

Riccardo Foschi

ORIGAMI **DESIGN STRATEGIES** **FOR ARCHITECTS** **AND DESIGNERS**

Analysis and Design
of Folded Surfaces
Through Algorithmic
Modelling

Bologna
University Press



alphabet **18**

Riccardo Foschi

ORIGAMI DESIGN STRATEGIES FOR ARCHITECTS AND DESIGNERS

Analysis and Design
of Folded Surfaces
Through Algorithmic
Modelling

Bologna
University Press

Il volume è tratto dalla tesi di dottorato *Algorithmic Modelling of Folded Surfaces. Analysis and Design of Folded Surfaces in Architecture and Manufacturing*, Alma Mater Studiorum - Università di Bologna, Dottorato di ricerca in Architettura, ciclo XXXI, depositata in AMSDottorato - Institutional Theses Repository (<http://amsdottorato.unibo.it/>)



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Progetto Open Access Consorzio Alfabeta

Il testo è stato sottoposto a peer review / This text has been peer reviewed

This work is licensed under a Creative Commons Attribution (CC) BY-NC-SA 4.0

This license allows you to reproduce, share and adapt the work, in whole or in part, for noncommercial purposes only, providing attribution is made to the authors (but not in any way that suggests that they endorse you or your use of the work). Attribution should include the following information:

Riccardo Foschi, *Origami Design Strategies for Architects and Designers. Analysis and Design of Folded Surfaces Through Algorithmic Modelling*, Bologna: Bologna University Press, 2022

Quest'opera è pubblicata sotto licenza Creative Commons (CC) BY-NC-SA 4.0

Questa licenza consente di riprodurre, condividere e adattare l'opera, in tutto o in parte, esclusivamente per scopi di tipo non commerciale, riconoscendo una menzione di paternità adeguata (non con modalità tali da suggerire che il licenziante avalli l'utilizzo dell'opera). La menzione dovrà includere le seguenti informazioni:

Riccardo Foschi, *Origami Design Strategies for Architects and Designers. Analysis and Design of Folded Surfaces Through Algorithmic Modelling*, Bologna: Bologna University Press, 2022

Fondazione Bologna University Press

Via Saragozza, 10

40123 Bologna

tel. (+39) 051 232882

fax (+39) 051 221019

www.buonline.com

ISSN 2724-0290

ISBN 979-12-5477-108-2

ISBN online 979-12-5477-109-9

Progetto grafico e impaginazione: Design People (Bologna)

Prima edizione: giugno 2022

ACKNOWLEDGEMENTS

This book is based on my PhD thesis, titled *Algorithmic Modelling of Folded Surfaces. Analysis and Design of Folded Surfaces in Architecture and Manufacturing*, discussed in 2019, which was possible thanks to the Department of Architecture of Bologna University, the Marco Polo grant and the School of Arts and Science of the Tokyo University. I want to thank all the persons that followed me in this journey into applied origami, in particular, the professors: Fabrizio Ivan Apollonio, Federico Fallavollita, Tomohiro Tachi that guided me during the PhD years, and the professor Yasushi Yamaguchi that allowed me to join his research team in Tokyo. I want also to thank professors Yoshinobu Miyamoto and Graziano Mario Valenti who revised the thesis precisely and gave me important suggestions to improve its form and clarity. Special thanks to all my PhD mates and research colleagues with whom I passed my PhD years sharing sweat and joy, in particular, Kai Suto for the profitable time we passed together at Komaba campus in Tokyo, without whom I could not achieve certain results on degree-4 vertices and prototyping. I also want to thank my dear friend Marco Mandia who helped me thinking with a programmer point of view and handling delicate mathematics and Carolina Gelsomini who helped me revising the text and was always by my side. Last but not least, I want to thank my family who helped me economically and humanly during my studies without whom the thesis, and thus this book, would not have been possible.

TABLE OF CONTENTS

CHAPTER 1	
AIMS, TOOLS AND BACKGROUND	11
1.1. Introduction	11
1.2. Research Field	12
1.3. Tools	14
1.3.1. Existing Software for Designing Origami	16
1.4. Background	19
1.4.1. Brief History of Origami	19
1.4.2. Origami in Education – Art, Design and Math	22
1.4.3. Math Meets Art – Most Known Methods to Design Origami	25
CHAPTER 2	
ORIGAMI-INSPIRED DESIGNS	29
2.1. Classification Criteria	30
2.2. Synoptic Tables	31
2.3. Synoptic Tables Data Analysis	45
2.4. Designing with Folded Surfaces – Critical Observations	48
CHAPTER 3	
DEFINITIONS AND THEOREMS	51
3.1. Fold Angle	51
3.1.1. Fold Angle Over Time – From Plot Analysis	51
3.2. Developability	55
3.3. Degree of Freedom (DOF)	57
3.4. Rigid-Foldability	58
3.4.1. Reciprocal Diagram to Judge the First-Order Rigid-Foldability	60
3.5. Flat-Foldability	61
3.5.1. Four Rules of Flat-Foldability – Kawasaki and Maekawa Conditions	61
3.5.2. Flat-Foldable Degree-4 Single Vertex – Relations Between Fold Angles	64
3.5.3. Flat-Foldable Degree-4 Single Vertex – Reciprocal Diagram and Fold Angles	66
3.5.4. Calculating k Through Reciprocal Diagram – Proved by Experimental Method	67
3.6. Non-Flat-Foldability	68

3.6.1. Non-Flat-Foldable Degree-4 Single Vertex – Huffman’s Formulations	69
3.6.2. The Blocking Crease	76
3.6.3. Understanding the Kinematics of a Non-Flat-Foldable Developable Degree-4 Vertex	77
3.6.4. First Blocking Crease in a Developable Degree-4 Vertex	79
3.6.5. Other Fold Angles at Blocked State – With the Spherical Law of Cosine	82
3.7. About the Relation Between Origami Functionality and Real Applications	83
CHAPTER 4	
CONSTRUCTIVE METHODS FOR SOLVING THE KINEMATICS OF ORIGAMI	85
4.1. Families of Folded Surfaces	85
4.2. Operative Tools	87
4.3. Analogy with Computer Programming and Terminology Clarification	87
4.3.1. Clustering and Nesting	87
4.3.2. Definition of “Algorithm”	88
4.4. Single Linear Crease	88
4.4.1. Single Linear Crease Between Equal Rectangular Faces, Two Edges Slide on Construction Plane – Intersecting Circles	89
4.4.2. Single Linear Crease Between Asymmetric Rectangular Faces, Two Edges Slide on Construction Plane – Intersecting Circles	91
4.4.3. Single Linear Crease Between Rectangular Faces, Crease on Construction Plane – Varying Fold Angle	91
4.4.4. Single Linear Crease Between Triangular Faces, Crease on Construction Plane – Varying Fold Angle	93
4.4.5. Single Linear Crease Between Triangular Faces, Two Edges Slide on Construction Plane – Intersecting Circles	94
4.4.6. Single Linear Crease Between Trapezoidal Faces, Two Edges Slide on the Construction Plane – Intersecting Circles	95
4.5. Patterns with Multiple (Non-Intersecting) Linear Creases	96
4.5.1. Straight Accordion – Array of “Single Linear Crease Between Rectangular Faces” Molecules	97
4.5.2. Straight Accordion Sliding on a Rail – Intersecting Circles	98
4.5.3. Straight Accordion on a Rail with Non-Uniform Fold Angle Distribution – Intersecting Circles and “Graph Mapper”	100
4.5.4. Accordion on Two Circular Rails – Intersecting Circles	101
4.5.5. Triangulated Accordion – Joining Multiple “Single Linear Crease Between Triangular Faces” Molecules	102
4.6. Single Degree-4 Vertex	105
4.6.1. Symmetric Reverse Fold – Reflecting a Single Linear Crease	106
4.6.2. Asymmetric Reverse Fold – Reflection and Collision Detection	107
4.6.3. Generic Degree-4 Vertex – Intersecting Cones	109
4.7. Multiple Degree-4 Vertices	111
4.7.1. Joining “Symmetric Reverse Fold” Molecules – Critical Observations About Global Rigid-Flat-Foldability	111
4.7.2. Joining “Asymmetric Reverse Fold” Molecules	114

4.7.3. Reverse Fold on Triangulated Accordion – Joining “Symmetric Reverse Fold” Molecules	115
4.7.4. The Miura Pattern – Planar Rectangular Array of “Symmetric Reverse Fold” Molecules – Intersecting Circle with Plane of Symmetry	119
4.7.5. The Sink Fold – Reflecting the Tip of a Degree-4 Vertex	121
4.8. Patterns with Single or Multiple Degree>4 Vertices	125
4.8.1. Degree>4 Vertices – Physical Simulation	126
4.8.2. Testing the Algorithm with Different Patterns	129
4.8.3. Limits of the Algorithm and Known Problems – Pop-Up and Pop-Down	131
CHAPTER 5	
PATTERN DESIGN FROM A GIVEN SHAPE	133
5.1. Lampshade – Vertices Extrusion and Reflection	133
5.2. Folded Facade – Vertices Extrusion from Reference Curved Rails	136
5.3. Building Envelope – Reflection of a Creased Developable Surface	137
5.4. Curve-Folded Table	139
5.4.1. Reflection of a Developable Curved Surface	139
5.4.2. Discretization of the Curved Crease	140
5.5. Conformable Corrugated Suspended Ceiling	142
5.5.1. Conformation of a Rigid Creased Surface to a Curved Surface	143
5.5.2. Conformation of a Flexible Miura-ori to a Curved Surface	145
5.5.3. Optimization of Supporting Cables and Anchor Points	147
5.5.4. Changing Shape to the Surface – Adjusting Cables Lengths	151
CHAPTER 6	
FABRICATION-AIMED DESIGNS	153
6.1. Known Origami Thickening Methods – State of the Art	153
6.1.1. Offset Panels	154
6.1.2. Hinge Shifting	155
6.1.3. Tapered Panels	155
6.1.4. Constant Thickness Attached Panels	156
6.1.5. Membrane Hinges	157
6.1.6. Rolling Contact or “SORCE” Technique	157
6.1.7. Strained Joint	158
6.1.8. Double Hinge	159
6.1.9. Symmetric Miura-ori Vertex by Shifted Hinges and Carved Panels	160
6.1.10. Slidable Hinges	160
6.1.11. Double Line	161
6.2. Case Study – One-DOF, Developable, Non-Flat Rigid-Foldable Ladder	162
6.2.1. Preliminary Paper Prototype and Digitalization	163
6.2.2. Fine-Tuning the Dimensions of the Ladder with Trigonometry	164
6.2.3. Thickening – “Offset Panels” Method	165
6.2.4. Thickening – “Tapered Panels” Method	166
6.2.5. Thickening – “Double Line” Method	168
6.2.6. Stability Problems and Possible Solutions	172

6.3. Case Study – One-DOF, Developable, Non-Flat Rigid-Foldable Chair 174

6.3.1. Preliminary Paper Prototype 174

6.3.2. Thickening – “Double Line” Method 176

6.3.3. Blocked Degree-4 Vertex – From a Non-Developable Corner of 3 Faces 178

6.3.4. Adjusting Shape and Dimensions to Improve Ergonomics and Stability 179

6.3.5. Human-Size Prototype of the Chair - Critical Observations 180

CONCLUSIONS 185

Notes 188

REFERENCES 191

GLOSSARY 199

AIMS, TOOLS AND BACKGROUND

1.1. Introduction

Both in the fields of manufacturing and architecture, origami is often taken as a reference for its kinetic properties, its elegant and geometric shapes, for its capability to rationalize the creative process following precise geometric rules and specific spatial references and its capacity to combine shape and motion in a functional or ornamental way. If the crease is replaced with a hinge and the paper with a panel of a rigid material, or the hands of the origami artist with a CNC machine, it is not hard to imagine the numerous possible applications of this art and technique. Dynamic facades, deployable structures, temporary shelters, portable furniture, retractile roofs, unfoldable boxes, are some examples of kinetic designs that can take advantage of origami strategies. Differently from structures with bars and panels, origami can be used to obtain continuous surfaces without assembling different parts, optimizing the constructive process, the transportation, and the cost, at the expense of the designing time, in fact designing with origami makes the shape and movement harder to control with the contemporary professional cad applications, typically used by designers, because they were not developed with origami in mind.

In the design process of such applied origami, it is very difficult for the designer to control the form to fit design contexts while preserving the necessary functionalities of the original patterns. Therefore, without sufficient knowledge or intelligent design systems, the result designs would end up in either just a mere copy and paste of an existing origami pattern or an 'origami-inspired' design which is not using the proprieties of origami in functional ways. (Demaine & Tachi, 2010)

In accordance with Tachi and Demaine's studies, it is observable that the cause of this lack of variety could be attributed to insufficient knowledge, or to the

inefficiency of the design tools. When designing kinetic structures with folded surfaces it is necessary to control both shape and motion at the same time without losing the developability property of the surfaces. Consequently, the use of mathematical or geometrical rules play an important role for the success of the project, which increase complexity and time consumption at the very beginning of the design stage.

This lack of efficient tools has been already pointed out by many researchers in the past few decades, they tried to reduce this gap by studying the mathematical implications of origami art or by developing specific computer applications, however theorems and formulations are often hardly directly applicable by designers and architects to the creative process, and the developed applications are still limited in number and usually not very well integrated into the typical workflow of a designer. In this book we propose several simplified methods to design origami-like geometries, using a synthetic approach based on geometric constructions typical of the descriptive geometry, applied with a parametrical node-based application (Grasshopper for McNeel Rhinoceros). Working with geometrical constructions and spatial references is more natural for most of the professionals that operate in the fields of manufacturing and architecture because it is related to the representation method typically used by architects and designers.

The variety of possibilities that origami offers is boundless, for this reason, the aim of the research that gave birth to this book was not to provide a specific command or a piece of software that performs specific tasks, because this would not provide extensive design freedom. Thus, this book presents a series of case studies and operative guidelines which will help optimizing the design process that involves origami geometries, while at the same time guaranteeing maximum design flexibility. Our targets are all those professionals interested in origami design but without a specific background in mathematics or computer science, thus the use of scripting and algebraic formulations will be limited as much as possible.

1.2. Research Field

The origami world for his capability to combine technique and art, static nature and dynamism, straight lines and curves, recursive patterns and sculpted figures, planar configurations and three-dimensional objects, stiffness and flexibility, is versatile and applicable in a vast number of fields like engineer-

ing, manufacturing, astronomy, medicine, chemistry, architecture, robotics, computer science, art, fashion, design. The application fields are innumerable as well as the researchers that use origami constructions to improve some aspect of their projects.

One of the most influential researchers in the field of applied origami is Tomohiro Tachi. He investigates the paper folding since the first decade of the 21st century (Tachi, n.d.-a). He developed some of the most versatile and powerful computer applications focused on the design of origami, his aim is to simplify the origami design process to make it accessible to a vaster number of designers. He developed several mathematical theorems about origami, and he demonstrated their usefulness applying them in many practical applications, such as the “Rigid-origami table” which folds and unfolds in a single rigid motion or the “Vault structure” designed with rigid-foldable curved tubular arches (cf. section 3.4 for more about rigid foldability). He also developed some techniques to thicken the zero-thickness study model while preserving the kinematics of the original pattern (Tachi, 2011b).

With similar aims and approaches, in the last few years, some researches in the “Sapienza” University of Rome in Italy explored the kinematic properties of origami from the point of view typical of the descriptive geometry, and they searched for solutions suitable to be applied to the field of kinetic architecture (Casale & Calvano, 2012; Casale *et al.*, 2013).

Erik Demaine, computer scientist, mathematician, artist and professor at MIT, has a pluriannual experience into origami science, he is nowadays one of the most active and influential theorists in the field of computational origami, he contributed to the development of some computer applications that solve some specific origami problems related to pattern design, rigid-foldability, flat-foldability and curve-folding.

Because computer applications for designing origami are lacking, many different researchers started developing their own digital tools, such as Tomohiro Tachi, Jun Mitani, Ke Liu and Glaucio H. Paulino, Zhonghua Xi, only to name a few (cf. section 1.3.1 for more about computer applications for origami). However, one of the first and most influential computer applications developer for origami design was Robert J. Lang.

Lang is considered one of the most important origami scientist and artist of all times. He contributed to refine and extend some of the most advanced techniques still used by scientists and artists to design complex origami patterns. He also worked on several projects for aerospace applications which is probably the most advanced frontier of this art and technique. For example, he contrib-

uted to the design of the “Eyeglass” telescope for Lawrence Livermore National Laboratory and to the “Starshade” project for NASA, which are respectively a foldable lens for a space-based telescope and an occulter for the sunlight that will be used to look for planets orbiting faraway stars (Feder, 2018). He also contributed developing an algorithm to optimize the air-bag flattening in collaboration with the EASi airbag company (Lang, 2015a).

Some other interesting uses of origami into practical applications are the “Origami-Based Deployable Ballistic Barrier” by Seymour *et al.* (Seymour *et al.*, 2018); the “Deployable Locomotive Fairing” designed to improve the aerodynamics of the locomotives by Tolman *et al.* (Tolman *et al.*, 2018); and the origami stent by Kuribayashi *et al.*, that facilitate the insertion of the stent inside the human body by folding it in a specific way (Kuribayashi *et al.*, 2006), only to name a few. All these applications were possible also thanks to Thomas Hull, Toshikazu Kawasaki, Humiaki Huzita, Koshiro Hatori, Koryo Miura and many others who contributed to set the basis and extend the fundamental theorems of the origami mathematics.

This book adds a small contribution in this vast landscape, it tries to narrow the gap between theories and applications, opening this field full of possibilities to all those professionals without a specific background in engineering, mathematics, and computer science, who want to use origami functionalities in their projects.

1.3. Tools

To model folded surfaces in a three-dimensional digital environment and to integrate them into the design of a building or a piece of furniture, it is highly preferred the use of computer applications which can exchange files with the software used by architects and designer. Less export/import operations are usually preferable. This is important to limit time consumption and conversion problems.

Parametric/procedural modelling/animation is the widespread approach that is usually used to control such type of complex geometries and integrate them into the projects while speeding up the shape-finding process at the same time. Traditional 3D modelling computer programs can produce parametrical models through the integrated scripting interface, which is usable by users that can script in the programming language that the software requires. This kind of applications are tricky to use, and the programming language usually changes

from software to software. Thus, they are usually harder to learn than the logic environment of a graphical parametric node-based modeller, such as Dynamo, a plug-in for Autodesk Revit (Autodesk, n.d.), or Grasshopper, a plug-in for Rhinoceros (Rutten, n.d.). This kind of applications converts code strings in visual “nodes” that can be connected reciprocally to compose complex algorithms for documentation, fabrication, coordination, simulation and analysis. The research of the tools has been carried out analysing the international professional and academic landscape. Ultimately, Grasshopper for McNeel Rhinoceros was chosen for the following reasons: it is integrated into applications already used by both architects and designers, it is one of the most versatile parametric node-based modelers available up to now, it is affordable, it is user-friendly, it has a lot of free add-ons that speed up different processes, it has an active online community, it has an efficient assistance, it allows to integrate missing nodes by scripting with common programming languages (Python, VB, C#). Although, any other analogous application can be used to achieve similar results, because of that the procedures will be explained in the most general way possible so that they can be easily transposed in a different software with different tools and interface.

According to the initial statement, the generative algorithms in this book will be carried out trying to limit as much as possible scripting and mathematical formulations, thus all the definitions will be based as much as possible on geometric constructive procedures, using visual references and geometric primitives as construction tools.

This approach is comparable to the “Synthetic method”. The synthetic method has been used for centuries by mathematicians and scientists as an alternative to the analytical method and it can be compared to the field known today as “Descriptive geometry” extensively studied and disseminated by the geometer and mathematician Gaspard Monge (1746-1818) (Cardone, 2017). From that time, where ruler and compass were the most used tools, the methods and the solutions are greatly improved. Riccardo Migliari *et al.* into the book *Geometria descrittiva* (Migliari, 2009a, 2009b) proposes several methods to solve some old and new problems of descriptive geometry with three-dimensional modelling applications. Today we can work into three-dimensional space, and we can move the point of view in space to verify spatial relations easier. We can also measure distances and angles without needing to project them into auxiliary planes, which does not only simplify the visualization of the problems, but it also may simplify the actual constructions needed to solve those problems in the first place. Furthermore, nowadays, the accuracy when drawing with this kind of applications is incredibly high, this gives to the synthetic method great pos-

sibilities when looking for new problems and new solutions. The researchers of the Roman school, to which Migliari belongs, studied for many decades, until nowadays, the synthetic method and its implications, and they use it to study and solve complex modern geometrical problems or to verify problems from the past (Carlevaris *et al.*, 2012; Fallavollita, 2008; Fallavollita & Salvatore, 2013; Migliari, 2008a, 2008b, 2012; Salvatore, 2012). An interesting contribution from the Roman school, based on the synthetic method but related to origami, is the book *Architettura delle superfici piegate* written by Andrea Casale and Graziano Mario Valenti (Casale *et al.*, 2013) who use geometric constructions applied with the computer to solve the kinematics of origami.

1.3.1. Existing Software for Designing Origami

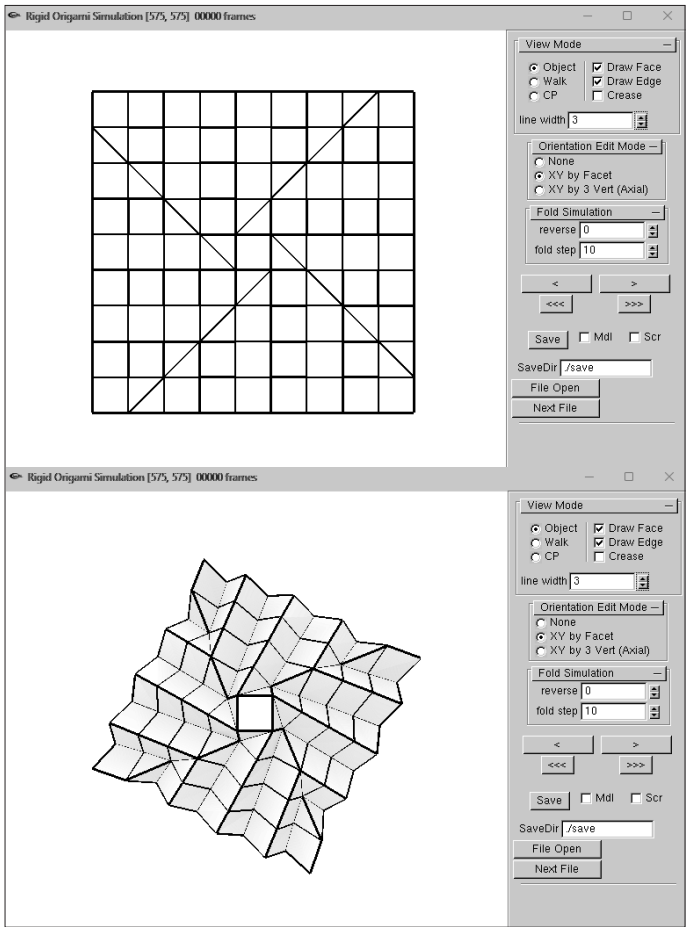


Figure 1 - Rigid Origami Simulator, by Tomohiro Tachi.

Architects, designers, engineers, and artists interested in origami, have until today developed their own tools for designing origami. Ron Resh was the pioneer of computer science applied to origami:

“[...] the design is a kind of feedback loop between the artist and the environment [...] the computer can really speed up this kind of loop, (design) and I think it greatly aid creativity [...] the excitement for me is to try to develop the computer as a medium for exploration and as a medium for expression.” (Resh, 1992).

As Resh stated, the computer does not only speed up the creative process, but it becomes a medium for expression and exploration. Kostas Terzidis shares the same point of view when he says:

“We shouldn’t consider the computer as an extension of the mind, but rather as a partner in the design process with fundamentally different aptitudes and ways to reason.” (Terzidis, 2006). And “Computers should be acknowledged not only as machines for imitating and appropriating what is understood but also as vehicles for exploring and visualizing what is not (yet) understood.” (Terzidis, 2009).

Thus, using the computer not only optimizes and speed up our design processes but also opens new unexpected possibilities. In fact, at the beginning of the twenty-first century, when computer applications for the design of origami started spreading, the complexity of the new origami models increased rapidly.

Robert J. Lang was the first to develop a stand-alone, publicly available, computer application to generate origami crease patterns called Treemaker (Lang, 2015c). This software was made for designing crease patterns using the circle river packing technique (cf. section 1.4.3 for a brief explanation of the circle river packing technique). In 1998 with version 4.0 this software was released outside the academic sector. Tomohiro Tachi developed several pieces of software to analyse and design origami-like geometries. Some of them are aimed to design non-flat foldable patterns or aimed to analyse the folding motion of any rigid-foldable pattern. The software developed until now by Tachi are: Freeform Origami, Origamizer, Rigid Origami Simulator (Figure 1), which allow the users to develop modify and analyse origami models with intuitive 2D and 3D interfaces (Tachi, n.d.-b). Jun Mitani developed: Oriref and Orirevo, made to design origami with reflections (Mitani & Igarashi, 2011) and revolutions (Mitani, 2009), and Oripa, which is a software to design planar origami patterns and returns their collapsed flat-folded configuration. All these apps can be found on Mitani’s official web page (Mitani, n.d.). Tess is a software aimed to design origami tessellations developed by Bateman (Bateman, n.d.). Pepakura is not properly an origami software for purists, but it is somehow related to origami, because it works with folds, other than cuts, to make 2D pattern starting from a generic three-dimensional mesh (Tamasoft, n.d.). Another interest-

ing recent application that returns the folding animation of a given origami pattern is the Origami Simulator by Amanda Ghassaei (Ghassaei, n.d.), which run directly on the internet browser. We want also to cite Merlin 2 by Ke Liu and Glucio H. Paulino which is a versatile powerful tool for the analysis of origami structures (Kawaguchi *et al.*, 2016; Liu & Paulino, 2018).

In the following table, we report some of the most important applications that concern origami and folding, at the present moment.

Table 1 - Origami computer applications.

Year	Software	Author
?	Pepakura	Tama Software
1998	Treemaker	Robert J. Lang
2003	Reference finder	Robert J. Lang
2007	Tess	Alex Bateman
2007	Rigid Origami Simulator	Tomohiro Tachi
2008	Origamizer	Tomohiro Tachi
2010	Freeform Origami	Tomohiro Tachi
2011	Oripa	Jun Mitani
2011	Oriref	Jun Mitani
2011	Orirevo	Jun Mitani
2011	Orirevo Morph	Jun Mitani
2013	Single Vertex Rigid Origami Simulator	Zhonghua Xi
2014	Origami Pattern Designer	Zhonghua Xi
2014	Tes Generator	Zhonghua Xi
2014	Rigid Origami Folder	Zhonghua Xi
2015	Origami Folder	Zhonghua Xi
2017	Origami Simulator	Amanda Ghassaei
2017	Merlin	Ke Liu and G. H. Paulino
2018	Merlin 2	Ke Liu and G. H. Paulino
2018	DeltaMod	Naoya Tsuruta

These applications (and many others not reported here), simplify the design of origami-inspired geometries and mechanisms to be used in architecture and manufacturing, but a deep theoretical origami knowledge is needed to properly use most of them. Furthermore, some applications often absolve only one task. They are useful tools, nevertheless, they require numerous exporting and importing operations to be able to use them into a real professional workflow. These file

conversions may cause loss of data while moving the model from one software to another, e.g. the mountain valley assignments, the folded or unfolded state, the folding animation, the overlapping sequence of the layers, the colour or the shader of the surface. Some researchers are already working trying to partially solve this problem through the creation of a file extension specific for origami. This innovation could revolutionize the design of these folded patterns. The file extension is called “.fold” from the GitHub repository by E. Demaine, it is defined as follows:

FOLD (Flexible Origami List Data-structure) is a file format (with extension .fold) for describing origami models: crease patterns, mountain/valley patterns, folded states, etc. Mainly, a FOLD file can store a mesh with vertices, edges, faces, and links between them, with optional 2D or 3D geometry, plus the topological stacking order of faces that overlap geometrically. A mesh can also easily store additional user-defined data. (Demaine, n.d.)

While we wait for the enhancing of the interoperability between these applications, the solution that we propose is trying to clarify how a designer needs to think while designing with origami and what strategies he needs to follow to make a creative and interesting origami-based design. To do so we will propose a variety of algorithms and case studies using the same parametric modelling software so that any designer would be able to work in the same platform without needing to switch software, saving time and preventing data loss.

1.4. Background

1.4.1. Brief History of Origami



Figure 2 - Shinto temples orned with zigzagging cut-and-folded paper streamers called “shide”. (Pictures courtesy of K. Toma and M. Winkler under the “Free to use Unsplash License”)

Origami has no certain origins. The paper degrades easily, for this reason, it is impossible to know who was the first who folded a piece of paper or when exactly origami has been invented. The implications of the origin of traditional origami on the field of origami applied to architecture and manufacturing are minimal because the origami as we know today is very different from the one practised in ancient times. Despite that, it is still interesting to see where it all started and in how many different fields origami played a more or less important role before becoming a reference for designers and engineers.

There are many theories about origami origins. Someone says that origami originated in China concurrently with the paper invention around 2000 years ago, this theory is based on the fact that many believe that paper was born in 105 BCE, when the Chinese official of Han dynasty, Cai Lun, wrote a document that explained the procedure to produce paper used at the time. Koshiro Hatori in his article *History of Origami in the East and the West before Interfusion* (Hatori, 2011) states that all these assumptions are wrong, because there is no evidence of origami from that period and, furthermore, the paper wasn't invented in China in that period. Hatori reports recent studies, by Imami Sakamoto, which dates high-quality foldable bark paper around 5000 years BCE, and there are proves of a similar type of rough paper found in different parts of the world at that time (Meso-America, Hawaii, Southeast Asia). Furthermore, even if they folded the paper in half or more it is hard to consider that as an actual origami. This philosophical observation about the number of folds needed to consider a folded sheet as a proper origami makes even harder trying to date its origins. This does not mean that an independent Chinese origami tradition does not exist. For example, the "Yuan bao" is a traditional Chinese origami representing a golden nugget which was invented by an unknown designer earlier than the tenth century CE when it was already a tradition folding and burn it at funerals (Mitchell, n.d.). There is also who believe that origami originated in Japan in the Heian era (circa 794-1192 CE). The theory is based on traditional anecdote where Abe-no Seimei took a piece of paper and he transmuted it into a real heron. However, even this hypothesis is not sufficient to prove that they were talking about origami as we know them nowadays, because, according to what Hatori reports, some version of the stories says that the heron was made by knotting the paper or drawing or cutting it instead of folding it. In addition, Hatori explains that the Japanese paper strips, "Shide" sometimes mounted on wooden sceptres called "Gohei", "Onbe" or "Heisoku", used in Shinto rituals and the paper dolls, "Hitogata", were not made of paper in ancient Japan, and they are not necessarily folded even now. The word "ori-gami", came from "oru"

meaning “to fold” and “kami” meaning “paper” or “divinity”. This leads us to think that there is a strict relationship between Japanese religion and the art of paper folding, but in ancient Japanese language the pronunciation of those words were different, so Hatori believe that it is hard to see a univocal connection even between the traditional origami and the Japanese religion.

The oldest unequivocal document about origami in Japan is a short poem composed by Ihara Saikaku in 1680, where he speaks about “Origami butterflies in Rosei’s dreams”. He refers to origami “Ocho” and “Mecho” which are male and female butterflies, Japanese people still use those folded paper models to ornate bottles at weddings. This means that origami was already deep-rooted in Japanese culture when the poem was written, in fact, the samurai warriors between 1603 and 1868 were supposed to fold wrapping paper, shaping it in a symmetric regular figure. Such type of folded figure named “Noshi” is probably dated between 1333-1573 and it was gifted as a token of good luck. What is surprising is that an older document reported by Vicente Palacios where we can recognize an origami boat was probably edited in Venice for the first time in the 13th century. It is the *Tractatus de Sphaera Mundi* by Giovanni Sacrobosco, according to Vicente, the image of the boat on the bottom has been found in the 1490 edition, but it could have been present even in an earlier edition. However, even in this case, Hatori discourages to take it as a certain clue because no written evidence of origami in Europe in that period has been found yet, and the picture could also have represented a simple stylized boat instead of an accurate origami boat.

Another European document that proves that origami was present in Europe in ancient times, which is probably unrelated to Japanese origami tradition, was

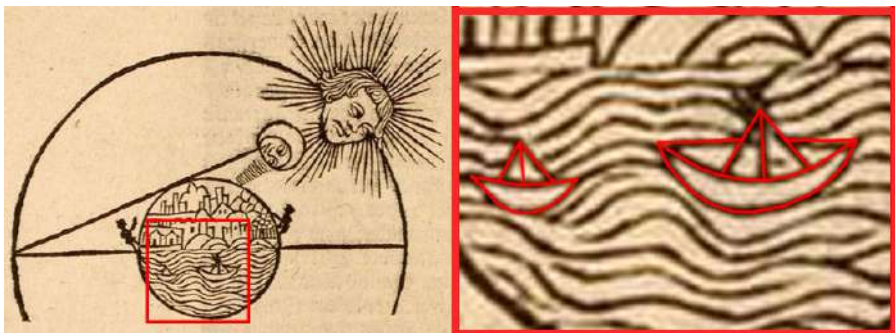


Figure 3 - *Tractatus de Sphaera Mundi* written by Giovanni Sacrobosco on the left, two figures similar to paper boats on the right.

the baptismal certificate which was folded in a way that is known today as the “double blintz base”, and it can be dated back to the 16th century before the protestant reformation. In addition, there are several examples of love letters dating back to the early nineteenth century folded in a similar way which could be related to an autonomous European tradition of paper folding.

What is almost certainly true is that there is not a univocal place or time where and when origami was born, but it probably had a concurrent diffusion in many different countries and ages because the paper by its nature invites to folding it. Nevertheless, it is still an open problem, which we will probably never be able to solve due to the lack of evidence. What is certain is that the origami, even if it could not be born exclusively in Japan, nowadays is considered strictly related to Japanese culture. The reason for this can be related, on the one hand, to the higher number of references to origami in the ancient and modern Japanese art compared to other countries; for example the first known book about ornamental origami is the Japanese book *Hiden Senbazuru Oriката* first published in 1797; and on the other hand, it can be related to the work of many Japanese artists who lived in 20th to 21st century such as Akira Yoshizawa (1911-2005) who is considered the father of modern origami and the one who redefined the graphical system which is used today to represent the folding procedures of origami, known as “Yoshizawa-Randlett system”. He probably created more than 50.000 original models, of which only a small amount was published in his 18 books. For his contribution, as an ambassador of the Japanese culture in the world, he has been awarded from Emperor Hirohito with the “Order of the Rising Sun”, which is the highest honour conferred in Japan.

1.4.2. Origami in Education – Art, Design and Math

In the past, origami was not only used for artistic or ceremonial purposes. Thanks to its intrinsic geometrical properties, it was often used for educational purposes. One of the first well-documented examples of origami used in classes was the experience of Joseph Albers who used origami as a tool to experience construction at Bauhaus in the 1920s. Quoting the artist Hans Beckmann words on his experience in Joseph Albers’s basic design course at Bauhaus:

I remember vividly the first day of Vorkurs, Josef Albers entered the room, carrying with him a bunch of newspapers. [...] he then addressed us saying: Ladies and gentlemen, we are poor, not rich. We cannot afford to waste materials or time. We have to make the most out of the least. All art starts with a material, and therefore we have first to investigate what our material can do. So, at the

beginning, we will experiment without aiming at making a product. At the moment, we prefer cleverness to beauty. [...] Our studies should lead to constructive thinking. [...] I want you now to take the newspapers [...] and try to make something out of them that is more than you have now. I want you to respect the material and use it in a way that makes sense – preserve its inherent characteristics. If you can do without tools like knives and scissors, and without glue, the better. (Roth *et al.*, 2013)

In the book *Geometric folding algorithms: linkages, origami, polyhedra* by Demaine and O’Rourke, the authors report an interesting and extensive research about the history of origami in math. They report a contribution precedent to the Bauhaus experience that used origami with educational purposes, which was the geometry essay by Rev. Dionysius Lardner written in 1840. This book illustrates several geometric concepts using paper folding. Furthermore, Sundara Row in 1893 wrote a text where origami was used as a tool to make geometrical constructions as an alternative to ruler and compass. These writings can also be considered as the firsts known contributions to origami in the field of mathematics, even if in these cases the origami is used as a tool and not as the focus of the study (Demaine & O’Rourke, 2007). In 1936 is dated the first known contribute signed by Margherita Piazzola Beloch, which considers origami as the focus of a research about mathematics. In this book, she starts the investigation of the origami axioms, which later will be investigated further by Humiaki Huzita, the Japanese-Italian mathematician who, in 1985, presented the first 6 of the 7 axioms which define the operations that can be made with a single piece of paper, folded with linear creases with no cuts and completed on a plane. Someone believed that the 7th axiom was discovered in 2002 by Koshiro Hatori, a Japanese folder who found a new type of single fold alignment which could not be attributed to any of the Huzita axioms. From that moment, the 7 axioms started to be known as “Huzita-Haori” axioms, but according to Robert J. Lang’s point of view, the seven axioms should have been named “Huzita-Justin axioms” (Lang, 2016), because in fact it turned out later that all the 7 axioms were already been completed by a French researcher Jacques Justin in 1989, who published the paper *Résolution par le pliage de l’équation du troisième degré et applications géométriques* in which he enumerated 7 possible combinations of one single fold alignments. This fact instilled the doubt in Lang that the axioms could have been not concluded, thus a few years later he proved their completeness mathematically (Lang, 2015b). The full set of axioms is reported in Table 2.

Table 2 - Huzita-Justin Axioms.

HUZITA-JUSTIN AXIOMS	
1	Given two points p_1 and p_2 we can fold a line connecting them.
2	Given two points p_1 and p_2 we can fold p_1 onto p_2
3	Given two lines l_1 and l_2 , we can fold line l_1 onto l_2
4	Given as point p_1 and a line l_1 , we can make a fold perpendicular to l_1 passing through the point p_1
5	Given two points p_1 and p_2 and a line l_1 , we can make a fold that places p_1 onto l_1 and passes through the point p_2
6	Given two points p_1 and p_2 and two lines l_1 and l_2 we can make a fold that places p_1 onto line l_1 and places p_2 onto line l_2
7	Given a points p_1 and two lines l_1 and l_2 , we can make a fold perpendicular to l_2 that places p_1 onto line l_1

For those who are interested to study further the origami axioms, more details can be found in: *Origami and geometric constructions* and *Huzita-Justin axioms* by Robert J. Lang (Lang, 2015b, 2016), *The mathematics of origami* by Sheri Yin (Yin, 2009), *Some results to the Huzita axioms* by H. R. Khademzadeh and H. Mazaheri (Khademzadeh & Mazaheri, 2007), *Résolution par le pliage de l'équation du troisième degré et applications géométriques* by Jacques Justin (Justin, 1989), and *Geometric folding algorithms: linkages, origami, polyhedra* by Erik D. Demaine and Joseph O'Rourke (Demaine & O'Rourke, 2007). A lot of studies about axioms, and about geometrical constructions that are possible thanks to the origami axioms, can be also found on Thomas Hull's web page and other publications by him (Hull, n.d., 2003a, 2003b, 2006).

Even if the seven axioms of origami are not easily directly applicable to practical designs in the field of engineering, manufacturing and architecture, they are the basics of the “mathematics of paper folding”. Starting from these basic axioms, the scientific community became more and more interested in origami mathematics. The studies were extended to many different problems, such as the flat-foldability, the definition of generalized techniques to design any shape only by folding, the degree of freedoms of a pattern and so on. There are many theorems about flat-foldability, extensively studied by Jun Maekawa, Toshikazu Kawasaki, Jacques Justin (cf. section 3). Later Thomas Hull continued their work on flat-foldability from the early nineties until today. On the contrary, there is not a large bibliography about non-flat foldable patterns, one of the few references is the paper written by David A. Huffman *Curvature and Creases: A*

Primer on Paper (Huffman, 1976). One of the first examples of computational origami is attributable to Ronald Resh. Between the late 1950s and early 1970s, he worked with paper folding both artistically and computationally. In the 70s he developed a computer program at the University of Huta that converted any space curve in a curve-folded edge (Schmidt & Stattmann, 2009).

Robert J. Lang developed around 1993 an algorithm, which became later a standalone software, to design crease patterns, this contribution determines a conjunction point between origami intended as an art and the origami intended as a technique, but probably the most important Lang's contribution is his book *Origami design secrets* (Lang, 2003, 2011). The first edition dates back to 2003 re-edited in 2011. His "magnum opus" on origami design methods was extended by another recent publication *Twists, Tilings and Tessellations. Mathematical methods for Geometric Origami* (Lang, 2018) which extends the previous publications with new methods to design tessellations, twists and corrugations. Erik Demaine asserted that "Lang's work may be viewed as the start of the recent trend to explore computational origami" (Demaine & O'Rourke, 2007).

The next frontier of the origami mathematics will probably be focused on the curved folding, which has already been approached by Tachi and Demaine *et al.* (Tachi, 2011a, 2013; Demaine *et al.*, 2015, 2018; Demaine *et al.*, 2011) but it still has a lot of open problems. Lang also asserted that he is interested in studying further this topic, which is still mostly unexplored.

1.4.3. Math Meets Art – Most Known Methods to Design Origami

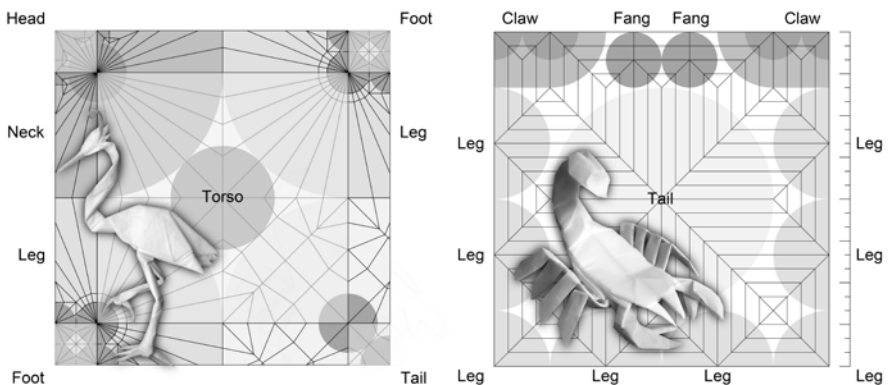


Figure 4 - Examples of crease patterns developed using 22.5 technique on the left (Egret by Riccardo Foschi); and box pleating technique on the right (Scorpion by Riccardo Foschi).

All the theorems developed since the second half of the twentieth century until today were not only a matter of mathematicians, engineers and scientists, also the artistic community started applying some of those theorems to increase their design possibilities. The result of that was a surprising increase in the complexity of the figurative origami models. This gave birth to a new artistic design approach based on mathematical rules.

Before this revolution, the traditional method to design origami usually consisted of a trial-and-error process. This method is still used by the most part of origami designers, and it starts by fixing the subject to design; then the designers try to get a schematic geometric figure by using their experience and a trial-and-error method. The geometrical base must be as much close to the desired subject as possible so that they can shape and sculpt it to achieve a more realistic look. The design method is focused on finding clear folding steps and reference easy points in spite of looking for an optimized crease pattern (CP).

Sometimes the process can also be reversed, the artist folds a random base without having in mind a particular subject searching for the inspiration while folding it, and only after folded the base he/she searches for a figure that could match the base and he/she starts shaping it accordingly.

The most recent mathematically-based approach is characterized by applying mathematical or graphical rules to draw an accurate crease pattern before even folding a single crease, only after the pattern is finished the artist will find the folding sequence. Sometimes there is not an easy step by step process to collapse complex patterns, and sometimes it is necessary to collapse it all at once making it way more difficult to fold than a traditional step-by-step sequence. The first approach is usually used for simple models, the second one is necessary for very complex models because the design time would be too long if approached with the trial-and-error method.

Ryujin by Kamiya Satoshi, for example, is one of the most complex models ever created. It is folded from a $2m \times 2m$ single square sheet of paper and represents an eastern style scaled dragon with several claws horns and fangs. It took Kamiya many months to design and fold its final version, and without a precise design strategy, and solid mathematical/geometrical basis, Kamiya would not have probably been able to design it.

Many different widespread mathematically-based approaches exist, some of them aim to optimize the paper usage, others aim to simplify the design process, others want to push to another level the possibilities of origami even if the method is not convenient for the folders, some of them aim to be as flexible and reliable as possible. The most known are explained in Lang's book *Origami*

design secrets (Lang, 2011), most of them come from the analysis of traditional techniques, enriched by Lang's experience. He explains how to design any origami figure, just by modifying a traditional known pattern, or by assembling different pieces of known patterns that he calls "tiles" or "molecules".

To understand how to mate two tiles it is necessary to understand the rivers and circle rules, which will not be explained in depth here. Suffice it to say that circles represent flaps and points, and rivers represent connectors between flaps and points in the folded geometric base model. To obtain a flat foldable origami composed by different tiles it is necessary to line up all the rivers and the circles of the adjacent tiles. The natural consequence of a tiles-based method (molecule method) is to investigate all the possible ways to arrange the tiles (thus the river and the circles) on the plane (the sheet of paper). The answer to this problem is a well-known geometrical problem called circle packing, which is defined as the study of the arrangement of circles on a given surface such that no overlapping occurs and so that all circles touch one another.

The tree theory explained by Lang is an evolution of the circle packing and molecule technique, and it consists in drawing a schematic figure where the lengths of the limbs are the lengths of the flaps, and thus, they are the radii of the circles which have to be arranged on the unfolded sheet as a guide to draw the crease pattern. Once drawn all the creases the verse of each crease must be assigned (valley or mountain) which can be done through the rules of flat foldability (cf. section 3.5). When connecting with creases the packed circles, the angles between creases are often odds, and thus it can be very hard to find references just by folding.

The box pleating technique became famous between origami artists to avoid this problem, due to its simplicity both in terms of designability and foldability. This technique was born to design box-like origami and evolved becoming a self-standing technique which can be used to design any kind of flat-foldable or non-flat-foldable origami. In the "box pleating" method, angles which are not multiple of 45° are not allowed thus it solves the problem of odd angles greatly simplifying the pre-creasing process. However, it partially limits the design freedom; Lang says "Because of their simple angles, box-pleated crease patterns can be much easier to develop linear folding sequences for. They come with a cost, however; not all circle patterns possess box-pleatable molecules." (Lang, 2011).

In architecture, manufacturing and engineering, the design processes differ from the ones used by artists. Usually, the shape-finding process starts from the context or it is aimed at finding particular movements to be applied in particular situations instead of finding a specific shape. The necessity to control the move-

ment makes the design process longer and harder. In these fields, it is not easy to identify a set of known approaches because they usually differ case by case. However, for sure the common ground is the use of computer applications and/or math, to generate, control, modify and analyse the behaviour of rigid-foldable surfaces.

ORIGAMI-INSPIRED DESIGNS

Origami existed for centuries, but it is from fairly recent times that it became a recurrent reference for designers and architects, however the examples of already existing objects and structures inspired to origami are already innumerable. So before to deep dive into the actual topic of this book, which is the development of algorithms to help designer designing origami with digital tools, it is important to have a clear idea of what it is already existing in this field, and use this knowledge as a starting point. So in the next sections there are collections of projects inspired from origami for a diversity of aspects. This extensive gathering has the aim to outline the general trend of the selected application fields. The projects were selected by filtering those which were from the fields of permanent architecture, temporary architecture, artistic installations, furniture and manufacturing, and fashion. The catalogue of projects does not in any way claim to provide a comprehensive and exhaustive cataloguing of all the existing origami-related projects because the number of suitable cases with regards to the assumed criteria would have been too wide to be listed in this context. Therefore, the collection here presented has to be considered as a limited list of references, but vast enough to be able to make some robust considerations. In accordance to all of these assumptions, the gathered projects will be subdivided into the 5 sub-groups: “Permanent Architecture”, “Temporary Architecture”, “Installations”, “Goods and Furniture”, “Fashion and Clothing” (Figure 5).



Figure 5 - Classification of projects into families based on the application fields, scale and functions. (Reference of the pictures in the next section)

2.1. Classification Criteria

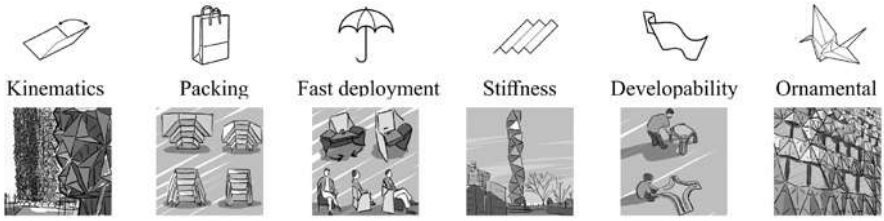





















Figure 6 - Origami-related characteristics outlined from the analysis of the projects.

All the projects selected are briefly presented in the next sections, gathered into synoptic tables where all their origami-related characteristics are synthesised visually with a set of icons (icons with relative meanings in Figure 6). The classes of origami-related functions are six: Kinematics, Packing, Fast Deployment, Stiffness, Developability, Ornamental. Every project can match with one or more characteristic. The Ornamental class includes all the designs which take inspiration from origami for aesthetical reasons. This class will be the only one described with 2 columns in the summary histograms (Figure 7, Figure 8): one enumerates the works inspired to origami while having other origami-related characteristics, the second enumerates the works inspired by origami exclusively for ornamental reasons. This distinction is important because all those projects that are related to origami only for ornamental reasons do not require to follow design strategies that make use of accurate origami rules. For “Stiffness” are intended those works which use folds to get a stronger structure. The “Kinematics” class includes those works that use origami rules to achieve specific movements. In the “Packing” class are included those works that exploit origami mechanisms to reduce their dimensions for transportation stocking or for space optimization. The “Fast deployment” class includes all those works where the origami mechanisms are used to pack and deploy rapidly the structure. The “Developability” class includes all those works which were fabricated by folding a flat sheet. The developability characteristic is often used to optimize the cutouts and the scraps during the production and to optimize the assembling time. The criteria adopted for this classification were as objective as possible, although some cases were hard to be assigned to one class univocally. Nevertheless, the objective of this analysis was not to extract a precise number of projects in one or the other class, but it was to trace a trend of the field, thus even if some projects may be not univocally classifiable, the high number of case studies considered will tend to globally flatten out eventual uncertainties, and the outlined trend of each group of projects will not lose relevance.

