instead of five rule applications. At four rule applications, all graphs except one are generated by three applications of  $r_1$  and one application of  $r_2$ , while only  $r_2$  can be applied thereafter. Rule  $r_2$  introduces constraints into the system without adding variables, which is why the number of designs that can grip  $P_G$  steadily decreases from four applications onward. For the same reason, the percentage of successful gripping designs per total number of designs is highest at three rule applications, where five of seven graphs are generated purely by  $r_1$ . The same peaks at four rule applications can be observed in Fig. 51a for the 148 designs without self-intersection and for the 36 designs without any intersection. When at first one could have argued not to apply  $r_2$  since it only introduces constraints, these results show that a certain number of applications of  $r_2$  can drive the

optimized designs out of intersecting configurations. Without specific constraints that prevent intersection within the optimization,  $r_2$  thus provides more variability in the crease pattern graphs that satisfy the design task. However, if  $r_2$  is applied too many times, the designs are either unable to grip  $P_G$  or lead to intersection, which leads to zero feasible designs at six and seven rule applications for the gripper design task.

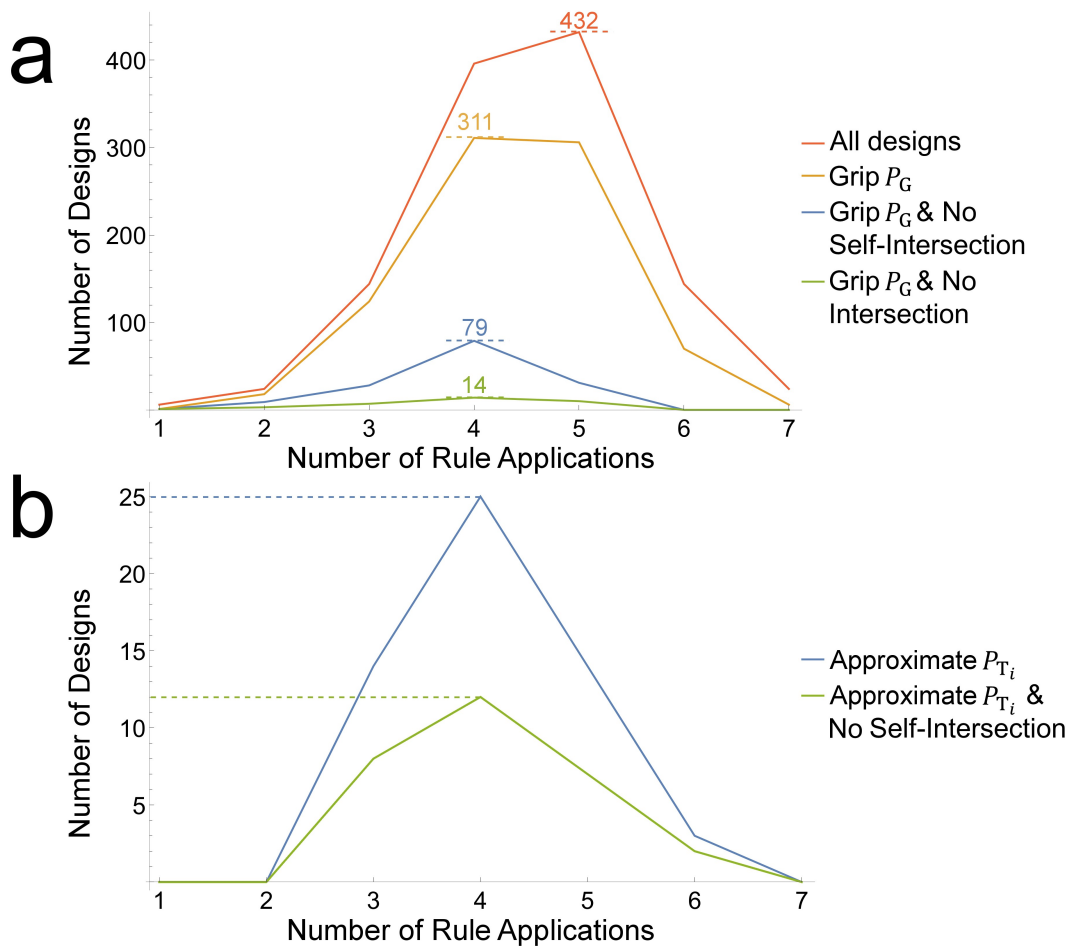


Fig. 51: (a) Plot corresponding to the gripper design task showing the distribution of the total number of designs, the number of designs that can grip  $P_G$ , that can grip  $P_G$  without self-intersection, and that can grip  $P_G$  without any intersection, over the number of rule applications. (b) Plot corresponding to the robotic arm design task showing the distribution of the designs that are able to approximate the points  $P_{T_i}$  and that do so without self-intersection, respectively.

Fig. 51b has the same axes as Fig. 51a but shows only the distribution of the 56 successful designs that are able to approximate the given trajectory and the 29 designs that in addition do not self-intersect. The same peaks appear at four rule applications, which again shows the usefulness of  $r_2$ . In contrast to the results of the gripper, for the robotic arm task there are feasible concepts generated by six rule applications, which probably stems from the wide motion defined by the trajectory that leads to longer crease lines and less tightly packed crease patterns (Fig. 50). The given trajectory represents a perfect quarter circle that seems simple to approximate. However, the maximum crease line length

$l_{\max} = 1.5$  prevents a simple solution, which renders a two-dimensional trajectory a hard problem for origami mechanisms that usually exhibit complex three-dimensional motion.

The future steps that correspond to the method itself involve the guidance of the method and the rule system. In both design tasks, there is no statistical difference in the performance of design alternatives with respect to the assignment of RBMs; both modes are equally represented across the range. Thus, a transfer from the design space enumeration to more efficient search methods, such as branch-and-bound algorithms [130], requires more investigation in the future to expand the application of the method to larger design spaces.

Another interesting path for future work involves the expansion of the presented graph grammar. The rule set  $\mathcal{R} = (r_1, r_2)$  enables the generative design of a variety of novel crease patterns such as the ones shown in Fig. 49 and Fig. 50, as well as existing crease patterns such as slender origami [122], origami strings [123], or the gripper in Ref. [3]. All of these achievable crease patterns have in common that they are kinematically determinate. However, many known crease patterns make use of specific sector angle configurations that fold rigidly although being overconstrained, as explained in Section 4.

To generate such overconstrained but rigidly foldable crease patterns, the rule system presented in Section 5.2 could be expanded by an adjusted version of rule  $r_2$ , here denoted as  $r'_2$ . Rule  $r'_2$  would also combine two vertices, but the vertex  $v'_{c_1}$  on the  $RHS_2$  in Fig. 45 would lie on the line segment between the predecessors  $v_{p_1}$  and  $v_{p_2}$  and would become a sink vertex ( $\deg^+(v'_{c_1}) = 0$ ) of the type  $T_{c'_1} = \emptyset$ . Fig. 52 illustrates how a Miura-ori pattern could be generated by the expansion of the presented rule system with rule  $r'_2$ .

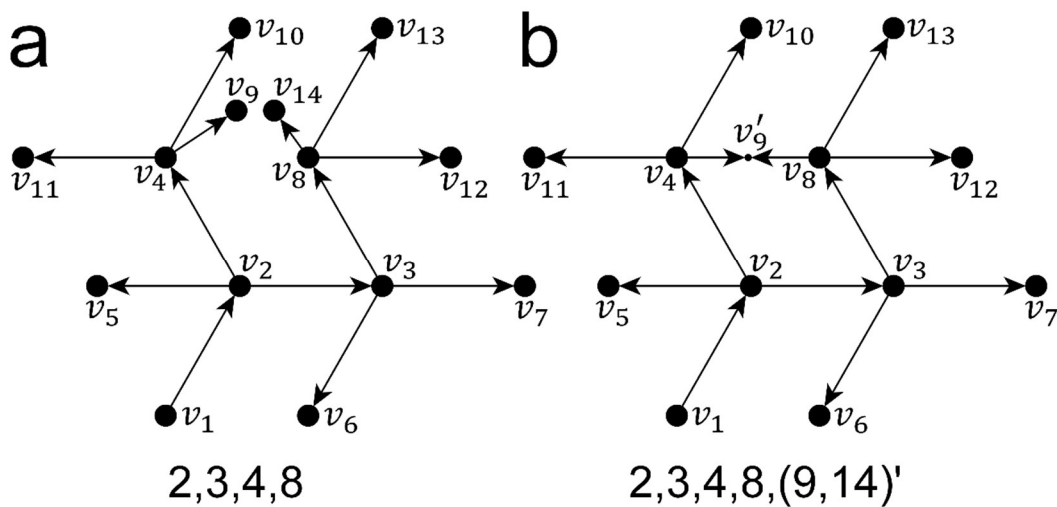


Fig. 52: (a) Crease pattern graph with rule application sequence 2, 3, 4, 8 that can be generated by the presented graph grammar. (b) Overconstrained Miura-ori pattern with rule application sequence 2, 3, 4, 8, (9, 14)' that could be generated by introducing and applying rule  $r'_2$ .

The crease pattern graph depicted in Fig. 52a can be generated by the initial graph  $G_0$  in Fig. 40 with the rule application sequence 2, 3, 4, 8. By applying  $r'_2$  to vertices  $v_9$  and  $v_{14}$ , resulting in the rule application sequence 2, 3, 4, 8,  $(9, 14)'$  and the crease pattern graph shown in Fig. 52b that depicts a Miura-ori unit cell.

The generation of the pattern in Fig. 52b would stipulate a condition for the dihedral angles,  $\rho_{8,9'} = \rho_{4,9'}$ , and thus demand a condition for the sector angle configuration of the entire crease pattern. As explained in Section 4.3.3, an overconstrained but rigidly foldable crease pattern can only be achieved by introducing some sort of symmetry. However, introducing such conditions within the constraint system generated by the graph grammar is complex and requires more future work. If such an implementation were successful, the author conjectures that the rule set consisting of  $r_1$ ,  $r_2$ , and  $r'_2$  would enable the generation of all possible rigidly foldable crease pattern graph topologies.