2.2. Synoptic Tables

	Permanent Architecture
	1 1923 - Hangars of the Orly Airport - Paris, FR - Eugène Freyssinet
	2 1953/1958 - Assembly Hall of the UNESCO Headquarters - Paris, FR - Pierluigi Nervi and Zehrfluss
	3 1958 - American Concrete Institute Building - Detroit, US - Minoru Yamasaki
	4 1962 - Steel Pre-Fab Houses - Palm Springs, CA, US - Donald Wexler
	5 1962 - USFA Cadet Chapel - El Paso, CO, US - Walter Netsch of Skidmore, Owings, & Merrill
	6 1963 - Miami Marine Stadium - Miami, US - Hilario Candela
	7 1966 - Shelter for sulphur factory - Pomezia, IT -Renzo Piano Studio
	8 1966-68 - St. Paulus Neuss Church - Mito, Ibaraki, JP - Fritz Schaller and Stefan Polonyi
	9 1973 - Teatro Regio di Torino - Torino, IT - Carlo Mollino
	10 1985 - Ernstings Warehouse - Coesfeld-Lette, DE - Santiago Calatrava
	11 1990 - Art Tower Mito - Mito, Ibaraki, JP - Arata Isozaki
	12 2000 - M-house - Gorman, California, USA - Michael Jantzen
	13 2002 - Meguro Persimmon Hall - Tokio, JP - Nihon Sekkei
	14 2002 - Rehearsal room in Tannhausen - Stuttgart, DE - Regina Schineis
	15 2003 - Bengt Sjostrom Starlight Theatre - Rockford, IL, US - Studio Gang Architects
	16 2007 - CaixaForum - Madrid, ES - Herzog & de Meuron
	17 2007 - Dom mieszkalny w Doktorgässchen - Niemcy, Augsburg, DE - Regina Schineis Architecten BDA
	18 2007 - Fuji television wangan studio - Tokyo, JP - Kajima design



Permanent Architecture

- | | |
|--|---|
| | 19 2007 - Monaco House - Melbourne VIC, AU - McBride Charles Ryan |
| | 20 2007 - Nestlé Chocolate Museum - Toluca de Lerdo, MX - Rojkind Arquitectos |
| | 21 2007 - Spertus Institute of Jewish Studies - Chicago, Illinois, US - Krueck & Sexton Architects |
| | 22 2008 - Klein Bottle house - Mornington Peninsula, VIC, AU - McBride Charles Ryan |
| | 23 2008 - Café-Restaurant OPEN - Amsterdam, NL - De Architekten Cie |
| | 24 2008 - Karuizawa Museum Complex - Karuizawa, JP - YASUI HIDEO ATELIER |
| | 25 2009 - Autobahn Church Siegerland - Wilnsdorf, DE - Schneider+Schumacher |
| | 26 2009 - Horten Headquarters - Copenhagen, DK - 3XN |
| | 27 2009 - Neo Solar Power Corporation - Hsinchu City, TW - J. J. Pan & Partners, Architects & Planners (JJPP) |
| | 28 2010 - Arthouse - Hangzhou, Zhejiang, CN - Joey Ho Design |
| | 29 2010 - Kiefer Technic Showroom - Bad Gleichenberg, AT - Ernst Giselbrecht + Partner |
| | 30 2010 - Museum of Art Amir Building - Tel Aviv, IL - Preston Scott Cohen |
| | 31 2010 - Office Building In Istanbul - Istanbul, TR - Tago Architects |
| | 32 2010 - Q1, ThyssenKrupp Quarter - Essen, DE - JSWD Architekten, Chaix & Morel et Associés |
| | 33 2011 - CIB, Biomedical Research Center - Pamplona, ES - Vaíllo & Irigaray & Galar |
| | 34 2011 - Glasgow Riverside Museum of Transport - Glasgow, GB - Zaha Hadid Architects |
| | 35 2011 - House 77 - Póvoa de Varzim, PT - dIONISO LAB |
| | 36 2011 - Origami - Paris, FR - Manuelle Gautrand Architecture |



Permanent Architecture

- | | | |
|--|----|---|
| | 37 | 2011 - Perry and Marty Granoff Center for the Creative Arts - Providence, Rhode Island, US - Diller Scofidio + Renfro |
| | 38 | 2011 - Stadshuis Nieuwegein - Nieuwegein, NL - 3XN |
| | 39 | 2011 - Vivida - Hawthorn, Melbourne, AU - ROTHELOWMAN |
| | 40 | 2012 - Al Bahar Towers - Abu Dhabi, AE - Aedas Architects |
| | 41 | 2012 - Dalian International Conference center - Dalian, CN - Coop Himmelb(l)au |
| | 42 | 2012 - Factory Building on the Vitra Campus - Weil am Rhein, DE - SANAA |
| | 43 | 2012 - FECHAC Regional Office - Ciudad Juarez, MX - Grupo ARKHOS |
| | 44 | 2012 - Festival Hall of the Tiroler Festspiele - Erl, AT - Delugan Meissl Associated Architects |
| | 45 | 2012 - Hooke Park Big Shed - Dorset, GB - AA Design & Make |
| | 46 | 2012 - Kilden - Kristiansand, NO - ALA Architects |
| | 47 | 2012 - Mülimatt Sports Education and Training Centre in Windisch - Brugg, CH - Studio Vacchini Architetti |
| | 48 | 2012 - Naturum Kosterhavet - Ekenäs, SE - White Arkitekter |
| | 49 | 2012 - Roberto Cantoral Cultural Center - Coyoacán México DF, MX - Broissin Architects |
| | 50 | 2012 - Salon Urbain - Montreal, QC, CA - Ædifica + Sid Lee Architecture |
| | 51 | 2013 - Aix en Provence Conservatory of Music - Aix-en-Provence, FR - Kengo Kuma and Associates |
| | 52 | 2013 - Assemble Studio - Northcote, VIC, AU - Assemble |
| | 53 | 2013 - Dear Ginza - Chuo, Tokyo, JP - Amano design office |
| | 54 | 2013 - High School Crinkled Wall - Kufstein, AT - Wiesflecker Architecture |



Permanent Architecture

- | | | |
|--|----|--|
| | 55 | 2013 - HygroSkin - Meteorosensitive pavilion - FRAC Centre Orleans, FR - Achim Menges, Oliver David Krieg and Steffen Reichert |
| | 56 | 2013 - Innovation and Technical and Technological Transfer Park - Chihuahua, MX - Grupo ARKHOS |
| | 57 | 2013 - Muqarnas Tower - Riyadh SA - Skidmore, Owings & Merrill, SOM |
| | 58 | 2013 - Siemens HQ in Masdar City - Masdar City - Abu Dhabi - AE - Sheppard Robson |
| | 59 | 2013 - Textilmacher - Munich, DE - tillicharchitektur |
| | 60 | 2014 - Commercial Foyer Space, 105 Wigmore Street - London, GB - Paul Nulty Lighting Design |
| | 61 | 2014 - New Wave Architecture Designs Rock Gym for Polur - Polur, IN - New Wave Architecture |
| | 62 | 2014 - SDU Campus Kolding - Kolding, DK - Henning Larsen Architects |
| | 63 | 2014 - Velenje Car Park - 3320 Velenje, SI - ENOTA |
| | 64 | 2014 - Yokohama International Passenger Terminal - Yokohama, JP - Foreign Office Architects (FOA) |
| | 65 | 2015 - Bespoke Theatre - Xishuangbanna, Yunnan, CN - Stufish Entertainment Architects |
| | 66 | 2015 - Cozzarelli price stage, National Academy of Science - Washington DC |
| | 67 | 2015 - Qingdao Cruise Terminal - Qingdao, Shandong, CN - CDDI - Mozhao Studio & Jing Studio |
| | 68 | 2016 - Poly International Plaza - Beijing Shi, CN - Skidmore, Owings & Merrill |
| | 69 | 2016 - Széll Kálmán Square - Budapest, HU - Építész Stúdió, Lépték-Terv |
| | 70 | 2016 - Tokyu plaza - Tokyo, JP - Nikken Sekkei |
| | 71 | 2017 - Low Carbon Energy Centre - London, GB - C.F. Møller Architects |
| | 72 | 2013 - Kyushu Geibunkan - Fukuoka, JP - Kengo Kuma |



Temporary Architecture

- | | |
|--|--|
| | 1 1964 - Pavillion Wehrhafte Schweiz - Lausanne, CH - Carl Fingerhuth |
| | 2 1967 - Plydome at Flash Peak camp - Indio, CA - Herbert Yates, Hirshen Van der Ryn Architects |
| | 3 2000 - Origami Shelter sculpture - JP - Yuko Nishimura |
| | 4 2005 - Jumbo Origamic Arch White - Jakarta (Indonesia, Pescara, IT - Atelier Bow - Wow |
| | 5 2005 - Jumbo Origamic Arch Orange - Kobe Art Village Center, Hyogo, JP - Atelier Bow-Wow |
| | 6 2006 - dB folding disco - BR - Fernanda Dolabella Dubal |
| | 7 2008 - Embedded Project - Shanghai, CN - HHD_FUN + Xu Wenkai |
| | 8 2008 - Chapel for the Deaconesses - Hôpital de St-Loup, CH - Localarchitecture + Danilo Mondada |
| | 9 2009 - Cardboard banquet paper pavilion - Cambridge, GB - Cambridge University |
| | 10 2011 - Bloomberg Pavilion - Tokyo, JP - Akihisa Hirata |
| | 11 2011 - Winnipeg Skating Shelters - Winnipeg, MB, CA - Patkau Architects |
| | 12 2012 - Burning Man yurt - Black Rock City, US - Joerg Student - ideoLABS |
| | 13 2012 - Canary Wharf Kiosk - London, GB - Make Architects |
| | 14 2012 - Insrtant Flat-Pack Origami -Shelter - Doowon Suh, KR |
| | 15 2012 - Public Toilets - Uster, CH - Gramazio & Kohler |
| | 16 2012 - The Bowooss Temporary Pavilion - Saarbrücken, DE - The School of Architecture at Saarland University |
| | 17 2013 - ArboSkin Pavilion - Stuttgart, DE - ITKE |
| | 18 2013 - Cardborigami, temporary cardboard shelter - Santa Monica, California, US - Callison LLC |



Temporary Architecture

	19 2013 - Emergency Shelter - Melbourne, AU - Woods Bagot
	20 2013 - Ferrocement Shelter - IN - Anupama Kundoo
	21 2013 - Indo - German Urban Mela pavillions - Pune, IN - Markus Heinsdorff
	22 2013 - Plate house - Oxford, GB - Jo Gattas and Zhong You
	23 2013 - Pop-up Dome Prototype - Delft, DE - University of Technology
	24 2014 - FFolded Bamboo + Paper House concept - CN - Ming Tang
	25 2014 - Flat-pack disaster housing - AU - Alastair Pryor
	26 2014 - Ha-Ori Shelter - Bolinas, CA - Joerg Student - ideoLABS
	27 2017 - Look! Look! Look! - Herefordshire, GB - Studio Morison



Installations

	1 2000 - La Patata - DE - Volker Flamm & Wolfgang Ohnmacht
	2 2002 - Your spiral view - Fondation Beyeler, Basel, CH - Olafur Wliasson
	3 2008 - Life Tunnel - The Hayward, London, UK - Atelier Bow Wow
	4 2009 - Ecological Installations - Greenmeme
	5 2010 - Move: Choreographing You exhibition design - London, UK - Amanda Levet Architects
	6 2011 - Le Fabrique Sonore - Reims, FR - Hyoung-Gul Kook, Ali Momeni and Robin Meier
	7 2011 - Overliner - MIT, US - Joel Lamere, Cynthia Gunadi
	8 2011 - Resonant chamber - University of Michigan, US - RVTR



Installations

- | | | | | | | |
|--|--|--|--|--|----|--|
| | | | | | 9 | 2011 - Tessel, kinetic sound installation - Lyon, FR – Brussels, BE - David Letellier and Lab[au] |
| | | | | | 10 | 2012 - Arm Shell - Biennale di Venezia, IT - Zaha Hadid Architects |
| | | | | | 11 | 2012 - Curved Folding, Metal Twins - Bangalore, IN - S. Chandra, S. Bhooshan, M. El-Sayed |
| | | | | | 12 | 2012 - INVOLUTION - Savannah, Georgia, US - (LAB)normal, Larry O. Martin |
| | | | | | 13 | 2012 - Rainbow Gateway - Burnley, GB - Tonkin Liu |
| | | | | | 14 | 2013 - Computational parabolic origami Shelter - Komaba Museum, Tokio, JP - Tomohiro Tachi |
| | | | | | 15 | 2013 - Rigid foldable generalized Miura - Komaba Museum, Tokio, JP - Tomohiro Tachi |
| | | | | | 16 | 2014 - Blumen Lumen - Black Rock City, US - Joerg Student - ideoLABS |
| | | | | | 17 | 2014 - One Fold - CA - Patkau Architects |
| | | | | | 18 | 2015 - Curved-Line Folding Workshop - School of Architecture, London, UK - Michael Weinstock, Axel Körner, Suryansh Chandra |
| | | | | | 19 | 2015 - Computing Curved-Folded Tessellations through Straight Folding Approximation - University of Stuttgart, DE - Michael Weinstock, Axel Körner, Suryansh Chandra |
| | | | | | 20 | 2016 - Origami Pavilion - Detmold University, DE - Tal Friedman |
| | | | | | 21 | 2016 - "Surface to Form" pavillion - University Innsbruck, AU - Rupert Maleczek, et al. |
| | | | | | 22 | 2016 - Arch(k)inetic workshop - University of Stuttgart, DE - Contemporary Architects Association Tehran, Axel Körner et al. |
| | | | | | 23 | 2017 - Curved folded wooden assemblies - Holzbau Saurer, Höfen, AU - Rupert Maleczek, Gabriel Stern, et al. |
| | | | | | 24 | 2018 - Curved-Folded Assemblies - University of Innsbruck, AU - Rupert Maleczek and Axel Körner |



Goods and Furniture



1 3D Surface - Caos - TRG, TRB



2 Petrucci Adrien - Paper cast teapot - Ceramic



3 Alice Minkina - Chair AMi - Fabric and plywood



4 Anthony Dickens & Tony Wilson - Origami table - Steel



5 Ariel Zuckerman - Wood And Light Origami - Wood



6 ARTEMIDE, Issey Miyake - Mendori - Paper



7 Bell Phillips - Origami stairs - Stainless steel



8 Ben Ryuki Miyagi - Elephant Seating - Felt



9 Boaz Mendel - Loop Chair - Wood and metal



10 Brett Mellor - Facetation Butterfly Chair - Fabric and metal



11 Brett Mellor - Flat Stanley Origami Chair - Wood, Canvas and Vinyl



12 Brett Mellor - The Morgan Felt Folding Stool - Felt impregnated with resin



13 BUILT - Origami Wine Tote - Neoprene



14 Cut and Fold by Andrea Kordos & Tony Round - Aperture Pendant - Metal and wood



15 Cut and Fold by Andrea Kordos & Tony Round - Origami Chair - Metal and wood



16 Daniel Milchtein Peltsverger - Biombo Chair - Wood



















































































































































17 Daniel Schipper - Green Green House - Recyclable plastic



18 DegreesOfFreedomCo by Brian Ignaut - Loop table - Wood



Goods and Furniture

- | | |
|--|--|
|         | 19 DegreesOfFreedomCo by Brian Ignaut - Legged wooden origami table - Wood |
|         | 20 DegreesOfFreedomCo - by Brian Ignaut - Kinetic Ring Box - Wood |
|         | 21 DOMUS by Tobias Krafczyk - Origami Intersections - Paper |
|         | 22 Donn Koh - Origami for superheroes - pencil sharpener - Stainless steel |
|         | 23 -FiberStore - Bubble - Polypropylene |
|         | 24 FiberStore - Pebble - Paper |
|         | 25 FiberStore - Poetry - Paper |
|         | 26 FiberStore - Ramekin - Paper |
|         | 27 FiberStore - Volant - Paper |
|         | 28 -Florian Kräutli - Magnetic origami curtain - Fabric, metal, magnets |
|         | 29 FLUX - Flux Chair - Plastic |
|         | 30 Foldschool - Cardboard kid furniture - Cardboard |
|         | 31 -FOSCARINI - Diesel Rock Light - ABS, Metal, nylon |
|         | 32 Four-o-nine - Pleat Diner chair - Polyethylene |
|         | 33 Fox & Freeze - Lounge Chair - Felt |
|         | 34 Fredrik Färg - The COAT Chair - Felt |
|         | 35 Gant Lights - Concrete hanging lamp - Concrete |
|         | 36 Gold Leaf Design Group - Origami Paper Lamps - Paper |



Goods and Furniture

- | | |
|--|---|
| | 37 HOID - Pleat Box - Ceramic, glazed enamel, metal |
| | 38 Ignatov Architects - News Coffee Table - Aluminium |
| | 39 Ilan Garibi - Tessellated origami mirrors - Aluminium |
| | 40 Ilan Garibi - Tessellated origami Table Lamp - Paper and Wood |
| | 41 Ilan Garibi - Tessellated origami tables - Wood |
| | 42 INC Architecture & Design - Origami Stairs - Metal |
| | 43 Issey Miyake - Fukurou Lamp - Recycled polyester fibre fabric |
| | 44 Issey Miyake - Minomushi Terra Lamp - Recycled polyester fibre fabric |
| | 45 Issey Miyake - Mogura Lamp - Recycled polyester fibre fabric |
| | 46 Issey Miyake - Tatsuno-Otoshigo Lamp - Recycled polyester fibre fabric |
| | 47 James Slack - ORI - Cardboard |
| | 48 Jan Sekula - Cardboard Playground - Cardboard |
| | 49 Jiangmei Wu - Torus Folded Lamp - Recycled cotton paper |
| | 50 Joseph Joseph - Folding Colander - Plastic |
| | 51 Jurmol Yao - Venom - Metal and carbon fibre |
| | 52 Kaj Franck - Origami Plate - K F 2 - Ceramic |
| | 53 KARTON - Cardboard Furniture - Cardboard |
| | 54 KARTON - Twist Table - Cardboard |



Goods and Furniture

	55 Keiji Ashizawa - Flat Pack Wall Magazine Holder - Steel
	56 Kelly Lohr - Origami Chair - PETG thermoplastic polymer
	57 KNOLL - Washington Prism, Lounge chair - Urethane moulded foam, Baltic Birch plywood panels covered in plastic
	58 Krings & Sebastian Mühlhäuser - Casulo - Cardboard
	59 Kyungeun Ko - Bentley tailormade - Aluminium
	60 Lapalma, Shin Azumi - AP Foot stool - Wood
	61 Le Klint - Pendant 150 - Lampshade foil
	62 Le Klint - Pendant 153 - Plastic Foil
	63 Le Klint - Pendant 169 - Plastic Foil
	64 Le Klint - Pendant 172 - Plastic Foil
	65 Le Klint - Pendant 178 - Plastic Foil
	66 Le Klint - Pendant 181 - Plastic Foil
	67 Lim Ruiwen - Origami Shade - Fabric
	68 -Matsuoka furniture - Origami Chest - New Guinea Walnut
	69 Max Hauser - Trifold - Anodised Aluminium
	70 McEwen Lighting Studio - Gear Ceiling Fixture - Satin Nickel
	71 Michael Sholk - Foldable wooden Spoon - Wood
	72 Polygons Measuring Spoon - Rahul Agarwal



Goods and Furniture



73 Kristina Wißling - Miura ori world map - Paper



74 Matteo Signorini - Origami Boat



75 Max Frommled & Arno Mathies - Folding Boat - Plastic



76 Milk Design Limited - Origami Glow - Paper and wood



77 Moritz Menacher - Urban Origami Bike - Aluminium cut from a single sheet



78 NEO design studios -Vanity - Fabric



79 Novague - EDGE chair - Aluminium



80 Patricia Urqiola - Antibody Chair - Reversible felt, wool fabric, stainless steel



81 Phil Cuttance - FACETURE lamps - Resin



82 Phil Cuttance - FACETURE vases - Resin



83 Po Shun Leong - Cookie stool, Counter Stool, Po Chair - Wood



84 TO DO Product design - Papero lamp - ABS, polypropylene



85 Rami Tareef - Foldigon, Outdoor Sofa - Wood and Textile



86 Ran Amitai - Folded aluminium chair - Aluminium



87 Sebastian Burdon - Sawn Table - Aluminium and glass



88 Semi Skimmed milk - Milk curved packaging - Cardboard



89 Shin Yamashita - Land Peel



90 Showroom Finland - Cardboard furniture - Cardboard



Goods and Furniture

- | | |
|--|---|
| | 91 SLAMP - Bach - Opaflex |
| | 92 SLAMP - Chapeau - Metallized-Mirror Polycarbonate |
| | 93 SLAMP - Diamond - Opaflex |
| | 94 So Takahashi - Origami Chair - Steel |
| | 95 Sooin Kim - Origami Chair - Plastic and Velcro |
| | 96 Studio Ayaskan - GROWTH, an origami-like pot that grows with plant - Polypropylene |
| | 97 Studio Dror - Pick Chair - Wood and Metal |
| | 98 Studio Snowpuppe - Chestnut wooden origami lamp - Birchwood veneer |
| | 99 -Taewon Hwang - Flat Pack Mouse - Polycarbonate, Silicone, Plastic |
| | 100 Tobias Labarque - Perforated aluminium folded chair - Aluminium |
| | 101 Tobias Labarque - Perforated aluminium folded chair - Aluminium |
| | 102 Tomohiro Tachi - Rigid foldable table - rigid panels with membrane hinges |
| | 103 Van Esch by Matthias Demacker - Origami Table - Aluminium |
| | 104 Vondom by Karim Rashid - VERTEX Chair - recyclable plastic |
| | 105 Uria Graver - CHAIR-IGAMI - Metal |
| | 106 Yuji Fujimura - ORIC, origami chair - Plastic |
| | 107 Zhang Zhoujie - Triangulation Series Furniture - Aluminium |
| | 108 Zipper 8 Lighting - Large Dakota Pendant Light - Paper |



Fashion and Clothing

- | | |
|--|--|
| | 1 EIN-TRITT - Catherine Meuter - Canvas and other materials |
| | 2 Tridimensional Origami Phone - Chengyuan Wei - Cardboard |
| | 3 Paper Bowtie - FiberStore - Paper |
| | 4 Melt - HOID - Cotton jersey, Iron on interface |
| | 5 MUYBRIDGE PT.2 - HOID - Cotton jersey, Iron on interface |
| | 6 NO 419 - HOID - Cotton jersey, Iron on interface |
| | 7 H+Bag - Hyo Jun Jeon - Paper |
| | 8 Tessellated origami ring - Ilan Garibi - Gold |
| | 9 Tessellated origami necklace - Ilan Garibi - Gold |
| | 10 Tessellated origami bracelet - Ilan Garibi - Metal and wood |
| | 11 Paper bags - Ilvy Jacobs - Paper |
| | 12 Origami dresses - Jum Nakao - Paper |
| | 13 Sa Umbrella - Justin Nagelberg - Recyclable plastic polymers |
| | 14 Folding Leather Stool - Louis Vuitton, Atelier Oi - Leather |
| | 15 Collapsible Cycling Helmet - Mike Rose - Metal fabric and plastic |
| | 16 Folded plate shirt - MILIVOJEVIC MILOS - Cotton |
| | 17 Ecstatic Spaces collection - Tara Keens Douglas - Paper |
| | 18 Ecstatic Spaces collection - Tara Keens Douglas - Paper |



Fashion and Clothing

	19 Flux Snowshoe - Eric Burnt
	20 Iittala X bag - Issey Miyake
	21 Silver-tone 'Prism' tote bag - Issey Miyake
	22 123 5. Standard - Issey Miyake
	23 Frame Pleats Bag - Issey Miyake
	24 Luna Pleats - Issey Miyake
	25 Origami Fashion - Diana Gamboa
	26 Paper couture - Sylwia Lewandowska
	27 Nintai, Origami-Inspired ,Geometric Dresses - Mercedes Arocena and Lucia Benitez

2.3. Synoptic Tables Data Analysis

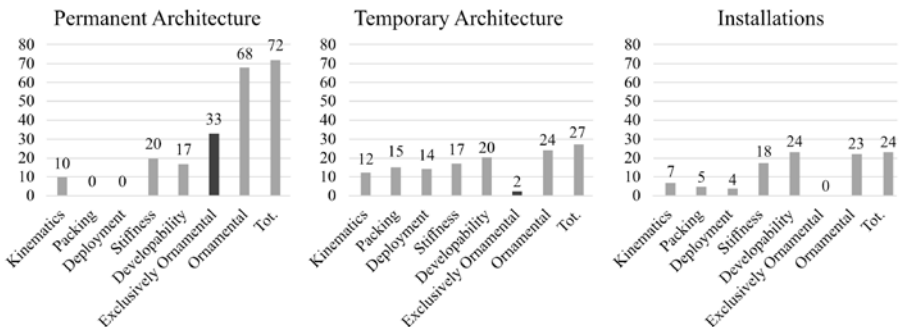


Figure 7 - Histograms of Permanent Architecture, Temporary Architecture, and installations.

In the field of “Permanent Architecture” (Figure 7), it is evident that a major part of projects is inspired by origami only for ornamental reasons without any other functional purpose. The Tokyu Plaza, the Fuji television Wangan studio,

the stage of the Cozzarelli price at the National Academy of Science and the Yokohama International Passenger Terminal are examples of this category. All these examples are clearly using folded surfaces only for ornamental purposes because, in these cases, the folding is neither contributing to make the structure stiffer, nor to make the surface movable, nor to optimize the assembling or transportation. When designing with origami, the designer often needs to control the shapes in a dynamic way, and not all architects are specialized in using advanced 3D-animation applications. In addition, static architecture cases are way more numerous than kinetic architecture ones, because they are less expensive and easier to maintain, but even if we analyse only the kinetic architecture field, the origami mechanisms are always simple and often copied from traditional well-known patterns. These numbers tell us that there is the desire of using origami as a reference because of its beautiful appearance and useful functionalities. Nevertheless, very often, due to the complexity of designing origami mechanisms at that scale, and probably due to the lack of digital tools specifically aimed to origami design, the examples of buildings that are referenced to origami in a functional way are still rare. On the contrary, it is clear that in “Temporary Architecture” the kinematic and mechanical properties of origami are way more used as tools to improve the projects functionalities, and the projects inspired from origami exclusively for ornamental purposes are rarer. One interesting example of origami-inspired temporary architecture is the “Plate house” by Gattas and You (Gattas & You, 2016) who used origami techniques to design a self-supporting sandwich structure made by cardboard.

One of the reasons of such different numbers between temporary and permanent architecture is probably related to the scale of the objects. Smaller dimensions allow the designers to use the self-supporting properties of the materials that benefits from creases; moreover, the developability is used a lot more because it can cut the production costs and time. Furthermore, the temporary architecture examples, by their nature, need to be moved, so the deployment and the packing characteristics become more relevant compared to the aesthetics. Nevertheless, the risk, highlighted earlier, of falling into the mere copy of traditional patterns while designing with origami is still an element strongly present also in this kind of projects. In fact, designers often search for solutions into existing patterns and standard constructions probably because of the lack of design tools or a lack in their knowledge and familiarity with origami constructions.

In the field of “Installations”, however, we can see increased efforts to find innovative solutions and patterns. For example, the Resonant chamber by RVTR

(Thün *et al.*, 2012) is a virtuous example of applied origami, where the origami properties are used to make a morphing suspended ceiling capable of changing its shape according to the variable acoustic conditions of the space where it is installed into.

Other interesting examples are the “Computing Curved-Folded Tessellations through Straight Folding Approximation” (Chandra *et al.*, 2015b) and the “Curved folding metal twins” installation by Chandra *et al.* (Chandra *et al.*, 2015a), where they used curved folding as a tool to design nice looking curved stiff sculptures by folding a developable surface. In this case, the creases are not used to generate motion, but they are utilised to optimize the fabrication process and to increase the stiffness of the structure.

The same researchers (in collaboration with Zaha Hadid Architects) also designed the “Arum Shell” installation, exhibited at the “Biennale di Venezia” in 2012 (Bhooshan, 2016), which is a beautiful example of modular structure constructed with curved-folded developable metal plates folded with robotized mechanical arms at RoboFold company (Epps, n.d., 2014; Epps & Verma, 2013). These projects are usually academic works or experimentations made by groups of researchers and artists. Sometimes they have the only function of displaying design skills or advertise some architecture firm. Often the projects focus just on exploring the shape or on testing the properties of the material or the efficiency of a certain technology. In many cases, we have noticed that the prototyping phase proceeds in parallel with the creation of the generative algorithms and the analysis of the digital model. In this way, the correctness of the model can be tested through a comparison between the digital surface and the physical prototype.

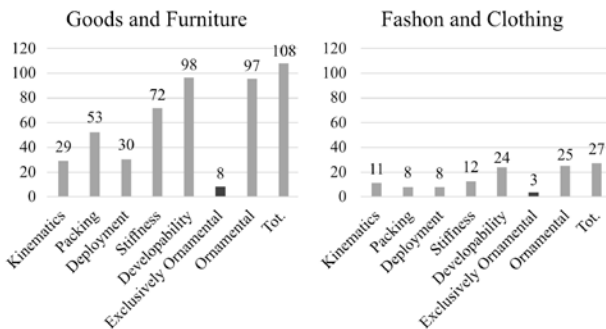


Figure 8 - Histograms of Goods and Furnitures, and Fashion and Clothing.

Differently, from what happens in the field of “Permanent Architecture”, in the “Goods and Furniture” and “Fashion” fields (Figure 8), we found a higher percentage of projects that take advantage of origami design techniques, rather than projects that use origami exclusively for ornamental purposes. For example, the developability is a relevant point in both fields, not only for the possibility to produce an object from a single sheet of the same material but also to optimize the space consumption and the assembling time. The reason of this difference is probably due to the fact that in large-scale projects inspired by origami even if the folded surface is designed to be globally developable, it would require to be assembled instead of being cut and folded from one single sheet of the same material. Thus, the global developability does not really bring a real advantage in architectonic-scale origami. Furthermore, even if it would be found a way to produce and fold a large-scale sheet of the same material (which is already a difficult task), the material should be flexible enough to be foldable while being stiff enough to be self-supporting, which is not an easy target to reach for any large-scale continuous surfaces.

2.4. Designing with Folded Surfaces – Critical Observations

According to the collected projects, and considering the analysed data, it is evident that small-scale projects rather than architectonic-scale projects make use of origami-related characteristics for many different functional purposes and not only for ornamental purposes. Furthermore, in both the fields of manufacturing and architecture, we can find a multitude of designs that use well-known patterns taken from traditional origami designs, or from previous projects (e.g. the Yoshimura pattern, the Miura pattern, the water-bomb base pattern). There are many different reasons that may explain this trend. The first is probably due to the scale of the objects because using origami for large-scale projects involves the problem of thickness, and it becomes harder to fold a large-scale surface from a single sheet of the same material. Furthermore, in big-size projects, the typical continuous surfaces proper of origami designs introduces new issues about the mechanical resistance of the joints or about the shape and the dimensions of the hinges that could possibly replace the creases. Moreover, origami mechanisms may have moving parts which are harder, more expensive and more time-consuming to design and maintain compared to static projects. Lastly, small-size designs can be prototyped by using directly the physical models instead of passing through digital simulations. On the contrary, for large-scale

projects the accuracy needed is much higher, thus the digital models of the folded surfaces are necessary. This makes the workflow harder because to be able to simulate digitally the folding animation of origami, the designer must deal with the kinematics of the specific mechanism that he is trying to design. This requires a deep understanding of the origami theories. Furthermore, the moving parts influence the design process from earlier stages. For example, if we want to design a traditional chair the design process would be similar to the one schematized into Figure 9. The design process would usually follow a linear sequence of steps starting from the idea, up to the final object, passing through the sketch, the technical drawing (2D drawings and 3D model) and the prototype. If we compare this process with the one of a foldable origami-inspired chair, the design workflow would appear similar to the one illustrated in Figure 10.



Figure 9 - An example of a design process of a traditional non-foldable chair.

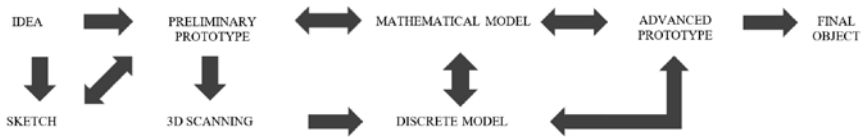


Figure 10 - An example of a design process of an origami-inspired foldable chair.

The sketch would have marginal importance because it would only contribute to the design of the preliminary aesthetic or functional aspects, which are not as crucial as the accurate analysis needed to address the kinematics. The preliminary conceiving phase would probably see the substitution of the sketch with a paper model which would already help the designer to reflect on some important aspects like the developability, the rigid-foldability, the blocking creases, and the DOF. The preliminary prototype, however, usually does not consider the thickness of the panels, which is something that it is usually postponed in a later step when a more accurate and rigid advanced prototype is built. However, the preliminary and advanced prototypes together are not sufficient to test every single aspect of the project, because the panels of which they are made may have a different elasticity compared to the one of the final object. Furthermore, the rigid-foldability and the DOF are difficult to verify only by using physical

models, thus a digitalization of the prototype may help to make an accurate analysis of these aspects. The conversion of the physical prototype into a digital model may be achieved by 3D scanning the model, this solution is often used in curved-folding designs (Kilian *et al.*, 2008) or by constructing and animating it by following the mathematical rules that regulate its pattern which would return much more versatile results with much higher accuracy.

However, also the digital model has some limits. For example, it is hard to simulate accurately the folding motion considering thickness, friction, elasticity and deformations. Furthermore, it might be easy to run into self-intersected configurations which are possible in the digital model but not allowed in the real one. Thus, if not checked carefully, this may cause the final object to block at a certain point because of a non-perfect rigid-foldability or overlooked collisions and self-intersections. Thus, once verified the real behaviour of the advanced prototype, the designer may have to update and eventually implement the analysis of the mathematical model or the discrete model developed previously. Because of all these reasons, we can clearly see that the design process is much harder, and it is not a linear process anymore.

DEFINITIONS AND THEOREMS

This chapter introduces, to those who are not origami experts, some important concepts of origami theory, without which it would be impossible to understand the discussions in the following chapters. In particular we will clarify the following terms and topics: Fold-angle (cf. section 3.1), Developability (cf. section 3.2), Degree of Freedom (DOF) (cf. section 3.3), Rigid-foldability (cf. section 3.4), Flat-foldability (cf. section 3.5), Non-flat-foldability (cf. section 3.6).

3.1. Fold Angle

For “Fold angle” it is intended the dihedral angle between two consecutive faces divided by a crease at any moment of the folding motion. The dihedral angle is an angle between two planes in a third plane, which is perpendicular to the intersection line between the former two planes. To measure the fold angle, we can measure the actual angle between the two planes from surface to surface, or we can measure the angle between their normal vectors. In the former case, the fold angle of a flat-foldable origami with one single linear crease goes from 180° (unfolded flat configuration) to 0° (folded configuration) or from 180° to 360° depending on the mountain/valley assignment. In the latter, the fold angle goes from 0° (unfolded flat configuration) to $\pm 180^\circ$ (folded configuration). Usually, the scientific community favours the normal-to-normal measuring method.

3.1.1. Fold Angle Over Time – From Plot Analysis

In Figure 11 you can see the plotted graph of the fold angle over time, however, it does not consider the sign of the crease or the verse of rotation of one or the other face. That happens because we asked the software to measure and return the angle between the normal vectors, and as a result, it plotted the smaller pos-

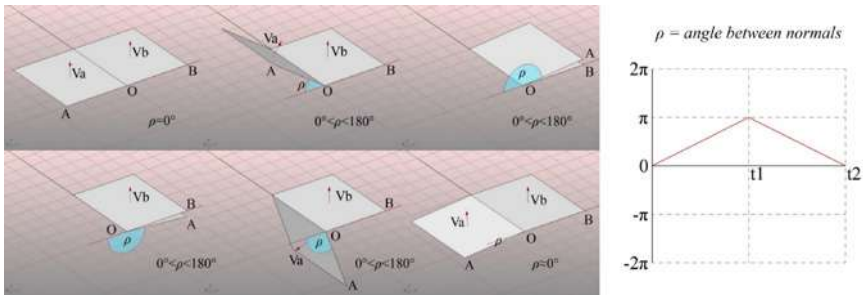


Figure 11 - Fold angle without folding verse information.

sible angle frame-by-frame. The two faces rotate around the crease, and at 180° they pass through each other and continue their motion until they reach the flat unfolded state again. After passing 180° the verse of the valley fold reverses instantly and it transforms to mountain, but in the graph, there is no evidence of that. We would have the same function shape if the two surfaces, once reached 180° , would reverse their verse of rotation returning to the unfolded state without passing one through each other. Therefore, we need to add a piece of new information to define the fold angle. We can, for example, multiply for -1 the angle value once the rotation angle is greater than 180° .

This is sufficient to define the mountain valley assignment. Mathematically it can be solved as follows. Consider a unit vector along the direction of the crease line, multiply it with the unitized cross product between the normal vectors of the two faces. This will return ± 1 according to the verse of the crease in relation to the verse of the normal vectors of the surfaces. Multiplying this value with the angle between the normal vectors of the faces will return a signed angle according to the folding verse of the crease, as shown in Figure 12. The formula used to obtain the graph in the figure is the following:

$$\rho_s = \text{sign} (|V_b, V_a, V_o|) \cdot \rho. \tag{1}$$

Which, in the explicit form is:

$$\rho_s = \rho \cdot \frac{V_b \times V_a}{|V_b \times V_a|} \cdot V_o. \tag{2}$$

Where:

ρ_s is the signed angle between the normal vectors V_a and V_b according to the verse of the fold;

ρ is the angle between the normal vectors V_a and V_b ;

V_a is the normal vector to face A;
 V_b is the normal vector to face B;
 V_o is the vector along the direction of the fold.

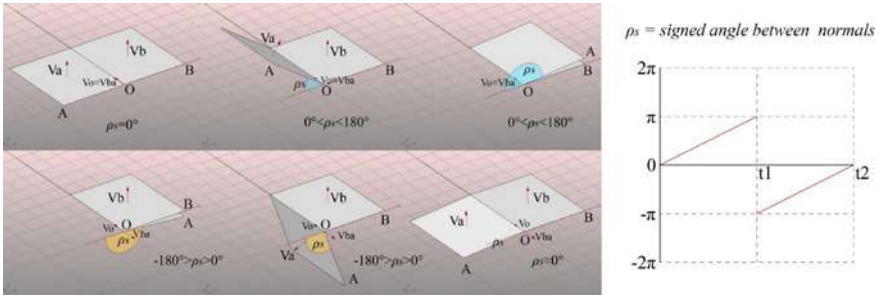


Figure 12 - Signed angle between normals.

As you can see the plotted function of the angle jumps from 180° to -180° at t_1 , which is what we were looking for because the jump corresponds to the instant flip of the verse of the fold at the moment of the self-intersection. In this case, the function is periodical, and every time it hits 0° it restarts equal to itself. Therefore, it gives us evidence about the mountain/valley assignment, but it does not keep track of the number of total rotations. Thus, if we want to also add that information, we need to implement the formula as follows:

$$\rho_t = \rho + 90^\circ - \left(90^\circ \cdot \frac{V_b \times V_a}{\|V_b \times V_a\|} \cdot V_o \right). \quad (3)$$

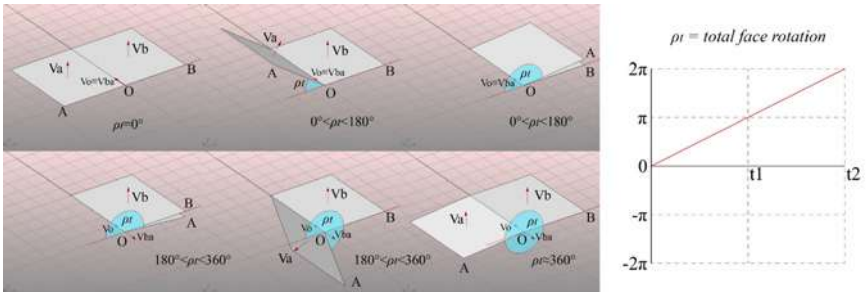


Figure 13 - Keeping track of the total rotation of the faces.

The graph shown in Figure 13 keeps track of the total reciprocal rotations of the faces A and B, and the fold changes verse when the function intersects the ordinates values multiples of $180^\circ (\pi)$. To be able to keep track of rotations bigger than $360^\circ (2\pi)$ though, the formula needs to be updated as follow:

$$\rho_t = 360^\circ \cdot n + \rho + 90^\circ - \left(90^\circ \cdot \frac{V_b \times V_a}{\|V_b \times V_a\|} \cdot V_o \right). \quad (4)$$

Where: n is the number of complete turns.

Now if we apply this method to measure the fold angles of every crease in a more complex pattern we would have something similar to the graph shown in Figure 14, which represents the plots of the fold angles over time of a degree-4 single vertex (cf. section 3.5 and 3.6 for more about flat foldability of degree-4 vertices).

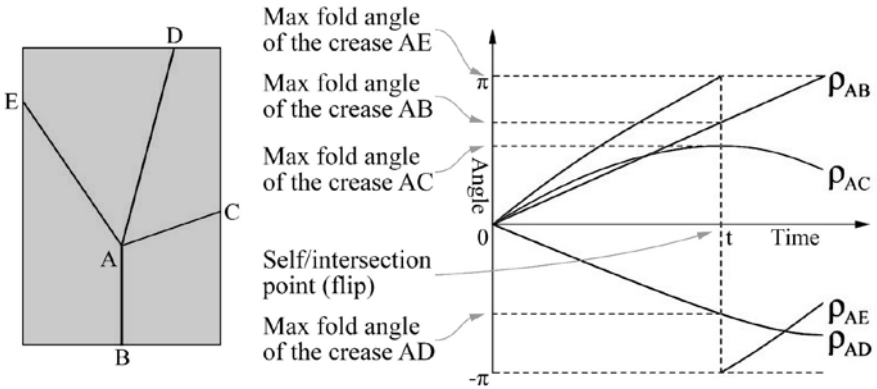


Figure 14 - Analysis of the fold angle speed for each fold in a degree-4 vertex CP (crease pattern).

The plot shown in figure follows the expression (2), therefore, when the function lies in the negative side of the Cartesian plane it means that the crease is mountain folded, on the contrary, when it lies in the positive side, the crease is valley folded. In Figure 14 the fold angle ρ_{AB} , which is the controller fold angle, has been animated with constant speed, therefore its function is the only one with a linear path, but it is not the one that hit 180° first. In fact, at the time t included in the Δt domain, one of the other folds flips and its function jumps from the positive space to the negative space of the Cartesian plane. The time t represents the moment when the vertex self-intersects and the verse of the fold

AE (once the two faces adjacent to it passes through each other) instantly flips changing its mountain/valley assignment.

To find the values of the maximum fold angles for each crease, it is sufficient to intersect all the fold angle functions with the abscissa t and extrapolate the relative ordinate values. Furthermore, in the same way, it is possible to extrapolate all the fold angle values for each fold at any t .

3.2. Developability

The developability is the property of any surface to be unfolded or unrolled into a plane without distortions or cuts. Conversely, it is a non-planar surface which can be shaped by transforming a plane by folding, rolling. The developable surfaces are always ruled surfaces, but not every ruled surface is developable, for example, the hyperboloid is a ruled surface which is not developable (Migliari, 2009a). Ruled surfaces that are developable are for example cylinders or cones. A developable surface mathematically is a surface with zero Gaussian curvature, on the contrary non-developable surfaces have double curvature or non-zero Gaussian curvature. The Gaussian curvature in differential geometry is called K and represents the product of the principal curvatures K_1 and K_2 at a given point of the surface.

For example, a sphere has a Gaussian curvature equal to $1/r^2$ in every point of its surface. On the contrary, a plane or a cylinder have Gaussian curvature equal to 0 everywhere. In these examples, the Gaussian curvature is equal in every point of the surfaces, but in general, it can be different from point to point, for example, a torus has negative Gaussian curvature in the inside and positive in the outside.

Because origami is mostly made by planar faces, we cannot use the Gaussian curvature to judge its developability. In rigid origami, the developability of a given pattern is measured by summing all the sector angles between the creases at every vertex. If all the summations of all the sector angles at every vertex are equal to 360° the pattern is developable.

Traditional origami is always developable because it starts from a flat sheet of paper, but in the last few years some researchers started to study the possible applications of non-developable vertices into origami-like mechanisms, for example in the paper *Folding Mechanisms with Discriminate Extremal Configurations for Structural Purposes* by Buffart *et al.* (Buffart *et al.*, 2018) make some considerations about using non-developable non-flat-foldable vertices to design movable mechanisms with given extremal configurations. Also, Tachi proposed a method using degree-4 non-developable vertices to convert three-dimensional

polyhedra (cut along some edges) into one-DOF mechanisms that can fold and unfold with a smooth motion without bifurcations (Tachi & Horiyama, 2018).

The vertices with the sector angles that sum up to an angle smaller than 360° can be configured into synclastic configurations (or pyramidal), The vertices with the sector angles that sum up to an angle bigger than 360° can be configured into anticlastic configurations (hyperbolic paraboloid, or saddle state). If they are degree-4, both the non-developable synclastic or anticlastic vertices have two extremal configurations that can be reached through a folding motion without bifurcations (because there is not any flat state), and the extremal configurations can be both flat-folded, or both non-flat-folded, or one flat-folded and one non-flat-folded. This characteristic increase greatly the design possibilities, but it is harder to design and fabricate than a developable vertex.

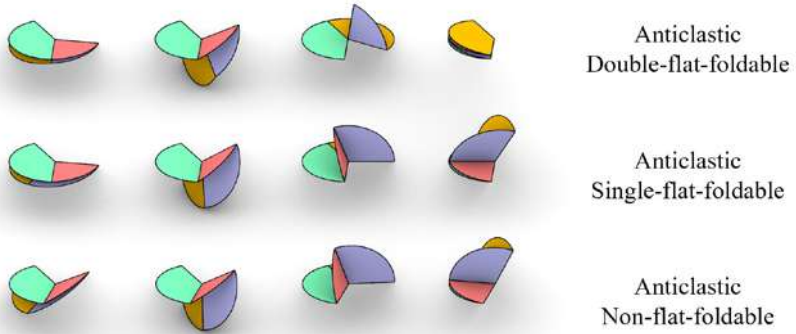


Figure 15 - Non-developable anticlastic degree-4 vertices – types.

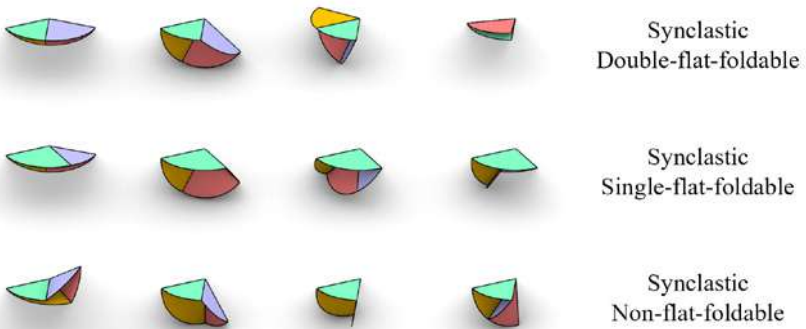


Figure 16 - Non-developable synclastic degree-4 vertices – types.

3.3. Degree of Freedom (DOF)

The “Degree of freedom” (DOF) of a system is the number of parameters that can vary independently. For example, a point in a plane has two degrees of freedom, which represent the possible translation axis or the coordinates which are needed to identify all the possible positions of the point. A non-infinitesimal object will have more than three degrees of freedom because it can also rotate in space. For example, a segment in three-dimensional space has six DOF, three for translations and three for rotations.

In rigid folding, the DOF usually represent the number of fold angles that can vary independently without bending, flexing or ripping the faces. The identification of the degree of freedom of folded surfaces is a problem that has not been generalized yet. Nevertheless, a rigid origami pattern can be compared to a rigid linkage, and there are many approaches that allow calculating the DOF of linkages. However, most of the times with symmetric and periodical patterns these methods give wrong results due to the occurrence of special conditions. For example, if we calculate the DOF of the Miura-ori (cf. section 4.7.4 for more about Miura-ori), without considering its symmetry conditions, it is apparently not foldable at all, but because of the symmetry conditions, the pattern still has 1 single DOF. In the Miura-ori case, as well as in all the cases with closed loops of faces, there are redundancies into the definition of the fold angles of each crease, which means that the same fold angle is constrained multiple times from different directions, but because of the symmetry conditions if the over-constrained angles are equals from all the directions then there are no inconsistencies and the mechanism can move anyway (Tachi, 2011a).

To calculate the DOF of a simple accordion it is sufficient to count all the creases, each one of them will increase the DOF by one. Thus, we can say that: if a pattern has some creases which do not converge to internal vertices, and there are no closed loops of faces, each new crease increases the global DOF by one. Another method to find the DOF of a pattern consists of counting the naked edges of the pattern and subtract 3 to that number, this approach works only with patterns with only triangular faces (e.g. Yoshimura pattern).

However, why is the DOF analysis so important for the animation of origami geometries? In the Miura-ori case, it is well known that it is sufficient to constrain only one single fold angle to control the folding motion because folding two consecutive faces will propagate the motion to all the other faces in the pattern univocally (the motion is univocal only if the mountain/valley assignment is given, thus it may generate problems at flat state where the

mountain valley creases are flattened into the plane). This means that you need to change only one input fold angle, to control the folds of the entire surface, being able to shape the folded surface in all the possible available configurations. If we consider a straight accordion with only two non-intersecting linear folds, we need to control the two fold angles separately to be able to shape the folded surface in any possible configuration. This means that each fold angle which can move independently from the others increases the DOF by one and requires one more controller input to configure the surface in all its possible configurations.

For accordions and symmetrical cases, it is relatively easy to foresee the number of necessary controller inputs by trial-and-error. However, it is harder in patterns that are more complex. For this reason, it is important to analyse the DOF in advance. Unfortunately, there is not a general method to calculate the DOF yet. An easy approach to test the degree of freedom of any pattern is by building a rigid physical model, but it may be deceptive for wide patterns or when the material is not rigid enough. Another approach is by simulating the animation digitally through physics engines, for example using the software “Freeform origami” by Tachi or with Grasshopper and its plug-in Kangaroo Physics. This last solution is more reliable than the physical model; however, both methods give only qualitative results. They do not return automatically exactly the DOF of the pattern if it is more than one.

It must be said that special cases are often the most interesting cases for movable mechanisms, thus it would be important and interesting to study further this aspect of paper folding and develop a generalized method that does not necessarily need physical simulations and trial-and-error methods. One of the most interesting one-DOF origami mechanism known is the rigid-foldable degree-4 single vertex, a vertex where only four creases meet (cf. sections 3.5., 3.6., 4.6., 4.7., 6.3.3. for more about degree-4 vertices).

3.4. Rigid-Foldability

An origami pattern is rigid-foldable when it can be folded and unfolded without bending, stretching, intersecting or cutting the faces. This kind of origami structure is not like paper origami, because the faces must be infinitely rigid. In fact, they are more related to thick-origami (cf. section 6.1 for more about thick origami), because in the real world the stiffness of any object is strictly dependent by the material and the area of its section.

Thus, the question now is how to judge if a pattern is rigid-foldable or not. There are some special cases where the rigid-foldability can be easily evaluated only by watching the distribution of the creases in the CP and their mountain/valley assignments. For example, in a developable degree-4 single vertex pattern, the rigid-foldability is guaranteed if and only if there are three creases with the same mountain/valley assignment spaced with sector angles smaller than 180° , plus one crease with the opposite sign (Abel *et al.*, 2016).

Another example of a rigid-foldable, one-DOF pattern, which is easily recognizable is a pattern composed by multiple degree-4 rigid-foldable vertices joined in a linear array (without making closed loops). Tachi refers to these patterns as a “ $2 \times n$ quadrangle array(s)”, and he says that they are always one-DOF rigid-foldable mechanisms, he also asserts that an “ $m \times n$ (where $m > 2$) quadrangle array [...] yields over-constrained static structure or a redundant one-DOF mechanism because fold angles are multiply defined” (Tachi, 2011a). Thus, how do we judge if whether an $m \times n$ quadrangle array is rigid-foldable or not?

Abel *et al.* give us a preliminary answer: “Rigid foldability has been represented using extrinsic parameters of the folded state, e.g., the existence of a set of fold angles satisfying compatibility conditions, or the existence of intermediate state” (Abel *et al.*, 2016). With this statement, Abe *et al.* refer to two different methods to judge rigid-foldability of a pattern. The first approach refers to the method presented by Belcastro and Hull (Belcastro & Hull, 2002), where they evaluate the rigid-foldability by calculating the fold angle of a closed loop of creases, if the fold angles of the loop are all compatible, then the pattern is rigid-foldable. The second method reported by Abe *et al.* was stated by Tachi as follows (Tachi, 2010a): “If and only if BDFFPQ mesh, homeomorphic to a disk with more than one interior vertex, has one intermediate folded state, the surface is finitely rigid- foldable.”, which means that in an origami pattern if a flat-unfolded configuration and at least an intermediate folding configuration free of deformations exist, then it is guaranteed that the faces during the whole motion do not deform.

We can try applying a qualitative approach based on this last assumption. However, to be able to test this condition qualitatively, we need to simulate the folding motion of the pattern, usually we do this operation with physical models or with digital simulators (like the software Freeform origami), but in both cases we do some errors related to the elasticity of the material or to the tolerance of the software. Thus, this kind of tests based on simulations may return positive results even if the pattern is not rigidly foldable for a small amount.

If we merge the two approaches in one, we can judge if a pattern is rigid-foldable precisely by calculating the fold angles of all the closed loops of a

crease pattern in only one intermediate configuration, and if the compatibility is confirmed then the pattern is rigid-foldable.

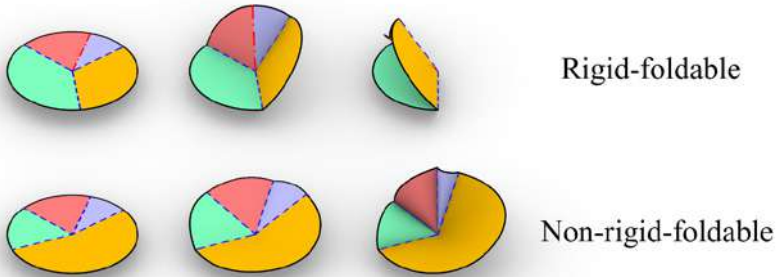


Figure 17 - Examples of a rigid-foldable and non-rigid-foldable degree-4 vertices.

3.4.1. Reciprocal Diagram to Judge the First-Order Rigid-Foldability

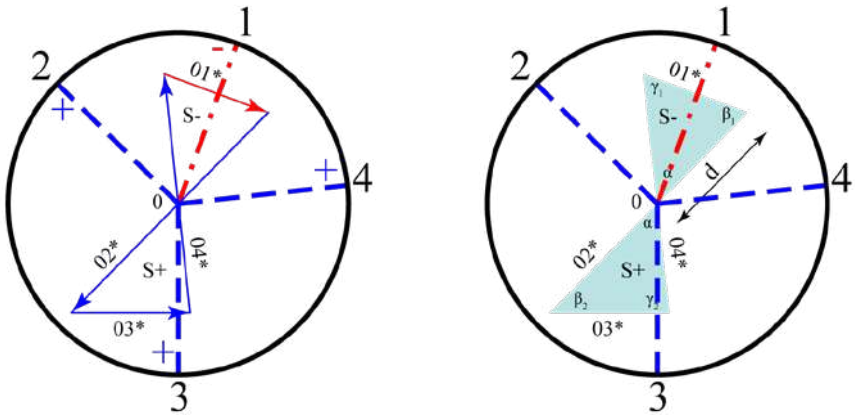


Figure 18 - Zero-area reciprocal diagram of a degree-4 single vertex pattern.

The reciprocal diagram is a well-known and powerful graphical tool for understanding and designing structural systems. It also has applications in the origami field, in which it has been introduced, for the first time, by D. A. Huffman in 1977 (Huffman, 1977). The reciprocal diagram is composed by straight segments perpendicular to each crease line in a CP, the perpendicular segments form a closed loop around the vertex, such that the direction of the perpen-

dicular segment is $\pm 90^\circ$ relatively to the direction of the creases. The sign, and thus the verse of the vector, depends on the mountain/valley assignment of the folds. The area of the resulting self-intersecting polygon must be equal to zero, as shown in Figure 18.

The reciprocal diagram has been investigated more recently by Demaine *et al.* in the paper *Zero-Area Reciprocal Diagram of Origami*, where they assert that the reciprocal diagram can be used to investigate the first order approximation of rigid origami:

We can view a polyhedral lifting as the first-order approximation of rigid origami, i.e., an origami surface is composed of rigid panels and rotational hinges connecting them together. Hence, it seems proper to use the reciprocal diagram for the analysis of rigid foldability and the design of rigidly foldable structures. However, it turns out soon that the existence of reciprocal diagram alone is a poor tool to judge rigid foldability of origami. For example, a degree-3 vertex has a nontrivial reciprocal diagram, but there is no valid folding for this pattern. (Demaine *et al.*, 2016)

According to what Demaine *et al.* assert the reciprocal diagram cannot be used to judge the rigid foldability in general, but it can be used as a tool to test the infinitesimal rigid-foldability, which is the property of an infinitely rigid creased surface to behave like a movable mechanism when it is close to the unfolded state.

We will see later in section 3.5.3 an alternative use of the reciprocal diagram as a tool to animate a degree-4 flat-foldable vertex.

3.5. Flat-Foldability

The flat-foldability is the property of an origami pattern to be collapsible into a plane without cutting, stretching or adding new creases to the pattern. Almost all the traditional figurative origami models are flat-foldable because the folding steps are performed flattening the model into a flat surface. Mathematically the flat foldability is described by simple rules that we are going to explain briefly in the next section.

3.5.1. Four Rules of Flat-Foldability – Kawasaki and Maekawa Conditions

The flat foldability of a crease pattern follows 4 simple rules, explained by Robert Lang in his presentation at “TED talks” in 2008 (Lang, 2008):

- At any interior vertex, $M - V = \pm 2$: mountain and valley creases always differ by 2, 2 more or 2 less.
- 2 colourability: crease patterns can be coloured with just two colours without ever having the same colour meeting.
- Alternate angles around a vertex sum to a straight line: considering angles between creases around a vertex of the crease pattern, numbering the angles on a circle all the even numbers head up to a straight line, the same happens for the odd numbers.
- No self-intersection at overlaps: no matter how you stack folds and sheets, a sheet can never penetrate a fold.

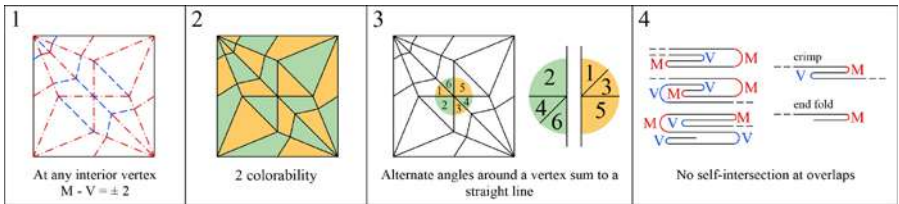


Figure 19 - four rules of flat foldability.

These four simple rules are everything we need for judging a flat foldable origami. They are based on mathematical theorems discovered in the last four decades. The first rule is described mathematically by the Maekawa theorem.

Theorem 1: Maekawa-Justin (Hull, 2003a)

$$M - V = \pm 2. \tag{5}$$

Where: M and V are respectively the mountain and valley creases adjacent to a vertex in a flat origami CP.

It means that the mountain folds and the valley folds in a flat-folded single vertex pattern differ always by 2. The second and third rules are described by the Kawasaki theorem. It was discovered by Kawasaki in 1989, although was also discovered independently by Justin in the same year. It gives a criterion to determine if a single vertex crease pattern can be folded to form a flat figure. The theorem statement is the following:

Theorem 2: Kawasaki-Justin theorem (Demaine & O'Rourke, 2007; Hull, 2003a)

A single-vertex crease pattern defined by angles $\theta_1 + \theta_2 + \dots + \theta_n = 360^\circ$ is flat foldable if and only if n is even and the sum of the odd angles (θ_{2i+1}) is equal to the sum of the even angles (θ_{2i}), or equivalently, either sum is equal to 180° :

$$\theta_1 + \theta_3 + \dots + \theta_{n-1} = \theta_2 + \theta_4 + \dots + \theta_n = 180^\circ. \quad (6)$$

It can also be written as follows:

$$\theta_1 - \theta_2 + \theta_3 - \dots - \theta_{2n} = 0. \quad (7)$$

The fact that the creases must be even implies that, for any flat-foldable crease pattern (even with multiple internal vertices), it is always possible to colour the regions between the creases with two colours, such that each crease separates two areas of different colours, this is always true for each side of the paper. Also, the fact that the sum of odd and even angles must be equal implies that either odds and evens angles sum to a straight line.

The Kawasaki theorem for developable flat-foldable vertices was generalized by Demaine in 2007 for non-developable pieces of paper as follows:

Theorem 3: Kawasaki-Justin-Demaine generalized for non-flat pieces of paper (Demaine & O'Rourke, 2007)

A single-vertex crease pattern defined by angles $\theta_1, \theta_2, \dots, \theta_n$, is flat foldable if and only if n is even and the alternating sum of the angles θ_i is equal to 0, 360° , or -360° :

$$\begin{aligned} \theta_1 - \theta_2 + \theta_3 - \theta_4 + \dots + \theta_{n-1} - \theta_n &= \\ = \sum_{i=1}^n (-1)^i \theta_i \in \{0, 360^\circ, -360^\circ\} \end{aligned} \quad (8)$$

The theorem 2 (Kawasaki-Justin) is included in the theorem 3 (Kawasaki-Justin-Demaine) and can be only used when working with flat pieces of paper.

For what concerns the self-intersection rule the problem can be approached dividing the cases into two simple groups of crease patterns: a pattern with parallel creases, and a pattern with creases incident in a single vertex. The flat-foldability of the pattern depends on many factors: the order of mountain and valley creases, the distance or the angle between consecutive creases. For example, if the pattern is composed by all parallel folds with alternated verses, it is always foldable, but if there are consecutive folds with the same verse the foldability depends on the se-

quence of the verses of the folds, and on the distances between the folds. The same can be said for a case where the creases converge in a single vertex, but instead of the distances, we must consider the angles between them. For a much comprehensive explanation in mathematical terms and more bibliographic references refer to the extensive investigation made by Demaine, in his book *Geometric folding algorithms: linkages, origami, polyhedral* (Demaine & O'Rourke, 2007).

3.5.2. Flat-Foldable Degree-4 Single Vertex – Relations Between Fold Angles

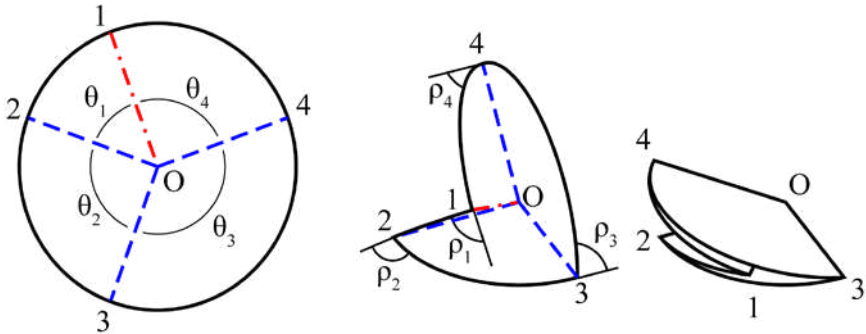


Figure 20 - Flat-foldable degree-4 vertex notation.

A particularly interesting flat-foldable pattern is the flat-foldable degree-4 single vertex. It is flat-foldable when it is characterized by a particular symmetry condition between opposite creases as shown in Figure 20. Because this particular pattern has only one-DOF every fold angle is univocally related to all the other fold angles. In this section, we are going to report the well-known formulations that relate the fold angles in a degree-4 vertex, and we are going to apply them to a folded surface with a Grasshopper's definition to test their correctness.

The relationship between the fold angles in a degree-4 vertex has been studied by many researchers such as David A. Huffman, Thomas C. Hull, Robert J. Lang and Tomohiro Tachi (Huffman, 1976; Hull, 2006; Lang *et al.*, 2016; Tachi, 2009). In particular, Lang in his paper *Single Degree-of-Freedom Rigidly Foldable Cut Origami Flashers* asserts that the vertex is flat foldable if and only if:

$$\theta_1 + \theta_3 = \theta_2 + \theta_4 = \pi. \tag{9}$$

and for flat foldable vertices the major fold angles are equal:

$$\rho_2 = \rho_4. \tag{10}$$

and the minor fold angles are equal but with a different sign:

$$\rho_1 = -\rho_3. \tag{11}$$

Furthermore, he derives the known formulations of the fold angles of a flat-foldable degree-4 vertex proposing a particularly simple expression that describes the relationship between adjacent fold angles as follows:

$$\frac{\tan\frac{1}{2}\rho_2}{\tan\frac{1}{2}\rho_1} = -\frac{\tan\frac{1}{2}\rho_2}{\tan\frac{1}{2}\rho_3} = \frac{\tan\frac{1}{2}\rho_4}{\tan\frac{1}{2}\rho_1} = -\frac{\tan\frac{1}{2}\rho_4}{\tan\frac{1}{2}\rho_3} = \frac{\sin\frac{1}{2(\theta_1+\theta_2)}}{\sin\frac{1}{2(\theta_1-\theta_2)}}. \tag{12}$$

This expression can be used to animate the pattern with Grasshopper by writing the following rearrangement of the expression (Expression 13) into the “Expression” component and using the result as the rotation value of the corresponding adjacent faces, as shown in Figure 21. The expression must be written in the following form, in order to be readable by the software:

$$2 * \text{Atan}(\sin((H1 + H2)/2)/(\sin((H1 - H2)/2))) * \tan(R1/2)) = R2. \tag{13}$$

Where:

Atan (...) is *tan*⁻¹ (...)

H1 is θ_1

H2 is θ_2

R1 is ρ_1

R2 is ρ_2

Whit this expression we calculate ρ_2 from ρ_1 , θ_1 and θ_2 , and because $\rho_3 = -\rho_1$ and $\rho_4 = \rho_2$ we already have all the four angles and we can animate all the faces of the degree-4 flat-foldable vertex.

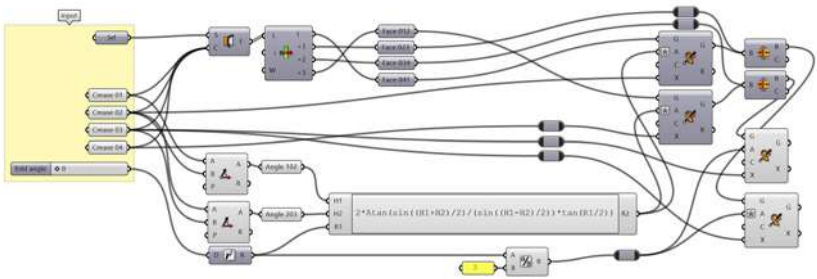


Figure 21 - Generative algorithm that animates a degree-4 flat-foldable vertex by calculating the fold angles with mathematical formulations.

3.5.3. Flat-Foldable Degree-4 Single Vertex – Reciprocal Diagram and Fold Angles

We anticipated in section 3.4.1 that the reciprocal diagram can also be used as a tool to identify the fold angles of a flat-foldable degree-4 vertex, this is possible because the lengths of the segments of the reciprocal diagram are strictly related to the sector angles, thus they have a relation to the expression (12).

The formulation (12) that relates consecutive angles can be rearranged in the following form:

$$\rho_i = 2 \tan^{-1}(k * \tan \frac{\rho_{i+1}}{2}). \tag{14}$$

Where k is:

$$k = \frac{\sin \frac{1}{2}(\theta_1 + \theta_2)}{\sin \frac{1}{2}(\theta_1 - \theta_2)}. \tag{15}$$

By experimental method, we discovered that the constant k , in a flat-foldable degree-4 vertex, can also be calculated by dividing the length of the longest segment with the length of the shortest segment of the reciprocal diagram as follows:

$$k = \frac{\text{max edge length}}{\text{min edge length}}. \tag{16}$$

To draw parametrically a reciprocal diagram, we need to identify the single crease with opposite sign and draw a vector with an angle equal to $+90^\circ$ relative to that crease. We do the same thing with the other three creases but rotating the relative vectors by an angle of -90° . Then we draw a line along the direction of each vector, and we extend each one of them until they meet the relative adjacent two lines, like so we find 4 intersection points. We connect the 4 points with a polyline following the verse of the vectors, obtaining a self-intersecting closed loop of four edges shaped like a ribbon. The ribbon forms two triangular areas, which need to be equalized to make a proper reciprocal diagram with zero-area. To equalize the areas, we translate one of the edges keeping it parallel to itself while extending the two adjacent edges of the ribbon until the two triangles have the same area. As shown in Figure 18, if we fix the position of the vectors 02^* , 03^* , 04^* we can find the position of 01^* by calculating d using the internal angles of the polygon as follows:

$$d = \sqrt{\frac{2S \sin \gamma_1}{\sin \alpha \sin \beta_1}}. \tag{17}$$

Once drawn the reciprocal diagram, we calculate k by measuring the maximum and minimum lengths of the edges. In a flat-foldable degree-4 vertex the reciprocal diagram is always symmetric, thus there are two equal segments that are the longest and two equal segments that are the shortest, we just pick one of each pair of segments to calculate k . With k , obtained with parametrical method (as shown in the full generative algorithm shown in Figure 22, we can now calculate all the fold angles at any folded state in any flat-foldable degree-4 vertex.

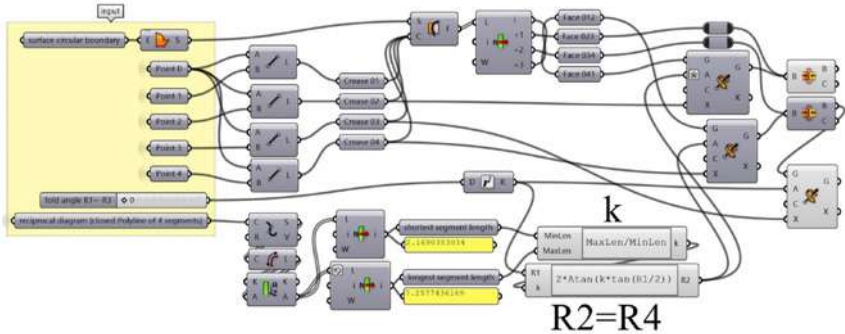


Figure 22 - Generative algorithm for the animation of a flat-foldable degree-4 vertex calculating k by using the reciprocal diagram.

3.5.4. Calculating k Through Reciprocal Diagram – Proved by Experimental Method

As stated in the previous section we proved by experimental method that the reciprocal diagram could be used to calculate the constant k needed in the equation 14 that allows to calculate all the fold angles of a flat-foldable degree-4 vertex. We proved it with the parametrical approach as follows.

Construct parametrically the reciprocal diagram of a given degree-4 vertex and calculate the constant k as explained in section 3.5.3. Choose one controller crease and fix its fold angle to a value between 0° and 180° . Calculate the fold angle of one of the adjacent creases applying the expression 14 using the constant k just calculated. Because the opposite fold angles in a degree-4 vertex are always equal we already have all the fold angles that we need to univocally define the position of each face of the degree-4 vertex. Now we rotate the faces around the adjacent creases to configure the flat pattern into the target folded configuration that matches the fold angle of the controller fold. If the angles are correct the faces will be configured into a closed loop of planar faces with no deformations in the faces.

Now, because the whole process is parametric, we can animate the folding and unfolding of the vertex by changing the input fold angle. By animating the vertex, we can prove that the faces always make a closed loop at any value of the controller fold angle. The preservation of the shape of the faces is guaranteed by construction, the preservation of the closed loop, instead, is verified by setting off the animation and testing the continuity between adjacent faces frame-by-frame. To prove that this approach works with every flat-foldable vertex we tested different flat-foldable degree-4 patterns at limit cases and at intermediate symmetric cases (we avoided trivial cases like symmetry reflections of the patterns and flipped mountain/valley assignment) as shown in Figure 23.

Of course, we can prove this also by the analytical method by relating the expression 14 with the construction of the reciprocal diagram by applying simple trigonometry rules and proportions between angles and lengths.

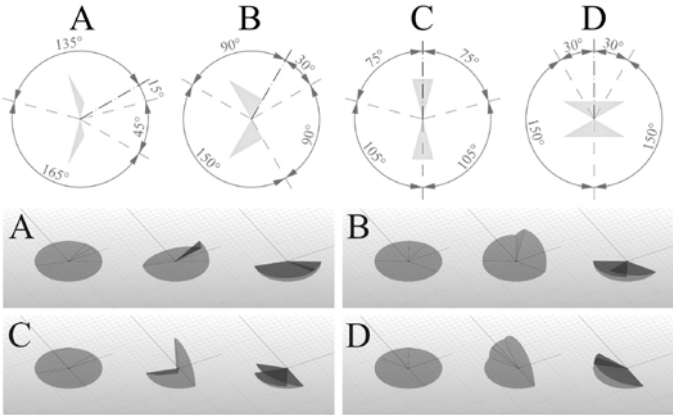


Figure 23 - Examples of test cases of asymmetric and symmetric flat-foldable vertices animated by calculating the constant k by dividing the longest and shortest segment of the reciprocal diagram.

3.6. Non-Flat-Foldability¹

A non-flat-foldable pattern is a pattern that cannot be flat-collapsed into the plane without adding new creases. It is the counterpart of the flat-foldable pattern introduced in the previous section, and it has some interesting properties that may be used into a movable mechanism. One of the most useful property is that it blocks at a certain three-dimensional configuration. If the pattern has one-DOF the configuration is univocal (considering a given mountain/valley

assignment) and this makes it very useful to design deployable or compactable objects. Two examples of self-blocking foldable mechanisms, designed using non-flat-foldable degree-4 vertices, are presented in section 6.3 and section 6.2 and they are a rigid-foldable chair and a rigid-foldable ladder.

The blocked configuration is often called “locked”, “arrested” (Buffart *et al.*, 2017), “binding” (Lang, 2018), and recently has been also called “Blockfaltung” by Lang (Lang, 2018). We use “blocked” (Klett & Drechsler, 2011) because some of the terms already used may be interpreted as configurations where the movement is obstructed in both directions, which is not the case.

3.6.1. Non-Flat-Foldable Degree-4 Single Vertex – Huffman’s Formulations

Flat-foldable degree-4 vertices are special cases of generic degree-4 vertices. If we trace 4 creases that converge into a point with random angles, it is more probable to come up with a non-flat-foldable degree-4 vertex instead of a flat-foldable one. Thus, we can say that the major part of degree-4 vertices is non-flat-foldable. Furthermore, as we previously stated, non-flat-foldable vertices may be very useful for practical applications, for this reason, we want to explain carefully how they work and how to treat them mathematically (in this section) and geometrically (in sections 4.6.2. and 4.6.3.).

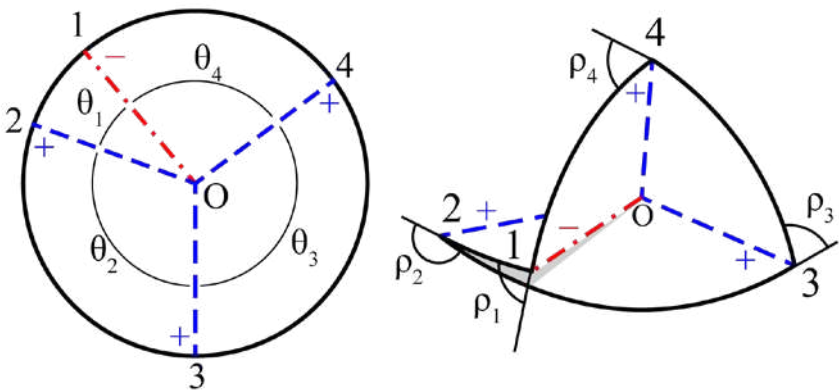


Figure 24 - Non-flat-foldable degree-4 single vertex notation.

Huffman in the paper *Curvature and Creases: A Primer on Paper* (Huffman, 1976) generalised the formulation, reported in section 3.5.2, for every degree-4 vertex. In fact, the formulations for flat-foldable degree-4 vertices are a simplification of the Huffman’s formulations which work both for flat and non-flat foldable vertices.

The following formulations relate ρ_4 to ρ_2 and ρ_3 to ρ_1 :

$$\frac{\sin^2 \frac{\rho_4}{2}}{\sin^2 \frac{\rho_2}{2}} = \frac{\sin \theta_2 \sin \theta_1}{\sin \theta_3 \sin \theta_4} \quad (18)$$

$$\frac{\sin^2 \frac{\rho_3}{2}}{\sin^2 \frac{\rho_1}{2}} = \frac{\sin \theta_4 \sin \theta_1}{\sin \theta_2 \sin \theta_3} \quad (19)$$

Huffman also associates ρ_2 to ρ_3 by a “*very difficult*”²² derivation:

$$\begin{aligned} & \left[\begin{array}{l} 1 \mp \sqrt{\frac{\sin \theta_3 \sin \theta_1}{\sin \theta_2 \sin \theta_4}} \sqrt{\frac{1 - \frac{\sin \theta_2 \sin \theta_1}{\sin \theta_3 \sin \theta_4} \sin^2 \frac{\rho_2}{2}}{1 - \sin^2 \frac{\rho_2}{2}}} \\ 1 \pm \sqrt{\frac{\sin \theta_3 \sin \theta_1}{\sin \theta_2 \sin \theta_4}} \sqrt{\frac{1 - \frac{\sin \theta_4 \sin \theta_1}{\sin \theta_2 \sin \theta_3} \sin^2 \frac{\rho_1}{2}}{1 - \sin^2 \frac{\rho_1}{2}}} \end{array} \right] = \\ & = 1 - \frac{\sin \theta_3 \sin \theta_1}{\sin \theta_2 \sin \theta_4} \end{aligned} \quad (20)$$

Rearranging these formulations in the forms $\rho_2, \rho_3 = f(\rho_1, \theta_1, \theta_2, \theta_3, \theta_4)$ and (once found ρ_2) $\rho_4 = f(\rho_2, \theta_1, \theta_2, \theta_3, \theta_4)$ we can find all the fold angles knowing only ρ_1 and the sector angles $\theta_1, \theta_2, \theta_3, \theta_4$.

Unfortunately, it is not trivial rearranging these functions, so we solved them with the help of mathematical software to calculate all the alternative forms and the possible solutions, which could be more than one depending on the symmetry conditions.

For the rearranged forms, to be able to input them into Grasshopper, we used the original notation by Huffman even if it is not the standard notation of a degree-4 vertex, because it uses ASCII characters that are the only character that the “Expression” component in Grasshopper accepts, thus the factors change as follows:

$$\rho_1 = q, \rho_2 = n, \rho_3 = p, \rho_4 = m, \theta_1 = D, \theta_2 = C, \theta_3 = A, \theta_4 = B.$$

Therefore, knowing n, A, B, C, D we can calculate q, p and m . For the purposes of this algorithm, we only consider interesting solutions, which are not simple mirror reflections of the only 2 possible folding modes, that can be found simply changing the sign of the result of the interesting solutions. The variable q has two interesting solutions, one for each possible folding mode, variable p is related to q . For flat-foldable degree-4 vertices, the interesting solution is only one as well as the folding mode. The variable m has only one interesting solution. Below we report the solutions of q, p , and m transcribed in ASCII characters solved in function of n, A, B, C, D . The fold angle n is the angle of the controller crease which is conveniently also the crease that blocks first in this case.

Fold angle q , solution 1, in function of n, A, B, C, D .

$$\begin{aligned}
& -2 * \text{Acos}(-1 * \text{Pow}(-1 * \text{Pow}(\text{Sin}(C), -1) - \\
& 2 * \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), - \\
& 1), 3) * \text{Pow}(\text{Sin}(D), 2) - \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), - \\
& 1), 4) * \text{Pow}(1 * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), - \\
& 1), 3) * \text{Pow}(\text{Sin}(B), 2) * \text{Pow}(\text{Sin}(D), 2) \\
& - 2 * \text{Pow}(1 * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(\text{Sin}(C), -1) * \text{Pow}(\text{Sin}(D), 2) * \text{Pow}(\text{Tan}(n/2.), 2) + 2 * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), - \\
& 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Sin}(D), \\
& 3) * \text{Pow}(\text{Tan}(n/2.), 2) - 2 * \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), - \\
& 1), 2) * \text{Pow}(1 * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), - \\
& 1), 3) * \text{Pow}(\text{Sin}(D), 4) * \text{Pow}(\text{Tan}(n/2.), 2) - \\
& 1 * \text{Pow}(1 * \text{Pow}(\text{Sin}(A), -1), 4) * \text{Pow}(\text{Sin}(C), -1) * \text{Pow}(\text{Sin}(D), 4) * \text{Pow}(\text{Tan}(n/2.), 4) + 1 * \text{Pow}(1 * \text{Pow}(\text{Sin}(A), - \\
& 1), 3) * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), -1), 2) * \text{Pow}(\text{Sin}(B), - \\
& 1) * \text{Pow}(\text{Sin}(D), 5) * \text{Pow}(\text{Tan}(n/2.), 4) + \\
& 1 * \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), -1), 4) * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), - \\
& 1), 4) * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(D), 3) * \text{Sin}(B) + \\
& 2 * \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(1 * \text{Pow}(\text{Sin}(A), - \\
& 1), 3) * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), -1), 2) * \text{Pow}(\text{Sin}(D), 3) * \text{Pow} \\
& \text{w}(\text{Tan}(n/2.), 2) * \text{Sin}(B) + 1 * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), - \\
& 1), 2) * \text{Pow}(\text{Sin}(B), -1) * \text{Sin}(A) * \text{Sin}(D) + \\
& 2 * \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(1 * \text{Pow}(\text{Sin}(C), - \\
& 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(- \\
& 1 * \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(\text{Sin}(C), -1) \\
& - 2 * \text{Pow}(1 * \text{Pow}(\text{Cos}(n/2.), -
\end{aligned}$$

$$\begin{aligned}
& 1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 3) * \text{Pow}(\text{Sin}(D), 2) \\
& \quad - 1. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
1), 2) * \text{Pow}(\text{Sin}(B), 2) * \text{Pow}(\text{Sin}(C), -1) * \text{Pow}(\text{Sin}(D), 2) \\
& \quad - 2. * \text{Pow}(\text{Sin}(C), -1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Po} \\
\text{w}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), - \\
1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(C) \\
& \quad * \text{Sin}(D), 0.5) + 2. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
1), 2) * \text{Pow}(\text{Sin}(C), -1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Po} \\
\text{w}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), - \\
1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(\\
C) * \text{Sin}(D), 0.5) - 4. * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
1), 2) * \text{Pow}(\text{Sin}(C), -1) * \text{Pow}(\text{Sin}(D), 2) * \text{Pow}(\text{T} \\
\text{an}(n/2.), 2) + 2. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(\text{Sin}(C), - \\
1) * \text{Pow}(\text{Sin}(D), 2) * \text{Pow}(\text{Tan}(n/2.), 2) + \\
2. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 2) * \text{Pow}(\text{Sin}(A), - \\
1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Sin}(D), 3) * \text{Pow}(\text{Ta} \\
n(n/2.), 2) - 2 * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
1), 3) * \text{Pow}(\text{Sin}(D), 4) * \text{Pow}(\text{Tan}(n/2.), 2) + \\
2. * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(\text{Sin}(C), - \\
1) * \text{Pow}(\text{Sin}(D), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5 \\
) * \text{Pow}(\text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - \\
1. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/ \\
2.), 2) * \text{Sin}(C) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Tan}(n/2.), 2) \\
- 1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 3) * \text{Pow}(\text{Sin}(B), - \\
1) * \text{Pow}(\text{Sin}(D), 3) * \text{Pow}(\text{Tan}(n/2.), 4) + \\
1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 3) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
1), 2) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Sin}(D), 5) * \text{Pow}(\text{T} \\
\text{an}(n/2.), 4) + 1. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
1), 4) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 4) * \text{Pow}(\text{Sin}(A), - \\
1) * \text{Pow}(\text{Sin}(D), 3) * \text{Sin}(B) + \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
1), 3) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 2) * \text{Pow}(\text{Sin}(D), 3 \\
) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(B) + 1. * \text{Pow}(\text{Sin}(A), -
\end{aligned}$$

$$\begin{aligned}
& 1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(D) \\
& \quad - 2. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), - \\
& \quad 1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(C), - \\
& \quad 1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Pow}(1. * \text{Pow}(\text{Cos} \\
& \quad (n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), - \\
& \quad 1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(C) * \text{Sin}(D), 0.5) * \text{Pow}(\text{T} \\
& \quad \text{an}(n/2.), 2) * \text{Sin}(D) + 1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(B), -1) * \text{Sin}(A) * \text{Sin}(D) + \\
& 4. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(\text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Sin}(B) * \text{Sin}(D) - \\
& 1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Sin}(B) * \text{Sin}(D) \\
& - 1. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 4) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Sin}(B) * \text{Sin}(D) - \\
& 2. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(\text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Po} \\
& \quad \text{w}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(C) * \text{Sin} \\
& \quad (D), 0.5) * \text{Sin}(B) * \text{Sin}(D) + 2. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Po} \\
& \quad \text{w}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(C) * \text{Sin}(D), 0 \\
& \quad .5) * \text{Sin}(B) * \text{Sin}(D), -0.5)
\end{aligned}$$

Fold angle q , solution 2 in function of n, A, B, C, D .

$$\begin{aligned}
& 2 * \text{AcOs}(1. * \text{Pow}(-1. * \text{Pow}(\text{Sin}(C), -1) - \\
& 2 * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 3) * \text{Pow}(\text{Sin}(D), 2) - \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
& 1), 4) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 3) * \text{Pow}(\text{Sin}(B), 2) * \text{Pow}(\text{Sin}(D), 2) \\
& - 2. * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(\text{Sin}(C), -1) * \text{Pow}(\text{S} \\
& \quad \text{in}(D), 2) * \text{Pow}(\text{Tan}(n/2.), 2) + 2. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Sin}(D), \\
& \quad 3) * \text{Pow}(\text{Tan}(n/2.), 2) - 2 * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
& 1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 3) * \text{Pow}(\text{Sin}(D), 4) * \text{Pow}(\text{Tan}(n/2.), 2) -
\end{aligned}$$

$$\begin{aligned}
& 1.*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 4)*\text{Pow}(\text{Sin}(C), -1)*\text{Pow}(\text{Sin}(D), 4)*\text{Pow}(\text{Tan}(n/2.), 4) + 1.*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 3)*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 2)*\text{Pow}(\text{Sin}(B), -1)*\text{Pow}(\text{Sin}(D), 5)*\text{Pow}(\text{Tan}(n/2.), 4) + \\
& 1.*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 4)*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 4)*\text{Pow}(\text{Sin}(A), -1)*\text{Pow}(\text{Sin}(D), 3)*\text{Sin}(B) + \\
& 2*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 3)*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 2)*\text{Pow}(\text{Sin}(D), 3)*\text{Pow}(\text{Tan}(n/2.), 2)*\text{Sin}(B) + 1.*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 2)*\text{Pow}(\text{Sin}(B), -1)*\text{Sin}(A)*\text{Sin}(D) + \\
& 2.*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 2)*\text{Pow}(\text{Sin}(A), -1)*\text{Sin}(B)*\text{Sin}(D), 0.5)*\text{Pow}(-1.*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(\text{Sin}(C), -1) \\
& \quad - 2*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 3)*\text{Pow}(\text{Sin}(D), 2) \\
& \quad - 1.*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 2)*\text{Pow}(\text{Sin}(B), 2)*\text{Pow}(\text{Sin}(C), -1)*\text{Pow}(\text{Sin}(D), 2) \\
& \quad + 2.*\text{Pow}(\text{Sin}(C), -1)*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1)*\text{Pow}(\text{Sin}(C), -1)*\text{Sin}(B)*\text{Sin}(D), 0.5)*\text{Pow}(\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2) - 1.*\text{Pow}(\text{Sin}(A), -1)*\text{Pow}(\text{Sin}(B), -1)*\text{Pow}(\text{Tan}(n/2.), 2)*\text{Sin}(C)*\text{Sin}(D), 0.5) - 2.*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(\text{Sin}(C), -1)*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1)*\text{Pow}(\text{Sin}(C), -1)*\text{Sin}(B)*\text{Sin}(D), 0.5)*\text{Pow}(\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2) - 1.*\text{Pow}(\text{Sin}(A), -1)*\text{Pow}(\text{Sin}(B), -1)*\text{Pow}(\text{Tan}(n/2.), 2)*\text{Sin}(C)*\text{Sin}(D), 0.5) - 4.*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 2)*\text{Pow}(\text{Sin}(C), -1)*\text{Pow}(\text{Sin}(D), 2)*\text{Pow}(\text{Tan}(n/2.), 2) + 2.*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 2)*\text{Pow}(\text{Sin}(C), -1)*\text{Pow}(\text{Sin}(D), 2)*\text{Pow}(\text{Tan}(n/2.), 2) + \\
& 2.*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 2)*\text{Pow}(\text{Sin}(A), -1)*\text{Pow}(\text{Sin}(B), -1)*\text{Pow}(\text{Sin}(D), 3)*\text{Pow}(\text{Tan}(n/2.), 2) - 2*\text{Pow}(1.*\text{Pow}(\text{Cos}(n/2.), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 2)*\text{Pow}(1.*\text{Pow}(\text{Sin}(C), -1), 3)*\text{Pow}(\text{Sin}(D), 4)*\text{Pow}(\text{Tan}(n/2.), 2) \\
& \quad - 2.*\text{Pow}(1.*\text{Pow}(\text{Sin}(A), -1), 2)*\text{Pow}(\text{Sin}(C), -1)
\end{aligned}$$

$$\begin{aligned}
& 1) * \text{Pow}(\text{Sin}(D), 2) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5 \\
& \quad) * \text{Pow}(\text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - \\
& 1. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/ \\
& \quad 2.), 2) * \text{Sin}(C) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Tan}(n/2.), 2) \\
& - 1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 3) * \text{Pow}(\text{Sin}(B), - \\
& \quad 1) * \text{Pow}(\text{Sin}(D), 3) * \text{Pow}(\text{Tan}(n/2.), 4) + \\
& 1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1), 3) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Sin}(D), 5) * \text{Pow}(\text{T} \\
& \quad \text{an}(n/2.), 4) + 1. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), - \\
& \quad 1), 4) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 4) * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(D), 3) * \text{Sin}(B) + \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1), 3) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 2) * \text{Pow}(\text{Sin}(D), 3 \\
& \quad) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(B) + 1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(D) \\
& \quad + 2. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), - \\
& \quad 1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(C), - \\
& \quad 1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Pow}(1. * \text{Pow}(\text{Cos} \\
& \quad (n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(\text{Sin}(B), - \\
& \quad 1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(C) * \text{Sin}(D), 0.5) * \text{Pow}(\text{T} \\
& \quad \text{an}(n/2.), 2) * \text{Sin}(D) + 1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(B), -1) * \text{Sin}(A) * \text{Sin}(D) + \\
& 4. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(\text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Sin}(B) * \text{Sin}(D) - \\
& 1. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), -1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Sin}(B) * \text{Sin}(D) \\
& - 1. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 4) * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Sin}(B) * \text{Sin}(D) + \\
& 2. * \text{Pow}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) * \text{Pow}(\text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Po} \\
& \quad \text{w}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(C) * \text{Sin}(\\
& \quad D), 0.5) * \text{Sin}(B) * \text{Sin}(D) - 2. * \text{Pow}(1. * \text{Pow}(\text{Sin}(C), - \\
& \quad 1), 2) * \text{Pow}(\text{Sin}(A), -1) * \text{Pow}(1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(C), -1) * \text{Sin}(B) * \text{Sin}(D), 0.5) * \text{Pow}(\text{Po} \\
& \quad \text{w}(1. * \text{Pow}(\text{Cos}(n/2.), -1), 2) - 1. * \text{Pow}(\text{Sin}(A), - \\
& \quad 1) * \text{Pow}(\text{Sin}(B), -1) * \text{Pow}(\text{Tan}(n/2.), 2) * \text{Sin}(C) * \text{Sin}(D), 0 \\
& \quad .5) * \text{Sin}(B) * \text{Sin}(D), -0.5))
\end{aligned}$$

Fold angle p is in function of q, A, B, C, D .

$$2 * \text{Asin}(\text{Pow}(\text{Pow}(\text{Sin}(q/2), 2) * \text{Sin}(A) * \text{Sin}(D)) / (\text{Sin}(C) * \text{Sin}(B)), 0.5))$$

Fold angle m in function of n, A, B, C, D .

$$2 * \text{Asin}(\text{Pow}(\text{Pow}(\text{Sin}(n/2), 2) * \text{Sin}(C) * \text{Sin}(D)) / (\text{Sin}(A) * \text{Sin}(B)), 0.5))$$

These textual solutions have been generated using the software Mathematica by Wolfram research (WolframResearch, n.d.). Obviously they could have been simplified with subfunctions or they could have been written in a slimmer manner using alternative forms, but we decided to leave them as they have been returned by Mathematica because this is what the application returns as a first calculation step, which might convince a nonprofessional user to judge it as too complex to handle, making him/her giving up at this stage.

This kind of intimidating steps is less likely to happen while approaching the same problem with the geometrical constructive approach, that we are going to present in sections 4.6.2. and 4.6.3.

This case study is a clear example of the greater complexity of the algebraic approach compared to the graphical constructive approach.

3.6.2. The Blocking Crease

The blocking crease is the first crease that hit 180° in a CP. In a degree-4 non-flat-foldable single vertex the blocking crease is always one, but they can be more than one in a CP with multiple vertices or multiple degrees of freedom. The identification of the blocking crease is important for animating the folding and unfolding of a one-DOF pattern because if it is used as the controller crease and the domain of its fold angle is limited between 0° and 180° it prevents the pattern to self-intersect.

To identify the blocking crease before folding the pattern we could use the well-known Huffman formulations that would allow us to find the maximum fold angles of every crease, but because those formulations are valid at any time t , they are unnecessarily complex for finding only the fold angles at blocked state.

Thus, in the following sections, we present a simplified method using spherical trigonometry and triangular inequalities on the sphere, that will allow us

to identify the crease that blocks first from the unfolded pattern, in a degree-4 single vertex, analysing exclusively the blocked state.

Adding this procedure at the beginning of the algorithms explained in 4.6.3 or 3.6.1 would solve the problem of the identification of the crease that blocks first before folding anything. This would allow us to choose the optimal controller crease from early stages of the animation process. Before explaining this new approach, we need to recall some concepts of the kinematics of a degree-4 vertex.

3.6.3. Understanding the Kinematics of a Non-Flat-Foldable Developable Degree-4 Vertex

A non-flat-foldable degree-4 developable vertex is a vertex with four incident creases forming a convex angle to adjacent ones, which does not satisfy Kawasaki's condition, i.e., the alternating sum of angles does not sum to 0° . The fold angle $\rho \in [-180^\circ, 180^\circ]$ is the angle between the normal vectors of adjacent faces. The sector angle $0 < \theta < 180^\circ$ is the angle between adjacent creases. First, we briefly characterize the kinematics of a non-flat-foldable degree-4 vertex. In the following statements, it is assumed that the faces can pass through each other, and any state with two coplanar faces is called a "blocked state".

- A non-flat-foldable degree-4 vertex forms a one-DOF mechanism with no bifurcations in its fold angle function except at the completely unfolded states.
- At the unfolded states, at most two folding modes intersect. The mountain/valley assignment of each folding mode always has three creases with the same signs forming a "Y" shape (convex angle to each other) and one crease with opposite sign (Abel *et al.*, 2016). For non-flat-foldable degree-4 vertices, exactly two creases can be the oppositely signed crease, one for each folding mode.
- In either mode, only one single crease folds flat first, and it blocks the movement when self-intersections are avoided. Hence, the state is called the "first blocked state".
- Ignoring self-intersection, if the folding motion continues, the faces can pass through each other and the single vertex pattern will reach another configuration where two different faces are coplanar, thus we call it the "second blocked state".
- The signs flip after the self-intersection, and the second blocked state is equivalent to the first blocked state of the other folding mode (refer to Figure 25).

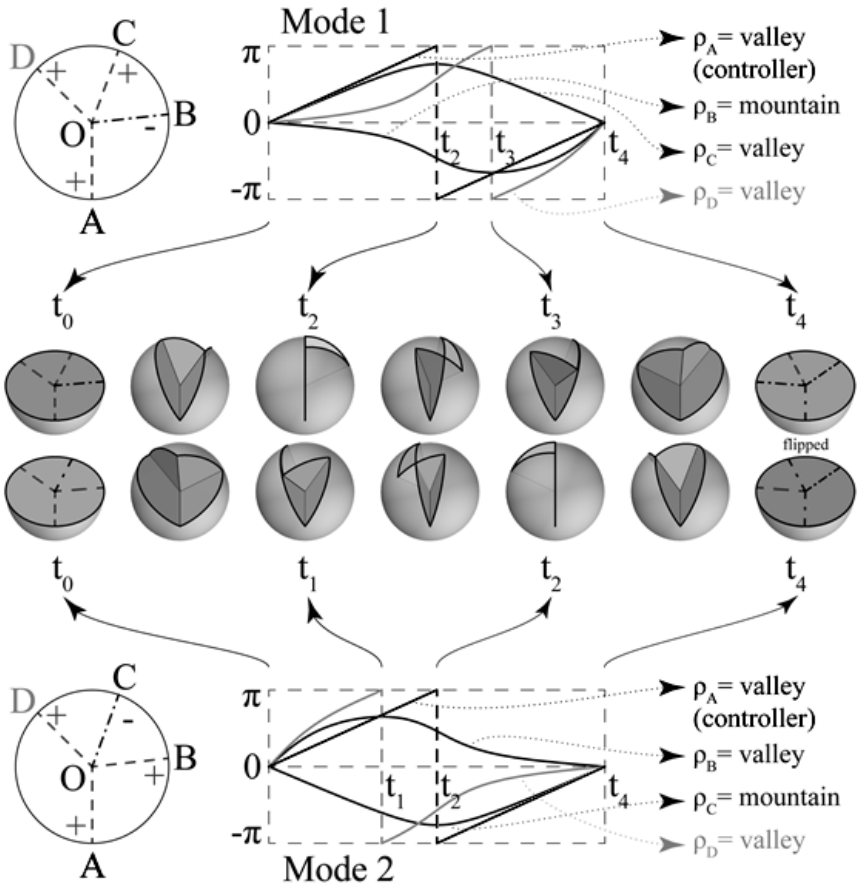


Figure 25 - Folding animation snapshots, and fold angle plots of a given degree-4 vertex (the folding mode one has the same kinematics as mode two played backwards with mirrored mountain valley assignment). Notice that the unfolded state in t_4 is flipped (upside-down compared to the flat-folded state in t_0).

Figure 25 shows the fold angles variation over time when ρ_A is the controller crease represented as a linear function of time. The fold angle functions of the other creases depend on the controller crease because it is a one-DOF mechanism. Focus on folding mode two. The folded state reaches the first blocked state at t_1 when the ρ_D function jumps from $+180^\circ$ (π) to -180° ($-\pi$). The jump happens because after that point, the faces pass through each other and the sign of the crease flips (passing from valley to mountain in this case). In t_2 , the folded surface reaches the second blocking configuration when ρ_A jumps from $+180^\circ$

to -180° . While two creases A and D rotate 360° , the other creases B and C rotate to some amount smaller than 180° and go back to θ . Function ρ_B reaches its maximum at t_1 , and ρ_C reaches its minimum at t_2 . At θ and t_4 all the fold angles go to θ at the same time, because at θ and t_4 the vertex is completely unfolded.

Because of the two creases ρ_A and ρ_D flipped their signs, when the pattern again reaches the flat state (at t_4), the mountain/valley assignment has changed to that of mode one except that all the signs are flipped. Thus, folding mode 1 is given by the backward play of folding mode two with mirrored mountain/valley assignment.

3.6.4. First Blocking Crease in a Developable Degree-4 Vertex

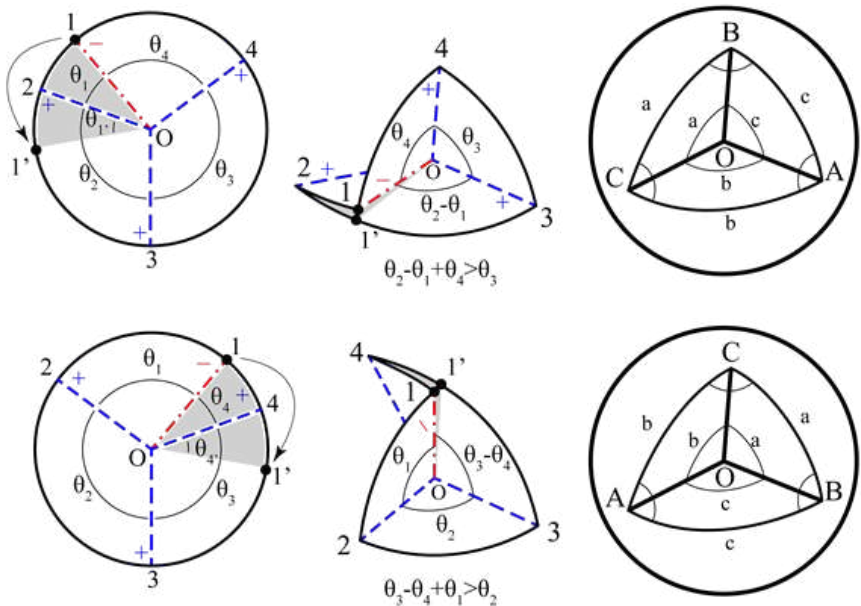


Figure 26 - Only two possible blocked cases in a generic degree-4 vertex (in the upper one the blocking crease is O2; in the bottom one the blocking crease is O4).

The following considerations prove that in a developable non-flat-foldable degree-4 single vertex pattern there are only two possible creases that can be the candidate creases among which there is only one that hit 180° first.

Focus on the first blocking state. The mountain/valley assignment must guarantee rigid foldability from the unfolded state, so there must be three creas-

es with the same sign and one crease with opposite sign (Abel *et al.*, 2016). Assume that they are three valleys and one mountain, without loss of generality.

In the blocked state, only one crease is flat-folded in non-flat-foldable cases. This is true because if at least two creases are flat at the same time, then all four creases are coplanar in this configuration, and the pattern would be flat-foldable.

Refer to Figure 26. Consider the first blocked state. Because only one crease is fully folded, a three-faced pyramid (OABC) is formed. Two edges of the pyramid (\overline{OA} and \overline{OB}) are formed by faces adjacent in the unfolded state, and one edge (\overline{OC}) is formed by two faces non-adjacent in the unfolded state but touching in the folded state. The former two edges have the same mountain/valley assignment, which is valley, and the other edge (\overline{OC}) is made by two creases of opposite signs ($\overline{O1}$ and $\overline{O2}$ or $\overline{O1}$ and $\overline{O4}$), one of which folds flat.

The turn angle at C is positive (valley) because ABC is a spherical triangle, but this should be equal to the summation of fold angles of the two creases forming \overline{OC} . This means that the crease hitting 180° (the one with bigger absolute value) is positive (valley). So, the crease that hits 180° first must be subsequent or precedent to the crease with opposite sign. This limits the solutions to two possible candidate flat-folded creases ($\overline{O2}$ or $\overline{O4}$).

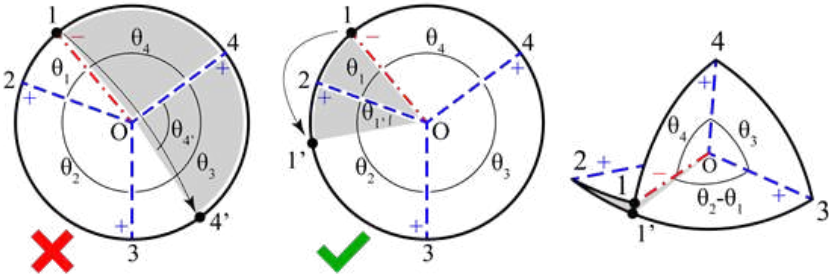


Figure 27 - Testing which crease blocks the movement first, among the two possible candidates.

Consider having an unfolded pattern where the crease that blocks the movement is still unknown (Figure 27). Considering the previous assumptions, it is known that the crease that hits 180° first is always one of the two valley creases adjacent to the single mountain crease. The following formulations are aimed to test which of the two identified candidate creases is the actual one that hits 180° first, without the need of folding the pattern. Spherical trigonometry and triangular inequality on a sphere are used. The pyramid can be considered as a spherical triangle on a unit sphere. Triangular inequality on a sphere is given by:

$$+b > c \text{ and } b + c > a \text{ and } c + a > b. \quad (21)$$

Because:

$$a, b, c < 180^\circ. \quad (22)$$

The mountain crease is called 1, and the valley creases are called 2, 3, and 4 counterclockwise. Considering the previous assumptions, the flat-folded crease is in general either 2 or 4. If crease 2 folds flat first, then creases 3 and 4 form edges \overline{OA} and \overline{OB} . Therefore, $a = \theta_4$, $b = \theta_2 - a = \theta_1$, $c = \theta_3$ and this should satisfy $\theta_2 + \theta_4 > \theta_1 + \theta_3$. If crease 4 folds flat first, then creases 2 and 3 form edges \overline{OA} and \overline{OB} . Therefore $a = \theta_3 - \theta_4$, $b = \theta_1$, $c = \theta_2$, and thus it should satisfy $\theta_2 + \theta_4 < \theta_1 + \theta_3$. Conversely, judging from the given set of angles, the following tests can be used:

$$\text{If: } \theta_2 + \theta_4 > \theta_1 + \theta_3 \text{ then } \overline{O2} \text{ folds flat first.} \quad (23)$$

$$\text{If: } \theta_2 + \theta_4 < \theta_1 + \theta_3 \text{ then } \overline{O4} \text{ folds flat first.} \quad (24)$$

$$\text{If: } \theta_2 + \theta_4 = \theta_1 + \theta_3 \text{ the degree-4 vertex is flat-foldable (limit case).} \quad (25)$$

Thus, if we want to use the crease that blocks first as controller crease into a Grasshopper definition, we can insert one of these very easy inequations into the “Expression” component of grasshopper, setting as input the sector angles, and with a “Key/Value search” component we can substitute to the “False” or “True” result the index of the crease $\overline{O2}$ or $\overline{O4}$ as shown in Figure 28. This index will afterwards be used to select the crease that will be the first axis of rotation and we will set the domain of the rotation angle of its adjacent faces from 0° to 180° .

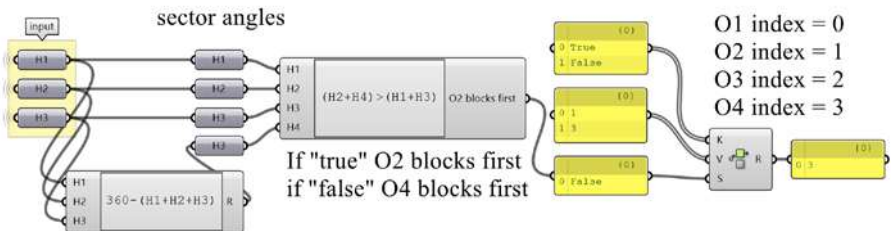


Figure 28 - Identification of the index of the crease that blocks first.

3.6.5. Other Fold Angles at Blocked State – With the Spherical Law of Cosine

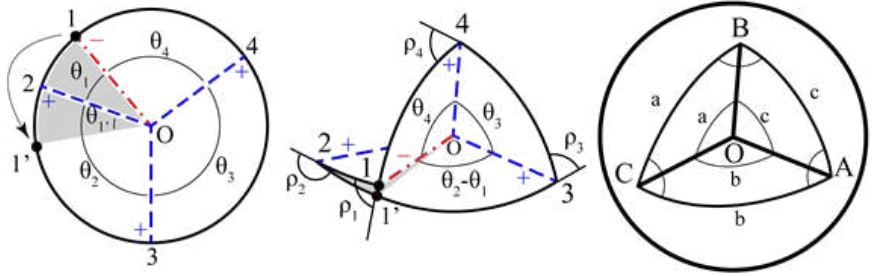


Figure 29 - Spherical trigonometry standard notation compared with the degree-4 vertex notation when O2 blocks first.

Once the crease that blocks the movement is identified, its fold angle is equal to 180° at the blocked state. The next question is how to calculate the fold angles of the other creases at the blocked state. Refer to Figure 29. Assume that the blocking crease is $\overline{O2}$. (For cases where the blocking crease is $\overline{O4}$, the notation is a mirror reflection). Then, by the cosine rule of spherical trigonometry,

$$\cos A = \frac{\cos a - \cos b \cos c}{\sin b \sin c} \quad (26)$$

Applying this to the fold angles in the degree-4 vertex we will get:

$$\rho_1 = -\cos^{-1} \frac{\cos \theta_3 - \cos(\theta_2 - \theta_1) \cos \theta_4}{\sin(\theta_2 - \theta_1) \sin \theta_4} \quad (27)$$

$$\rho_2 = 180^\circ \quad (28)$$

$$\rho_3 = 180^\circ - \cos^{-1} \frac{\cos \theta_4 - \cos \theta_3 \cos(\theta_2 - \theta_1)}{\sin \theta_3 \sin(\theta_2 - \theta_1)} \quad (29)$$

$$\rho_4 = 180^\circ - \cos^{-1} \frac{\cos(\theta_2 - \theta_1) - \cos \theta_4 \cos \theta_3}{\sin \theta_4 \sin \theta_3} \quad (30)$$

These expressions to calculate the fold angles are valid only at the blocked state. We can use this approach to design patterns of developable foldable three-dimensional structures with specific fold angles as shown in section 6.2.2.

3.7. About the Relation Between Origami Functionality and Real Applications

Some of these definitions highlight possible connections with the functionality of the origami when applied to real projects. For example, designing considering developability may be a way to reduce the cut-outs and scraps from the production, or it may be a way to produce assembling-free objects reducing the production costs and eventually the assembling time and issues. Furthermore, the rigid-foldability is strictly related to the fabrication with rigid panels and stiff materials, also the non-flat-foldability is related to patterns that block at a certain non-planar configuration thus realizing something with folded rigid panels will open new possibilities in the creation of mechanisms that self-block at a certain three-dimensional configuration or that can use blocking folds to increase the structural stiffness and stability. The study of the DOF could be useful to design deployable systems, for example designing a one-DOF mechanism helps preventing unexpected behaviour, it would also prevent the mechanisms to jam because of bifurcations in its folding and unfolding, it would decrease the amount of the necessary actuators and motors to move the mechanism. Lastly, designing considering flat-foldability may be an efficient way to design objects that have to be stored in a small space for transportation or stoking. All these relations are clearly reflected in the projects that we collected and analysed in chapter 2.

CONSTRUCTIVE METHODS FOR SOLVING THE KINEMATICS OF ORIGAMI

4.1. Families of Folded Surfaces























Single linear crease	 Single linear crease between rectangular faces	 Single linear crease between asymmetric rectangular faces	 Single linear crease between trapezoidal faces	 Single linear crease between triangular faces	
Multiple non-intersecting linear creases	 Straight accordion	 Accordion with converging creases	 Triangulated accordion with alternated diagonals	 Triangulated accordion with parallel diagonal	
Degree-4 single vertex	 Single symmetrical reverse fold	 Single simmetrical reverse fold on triangular faces	 Single asymmetrical reverse fold	 Generic degree-4 vertex	
Chain of degree-4 vertices	 Flat-foldable reverse fold on a straight accordion	 Reverse fold on a triangulated accordion	 Non-flat-foldable reverse fold on a straight accordion	 Multiple reverse folds on a single straight fold	 Discretized curve fold
Multiple degree-4 vertices	 Miura-ori	 Sink fold on a single degree-4 vertex	 Multiple discretized curve folds		
Multiple degree>4 vertices	 waterbomb base pattern/ magic ball pattern		 yoshimura pattern		

Figure 30 - Families of folded surfaces, the list obviously does not represent all the existing crease pattern, it represent only the set of families and subfamilies that we studied in this section.