## 6 Discussion

This section discusses the findings and the contributions of the thesis with respect to the research questions and the objective. The objective is addressed first, followed by the research questions RQ1-RQ4 and the limitations that lead to the possible future paths.

### 6.1 Objective

*The objective of this thesis is to develop a computational method for the synthesis of rigidly foldable crease patterns to support the application of origami in engineering design tasks.*

Fig. 53 illustrates the overview of the computational method presented in this thesis. At the bottom of Fig. 53, the PTU introduced in Section 4 represents the basis for the method, stating that the kinematic behavior of a single vertex is determined by the vertex triangle. The PTU then leads to three major implications corresponding to the research question RQ1-RQ3. The PTU provides a condition for the rigid foldability of degree- $n$  vertices (RQ1), a kinematic model incorporating the RBMs (RQ2), and guidelines for the generation of crease patterns that arise from the relations between the kinematic determinacy, the number of DOF, and symmetry (RQ3). In Section 5, the answers to RQ1-RQ3 are embedded into a graph grammar system (RQ4) that comprises two rules with which a range of known and novel crease patterns can be generated. The graph grammar system is then integrated into the automated computational method that is structured according to a CDS method containing the following parts.

**Input:** The input to the computational method involves an initial graph and its actuation, a maximum number of internal vertices  $N_{\max}$ , and an engineering design task with target criteria. Since all design tasks realized through origami require crease patterns that exhibit an appropriate kinematic behavior, the computational method focuses on purely kinematic aspects of the design tasks to explore the geometric capabilities of origami. To capitalize on the full range of benefits offered by origami, the input should further include only actuations that exhibit a low number of DOF to facilitate a realization with the least possible amount of actuators, thus minimizing the available resources and enabling a reliable control [12] in remote or hardly accessible environments for which origami engineering applications are often designed [42].

**Representation:** Crease patterns are represented by labeled, directed graphs defined by the sets of vertices, edges, vertex labels, and edge labels. The vertex coordinates are represented in the Cartesian

coordinate system through spherical, relative coordinates [75] prompted by the kinematic model of the PTU. This representation yields a basis for the generative system in the generation part of the computational method.

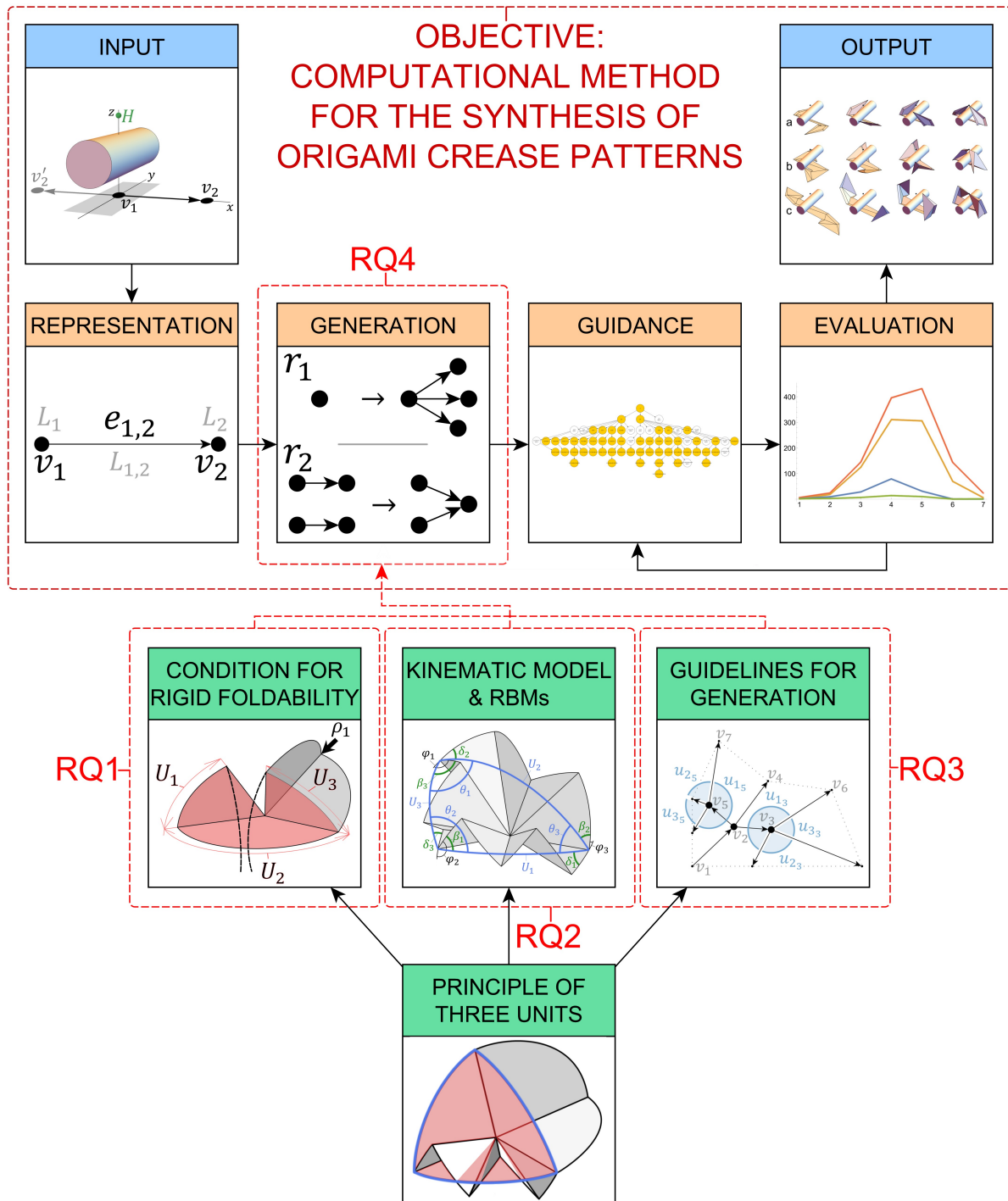


Fig. 53: Overview of the computational method for the synthesis of origami crease patterns presented in this thesis with the Principle of Three Units (PTU), research questions RQ1-RQ3, and objective, structured as a CDS method with input, representation, generation, evaluation, guidance, and output

**Generation:** The generation of crease patterns is enabled by the graph grammar that entails the constraints for rigid foldability, the kinematic model including the RBMs, and the guidelines for the

generation of kinematically determinate crease patterns. Automated applications of the two rules  $r_1$  and  $r_2$  generate all crease pattern graphs arising from the initial graph and the maximum number of internal vertices  $N_{\max}$  of the input.

**Evaluation:** The evaluation is constituted by both the optimization of a crease pattern and an intersection check. The objective function corresponding to the design task is minimized by the function *NMinimize* in Mathematica 11 that reformulates the nonlinear optimization problem as an unconstrained problem using penalty functions, which is by default solved with Differential Evolution [127]. Then, the intersection check is conducted only if the optimized geometry of a crease pattern satisfies the given target criteria to decrease the time consumption of the method.

**Guidance:** Due to the completely symbolic formulation of all components within the optimization scheme, the evaluation of a design alternative takes seconds to minutes, allowing for the enumeration of thousands of possible crease pattern graphs and RBM assignments, at least for the design tasks addressed in Section 5.

**Output:** After the automated loop that involves the evaluation and the guidance (Fig. 53) is terminated, the method outputs a range of origami crease pattern concepts that satisfy the given engineering design task, fold rigidly, and avoid self-intersection as well as obstacles. The origami concepts automatically synthesized in this thesis include rigid origami grippers and robotic arms.

**Contributions:** While the conventional process is tedious and time-consuming, the input required of a human designer to run the presented computational method is simple and straightforward. Due to the incorporation of fundamental knowledge about the kinematic behavior of origami crease patterns, the method enables a direct generation approach that relies on a purely symbolic optimization model. This symbolic model enables the computational method to generate and evaluate a large number of possible design alternatives and to search the vast design space offered by origami. In addition, the results of the grippers and the robotic arms in Section 5.6 demonstrate the ability of the method to synthesize novel crease patterns that satisfy the given engineering tasks. Moreover, to the best knowledge of the author the robotic arm design task is the first application of the origami principle to the approximation of a trajectory, signifying that the computational method developed in this thesis can expand the range of possible applications of origami to an even broader set of engineering design tasks. These contributions are enabled by addressing the research questions whose answers are summarized in the following.

## 6.2 Research Question 1, Part 1

*What are the characteristics of the search space of rigid foldability?*

Section 3 presents a numerical approach to the computational method and introduces a new simulation method that enables the examination of the kinematic behavior of an origami. The simulation method is independent of any constraints for rigid foldability and thus allows for the assessment of both rigidly and non-rigidly foldable crease patterns. This assessment provides the possibility to visualize the entire kinematic search space of an origami instead of only the rigidly foldable space, which is common in related works, such as [131]. The visualization then reveals the formation of rigidly foldable, three-dimensional regions in which the kinematic behavior of the crease pattern is smooth and monotonic. Some of these regions are delimited by abrupt boundaries that signify the advent of distortion within the crease pattern. For a single vertex, these boundaries between rigid and non-rigid regions coincide with flat foldable configurations, which is an interesting finding considering that the exact nature of the connection between rigid and flat foldability has not been previously examined. In addition, the visualization in Fig. 14 demonstrates a hierarchy of layers and vertices, where the kinematic behavior of vertices closer to the actuation are not influenced by vertices further away if the crease pattern folds rigidly. This hierarchy implies a flow of information from the actuation outward, which hints at the possibility to add rigidly foldable vertices to an existing rigidly foldable crease pattern without compromising the original rigid behavior.

**Contributions:** In general, the assessment of the first part of RQ1 contributes by enabling the visualization of the search space of the kinematic behavior of an origami. Visualization offers the potential for human designers to process complex behavior and discern previously hidden patterns whose discovery can lead to a new perspective on the underlying problem [132]. In particular, the assessment of the first part of RQ1 reveals rigidly foldable regions within the search space of rigid foldability, a connection between rigid and flat foldability, and the hierarchy of layers, all of which motivate a more in-depth analytical investigation of RQ1 in Section 4.