In the book *Architettura delle superfici piegate* Casale *et al.* assert that the infinite variety of configurations that can be obtained folding a planar surface can be divided into three groups: “Chaotic”, “Shape-oriented”, “Structured”.

In the “Chaotic” family, the surface is crossed by a dense mesh of irregular creases. Once collapsed, the surface can be configured in various three-dimensional irrational shapes, this behaviour is hard to control and to foresee. Furthermore, in architecture, there is no reason to try to analyse it from a kinematic point of view.

If we take a sheet of paper and we crumple it strongly with our hands, we can force it to assume infinite different configurations, opening it and stretching it properly. The more is dense the starting crease pattern, the more the surface will be capable to adapt to specific configurations. Apart from the obvious difficulty in determining the relationship between the initial subdivision and the final shape, it is equally difficult and meaningless searching for the geometrical relationship between the parts participating in this kind of spatial configuration. (Casale *et al.*, 2013)

In the “Shape-oriented” family, the plane is divided into a multitude of different polygons, placed side by side to generate a pattern that must assume a specific spatial configuration. Figurative traditional origami and packaging can be categorized in this group. Some of the projects, among the ones that we selected in section 2, which could be added to this category are, for example, the “Origami pavilion” by Tal Friedman, the “Common Ground” by Zaha Hadid Architects (Bhooshan, 2016), the “Folding Table” by Tachi or the “Curved Folding Metal Twins” by Chandra *et al.* (Bhooshan, 2015; Chandra *et al.*, 2015a).

The “Structured” family is defined by groups of equal tiles. The surface can be configured into a lot of different shapes, using a little number of different tiles, this characteristic makes this family of folded surfaces easier to animate and to design. The well-known solar panel by Koryo Miura (Miura, 1985), the self-deployable cardiac stent by Kuribayashi *et al.* (Kuribayashi *et al.*, 2006), or the “Resonant Chamber” by RVTR (Thün *et al.*, 2012), are related to this family of surfaces. In general, the tessellations and the corrugations from traditional origami, are perfect examples of this family of folded surfaces. We focus only on the “Shape-oriented” and “structured” families. The structured folded surfaces are easy to design because they are characterized by groups of equal faces, also the variety of shapes that they can assume is limited, thus the algorithms will be more focused on the animation of the surfaces instead of being focused on the final folded shape. The “Structured” family will be explored in this chapter. The “Shape-oriented” family will be addressed in chapter 5 and 6.

To study the “Structured” family, we divided it into six sub-classes of patterns which will be presented following a growing complexity criterion. The sub-classes synthesized in Figure 30 are patterns with: a single linear crease, multiple non-intersecting linear creases, a single degree-4 internal vertex, a chain of degree-4 vertices, multiple internal degree-4 vertices and multiple degree>4 vertices.

4.2. Operative Tools

All the crease patterns proposed will be created and animated with a constructive synthetic approach applied with a parametric three-dimensional modeller, Grasshopper (Rutten, n.d.), a node based parametric modeller integrated into Rhinoceros 6 by McNeel associates (McNeel, n.d.). Because of that, for clarity sake, sometimes we will refer to specific components of Grasshopper or Rhino. Nevertheless, we will try to put the reader in the condition to reproduce the same constructions also with different parametric\mathematical modellers explaining the processes and describing the algorithms step by step.

4.3. Analogy with Computer Programming and Terminology Clarification

4.3.1. Clustering and Nesting

In computer programming, a widespread practice that helps to keep the code simple and clear is to divide the code into clusters and save them for future uses. This type of approach is very common in Object-Oriented-Programming (OOP), where these clusters of code are called objects and classes. This approach is useful when scripting because it allows the programmer to keep the code short and clear and more importantly to find the errors faster because there is no need to test the previously tested clusters of code. Furthermore, it helps to keep the script easily readable, transmissible and editable.

To make digital three-dimensional parametric animated folded surfaces, it can be followed a similar approach, starting from elementary cases grouped in families, up to harder ones by referring to already solved simpler problems adding variables and components to them. The algorithms will be presented with a growing complexity criterion, starting from patterns with one or more single non-intersecting linear creases, followed by patterns with one or more internal degree-4 vertices and patterns with vertices with a degree greater than 4. For

this reason, the first cases may appear trivial, but they are the necessary building blocks for the following algorithms of higher complexity.

The operation of grouping some nodes inside a single new component is called “nesting”, we will refer to these nested nodes as “clusters” as they are called in Grasshopper. Furthermore, we will refer to a small part of a folded surface as a “molecule”, because we consider it as a group of “atoms” (creases and faces) that can be joined to make bigger patterns or “macro-molecules”. The word “molecule” is usually used to define a part of a pattern composed by many different folds. Robert J. Lang in the book *Origami design secrets* provides a comprehensive explanation of a design method based on the stitching of different molecules (Lang, 2011).

4.3.2. Definition of “Algorithm”

We consider the “Algorithm” as a process or set of rules that we need to follow to reach an expected result. Another interesting definition of “Algorithm” is the one given by Kostas Terzidis in his book *Algorithmic architecture*:

An algorithm is a computational procedure for addressing a problem in a finite number of steps. It involves deduction induction, abstraction, generalization, and structured logic, it is the systematic extraction of logical principles and the development of a generic solution plan. Algorithmic strategies utilize the search for repetitive patterns, universal principles, interchangeable modules, and inductive links [...] An algorithm may be compared to the steps in a recipe; the steps of gathering the ingredients, preparing them, combining them, cooking, and serving are algorithmic steps in the preparation of food [...] Theoretically, an algorithm is the abstraction of a process and serves as a sequential pattern that leads towards the accomplishment of a desired task. (Terzidis, 2006)

In computational modelling, the word “Algorithm” is often used to refer to the parametric generative procedures.

4.4. Single Linear Crease

We start the study of the animation of the folded surfaces with the most elementary crease pattern, which is composed of one single linear crease. This kind of fold has been called by Casale *et al.* “First fold” (Casale *et al.*, 2013). A rectangular piece of paper can be folded in half along its middle line gen-

erating two rectangular equal faces, or it can be folded along any other line, dividing the piece of paper into trapezoidal, parallelogrammatical, or triangular faces. Even if the fold line is only one, there are many possible ways to animate the folding of a single creased piece of paper. For example, we can simply rotate the faces around the crease, or we can intersect the paths of the vertices or the edges of the faces and use the intersections as geometric references. Furthermore, we can anchor different parts of the piece of paper to the construction plane, for example, we can make the crease lifting from the construction plane while constraining the opposite edges to slide on the plane. Or we can anchor a face, the crease, or an edge to the construction plane rotating the unconstrained elements out of plane. We can even use a reference curve or a curved surface as a rail on which the edges would slide. We will explore many different methods and we will apply them on patterns with a single crease line before approaching more complex cases.

4.4.1. Single Linear Crease Between Equal Rectangular Faces, Two Edges Slide on Construction Plane – Intersecting Circles

This method uses intersecting circles as geometric references to find all the possible configurations. The circles represent the paths of the vertices opposed to the crease when the adjacent faces rotate around it. This approach works only if the starting rectangular surface is folded in half generating equal rectangular faces. We perform a symmetrical motion of the two faces making the opposite edges sliding on the construction plane of the same amount while lifting the crease from the plane. The following method refers to Figure 31.

Draw two rectangular surfaces, then draw the crease AB, and the edges BC and BF, we will use these geometries as inputs of the Grasshopper's definition. Now, reparametrize the domain of the curves BC and BF between

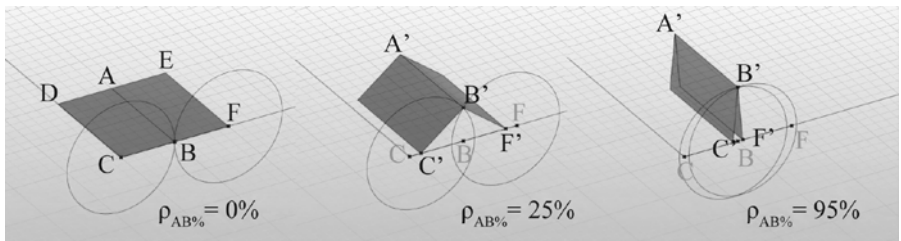


Figure 31 - Single linear crease between rectangular faces with intersecting circles. The input slider controls the distance from C to B and from F to B.

0 and 1. Match the parameter 1 of each curve with the point B and the parameters 0 with points C and F. On an input slider (that we call “Folding percentage”) set a value between 0.00 and 1.00 and extract a point on both BC and BF curve at that parameter and call those points C’ and F’. Draw two circles on C’ and F’. The circle radii are equal to the length of the edges BC and BF (in this case they are equal one to each other as well because the starting rectangles are equal). The circles are drawn on a plane perpendicular to the crease AB. If we move the cursor of the slider, we will see the circles moving along the edges BC and BF. At this point, intersect the circles getting two points, and select the upper one or the lower one (in grasshopper we can use the “Boolean toggle” component or the “Value List” component to make the selection). This last operation allows us to switch between mountain and valley folding. We call B’ the chosen intersection of the circles. Lastly, draw a polyline passing through C’, B’, and F’ and extrude it along AB. In this way, we construct the rectangular faces and we can see them folding and unfolding by moving the cursor of the folding-percentage slider. The 0.00 and the 1.00 values (we could also use integer values, but adding decimals makes the animation smoother.) on the slider represent the unfolded state (0%) and the completely folded state (100%). We can notice that with this method the edges DC and EF slide on the construction plane and the crease AB leaves the plane moving along the Z axis. The full algorithm with the used nodes is shown in Figure 32.

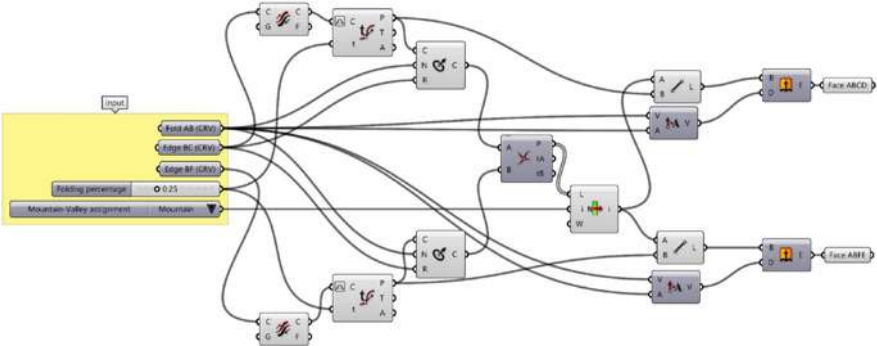


Figure 32 - Generative algorithm for the single linear crease between equal rectangular faces with intersecting circles.

4.4.2. Single Linear Crease Between Asymmetric Rectangular Faces, Two Edges Slide on Construction Plane – Intersecting Circles

In the previous example, we solved the kinematics of a rectangular surface creased exactly in half. But what if we move the crease a bit toward one side or the other? We can use intersecting circles to identify the position of the crease as we did in the last example. However, if we want to constrain the opposite edges (CD and EF) to slide on the construction plane, we cannot consider a vertical symmetry. So, the algorithm works as follows.

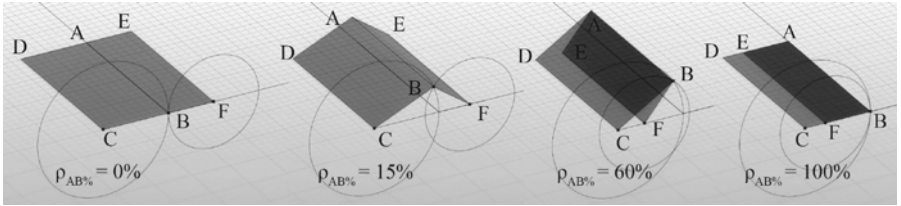


Figure 33 - Single linear crease between asymmetric rectangular faces animated with intersecting circles, making the edges EF sliding on the construction plane.

First, draw the edges CB and BF and draw the crease AB. Set a slider between 0.00 and 1.00 and remap the output value to a domain between 0 and the length of the segment BF. Multiply this value by two and use the result to move the point F along the segment CF. Like so the point F will never be farther than its original distance from point B. Now, draw two circles on C and the moving point F with radii respectively equal to the initial edges CB and FB. Move the point F together with its relative circle and intersect the two circles. Select one intersection point and draw a polyline passing through the points CBF as we did in the previous algorithm. Extrude the polyline to generate the folded surface¹ (Foschi, 2019).

4.4.3. Single Linear Crease Between Rectangular Faces, Crease on Construction Plane – Varying Fold Angle

In this section, we show a different approach. Now we use the fold angle as a variable parameter instead of moving and intersecting two circles on the construction plane. This method is conceptually easier to understand but it gives a different result. Compared to the method with intersecting circles, the method with fold angle returns a fold animation of the surface where the crease stays on the construction plane and the edges leave the plane along an arch of circumference centred in B. Furthermore, we use an angular input instead of a percentage value.

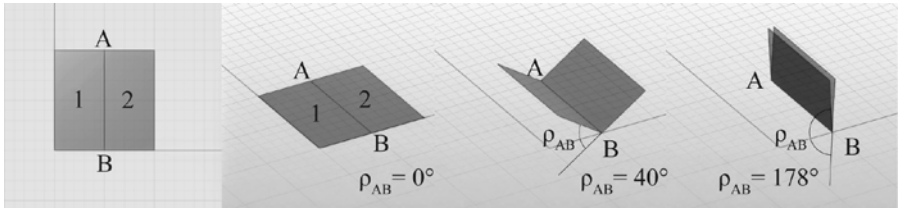


Figure 34 - Single linear crease between rectangular faces with the fold angle as a variable input.

The algorithm is developed as follows. Refer to Figure 34. Draw two rectangular faces and the crease AB. After that, rotate both faces around the crease AB, face 1 clockwise, and face 2 counterclockwise. To match the input angle to the exact angle between the normal vectors of the two faces, it is necessary to divide it by two before plugging it into the corresponding “Rotate axis” components. This step is necessary because we plugged the same angle into both the “Rotate axis” nodes to have vertical symmetry. On the contrary if we would have rotated only one face keeping the other on the plane, we would not have needed to halve the angle.

To animate the folding process of the surface it is sufficient to move the angle slider which has as boundaries 0° and +180°. We could also set the domain boundaries from -180° to +180° so that the crease assignment would flip from valley to mountain once passed the flat state. Nevertheless, if we choose bigger absolute values to limit the domain, the surface would self-intersect after passed 180°. Figure 35 shows the complete generative algorithm.

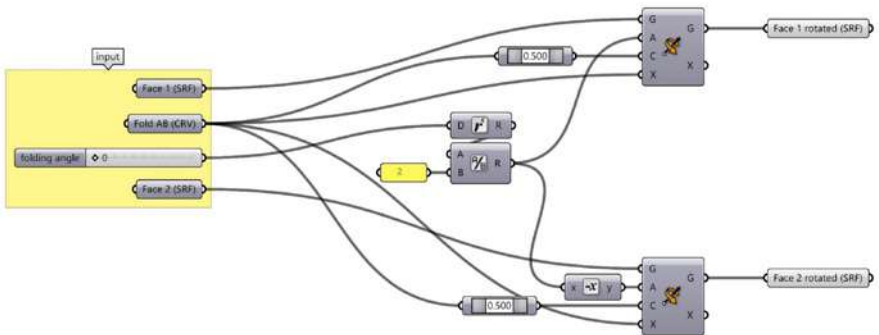


Figure 35 - Generative algorithm for single linear crease between equal rectangular faces with fold angle input.

4.4.4. Single Linear Crease Between Triangular Faces, Crease on Construction Plane – Varying Fold Angle

The single linear crease between triangular faces (Figure 36) is conceptually no different from the single linear crease between rectangular faces (Figure 34) explained in section 4.4.3, because we rotate the two faces around the fold line on the construction plane as we did in the previous case. However, we decided to present also this case because we used a little different method to generate faces. Furthermore, this algorithm will be part of a more advanced algorithm later. In particular, we used the “Extrude Point” node instead of the “Extrude” node. The “Extrude Point” node creates a triangular face starting from a segment and a point which is a faster solution to re-create the faces in this case. As in the case of the single linear crease between rectangular faces, we considered vertical symmetry, and we divided the input angle by two to make the slider value matching exactly the angle between the normal vectors of the two faces. In Figure 37 it is shown the full generative algorithm.

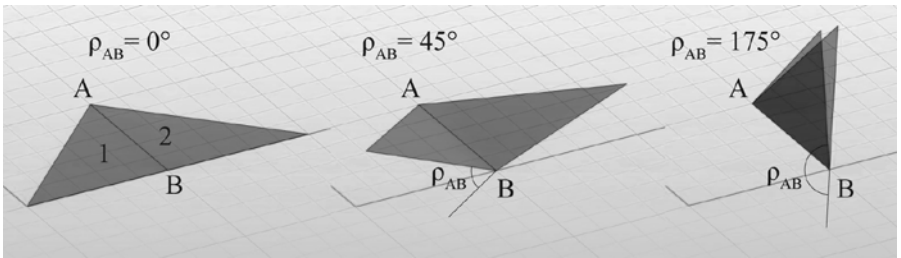


Figure 36 - Single linear crease between triangular faces animated by varying the fold angle.

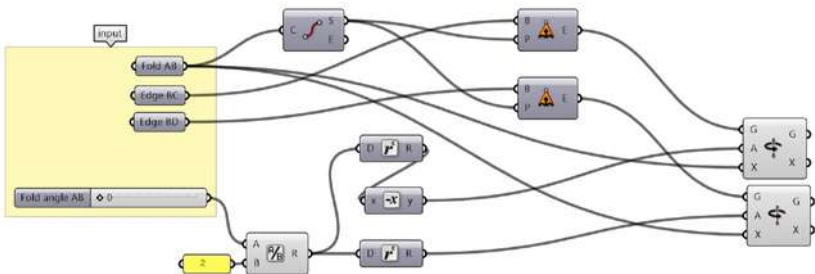


Figure 37 - Generative algorithm for the single linear crease on triangular faces animated with fold angle AB.

4.4.5. Single Linear Crease Between Triangular Faces, Two Edges Slide on Construction Plane – Intersecting Circles

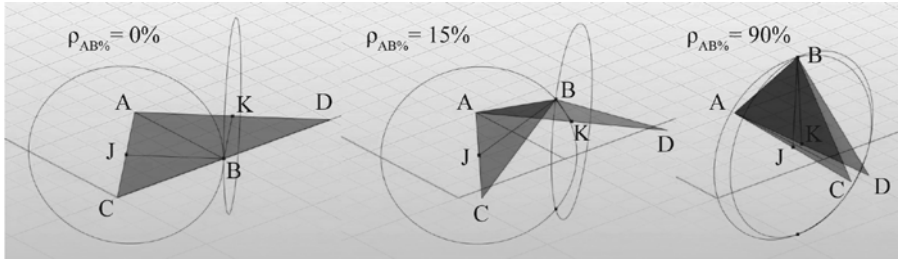


Figure 38 - Single linear crease between triangular faces with intersecting circles, the outer edges slide on the construction plane.

In this case, we use again intersecting circles to find the only two possible positions of the crease while constraining the edges AC and AD to slide on the construction plane. However, this time the circles are drawn on different planes because the crease and the outer edges are not parallel anymore.

The algorithm works as follows. Draw the crease AB, the edges AC and AD. Measure the absolute value of the angle between AC and AD (we will use this value in a moment). Project the point B on the edges AC and AD finding the relative points J and K. Draw the two planes passing through the points J and K and perpendicular respectively to AC and AD. Draw one circle on each plane with centres respectively on J and K and radii equal to JB and KB. Set one input slider with a domain that goes from 0.00 to 1.00 and remap the output value to a domain that goes from 0.00 to the angle between AC and AD just measured. Rotate the edges AC and AD around the point A (on the construction plane) of half of the remapped value and rotate with them the relative circles centred on J and K. Find the intersection points between the two circles and chose one of them. That point is now the new position of B. Draw a polyline passing from the moved points CBD and extrude it toward point A. Like so we generated the two triangular faces that we can animate by moving the slider from 0.00 to 1.00. In most of the definitions that use intersecting circles, at the limit cases, the circles perfectly overlap (when the surface is 100% folded in this case), this causes the “Curve | Curve” node (which solves intersection events between two input curves) to return a “null” value. To solve this problem, we can both extend the generative algorithm with an “if” statement that triggers only when the problematic configuration occurs, or it can be simply limited the domain of the input slider to stop a bit earlier of the critical values (99,999% in this case). This last method is the one we used in the generative algorithm shown in Figure 39.

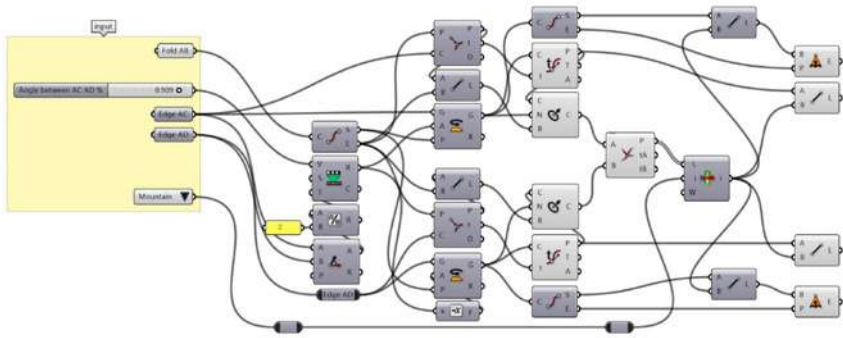


Figure 39 - Generative algorithm for the single linear crease between triangular faces with intersecting circles, the outer edges slide on the construction plane.

4.4.6. Single Linear Crease Between Trapezoidal Faces, Two Edges Slide on the Construction Plane – Intersecting Circles

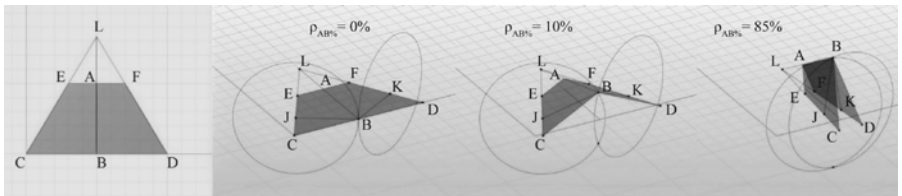


Figure 40 - Single linear crease between trapezoidal faces, built and animated with intersecting circles.

The method to build and animate two trapezoidal faces with outer edges and crease converging to a point, while constraining the edges EC and FD to slide on the construction plane, is analogous to the triangular faces case shown in section 4.4.5. The trapezoidal faces can be asymmetric, but the edges EC and FD and the crease AB must converge to the same point L so that we can use the same approach used for the triangular faces, because the trapezoid faces would be simply a slice of two triangular faces.

In general, if we have two trapezoid faces that do not have the outer edges and the crease converging to the same point, we would not be able to force the edges EC and FD to slide on the same plane (while rigid-folding the surface) because the edges would belong to the same plane only at completely unfolded (and folded) configuration.

On the contrary, if we do not want to keep the outer edges CE and DF on the same plane, we can apply the method that uses the fold angle as the only variable input, as shown in section 4.4.3 and 4.4.4. In that way, any combination of trapezoidal faces would work.

The algorithm that we used to animate the surface shown in Figure 40 has the same structure of the algorithm presented in section 4.4.5 with some additional nodes that cut the triangular faces into two trapezoidal faces. Refer to Figure 41 for the full algorithm.

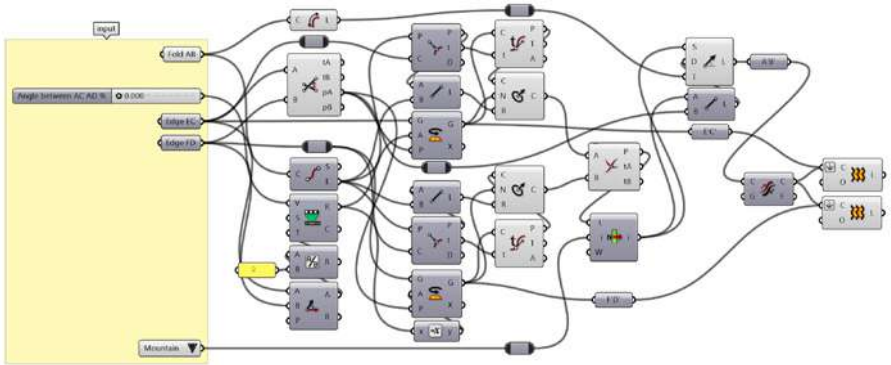


Figure 41 - Generative algorithm for the folding animation of two trapezoidal faces with outer edges and crease converging into the same point. The outer edges are constrained to slide on the construction plane.

4.5. Patterns with Multiple (Non-Intersecting) Linear Creases

If we fold a piece of paper with two non-intersecting creases, we obtain a crease pattern with two DOF. Adding another non-intersecting crease, the DOF will increase to 3, and so on. Thus, an ideal parametric generative algorithm should have an input controller parameter for every DOF. This approach would guarantee the maximum shaping freedom, and the surface could be conformed to every possible configuration. When the creases are a small number this is feasible, but when the pattern has a high number of creases, even if having a controller for each fold would be the most versatile solution, working with it would rapidly become cumbersome. However, we can forcefully limit the number of controllers to make it more manageable. For example, we can force all the creases to fold of the same amount at the same time, by using as inputs the same fold angle for every crease. Alternatively, if we want a more versatile behaviour, we can make the surface slid-

ing along a linear or a curved rail. We can also set a mathematical rule to control automatically the propagation of the fold angle non-linearly.

What is important to keep in mind is that using a smaller number of input parameters compared to the number of DOF would reduce the available configurations into which the pattern could be shaped. Furthermore, if we want to fabricate a pattern with more than one DOF where several creases must move at the same time of a precise amount, we must add an actuator for each one of those creases, even if in our algorithm we used the same input angle. In this section we propose some examples starting from the straight accordion, up to the triangulated accordion, animated with specific fold angles applied to groups of creases, or constraining the creases to slide on a curved and linear rail.

4.5.1. Straight Accordion – Array of “Single Linear Crease Between Rectangular Faces” Molecules

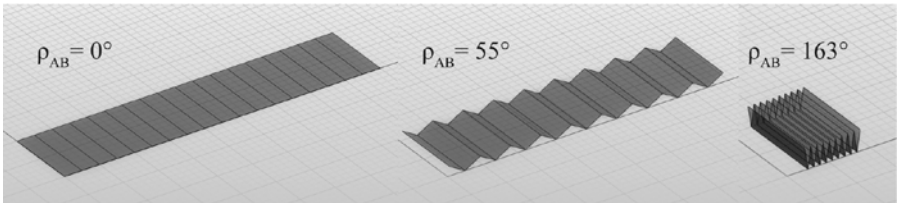


Figure 42 - Straight accordion joining rectangular molecules animated with fold angles.

In section 4.4.1 we have shown how to make a base generative algorithm to animate a surface folded into two equal rectangular faces. Now, that we have the base molecule, it is easy to make a sequence of parallel linear creases. To do that, we join multiple base molecules linearly. The new generative algorithm starts with the clustered algorithm shown in Figure 35. Then extract the bounding box of the base molecule finding its dimensions along X, Y, and Z axes. After that, use the X dimension as the input of the “Linear array” node which uses it to space the consecutive copies of an amount equal to the animated base molecule width. In this way, the distance is related to the fold angle, which reaches its maximum at the completely unfolded state and its minimum at its completely folded state. The full generative algorithm is shown in Figure 43. This algorithm returns an animated straight accordion where all the fold angles are equal and fold at the same time with constant propagation. However, as we said this does not cover all the possible configurations of a straight accordion, because its DOF is higher than the number of the controller inputs used. Thus, to increase the animation freedom in the next section we will show how to make the accordion sliding on a linear and a curved rail.

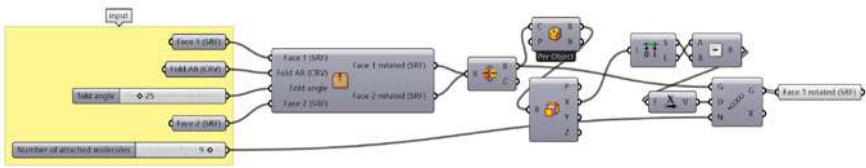


Figure 43 - Generative algorithm for the straight accordion joining 9 molecules with a single linear crease. Inside the initial cluster node, there is the generative algorithm shown in Figure 35.

4.5.2. Straight Accordion Sliding on a Rail – Intersecting Circles

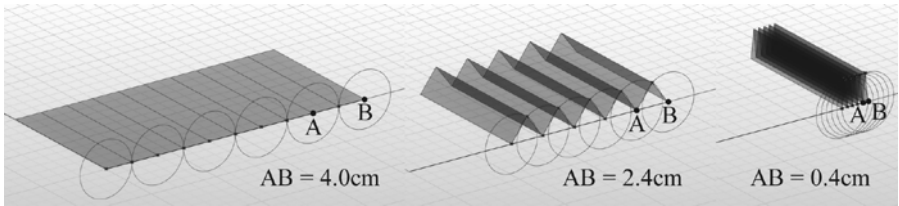


Figure 44 - Straight accordion, uniform motion on a linear rail.

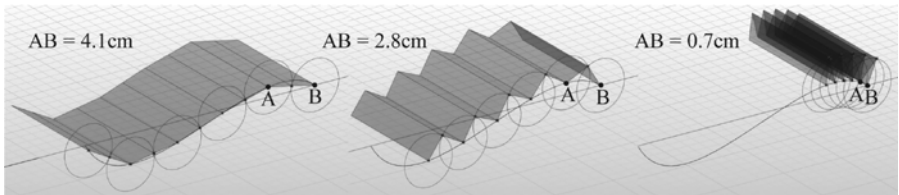


Figure 45 - Straight accordion, folding and unfolding along a curved path.

In Figure 44 it is synthesized the construction of a straight accordion made with intersecting circles. This approach gives a similar result to the one shown in Figure 42, but it is much more versatile, because the molecules are not equal copies, thus they can be conformed not only to a linear segment but also to a curved path as shown in Figure 45.

The generative process is similar to the one explained in section 4.4.1, but it starts with a curved rail. This curve (it can also be a straight line) is divided in “n” parts (5 in this case), which represent the number of mountain folds (or valley). Between adjacent points, there must be the same linear distance. To achieve this result, we used Grasshopper’s “DivDist” node. On each point we draw a circle, the construction plane of the circle is perpendicular to the creases, and

their radii are equal to half the initial distance between two consecutive points on the curve. Then, all the points are moved uniformly along the curve to make all the circles intersecting. The intersection points are filtered and used to build and animate the surface with the same approach used in section 4.4.1. The full definition is shown in Figure 46 and Figure 47.

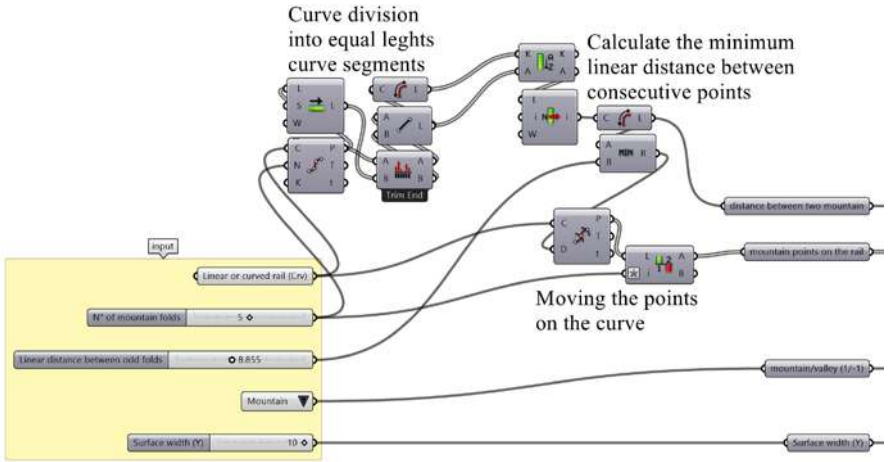


Figure 46 - Generative algorithm to animate a straight accordion on a linear or curved rail – Part 1.

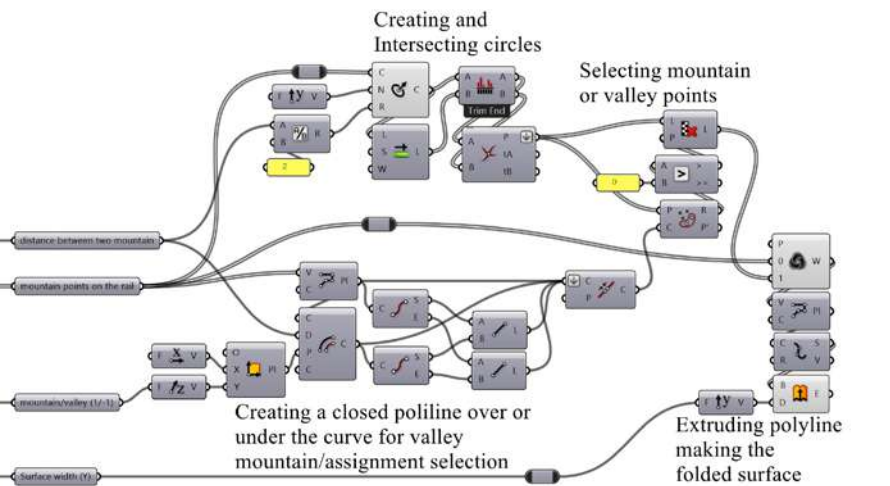


Figure 47 - Generative algorithm to animate a straight accordion on a linear or curved rail – Part 2.

4.5.3. Straight Accordion on a Rail with Non-Uniform Fold Angle Distribution – Intersecting Circles and “Graph Mapper”

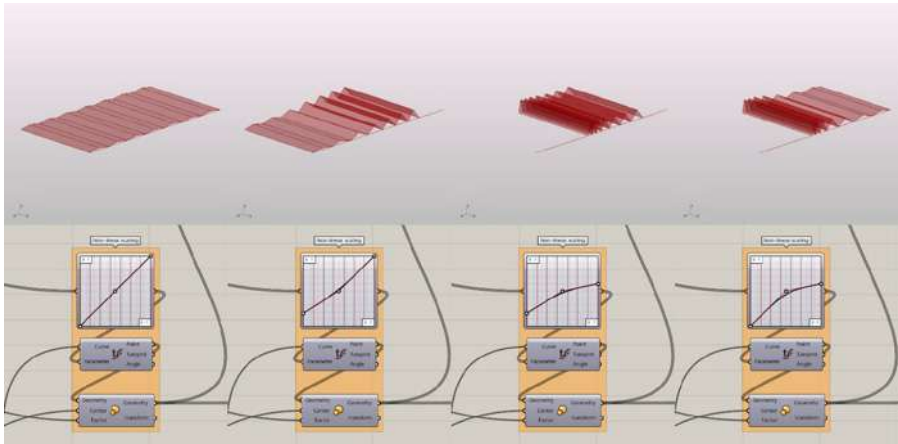


Figure 48 - Straight accordion animation with non-uniform fold angle distribution on a linear rail.

To increase even more the shaping freedom without increase too much the number of controller inputs we can remap non-uniformly the fold angles on each fold without necessarily controlling them one by one. We can do that with the “Graph Mapper” component, this component remaps the values of a list according to a chosen mathematical function. In this example, we used a Bezier function. We set as the input of the graph mapper the parameters of each point along the curve so that we can remap and use them as new parameters that define the new position of each point along the curve. When the Bezier is a linear function all the points stay at the original position, when we move the Bezier control points the surface starts folding non-uniformly because the points on the curve are no more uniformly spaced. In this way, we can partially fold some section of the surface leaving completely unfolded other sections as shown in Figure 48. It must be pointed out that with this approach to get correct behaviours, we must know which shape of the remapping function are usable and which are not, with some particular shape of the Bezier curve, the points, when remapped, may be too far one from the other and the surface would lose its developability or behave in an unpredictable way. To avoid this kind of inconveniences it is suggested to use a Bezier curve and to move one control point at a time checking in real time if the surface does not stretch or self-intersect as shown in Figure 48² (Foschi, 2019).

4.5.4. Accordion on Two Circular Rails – Intersecting Circles

The accordion on two circular rails has a behaviour similar to the straight accordion on a linear or a curved rail, but the unfolded pattern has the creases converging into a point rather than being parallel to each other, thus the faces are trapezoids instead of rectangles. The approach used here starts from multiple adjacent “Single linear crease between trapezoidal faces” molecules constructed following the same approach used in section 4.4.6. An accordion with parallel creases can be conformed only into a cylindrical surface. On the contrary, the accordion with all the creases converging into a point can be conformed into a cone. An example of the conical motion is shown in Figure 49.

The animated surface is obtained from a generative algorithm that takes as inputs two concentric circles used as rails, and an integer numeric slider which defines the number of folds that are distributed at equal distances along the two circles. The circles can be coplanar, or they can belong to two different parallel planes. The circles are divided into a given number of parts, the points so obtained are remapped to make them slide along the circular paths, similarly to what done in section 4.5.2. The domain of each circle is reparametrized between 0.00 and 1.00, then the parameter of the curve where each point is located is extracted. Afterwards, all the parameters are remapped to a new domain smaller than 1, and they are redistributed again on the circles. Moving continuously the top boundary of the domain between 1.00 and 0.00 will make the point slide on the circle. The point located on the start of the circular path will remain fixed. The other points, when remapped, will tend to compress getting closer to the initial point sliding along the reference curve. On each odd point, we draw a circle that at the completely unfolded state is tangent to the adjacent circles drawn centred on the adjacent odd points. When the points slide on the rails, they get closer and every circle intersect the adjacent ones in two points. Only one of two intersecting points is chosen for each pair of adjacent circles (the upper or the bottom one). Choosing one or the other intersection will flip the mountain /valley assignment of the surface and will make it fold inside or outside the reference cone defined by the two starting circular rails. Once selected all the intersections all the points are connected to make a zig-zag line that alternates the intersection points and the odd points on the circles. Lastly, the two zig-zag lines are plugged into a “Loft” component which generates a surface between each segment of the upper zig-zag line with the correspondent aligned segment of the zig-zag line constructed on the bottom circle. To animate the surface, it is sufficient to move the top boundary of the remapped domain with a slider with a range that goes from 0.00 to 1.00. When the top boundary of the

remapped domain is 0.5 the surface is exactly halfway to be completely folded, so the slider that controls that number can be considered as a percentage of the folding motion³ (Foschi, 2019).

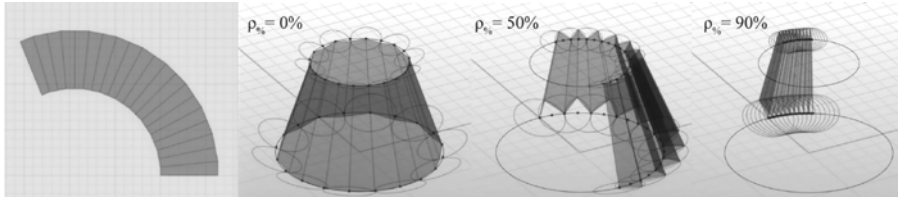


Figure 49 - Radial accordion sliding on two non-coplanar circular rails.

4.5.5. Triangulated Accordion – Joining Multiple “Single Linear Crease Between Triangular Faces” Molecules

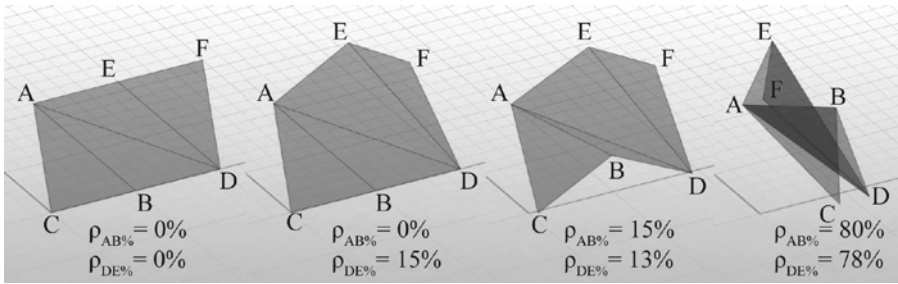


Figure 50 - Two equal molecules joined that can fold independently making a macro-molecule composed of 4 faces.

A triangulated accordion is an accordion where all the faces are triangular, and the mountain valley assignment is alternated. Thus, the liner folds are not parallel anymore and they touch on the perimeter of the paper. To model and animate it we can join many “Single linear crease between triangular faces” molecules (explained in section 4.4.5). We can both use as input a single slider that controls the fold angles of all the molecules at once, or we can use one input slider for each newly attached molecule. The former approach returns a less versatile result in terms of shaping freedom. In this section, we propose a method where two different sliders are used to control separately the folding of all the odd and even triangular molecules. In this way, the formers can be folded independently from the latter molecules and this approach allows to configure the folded surface into a fan, as shown in Figure 53. Figure 51 shows the full

generative algorithm to join 2 molecules of 2 triangular faces. The clustered nodes contain the algorithm proposed in section 4.4.5. The two molecules can be folded independently by moving the sliders that control the folding amount of each one of them. The algorithm is developed as follows (refer to Figure 50 for notation). We refer to the vertices of the first molecule as A_1, B_1, C_1, D_1 , and the vertices of the copied molecule as A_2, B_2, C_2, D_2 . Copy the first molecule and move it along the A_1D_1 edge matching the vertex A_2 of the copied molecule with the vertex D_1 of the first molecule, then rotate the copied molecule around the vertex D_1 of the first molecule to match the vertex D_2 of the second molecule with the vertex A_1 of the first molecule. Now the two molecules are joined, and they can be controlled with different input sliders. Now the vertices D_1 and A_2 become D , the vertices A_1 and D_2 become A , the vertices C_1 and B_1 become C and B , and the vertices C_2 and B_2 become F and E .

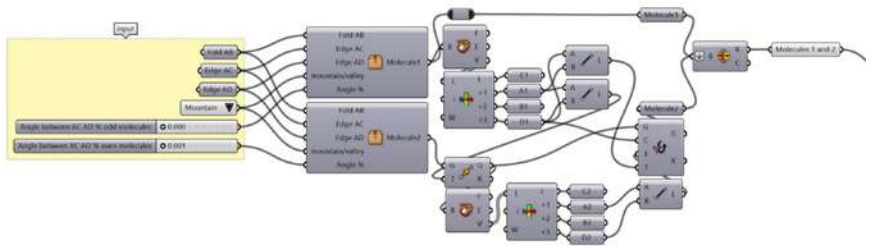


Figure 51 - Joining 2 equal molecules controlled by 2 different fold angle (%) sliders.

The new macro-molecule (with vertices ABCDEF) is made by 2 smaller molecules of two faces each. We can now mate as many macromolecules as we want to make a chain of them where the odd ones fold together independently from the even ones as shown in Figure 53. Grasshopper, unfortunately, does not allow looping or recursive definitions, so each molecule must be copied manually as many times as needed. A more elegant alternative can be achieved using a plugin called Anemone (Zwierzycki, n.d.). Anemone implements Grasshopper with 2 new nodes that perform loops. The “Loop Start” that needs as input one or more variables (in our case the animated macro-molecule of 4 faces) and the “Loop End” that records the data from each cycle and resend the resultant data to the “Loop Start” component. The number of cycles can be set attaching an integer number as an input into the “Loop Start” node. In-between the two nodes there must be connected the definition that needs to be repeated. In our case the copy and rotation of the macro-molecule (A_{i+1} and C_{i+1} must be

matched with D_i and F_i). The loop definition (that follows the definition shown in Figure 51) is shown in Figure 52. It must be pointed out that in the solution that uses Anemone, the animation has a flickering problem. Every time that the cursor of the slider that animates the surface moves, the surface disappears for an instant because the loop definition needs to update. For a flicker-free animation, the copy and paste method (with no loops) is suggested.

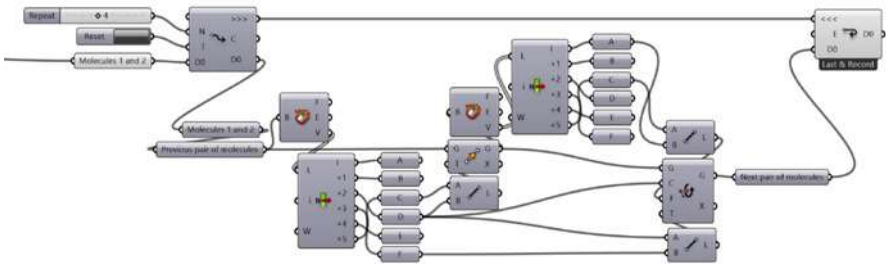


Figure 52 - Looping generative algorithm that repeats the cycle 4 times. The definition that is cycled copies and rotates the macro-molecule 4 times starting every time from the newly created macro-molecule. The two upper nodes, with two arrows in their icons, are from Anemone plug-in for Grasshopper.

The cases in Figure 53 and Figure 54 are particular cases where all the faces are equal, but with a different the orientation of the diagonals, thus the kinematics of the surface changes. They both can be folded starting from straight accordions creased along the diagonal of each face. A faster and easier approach to triangulate and animate a straight accordion is by reflecting a straight accordion with respect to a plane that cuts all the faces along their diagonals as shown in Figure 55. However, this method is limited in terms of possible configurations compared to the one explained above, because in this method we use only one single slider to control all the folds at once.

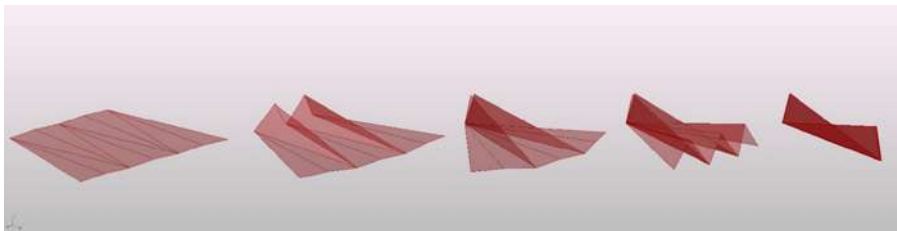


Figure 53 - Triangulated accordion, alternated diagonals, non-uniform motion.

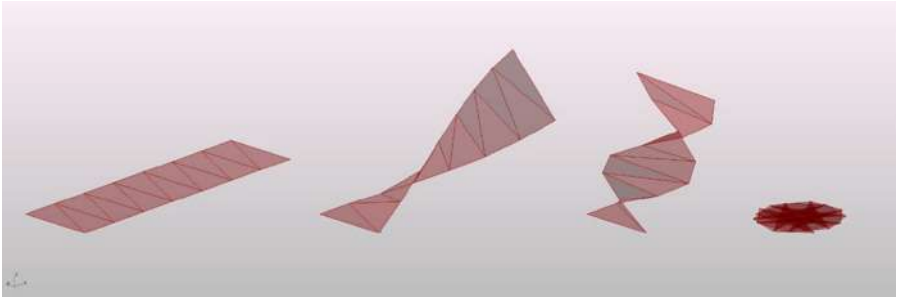


Figure 54 - Triangulated accordion, parallel diagonals, uniform motion.

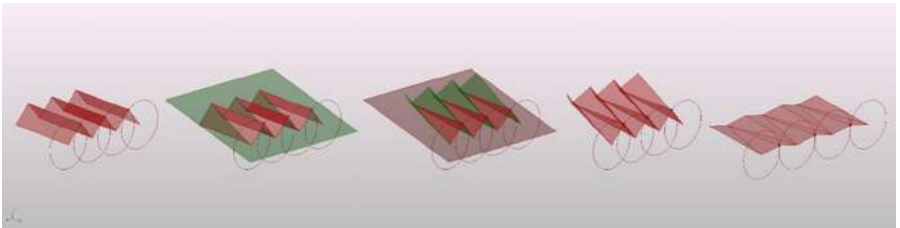


Figure 55 - Triangulated accordion obtained by reflecting the straight accordion, uniform motion.

4.6. Single Degree-4 Vertex⁴

Now that we explored the animation of patterns with multiple linear non-intersecting creases, according to the increasing complexity criterion, we investigate patterns with a single internal vertex. The simplest rigid-foldable pattern with only one internal vertex is the degree-4 vertex as demonstrated by Abel *et al.* in the paper *Rigid Origami Vertices: Conditions and Forcing Sets* (Abel *et al.*, 2016). They also proved that any degree-4 vertex must have one and only one crease with the opposite verse with respect of the other three to be rigidly foldable. Also, they asserted that “A single-vertex crease pattern (C, μ) can be continuously parameterized in a family of rigid origami folds if and only if (C, μ) contains a bird’s foot.” This means that in a degree-4 vertex there must be a “Y” shaped family of three folds to be rigid-foldable.

As the reader can see, the problem increased rapidly in complexity. The patterns with more non-intersecting creases studied previously, only involved animation problems related to the increasing DOF and controller inputs. On the contrary, now we cannot focus only on the animation of the pattern, but also

with the correct pattern design. Furthermore, in patterns with internal vertices, the fold angle of each crease may be influenced by the adjacent creases. In fact, the degree-4 single vertex as we saw in section 3.5.2 is a one-DOF mechanism (Tachi, 2011a), thus, to animate it, we do not need to deal with an increasing number of controller inputs.

4.6.1. Symmetric Reverse Fold – Reflecting a Single Linear Crease

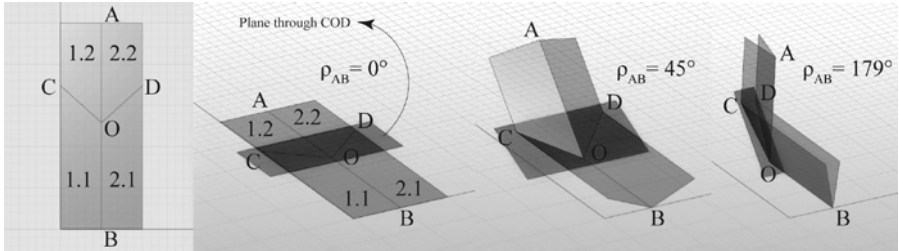


Figure 56 - Symmetric reverse fold on a single linear fold molecule by reflection with respect of the plane passing through COD points.

The easiest way to make and animate a degree-4 vertex is by reflecting a “single linear fold” molecule. This kind of approach, based on reflection, has been extensively used also for curve folded designs by Mitani (Mitani & Igarashi, 2011). This type of fold in traditional origami is called “Reverse fold”. The “Reverse fold” can be performed reflecting any completely flat-folded or partially folded surface with respects of a reflection plane. We used the same technique to make the triangulated accordion in section 4.5.5. The triangulated accordion is a limit cases of a reverse fold. Casale *et al.* call it “Second fold” and they define it as a valley or mountain zig-zag crease running in a transversal direction relative to a set of linear folds which they call “First folds” (Casale *et al.*, 2013). This kind of terminology (first and second folds) focuses the attention on the order of the steps of the folding process, on the contrary, the traditional terminology (reverse fold) focuses the attention on the nature of the fold itself. When the first creases intersect the second creases, the verse of the first creases flips changing from valley to mountain or vice-versa. The reverse fold can generate flat-foldable or non-flat-foldable patterns depending on the angle of the reflection plane and the configuration of the set of first creases. Performing a reverse fold to a pattern with “n” non-intersecting creases will change its DOF from “n” to one. For example, if we add to a straight accordion with five linear creases, a transversal reverse zig-zag crease intersecting all the linear creases, the DOF changes from 5 to 1.

The construction method for the asymmetric reverse fold is analogous to the method explained in the previous section (4.6.1). However, if we do not limit the domain of the controller fold angle properly, the surface will self-intersect after passed the blocking configuration as shown in Figure 58. Thus, if we want to know which angles make the vertex self-intersecting, we can apply the following method.

First, explode the input 4 faces 1.1, 1.2, 2.1, 2.2 (not animated), and count how many collisions there are between the unfolded faces (with the “Collisions Many|Many” component). Then, perform the reverse fold as explained 4.6.1. Partially fold the vertex and test the collisions again. Compare the collisions at flat state with the collisions at folded state. If the collisions increased the algorithm will return “True”, if they remain unvaried it will return “False”. When we change the fold angle value ρ_{AB} if it is “True”, then we passed the blocking point, or we are exactly on the blocking point.

Now if we want to synthetically and automatically extract the maximum value that we can assign to ρ_{AB} without self-intersecting the vertex, we can apply the following method. Move slowly the cursor of the slider ρ_{AB} from 0% to 100%, and record frame by frame (with a “Data recorder” component), in two separate parallel lists, the values of ρ_{AB} and the relative Boolean values resultant from the collision test. Now compare the two lists and erase all the angular values that correspond to “True” (with a “Cull pattern” component). Sort the angles from the smallest to the largest and choose the largest one. In this way, we just found an approximation of the blocking angle. The accuracy of the maximum angle value is proportional to the number of frames between the flat and the folded state. The full definition is shown in Figure 59.

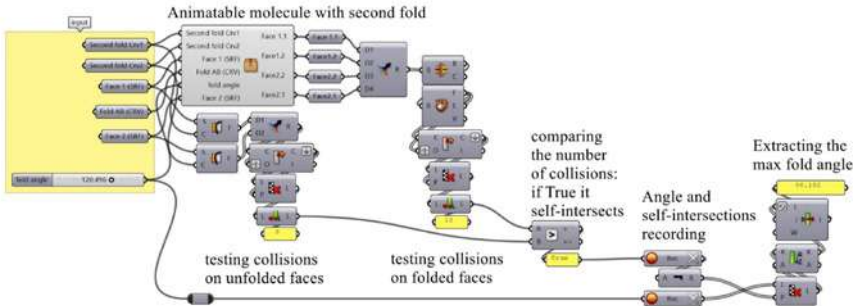


Figure 59 - The generative algorithm for the identification of the maximum fold angle, testing the animated surface for collisions; the maximum fold angle is generated within a given tolerance, thus it is an approximation; the cluster contains the definition shown in Figure 57.

We can also use the maximum fold angle found to limit the domain of the ρ_{OB} value by inputting the calculated maximum ρ_{OB} and the fold angle slider into the same “Minimum” component as shown in Figure 60. In this way, even if we set a fold angle greater than the maximum ρ_{OB} , the output value will stop at the maximum ρ_{OB} .

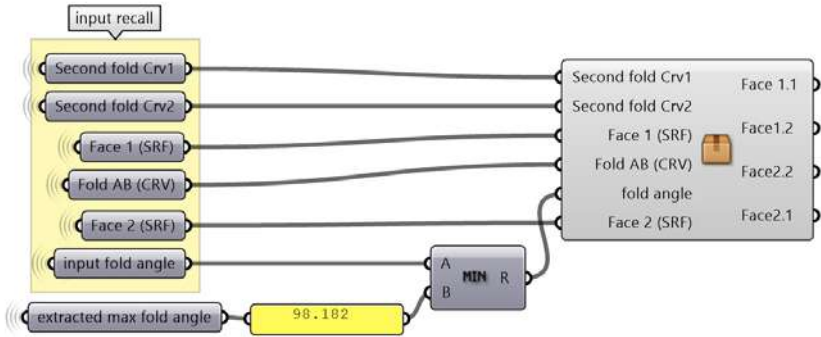


Figure 60 - Re-animating the second fold molecule avoiding self-intersections, the extracted max fold angle node is the result of the algorithm in Figure 59.

4.6.3. Generic Degree-4 Vertex – Intersecting Cones

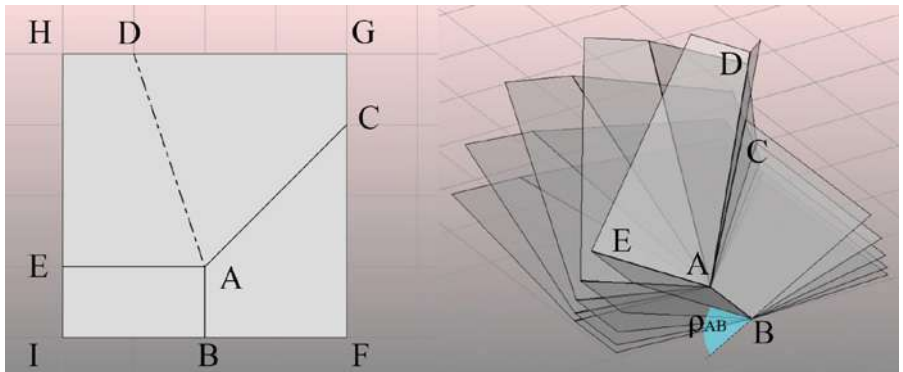


Figure 61 - Animation of a degree-4 single vertex CP.

The reflection method shown in the previous sections is an easy approach, but it does not solve all the possible degree-4 vertices. It can be used only to animate vertices where the opposite mountain and valley creases are colinear. The vertex shown in Figure 61, for example, cannot be solved with the reflection method.

In this section, we propose a different approach to animate any rigid-foldable degree-4 vertex.

The algorithm is visualized in Figure 62 and Figure 63 and it works as follows. Draw 4 adjacent faces ABFC, ACGD, ADHE and AEIB as shown in Figure 61, the faces must be drawn considering the rigid-foldability conditions of a degree-4 vertex presented in section 3.4. Revolve the crease AD using as axis the fold AE generating a cone with apex in A. Similarly revolve the AD fold around the axis AC. Rotate the faces ABFC and ACGD around the crease AB clockwise, and the faces AEIB and ADHE around the same axis of the same amount but counterclockwise. Doing so the vertex D will split in D' and D'' and the two cones intersect generating 2 segments, meeting into the apex of the cones. These two segments represent the possible positions of the AD fold so that rotating the faces ACGD and ADHE to that point their AD edge would perfectly match. Choose the intersection line AK' which matches the mountain valley assignment of the given pattern, choosing the other intersection line will generate the other folding mode of the same pattern. Rotate AD''HE around AE to mate D'' with K'. Do the same thing with the face ACGD' rotating it around the AC axis mating D' with K'. The algorithm is finished.

Now, if we change the value of θ we will be able to animate the surface. If the pattern is non-flat-foldable, and if the controller fold AB is not the fold that hit 180° first, the model will self-intersect during motion, we can solve it by testing the collisions as shown in the previous section (4.6.2).⁵

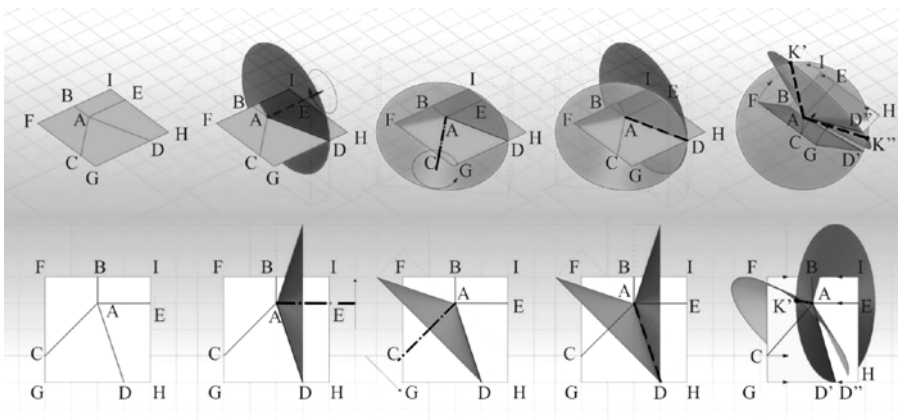


Figure 62 - Three-dimensional modelling and animation of a generic degree-4 single vertex with intersecting cones – Part 1.

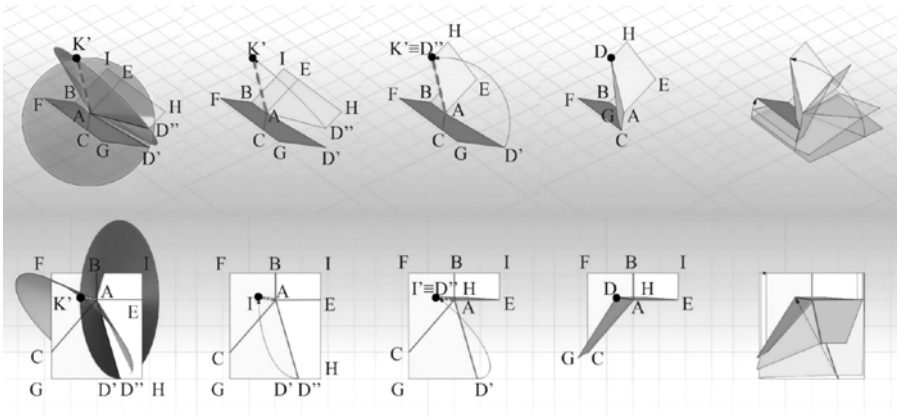


Figure 63 - Three-dimensional modelling and animation of a generic degree-4 single vertex with intersecting cones – Part 2.

4.7. Multiple Degree-4 Vertices

In section 3.3 we introduced some concepts about the DOF, and in section 3.5 and 3.6 we explained the kinematics of degree-4 vertices. As we said, patterns with multiple degree-4 vertices may not be rigid-foldable if some symmetry conditions do not occur. However, it is not always easy to recognize those patterns. Furthermore, testing the compatibility of every fold angle in every closed loop of faces may be a very time consuming and cumbersome operation. Thus, in this section, we will start solving the kinematics of easy symmetric well-known patterns with multiple degree-4 vertices that are known to be rigid-foldable.

4.7.1. Joining “Symmetric Reverse Fold” Molecules – Critical Observations About Global Rigid-Flat-Foldability

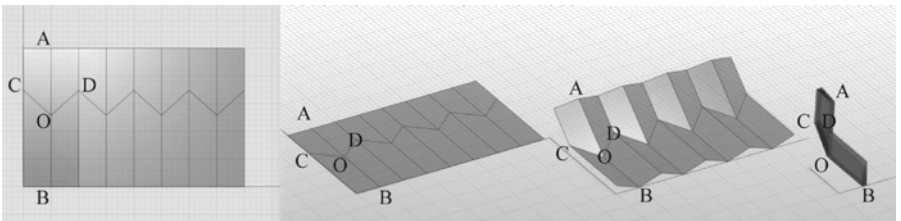


Figure 64 - Reverse fold on straight accordion, joining “symmetric reverse fold” molecules.

A straight accordion without reverse folds has more than one DOF. If we add one reverse fold to the same accordion, as shown in Figure 64, its DOF will decrease to one. Furthermore, if every segment of the reverse fold is symmetric with respect to the adjacent linear first folds, the pattern is also flat-foldable.

To animate a flat-foldable straight accordion with a single reverse fold, we can reflect a straight accordion with respects of a reflection plane following the same approach explained in section 4.6.1, or we can perform a linear array of a “symmetric reverse fold” molecule following the same approach that we proposed in section 4.5.1 (the definition using this approach is shown in Figure 66).

If we want to use the former approach (the “reflection of a straight accordion” method), we need to pay attention to the fact that the straight accordion must be animated using the same fold angle for every crease, so that we can reflect the first two consecutive animated faces of the accordion as shown in 4.6.1 and using the same animated plane to reflect all the other faces.

If we want to apply the second method (the linear array of “symmetric reverse fold” molecules), we can follow two main approaches: we can join the molecules side-by-side or along their linear first folds. If they are mated side by side we will have a pattern with one long single zig-zag shaped reverse fold as shown in Figure 64, if we mate them along their linear first fold we will have a surface with more V-shaped reverse folds and one single first fold, as shown in Figure 65. A pattern with more than one reverse fold can be shaped in different ways depending on the orientation of the molecules and the angle of the reverse folds.

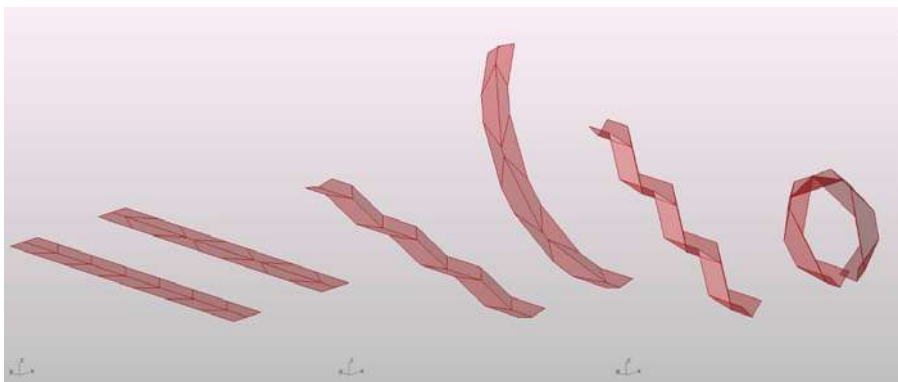


Figure 65 - Different behaviour during the collapsing, depending on the direction of the second folds.

If we join more than one “symmetric reverse fold” molecule along their linear first crease with alternated orientation, we would risk of ending up having self-collisions between the ending molecules at the opposite sides of the pattern, and it would be impossible to complete the folding motion without self-intersecting or flexing the faces, as shown in Figure 67. This means that it is not sufficient to attach rigid-flat-foldable molecules to have a guaranteed globally rigid-foldable pattern. Furthermore, if we attach side by side two molecules with symmetric second folds but different angles, we obtain a globally non-fat foldable pattern because the new internal vertex (resulting from the attaching of the two molecules) does not respect the Kawasaki’s condition, as shown in Figure 68. The problem of global rigid-flat-foldability avoiding stretches and self-intersections, for a generic pattern, is proven to be at least NP-hard (Akitaya *et al.*, 2016), thus simulating the folding motion, for now, is one of the most reliable methods to test if a pattern is rigid-flat-foldable or not.

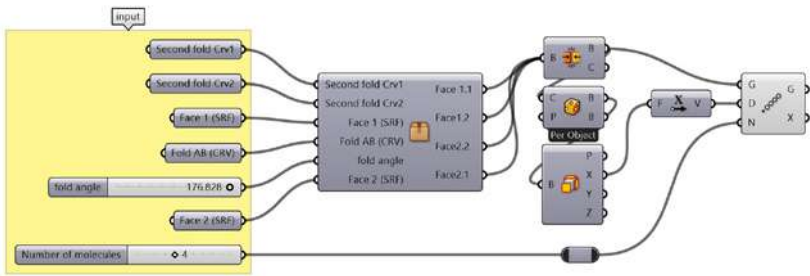


Figure 66 - Generative algorithm for the symmetric reverse fold on a straight accordion, joining “symmetric reverse fold” molecules, the cluster on the right contains the generative algorithm explained in section 4.6.1.

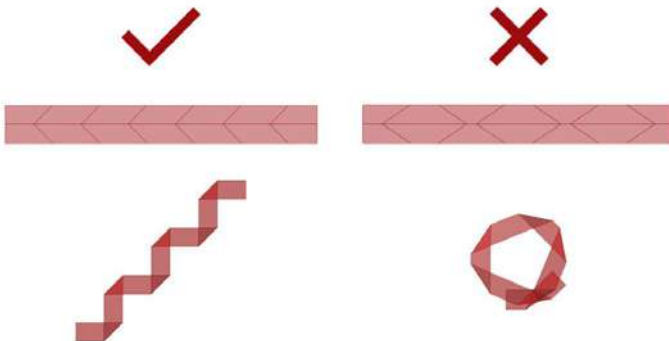


Figure 67 - Possible and impossible rigid flat-foldability due to self-intersections, even if the Kawasaki theorem is verified in every internal vertex.

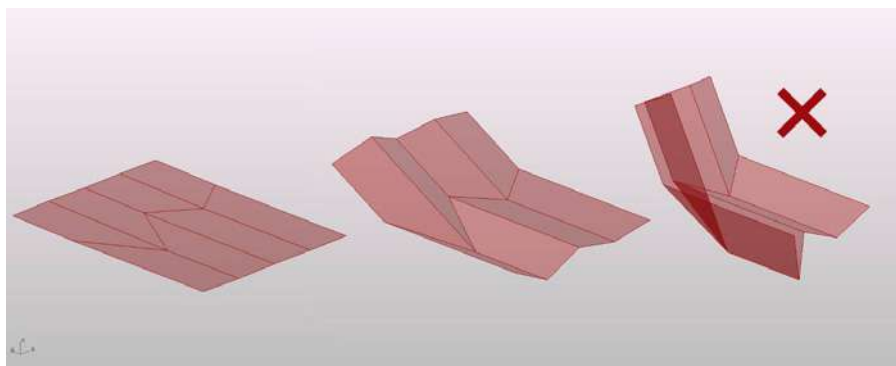


Figure 68 - Stitching two different symmetric flat foldable molecules side by side generates a globally non-flat-foldable pattern.

4.7.2. Joining “Asymmetric Reverse Fold” Molecules

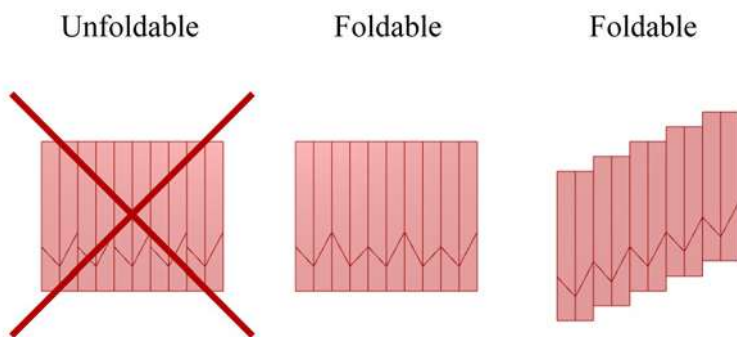


Figure 69 - Joining asymmetric molecules, the first case on the left is unfoldable, the second and third cases are non-flat-foldable.

In section 4.7.1 we placed side by side a series of “Symmetric reverse fold” molecules making a reverse-folded flat-foldable accordion. With the “Asymmetric reverse fold”, the result will be non-flat-foldable, but the procedure is similar. However, the molecules must be mated matching the endpoints of the reverse fold instead of aligning their outer perimeter. Alternatively, to match the endpoints of the reverse fold and also the outer perimeter of adjacent molecules, we can mirror the even or odd molecules as shown in Figure 69. Alternatively, instead of mating equal molecules side-by-side we can reflect a straight accordion with an angled plane as we do in the following algorithm.

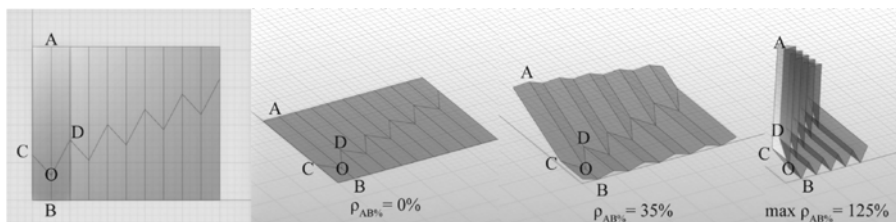


Figure 70 - Animation of an asymmetric reverse fold on a straight accordion, with the reflection of a straight accordion with respects of a mirror plane constructed on CO and DO.

Refer to Figure 70. First, animate the straight accordion as explained in 4.5.1. Then, draw two asymmetric creases OC and OD on the first two faces. Reverse fold the first two faces of the accordion following the method explained in section 4.6.1 and 4.6.2. After that, slice the whole accordion with the same plane and reflect it consistently with the first two faces. The full generative algorithm is shown in Figure 71. To keep the structure of the nodes simple and clear we animated the surface with self-intersections. However, if there is the need to stop the animation at the blocked state, we can apply the collision detection method explained in 4.6.2.

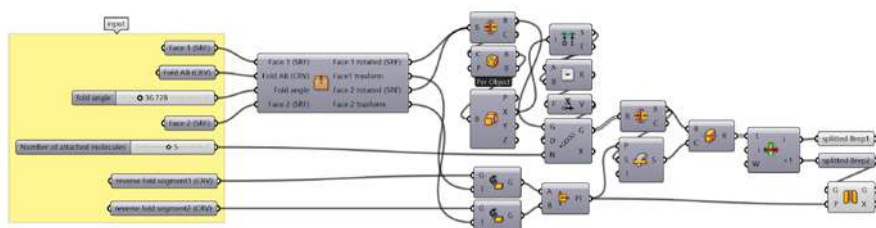


Figure 71 - Generative algorithm to animate a straight accordion with an asymmetric reverse fold the cluster on the right contains the algorithm explained in section 4.4.2.

4.7.3. Reverse Fold on Triangulated Accordion – Joining “Symmetric Reverse Fold” Molecules

The reverse fold on a triangulated accordion is comparable to the reverse fold on a straight accordion, but in the starting rectangular molecule, the first fold is drawn along the diagonal AD instead of along the symmetry vertical axis of the rectangle. The algorithm works as follows.

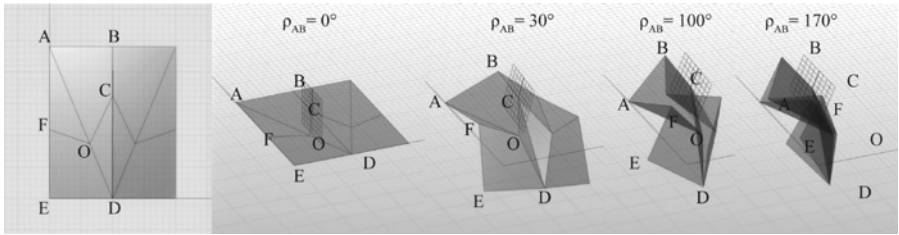


Figure 72 - Reverse fold on a triangulated accordion – construction and animation of the macro-molecule.

Refer to Figure 72. Animate the faces AED and ABD by rotating them around the crease-line AD symmetrically. Draw the creases FO and CO to be symmetrical with respects of the crease AD. Once drawn the two creases apply the reflection method to perform a reverse fold as shown in section 4.6.1. Now draw a plane passing through B, C and D and use it as a reflection plane to mirror the first molecule. We now obtained a symmetrical macro-molecule with 8 faces, which we are going to copy and paste one after another, as shown in Figure 73, using a looping definition. We use a looping definition because the macro-molecules need to be rotated other than translated to match perfectly the following molecule edges, thus we cannot use a simple translational array as we did in the previous section. The looping definition is shown in Figure 75 and it consists into matching the two planes constructed on the points A, F and E and the plane passing from their respective reflected points A', F' and E'.

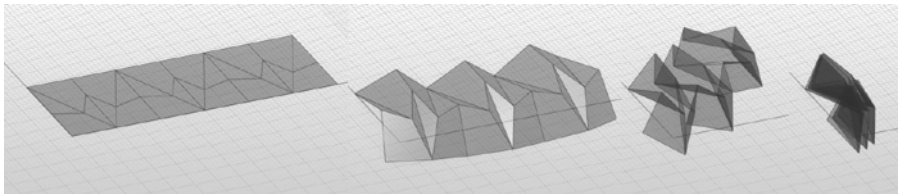


Figure 73 - Joined macro-molecules to make a longer triangulated accordion with the reverse fold.

Even in this case, it is not easy to foresee if the surface will self-intersect just by watching its CP. It is much easier to judge it by animating the surface. In fact, not every configuration of the reverse-fold gives rigid folding motions from start to finish. For example, in Figure 76 we can see two examples of flat-foldable reverse folded triangulated accordions with different behaviours. Both can be folded in-plane, but in the second case, the surface can fold flat and rigidly,

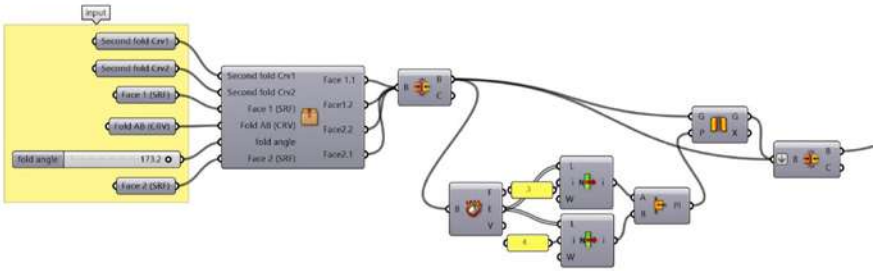


Figure 74 - Generative algorithm for a reverse-folded triangulated accordion – Part 1.

without colliding or self-intersecting. On the contrary in the first case, the surface is rigid-foldable only for a certain amount, and once reached a specific fold angle it starts self-intersecting. Thus, to be able to reach the flat configuration continuing the folding motion without self-intersecting the surface, it would be necessary to flex the faces or slide a bit the fold-lines (as it would happen into a real model made by paper). In Figure 77 we can see an example of a paper model that apparently can be folded rigidly, but the digital simulation proves the opposite. This misalignment between the two models is caused by the fact that the paper is flexible, and the digital model is not. Furthermore, in the paper model the creases try to self-correct shifting from the original position for a small amount so that the initial pattern would slightly change and in some cases, it would be barely perceptible.

Testing the rigid foldability is very important especially if we want to use the pattern for real applications made by rigid patterns, because using a pattern that is not perfectly rigid-foldable may cause serious problems over the short or the long term. For example, if the faces are very rigid and connected with hinges, but the used pattern is not perfectly rigid-foldable (even for a small amount as in Figure 77), given that we are able to fold it forcefully exploiting the elasticity

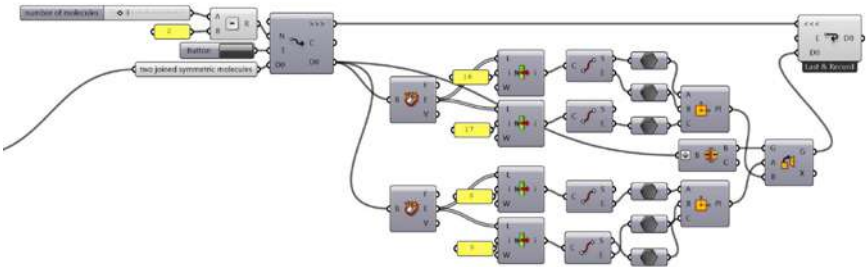


Figure 75 - Generative algorithm for a reverse-folded triangulated accordion – Part 2.

of the material, every time the mechanism is folded and unfolded, it would be subject to stresses and deformations caused by the colliding faces and that may cause faster deterioration of the joints.

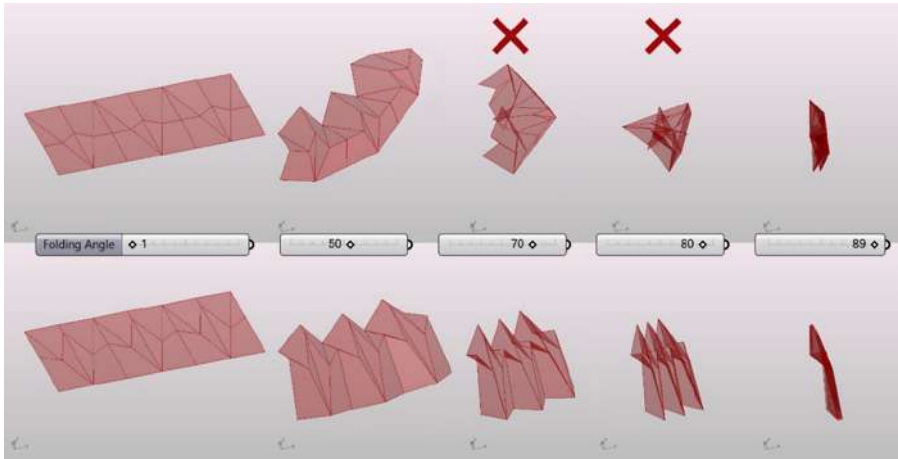


Figure 76 - Non-flat-rigid-foldability and rigid-flat-foldability of the triangulated accordion, varying the angle of the reverse fold.

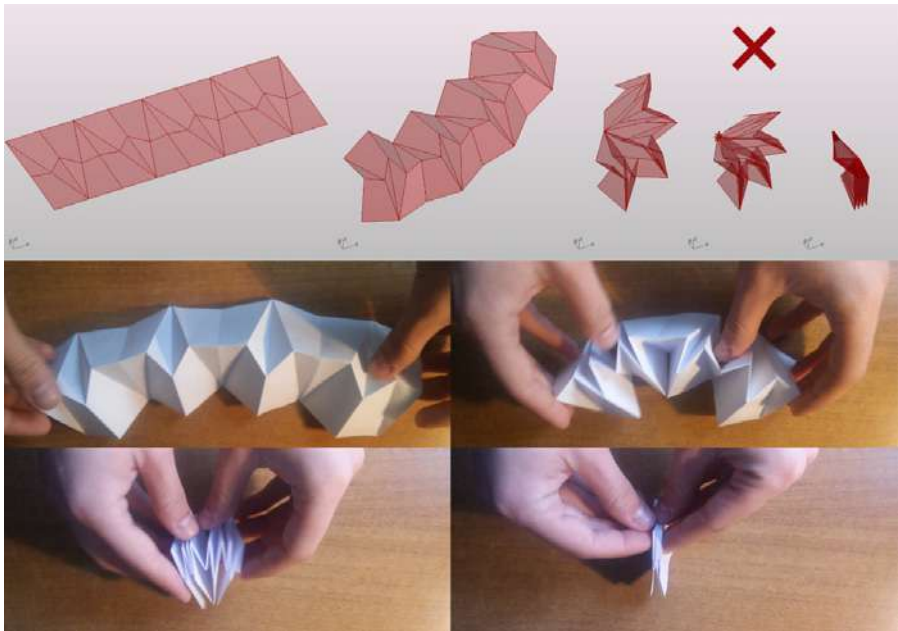


Figure 77 - Apparent incongruence between the digital and physical models; the faces of the former self-intersect, the faces of the latter remain apparently planar.

4.7.4. The Miura Pattern – Planar Rectangular Array of “Symmetric Reverse Fold” Molecules – Intersecting Circle with Plane of Symmetry

The Miura pattern, or Miura-ori, is probably the most famous and most used corrugations of all times, it has been studied and used for actual applications in several fields. Its diffusion is due to the fact that it is a one-DOF mechanism and it has a particular property of expanding and contracting with negative Poisson's ratio. The Japanese engineer Koryo Miura, in the mid-80s, studied this pattern, which derives from the uniform compression buckling pattern of a thin plate (Miura, 1997), to create a solar panel deploying system. This system was revolutionary because the solar panel so folded was able to unfold univocally with minimal fold actuators and without the help of the human interaction (Miura, 1985), for this reason, it has been renamed with his name. However, according to what recently stated by Hellmuth Stachel, the Miura-ori might already have been known before: “However, already before K. Miura, this technique was known; according to a personal communication with Gy. Darvas, it was kept as a military secret in Russia in the thirties of the last century. There are many applications of Miura-ori.” (Stachel, 2015).

The Miura-ori pattern is composed of equal parallelogram faces, this guarantees that the surface has an in-plane expansion and contraction. Recent studies developed methods to modify the traditional pattern but preserving the rigid foldability, in order to give to the surface different spatial configuration (Tachi, 2010b; Tachi, 2009).

The traditional Miura pattern can be divided into elemental equal molecules composed by four adjacent faces and it can be digitally modelled and animated by reflecting many times a straight accordion, or by joining many equal molecules of four equal parallelogrammatical faces. In this section, we are going to follow the second approach.

We can use the method explained in section 4.6.1 to animate the molecules, but the faces must be all equal and with a parallelogrammatical perimeter before applying a rectangular array. However, we highlighted that this method generates problems in the flat-folded configuration because when the two segments of the reverse fold converge into one (at flat-folded state) the reflection plane would be no more univocally identified. Thus, the algorithm at that limit case could behave in an unexpected way. In the same section we solved this problem limiting the domain of the controller fold angle, however, in this section we want to propose an alternative method to solve the problem at the limit configurations without limiting the domain or using cumbersome “if statements”.

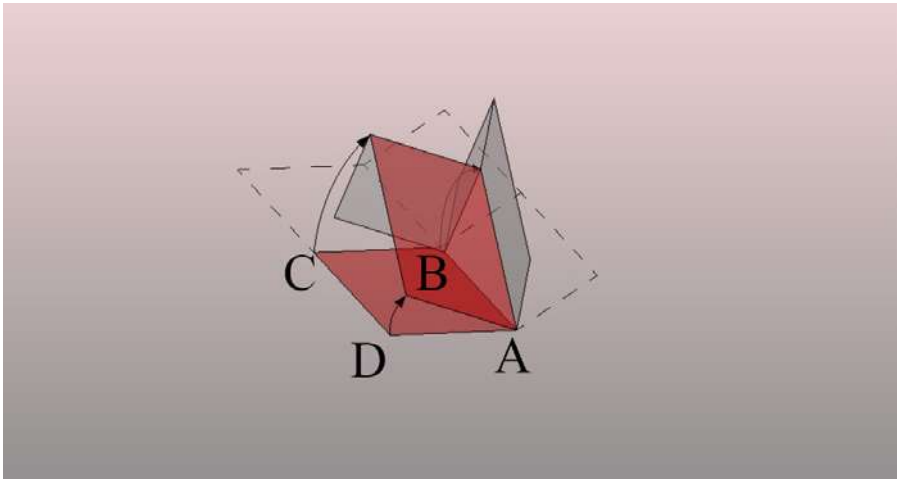


Figure 78 - Single Miura-ori face used to model the symmetrical molecule of the Miura-ori.

In Figure 78 it is represented one single Miura molecule. We can further simplify the problem focusing only on one single face (ABCD), because the faces are all equal, thus any other face is a simple translation or reflection of the same reference face. To be able to obtain an XY in-plane expansion of the Miura-ori during folding we want the segment AD to stay on the XY plane and the segment AB on the YZ plane.

The algorithm for the construction of the parametric animatable Miura-ori works as shown in Figure 79. Define a construction plane XY. Define point A in the origin of the plane, copy and move that point along X and Y to define other 3 points: B, C and D to form a rectangle. Move the points C and D along Y to shear the rectangle to get a rhomboid. Connect three sliders to these nodes to be able to control the width the length, and the amount of shear of the rectangle. Rotate clockwise ABCD around the Z-axis passing from A. Set the angular domain between the segment AD and the Y-axis. The rotation angle will be remapped in 0-100 domain to be controlled with a slider that we call “Collapsing (%)”. Draw a circle on the plane perpendicular to AD passing from B. Intersect the circle with the plane YZ obtaining 2 points, (I’ above the XY plane, I” under the XY plane). Select the upper point. Rotate the polygon ABCD around the AD axis to match B to I’. Mirror ABCD along YZ plane and mirror the two polygons together along XY plane. Then, translate the last two polygons along AB. The Miura molecule is finished. To make a wider Miura-ori pattern just copy and move this molecule as many times as you need along the direction X

and Y. To animate the collapsing of the Miura-ori just move the Collapsing% slider, and to change the initial Miura pattern proportions, move the sliders connected to the “Width (cm)”, “Length (cm)” and “Shear (cm)” inputs. With this method, the circle has always one or at most two intersection points with the plane, so that the limit cases are univocally defined, and the algorithm never returns unexpected behaviours. This approach works only for modular symmetric flat-foldable Miura-ori⁶ (Foschi, 2019).

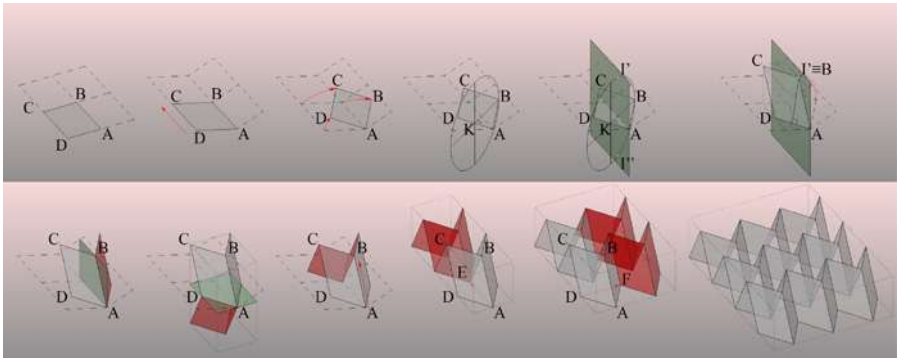


Figure 79 - Miura-ori constructions and animation.

4.7.5. The Sink Fold – Reflecting the Tip of a Degree-4 Vertex

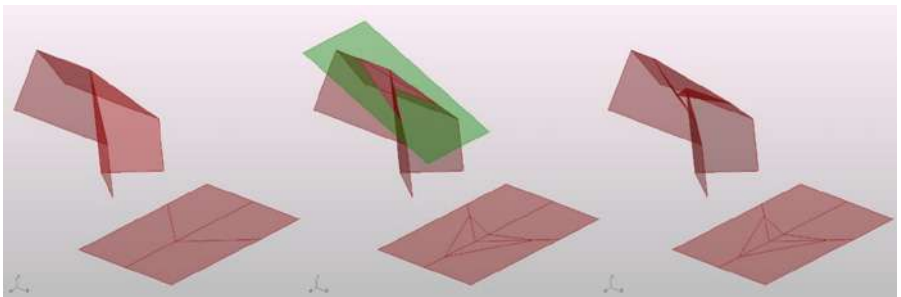


Figure 80 - Sink fold obtained by reflection.

Another known pattern with multiple degree-4 vertices is the sink fold. It consists into a closed loop of creases around a single vertex. Consecutive edges of the loop are symmetrical with respects of the crease that they touch. In particular we are going to sink fold a degree-4 vertex. The sink fold is a traditional fold that pushes a folded vertex inside the model using the reflection method. There

are two types of sink folds in traditional origami, the closed sink and the open sink. The former does not fold rigidly, thus we will only consider the open sink in this section. The sink fold is called by Casale *et al.* “third fold”. They define it as the fold that articulates the structure in the points where the direction of the configuration changes, thus in concomitance with the intersection between what they call “second fold” and “first fold”. They assert that this fold changes the formal quality of the surface and changes its structural characteristics, because adding ribs to the structure affects its static behaviour, but it does not change the DOF or the global behaviour of the previous pattern during motion, it only changes the local geometry (Casale *et al.*, 2013). Nevertheless, sometimes the global rigid-flat-foldability could be compromised, depending on the configuration of the sink fold.

To make a sink fold with paper in the traditional way it is sufficient to fold a piece of paper making a single flat-folded vertex, at this point fold and unfold its tip marking a new crease through multiple layers of paper. Once unfolded the whole pattern we see a closed loop of new creases around the vertex which have to be marked as a mountain. Lastly, we reverse all the creases inside the loop of new creases and fold again the vertex pushing the tip inside the model.

In digital reconstruction, we can perform the same action using a reflection plane. However, even if we start from a flat-foldable degree-4 vertex, not every reflection plane guarantees a flat-rigid-foldable sink-folded pattern, because once pushed the tip of the vertex inside the model if the plane has not a specific orientation, it could cause collisions during folding. The following exemplification will help us understanding how the angle of the reflection plane may affect the rigid-flat-foldability of the pattern.

In the example shown in Figure 81, we perform a sink fold on a molecule with a single symmetric reverse fold constructed with the method proposed in section 4.6.1 and reflected with respect of a plane. To prevent the pattern with sink fold to be no more rigid-flat-foldable the angle between the reverse-folded linear crease and the sink fold plane must be bigger than 0° and smaller than 90° . In general, an easy way to construct a sink folded degree-4 vertex preserving the same kinematic of the original pattern without sink fold is to fold the pattern all the way to reach its blocking configuration, at this point reflect the tip of the degree-4 vertex with respect of the chosen reflection plane. The reflection plane must be angled making sure that self-intersections are avoided. If there are no self-intersections at the blocked state (or flat-folded state), then the kinematics of the mechanism is preserved, and the surface will not intersect during the whole motion.

Another issue that it needs to be considered when making a digital simulation of a sink fold, is that the reflection plane must intersect all the creases of the vertex inside the boundaries of the surface because, differently, it would generate a pair of new reverse folds instead of a sink fold (Figure 82).

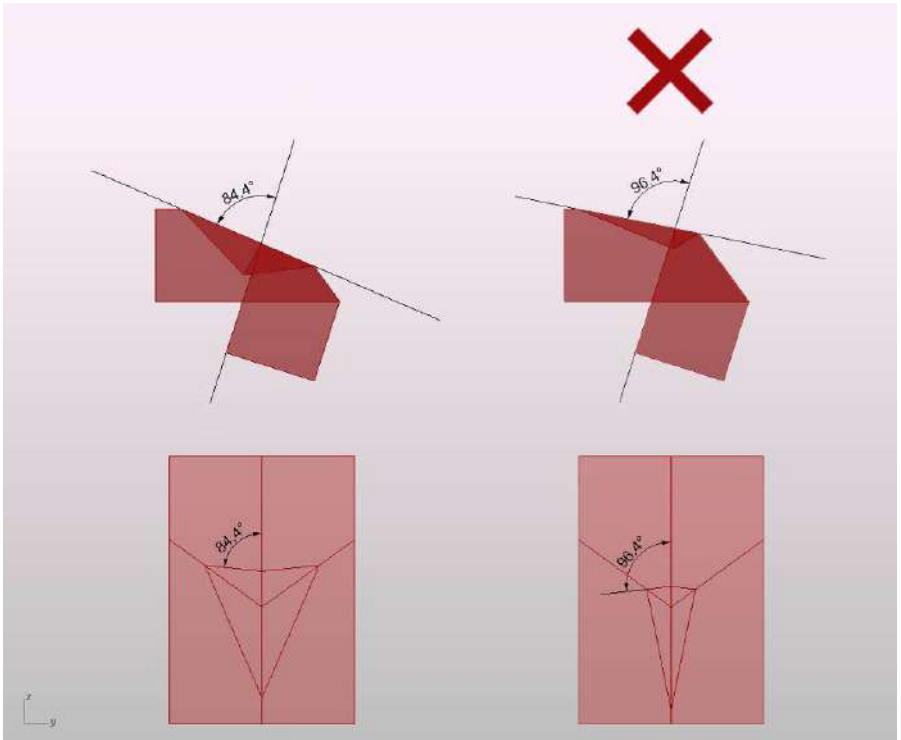


Figure 81 - Rigid-flat-foldability condition for the sink fold on reverse folded degree-4 vertex.

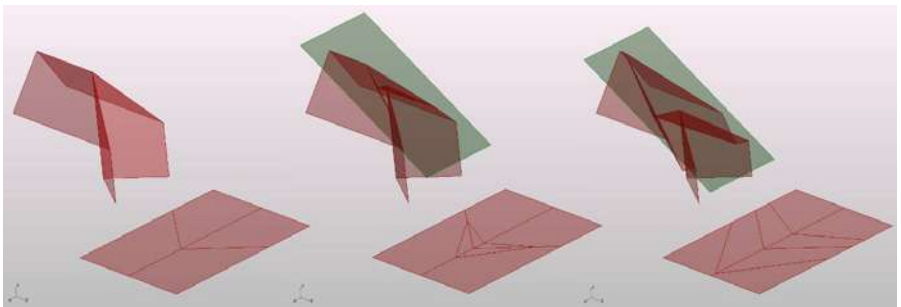


Figure 82 - Sink fold that degenerates into two simple reverse folds. This happens when the reflection plane does not intersect all the creases of the initial degree-4 vertex.

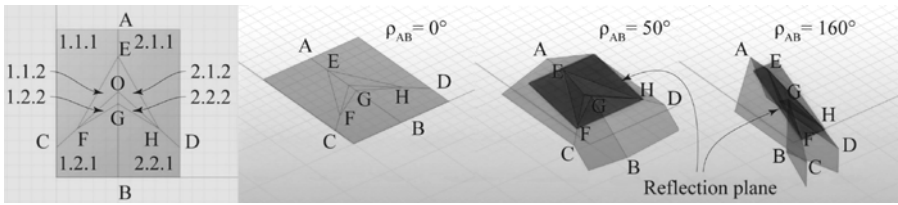


Figure 83 - Sink fold by reflecting a reverse folded molecule explained in 4.6.1.

To animate a pattern with sink fold we start from the reverse fold molecule and we reflect the tip of the degree-4 vertex. Refer to Figure 83 for the notation. First, start from a flat-foldable degree-4 vertex pattern, then add to that pattern some new creases that form a closed loop around the single vertex. The segments of the newly drawn loop of creases at the unfolded state must be symmetric with respects of the creases that they touch. Then use the same definition shown in 4.6.1 to animate the molecule with “symmetric reverse fold”. After this, move and rotate two of the segments of the sink fold drawn on the plane by copying and re-applying the rotations and translations of the relative faces to which they belong. In this case, we copy the transformations of the faces 1.1 and 2.1 of the original reverse fold molecule to the new segments EF and EH. Draw a plane passing through EF and EH segments and use it to split the surface. The folded surface is now split into two independent poly-surfaces, which are called in Grasshopper B-reps, they are split along the EFGH planar polygon. Pick the upper or the lower B-rep and reflect it with respects of the same plane. Lastly, we can animate the surface with the sink fold by moving the cursor of the slider that controls the fold angle. With this approach, the kinematics of the original starting molecule without sink fold is preserved. The generative algorithm is shown in Figure 84 and the initial cluster contains the algorithm explained in 4.6.1.

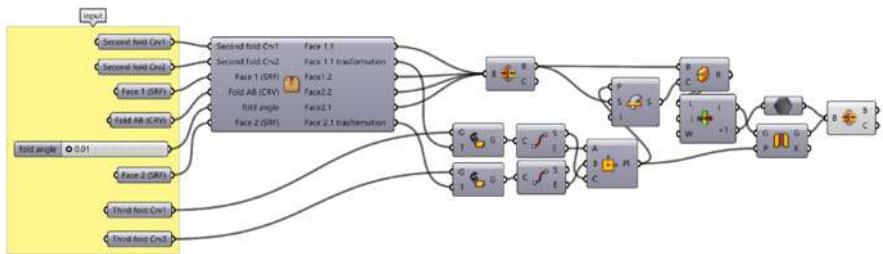


Figure 84 - Generative algorithm of the sink fold molecule animated with reflection with respect of a plane passing from at least two segments of the sink fold, the cluster node contains the algorithm explained in section 4.6.1.

4.8. Patterns with Single or Multiple Degree >4 Vertices

So far, all the patterns with internal vertices that we studied were made only by degree-4 vertices, and they had only one-DOF and for this reason, we were able to animate them with only one single controller crease that propagated its motion to all the other creases univocally. But how can we animate all the other patterns with vertices with a degree greater than four and with a DOF greater than one?

The possibilities are several, and the complexity of the problem increases drastically. At the present moment, the approach which is mostly used by the scientific community to animate a generic origami pattern is based on physical simulations that iteratively distribute the errors on every face and crease, preserving its developability and rigidity of the faces according to a certain tolerance. The software Freeform Origami by Tachi (Tachi, 2010b) works with this principle, as well as the web-based application Origami Simulator by Amanda Ghassaei (Ghassaei, n.d.). These simulators try to fold the crease pattern all at once. To do so, the surface is stretched and displaced during motion and it temporarily loses its developability for a small amount. These deformations are due to forces exerted by mountain and valley creases. Because there are forces involved, we often refer to this kind of simulations as physical simulations.

Even if some applications that fold almost any origami pattern exist, they are hard to include into a professional designing pipeline, due to the reasons we highlighted in section 1.3.1. Furthermore, integrating the simulation into the original design context would open new possibilities about the interaction with the context and other parts of the project that are not necessarily origami-related.

Fortunately, even if Grasshopper does not have built-in components for physical simulations, a plug-in called Kangaroo physics (Piker, n.d.) scripted by Daniel Piker will allow us to work with such type of simulations into Grasshopper without the necessity of scripting our own custom components. Kangaroo Physics is a “live physics engine for interactive simulation, form-finding, optimization and constraint solving”. This plugin implements in Grasshopper a set of new tools that facilitate working with (e.g.) forces, meshes, point clouds, and interconnected points. It also implements a wide set of “Goal nodes” which constrain some properties of the input geometry during the simulation, and a “Kangaroo Solver” which animates a given geometry according to the forces and constraints set by the goal nodes. Once set off the simulation, the solver generates a set of motion vectors for each goal node plugged into the “GoalObjects” input. The move vectors are applied to each particle they act on (e.g. the vertices

of a mesh, the endpoints of some segments, the origin points of a group of solids), and the solver iteratively minimises the total sum of the weighted squares of all the move distances. With this method all the constraints have a certain amount of error inversely proportional to the strength we set for each of them, thus the error is distributed non-uniformly to all the constraints.

The formulation that the Kangaroo solver uses to calculate the position P of each point is the following:

$$P_{i,new} = P_{i,cur} + \frac{\sum_{j=1}^n \omega_j G_j}{\sum_{j=1}^n \omega_j} \quad (31)$$

Where: i refers to the particle index, n is the number of goals acting on that particle, ω is the weighting and G is the moving vector for goal j (Brandt-Olsen, 2016).

For example, we can set the strength of the nodes that constrains the preservation of the developability of the original pattern to a very high value, and we can set the strength of the node that constrains the fold angle of each crease to a certain given fold angle with a lower strength. Like so even if the real fold angles of each crease may be different from the one that we set while preserving the invariance of the shape and planarity of the faces as much as possible.

This approach is similar to the one used by Tachi and Ghassaei in their applications, it is fast and efficient to calculate because it performs the folding of each fold all at once, but it stretches and deforms a bit the surface during motion. The same approach could solve all the cases shown in the previous sections faster, but with lower accuracy, thus it can be considered an alternative method but with different aims and needs.

4.8.1. Degree>4 Vertices – Physical Simulation

The aim of the algorithm presented in this section is to simulate the folding animation of an origami pattern with one or more degree-4 vertices or vertices with a degree greater than four, and a DOF greater than one, with a given mountain/valley assignment. To help the designer controlling the animated surface we implement the possibility of anchoring some points and sliding some other points on a given plane. Furthermore, for vertices with a DOF greater than one it will be possible to set a different fold angle for each crease or group of creases. The first part of the algorithm is focused on identifying the creases which are mountain, valley or unassigned, the second part is focused on setting some constraints that will be inputted into the “Kangaroo Solver” component. As a result, we will obtain the real-time folding simulation of the given pattern.

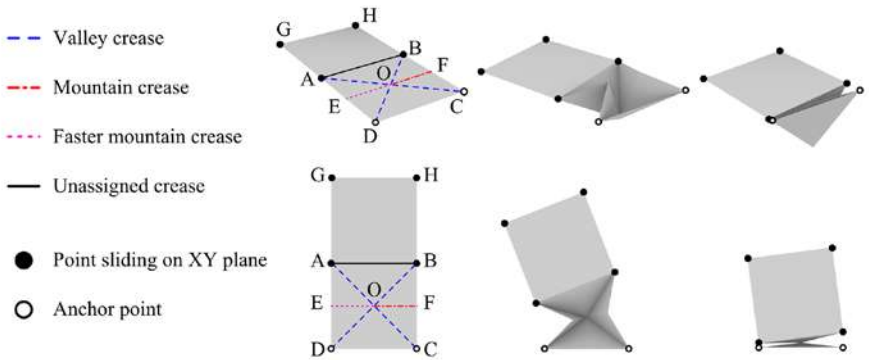


Figure 85 - An example of a degree>4 vertex, animated with physical simulation, applying different fold angles to the creases, in this way the animation is asymmetric even if the pattern is symmetric.

In Figure 85 we show an exemplification pattern with a generic degree-6 vertex with a DOF greater than one. The crease AB is left unassigned, the creases AO, BO, CO, DO are set to be folded with a positive given fold angle, the same angle, but negative, is assigned to FO. EO is a mountain crease as FO but its fold angle is set to be the double of the other crease angles. In this way, we experiment asymmetric folding motions on symmetric patterns.

The steps of the algorithm are the following. First, set the inputs of the definition: a planar surface which can be compared to the piece of paper that we are going to fold, and a set of straight lines arranged on the surface which are going to form the CP. Now divide the creases into two lists: mountain and valley. Then slice the surface along the creases so that each face of the crease pattern is a separate piece. Join all the surfaces into a single poly-surfaces (B-rep) and convert it into a simple mesh to allow kangaroo to process it. Now, compare the position of each input valley crease and mountain crease to the position of each edge of the mesh and divide them in two separate lists. To do so, use the “Closest Points” component comparing the mid-point of each mesh edge to the mid-point of each crease line. Extract for each mountain or valley edge of the mesh the two adjacent mesh faces. As a result of these steps, we obtained two branched lists. Each branch represents the mesh edge index and it contains a list made of two elements which are the two indices of the two adjacent faces. To do this last step more easily we used the Sandbox plugin for Grasshopper (Schwinn, n.d.) which analyses the input mesh topology and returns for each edge the list of adjacent faces automatically. Lastly, triangulate the mesh, because if the faces are all triangular it is sufficient to constrain the length of each edge to be able to preserve the developability of the surface.

Now we have all we need to set up the Kangaroo goal nodes. The goal nodes that we are going to use are: “CoPlanar”, “Length(line)”, “Anchor”, “OnPlane”, “NoFoldThrough”, “Hinge”, “Show” and “Grab”. The “CoPlanar” and “Length(line)” nodes are used to constrain the surface to remain developable and rigid. These two components are sufficient to guarantee and preserve the rigid-foldability of the surface because the “Length(line)” takes as input the edges of the triangulated mesh so that the developability is preserved, and the “CoPlanar” node takes as input the edges of each face of the mesh before the triangulation, so that the original faces will try to remain planar preserving the rigidity of the faces. Then we use the “Anchor” and “OnPlane” goal nodes to lock the surface in one place or to constrain the expansion and contraction of the surface along one particular plane so we prevent the surface to navigate the three-dimensional space uncontrollably while folding. Specifically, the “Anchor” node tries to keep a group of points on a given location, and the “OnPlane” node tries to keep a group of points on a given plane. The “OnPlane” node is different from the “CoPlanar” node, because the “OnPlane” node tries to keep the points on a given plane, if no plane is set as input it considers the plane as target attractor plane, the “CoPlanar” node instead, tries to keep the points on a plane (recomputed for each iteration) that is the interpolation of the input point cloud. The “NoFoldThrough” and “Hinge” nodes are the components responsible of the folding motion, the former prevents the surface to intersect once the adjacent faces reach their maximum fold angle, the latter takes two triangular faces as input and rotates them around the common edge of a certain given angle. We need two different “Hinge” nodes to perform mountain and valley folds, we multiply to -1 the angle that is inputted into the mountain “Hinge” node.

We said that only triangular faces can be processed by the “Hinge” node but also patterns with polygonal faces can be animated. It is sufficient to split their faces into triangles before plugging them into the “Hinge” node. However, we need to add an additional “Coplanar” constraint to keep the triangular faces that belong to the same polygonal face on the same plane because otherwise, the pattern would fold as if we added new unassigned creases. The higher the strength of this constraint is, the stiffer the faces are; and if we want the surface to behave more elastically, we can simply decrease the strength of this constraint.

To make some creases folding faster or slower than other as shown in Figure 85 we need a third “Hinge” node which is going to take as input the same angle multiplied or divided by a certain number (in this case we multiplied it by two). There is no limit to the number of “Hinge” nodes that can be used. We potentially could add one hinge node for every crease, setting a different fold angle for every one of them. In this way, we would increase the shaping freedom at

the expenses of easier operability. In the example shown in the figure, we added only one additional “Hinge” node as an exemplification.

Lastly, we use the “Show” node to set the geometry that we want to see during the simulation, and the “Grab” node to add the possibility to interact with the surface in real time into the Rhino viewport by grabbing the vertices with the mouse pointer. This node is useful to help the designers judging the DOF of the surface or helping the surface to fold properly if it gets stuck for any reason⁷ (Foschi, 2019). This algorithm can be used to simulate, potentially, the folding animation of any foldable pattern, even one-DOF patterns. This versatility makes this approach very effective.