## 6.3 Research Question 1, Part 2

*What are the conditions for an origami to fold rigidly?*

Since the numerical approach in Section 3 does not offer the means required for an automated synthesis of origami crease patterns, Section 4 delves deeper into the mathematical underpinnings of

rigid foldability by investigating analytical formulations for origami kinematics. The investigation results in a sufficient and necessary condition for the rigid foldability of degree-4 vertices in Eq. (9) that is then reduced to a purely sufficient condition in Eq. (13). The sufficient condition states that a degree-4 vertex with driving angle  $\rho_1$  folds rigidly if  $|\alpha_1 - \alpha_4| \geq |\alpha_2 - \alpha_3|$ , which is translated into a visual example in Fig. 23 that confirms the rigidly foldable regions found in Section 3. Analyses of Eqs. (9) and (13) further explain the connection between flat and rigid foldability in a degree-4 vertex, stating that flat foldability signifies the last rigidly foldable state of a vertex whose driving angle goes to its extremes at  $\pm\pi$ . This result confirms that flat foldability does represent the boundary between rigidly and non-rigidly foldable regions as prompted by Section 3, at least for degree-4 vertices.

The recurring emergence of spherical triangles in the analytical assessment of degree-4 vertices then motivates the closer examination of the kinematics of degree- $n$  vertices. This examination results in the PTU, a principle that yields unprecedented insight into origami kinematics and answers the second part of RQ1 in three parts. First, in comparison to related methods [89], the PTU recognizes that the kinematic behavior of a single vertex is determined only by a single spherical triangle that is called the vertex triangle. By applying the triangle inequality to the vertex triangle, the PTU determines the condition that for a single vertex to fold rigidly none of its unit angles  $U$  can be greater than the sum of the two remaining unit angles,  $U_{\max} \leq U_{\text{med}} + U_{\min}$  (Eq. (25)). Since the unit angles can be expressed analytically using Eq. (18), this condition includes both the driving and the sector angles in symbolic form. Moreover, the condition is independent of the chosen set of driven crease lines, the degrees of the vertex (for  $n \geq 4$ ), and the developability of the surface, which means that it can be applied universally to single origami vertices and even more generally to spherical mechanisms with purely rotational joints of degree  $n$ . This constitutes the answer to the second part of RQ1 with respect to single-vertex origami.

Second, the PTU yields a kinematic model to determine the unknown dihedral angles of a single vertex, which is important for the following reasons. Any rigidly foldable single vertex is a node that takes  $n - 3$  inputs and transforms these into three outputs [74], which is proposed in Section 3 in slightly different form as the hierarchy of vertices. This notion turns an origami into a directed graph, signifying that if any two vertices are connected, the shared crease line transmits the output of the first as an input to the second vertex. The PTU specifies that the nature of these inputs and outputs, and thus the information transmitted between the vertices, are the unit angles. This renders an origami crease pattern

a complex network in which the conserved entity is the surface area of the origami. A time-dependent unit angle expresses the change in the curvature of the surface around a vertex, which needs to be balanced by the remaining unit angles, leading to the formation of the vertex triangle and, more fundamentally, to the folding motion. However, to determine the unit angles all unknown dihedral angles must be known, which requires a kinematic model. The kinematic model provided by the PTU is thus instrumental for the transfer of the condition for rigid foldability from single vertices to multi-vertex crease patterns.

Finally, the PTU is able to determine the global rigid foldability of a large subset of origami crease patterns. This subset corresponds to kinematically determinate and acyclic crease patterns for which the condition in Eq. (25) can be applied successively to each vertex within the crease pattern. The rigid foldability of all individual vertices then collectively determines the global rigid foldability of the mentioned subset of crease patterns. A model for cyclic graphs is not established in this thesis, but the size of the design space constituted by acyclic graphs is undoubtedly large enough to support the design of origami crease patterns for technical solutions. Moreover, the answer to the second part of RQ1 is not tied to the kinematic model of the PTU but to the vertex triangle and the kinematic determinacy: the condition for an origami to fold rigidly is that the underlying crease pattern is kinematically determinate and that each of its vertices individually satisfies  $U_{\max} \leq U_{\text{med}} + U_{\min}$ .

**Contributions:** The PTU offers a range of contributions with respect to the second part of RQ1. The condition for the rigid foldability of degree- $n$  vertices has long been an outstanding problem in origami [62] and its discovery opens up new possibilities for the embedding of higher order vertices within crease patterns because any single degree- $n$  vertex can be directly designed to be rigidly foldable. Degree- $n$  vertices can be advantageous, e.g., in metamaterials to achieve multi-stability [63] or in the approximation of surfaces that exhibit extreme Gaussian curvatures [133].

The PTU further offers an analytical kinematic model for single vertices and for entire crease patterns. The kinematic model computes the equations for the dihedral angles of degree- $n$  vertices in real time because no matrix inverse calculations are involved. In addition, these equations are given in symbolic form, which speeds up the simulation process as well as the optimization of crease pattern geometries. While the optimization of the adapted flasher pattern in Section 3 takes multiple hours, the optimization of comparably complex crease patterns in Section 5 takes seconds to minutes. This increase in efficiency renders the kinematic model of the PTU perfectly suitable for an application within an automated computational method.

The PTU in general leads to a deeper understanding of origami mathematics by explaining the connection between rigid and flat foldability, the two biggest mathematical notions in origami, as well as the information flow in origami crease patterns. The understanding of such phenomena contributes fundamentally to origami research.

#### 6.4 Research Question 2

*How can the exponential number of RBMs be modeled kinematically?*

Due to the singular flat initial state of an origami, kinematically modeling different RBMs through existing approaches is a cumbersome process and conventionally involves vertex perturbations [102] or prior knowledge about the MV assignment of the crease pattern [40]. The kinematic simulation method introduced in Section 3 allows for the kinematic modeling of RBMs, but it does so mainly by assessing non-developable configurations similar to the perturbation of vertices. Moreover, it resembles a search for rather than a choice of RBMs, which precipitates multiple objective evaluations of the same pattern in different configurations. This repeated assessment of each crease pattern is one of the reasons why the numerical approach in Section 3 eventually necessitates a stochastic search method to guide the optimization of the crease pattern geometry, leading to an ineffective design process.

The PTU introduced in Section 4 provides a new perspective on RBMs and offers a solution to RQ2. The PTU states that the RBMs are just a consequence of the two possible formations of the vertex triangle (Fig. 25), which is why the internal angles of the vertex triangle can simply be applied in opposite directions. Then, the unknown dihedral angles corresponding to either RBM are automatically determined by the kinematic model of the PTU based on the decision made by a human designer or a computational method.

**Contributions:** With respect to the RBMs, the introduction of the PTU completely changes the design process: origami research focuses predominantly on the MV assignment instead of RBMs, which leads to an unnecessarily complicated, NP-hard problem [64]. The task of finding a valid MV assignment does not include the actuation of the crease pattern and assumes that every crease line can become a mountain or a valley. Since there are usually more crease lines than vertices within a crease pattern, the number of MV assignments is even higher than the number of RBMs. The only existing method [89] that is able to analytically determine the unknown dihedral angles of degree- $n$  vertices yields multivalued solutions that contain no information about the sign of the dihedral angles. In contrast, the kinematic model of the PTU incorporates the signs of these angles and offers a choice from only two options for

each individual vertex. Hence, instead of searching through the exponential number of RBMs, specific RBMs can be chosen and then kinematically modeled by the PTU. This approach supersedes the repeated evaluation or the prior knowledge about the MV assignment of a crease pattern and renders the PTU perfectly suitable for an application within an automated computational method. Then, the exponential number of different RBMs is turned from a potential into a definitive benefit of origami.

### 6.5 Research Question 3

*What are the relations between the kinematic determinacy, the number of DOF, and the symmetry of an origami crease pattern?*

RQ3 is assessed in Section 4 by transferring the implications of the PTU from single-vertex to multi-vertex crease patterns. Each individual vertex of degree  $n$  requires  $n - 3$  inputs [74] to actuate the folding motion, which is equal to the number of DOF that need to be prescribed for a single vertex to fold in a kinematically determinate manner. As an example, Fig. 54a shows two unconnected vertices  $v_1$  and  $v_2$  that are both driven by a driving angle  $t$  at the allocated crease lines. Since both vertices are of degree four, the single driving angle for each vertex renders both vertices kinematically determinate.

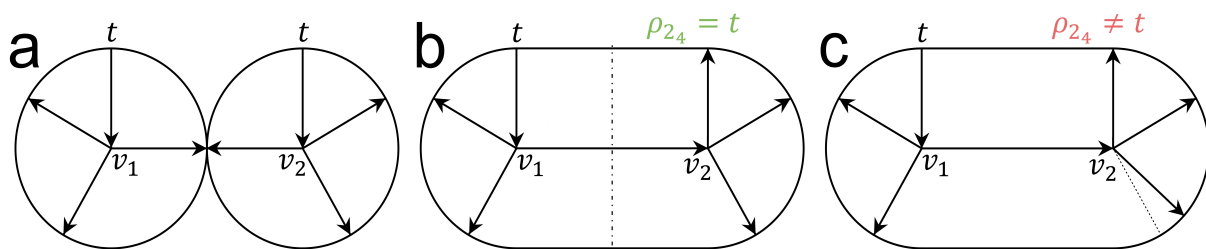


Fig. 54: (a) Two unconnected single vertices  $v_1$  and  $v_2$ , (b) two connected, symmetric vertices that are kinematically determinate even when the driving angle  $t$  from (a) would be applied to  $v_2$ , and (c) two connected, asymmetric vertices that are kinematically determinate only if the driving angle  $t$  from (a) is not applied to  $v_2$ .

When multiple vertices are connected, the PTU states that every vertex within the crease pattern requires 3 outputs to be kinematically determinate. In Fig. 54b,  $v_1$  and  $v_2$  are connected and the driving angle  $t$  previously applied to  $v_2$  is suspended, in which case both vertices exhibit 3 outgoing crease lines and are thus kinematically determinate. However, since  $v_1$  and  $v_2$  are mirror-symmetric, the outgoing dihedral angle  $\rho_{2,4}$  is identical to the previously suspended driving angle  $t$ . Reinstating the driving angle to  $v_2$  would thus lead to a redundant constraint that would not compromise the kinematic determinacy. The same statement is not true for Fig. 54c, where the location of one crease line incident to  $v_2$  changes its location, leading to  $\rho_{2,4} \neq t$ . In this case, additionally applying the driving angle  $t$  would then overconstrain the system. Although simple, the exemplary situation in Fig. 54 can be transferred to



rotational symmetry, for which the PTU demonstrates exactly which constraints need to be redundant to enable folding motion, as illustrated for the Miura-ori pattern in Fig. 52.

To answer RQ3, symmetry in origami leads to highly specific sector angle configurations in which some of the involved constraints are redundant, allowing a crease pattern to fold in a kinematically determinate way although the same crease pattern in any asymmetric sector angle configuration would be overconstrained when actuated by the same number of DOF. Symmetry can arise through either geometric symmetry or fold angle multipliers, both representing cases in which the underlying information is symmetric.

Although this concept has been described for degree-4 vertices [99], with the exception of the fold angle multipliers [91] the relations between the kinematic determinacy, the number of DOF, and symmetry are subject to analyses in order to determine the mobility of crease patterns [37, 50-52] instead of useful properties for generative methods. Hence, while analyses of given crease patterns are interested in assessing the number of DOF and thus the appropriate actuation of the folding motion, this thesis specifies the actuation as an input and is interested in generating kinematically determinate crease pattern topologies. The relations between the kinematic factors mentioned in RQ3 are thus slightly different in both cases. To clarify these relations with respect to the design perspective, Fig. 55 depicts the ontology of an origami crease pattern in which the inputs of a design practitioner or a computational method are highlighted in green and in which the research questions RQ1-RQ3 are colored in orange.

Defining an origami crease pattern (Fig. 55, top) determines both the topology and the geometry of the crease pattern. The topology constitutes the number of vertices and the number and distribution of crease lines, whereas the geometry determines the sector angle configuration, i.e. the size of the sector angles around each vertex.

The number of vertices directly relates to the number of unknowns within the system since every vertex in motion exhibits three unknown coordinates corresponding to the three-dimensional space [102]. The crease lines between the vertices define the distribution of facets and thus the total number of constraints: each crease line introduces a single constraint by fixing the Euclidean distance between two vertices, facets with more than three sides add constraints along their diagonals, and the closed surface around each internal vertex adds a closure constraint. Together with the sector angle configuration, the distribution of crease lines may lead to **symmetry** (RQ3), which can render some of the involved constraints redundant, resulting in the number of independent constraints.

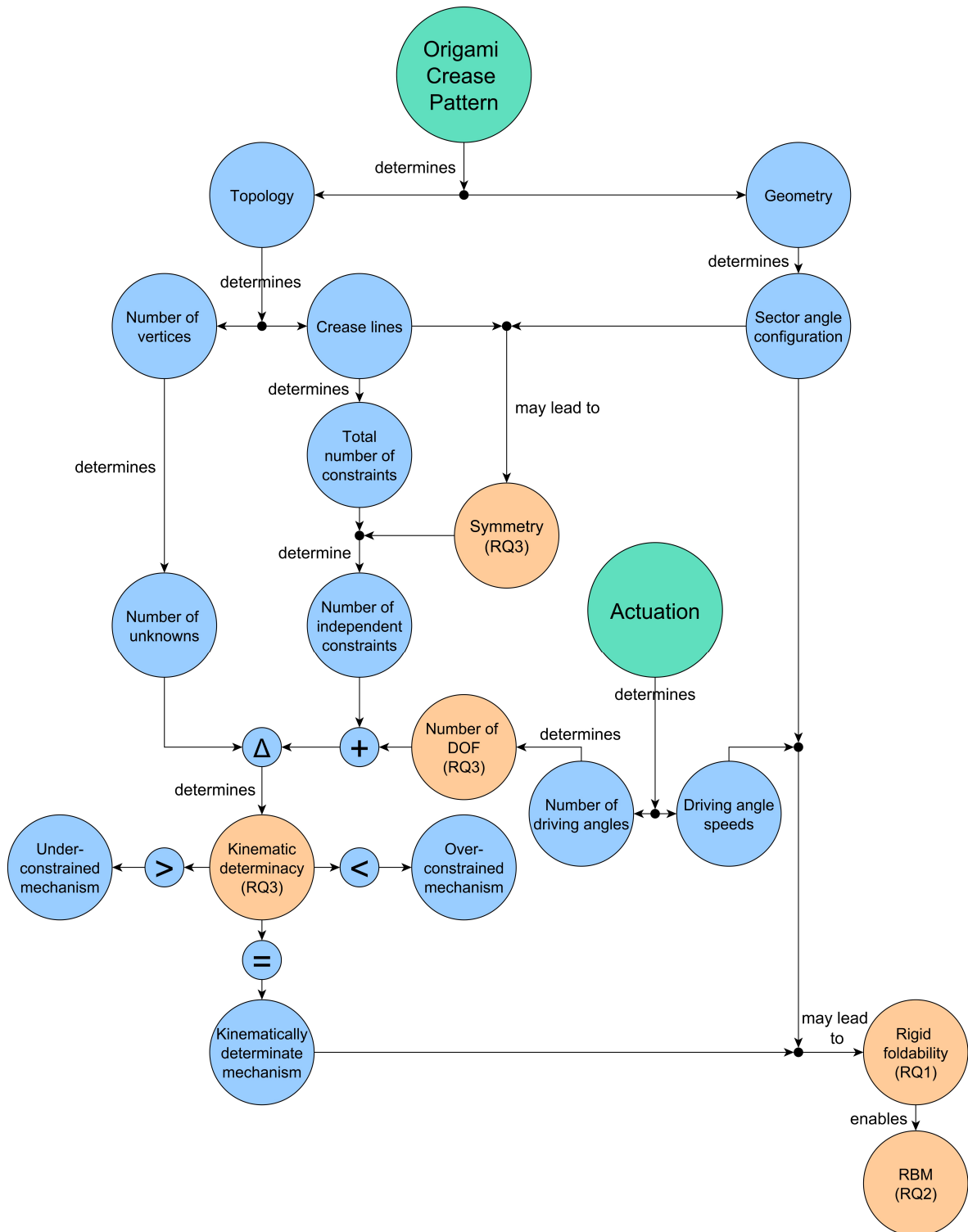


Fig. 55: Ontology of an origami crease pattern from a design perspective with the inputs of design practitioners or computational method shown in green and the research question RQ1-RQ3 colored in orange

Folding an origami crease pattern demands for an actuation of the crease pattern that determines both the number of driving angles as well as their speeds. The number of driving angles directly determines the **number of DOF** (RQ3) that can be added to the number of independent

constraints and then compared to the number of unknowns, determining the **kinematic determinacy** (RQ3) of the crease pattern. If the number of unknowns is greater or smaller than the sum of the number of independent constraints and the **number of DOF**, the crease pattern is equivalent to an underconstrained or an overconstrained mechanism, respectively. Note that the above comparison already includes the effect of symmetry, which is why the under- and overconstrained mechanisms in Fig. 55 do not fold in a kinematically determinate way (in comparison to the conventional usage of “overconstrained but still rigidly foldable” [99]). Kinematically determinate mechanisms only emerge when the sum of the number of independent constraints and the **number of DOF** is equal to the number of unknowns.

A kinematically determinate mechanism is an important requirement for rigid foldability (RQ1) that further depends on the sector angle configuration and the actuation of the crease pattern. While the actuation also determines the number of driving angles, its more direct influence on rigid foldability arises from the speed of the driving angles as defined by the input of a human design practitioner or the computational method. Finally, rigid foldability enables the emergence of different RBMs (RQ2).

**Contributions:** A subtle but crucial detail in Fig. 55 is the directions of the arrows that hint at a major contribution of this work: by unraveling the precise relations between the factors that determine the kinematic behavior from a design perspective, three aspects of the origami design process are inverted.

First, while related works [52, 102, 134] present methods to analyze the composition of crease patterns in order to determine the number of DOF with which a feasible motion can be achieved, in this thesis the actuation is an input and the entire design process tailors an origami specifically to the number of DOF. This inversed approach contributes by enabling the actuation of the folding motion through a desired low number of DOF, which reduces the resources required to actuate the origami, facilitates the control [99], and renders the synthesized mechanisms more reliable [12].

Second, related works focus on symmetry as the subject of considerable analytical efforts [50, 51] because it complicates the understanding of the kinematic behavior. Possibly sparked by the many known symmetric tessellations and natural fold pattern, symmetry is commonly perceived as an inevitable feature of origami and is utilized in the design of symmetric crease line distributions and sector angle configurations (such as in [91]). To ensure a reliable actuation, in this thesis symmetry is perceived as a phenomenon best prevented so that it exerts minimal impact on the difference between the total

number of constraints and the number of independent constraints (Fig. 55). In other words, instead of focusing on patterns that can only fold in the presence of perfect symmetry, this work synthesizes crease patterns that are always able to fold except in the rare case of symmetry. These synthesized crease patterns are less prone to imperfections in the physical realization and thus generally more robust than symmetric crease patterns.

Third, when symmetry is avoided as a design feature, the number of total constraints is equal to the number of independent constraints (Fig. 55). Together with the number of DOF that are defined as an input to the generation, the number of independent constraints then determine the kinematic determinacy in a straightforward and reliable manner since the generative method can simply guarantee that each individual vertex is kinematically determinate. In a directed graph, as stipulated by the PTU, each single vertex is kinematically determinate if it exhibits three outgoing crease lines, and its kinematic behavior can be modeled using the PTU if the crease pattern graph is acyclic. These two guidelines enable the generation of kinematically determinate crease pattern topologies and are thus perfectly suited for an application within a computational method that targets the design of origami crease patterns.