4.8.2. Testing the Algorithm with Different Patterns

In the last section, we asserted that the folding animation based on physical simulation is very efficient and versatile. Once built the generative algorithm, the simulation is easy to set up because we only need to draw the CP and divide the creases into groups: mountain, valley and unassigned (and faster mountain and valley if needed). In this section we show some tests we did on well-known traditional patterns, to prove its efficiency and versatility. In Figure 86 and Figure 87 we simulated the folding of the traditional “magic ball” and “Yoshimura” patterns, both patterns have more than one DOF, thus we tested them with uniform and non-uniform fold-angle distribution. As the reader can see, the surface can be folded symmetrically or asymmetrically just by changing the fold angle speed of some designated creases.

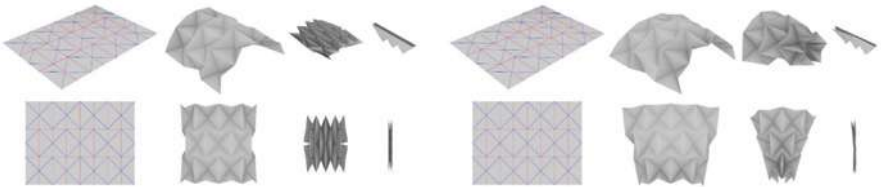


Figure 86 - Folding physical simulation of a traditional “magic ball” origami pattern, with uniform and non-uniform fold angle distribution.



Figure 87 - Folding physical simulation of a traditional “Yoshimura” origami pattern, with uniform and non-uniform fold angle distribution.

We verified the preservation of the developability of the pattern during and after the simulation, and we compared the area of the surface and the unfolded pattern before and after the animation, and as expected it loses the developability only for a small amount during motion. Nevertheless, once reached the equilibrium state, the global error is minimal, and, in most scenarios, it would be already acceptable. However, we can minimize the developability and rigidity errors even more, by increasing the strengths of the “CoPlanar” and “Length(line)” nodes by a great amount. Nevertheless, if we increase the strengths of these two nodes too much the animation would slow down drastically because the forces applied by the “Hinge” nodes would be overwhelmed by the forces applied by the “CoPlanar” and “Length(line)” components. Thus, to minimize the developability and rigidity errors without slowing down the animation too much, we can balance the strength of the various goal nodes during the simulation, and once reached the desired configuration, we can make the strengths of the goal nodes equal to zero all at once except for the strength of the “CoPlanar” and “Length(line)” nodes. In this way, the forces applied on the surface disappear except for the forces responsible of preserving the developability and the rigidity properties, so that the surface self-adjusts finding a new equilibrium configuration which is almost equal to the previous equilibrium configuration, but with minimal geometry errors.

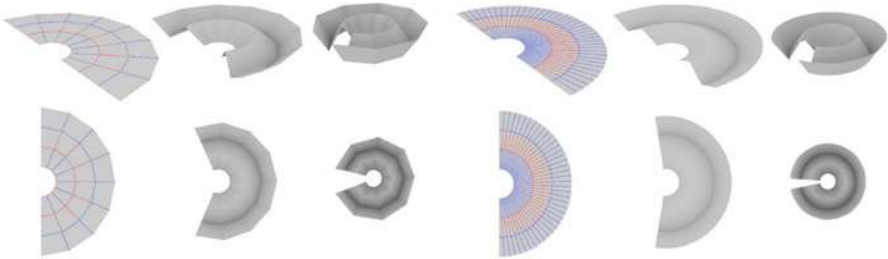


Figure 88 - Folding simulation of two discretized curve creases (with a different discretization degree) with physical simulation.

In Figure 88 it is shown the simulation of a pattern with two discretized curved creases. The patterns on the left and on the right start from the same two curve creases but with different discretization degrees. The ruling of the curved surface has mountain or valley assignments, in this way the folding animation is more stable, nevertheless it could also be left unassigned and they would assume a mountain or valley assignment automatically from the adjacent creases with an assigned

verse. This example approximates a curve crease because with discretization all the vertices are degree-4, thus the surface has only one-DOF. Nevertheless, in the real world, the curved creases can be performed only on flexible materials and they are usually made without deciding the ruling in advance, thus the ruling can change over time and the DOF would be more than one. This case is an easy special case that uses two concentric arcs of circumference as curved creases, thus the ruling match the direction of the radii of the circles, however for more complex curved creases the calculation of the ruling that is necessary to connect the curves while preserving the rigid foldability of the discretized pattern may not be as easy to calculate as in this case. There are some interesting studies about this problem (Bhooshan *et al.*, 2015; Demaine *et al.*, 2015, 2018; Dias *et al.*, 2012; Kilian *et al.*, 2008; Tachi, 2013; Tachi & Epps, 2011) but there is not a generalized theory yet. An interesting tool to study curve folding is the “Origami Simulator” by Amanda Ghassaei (cf. section 1.3.1).

4.8.3. Limits of the Algorithm and Known Problems – Pop-Up and Pop-Down

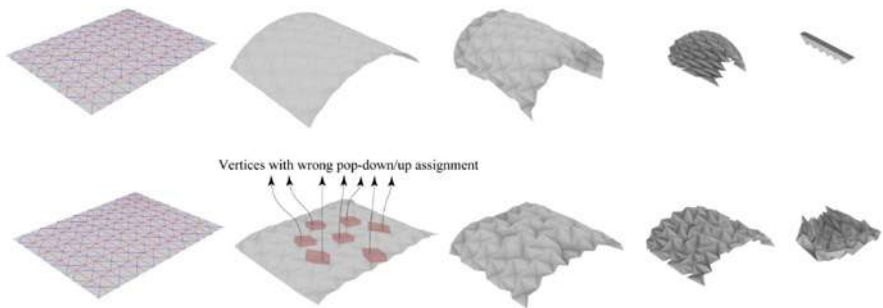


Figure 89 - Self-folding animation of a wide “magic ball” pattern, without and with the pop-up assignment problem.

Unfortunately, the physical simulation is not always reliable, the reason of that is that folding all the creases at the same time may cause some defects and problems on some vertices, it happens especially when there are a big number of creases in the pattern. We can try to tweak the strength values of the goal nodes or move the slider that animates the folding very slowly or grabbing the surface directly from the Rhinoceros viewport (by using the Kangaroo “Grab” goal node) trying to correct the defects while they occur, and most of the times these tricks work, but sometimes the simulation still fail for a known problem which sometimes occurs in self-foldable origami. This problem is caused by the fact

that most of the origami vertices can be folded in two different ways without changing the mountain/valley assignment. These two possible configurations are known as “pop-up” and “pop-down” states (Tachi & Hull, 2016). Some origami single vertex patterns can be folded pushing the central vertex upward or downward without changing their mountain/valley assignment, and the bifurcation of the motion happens at the completely unfolded state. Thus, when we try folding a pattern just by constraining the mountain valley assignment there are equal chances to get a pop-up or pop-down result if no other forces are applied. Furthermore, in patterns with a big number of vertices, because the physical simulation for its nature displace the vertices for a small amount while distributing the errors, it makes the surface behaving as a flexible surface, thus some vertices could take the path of the “pop-up” state and some other of the “pop-down” state even if they are not compatible, blocking the folding of the surface as shown in Figure 89.

To solve the problem, we can use many different methods, for the “magic ball” pattern shown in figure it was sufficient to slow down the animation of a big amount giving the necessary time to the Kangaroo solver to distribute the error to all the faces and to let the surface self-adjusting before passing the point of no-return where some vertices take the wrong pop-up/down assignment. Other methods could use attractor forces that may help the vertices to move in the correct direction from the first instant, or we could decrease the stiffness of the edges making the pattern more willing to self-correct exploiting the flexibility of the faces. Anyway, this kind of solutions has to be evaluated and tested case by case.

This kind of defects does not happen only in digital simulations, but also in the real world if the surface is flexible enough. The implications of these defects on real folded materials has been extensively studied by Silverberg *et al.* who consider a Miura-ori defected pattern as a case study to make some considerations about the design of reprogrammable mechanical metamaterials (Silverberg *et al.*, 2014).

PATTERN DESIGN FROM A GIVEN SHAPE

Designing an origami-inspired object does not always require solving its kinematics and digitally simulate the folding and unfolding of the surface. Animating the surface is useful for example for rigid-foldable kinetic mechanisms. However, movable mechanisms, even if they are one of the most important targets of applied origami, are not the only cases that may benefit from origami properties. For example, for a paper lampshade, or a cardboard box, we do not necessarily need to simulate the folding and unfolding of the pattern, but we may want to manipulate the three-dimensional object in space while preserving its developability. Thus, in this part of the book, we present some case studies that exemplify the design of objects or buildings inspired by already existing projects, through the use of specific algorithms developed with Grasshopper (Rutten, n.d.; Tedeschi, 2014) that are aimed to achieve a three-dimensional folded configuration that is developable. Thus, instead of starting from the unfolded pattern, we start from reference geometries in space (e.g. curves, surfaces, meshes) that we consider as attractors, guides, rails, anchor points, and we build the folded geometry on them while following strict rules that guarantee the developability of the CP. These case studies are mainly focused on the construction of the generative algorithms, and they do not solve all the issues that may arise from fabrication. We will focus on the fabrication problems in chapter 6.

5.1. Lampshade – Vertices Extrusion and Reflection

Imagine having an old lamp by Le Klint (Le Klint, n.d.), that is missing its original shade, and we want to renew it by making a different shade that fits perfectly the old structure. Because we already have a reference structure, we cannot use any accordion-shaped piece of paper because with a high probability it would not perfectly fit on it. For example, we cannot use a straight accordion, because, as we saw in section 4.5.2 it is only conformable into cones if no elastic

deformations are allowed. We could instead use the method explained in section 4.5.4 to design an accordion with converging creases that would fit perfectly on a cone. However, that solution leaves no room for creativity, thus, because the design possibilities are endless, in this section we propose a more versatile procedure that allows us to manipulate the folded shape constraining it to the two circular rails while preserving its developability. We start generating a triangulated accordion between the two rails, and to make it more appealing we reflect the bottom points with respects of specifically placed reflection planes, and we deform and manipulate the global shape to make it asymmetric.

Refer to Figure 90. First, draw two concentric circles on the same plane, then move on the Z axis one of the two circles. Draw an equal number of equally spaced points on the two circles. Take the even points of the upper

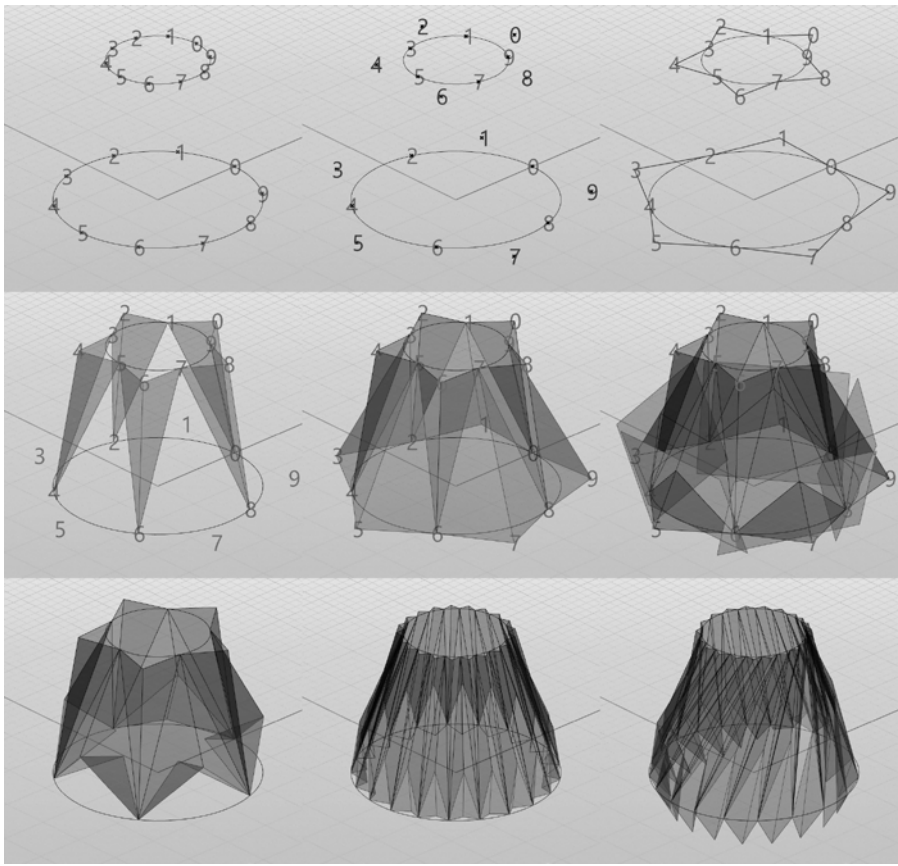


Figure 90 - Folded lamp shade algorithm, steps.

circle and move them away from the circle along the radii directions. Do the same thing with the odd points of the bottom circle. Connect the points of the upper circle to make a zig-zag-shaped closed polyline. Do the same thing with the bottom points. Select the two segments adjacent to each even point of the upper polyline and extrude them to the relative even point of the bottom polyline making triangular faces. Do the same thing with the bottom polyline but using the odd points. With this method, we constructed a “chain” of triangular faces that is guaranteed to be a developable surface because the triangles are always planar. Thus, starting from this developable surface, to push even further the research of an appealing shape, we reflect the bottom points inside with planes passing from pairs of even points and a third point, not on the horizontal plane. Now the algorithm is ready, and we can play with the inputs to explore different shapes. We can add more faces, or change the angle of the reflection planes, or even rotate independently the upper and bottom circles to get a twisted look. It is also possible to add variation to the design by adding “Graph Mappers” nodes as shown in section 4.5.3 to distribute non-uniformly distances, lengths and angles. In the example shown in Figure 91 we reported two examples of possible shapes that can be obtained with this approach¹ (Foschi, 2019).



Figure 91 - Two possible solutions of lampshades for the same support structure, achieved with the same algorithm.

The last thing to do to be able to fabricate this lamp cover is to develop on a plane the folded surface and check if there are no overlapping parts on the unfolded pattern. If some parts are overlapping, we can either split the pattern into different parts and assemble them later, or we can change a bit the design, for example lowering the number of points or shortening the moving distances of the points, to be able to get a CP which does not overlap.

5.2. Folded Façade – Vertices Extrusion from Reference Curved Rails

The same technique used for the lampshade can be used to design any other geometry that uses triangulated accordions placed on support rails, like the façade of the Biomedical Research Center by Vaillo & Irigaray Architects. In this section we show a variation of the project by Vaillo and Irigaray, using curved support rails instead of straight ones. In this case, the property of being developable is probably not crucial for construction purposes due to the bigger dimensions, but it may be useful for decreasing the waste and trims from the production phase. The algorithm has the same structure of the algorithm shown in section 5.1, but we add variation by drawing additional reference curve rails instead of using “Graph Mapper” nodes.

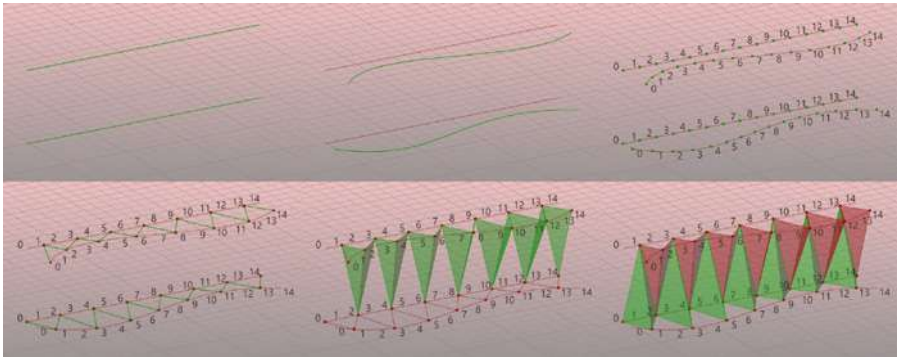


Figure 92 - Folded triangulated façade algorithm, steps.

Refer to Figure 92. Draw one straight line, copy it and move it along the Z axis. Draw two freeform curves close to the straight lines. Divide the curves into an equal number of pieces. Connect the even points of the bottom freeform curve with the odd points of the relative closest straight line. Do the same thing with the upper points but with inverted even/odd assignment. Explode the upper polyline in independent segments and split the list of segments into sub-lists with two elements each. Extrude each pair of segments to the relative closest even point of the bottom straight line. Do the same thing with the bottom segments. As well as the algorithm shown in 5.1, this algorithm generates a developable “” chain of triangular faces. The developability is guaranteed by the fact that the triangles are always planar and there are not internal vertices in the CP^2 (Foschi, 2019). In Figure 93 it is shown a possible application where many folded patches are anchored to a generic building façade.

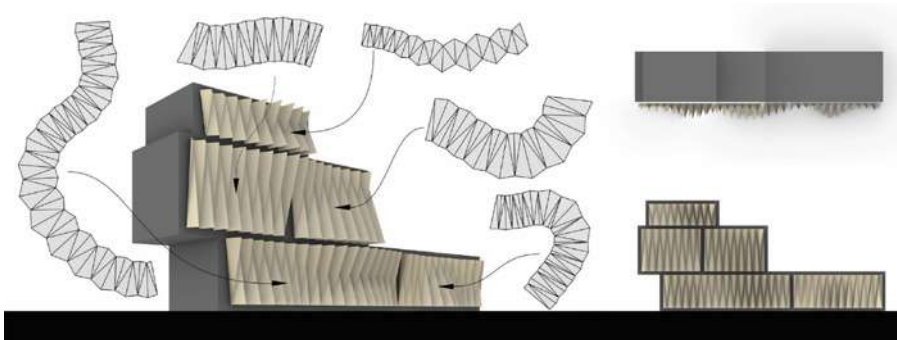


Figure 93 - An example of a foldable facade shaped like a triangulated accordion on curved rails.

5.3. Building Envelope – Reflection of a Creased Developable Surface

The project by Localarchitecture and Danilo Mondada of the temporary chapel for the Deaconesses in Saint-Loup is one beautiful example of architecture designed with origami rules. In this case, it has been used the reflection method (cf. section 4.6.1) (Mitani & Igarashi, 2011) as a tool for the shape-finding phase (Buri & Weinand, 2008). In this case, from a structural point of view, the developability property may even have a counterproductive effect on the project because being developable entails the fact that it may tend to slide until it lies on a plane, which is not desirable for a structure that must be self-supporting. However, with properly designed anchor points this factor may have no relevance at all. Nevertheless, developability may play an important role in the manufacturing process. For example, for a wood and hinges structure, the panels might be assembled on the ground and lifted up all at once in a single motion limiting the need of scaffoldings and cranes; or if built in concrete, the panels may be pre-casted in situ, placing them one next to each other, without empty spaces between them, with minimal shuttering, minimal soil occupation, and minimal costs of transportation.

In this section we explore an approach to design this inspiring and versatile shape proposed by Buri and Weinand, starting from a polyline path and a section profile.

Refer to Figure 94. Draw a polyline with the first point on 0 on the XZ plane (this will be the path). Draw a polyline with the first point on 0 on the XY plane (this will be the profile). Build the first reflection plane on the first kink point of the path, its normal vector is directed like the bisector of the angle between the

first two segments of the path. Now project the profile points to the reflection plane along the direction of the first segment of the path, and draw a new polyline passing through them. Now perform a straight loft with the first profile and the projected profile as sections. The first section of the folded surface is built, and it is an accordion with parallel creases. To build the restart sections we need to repeat this process on all the other segments of the path. To do so, we need to exclude the first segment of the path and switch the original profile with the new projected profile and apply the same algorithm over and over until all the segments of the path are processed. To perform this kind of looping definitions in grasshopper, we need to use the Anemone plug-in that we already introduced in section 4.5.5.

The algorithm explained above fails when it reaches the last segment because there is not a subsequent segment to calculate the bisector with. Thus, to solve this problem, we need to set an exception for the last segment as follows. Every time the algorithm repeats, it erases the first segment of the path, when only one segment remains it returns “True”. “True” in Grasshopper can also be interpreted as a 1, and False as a 0, 1 and 0 are also the indices of the list containing the two planes, if the algorithm returns “True” use the XY plane, if it is “False” build the plane perpendicular to the bisector of two consecutive segments³ (Foschi, 2019).

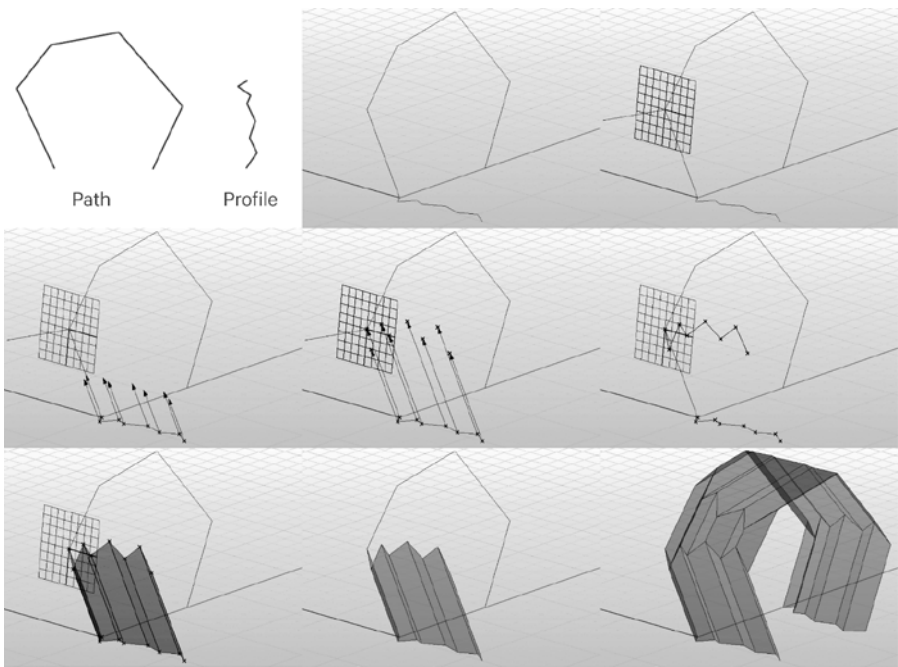


Figure 94 - Folded roof by reflection, steps of the algorithm.



Figure 95 - Building envelope example shaped like a reflected accordion.

5.4. Curve-Folded Table

5.4.1. Reflection of a Developable Curved Surface

The reflection method shown in section 5.3 is not only valid to make reverse folds starting from folded surfaces with linear creases, but it can also be used to reverse fold generic developable ruled surfaces (Mitani & Igarashi, 2011) generating curved creases. For example, we can use the reflection method to design a curved folded table similar to the one designed by Tachi applying the method shown in Figure 96.

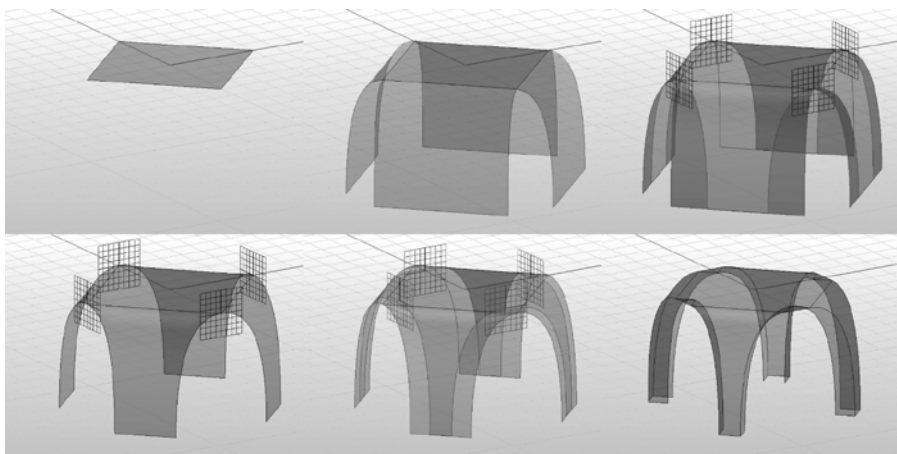


Figure 96 - Curve-folded table algorithm, steps.

The method works as follows. Draw a planar horizontal polygon (a square in this case). Extrude the edges along curved paths tangent to the surface. Draw vertical planes with origins coincident to the vertices of the polygon and rotate them around their relative Z axes to make them intersect both the adjacent surfaces (the planes are perpendicular to the relative angle bisector in this case). Split the poly-surface and erase the outer split as shown in the figure. Move the planes along their local Z-axis to make them intersect the horizontal polygon and the four curved legs. Then, split the polysurface again with the planes paying attention to not intersect the curved sections. Mirror the outer splits with respect of the relative splitting plane, so that we obtained a developable surface with curved creases. As we show in Figure 97, we can achieve different results by changing the number of sides, the shape of the initial polygon or the orientation of the reflection planes⁴ (Foschi, 2019).

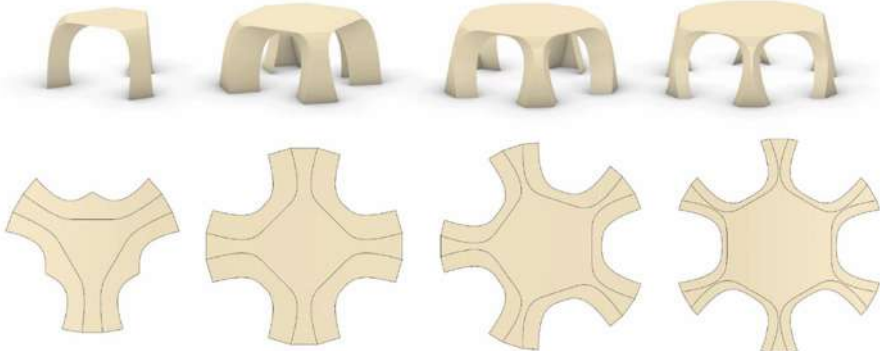


Figure 97 - Curve-folded table variations and unrolled CPs.

5.4.2. Discretization of the Curved Crease

The pattern that we obtained could already be used as a cutting template to be fed to a CNC machine that would return a pre-creased thin sheet of metal ready to be folded. However, folded metal has a limited number of folding cycles, thus, to make a kinetic deployable table this might not be the optimal solution. An alternative solution, more suitable for making moving mechanisms, might be the discretization of the curved surface into planar quadrangular faces. In general, the discretization of a curved folded surface into planar faces is not an easy problem to solve. However, it becomes very easy if we generate the curve fold by reflecting a developable cylinder or cone. This is because in general, in a developable surface, two consecutive linear generatrices

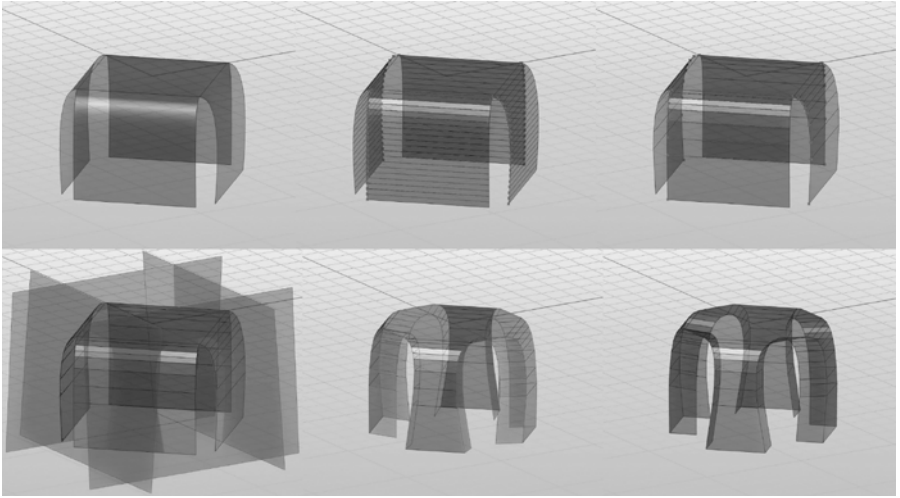


Figure 98 - Discretization of a curve-folded table.

do not always lie on the same plane unless they are infinitesimally spaced. Cylinders and cones however are special cases where all the consecutive generatrices lie on the same plane regardless of the distance between them. This means that we can simply convert the curved surfaces that we already used in the previous algorithm into a discretized polysurface where all the faces are planar and quadrangular, and we can reflect it with the same planes of reflection. We started from the same algorithm presented in the last section and we implemented it as follows.

Refer to Figure 98. Draw a developable polysurface shaped like a table as we did in section 5.4.1. Extract the lateral boundary curve of each leg and divide it into a certain number of parts. We want to add more points where the curvature is high. There are many approaches to solve this problem, because we do not need to have a perfect relation between the position of points and the curvature, we can use the following workaround. Create a very dense polyline that approximate the curve and reduce it with the “Reduce polyline” component, this component compares the angles between the segments adjacent to each point and according to a certain tolerance keeps or erases the point. After that extrude the polyline to generate a polysurface that approximates the initial curved surface. The polysurface so generated is a developable surface made by planar rectangular faces, thus we can use the reflection method as we did in the previous section. In Figure 106 we show the correct unfolding of the discretized pattern.

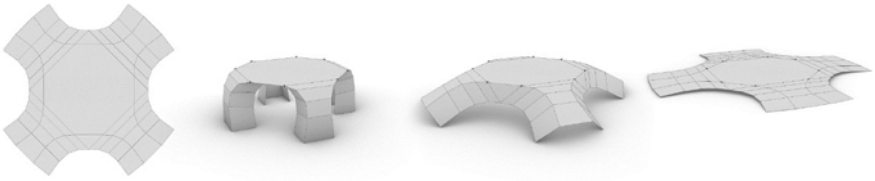


Figure 99 - Unfolding of the discretized curve-folded table, which is only made by planar quadrangular faces.

5.5. Conformable Corrugated Suspended Ceiling⁵

The installations “Tessel” by David Letellier and Lab[au] and “Resonant chamber” by RVTR, are two perfect examples of sculptures in form of suspended ceilings that make use of a folded corrugated surface as a tool to create interesting shapes and movements. Both installations are kinetic sculptures that change their shape interacting with the perception of sound in a certain space. The first one, “Tessel”, is programmed to react to the sound and “dancing” with it. The aim of this sculpture is “[...] combining influences that question the link between geometry, movement and chaos, thus continuing the quest for beauty in the synesthetic perception of sound and spatial phenomena.” (Letellier, 2010). Perceptively it creates a beautiful visual and acoustic effect that is similar to the behaviour of an almost living creature. The second project, “Resonant chamber”, has a more sophisticated behaviour and connects a beautiful geometric and enigmatic appearance to the practical need of acoustic optimization. It also combines the principles of rigid origami and dynamic, spatial, material and electro-acoustic technologies (Furuto, 2012). The intention of this project is creating a continuous surface that changes automatically according to the sound conditions to influence its perception in the environment where it is placed, one of the possible application fields pointed out by the designers is a theatre or any other place with a variable audience, where the acoustic conditions are very important and they may vary in relation to the distribution and number of the spectators.

Some of the most obvious challenges that the designers of these sculpture had to face were how to control the motion of these surfaces; and how to hang them with an optimized number of cables. First of all, they have chosen patterns with more than one single DOF, to be able to increase the shaping possibilities, then they connected motors to the hinges, or they used retractile cables to be able to change the shape of the surface dynamically. In this section we are going

to explore this type of concept through parametric design, to be able to conform the shape as a given reference surface and to optimize the number of necessary cables to keep the shape in position once hanged.

5.5.1. Conformation of a Rigid Creased Surface to a Curved Surface

First of all, we focus on the shaping of the corrugation. The aim of the following method is to change the shape of a given creased rigid surface conforming it to a given curved surface without changing the pattern. To do so we are going to use Kangaroo Physics (Piker, n.d.), the plug-in for Grasshopper we already introduced in 4.8, where it has been used to animate the folding and unfolding of degree >4 CPs. Contrary to what we did in 4.8, this time we are not going to set mountain/valley assignment, because we are more concerned about the global shape of the corrugation.

First of all, draw a planar surface. Then, draw the creases, split the surface with them, generating a polysurface, and convert the polysurface into a mesh. In this case, we draw the creases to generate only triangular faces. Draw a reference curved surface in the proximity of the original surface with similar proportions, this surface will be the attractor surface. Now we need to set up the Kangaroo nodes as shown in Figure 101 in order to move the mesh vertices to the attractor surface along the lines that connects the vertices and their relative closest point on the surface and at the same time preserving the developability of the mesh. To do so, we add a “Length(Line)” and an “OnMesh” goal nodes and connect them to the Kangaroo solver. The “Length(Line)” node constrains the length of the edges of the mesh, and the “OnMesh” node will try to bring the mesh vertices on the surface. Because the Kangaroo solver, once set off, tries to minimize the errors of the goal nodes by weighing them with the strength input of each node, we need to set the strength of the “Length(Line)” node much higher than the strength of the “OnMesh” node in order to prioritize the preservation of the developability of the surface more than the minimization of the distances between attracted points and surface. Like so, when the simulation starts, the mesh vertices start moving toward the surface while keeping their relative distance, thus keeping the faces of the surface rigid. By changing the attractor surface, the corrugation updates its shape trying to conform as better as possible to the new configuration.

In this case study, we used only triangular faces as in the “Tessel” reference, but what if we wanted to use a CP with quadrangular faces? We know that degree-4 vertices may generate patterns with only one-DOF. Thus, first of all, we need to test if the DOF of the CP is one or more. This is important because

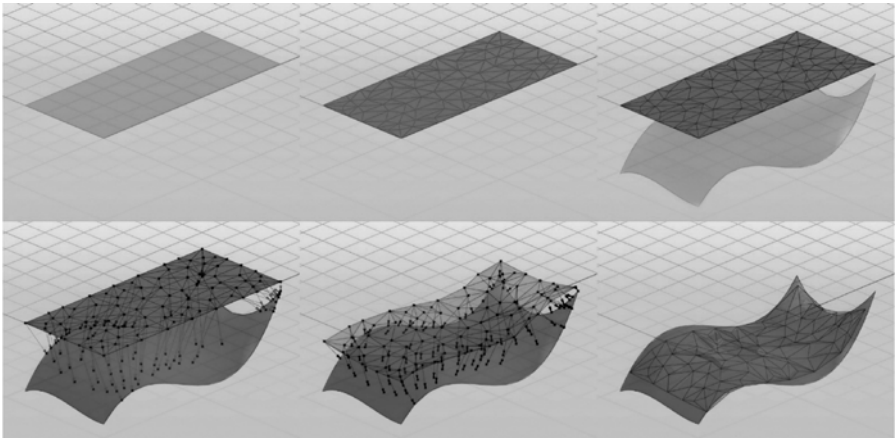


Figure 100 - Conformation of a corrugation to curved surfaces without changing the CP.

if it is one, the surface will not be able to be conformed to a freeform surface, thus a one-DOF pattern would not be suitable for this kind of applications. If it is more than one and we want to keep the quadrangular faces planar we need to add a “CoPlanar” node setting as input every vertex relative to each face of the mesh so that the faces will remain planar. Also, we will need to triangulate the mesh before setting it as input for the “Length(Line)” node. Constraining the diagonals lengths and keeping the boundaries of the quadrangular faces planar we guarantee the preservation of the developability and rigidity of the faces.

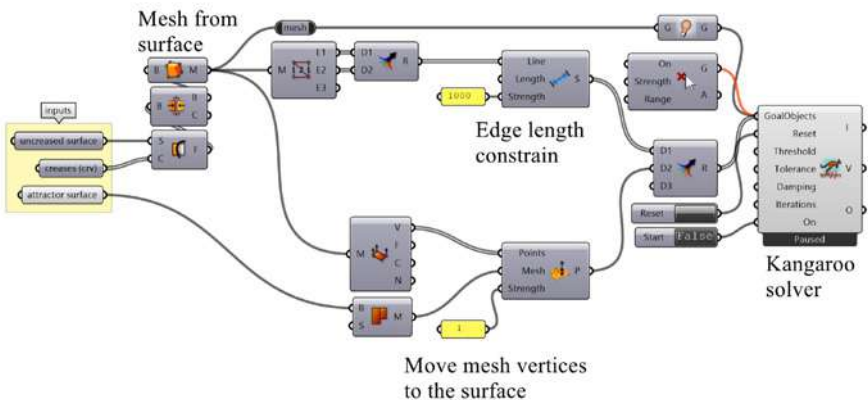


Figure 101 - Generative algorithm of the conformation of a corrugation to a curved surface using Kangaroo.

5.5.2. Conformation of a Flexible Miura-ori to a Curved Surface

Not only an unfolded pre-creased pattern can be conformed to curved surfaces, but also semi-pre-folded patterns. With semi-pre-folded pattern, however, we cannot make all the vertices of the mesh being attracted from the surface, because otherwise, the corrugation would flatten out completely, losing the mountain/valley assignment, trying to conform as best as it can to the curved surface. In the case of the Miura-ori, shown in Figure 102, the vertices that are attracted from the surface are all the bottom vertices. Furthermore, the Miura-ori is a pattern with one single DOF, thus we had to triangulate all the faces to increase the DOF allowing non-uniform out-of-plane deformations of the global shape. We treated the diagonals as loose hinges as all the other creases. However, we can also simulate a flexible behaviour of the Miura-ori by adding a rotational stiffness to the faces adjacent to the diagonals. We can do that by adding a “Hinge” node that tries to keep the dihedral angles between the triangles divided by the diagonal equal to 180° . Thus, the quadrangular faces would have a more or less flexible behaviour according to the “Strength” value set to be very high for a stiffer behaviour, or low for a more flexible behaviour. We also added a “SphereCollide” node among the goal nodes, to avoid collisions between faces⁶ (Foschi, 2019).

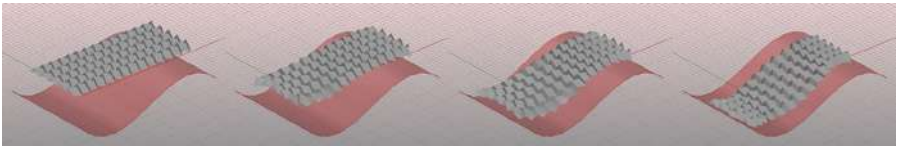


Figure 102 - Conformation simulation keyframes.

Most of the origami pattern cannot be conformed to any curved surface. The accuracy of the conformation depends on the pattern itself or from its scale in relation to the local curvature of the attractor surface. For example, in Figure 103 the triangulated Miura-ori pattern is conformed to the same cylindrical surface with different results according to the different orientation of the pattern in relation to the surface. In the first case, the Miura-ori conforms better than in the second case. The average distance between the attracted points and the surface in the former is approximately ten times smaller than the one in the latter. The average distance between the attracted points and the surface is an important factor that contributes evaluating the quality of the conformation, but most of the times not every section of the corrugation conforms equally to the surface, so how can we visualize better the error distribution?

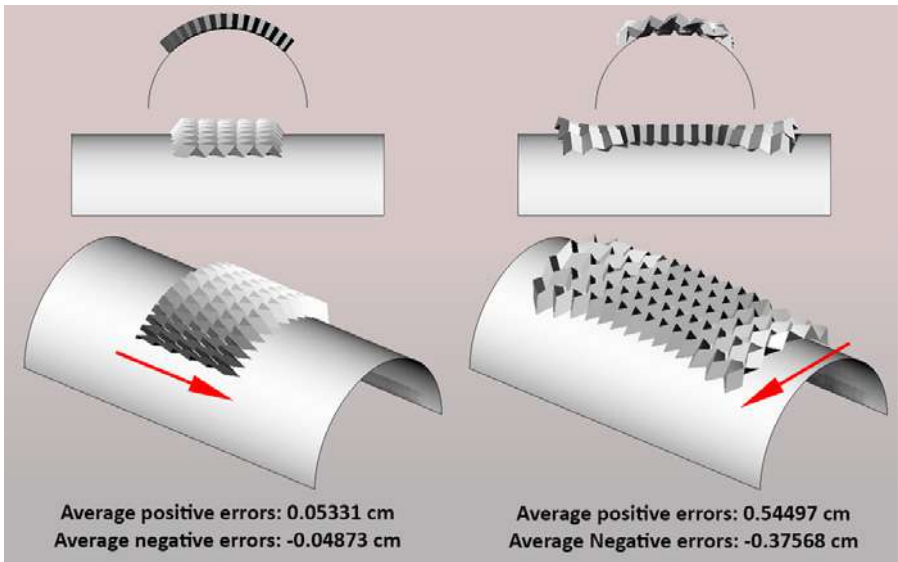


Figure 103 - Different results due to the different orientation of the pattern compared to the attractor surface.

To visualize better the error distribution we applied a gradient texture map on the surface coloured in grey scale that colours with black the areas of the surface close to the projection of the most distant points from the negative side of the surface, with white the areas close to the projection of the most distant points from the positive side of the surface and with grey the areas close to the projections of the most accurate points, as shown in Figure 105. In Figure 104 it is shown the generative algorithm to colour a surface with Grasshopper on the left, and the histogram of the error distribution of the configuration shown in Figure 105 on the right.

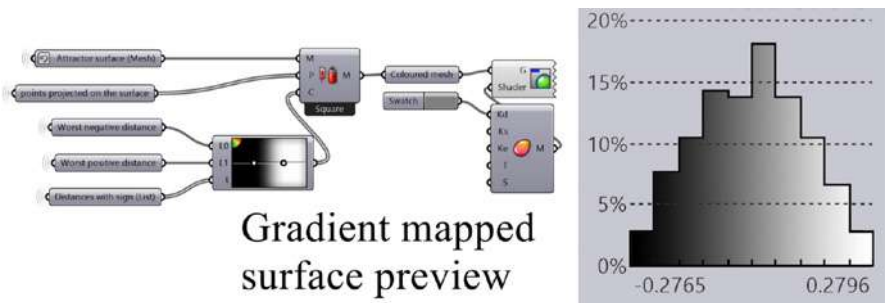


Figure 104 - Gradient error map, generative algorithm, and the histogram of the error distribution.

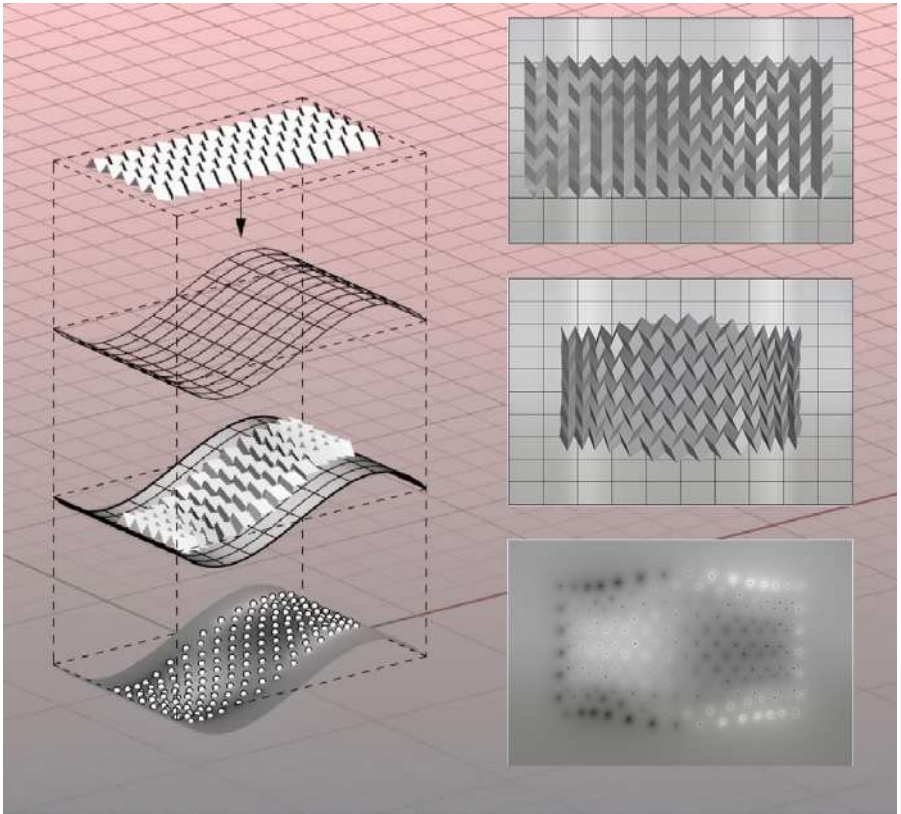


Figure 105 - Gradient error map on a single curvature curved surface. Axonometric and top view.

5.5.3. Optimization of Supporting Cables and Anchor Points

The most trivial way of hanging a corrugated surface while keeping its shape perfectly is by anchoring all the vertices of the mesh, attaching them to wires with specific lengths. The lengths are given by the distances of the vertices themselves and their relative projected points on the ceiling. Because the force of gravity pulls the vertices vertically, they try to move downward, but because all the loads are absorbed entirely by the wires, the system is balanced and there is no residual movement. However, for complex tessellations and corrugations hanging hundreds of vertices may be not optimal in the real world. However, most of the corrugations built with loose hinges are impossible to hang with fewer cables than vertices, while keeping perfectly their shape, thus we need to choose either if anchoring all vertices, or if partially lose the initial shape. This

is true for corrugations with faces built for example with wood attached with hinges, but it is not necessarily true for corrugations folded for example with cardboard. Usually, a continuous material creased does not have a loose behaviour, because the creases still have some sort of stiffness, thus we can exploit that stiffness to lower the number of anchor points while preserving their global shape. Furthermore, some corrugations have a lower number of DOF and they behave more rigidly than others. Therefore, we need to analyse case by case to be able to find the optimal solution. In the digital world is very hard to make a physically accurate simulation of a creased sheet made by a continuous material, because in the real world a crease corresponds to a plastic deformation, thus the stresses and the internal forces are hard to foresee and to replicate precisely digitally, because they may be not homogeneous according to how much strength we used while creasing or how many times we folded and unfolded that specific crease. Thus, for those cases where we need to hang a corrugation made by a folded sheet of continuous material, once tested if the conformation is possible with that pattern on a specific curved surface, it is preferred to realize the prototype of the surface with the same material at a similar scale and make tests in the real world. For those cases with loose hinges, however, we can simulate their behaviour more easily.

In this section, we are going to test some specific set of anchor points on a conformed Miura-ori by adding the force of gravity to our digital environment. The algorithm that we used to add the wires and to test the equilibrium of the system with the force of gravity works as follows⁷ (Foschi, 2019).

First, set up the standard nodes of Kangaroo as shown in previous sections. Among the standard goal nodes used to preserve the developability of the corrugation, add another “Length(Line)” node for the wires with a lower strength compared to the “Length(Line)” node used for the Miura-ori edges. Then, add a “Load” node which pulls the assigned vertices along an assigned vector (we assigned a vector directed as the Z-axis with an amplitude of -1, and we set as input all the vertices of the Miura-ori). Then project all the Miura-ori vertices to a virtual planar ceiling, placed above the conformed Miura-ori, and anchor those points in place adding an “Anchor” goal node.

By setting off the Kangaroo solver now, we would not see any change into the system, because all the vertices are anchored, and all the forces are balanced, thus we need first to decrease the number of wires to see some changes. Decrease the number of wires by culling some vertices among the list of vertices that are projected to the ceiling plane, while doing that try to find a set of vertices that, once attached to the wires and hanged, preserves the global shape of the corru-

gation without making it collapse. After tested many sets of vertices, we found out that anchoring the first ring of vertices of the Miura-ori is not sufficient to keep the global shape, however, anchoring the second ring of vertices is sufficient to find an equilibrium state which is very close to the initial configuration as shown in Figure 106. In the figure we set the Miura-ori to have loose hinges, in fact even if the global shape is preserved while anchoring the second ring of vertices, the outer vertices sometimes collapse, it is particularly evident for the vertices of the outer left border.

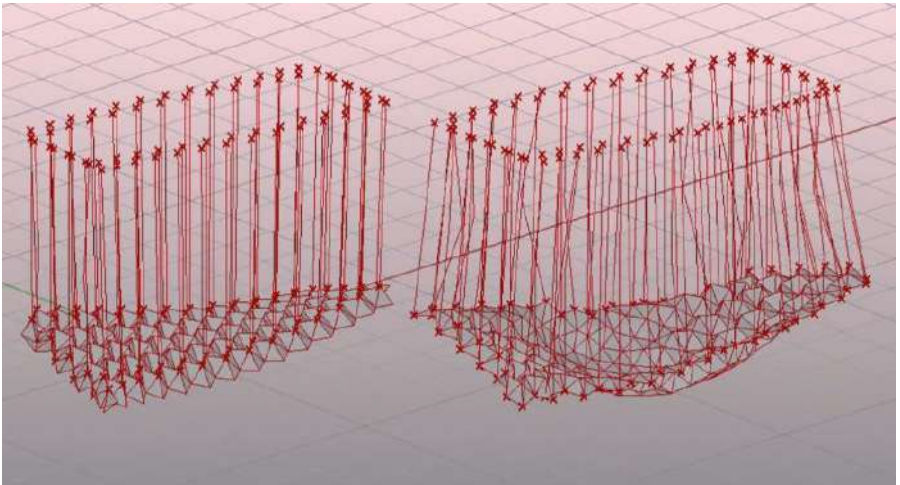


Figure 106 - Gravity simulations comparison of a conformed Miura-ori with different patterns of anchored vertices. On the left we anchored the upper ring of external vertices and the system is in an equilibrium state, on the right we anchored the bottom ring of external vertices and the system collapse.

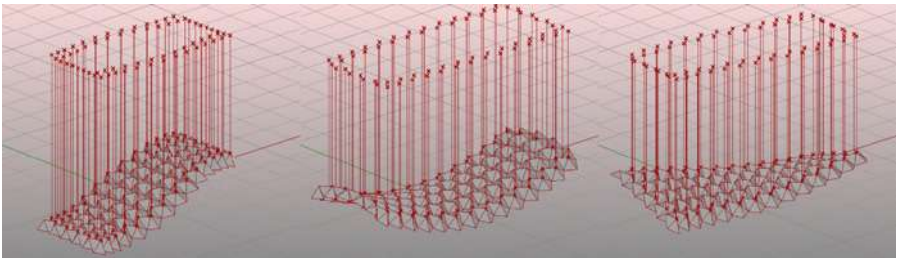


Figure 107 - Anchoring the second ring of vertices to find an equilibrium state once applied the force of gravity.

In Figure 107 on the contrary we added a little bit of reciprocal rotational stiffness to the faces adjacent to the diagonals, simulating a behaviour qualitatively similar to a flexible Miura-ori folded with paper, and as you can see only by anchoring the second ring of vertices the global shape is almost perfectly preserved in all the cases we tested.

Because we added some reciprocal rotational stiffness to the faces simulating a behaviour similar to a sheet of paper, anchoring the second ring of vertices entirely is probably an over-kill. Thus, the next question is how can we find a more optimized solution without necessarily guessing a better set of anchored vertices? To optimize even more the vertices without guessing, we propose a method that compares the lengths of the wires before and after applying gravity, so that we can identify the wires that are completely or almost completely unloaded and exclude them. Because the lengths of the cables are constrained with a “Line(length)” node also known as “spring” node, they behave exactly like springs. This means that they obey the relationship:

$$P_{i,new} = P_{i,cur} + \frac{\sum_{j=1}^n \omega_j \cdot G_j}{\sum_{j=1}^n \omega_j} \tag{32}$$

Where F is the Force, k is the stiffness and x is the extension (or Δ length). Thus, Δ length and tension are proportional. Therefore, we can use the length variation to apply a colour to the wires and visualize qualitatively the tension applied to each cable using the standard gradient map for tensions that goes from green to yellow to red as shown in Figure 108.

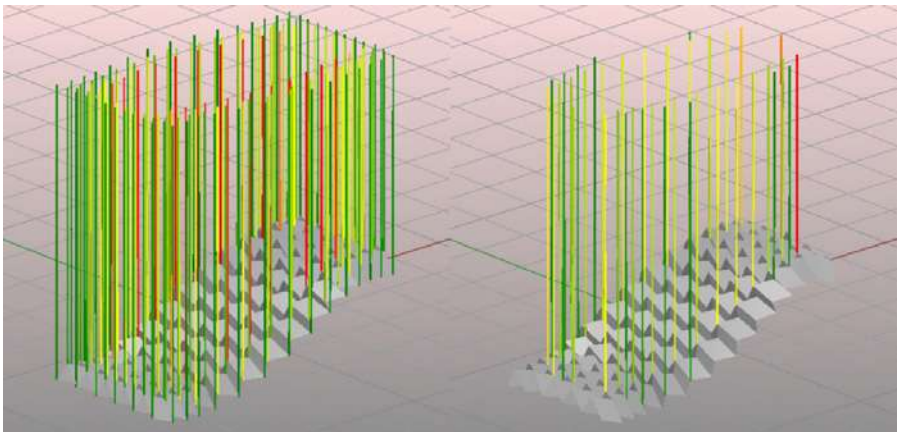


Figure 108 - Optimization of the cables through tension analysis.

After erasing unloaded (or almost unloaded) wires we can run the simulation again and check if the corrugation keeps its shape. If we try to apply this method before erasing some cables by guessing we would not have found any unloaded cable, thus it may not work properly. Furthermore, in some corrugations even erasing a single wire would change its shape greatly, thus every case must be analysed specifically.

5.5.4. Changing Shape to the Surface – Adjusting Cables Lengths

In section 5.5.1 and 5.5.2 we show how to conform a creased surface to a curved reference surface and in section 5.5.3 we show how to hang it with wires. In this section, we are going to propose a reversed approach, that starts from a hanged un-conformed surface that we conform changing the lengths of the cables.

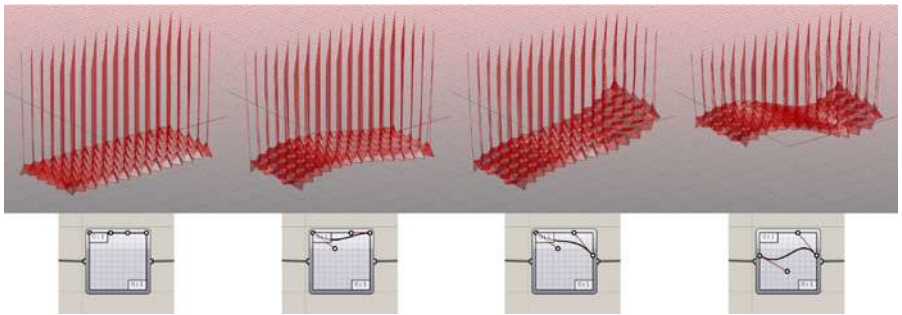


Figure 109 - Moving the surface changing cables lengths using a “Graph Mapper” node.