#### 6.6 Research Question 4

*How can the answers to RQ1-RQ3 be embedded within an automated, generative method to synthesize origami crease patterns?*

Section 5 addresses RQ4 by embedding the answers to RQ1-RQ3 within an automated, rule-based graph grammar system. The rule set  $\mathcal{R}$  is constituted by two rules  $r_1$  and  $r_2$ , the first of which extends vertices with three outgoing crease lines to satisfy the first guideline arising from the answer to RQ3. The application of  $r_1$  turns a vertex into an internal vertex whose kinematic behavior can then be assessed using the PTU. Rule  $r_1$  automatically models the kinematics of a vertex and further constrains the sector angles around the respective vertex to fold rigidly by introducing the condition in Eq. (33). As addressed by RQ2, the kinematic model applied by  $r_1$  also adds the RBMs as variables to the optimization. The second rule  $r_2$  combines two vertices in order to enable the generation of higher order vertices and guarantees that the crease pattern graphs generated are acyclic, satisfying the second guideline that emerges from answering RQ3. In addition, rule  $r_2$  performs adjustments to the constraint system and the set of variables so that the crease pattern graphs generated can be optimized with respect to the engineering design task. The application of both rules can then be enumerated by

matching their respective LHSs, leading to an automated generative method for the synthesis of origami crease patterns.

**Contributions:** In contrast to related methods [43], the graph grammar system incorporates fundamental knowledge about the kinematics of origami and thus offers a more direct approach to the design of origami crease patterns. As such, the graph grammar system is an integral part of the computational method that enables the achievement of the objective of this thesis.

### 6.7 Limitations and Future Work

The work presented in this thesis suggests five main limitations and respective possible paths for the future. First, the PTU enables only the assessment of acyclic crease patterns. Investigating cyclic graphs and determining a kinematic model and a corresponding condition for rigid foldability in the future would result in a generic condition for the global rigid foldability of all origami crease patterns. Together with the expansion of the rule system explained in Section 5.7, such a condition would broaden the design freedom of the computational method even further.

Second, although the computational method includes an intersection check, this check is performed numerically and applied only after the optimization of a design alternative to reduce the time-consumption. Intersection is thus only prevented instead of optimized for, leading to the rejection of possibly feasible designs and thus requiring more objective function evaluations than theoretically necessary. Determining and integrating an analytical constraint for self-intersection in the future could decrease the number of objective function evaluations required to discover feasible origami concepts.

Third, the guidance strategy chosen in this thesis equates to the enumeration of the design space, which probably represents a less-than-ideal search through the design space. The vast design space offered by origami is constituted by the endless possibilities of different crease pattern topologies and the exponential number of RBMs. With respect to the former, Section 5.7 explains the influence of the rules on the overall performance of different topologies: while an application of  $r_1$  facilitates the achievement of a design task by introducing additional optimization variables, an application of  $r_2$  complicates the satisfaction of a design task by introducing additional constraints. However,  $r_2$  also leads to higher order vertices that generally enable more flexibility in the achievable shapes, and the trade-off between the applications of the two rules is not entirely clear. The results in Section 5.6 did also not reveal a difference in performance with respect to the RBMs. Hence, a future investigation of

the influence of crease pattern topologies and RBMs on the performance of generated crease pattern graphs could lead to an improved guidance strategy that would increase the time-efficiency of the proposed computational method.

Fourth, the concepts developed within this thesis are of zero thickness, and applying them in real-life scenarios requires adaptations for finitely thick materials. Adapting zero thickness concepts works best for crease patterns that exhibit only degree-4 vertices and uniform sector angle configurations. To achieve only degree-4 vertices, users can select for the feasible designs generated only by applications of rule  $r_1$  or a minimum number of applications of rule  $r_2$ . To achieve uniform sector angle configurations, the variable boundaries  $\alpha_{\min}$  and  $\alpha_{\max}$  in Eqs. (31) and (34) can be defined to lie within a smaller interval than currently done in the case studies of the gripper and the robotic arm. However, introducing such intervals might deplete the number of feasible design concepts, and the user of the method must estimate the trade-off between the number and the realizability of such solutions. Nonetheless, adapting the feasible concepts to finite thickness requires future work. A range of possible adaptation techniques is given in [66].

Finally, the computational method is applied to only two engineering design tasks in this thesis. Although the results for both design tasks demonstrate the usefulness of the method, targeting more design tasks in the future could further highlight the benefits of the method for the synthesis of rigid crease patterns to support the application of origami in engineering design tasks.

## 7 Summary and Conclusion

While origami is an ancient art form, its application in engineering science has only been popularized in recent decades when the scientific community recognized its numerous benefits. These benefits include scale-independence, compactly stowed states and complex three-dimensional motion, actuation through low numbers of Degrees-Of-Freedom (DOF), programmable mechanical properties, shapes, and functions, as well as facilitated manufacturing in the flat state that reduces cost and assembly time. Due to its benefits, today origami finds application in various scientific domains from micro scale such as DNA research to macro scale such as deployable solar panels for outer space.

The benefits and the widespread applicability of origami are accompanied by a set of geometric and kinematic challenges involving rigid foldability, an exponential number of Rigid Body Modes (RBMs), complex relations between the kinematic determinacy, the number of DOF, and symmetry, as well as intersection and the adaptation to finitely thick materials. These challenges complicate the manual adoption of origami principles for scientific purposes, which led to the development of various computational methods that support the application of origami in engineering design tasks. However, most of these methods isolate and address specific challenges and focus on the adaptation of existing crease patterns rather than the design of novel crease patterns that are tailored to the given design tasks. Consequently, today the design of rigid origami crease patterns for engineering applications is tedious, time-consuming, and limited to a handful of experienced scientists.

This gap motivates the present thesis and defines its objective as the development of a computational method for the synthesis of rigidly foldable crease patterns to support the application of origami in engineering design tasks. To achieve the objective, the thesis addresses four research questions of which the first three target the geometric challenges of rigid foldability (RQ1), RBMs (RQ2), and the relations between the kinematic determinacy, the number of DOF, and symmetry (RQ3). RQ4 then aims at embedding the answers to RQ1-RQ3 within an automated computational method that conforms to a Computational Design Synthesis (CDS) method with six parts involving input, representation, generation, evaluation, guidance, and output.

The first approach to the computational method is presented in Section 3 that introduces a new kinematic simulation method. Using this simulation, a manually adapted flasher pattern is subjected to a sensitivity analysis that maps parts of the search space, first for a single vertex and then for multiple connected vertices. Following the results of the analysis, the geometry of the adapted flasher pattern is

then optimized to exhibit a maximally large surface area and to fold into a given cuboid shape. Although successful in the optimization of a single crease pattern topology, the numerical approach is too time-intensive for an automated computational method targeted at generating and evaluating large numbers of different crease pattern topologies. Nevertheless, Section 3 reveals some important findings for the remainder of the work including the visualization of the search space and the discovery of rigidly foldable regions.

Based on these findings, Section 4 examines the involved kinematics from an analytical perspective to gather more knowledge about the mathematical underpinnings and build a stronger foundation for a more efficient approach to the computational method. This investigation leads to the inception of the Principle of Three Units (PTU) stating that the kinematic behavior of a single vertex is only dependent on its vertex triangle. By subjecting the vertex triangle to the triangle inequality, the PTU yields the conditions for the rigid and flat foldability of single vertices of degree  $n$ . The corresponding kinematic model enables the assessment of different RBMs and further offers the active selection of RBMs to the human designer or the computational method. In addition, the kinematic model analytically expresses the unknown dihedral angles, allowing for the extension of the condition for rigid foldability from single vertices to crease patterns. This extension leads to the guidelines for the generation of kinematically determinate crease patterns that can be modeled by the PTU. These guidelines are then manually applied to find novel crease patterns, such as the rigid flasher (Fig. 32), the star pattern (Fig. 34), and an origami chair pattern (Fig. 37). This application also demonstrates the tedious, manual design process and again motivates an automation of the generative process.

The analytical basis provided by Section 4 is then embedded within the computational method in Section 5. The core of this method is constituted by a graph grammar that simultaneously performs the graph transformations, introduces the optimization variables, and builds the system of constraints. All possible combinations of crease pattern topologies and RBM assignments arising from the definition of the design task and the graph grammar are then enumerated and optimized to yield a set of feasible origami concepts that satisfy the given engineering design tasks. In this thesis, the method is applied to two design tasks involving a gripper and a robotic arm problem, resulting in novel origami crease patterns (Fig. 49 and Fig. 50) that achieve the prescribed task, fold rigidly, and are free of intersection.



In summary, the major contributions of this work are:

- The visualization and characterization of the kinematic search space and the discovery of the rigidly foldable regions (RQ1, part 1)
- The conditions for rigid and flat foldability of degree- $n$  vertices that can be applied to generic single vertices or kinematically determinate and acyclic crease patterns (RQ1, part 2)
- The analytical kinematic model that determines the unknown dihedral angles and the RBMs (RQ2)
- The explanation of the relations between the kinematic determinacy, the number of DOF, and symmetry from a design perspective as well as the guidelines for the generation of kinematically determinate crease patterns (RQ3)
- The development of a graph grammar (RQ4) that incorporates the analytical foundation arising from the answers to RQ1-RQ3
- The achievement of the objective augments today's predominantly manual design process with a computational method that searches through the vast design space offered by origami, leads to the synthesis of novel crease patterns, and yields the potential to apply the origami principle to yet uncharted territories in engineering design.

The five main limitations of this work suggest different options for future investigations. First, determining the condition for the rigid foldability of cyclic crease pattern graphs and expanding the rules of the graph grammar would enable the generic assessment of the global rigid foldability and the generation of all possible origami crease patterns. Second, finding analytical constraints for self-intersection and incorporating these constraints into the optimization scheme would reduce the number of evaluations and thus speed up the computational search for suitable origami concepts. Third, examining the influence of the crease pattern topology and the RBMs could lead to a more efficient guidance strategy. Finally, applying the developed method to more engineering design tasks would further highlight the contributions of this work.

The author considers the objective of this thesis as achieved due to the successful development of an automated computational method that enables the synthesis of rigidly foldable origami crease patterns to support the application of origami in engineering design tasks. Moreover, the findings and contributions of this thesis lead to a deeper understanding of origami kinematics and offer the potential to advance the field of scientific origami.



## References

- [1] C. M. Dobson, "Protein Folding and Misfolding," *Nature*, vol. 426, no. 6968, p. 884, 2003.
- [2] K. Saito, S. Nomura, S. Yamamoto, R. Niiyama, and Y. Okabe, "Investigation of Hindwing Folding in Ladybird Beetles by Artificial Elytron Transplantation and Microcomputed Tomography," *Proceedings of the National Academy of Sciences*, vol. 114, no. 22, pp. 5624-5628, 2017.
- [3] J. A. Faber, A. F. Arrieta, and A. R. Studart, "Bioinspired spring origami," *Science*, vol. 359, no. 6382, pp. 1386-1391, 2018.
- [4] T. Tallinen, J. Y. Chung, F. Rousseau, N. Girard, J. Lefèvre, and L. Mahadevan, "On the Growth and Form of Cortical Convolution," *Nature Physics*, vol. 12, no. 6, p. 588, 2016.
- [5] J. S. Shim, A. Kubota, and T. Imaizumi, "Circadian Clock and Photoperiodic Flowering in Arabidopsis: CONSTANS is a Hub for Signal Integration," *Plant physiology*, vol. 173, no. 1, pp. 5-15, 2017.
- [6] N. I. Libeskind *et al.*, "Tracing the Cosmic Web," *Monthly Notices of the Royal Astronomical Society*, vol. 473, no. 1, pp. 1195-1217, 2018.
- [7] R. J. Lang, *Origami Design Secrets: Mathematical Methods for an Ancient Art*. AK Peters/CRC Press, 2011.
- [8] K. Hatori, "History of Origami in the East and the West before Interfusion," *Origami*, vol. 5, p. 5th, 2011.
- [9] R. J. Lang, *The Complete Book of Origami: Step-by-step Instructions in Over 1000 Diagrams: 37 Original Models*. Courier Corporation, 1988.
- [10] Z. Wang *et al.*, "Origami-based Reconfigurable Metamaterials for Tunable Chirality," *Advanced Materials*, vol. 29, no. 27, p. 1700412, 2017.
- [11] J. Morgan, S. P. Magleby, R. J. Lang, and L. L. Howell, "A Preliminary Process for Origami-Adapted Design," in *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2015: American Society of Mechanical Engineers, pp. V05BT08A053-V05BT08A053.
- [12] M. Stern, M. B. Pinson, and A. Murugan, "The Complexity of Folding Self-folding Origami," *Physical Review X*, vol. 7, no. 4, p. 041070, 2017.
- [13] R. V. Martinez, C. R. Fish, X. Chen, and G. M. Whitesides, "Elastomeric Origami: Programmable Paper-elastomer Composites as Pneumatic Actuators," *Advanced functional materials*, vol. 22, no. 7, pp. 1376-1384, 2012.
- [14] E. Hawkes *et al.*, "Programmable Matter by Folding," *Proceedings of the National Academy of Sciences*, vol. 107, no. 28, pp. 12441-12445, 2010.
- [15] T. Chen, O. R. Bilal, R. J. Lang, C. Daraio, and K. Shea, "Autonomous Deployment of a Solar Panel Using Elastic Origami and Distributed Shape-Memory-Polymer Actuators," *Physical Review Applied*, vol. 11, no. 6, p. 064069, 2019.
- [16] K. C. Francis, L. T. Rupert, R. J. Lang, D. C. Morgan, S. P. Magleby, and L. L. Howell, "From Crease Pattern to Product: Considerations to Engineering Origami-adapted Designs," in *ASME 2014 international design engineering technical conferences and computers and information in engineering conference*, 2015: American Society of Mechanical Engineers Digital Collection.
- [17] S. A. Zirbel *et al.*, "Accommodating Thickness in Origami-Based Deployable Arrays," *Journal of Mechanical Design*, vol. 135, no. 11, p. 111005, 2013.
- [18] J. Morgan, S. P. Magleby, and L. L. Howell, "An Approach to Designing Origami-Adapted Aerospace Mechanisms," *Journal of Mechanical Design*, vol. 138, no. 5, p. 052301, 2016.

- [19] E. D. Demaine, M. L. Demaine, and J. S. B. Mitchell, "Folding Flat Silhouettes and Wrapping Polyhedral Packages: New Results in Computational Origami," *Computational Geometry*, vol. 16, no. 1, pp. 3-21, 2000.
- [20] B. Liu *et al.*, "Topological Kinematics of Origami Metamaterials," *Nature Physics*, vol. 14, no. 8, p. 811, 2018.
- [21] M. Schenk and S. D. Guest, "Geometry of Miura-Folded Metamaterials," *Proceedings of the National Academy of Sciences*, vol. 110, no. 9, pp. 3276-3281, 2013.
- [22] D. Han, S. Pal, J. Nangreave, Z. Deng, Y. Liu, and H. Yan, "DNA Origami with Complex Curvatures in Three-dimensional Space," *Science*, vol. 332, no. 6027, pp. 342-346, 2011.
- [23] K. Kuribayashi *et al.*, "Self-deployable Origami Stent Grafts as a Biomedical Application of Ni-rich TiNi Shape Memory Alloy Foil," *Materials Science and Engineering: A*, vol. 419, no. 1-2, pp. 131-137, 2006.
- [24] E. A. Peraza-Hernandez, D. J. Hartl, R. J. Malak Jr, and D. C. Lagoudas, "Origami-Inspired Active Structures: A Synthesis and Review," *Smart Materials and Structures*, vol. 23, no. 9, p. 094001, 2014.
- [25] B. J. Edmondson, L. A. Bowen, C. L. Grames, S. P. Magleby, L. L. Howell, and T. C. Bateman, "Oriceps: Origami-Inspired Forceps," in *ASME 2013 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, Snowbird, Utah, USA, 2013: American Society of Mechanical Engineers, pp. V001T01A027-V001T01A027.
- [26] E. Vander Hoff, D. Jeong, and K. Lee, "OrigamiBot-I: A Thread-actuated Origami Robot for Manipulation and Locomotion," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014: IEEE, pp. 1421-1426.
- [27] S. Mintchev, L. Daler, G. L'Eplattenier, L. Saint-Raymond, and D. Floreano, "Foldable and Self-Deployable Pocket Sized Quadrotor," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015: IEEE, pp. 2190-2195.
- [28] S. M. Felton, M. Tolley, E. D. Demaine, D. Rus, and R. J. Wood, "A Method for Building Self-folding Machines," *Science*, vol. 345, no. 6197, pp. 644-646, 2014.
- [29] E. Morris, D. A. McAdams, and R. Malak, "The State of the Art of Origami-inspired Products: A Review," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016: American Society of Mechanical Engineers, p. V05BT07A014.
- [30] T. Tachi, "Geometric Considerations for the Design of Rigid Origami Structures," in *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium*, 2010, vol. 12, no. 10, pp. 458-460.
- [31] L. Wilson, S. Pellegrino, and R. Danner, "Origami Sunshield Concepts for Space Telescopes," in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013, p. 1594.
- [32] M. C. Natori, N. Katsumata, H. Yamakawa, H. Sakamoto, and N. Kishimoto, "Conceptual Model Study Using Origami for Membrane Space Structures. ," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013: American Society of Mechanical Engineers, p. V06BT07A047.
- [33] A. Hanaor and R. Levy, "Evaluation of Deployable Structures for Space Enclosures," *International Journal of Space Structures*, vol. 16, no. 4, pp. 211-229, 2001.

- [34] G. Scholar. [Online], accessed in October 2019, Available: [https://scholar.google.ch/scholar?hl=de&as\\_sdt=0%2C5&q=origami&btnG=](https://scholar.google.ch/scholar?hl=de&as_sdt=0%2C5&q=origami&btnG=)
- [35] 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*, vol. 124, pp. 26-45, 2017.
- [36] A. R. Diaz, "Origami Folding and Bar Frameworks," in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2014: American Society of Mechanical Engineers, p. V05BT08A031.
- [37] J. Cai, Z. Qian, C. Jiang, J. Feng, and Y. Xu, "Mobility and Kinematic Analysis of Foldable Plate Structures Based on Rigid Origami," *Journal of Mechanisms and Robotics*, vol. 8, no. 6, p. 064502, 2016.
- [38] M. Schenk and S. D. Guest, "Origami Folding: A Structural Engineering Approach," in *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, 2011: CRC Press, Boca Raton, FL, pp. 291-304.
- [39] J. Cai, Z. Ren, Y. Ding, X. Deng, Y. Xu, and J. Feng, "Deployment Simulation of Foldable Origami Membrane Structures," *Aerospace Science and Technology*, vol. 67, pp. 343-353, 2017.
- [40] T. Tachi, "Simulation of Rigid Origami," presented at the Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education, 2009.
- [41] K. Fuchi, P. R. Buskohl, J. J. Joo, G. W. Reich, and R. A. Vaia, "Topology Optimization for Design of Origami-Based Active Mechanisms," in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2014: American Society of Mechanical Engineers, p. V05BT08A049.
- [42] K. Fuchi and A. R. Diaz, "Origami Design by Topology Optimization," *Journal of Mechanical Design*, vol. 135, no. 11, p. 111003, 2013.
- [43] A. S. Gillman, K. Fuchi, and P. R. Buskohl, "Discovering Sequenced Origami Folding Through Nonlinear Mechanics and Topology Optimization," *Journal of Mechanical Design*, vol. 141, no. 4, p. 041401, 2019.
- [44] R. J. Lang, "A Computational Algorithm for Origami Design," in *Proceedings of the twelfth annual symposium on Computational geometry*, 1996: ACM, pp. 98-105.
- [45] R. J. Lang and L. L. Howell, "Rigidly Foldable Quadrilateral Meshes from Angle Arrays," *Journal of Mechanisms and Robotics*, vol. 10, no. 2, p. 021004, 2018.
- [46] T. Tachi, "Origamizing Polyhedral Surfaces," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 2, pp. 298-311, 2009.
- [47] T. Tachi, "Freeform Variations of Origami," *J. Geom. Graph*, vol. 14, no. 2, pp. 203-215, 2010.
- [48] T. Tachi, "Designing Freeform Origami Tessellations by Generalizing Resch's Patterns," *Journal of Mechanical Design*, vol. 135, no. 11, p. 111006, 2013.
- [49] B. An, N. Benbernou, E. D. Demaine, and D. Rus, "Planning to Fold Multiple Objects from a Single Self-folding Sheet," *Robotica*, vol. 29, no. 1, pp. 87-102, 2011.
- [50] Y. Chen, J. Feng, and Y. Liu, "A Group-Theoretic Approach to the Mobility and Kinematic of Symmetric Over-constrained Structures," *Mechanism and Machine Theory*, vol. 105, pp. 91-107, 2016.
- [51] Y. Chen, P. Sareh, J. Yan, A. S. Fallah, and J. Feng, "An Integrated Geometric-Graph-Theoretic Approach to Representing Origami Structures and Their Corresponding Truss Frameworks," *Journal of Mechanical Design*, vol. 141, no. 9, p. 091402, 2019.