In Figure 109 we show how a semi-pre-creased Miura-ori can be shaped by using the “Graph Mapper” node. The generative algorithm works as follows⁸ (Foschi, 2019).

Import the semi-pre-creased Miura-ori as a mesh. Extract all the vertices of the mesh, sort the list of vertices according to their z position and select all the upper vertices. Project the vertices on a plane placed above the corrugation. Connect with a line the upper vertices and their relative projection. Rescale the lines using as base points the relative projected points. To rescale the wires gradually and following the shape of a function, use a “Graph Mapper” node. Then set up the Kangaroo nodes like explained in section 5.5.3 adding “Anchor”, “Load” and “Length(Line)” goal nodes attached to the Kangaroo “Bouncy Solver”. Before setting off the simulation set a Bezier function in the “Graph Mapper” and move all the control points on the top to make it shaped like a constant function $y=1$ so that the wires as the simulation begins are non-scaled. After

setting off the simulation change the function shape in the “Graph Mapper” moving the control points of the Bezier curve. As the points are moved, the wire length changes and the corrugation while being pulled down by the force of gravity and pulled up by the cables, finds a new equilibrium state changing its shape according to the lengths of the cables.

FABRICATION-AIMED DESIGNS

The case studies that we presented so far were all focused on the creation of patterns aiming to achieve a specific movement or aiming to a specific shape, but we always visualized and studied them through zero-thickness conceptual models. However, in the manufacturing and architecture fields, we may need to produce rigid-foldable mechanisms, or folded structures able to resist to certain loads. Rigid-folding and structural properties are strictly related to the stiffness of the material used, and as a matter of fact, the stiffness of any material is strictly related to its elasticity and to the area of its section (thus to its thickness). Thus, in this section we will study the origami-like mechanisms from another point of view, considering thickness, bending, forces, gravity, stability. In this chapter, we are going to present two case studies: a foldable ladder and a deployable chair. Both projects are originally designed, and we present them from start to finish focusing especially on the fabrication issues and on the possible solutions.

6.1. Known Origami Thickening Methods – State of the Art¹

A zero-thickness surface is a good approximation to model an origami-like geometry folded with paper. However, in manufacturing, architecture and engineering, is often necessary to use thick panels to enhance stiffness and rigidity of the folded mechanism. Most of the times the designers start modelling using zero-thickness surfaces, focusing only on the kinematics of the mechanism, but when they get to the prototyping phase, they cannot disregard the thickness. Thus, the problem that they face, is finding a way to add thickness to the zero-thickness surface without losing the original kinematics, preserving the DOF and avoiding self-intersections or collisions between the panels that would stop the folding motion. Many solutions have been already studied, proposed and used by many researchers. The most relevant known techniques are:

- “Offset panels” (Edmondson *et al.*, 2015; Lang *et al.*, 2018).
- “Hinge shifting”, also known as “Axis shifting” (Chen *et al.*, 2015; Lang *et al.*, 2018; Tachi, 2011b).
- “Tapered panels” (Tachi, 2011b).
- “Constant thickness attached panels” (Tachi, 2011b).
- “Membrane hinges” (Lang *et al.*, 2018; Zirbel *et al.*, 2013).
- “Rolling contacts” or “SORCE technique” (Lang *et al.*, 2017, Patent No. US 2017/0219007 A1; Lang *et al.*, 2018).
- “Strained joint” (Lang *et al.*, 2018; Pehrson *et al.*, 2016).
- “Double hinge” (Ku & Demaine, 2016; Lang *et al.*, 2018).
- “Symmetric Miura-ori vertex by shifted hinges and carved panels” (Hoberman, 1988, Patent No. 4780344; Lang *et al.*, 2018; Tachi, 2011b).
- “Slidable Hinges” (Lang *et al.*, 2018; Tachi, 2011b; Trautz & Künstler, 2009).
- “Double line” (Hull & Tachi, 2017).

6.1.1. Offset Panels

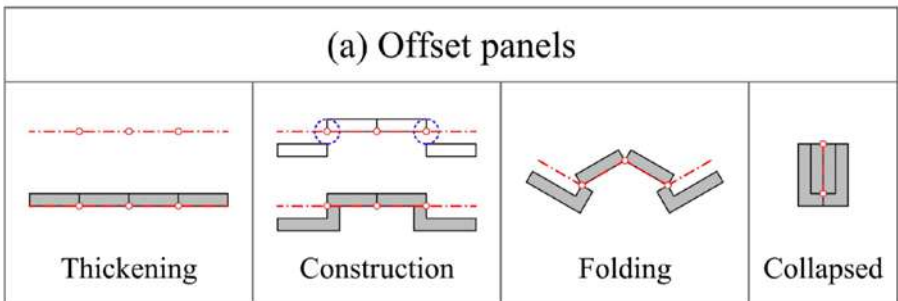


Figure 110 - “Offset panels” method.

The “Offset panels” method, theorised by Edmondson *et al.* (Edmondson *et al.*, 2015), is a versatile method that works with both flat-foldable and non-flat-foldable vertices. It allows the overlap of multiple layers, but it must be necessarily assembled, and the unfolded configuration does not lie on a plane. Furthermore, self-collisions are very common, and they must be checked carefully during the design phase. To avoid self-collisions, it may be necessary to carve out or cuts some parts from the faces, thus the mechanism may present holes which is not optimal for applications that require material continuity. Refer to section 6.2.3 for an example of this technique applied to a deployable ladder.

6.1.2. Hinge Shifting

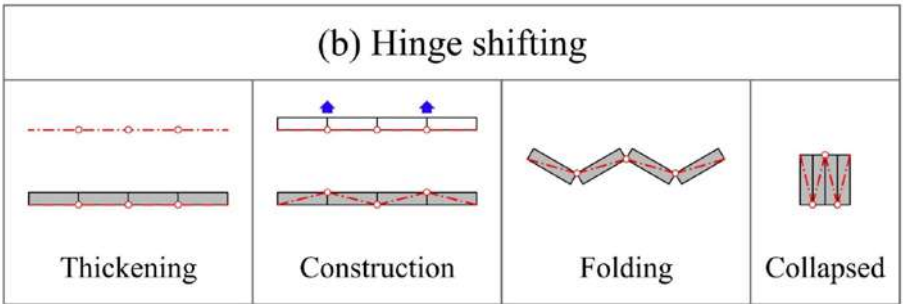


Figure 111 - “Hinge shifting” technique.

The “Hinge shifting” is one of the oldest known techniques for thickening origami, and it is cited by Lang *et al.* and Tachi in their reviews on thickening methods (Lang *et al.*, 2018; Tachi, 2011b), and extensively studied by Chen *et al.* (Chen *et al.*, 2015). This technique is easy to apply for a single linear crease, and it consists into moving the hinge on the valley side of the fold after thickened the panel. It may seem easy at a first glance, but it becomes trickier when it is used to solve patterns with internal vertices. Furthermore, the panels often need to be carved in some areas or fabricated with different thicknesses to be able to reach the perfectly flat-folded state. The dihedral angles between the adjacent faces are preserved, but because the hinges are shifted, the relative translations are not preserved, thus some holes generate in correspondence of the internal vertices when we start folding, and the DOF may change because we moved the creases out of plane.

6.1.3. Tapered Panels

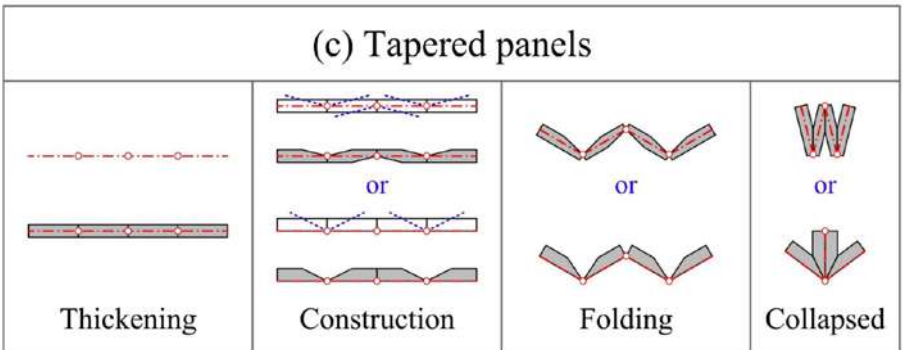


Figure 112 - “Tapered panels” method.

The “Tapered panels” method proposed by Tachi (Tachi, 2011b), is one of the best methods to preserve the original kinematics of the zero-thickness model, because it is easy to design and it preserves the position of the hinges, thus the thickened surface lies flat on a plane when unfolded. Furthermore, this method preserves the continuity of the material without generating holes. It can be built by marking or carving the crease lines from a single panel of the same material, with a specific tapering angle, which is equal to half the relative fold angle at the chosen partially-folded configuration. However, there are some major cons such as the fact that it is not possible to overlap more than two faces and the fact that the blocking creases cannot reach their maximum fold angle. The maximum collapsing amount can be regulated by changing the tapering angles of the panels, the smaller the tapering angles are, the smaller the collapsing amount is; the bigger the tapering angles are, the greater the collapsing amount is, and the thinner the panels become.

6.1.4. Constant Thickness Attached Panels

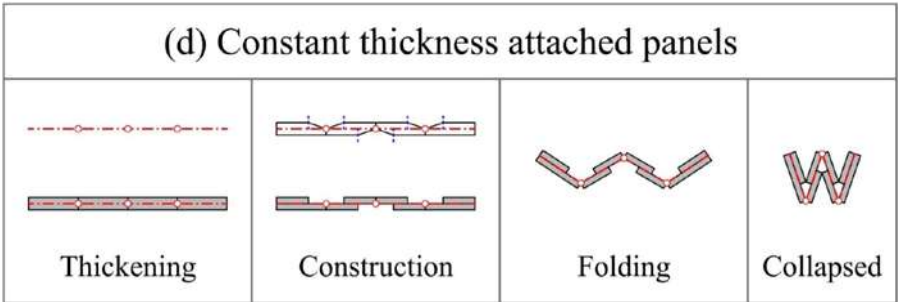


Figure 113 - “Constant thickness attached panels” method.

The “Constant thickness attached panels” is a direct consequence of the “Tapered panels” method demonstrated by Tachi (Tachi, 2011b). It consists into substituting each tapered panel by glueing together two non-tapered panels with a constant thickness equal to half the thickness of the substituted tapered panel. This is possible only when the top and bottom faces of the tapered panel overlap for a significant amount. The mechanism so configured can be produced, by sandwiching a strong fabric or film between the two shifted panels (not by folding or carving a single panel). This method is easier to fabricate by hand rather than the tapered panel method. That is why is often used for prototyping. However, it suffers the same problems of the “Tapered panels” technique, about the limited fold angle. Furthermore, the mechanism

at the folded state tends to work like a lever which tries to pull apart the film and the attached panels, thus the durability is strongly dependent on the film and glue quality.

6.1.5. Membrane Hinges

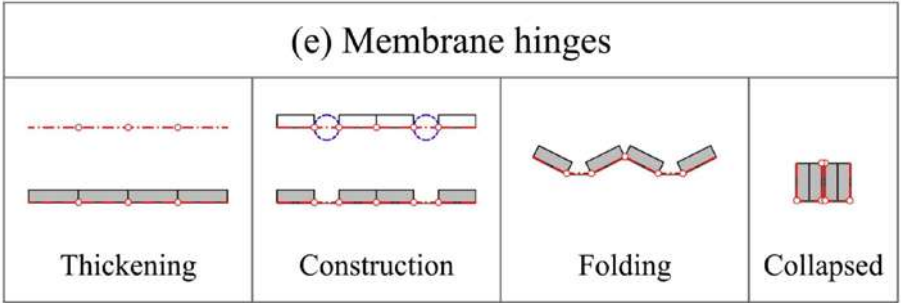


Figure 114 - “Membrane hinges” method.

The “Membrane hinges” technique explored by Zirbel *et al.* (Zirbel *et al.*, 2013) is easier to assemble and apply compared to other thickening techniques. However, with this method, the structure has a certain level of flexibility, because the faces are glued to a flexible membrane leaving a little space in correspondence of the valley (or mountain) creases equal to double the thickness of the panels (or even more in some cases), which makes the hinges loose and flexible. This technique has been proposed for structures that have to move in zero-gravity, where the stiffness is not as crucial as in other circumstances.

6.1.6. Rolling Contact or “SORCE” Technique

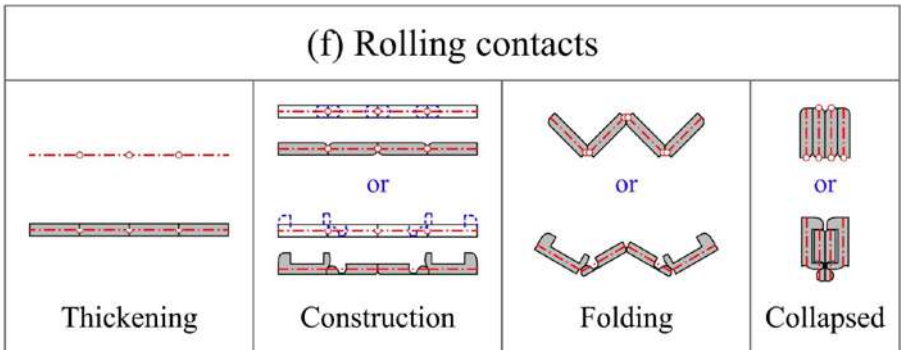


Figure 115 - “Rolling contact” or “SORCE” technique.

The “Rolling contacts” method, also known as “SORCE technique” has been demonstrated and patented by Lang *et al.* (Lang *et al.*, 2017, Patent No. US 2017/0219007 A1). It is a method inspired by Jacob’s ladder toy, and similar mechanisms have been used in spinal implants, robot fingers and prosthetic knee joints. Lang *et al.* in the paper *Thickness-Accommodation Techniques in Origami-Inspired Engineering* assert that “A notable aspect of the SORCE technique is that it marries a fully flat unfolded state [...] with a folded state incorporating arbitrary offsets between panels; furthermore, the DOF of the mechanism exactly reproduces the DOF of the zero-thickness model.” (Lang *et al.*, 2018).

However, like other thickening techniques, also this method has some cons such as the fact that flexible membranes are used to keep together the rolling contact hinges, thus the elasticity of the membranes may increase the unexpected deformations of the theoretical model. Furthermore, this kind of mechanisms must be assembled, and their robustness is strictly related to the assembling quality. Relating to this problem Lang also asserts that “While conceptually simple to implement, modelling flexible membrane hinges is considerably more complicated than mechanisms with discrete hinges. Also, the curvature and convexity of the rolling contact surfaces and the tolerances must be considered during design to ensure robust joints.” (Lang *et al.*, 2018).

6.1.7. Strained Joint

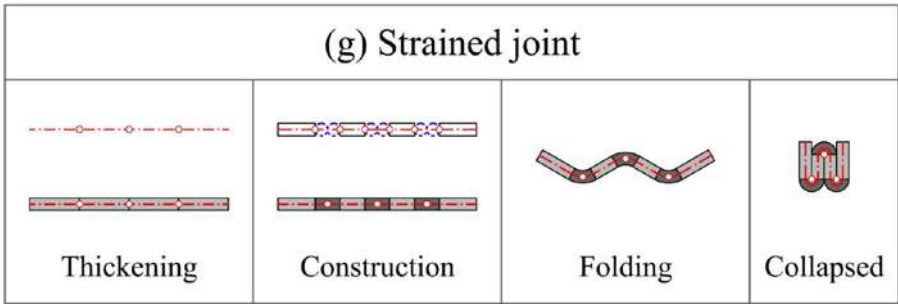


Figure 116 - “Strained joint” Technique.

The “Strained joint” technique, is a method demonstrated by Pehrson *et al.* (Pehrson *et al.*, 2016).

Lang *et al.* in the paper *A Review of Thickness-Accommodation Techniques in Origami-Inspired Engineering* assert that “The strained joint technique for accommodating thickness [...] is related to the membrane technique. Instead of using a thin membrane, the thick material itself acts as an effective membrane,

i.e., one in which the ‘fold’ is distributed across a region, rather than being a discrete revolute joint. In this case, the panel material itself is dissected so as to be flexible along desired hinge lines” (Lang *et al.*, 2018). The main con of this method is that we need to use different materials to realize a mechanism with strain-able joints while having rigid panels, which makes the fabrication harder than other methods that uses one single material. Instead of mixing different materials with different elasticity, there is a monolithic alternative that makes use of particular cuts in correspondence of the creases area. The system consists of dissecting the panels isolating some linked bars that can be flexed and twisted which makes the connections behaving like flexible joints. The larger the holes and the thinner the bars are, the more flexible will be the joint. However, if we chose to apply the dissection technique, we must use a flexible enough material, otherwise, the joints would break after a few folding cycles. This means that to be able to have flexible hinges also the faces would be a little bit flexible, thus the rigid-motion would not be perfect.

6.1.8. Double Hinge

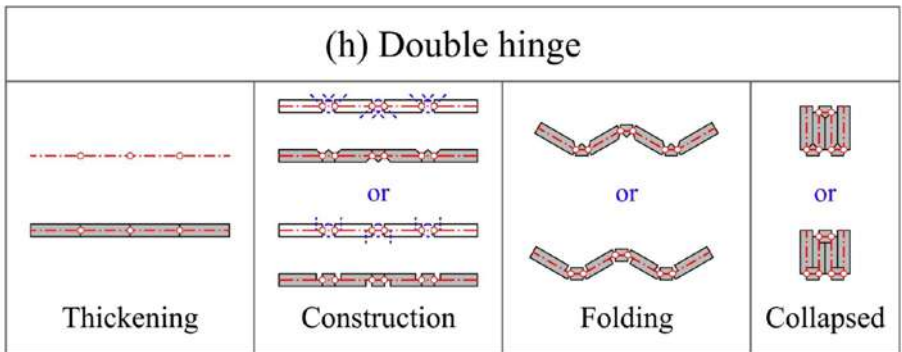


Figure 117 - “Double hinge” method.

The “Double hinge” method explored by Ku and Demaine (Ku & Demaine, 2016), is a method studied to thicken flat foldable patterns by doubling all the creases similarly to what happens in thick cardboard boxes. They proved that a double hinged solution for flat-foldable patterns with a single-vertex always exists, but the problem of thickening a flat-foldable pattern with multiple-vertices is still open. The main con of this method is that it needs holes to solve the internal vertices.

6.1.9. Symmetric Miura-ori Vertex by Shifted Hinges and Carved Panels

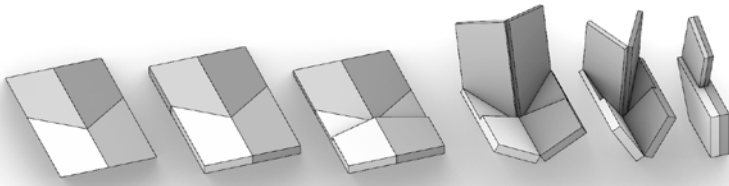


Figure 118 - “Symmetric Miura-ori vertex by shifted hinges and carved panels” method.

The method patented by Hoberman in 1988 (Hoberman, 1988, Patent No. 4780344) was cited by Tachi as a thickening method in “Rigid-Foldable Thick Origami” (Tachi, 2011b). It is a special application of the “Hinge shifting” method as explained by Lang *et al.* (Lang *et al.*, 2018). A limit of this method is that only symmetric Miura-like degree-4 vertices (bird’s foot vertices) can be thus accommodated.

6.1.10. Slidable Hinges

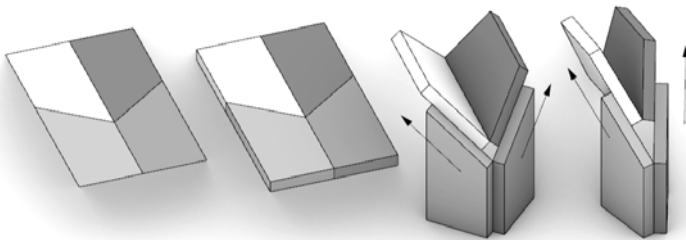


Figure 119 - “Slidable hinges” method.

The “Slidable hinges” method has been studied for the first time by Trautz and Künstler (Trautz & Künstler, 2009) on degree-4 vertices and it consists in sliding the faces longitudinally in relation to the adjacent faces along the common crease lines while folding the pattern. The more the panels slide, the more the collapsing amount increases. The main cons of this method are that it cannot be built with material continuity, it does not reach the completely collapsed state,

it generates holes during folding, and the technology to realize slidable hinges is very likely more complex and more subject to friction than traditional hinges.

6.1.11. Double Line

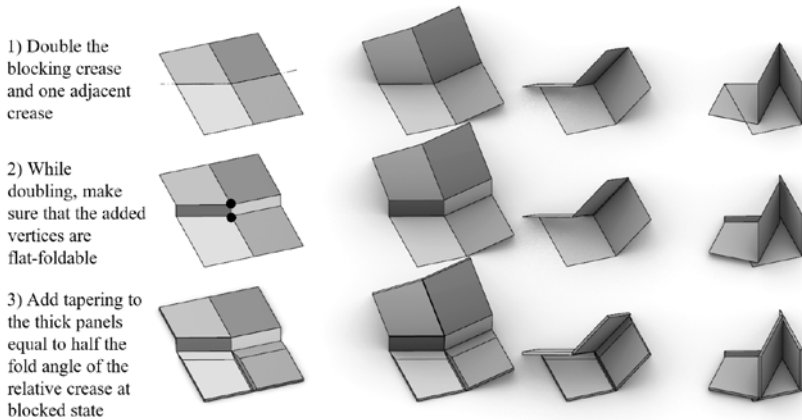


Figure 120 - “Double line” technique.

The last method we review here is the “Double line” method proposed by Hull and Tachi (Hull & Tachi, 2017). This method is probably the most versatile even if it is probably harder to apply than most of the other methods. The “Double line” method consists in offsetting some critical creases, mainly the creases that block and some incident creases, as in the “double hinge” method proposed by Ku and Demaine (Ku & Demaine, 2016), but with additional creased structures at the vertex. The additional creased structure at the vertex is a set of new flat-foldable vertices that solves the intersection of the doubled creases and the other non-doubled creases that converged in the same vertex. The new vertices are connected with a single polyline (which generates an additional planar face if it forms a closed loop). Once doubled those creases the faces can be thickened and tapered following the same approach used in the “Tapered panels” method. Doubling the correct creases allows to space the touching faces while keeping their relative orientation during motion, creating room for thickening the faces. The main pros of this method are that it can be produced by a monolithic single panel. The crease lines can be marked by carving or stamping as it happens in the “Tapered panels” method, but with lower tapering angles, which is better to preserve the thickness and stiffness of the panels. The collapsing amount is

not limited, and the relative rotation of adjacent faces is preserved. Also, all the mountain and valley creases can be marked from the same side of the panel with half-cuts if they are mountain and triangular-groves if they are valley. Traditional CNC machines or hot-stamping machines can be used to mark the creases and there is no need to build specifically designed machines helping to decrease the production costs. Furthermore, without any assembling phase, the product is ready to use. For all these reasons, this method is very suitable for industrial production. The main cons of this method are that it is harder to apply than most of the other methods we reviewed, and it preserves the kinematics of the mechanism only for what concerns the reciprocal rotation of the faces, not the translation. Furthermore, because every internal vertex split into multiple degree-4 flat-foldable vertices the DOF of the pattern may change to one. However, even if the translations are not preserved, no holes appear during motion, because the structure at the vertex of the split line has its own rigid cinematic that preserve material continuity. Lastly, because doubling the lines spaces the faces, to be able to achieve a result with the same dimensions of the original non-doubled model, the original faces must be resized and rearranged case by case after doubling the critical creases. In the following case studies, we tested different thickening method, but we ended up always using the “Double line” method by Tachi and Hull because, for our purposes, it was more efficient than other techniques.

6.2. Case Study – One-DOF, Developable, Non-Flat Rigid-Foldable Ladder²

The first proposed case study is a one-DOF, developable, rigid non-flat-foldable ladder. This design is an exemplification of a one-DOF thick structure with multiple blocking degree-4 vertices. The structure self-blocks (or self-arrests) when the blocking creases reach the maximum fold-angle of 180°. The main issues raised during the design phase were: making a one-DOF rigid foldable mechanism that self-blocks at a ladder-shaped configuration; dimensioning the steps correctly while preserving the developability of the ladder; thicken the panels without losing the original kinematics and making it self-supporting and self-balanced. All these issues influenced the design process from the early design stages and will be discussed further in the next sections, thus even if the generative process is presented as a linear sequence of steps, for clarity's sake, the reader should keep in mind that most of the stages of the process were de-

veloped in parallel, and they were redesigned and discussed several times before reaching the final iteration.

6.2.1. Preliminary Paper Prototype and Digitalization

When designing a new origami-inspired object often the first step is folding a paper mock-up, it helps materializing the idea and testing if it works as imagined. Also, it helps to make small changes to the initial pattern because, as long as the paper does not tear, the developability of the paper is preserved and guaranteed. This preliminary phase is crucial, and it helps the designer understanding fast how the global shape would look. However, because the paper model is elastic and flexible, it may mislead the understating of some formal or kinetic aspects, especially for what concerns the rigid foldability and the identification of the blocking configuration. Thus, it is recommended as from now, to convert the paper model into a much more accurate digital model. The preliminary digital model is not only useful to verify the rigid foldability and to make the first considerations about the shaping and folding motion, but it is the basic requirement for later accurate proportioning, and it is the fundamental starting point for the thickening phase.

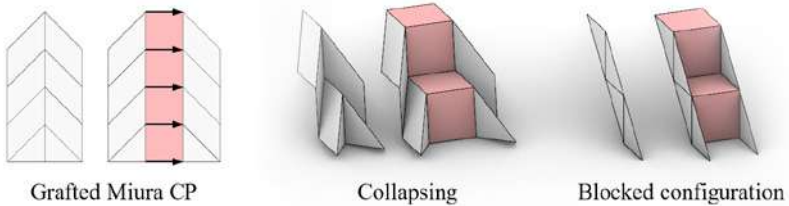


Figure 121 - Ladder concept by grafting a chain of three Miura-like vertices.

The concept of the ladder came from the grafting of a Miura-ori pattern as shown in Figure 121. Similar patterns are used in the aerospace industry for making ultra-light sandwich panels (Klett & Drechsler, 2011). The preliminary digitalized prototype of the ladder was designed by trial-and-error method. The model is very simple and because of that it is possible to get sufficiently accurate results using only reference points and simple graphical constructions. By trial-and-error method we tried to give a first rough shaping to the ladder. We angled the rises of the steps to make it steeper while keeping the treads long enough to fit a foot, and we dimensioned the steps to make the external corners aligned to the same oblique crease generated from the exceeding paper of the highest step as shown in Figure 122.

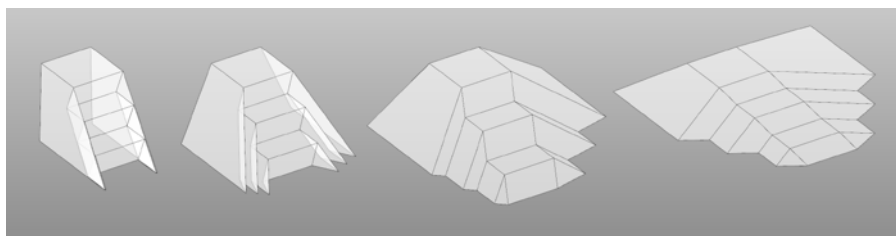


Figure 122 - The unfolding of the preliminary digitalized model of the ladder with angled rises.

6.2.2. Fine-Tuning the Dimensions of the Ladder with Trigonometry

Because the model is very simple, it is possible to get sufficiently accurate results by applying the trial-and-error method, only using reference points and simple graphical constructions. Although, to enhance even more the accuracy and to have total control on the shape and proportions of the model, we propose below a method based on simple mathematical formulations that allow the designer to get the wanted blocked state with fixed proportions without even needing to fold the CP. This method is a more accurate alternative of the trial-and-error method, but it may return similar results.

We start fixing some parameters according to the desired shape, refer to Figure 123:

- The angle ρ_4 between each riser and tread of each step is fixed ($\rho_4 = 80^\circ$).
- Each riser and tread must be rectangular ($\theta_4 = \theta_3 = 90^\circ$).
- The risers and the treads lengths (l) must be equal (to have the ribs $\overline{O2}$ and $\overline{65}$ aligned at blocked state).
- Each step height (h) is fixed (30 cm).

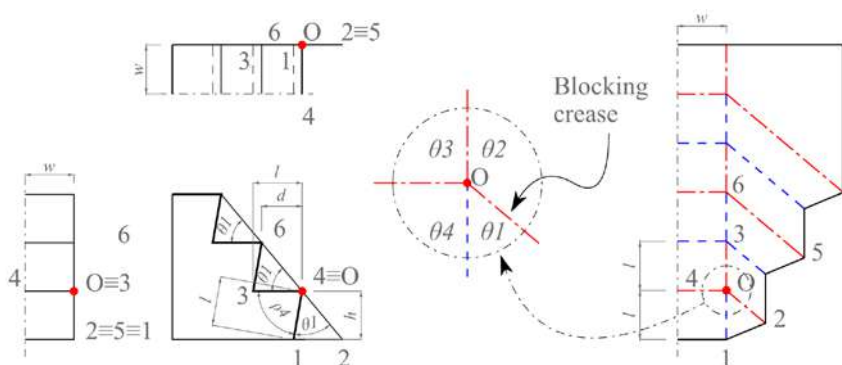


Figure 123 - Ladder design fixing some variables ($\theta_4 = \theta_3 = 90^\circ$, $\theta_2 = 180^\circ - \theta_1$, $h = 30\text{cm}$, $\rho_4 = 80^\circ$). θ_1 must be found.

Because the ladder has mirror symmetry, we can focus on one half of the pattern. The half pattern presents a chain of degree-4 vertices which is known to be a one-DOF mechanism (Tachi, 2011a). Also, because all the vertices have the same pattern, they all block at the same time. This allows us to focus only on one vertex (refer to vertex O in Figure 123). The creases $\overline{O1}$, $\overline{O3}$, and $\overline{O4}$ have fixed lengths and angles, thus the only crease that needs to be tuned is $\overline{O2}$, which is also the crease that blocks the degree-4 vertex. To find the $\overline{O2}$ direction, either θ_1 or θ_2 are needed. Because $\theta_1 + \theta_2 = 180^\circ$, we have $\theta_2 = 180^\circ - \theta_1$. So, finding θ_1 is all we need to solve the problem. The fixed parameters are the fold angle $\rho_4 = 180^\circ$, the angles $\theta_4 = \theta_3 = 90^\circ$, $\theta_2 = 180^\circ - \theta_1$, and $\theta_1 + \theta_2 + \theta_3 + \theta_4 = 360^\circ$. If we substitute these values into the equation 13 and rearranging it, we obtain the following simplified expression:

$$\theta_1 = \frac{180^\circ - \rho_4}{2}. \quad (33)$$

Once we know all the angles, we can calculate the riser and the tread length (l), fixing h and ρ_4 :

$$\overline{O1} = \overline{O3} = \overline{O4} = l = \frac{h}{\sin \rho_4} = \frac{30 \text{ cm}}{\sin 80^\circ} = 30.46 \text{ cm}. \quad (34)$$

$$\overline{O2} = 2(\overline{O1}) \cos \theta_1 = 2 \times 30.46 \text{ cm} \times \cos 50^\circ = 39.16 \text{ cm}. \quad (35)$$

It is also possible to calculate $\overline{O1}$, $\overline{O3}$, $\overline{O4} = l$ and ρ_4 fixing h and d using the Pythagorean theorem and trigonometry:

$$\overline{O1}, \overline{O3}, \overline{O4} = l = \frac{h^2 + d^2}{2d}. \quad (36)$$

$$\rho_4 = 90^\circ - \cos^{-1} \left(\frac{h}{l} \right). \quad (37)$$

6.2.3. Thickening – “Offset Panels” Method

Once defined the zero-thickness pattern we moved to the thickening phase. The first method we tested was the “Offset panels” method. The Figure 124 shows the steps that we followed to design the first thick version of the ladder. To be able to offset the panels properly we folded and unfolded continuously the ladder, so we were able to visualize and measure better the correct offset distances and position

of the vertical connections of the joints. This method is convenient because it allows the designer to animate the thickened surface (Figure 125) by matching the transformations of the original zero-thickness animated folded surface. This is possible because the kinematics of the faces is perfectly preserved after thickening. However, due to the multiple overlapping layers, we had to cut numerous parts in correspondence of the hinges to avoid collisions, this problem caused the physical prototype to have much floppier behaviour than expected. Furthermore, the overlapping layers forced us to double the offset distance of the panel for every step, this caused the unfolded model to be too thick and not very useful for actual applications. For all these cons, we had to move to another thickening method.

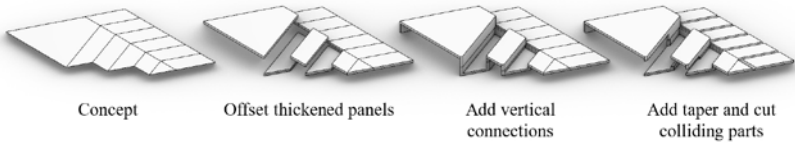


Figure 124 - Thicken the ladder with “Offset panels”.

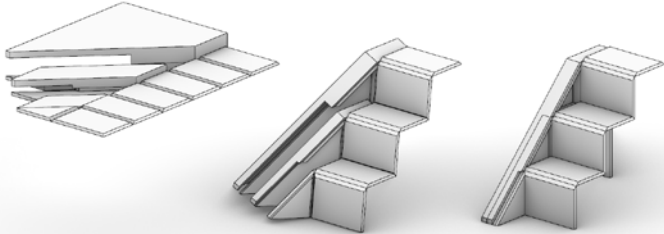


Figure 125 - Folding of the ladder thickened with “Offset panels” technique.

6.2.4. Thickening – “Tapered Panels” Method

The next thickening method we tested was the “Tapered panels” method. This method is convenient because it does not require to move the faces or the creases. The method consists into extruding the faces in one direction (perpendicular to the plane of the unfolded model) and taper the panels along the bisecting planes, relative to each pair of adjacent faces, at the blocked configuration. So far, everything appears easy and straightforward, however, at the blocked configuration, there are faces that touch, and because the bisecting plane of two touching

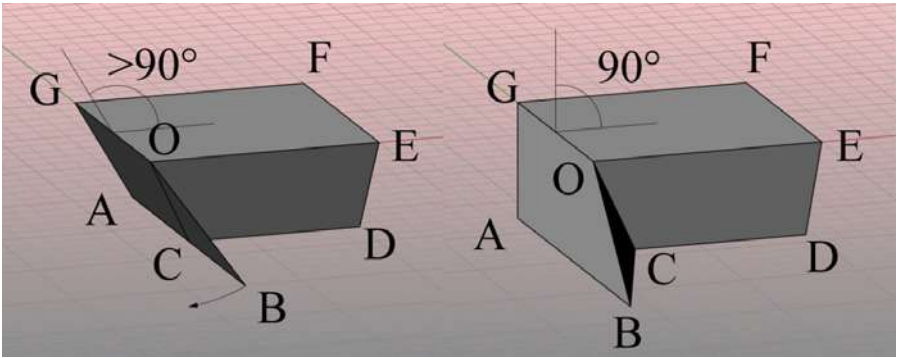


Figure 126 - Correcting the pattern to make the top and the side faces blocking at a fold angle greater than 90° .

faces lies into the same plane of the two faces themselves, we cannot add thickness to those faces at their maximum fold angle. So, we must stop folding before reaching the blocking configuration so that the faces are spaced of a certain amount allowing us to thicken and taper the faces without self-intersections or excessive tapering angles. Stopping the folding before reaching the blocking configuration, however, will prevent the model to reach the three-dimensional configuration that we designed in the first place. Thus, to be able to apply this thickening technique we must change a little bit the pattern to be able to reach the wanted three-dimensional configuration before reaching the blocked state. In Figure 126 you can see how we corrected the pattern of each vertex. With this new pattern, the face on the side reaches the vertical position before reaching the blocking configuration so that there is room to add a certain thickness between the faces ABOG and BCO. Then, we corrected the whole pattern according to the single-vertex test, and we folded it to reach the semi-folded configuration. After that, we thickened the panels by extruding them along the direction of their normal vectors to the inside, and we tapered them slicing the extruded panels with the bisecting planes between each pair of adjacent faces as shown in Figure 127.

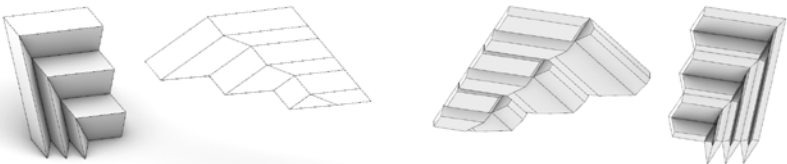


Figure 127 - Thickened ladder from the bottom, with tapered panels method.

This method is efficient and reliable; however, the adjacent panels are contacting only along the faces of the panels that are resultant from the tapering. Thus, the new blocked configuration is less stable than the previous blocked configuration because the contacting areas are smaller, and this may generate additional stresses on the hinges.

We may find a more stable blocked configuration by spacing for a smaller amount the blocking faces while keeping the same thickness of the panels, or by keeping the same semi-folded configuration and increasing the thickness. In both ways, after tapered the panels, the contacting area would be larger. However, such type of tapered panels is harder to fabricate, because traditional folding machines are not designed to taper the panel on such a wide area. It may be possible to fabricate the ladder with this thickening solution by tapering the panels one by one by CNC milling and glueing them onto a thin membrane or film, but because we wanted to limit the assembling, we decided to test a different thickening method.

6.2.5. Thickening – “Double Line” Method

The last thickening method we applied was the “Double line” method by Hull and Tachi. In the original approach that they propose, all the creases are doubled and the internal vertices resulting from the doubling process are forced to be always flat-foldable, in this way there is only one possible solution for any flat-foldable CP. However, the ladder is a non-flat-foldable special case, thus we discovered that there is no need to double all the creases to be able to add thickness while preserving the maximum fold angle. Furthermore, to increase the design freedom, we did not constrain the internal vertices to be flat-foldable.

The ladder has mirror symmetry and even if all the vertices have the same CPs, some of them have inverted mountain\valley assignments. Thus, there are only two types of degree-4 non-flat-foldable vertices. If we add thickness to the panels downwards, we can solve the vertices with the valley blocking fold simply applying the “Tapered panels” method, thus we focused only on the vertex with the mountain blocking fold as shown in Figure 128.

We started doubling the mountain blocking crease, and after that, we solved the intersections with other existing creases by doubling only one of the other creases. In Figure 128 we illustrated two possible solutions with two doubled creases. Nevertheless, the ladder has multiple vertices connected, thus the illustrated solution where the creases 01 and 04 are doubled, is not applicable, because to continue the pattern of the ladder we should attach the crease 04 to the crease 02 of the adjacent molecule, which are not compatible because they

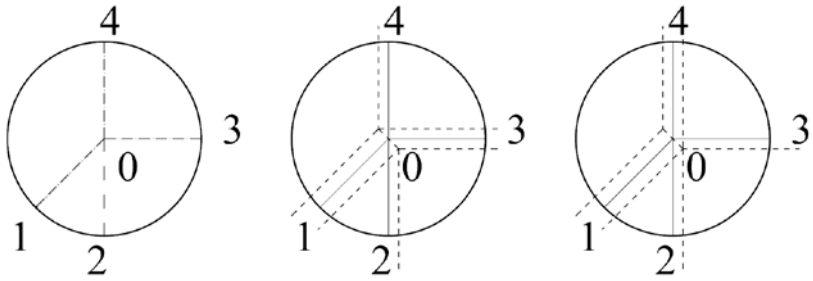


Figure 128 - Valid “Double line” solutions of a degree-4 vertex when the crease 01 is mountain and blocks.

are respectively a doubled and non-doubled crease. Thus, the solution where the creases 01 and 03 are doubled is the only valid solution in this case. After we decided which were the creases that needed to be doubled and because we chose to not constrain the internal vertices to be flat-foldable, we had to choose the offset distance and the angle of the crease connecting the two new vertices. In Figure 129 two possible solutions are illustrated. The creases AJ and BK are parallel, and their offset distance is equal in both cases. What it is changing are the angles $\theta_{j1,2}$ and $\theta_{k1,2}$, that make in the first case two non-flat-foldable vertices, and in the second case two flat-foldable vertices. Changing the internal angles of the vertices changes the distance of the parallel faces AHIJ and BCK at blocked state, thus the maximum allowed thickness of the panels also changes. The case with internal flat-foldable vertices is apparently more efficient because with the same offset we can get a result where there is more space between the faces AJHI and BCK, thus apparently, we can use thicker panels. However, the thickness of the face EFJK, which is one of the most stressed during use, is almost halved because of a higher tapering angle. Keeping tapering angles small makes the thickness more homogeneous, and thus also the loads are better distributed. Furthermore, with smaller tapering angles it is easier to be produced by CNC or folding machines. Moreover, in the first case the face EFJK is more perpendicular to the ground, thus it can transmit the load better to the bottom face.

Once designed the single vertex, we repeated the same structure to every equal vertex of the ladder. This method allowed us to preserve the relative rotations of the thick faces while keeping thin enough the unfolded configuration. However, the conversion from thin to thick model while keeping the same height and width of the steps was not trivial. After doubled the critical creases and solved all the internal vertices, we followed the process illustrated in Figure 130 to rectify the shape and dimension of the ladder matching the concept model dimensions.

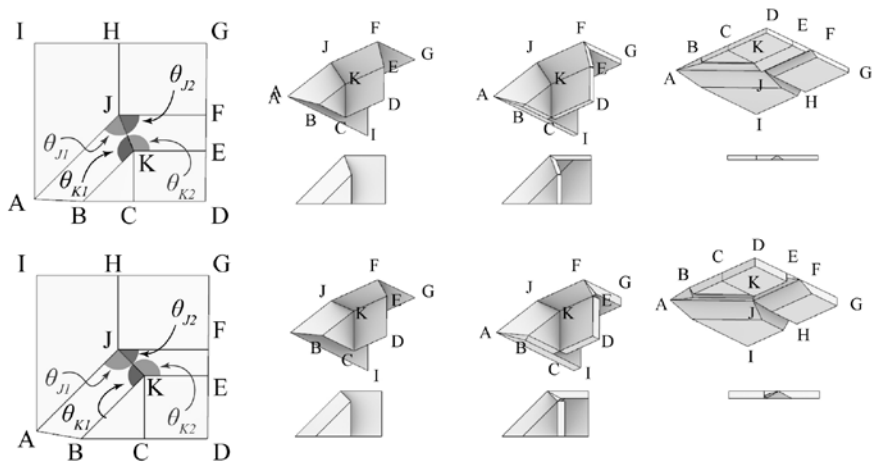


Figure 129 - Comparison of different internal vertices after doubling the internal critical crease in a single degree-4 corner.

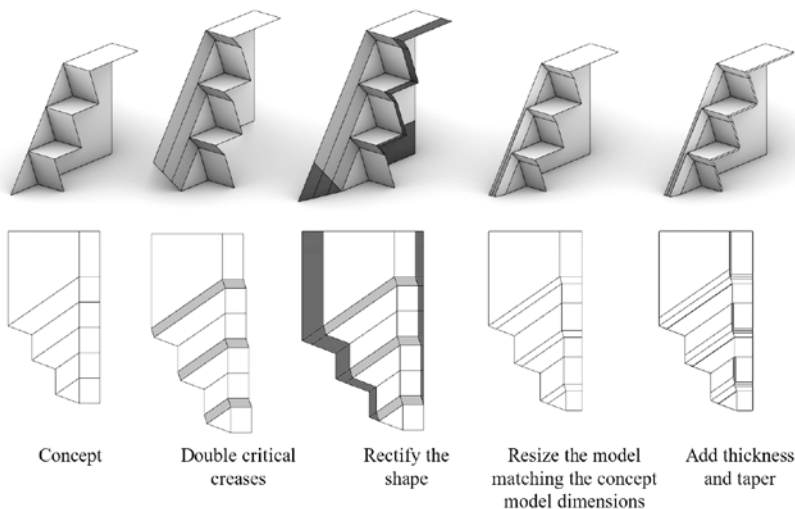


Figure 130 - Design process of the ladder, from the concept to the model with thick panels.

First, we identified the critical creases that we needed to double, and we added new strips of paper (light blue) to the surface following the angles defined previously on the single vertex. Then, because we expanded the surface and we added

some creases, the final dimensions and overall shape at folded state changed a little bit, thus we rectified it by adding missing parts and resizing everything to match the concept prototype as much as possible. Lastly, we added the thickness and we tapered the panels by bisecting the angles at folded configuration. As a final step before building the prototype, we simulated the folding and unfolding with thick panels. We checked and corrected possible problems like self-intersections, colliding panels, missing tapers, missing parts etc.

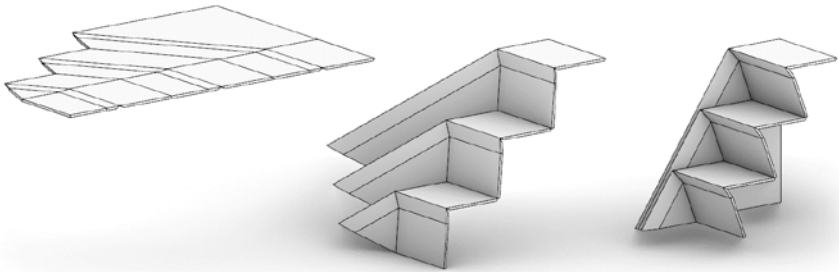


Figure 131 - Folding of the ladder thickened with “Double line” method.

We built the first prototype by 3D printing the tapered panels and assembling them by stitching them with common adhesive tape. After verified that it worked as expected, we built a full-size prototype by CNC milling wooden panels and gluing them on a thin membrane. The prototype gave us some hints about the stiffness of the structure and the possible problems that we will discuss in the next section.

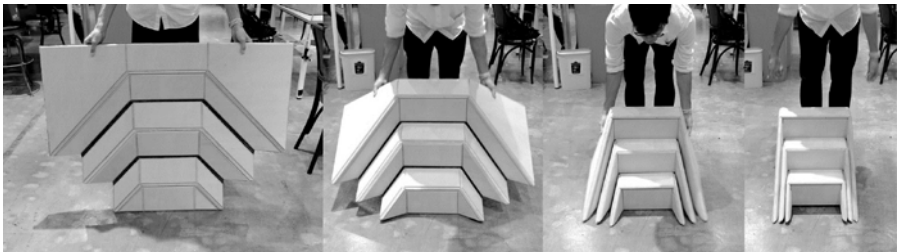


Figure 132 - Full-scale wooden prototype, folding test.

6.2.6. Stability Problems and Possible Solutions

The first problem we tried to solve was that the ladder tended to collapse when loaded. Because of its shape and its particular kinematics, surprisingly applying loads on the front two vertices of the first two steps did not interfere with the equilibrium state. However, when we applied two loads on the two back corners of the last step, it caused the collapsing of the whole ladder (as shown in Figure 133).

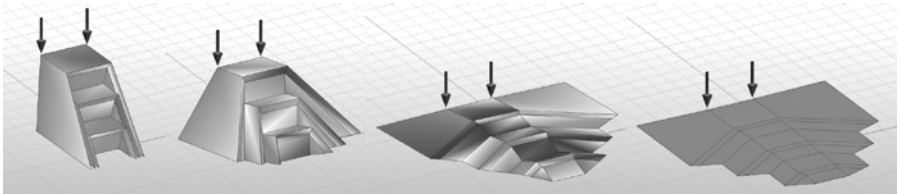


Figure 133 - Collapsing of the model of the ladder under given loads.

The three solutions in Figure 134 are possible locking systems that preserve the CP and prevent the collapsing of the ladder once loaded. All three systems: hooks, clips, and belt, are aimed to keep the ladder in position when loaded and they are all removable mechanisms which allow to fold and unfold the ladder fast. A different solution that helps keeping the ladder at blocked configuration is the solution proposed in Figure 135. This solution does not need any additional device but requires anchoring the ladder to a vertical support like a wall or a self-supported panel. This solution utilizes the gravity to keep the ladder at the folded configuration. It is based on the fact that anchoring the side panel distributes the loads in a way that when a user steps on it the ladder tends to fold and to lock. The ladder so anchored once reached the blocked configuration, will tend to keep it as long as we apply a vertical force from the bottom directed upward. Another possible solution may be aimed to change a bit the pattern to angle the side faces and make them converge toward the base so that the forces transmitted to the side faces will point inward and will help keeping the ladder closed, however, this solution may fail with elastic panels and it would decrease the area of the footing making it less stable.

Another issue, raised by the physical simulations, was a problem encountered while folding the ladder caused by the well-known pop-up and pop-down problem that we already mentioned in 4.8.3. This problem is critical, and it may cause the one-DOF mechanism to fold incorrectly or even to jam or break. This problem is caused by the fact that the degree-4 vertices have two possible folding modes where the mountain/valley assignment changes. At completely flat state there is no way to know whether a crease will start folding as a valley or a mountain crease. And

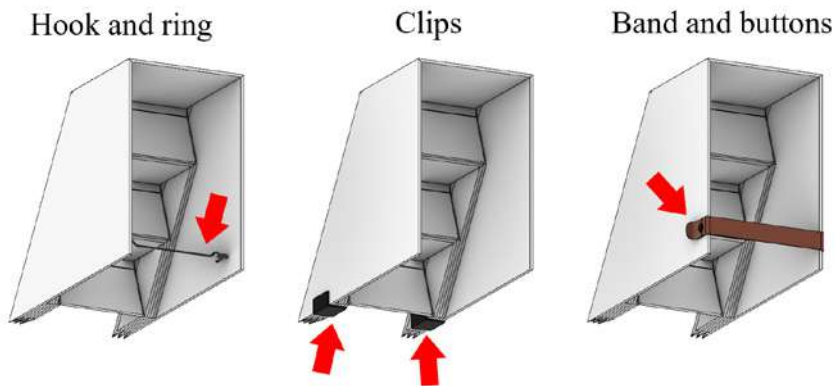


Figure 134 - Possible systems to keep the ladder at the blocked state even under load.

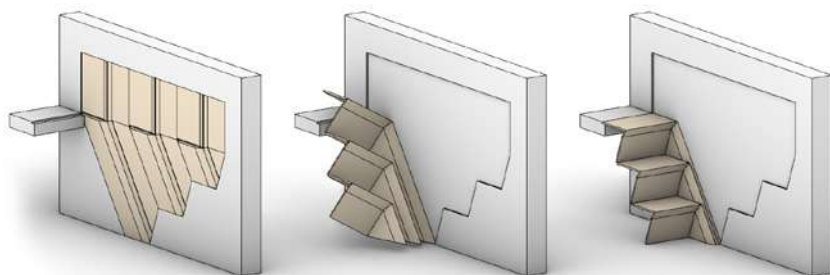


Figure 135 - Self-supporting ladder attached to a wall, it keeps the folded configuration by gravity.

even if a single crease starts folding with a wrong assignment it will cause a jam in the mechanism. A possible solution may be, for example, shaping the wall in a way that it would be impossible to unfold completely the ladder. This would solve the problem because there are no bifurcations in the motion except at the unfolded state, thus skipping the unfolded state will exclude the critical points where pop-up and pop-down problems occur. Another solution may be using non-developable non-flat-foldable vertices that skips the completely unfolded state making it impossible to change the pop-up or pop-down assignment, because a planar unfolded configuration does not exist into non-developable vertices. However, these solutions have been both excluded for portable ladder because not having a completely flat configuration would have compromised the portability and the stocking efficiency. This design, thus, is preferable for ladders with a small number of steps where the pop-up\pop-down assignment can be controlled manually.³

6.3. Case Study – One-DOF, Developable, Non-Flat Rigid-Foldable Chair⁴

Designing an origami-like chair, presents some issues similar to the ones encountered while designing the ladder, like the rigid foldability, the DOF, the proportions, the thickness and stiffness of the panels, the centre of gravity and the equilibrium conditions, but in this case also the aesthetics and the ergonomics must be taken into account. For simpler designs, it is possible to modify directly the unfolded CP being able to foresee the folded result easily without necessarily using complex mathematical formulations, as we did for the ladder. However, while designing a piece of furniture, especially when there is ergonomics involved, it would be better for designers to develop a system to control the shape at the blocked configuration directly in three-dimensions, while preserving the developability, without necessarily needing to work on the unfolded CP. Nevertheless, modifying the folded state without losing the developability may not be trivial, because moving a single vertex of the model, without particular precautions, will change the planar angles summation at the vertex making the model no more developable instantly. Therefore, in the next sections, we show the workflow we followed that allowed us to realize a one-DOF developable rigid-foldable self-blocking chair working directly in three-dimensional space while preserving the developability of the unfolded planar CP.

6.3.1. Preliminary Paper Prototype

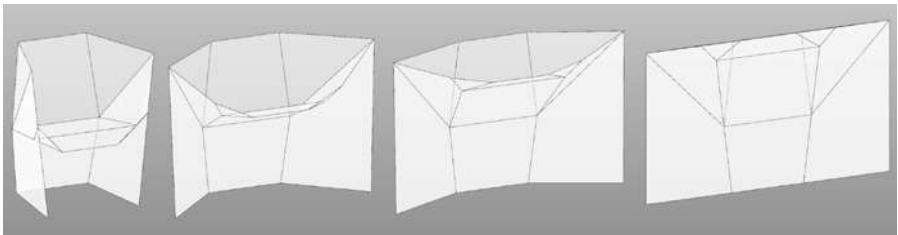


Figure 136 - The unfolding of the preliminary digitalized model of the chair.