- [52] H. Yu, Z. Guo, and J. Wang, "A Method of Calculating the Degree of Freedom of Foldable Plate Rigid Origami with Adjacency Matrix," *Advances in Mechanical Engineering*, vol. 10, no. 6, pp. 1-20, 2018.
- [53] L. H. Dudte, E. Vouga, T. Tachi, and L. Mahadevan, "Programming Curvature Using Origami Tessellations," *Nature materials*, vol. 15, no. 5, p. 583, 2016.
- [54] K. Miura, "Method of Packaging and Deployment of Large Membranes in Space," *The Institute of Space and Astronautical Science Report*, vol. 618, p. 1, 1985.
- [55] K. Miura and M. C. Natori, "2-D Array Experiment on Board a Space Flyer Unit," *Space Solar Power Review*, vol. 5, no. 4, pp. 345-356, 1985.
- [56] E. T. Filipov, G. H. Paulino, and T. Tachi, "Origami Tubes with Reconfigurable Polygonal Cross-Sections," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 472, no. 2185, p. 20150607, 2016.
- [57] T. Tachi and K. Miura, "Rigid-Foldable Cylinders and Cells," *Journal of the international association for shell and spatial structures*, vol. 53, no. 4, pp. 217-226, 2012.
- [58] J. L. Silverberg *et al.*, "Using Origami Design Principles to Fold Reprogrammable Mechanical Metamaterials," *Science*, vol. 345, no. 6197, pp. 647-650, 2014.
- [59] J. Cai, X. Deng, Y. Xu, and J. Feng, "Motion Analysis of a Foldable Barrel Vault Based on Regular and Irregular Yoshimura Origami," *Journal of Mechanisms and Robotics*, vol. 8, no. 2, p. 021017, 2016.
- [60] D. A. Huffman, "Curvature and Creases: A Primer on Paper," *IEEE Transactions on computers*, no. 10, pp. 1010-1019, 1976.
- [61] T. A. Evans, R. J. Lang, S. P. Magleby, and L. L. Howell, "Rigidly Foldable Origami Gadgets and Tessellations," *Royal Society open science*, vol. 2, no. 9, p. 150067, 2015.
- [62] Z. Abel *et al.*, "Rigid Origami Vertices: Conditions and Forcing Sets," *arXiv preprint arXiv:1507.01644*, 2015.
- [63] S. Waitukaitis, R. Menaut, B. G. Chen, and M. van Hecke, "Origami Multistability: From Single Vertices to Metasheets," *Physical review letters*, vol. 114, no. 5, p. 055503, 2015.
- [64] M. Bern and B. Hayes, "The Complexity of Flat Origami," in *SODA*, 1996, vol. 96, pp. 175-183.
- [65] G. Gogu, "Mobility of Mechanisms: A Critical Review," *Mechanism and Machine Theory*, vol. 40, no. 9, pp. 1068-1097, 2005.
- [66] R. J. Lang, K. A. Tolman, E. B. Crampton, S. P. Magleby, and L. L. Howell, "A Review of Thickness-Accommodation Techniques in Origami-Inspired Engineering," *Applied Mechanics Reviews*, vol. 70, no. 1, p. 010805, 2018.
- [67] E. D. Demaine and M. L. Demaine, "Recent Results in Computational Origami," in *Origami3: Third International Meeting of Origami Science, Mathematics and Education*, 2002, pp. 3-16.
- [68] E. T. Filipov, T. Tachi, and G. H. Paulino, "Origami Tubes Assembled into Stiff, yet Reconfigurable Structures and Metamaterials," *Proceedings of the National Academy of Sciences*, vol. 112, no. 40, pp. 12321-12326, 2015.
- [69] X. Liu, S. Yao, B. S. Cook, M. M. Tentzeris, and S. V. Georgakopoulos, "An Origami Reconfigurable Axial-mode Bifilar Helical Antenna," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 12, pp. 5897-5903, 2015.
- [70] H. Yasuda, C. Chong, E. G. Charalampidis, P. G. Kevrekidis, and J. Yang, "Formation of Rarefaction Waves in Origami-based Metamaterials," *Physical review E*, vol. 93, no. 4, p. 043004, 2016.

- [71] J. Cagan, M. I. Campbell, S. Finger, and T. Tomiyama, "A Framework for Computational Design Synthesis: Model and Applications," *Journal of Computing and Information Science in Engineering*, vol. 5, no. 3, pp. 171-181, 2005.
- [72] N. Watanabe and K. Kawaguchi, "The Method for Judging Rigid Foldability," *Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education*, pp. 165-174, 2009.
- [73] T. Ida and H. Takahashi, "Origami Fold as Algebraic Graph Rewriting," in *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009: ACM, pp. 1132-1137.
- [74] D. J. Balkcom and M. T. Mason, "Robotic Origami Folding," *The International Journal of Robotics Research*, vol. 27, no. 5, pp. 613-627, 2008.
- [75] J. G. De Jalon and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. Springer Science & Business Media, 2012.
- [76] G. M. Hallihan, H. Cheong, and L. H. Shu, "Confirmation and Cognitive Bias in Design Cognition," in *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013: American Society of Mechanical Engineers Digital Collection, pp. 913-924.
- [77] A. T. Purcell and J. S. Gero, "Design and Other Types of Fixation," *Design studies*, vol. 17, no. 4, pp. 363-383, 1996.
- [78] Y. Bar-Yam, "When Systems Engineering Fails - Toward Complex Systems Engineering," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, 2003, vol. 2: IEEE, pp. 2021-2028.
- [79] E. K. Antonsson and J. Cagan, *Formal Engineering Design Synthesis*. Cambridge University Press, 2005.
- [80] J. Rogers, Y. Huang, O. G. Schmidt, and D. H. Gracias, "Origami Mems and Nems," *Mrs Bulletin*, vol. 41, no. 2, pp. 123-129, 2016.
- [81] C. D. Onal, M. T. Tolley, R. J. Wood, and D. Rus, "Origami-inspired Printed Robots," *IEEE/ASME transactions on mechatronics*, vol. 20, no. 5, pp. 2214-2221, 2014.
- [82] A. P. Thrall and C. P. Quaglia, "Accordion Shelters: A Historical Review of Origami-like Deployable Shelters Developed by the US Military," *Engineering structures*, vol. 59, pp. 686-692, 2014.
- [83] K. Seymour *et al.*, "Origami-Based Deployable Ballistic Barrier," 2018: Proceedings of the 7th International Meeting on Origami in Science ....
- [84] S. Li *et al.*, "A Vacuum-driven Origami "Magic-ball" Soft Gripper," in *Proceedings of the 2019 IEEE International Conference on Robotics and Automation*, 2019: Institute of Electrical and Electronics Engineers (IEEE).
- [85] K. Miura, "A Note on Intrinsic Geometry of Origami," *Research of Pattern Formation, KTK Scientific Publishers, Tokyo, Japan*, pp. 91-102, 1989.
- [86] Z. He and S. D. Guest, "On Rigid Origami II: Quadrilateral Creased Papers," *arXiv preprint arXiv:1804.06483*, 2018.
- [87] R. J. Lang, S. P. Magleby, and L. L. Howell, "Single-degree-of-freedom Rigidly Foldable Origami Flashers," in *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2015: American Society of Mechanical Engineers, pp. V05BT08A043-V05BT08A043.

- [88] S. Waitukaitis and M. van Hecke, "Origami Building Blocks: Generic and Special Four-Vertices," *Physical Review E*, vol. 93, no. 2, p. 023003, 2016.
- [89] A. A. Evans, J. L. Silverberg, and C. D. Santangelo, "Lattice Mechanics of Origami Tessellations," *Physical Review E*, vol. 92, no. 1, p. 013205, 2015.
- [90] J. Cai, X. Deng, Y. Zhang, J. Feng, and Y. Zhou, "Folding Behavior of a Foldable Prismatic Mast With Kresling Origami Pattern," *Journal of Mechanisms and Robotics*, vol. 8, no. 3, p. 031004, 2016.
- [91] T. A. Evans, R. J. Lang, S. P. Magleby, and L. L. Howell, "Rigidly Foldable Origami Twists," *Origami 6: Sixth International Meeting of Origami Science, Mathematics, and Education*, no. 1, pp. 119-130, 2015.
- [92] H. Feng, R. Peng, J. Ma, and Y. Chen, "Rigid Foldability of Generalized Triangle Twist Origami Pattern and Its Derived 6R Linkages," *Journal of Mechanisms and Robotics*, vol. 10, no. 5, p. 051003, 2018.
- [93] T. Tachi, "Freeform Rigid-Foldable Structure Using Bidirectionally Flat-Foldable Planar Quadrilateral Mesh," *Advances in architectural geometry 2010*, pp. 87-102, 2010.
- [94] S. Liu, W. Lv, Y. Chen, and G. Lu, "Deployable Prismatic Structures with Rigid Origami Patterns," *Journal of Mechanisms and Robotics*, vol. 8, no. 3, p. 031002, 2016.
- [95] S. M. Belcastro and T. C. Hull, "A Mathematical Model for Non-flat Origami," in *Proceedings of the 3rd International Meeting of Origami Mathematics, Science, and Education*, 2002, pp. 39-51.
- [96] T. C. Hull, "The Combinatorics of Flat Folds: A Survey," in *Third International Meeting of Origami Science*, 2002, pp. 29-38.
- [97] T. Kawasaki, "On the Relation Between Mountain-Creases and Valley-Creases of a Flat Origami," in *Origami 1: Proceedings of the First International Meeting of Origami Science and Technology*, 1991, 1991.
- [98] J. Justin, "Mathematics of Origami, part 9," *British Origami*, vol. 118, pp. 28-30, 1986.
- [99] T. Tachi, "Generalization of Rigid-Foldable Quadrilateral-Mesh Origami," *Journal of the International Association for Shell and Spatial Structures*, vol. 50, no. 3, pp. 173-179, 2009.
- [100] J. Cai, Q. Zhang, J. Feng, and Y. Xu, "Modeling and Kinematic Path Selection of Retractable Kirigami Roof Structures," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 4, pp. 352-363, 2019.
- [101] Z. Xi and J.-M. Lien, "Folding Rigid Origami with Closure Constraints," in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2014: American Society of Mechanical Engineers, pp. V05BT08A052-V05BT08A052.
- [102] L. Zimmermann, T. Stanković, and K. Shea, "Finding Rigid Body Modes of Rigid-Foldable Origami Through the Simulation of Vertex Motion," in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2017: American Society of Mechanical Engineers, pp. V05BT08A046-V05BT08A046.
- [103] S. Pellegrino and C. R. Calladine, "Matrix Analysis of Statically and Kinematically Indeterminate Frameworks," *International Journal of Solids and Structures*, vol. 22, no. 4, pp. 409-428, 1986.
- [104] K. Miura, "The Science of Miura-Ori: A Review," *Origami*, vol. 4, pp. 87-100, 2009.
- [105] B. Kresling, "Origami-structures in Nature: Lessons in Designing "Smart" Materials," *MRS Online Proceedings Library Archive*, vol. 1420, 2012.



- [106] L. Zimmermann, K. Shea, and T. Stanković, "Origami Sensitivity - On the Influence of Vertex Geometry," in *7th International Meeting on Origami in Science, Mathematics and Education (7OSME)*, 2018, pp. p. 1087-1102.
- [107] N. Karmarkar, "A New Polynomial-time Algorithm for Linear Programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 1984: ACM, pp. 302-311.
- [108] T. Möller, "A Fast Triangle-Triangle Intersection Test," *Journal of graphics tools*, vol. 2, no. 2, pp. 25-30, 1997.
- [109] R. J. Lang, *Origami In Action: Paper Toys That Fly, Flag, Gobble and Inflate!* Macmillan, 1997.
- [110] J. Gleick, *Chaos: Making a New Science*. Open Road Media, 2011.
- [111] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
- [112] L. A. Bowen, C. L. Grames, S. P. Magleby, L. L. Howell, and R. J. Lang, "A Classification of Action Origami as Systems of Spherical Mechanisms," *Journal of Mechanical Design*, vol. 135, no. 11, p. 111008, 2013.
- [113] K. A. Dowland and J. M. Thompson, "Simulated Annealing," *Handbook of natural computing*, pp. 1623-1655, 2012.
- [114] L. Zimmermann, K. Shea, and T. Stanković, "A Heuristic Algorithm for Rigid Foldability," *Journal of Mechanisms and Robotics*, vol. 11, no. 3, p. 031004, 2019.
- [115] P. Sareh and S. D. Guest, "Design of Isomorphic Symmetric Descendants of the Miura-ori," *Smart Materials and Structures*, vol. 24, no. 8, p. 085001, 2015.
- [116] L. Zimmermann and T. Stanković, "Rigid and Flat Foldability of a Degree-Four Vertex in Origami," *Journal of Mechanisms and Robotics*, vol. 12, no. 1, 2019.
- [117] I. Streinu and W. Whiteley, "Single-Vertex Origami and Spherical Expansive Motions," in *Japanese Conference on Discrete and Computational Geometry*, 2004: Springer, pp. 161-173.
- [118] Y. Chen, R. Peng, and Z. You, "Origami of Thick Panels," *Science*, vol. 349, no. 6246, pp. 396-400, 2015.
- [119] M. Savage and A. S. Hall, "Unique Descriptions of All Spherical Four-Bar Linkages," *Journal of Engineering for Industry*, vol. 92, no. 3, pp. 559-563, 1970.
- [120] A. P. Murray and P. Larochelle, "A Classification Scheme for Planar 4R, Spherical 4R, and Spatial RCCC Linkages to Facilitate Computer Animation," presented at the ASME 1998 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 1998.
- [121] L. Zimmermann, K. Shea, and T. Stanković, "Conditions for Rigid and Flat Foldability of Degree-n Vertices in Origami," *Journal of Mechanisms and Robotics*, 2019, doi: doi:10.1115/1.4045249.
- [122] S. Kamrava, R. Ghosh, Y. Yang, and A. Vaziri, "Slender Origami with Complex 3D Folding Shapes," *EPL (Europhysics Letters)*, vol. 124, no. 5, p. 58001, 2018.
- [123] S. Kamrava, D. Mousanezhad, S. M. Felton, and A. Vaziri, "Programmable Origami Strings," *Advanced Materials Technologies*, vol. 3, no. 3, p. 1700276, 2018.
- [124] A. Ramsay and R. D. Richtmyer, *Introduction to Hyperbolic Geometry*. Springer Science & Business Media, 2013.
- [125] H.-J. Kreowski, R. Klempien-Hinrichs, and S. Kuske, "Some Essentials of Graph Transformation," *Recent advances in formal languages and applications*, vol. 25, pp. 229-254, 2006.

- [126] C. Königseder, T. Stanković, and K. Shea, "Improving Design Grammar Development and Application through Network-based Analysis of Transition Graphs," *Design Science*, vol. 2, 2016.
- [127] R. Storn, "On the Usage of Differential Evolution for Function Optimization," in *Proceedings of North American Fuzzy Information Processing*, 1996: IEEE, pp. 519-523.
- [128] J. M. Jiménez, G. Alvarez, J. Cardenal, and J. Cuadrado, "A Simple and General Method for Kinematic Synthesis of Spatial Mechanisms," *Mechanism and Machine Theory*, vol. 32, no. 3, pp. 323-341, 1997.
- [129] T. Ma, J. Chen, and C. Xiao, "Constrained Generation of Semantically Valid Graphs via Regularizing Variational Autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 7113-7124.
- [130] E. L. Lawler and D. E. Wood, "Branch-and-bound Methods: A Survey," *Operations research*, vol. 14, no. 4, pp. 699-719, 1966.
- [131] T. C. Hull and M. T. Urbanski, "Rigid Foldability of the Augmented Square Twist," *arXiv preprint arXiv:1809.04899*, 2018.
- [132] C. Ware, *Information Visualization: Perception for Design*. Elsevier, 2012.
- [133] A. Glassner, "Origami Platonic Solids," *IEEE Computer Graphics and Applications*, vol. 16, no. 4, pp. 85-91, 1996.
- [134] H. Yu, Z. Guo, and J. Wang, "A Method of Calculating the Degree of Freedom of Foldable Plate Rigid Origami with Adjacency Matrix," *Advances in Mechanical Engineering*, vol. 10, no. 6, p. 1687814018779696, 2018.

# Curriculum Vitae

## PERSONAL DATA

Name Luca Zimmermann  
Date of Birth 04.02.1991  
Place of Birth Zurich  
Citizen of Zurich

## EDUCATION

**Doctor of Science**, Mechanical Engineering 2020  
ETH Zurich, Switzerland  
Advisor: Prof. Kristina Shea

**Master of Science**, Mechanical Engineering 2015  
ETH Zurich, Switzerland

**Bachelor of Science**, Mechanical Engineering 2013  
ETH Zurich, Switzerland

## TEACHING

Engineering Design Optimization, ETH Zurich 2017-2018  
CAD and Technical Drawing, ETH Zurich 2018  
Engineering Tools Courses:  
    Design Optimization and CAD 2014-2019  
    Computer-Aided Design Methods 2014-2019

## SUPERVISION

Vanoni, M., 2016, "Optimization of a Floorball Blade", Bachelor Thesis  
Radke, F., 2016, "Origami-adapted Folding Mechanism for Flettner Rotor", Bachelor Thesis  
Schwery, O., 2017, "Optimization-Based Simulation and Investigation of a New Folding Concept", Semester Thesis  
Schacher, B., 2018, "Design of Foldable Solar Panels for CubeSats", Bachelor Thesis  
Wuite, G., 2018, "Building a Prototype for an Actuated Flasher", Bachelor Thesis  
Mauz, F., 2018, "Automated Packaging for Serial Production of Additive Manufactured Keys", Semester Thesis

Guardiola, R., 2018, "Automated Generation of Keyheads for Additive Manufacturing", Bachelor Thesis

Torredimare, M., 2019, "Surface Approximation using Single Degree of Freedom Origami", Semester Thesis

## **INDUSTRY PROJECTS**

**UrbanAlps**, 3D Printing and CAD modeling of the Stealth Key, the "key of the future"

**KOVE**, Foldable umbrella for amniotic membrane defect

**Orbitare**, Foldable antenna for CubeSat

## **PUBLICATIONS**

Zimmermann, L., Chen, T., and Shea, K., 2017, "Generative Shape Design Using 3D Spatial Grammars, Simulation, and Optimization", *Design Computing and Cognition'16*, p. 279-297

Zimmermann, L., Stanković, T., and Shea, K., 2017, "Finding Rigid Body Modes of Rigid-Foldable Origami Through the Simulation of Vertex Motion", *Proceedings of the IDETC 2017*, V05BT08A046.

Zimmermann, L., Chen, T., and Shea, K., 2018, "A 3D, Performance-driven Generative Design Framework: Automating the Link from a 3D Spatial Grammar Interpreter to Structural Finite Element Analysis and Stochastic Optimization", *AI EDAM*, 32(2), p. 189-199.

Zimmermann, L., Shea, K., and Stanković, T., 2018, "Origami Sensitivity – On the Influence of Vertex Geometry", 7<sup>th</sup> International Meeting of Origami Science, Mathematics and Education (7OSME), p. 1087-1102.

Zimmermann, L., Shea, K., and Stanković, T., 2019, "A Heuristic Algorithm for Rigid Foldability", *Journal of Mechanisms and Robotics*, 11(3), 031004.

Zimmermann, L., and Stanković, T., 2020, "Rigid and Flat Foldability of a Degree-Four Vertex in Origami", *Journal of Mechanisms and Robotics*, 12(1), 011004.

Zimmermann, L., Shea, K., and Stanković, T., 2020, "Conditions for Rigid and Flat Foldability of Degree-n Vertices in Origami", *Journal of Mechanisms and Robotics*, 12(1), 011020.