In section 6.2.6 we saw that, if not correctly designed, a pattern may be not suitable to self-lock under certain loads conditions. For the ladder we solved that problem by glueing the side face to a wall so that we were able to achieve a result where once added loads on the steps, the faces tended to push one against the other making it block firmly; or we studied some other portable solutions involving additional locking systems like clips, hooks, and belts. However, to

simplify the usability of the chair, we tried to avoid external locking systems by exploiting the kinematics of the pattern itself and the force of gravity making it self-lockable (we will show later that in the first full-size prototype this was not achieved perfectly, in fact we had to stabilise the chair with external locking devices. However, we still think that a perfect self-locking chair is possible with a proper manufacturing process and selection of materials). Thus, this time, we considered this necessity from the early stages, and, after a certain number of paper test prototypes, we found a solution that apparently was working as wanted.

However, those models were not returning the results we expected, because even with very small forces the paper bent and the chair collapsed. Thus, to test the behaviour with rigid faces, we digitalized the model (Figure 136) we tested its rigid kinematics by folding and unfolding it with the same algorithm we presented in 4.8.1, then we added loads on the front vertices while keeping the faces planar and rigid as we did for the ladder. The simulation confirmed that the chair is very stable under vertical loads as shown in Figure 137, and as more loads, we add as more it tends to tighten at blocked configuration. We also tested the structural stability under rotating torsional movement around the Z-axis through the centre of gravity, but as far as the faces remain rigid the digital model of the chair does not collapse. Nevertheless, without properly rigid panels this perfect rigid behaviour may fail, thus the prototyping phase and the testing phase with different materials and thicknesses are crucial. A more specific analysis about the collapsing conditions under certain stresses, considering elasticity, and precise analysis of deformations may also be useful, but we decided to skip these digital simulations and test the resistance and the elastic deformations directly on the physical full-scale prototype.

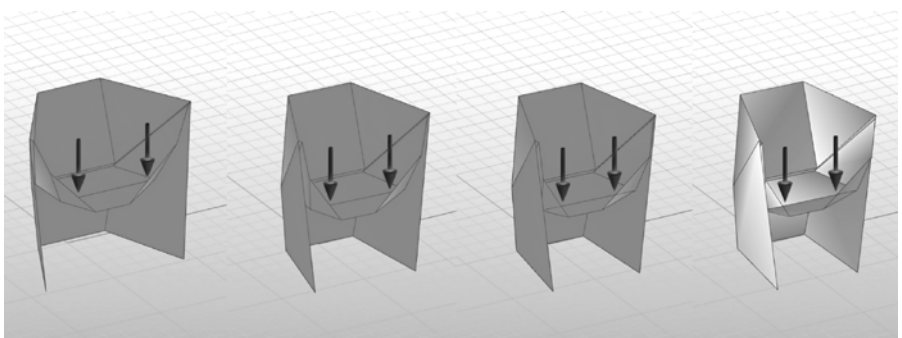


Figure 137 - Digital physical simulation of the chair under given loads; the more loads we add, the more the chair tightens at blocked configuration as far as the faces remain rigid.

6.3.2. Thickening – “Double Line” Method

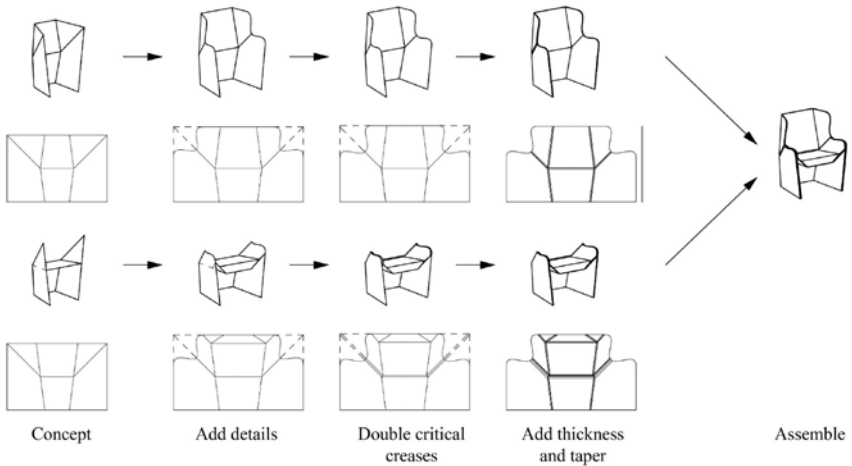


Figure 138 - Design process of the chair, from the concept to the model with thick panels.

To further enhance the ergonomics, to verify the stiffness and to correct the dimensioning of the chair we realized a prototype at human-scale. Thus, we converted the zero-thickness prototype to a thick prototype, and based on the experience with the ladder, we used the “Double line” method, which appeared to be the most suitable also for this application. The workflow we followed for thickening the chair is illustrated in Figure 138. In “Double line” method we need to double the critical creases and some connected creases, and the first prototype of the chair had only two blocking vertices in the seat, with only two blocking creases. Thus, we doubled the two blocking creases and we solved the two blocking vertices by doubling also the crease that connects them. Even if in this case it is not crucial, we designed the two resulting vertices as flat foldable vertices. The back part did not need any double crease because none of its creases blocks and the only double crease shared with the seat is one of the boundary creases of the glued faces, thus it is sufficient to offset that crease toward the unglued face to be able to not interfere with the back part. After doubled the creases, we refolded the model matching the original maximum fold angle of the model without double creases, and we added thickness to the panels. The maximum thickness of the panels that we can use without intersecting the panels, is half the distance between the blocking faces. However, the offset distance of the doubled creases in the CP does not correspond to the distance of the faces when doubled the creases, because the angle between

adjacent faces is different by 90°, as shown in Figure 139. Because of that, in order to calculate the offset distance, given the thickness of the panels, we need to calculate or measure the fold angles of the two doubled creases (Figure 140) and apply the simple trigonometry formulations reported below.

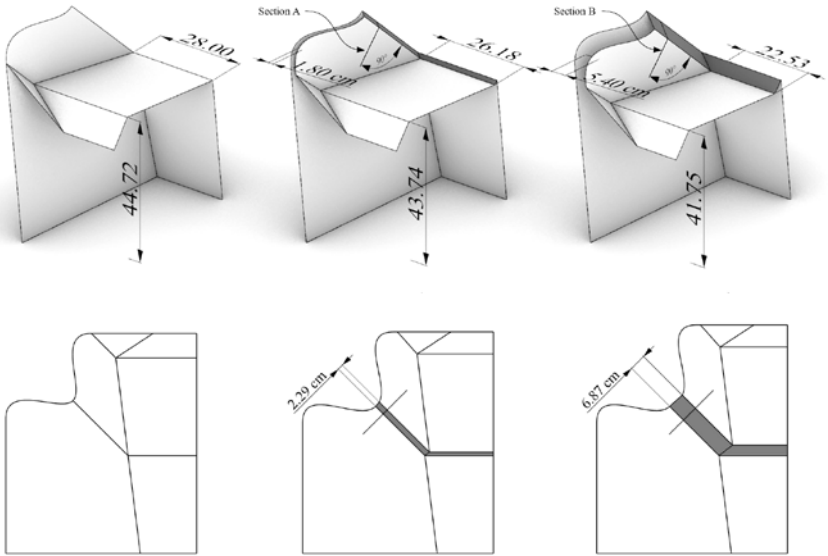


Figure 139 - Changing the offset of the double lines changes the space between the blocking faces.

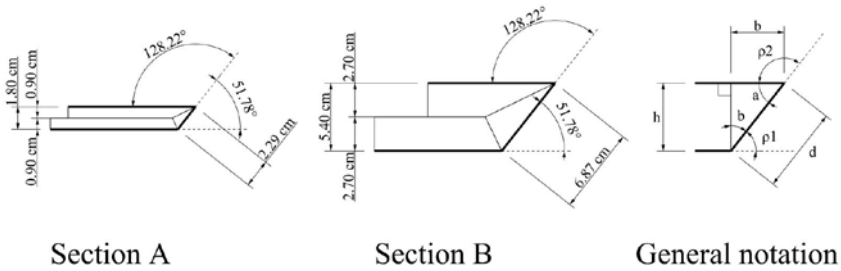


Figure 140 - Section A and B, and general notation.

Apply the following formulation to calculate the offset distance d given the thickness of the panels $\frac{h}{2}$ and the fold angles ρ_1 and ρ_2 :

$$d = \frac{h}{\sin a} \tag{38}$$

Where:

$$a = 180^\circ$$

$b = 2 \times$ thickness of the panel

$d =$ double line offset distance

Once doubled the critical creases with the correct offset, we thickened the panels and tapered them. The tapering angle is given by half the fold angle at blocked state. Before building a human scale prototype the project was tested with a 3D-printed 1:5 mock-up. The 3D-printed thick panels were glued to a sheet of plastic film (Tyvek) that functioned as a hinge between adjacent panels. After verified the correct behaviour of the small-scale model, a 1:1 prototype was built with a plastic sandwich panel with a thickness of 9 mm, the model was cut and folded with a CNC machine.

6.3.3. Blocked Degree-4 Vertex – From a Non-Developable Corner of 3 Faces

The tests we made on human-sized prototype highlighted some issues about the ergonomics, the weight, the portability, the stiffness and the size. We started correcting all these problems by re-modelling the zero-thickness digital model starting from the blocking vertices. To enhance the stiffness of the seat, we decided to change the pattern of the structure on the front making all the four vertices of the seat blocking at the same time, to do so we had to develop a method to work directly on the three-dimensional model while preserving the developability of the pattern.

The method we propose here is based on graphical constructions and a few trivial algebraic calculations. This method allows the designer to transform any non-developable corner of three faces into a developable degree-4 non-flat-foldable vertex. In this way, the designer can work directly on the three-dimensional folded model instead of working only on the unfolded CP. This method uses a zero-thickness mesh with planar faces, thus the thickness of the panels is not considered yet.

Any three-faces non-developable corner can be developed by cutting along any of the three edges splitting two faces. Thus, when we develop it, a gap will form between the two split faces.

Because we want to make it developable, we must fill that gap by extending one of the two split faces matching the bisector of the gap and adding an additional face in the remaining empty angle. Once refolded the three-faces corner the added face and the extended face will exceed outside the corner along the same plane of the extended faces. For each edge, there are only two possible

directions to which the exceeding part can be oriented. Thus, the possible developable degree-4 single vertex patterns, into which any three-faces corner can be transformed, are six in total as shown in Figure 141.

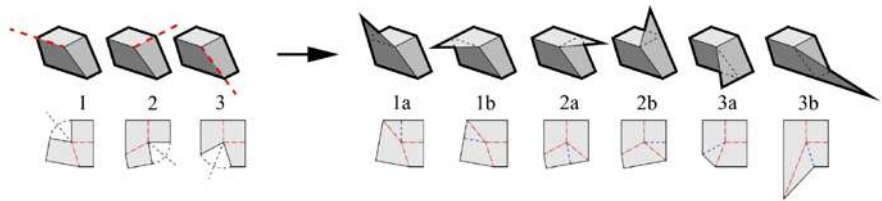


Figure 141 - On the left: all the possible developments of the same three-faces corner cutting along the 3 edges; on the right: all the possible degree-4 single vertex patterns without cuts.

To transform any non-developable corner of three faces into a developable degree-4 vertex by graphical approach, apply the following steps. (1) Chose the edge to be cut, of a given three-faces three-dimensional corner (three possible choices). (2) Extend one of the faces adjacent to the chosen edge (two possible choices). (3) Measure the total angle at the corner of the 3 starting faces, subtract that angle to 360° , and divide the result by two. (4) Draw a reference line along the chosen edge. (5) Rotate the reference line around the corner of the calculated angle (the rotation happens in the same plane of the extended face, with the centre of rotation in the corner). (6) Cut the extended face with the rotated reference line. (7) Draw an overlapped triangle on the outside triangular extrusion to close the loop of 4 faces. (8) This poly-surface made by 4 faces can now be developed in-plane without ripping, stretching or bending it. The proposed method has been implemented using Grasshopper (Rutten, n.d.)⁵ (Foschi, 2019). Applying this method to differently connected vertices allows the designer to obtain a CP where different multiple vertices block exactly at the same time.

6.3.4. Adjusting Shape and Dimensions to Improve Ergonomics and Stability

The method explained in the previous section (6.3.3) comes in handy to redesign the chair seat. Given that the chair has mirror symmetry, we focused only on one half of it, modelling only two of the four vertices of the seat as shown in Figure 142. The dimensions, the shape and the global proportions of the three-dimensional zero-thickness model were corrected according to the problems highlighted by the prototype at human-size. After redesigned all the four corners of the seat as blocking vertices, the perimeter of the pattern was not rec-

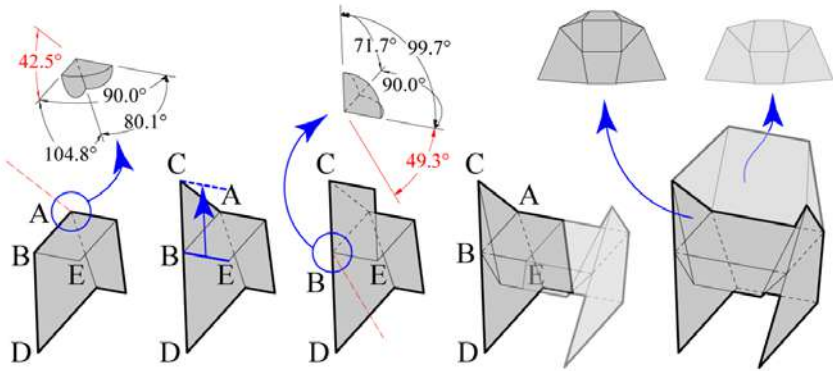


Figure 142 - Designing the chair seat with 4 symmetric degree-4 self-blocking vertices.

tangular, so with minor additional tweaking, we were able to restore the rectangular perimeter of the CP as shown in Figure 143 without losing the blocking four vertices. Lastly, we added some details like the hole for the handle, and the curve cuts for the armrests and we re-applied the thickening method based on “Double line” technique explained in 6.3.2.

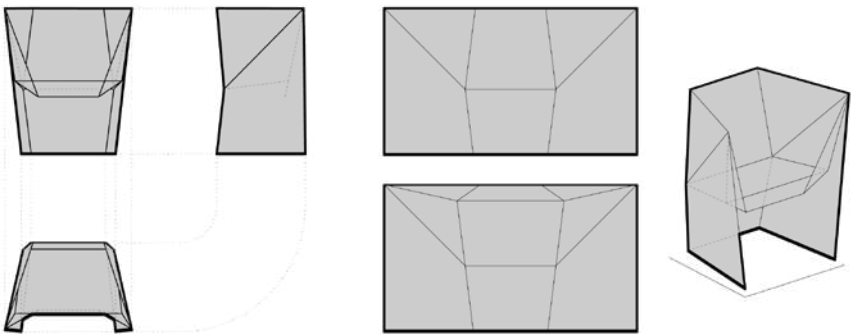


Figure 143 - Improved version of the foldable chair, all the degree-4 vertices of the seat block at the same time, and the perimeter of the unfolded CP is rectangular.

6.3.5. Human-Size Prototype of the Chair – Critical Observations

In Figure 144 we reported the final digital model which was subsequently produced by Kawakami Sangyo company by CNC cutting and attaching two plastic sandwich boards that they call “Plaperl”. This sandwich panel consists of two plastic boards with vacuum-formed cylinder in between. It has excellent rigidity

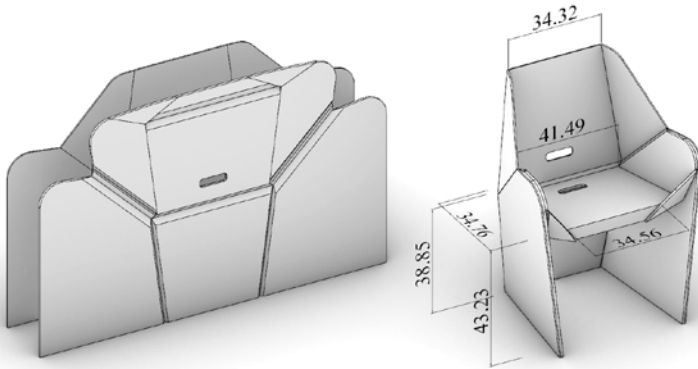


Figure 144 - Corrected design with better size, weight and ergonomics.

and lightweight and it is made of polyolefin that does not create toxic gases such as hydrogen chloride and dioxin. Thus, it is environmentally friendly and excellent in recycling (<http://www.putiputi.co.jp/en/>). However, this material may not be the optimal choice for what concerns the aesthetics, because once creased or cut it exposes the inside core which is not very appealing. The final physical prototype at human-scale is shown in Figure 145.

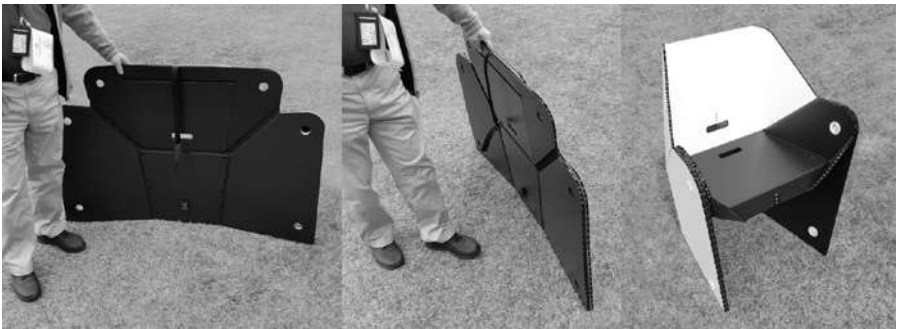


Figure 145 - Unfolded and folded chair prototype at human-scale.

The human-size prototype works almost as expected, but it has some critical points that should be studied further, discussed and improved. We report below some observations about the human-scale prototype that were also possible thanks to the comments of the users that examined the product.

- It has a unique original look that exposes the production process and folding function without losing the ergonomic shape and its elegant at-

tractive simplicity. However, the sandwich panel exposes the core section at the edges of the panels, and in some other areas, which is acceptable for study experimental models or temporary use, but not for a final product and long-term use. The research of a better material is still in progress, we are thinking to use a semi-rigid felt-based material which is light enough, comfortable, but still rigid, easy to crease and durable even if folded and unfolded several times.

- The structural stability of the chair at folded state is satisfactory, it has no tendency of instability in rotational movement around the vertical axis, but when loaded it tends to generate a local buckling at the midpoint of the front mountain fold in the seat. We expected this kind of elastic/plastic deformations in some areas of the chair, the prototype was also made to verify this kind of problems in all day use. To solve it we could use a thicker panel for the seat or add a reinforced ribbon where the deformations occur.
- The minimal thickness when unfolded is remarkable, however, the overall dimension at flat state is three times larger than a typical foldable chair with armrests. To solve this problem the pattern may be improved by adding one or more transversal creases that activate only at flat state to be able to fold it in half once flattened. An additional crease in the middle may help to avoid the buckling effect mentioned earlier, however, it may also compromise the stiffens at folded state, more tests are needed before drawing any conclusions.
- The panels used are not perfectly rigid, thus the bending of the faces causes the chair to not behave like a perfect rigid one-DOF mechanism. Unfortunately, we have no examples of realized origami mechanisms that behave like perfect one-DOF rigid mechanisms as in the simulations, thus a non-perfect rigid behaviour was expected. Filipov, Tachi and Paulino studied an approach to improve the preservation of rigidity during folding and unfolding by over-constraining the faces, glueing more folded sheets together with particular orientations to counterbalance the dynamic deformations (Filipov *et al.*, 2015). However, this approach, for now, has been only tested on tubular structures, which are not comparable to the chair at its actual version. Nevertheless, in this project, there are a small number of internal vertices, and the actuation of the folding and unfolding can be easily helped by the user even if it is not a perfect rigid one-DOF mechanism. Thus, we considered acceptable a non-perfect rigid behaviour for now. In addition, the chosen material, and the

manufacturing process are not optimal yet, thus improving these points would probably improve also the overall kinematics of the chair.

- Residual deformation tends to open the chair at flat state. This problem is probably also solvable changing the material, however, the residual deformation in this type of foldable furniture can be considered as an advantage because the memory of the material helps to avoid pop-up and pop-down problems at the beginning of the folding phase. We added a clip and a belt to keep the model flat, the same belt is used to stabilize the chair at the folded configuration.
- Wrong cut angles make the adjacent panels to push one against each other causing a resistance when almost reached the folded configuration. This is a crucial problem, the tapering of the panels is strictly related to the fold angle at folded configuration, thus a wrong tapering angle may cause instability, or it may cause the mechanism to block before reaching the final configuration. Fortunately, even if the CNC machine used is not designed to cut at any angle, we were able to achieve a satisfactory result exploiting the elasticity of the material. However, there is still room of improvement, because the pushing panels causes the chair to hardly keep the folded configuration and it causes a misalignment of some panels (especially in the armrests), we solved this problem by adding external locking devices as we did for the ladder design, which help to stabilize and lock the chair at the correct folded configuration.

CONCLUSIONS

In the last decades, in the field of architecture, furniture, manufacturing and fashion, origami was often taken as a reference for its functional and ornamental properties. The analysis of several projects inspired to origami, carried out in this book, reveals that in the field of permanent architecture not many of them uses origami as a reference for functional purposes, while the opposite can be observed in the other analysed fields. This may be caused by four main reasons: the lack of digital tools specifically designed for precise origami modelling; the fact that designing origami is hard especially at bigger scale or for kinetic applications where high accuracy is needed; the lack of specialized workforce that can build origami-like structures at architectonic scale; and the fact that kinetic architecture has higher costs of realization and maintenance than static architecture.

In this book we focused on trying to give solutions for the former two problems, which are a matter of drawing with the tools of descriptive geometry. Origami has endless possibilities, but the digital applications for designers have limited tools that can help designing foldable structures. We contributed to implement those tools by presenting a collection of algorithms, procedures, examples and case studies that the designers can use as starting points or as references for their projects. To do that we extensively investigated three main topics: solving the kinematics of given patterns from simple to complex; defining procedures to design a pattern while working directly on the folded 3D configuration guaranteeing that it remains developable while we manipulate it; and addressing some problems that may arise during fabrication, for example the thickening of the faces.

We studied these topics with the synthetic approach, which is the approach usually used by architects and designers, which is based on geometrical constructions. In most of the cases involving designing with origami, the synthetic method is highly preferable to the analytical method because it permits to visualise, and thus understand better and evolve procedures and results. Fur-

thermore, it is easier to be used by who do not have a background in mathematics or computer science, while still returning highly accurate results. To study the procedures presented, we used Grasshopper for Rhinoceros, because it is one of the most used tools that architects and designers already use for parametric and computational modelling. This node-based tool is not the only tool capable to perform such operations, in fact specific applications for designing digital origami exist, however they usually absolve very specific tasks which do not provide a flexible enough workflow for most projects. On the contrary, with the parametrical approach, the possibilities are endless, and we can establish relations between other digital objects in the same virtual environment, so that we can extract from them pieces of information (e.g. the distance from other buildings, the proportions and shape of the structure, the position of the sun and the direction of the light, the length of anchoring cables or support rails) and we can use them as design constraints or as references and inputs to generate the origami structure. We also highlighted the importance of working with software already used by professionals, because limiting the file conversions between different applications helps decreasing problems like incompatibility and data loss that may slow down the workflow.

With these premises, we made a catalogue of algorithms aimed to generate and solve the kinematics of specific rigid-foldable patterns, starting from the easiest cases (patterns with a single crease) up to more complex cases (patterns with multiple internal degree >4 vertices). These algorithms are a robust starting point for those designers who need to study the kinematics of their original patterns. We verified the validity, robustness and usefulness of the presented catalogue of generative algorithms by exemplifying their use into possible design workflows of buildings, furniture and objects, reproducing some existing projects. Thanks to the parametrical approach, the initial shapes, which mimicked the ones of the reference projects, evolved into different objects by just changing the input values.

Then, we focused on finding solutions for fabrication-related problems. In this context the case studies of the foldable chair and a foldable ladder allowed us to explore every generative step from start to finish, from the idea to the prototyping, passing from the study of kinematics, stability, flexibility, anchoring, locking and finally thickening. We studied and tested many different known thickening methods, and we identified the most suitable one for our cases: the “Double line” method by Tachi and Hull.

Lastly, we highlighted the importance of designing by comparing the physical paper prototype with the digital model because both approximate some

aspects of the final thick-panels prototype. Both the physical and the digital models are not self-sufficient, and we must always compare them to limit the risk of misinterpretation and bad designing.

This is the content unravelled in this book, however a lot could still be said about the field of applied origami. Future works may focus on non-developable degree-4 vertices, because their characteristic of being single-flat-foldable or double-flat-foldable has not been extensively studied yet and might hide interesting properties to be exploited in many fields. It may open new possibilities to solve the pop-up and pop-down problem in one-DOF mechanisms, and at the same time, degree-4 non-developable vertices can preserve or even improve the compactness of the folded and unfolded pattern while still having a self-blocking configuration (e.g. both the ladder and the chair designs could benefit from this type of vertices). Improving the chair and the ladder are works in progress and will eventually be mass-produced in the future.

For what concerns architecture it may be challenging and inspiring to further develop the field of responsive folded surfaces, as we did not explore it in depth in this book, and it is a major topic in the field of kinetic architecture. Another interesting ongoing research by our research group is about the topic of the elastic deformations and soft curve folds in digital folding simulations.

In conclusion, our wish is that this text inspired those researchers who are already into applied origami but also those designers that just approached this field while giving them the tools to develop original origami-related projects more smoothly and efficiently.

Notes

Chapter 3. Definitions and Theorems

¹ Some parts of this sections are also published in the paper *Designing Self-Blocking Systems with Non-Flat-Foldable Degree-4 Vertices* (Foschi & Tachi, 2018) written by the author of this book and the co-supervisor Tomohiro Tachi. The paper has been presented at the 7-OSME (The 7th International Meeting on Origami in Science, Mathematics and Education). The meeting took place in Oxford between 5th and 7th September 2018.

² Huffman himself defines it as “very difficult”.

Chapter 4. Constructive Methods for Solving the Kinematics of Origami

¹ To visualize the full grasshopper definition refer to the Appendix B.2. p. 166 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

² To visualize the full grasshopper definition refer to the Appendix B.9. pp. 171-172 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

³ To visualize the full grasshopper definition refer to the Appendix B.10. pp. 173-175 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

⁴ The reader may find confusing the fact that in this section we used a different notation compared to the one used in sections 3.5.2, 3.5.3, 3.6.1, although in the generative algorithms there are already a lot of nodes containing integer numbers, thus for the sake of clarity the use of numbers to indicate points it is not suggested. Furthermore, we used a rectangular perimeter instead of a disc perimeter because some of the algorithms that generates a single vertex will be used as building blocks to generate more complex patterns where the rectangular shape is preferable.

⁵ To visualize the full grasshopper definition refer to the Appendix B.14. pp. 178-180 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

⁶ To visualize the full grasshopper definition

refer to the Appendix B.18. pp. 183-184 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

⁷ To visualize the full grasshopper definition refer to the Appendix B.20. pp. 186-187 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

Chapter 5. Pattern Design from a Given Shape

¹ To visualize the full grasshopper definition refer to the Appendix C.1. pp. 189-190 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

² To visualize the full grasshopper definition refer to the Appendix C.2. p. 191 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

³ To visualize the full grasshopper definition refer to the Appendix C.3. p. 192 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

⁴ To visualize the full grasshopper definition refer to the Appendix C.4.1. p. 193 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

⁵ This section is excerpts from the paper *Conformation of a flexible Miura pattern on a double curvature surface* written by the author of this book. The paper has been presented at the AFGS 2017 (the 11th Asian Forum on Graphical Science). The meeting took place in Tokyo between 6th and 10th August 2017. The paper is part of the results of the research carried out during the PhD course (Foschi, 2017).

⁶ To visualize the full grasshopper definition refer to the Appendix C.5.2. pp. 197-198 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

⁷ To visualize the full grasshopper definition refer to the Appendix C.5.3. pp. 199-200 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

⁸ To visualize the full grasshopper definition refer to the Appendix C.5.4. pp. 201-202 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

Chapter 6. Fabrication-Aimed Designs

¹ The study of the thickening methods presented in this chapter was in a large part referenced to the paper *A Review of Thickness-Accommodation Techniques in Origami-Inspired Engineering* by Lang *et al.* (Lang *et al.*, 2018) and to *Considering Manufacturing in the Design of Thick-Panel Origami Mechanisms* by Crampton (Crampton, 2017); we extended the study of these methods and we commented them case by case.

² The ladder design is also published in the paper *Designing Self-Blocking Systems with Non-Flat-Foldable Degree-4 Vertices* written by the author of this book and Tomohiro Tachi, presented at the 7-OSME (The 7th International Meeting on Origami in Science, Mathematics and Education). The meeting took place in Oxford between 5th and 7th September 2018. The paper is part of the results of the research carried out during the period abroad encouraged by the PhD course (Foschi & Tachi, 2018).

³ The three-step ladder was further developed after these first iterations developed during my PhD thesis. Now is much more stable and does not require external locking devices to keep its folded configuration, however because it might become an actual product one day we cannot publish the result here yet.

⁴ The chair design is also published in the paper *Designing Self-Blocking Systems with Non-Flat-Foldable Degree-4 Vertices* written by the author of this book and Tomohiro Tachi, presented at the 7-OSME (The 7th International Meeting on Origami in Science, Mathematics and Education). The meeting took place in Oxford between 5th and 7th September 2018. The paper is part of the results of the research carried out during the period abroad encouraged by the PhD course (Foschi & Tachi, 2018).

⁵ To visualize the full grasshopper definition refer to the Appendix D.1. pp. 204-205 of the PhD thesis by the same author available at DOI:10.6092/unibo/amsdottorato/8871.

REFERENCES

- Abel, Z., Cantarella, J., Demaine, E. D., Eppstein, D., Hull, T. C., Ku, J. S., Lang, R. J., Tachi, T. (2016). Rigid Origami Vertices: Conditions and Forcing Sets. *Journal of Computational Geometry*, 7(1): 171-184. <https://doi.org/10.20382/jocg.v7i1a9>.
- Akitaya, H. A., Cheung, K. C., Demaine, E. D., Horiyama, T., Hull, T. C., Ku, J. S., Tachi, T. (2016). Box Pleating is Hard. In: *Discrete and Computational Geometry and Graphs*. JCDCGG 2015. Lecture Notes in Computer Science, vol. 9943. Cham: Springer, pp. 167-179. https://doi.org/10.1007/978-3-319-48532-4_15.
- Autodesk (n.d.). Dynamo. Accessed February 4, 2019, from <https://dynamobim.org/>.
- Bateman, A. (n.d.). Tess. Accessed September 6, 2016, from <http://www.papermosaics.co.uk/software.html>.
- Belcastro, S. M., Hull, T. C. (2002). Modelling the folding of paper into three dimensions using affine transformations. *Linear Algebra and Its Applications*, 348(1-3): 273-282. [https://doi.org/10.1016/S0024-3795\(01\)00608-5](https://doi.org/10.1016/S0024-3795(01)00608-5).
- Bhooshan, S. (2015). *Interactive Design of Curved-Crease-Folding*. Master Thesis, University of Bath, Department of Architecture & Civil Engineering.
- Bhooshan, S. (2016). Upgrading Computational Design. *Architectural Design*, 86(2): 44-53. <https://doi.org/https://doi.org/10.1002/ad.2023>.
- Bhooshan, S., Bhooshan, V., El-Sayed, M., Chandra, S., Richens, P., Shepherd, P. (2015). Applying dynamic relaxation techniques to form-find and manufacture curve-crease folded panels. *Simulation*, 91(9): 773-786. <https://doi.org/10.1177/0037549715599849>.
- Brandt-Olsen, C. S. (2016). *Calibrated Modelling of Form-active Structures*. Master Thesis in Architectural Engineering, The Technical University of Denmark.
- Buffart, H., Hoffmann, S., Paris, J., Siebrecht, J., Corves, B., Trautz, M. (2017). Non-flat folding mechanisms for structural purposes. *Proceedings of IASS Annual Symposia*, 2017: 1-8 (IASS 2017 Hamburg Symposium: Deployable Structures & Origami).
- Buffart, H., Hoffmann, S., Paris, J., Weigel, C., Siebrecht, J., Corves, B., Trautz, M. (2018). Folding Mechanisms with Discriminate Extremal Configurations for Structural Purposes. In: *Origami 7, The Proceedings from the 7th International Meeting on Origami in Science, Mathematics and Education*, vol. 3, *Engineering One*. St. Albans: Tarquin, pp. 685-697.
- Buri, H., Weinand, Y. (2008). ORIGAMI-Folded plate structures, architecture. In: *10th World Conference on Timber Engineering*, Miyazaki, Japan, 2-5 June.

- Cardone, V. (2017). *Gaspard Monge, padre dell'ingegnere contemporaneo*. Roma: DEI Tipografia del Genio Civile.
- Carlevaris, L., De Carlo, L., Migliari, R. (2012). *Attualità della geometria descrittiva*. Roma: Gangemi editore.
- Casale, A., Calvano, M. (2012). House of cards. The fold for the construction of articulated surfaces. *Disegnarecon*, 5(9): 289-300.
- Casale, A., Valenti, G. M., Calvano, M. (2013). *Architettura delle superfici piegate: le geometrie che muovono gli origami*. Roma: Edizioni K.
- Chandra, S., Bhooshan, S., El-Sayed, M. (2015a). Curve-Folding Polyhedra Skeletons through Smoothing. In: *Origami 6, Proceedings of the 6th International Meeting on Origami in Science, Mathematics and Education*, vol. I, *Mathematics*. American Mathematical Society, pp. 231-240.
- Chandra, S., Körner, A., Koronaki, A., Spiteri, R., Amin, R., Kowli, S., Weinstock, M. (2015b). Computing curved-folded tessellations through straight-folding approximation. In: *SimAUD '15, Proceedings of the Symposium on Simulation for Architecture & Urban Design*, pp. 152-159.
- Chen, Y., Peng, R., & You, Z. (2015). Origami of thick panels. *Science*, 349(6246): 396-400. <https://doi.org/10.1126/science.aab2870>.
- Crampton, E. B. (2017). *Considering Manufacturing in the Design of Thick-Panel Origami Mechanisms*. Master Thesis, Brigham Young University, Department of Mechanical Engineering.
- Demaine, E. D. (n.d.). .Fold file format. Accessed February 4, 2019 from <https://github.com/edemaine/fold>.
- Demaine, E. D., Demaine, M. L., Huffman, D. A., Hull, T. C., Koschitz, D., Tachi, T. (2016). Zero-Area Reciprocal Diagram of Origami. *Proceedings of IASS Annual Symposia*, 2016: 1-10 (IASS 2016 Tokyo Symposium: Spatial Structures in the 21st Century – Origami).
- Demaine, E. D., Demaine, M. L., Huffman, D. A., Koschitz, D., Tachi, T. (2015). Characterization of curved creases and rulings: design and analysis of lens tessellations. In: *Origami 6, Proceedings of the 6th International Meeting on Origami in Science, Mathematics and Education*, vol. I, *Mathematics*. American Mathematical Society, pp. 209-230.
- Demaine, E. D., Demaine, M. L., Huffman, D. A., Koschitz, D., Tachi, T. (2018). Conic crease patterns with reflecting rule lines. In: *Origami 7, The Proceedings from the 7th International Meeting on Origami in Science, Mathematics and Education*, vol. 2, *Mathematics*. St. Albans: Tarquin, pp. 573-589.
- Demaine, E. D., Demaine, M. L., Koschitz, D., Tachi, T. (2011). Curved crease folding: a review on art, design and mathematics. In: *Proceedings of the IABSE-IASS Symposium: Taller, Longer, Lighter (IABSE-IASS2011)*, London, September 20-23, 8 pp.
- Demaine, E. D., O'Rourke, J. (2007). *Geometric folding algorithms: linkages, origami, polyhedra*. New York: Cambridge University Press.
- Demaine, E. D., Tachi, T. (2010). Lecture 23 Video, Architectural Origami. Accessed October 1, 2018, from <http://courses.csail.mit.edu/6.849/fall10/lectures/L23.html>.

- Dias, M. A., Dudte, L. H., Mahadevan, L., & Santangelo, C. D. (2012). Geometric mechanics of curved crease origami. *Physical Review Letters*, 109(11): 114301. <https://doi.org/10.1103/PhysRevLett.109.114301>.
- Edmondson, B., Lang, R. J., Morgan, M. R., Magleby, S. P., Howell, L. L. (2015). Thick Rigidly Foldable Structures Realized by an Offset Panel Technique. In: *Origami 6, Proceedings of the 6th International Meeting on Origami in Science, Mathematics and Education*, vol. I, *Mathematics*. American Mathematical Society, pp. 149-161.
- Epps, G. (n.d.). Robofold. Accessed February 14, 2016, from <http://www.robofold.com/>.
- Epps, G. (2014). Robofold and Robots.Io. *Architectural Design*, 84(3): 68-69. <https://doi.org/10.1002/ad.1757>.
- Epps, G., Verma, S. (2013). Curved Folding: Design to Fabrication process of Robo-Fold. In: *Proceedings of SMI/ISAMA2013, Shape Modeling International 2013 and Fabrication and Sculpting Event: Thirteenth Interdisciplinary Conference of the International Society of the Arts, Mathematics, and Architecture*, pp. 75-83.
- Fallavollita, F. (2008). L'estensione del problema di Apollonio nello spazio e L'Ecole Polytechnique. *Ikhnos*, 2008: 29-42.
- Fallavollita, F., Salvatore, M. (2013). La costruzione degli assi principali delle superfici quadriche. *Disegnare Idee Immagini*, 46: 42-51.
- Feder, T. (2018). Q&A: Robert Lang, origami master. *Physics Today, People & History*, January 8. <https://doi.org/10.1063/PT.6.4.20180108a>.
- Filipov, E. T., Tachi, T., Paulino, G. H. (2015). Origami tubes assembled into stiff, yet reconfigurable structures and metamaterials. *Proceedings of the National Academy of Sciences (PNAS)*, 112(40): 12321-12326. <https://doi.org/10.1073/pnas.1509465112>.
- Foschi, R. (2017). Conformation of a Flexible Miura Pattern on a Double Curvature Surface. In: *AFGS 2017, Proceedings of the 11th Asian Forum on Graphic Science*, 10 pp.
- Foschi, R. (2019). Algorithmic Modelling of Folded Surfaces. Analysis and Design of Folded Surfaces in Architecture and Manufacturing. PhD thesis, Department of Architecture of Bologna University, a.a. 2018-2019. Available at: <http://amsdottorato.unibo.it/8871/>. Doi:10.6092/unibo/amsdottorato/8871.
- Foschi, R., Tachi, T. (2018). Designing Self-Blocking Systems with Non-Flat-Foldable Degree-4 Vertices. In: *Origami 7, The Proceedings from the 7th International Meeting on Origami in Science, Mathematics and Education*, vol. 3, *Engineering One*. St. Albans: Tarquin, pp. 795-809.
- Furuto, A. (2012). Resonant Chamber / rvtr. Accessed December 29, 2018 from <https://www.archdaily.com/227233/resonant-chamber-rvtr>.
- Gattas, J. M., You, Z. (2016). Design and digital fabrication of folded sandwich structures. *Automation in Construction*, 63: 79-87. <https://doi.org/10.1016/j.aut-con.2015.12.002>.
- Ghassaei, A. (n.d.). Origami Simulator. Accessed July 31, 2018, from <http://apps.amandaghassaei.com/OrigamiSimulator/>.
- Hatori, K. (2011). History of Origami in the East and the West before Interfusion. In: *Origami 5, Fifth International Meeting of Origami Science, Mathematics, and Education*. Natick, MA/Boca Raton, FL: A. K. Peters/CRC Press, pp. 3-11.

- Hoberman, C. S. (1988). Reversibly Expandable Three- Dimensional Structure. Patent No. 4780344. United States.
- Huffman, D. A. (1976). Curvature and Creases: A Primer on Paper. *IEEE Transactions on Computers*, C-25(10): 1010-1019. <https://doi.org/10.1109/TC.1976.1674542>.
- Huffman, D. A. (1977). Surface curvature and applications of the dual representation. In: *Computer Vision Systems*. New York: Academic Press, pp. 213-222.
- Hull, T. C. (n.d.). Tom Hull's page. Accessed February 4, 2019, from <http://mars.wne.edu/~thull/>.
- Hull, T. C. (2003a). Counting mountain/valley assignments for flat folds. *Ars Combinatoria*, 67: 175-187.
- Hull, T. C. (2003b). *Origami and Geometric Constructions: a comparison between straight edge and compass constructions and origami*. Accessed February October 10, 2018, from <http://mars.wne.edu/~thull/omfiles/geoconst.html>.
- Hull, T. C. (2006). *Project origami*. Natick, MA/Boca Raton, FL: A. K. Peters/CRC Press.
- Hull, T. C., Tachi, T. (2017). Double-line rigid origami. In: *AFGS 2017, Proceedings of the 11th Asian Forum on Graphic Science*, 13 pp.
- Justin, J. (1989). Résolution par le pliage de l'équation du troisième degré et applications géométriques. In: *Proceedings of the First International Meeting of Origami Science and Technology*, pp. 251-261.
- Kawaguchi, K., Ohsaki, M., Takeuchi, T., Liu, K., Paulino, G. H. (2016). MERLIN: A MATLAB implementation to capture highly nonlinear behavior of non-rigid origami. *Proceedings of IASS Annual Symposia*, 2016: 1-10 (IASS 2016 Tokyo Symposium: Spatial Structures in the 21st Century – Origami).
- Khademzadeh, H. R., Mazaheri, H. (2007). Some results to the Huzita axioms. *International Mathematical Forum*, 2(14): 699-704.
- Kilian, M., Flöry, S., Chen, Z., Mitra, N. J., Sheffer, A., Pottmann, H. (2008). Curved folding. *ACM Transactions on Graphics*, 27(3): 1-9. <https://doi.org/10.1145/1360612.1360674>.
- Klett, Y., Drechsler, K. (2011). Designing technical tessellations. In: *Origami 5, Fifth International Meeting of Origami Science, Mathematics, and Education*. Natick, MA/ Boca Raton, FL: A. K. Peters/CRC Press, pp. 305-322.
- Ku, J. S., Demaine, E. D. (2016). Folding Flat Crease Patterns with Thick Materials. *Journal of Mechanisms and Robotics*, 8(3): 031003. <https://doi.org/10.1115/1.4031954>.
- Kuribayashi, K., Tsuchiya, K., You, Z., Tomus, D., Umemoto, M., Ito, T., Sasaki, M. (2006). Self-deployable origami stent grafts as a biomedical application of Ni-rich TiNi shape memory alloy foil. *Materials Science and Engineering A*, 419(1-2): 131-137. <https://doi.org/10.1016/j.msea.2005.12.016>.
- Lang, R. J. (2003). *Origami design secrets: mathematical methods for an ancient art*. Natick, MA: A. K. Peters.
- Lang, R. J. (2008). The math and magic of origami. TED talk. ted.com. Accessed June 21, 2018, from https://www.ted.com/talks/robert_lang_folds_way_new_origami#t-156890.
- Lang, R. J. (2011). *Origami design secrets: mathematical methods for an ancient art*, 2nd ed. Boca Raton, FL: CRC Press.

- Lang, R. J. (2015a). *Air Bag Folding*. Self-Published. Accessed April 10, 2018, from <https://langorigami.com/article/airbag-folding/>.
- Lang, R. J. (2015b). *Origami and geometric constructions*. Self-Published, pp. 1-55. Accessed April 10, 2018 from http://langorigami.com/science/hha/origami_constructions.pdf.
- Lang, R. J. (2015c). *Treemaker*. Accessed April 10, 2018, from <https://langorigami.com/article/treemaker/>.
- Lang, R. J. (2016). *Huzita-Justin Axioms*. Accessed January 1, 2017, from <http://www.langorigami.com/article/huzita-justin-axioms>.
- Lang, R. J. (2018). *Twists, Tilings and Tessellations. Mathematical methods for Geometric Origami*, 1st ed. Boca Raton, FL: CRC Press.
- Lang, R. J., Howell, L. L., Magleby, S. P., Nelson, T. G. (2017). Non-planar closed-loop hinge mechanism with rolling-contact hinge. Patent No. US 2017/0219007 A1. United States.
- Lang, R. J., Magleby, S. P., Howell, L. L. (2016). Single-Degree-of-Freedom Rigidly Foldable Cut Origami Flashers. *Journal of Mechanisms and Robotics*, 8(3): 031005. <https://doi.org/10.1115/1.4032102>.
- Lang, R. J., Nelson, T., Magleby, S., Howell, L. (2017). Thick Rigidly Foldable Origami Mechanisms Based on Synchronized Offset Rolling Contact Elements. *Journal of Mechanisms and Robotics*, 9(2): 021013. <https://doi.org/10.1115/1.4035686>.
- Lang, R. J., Tolman, K. A., Crampton, E. B., Magleby, S. P., Howell, L. L. (2018). A Review of Thickness-Accommodation Techniques in Origami-Inspired Engineering. *Applied Mechanics Reviews*, 70(1): 010805. <https://doi.org/doi:10.1115/1.4039314>.
- Le Klint, P. V. J. (n.d.). The story of LE KLINT. Accessed July 19, 2018, from <https://www.leklint.com/en-GB/About-LE-KLINT/History.aspx>.
- Letellier, D. (2010). Tessel. Accessed November 29, 2018, from <https://www.davidletellier.net/TESEL>.
- Liu, K., Paulino, G. H. (2018). Highly efficient nonlinear structural analysis of origami assemblages using the MERLIN2 software. In: *Origami 7, The Proceedings from the 7th International Meeting on Origami in Science, Mathematics and Education*, vol. 4, *Engineering Two*. St. Albans: Tarquin, pp. 1167-1182.
- Loria, G. (1935). *Metodi matematici: essenza, tecnica, applicazioni*. Milano: Hoepli.
- McNeel (n.d.). Rhinoceros. Accessed August 20, 2010, from <https://www.rhino3d.com/>.
- Migliari, R. (2008a). Il problema di Apollonio e la Geometria descrittiva. *Disegnare Idee Immagini*, 36: 22-37.
- Migliari, R. (2008b). Rappresentazione come sperimentazione. *Ikhnos*, 2008: 11-28.
- Migliari, R. (2009a). *Geometria Descrittiva - metodi e costruzioni*, vol. 1. Novara: Città Studi edizioni.
- Migliari, R. (2009b). *Geometria Descrittiva - tecniche e applicazioni*, vol. 2. Novara: Città Studi edizioni.
- Migliari, R. (2012). Descriptive Geometry: From its Past to its Future. *Nexus Network Journal*, 14(3): 555-571. <https://doi.org/10.1007/s00004-012-0127-3>.
- Mitani, J. (n.d.). Jun Mitani's page. Accessed March 12, 2018, from <http://mitani.cs.tsukuba.ac.jp/en/>.

- Mitani, J. (2009). A design method for 3D origami based on rotational sweep. *Computer-Aided Design and Applications*, 6(1): 69-79. <https://doi.org/10.3722/cadaps.2009.69-79>.
- Mitani, J., Igarashi, T. (2011). Interactive Design of Planar Curved Folding by Reflection. In: *Pacific Graphics 2011 - Short Papers*. The 19th Pacific Conference on Computer Graphics and Applications. The Eurographics Association, pp. 77-81. Available at: <https://www.jst.go.jp/erato/igarashi/publications/001/PG2011.pdf>.
- Mitchell, D. (n.d.). *A brief outline of origami design history*. Accessed April 10, 2017, from <http://www.origamiheaven.com/origamidesignhistory.htm>.
- Miura, K. (1985). Method of Packaging and Deployment of Large Membranes in Space. *The Institute of Space and Astronautical Science Report*, 618.
- Miura, K. (1997). Fold - its physical and mathematical principles. In: *Proceedings of the Second International Meeting of Origami Science and Scientific Origami*, pp. 41-50.
- Pehrson, N. A., Magleby, S. P., Lang, R. J., Howell, L. L. (2016). Introduction of monolithic origami with thick-sheet materials. *Proceedings of IASS Annual Symposia*, 2016: 1-10 (IASS 2016 Tokyo Symposium: Spatial Structures in the 21st Century – Origami).
- Pentak, S., Roth, R., Lauer, D. A. (2013). *Design Basics: 2D and 3D*, 8th ed. Boston: Wadsworth.
- Piker, D. (n.d.). Kangaroo Physics. Accessed June 27, 2018, from <https://www.food4rhino.com/app/kangaroo-physics>.
- Resh, R. (1992). The Ron Resch Paper and Stick Film. Accessed April 10, 2017, from <http://www.ronresch.org/ronresch/gallery/paper-and-stick-film/>.
- Rutten, D. (n.d.). Grasshopper official page. Accessed July 31, 2018, from <http://www.grasshopper3d.com/>.
- Salvatore, M. (2012). *La stereotomia scientifica in Amédée François Frézier*. Firenze: Firenze University Press.
- Schmidt, P., Stattmann, N. (2009). *Unfolded: Paper in Design, Art, Architecture and Industry*. Berlin, Boston: Birkhäuser.
- Schwinn, T. (n.d.). Sandbox Topology. Accessed June 27, 2018, from <https://www.food4rhino.com/app/sandbox-topology>.
- Seymour, K., Burrow, D., Avila, A., Bateman, T., Morgan, D. C., Magleby, S. P., Howell, L. L. (2018). Origami-Based Deployable Ballistic Barrier. In: *Origami 7, The Proceedings from the 7th International Meeting on Origami in Science, Mathematics and Education*, vol. 3, *Engineering One*. St. Albans: Tarquin, pp. 763-777.
- Silverberg, J. L., Evans, A. A., McLeod, L., Hayward, R. C., Hull, T., Santangelo, C. D., Cohen, I. (2014). Using origami design principles to fold reprogrammable mechanical metamaterials. *Science*, 345(6197): 647-650. <https://doi.org/10.1126/science.1252876>.
- Stachel, H. (2015). Flexible Polyhedral Surfaces with Two Flat Poses. *Symmetry*, 7(2): 774-787. <https://doi.org/10.3390/sym7020774>.
- Tachi, T. (n.d.-a). Tachi's web Page. Accessed September 6, 2017, from <http://www.tsg.ne.jp/TT/>.
- Tachi, T. (n.d.-b). Tachi's Software. Accessed September 6, 2017, from <http://www.tsg.ne.jp/TT/software/>.

- Tachi, T. (2009). Generalization of rigid foldable quadrilateral mesh origami. *Journal of the International Association for Shell and Spatial Structures*, 50(3): 173-179.
- Tachi, T. (2010a). Freeform rigid-foldable structure using bidirectionally flat-foldable planar quadrilateral mesh. In: *Advances in Architectural Geometry 2010*. Vienna: Springer, pp. 87-102. https://doi.org/10.1007/978-3-7091-0309-8_6.
- Tachi, T. (2010b). Freeform variations of origami. *Journal for Geometry and Graphics*, 14(2): 203-215.
- Tachi, T. (2010c). Geometric Considerations for the Design of Rigid Origami Structures. *Proceedings of IASS Annual Symposia*, 2010: 1-12 (IASS 2010 Shanghai Symposium: Spatial Structures – Permanent and Temporary). Available at: https://origami.c.u-tokyo.ac.jp/~tachi/cg/DesignOfRigidOrigamiStructures_tachi_IASS2010.pdf.
- Tachi, T. (2011a). One-DOF rigid foldable structures from space curves. In: *Proceedings of the IABSE-IASS Symposium: Taller, Longer, Lighter (IABSE-IASS2011)*, London, September 20-23, 8 pp. Available at: <https://origami.c.u-tokyo.ac.jp/~tachi/cg/Rigid-FoldableStructureFromSpaceCurveTachiIASS2011.pdf>.
- Tachi, T. (2011b) Rigid-Foldable Thick Origami. In: *Origami 5, Fifth International Meeting of Origami Science, Mathematics, and Education*. Natick, MA/Boca Raton, FL: A. K. Peters/CRC Press, pp. 253-264.
- Tachi, T. (2013). Composite rigid-foldable curved origami structure. In: *Proceedings of the 1st International Conference on Transformable Architecture (Transformables 2013)*, Seville, September 18-20, 6 pp. Available at: https://tsg.ne.jp/TT/cg/CellularCurvedOrigami_Tachi_Transformables2013.pdf.
- Tachi, T., Epps, G. (2011). Designing One-DOF Mechanisms for Architecture by Rationalizing Curved Folding. In: *Proceedings of the International Symposium on Algorithmic Design for Architecture and Urban Design (AlgoDe Tokyo 2011)*, Tokyo, March 14-16, 14 pp. Available at: <https://origami.c.u-tokyo.ac.jp/~tachi/cg/RigidOrigami-CurvedFoldingTachiEppsALGODE2011.pdf>.
- Tachi, T., Horiyama, T. (2018). 1-DOF Pattern to Fold into Multiple Polyhedra. In: *Abstracts of the 7th International Meeting on Origami in Science, Mathematics and Education (7OSME)*, Oxford, September 5-7.
- Tachi, T., Hull, T. C. (2017). Self-foldability of Rigid Origami. *Journal of Mechanisms and Robotics*, 9(2): 021008. <https://doi.org/10.1115/1.4035558>.
- Tamasoft. (n.d.). Pepakura designer. Accessed September 6, 2016, from <http://www.tamasoft.co.jp/pepakura-en/>.
- Tedeschi, A. (2014). *AAD Algorithms-Aided Design. Parametric Strategies Using Grasshopper*, 1st ed. Brienza: Le Penseur.
- Terzidis, K. (2006). *Algorithmic architecture*. London: Routledge.
- Terzidis, K. (2009). *Algorithms for visual design using the processing language*. Indianapolis: Wiley Publishing.
- Thün, G., Velikov, K., Ripley, C., Sauvé, L., McGee, W. (2012). Soundspheres: Resonant chamber. *Leonardo*, 45(4): 348-357. https://doi.org/10.1162/LEON_a_00409.
- Tolman, K., Crampton, E., Stucki, C., Mayenes, D., Howell, L. L. (2018). Design of an Origami-Inspired Deployable Aerodynamic Locomotive Fairing. In: *Origami 7, The*

- Proceedings from the 7th International Meeting on Origami in Science, Mathematics and Education*, vol. 3, *Engineering One*. St. Albans: Tarquin, pp. 669-684.
- Trautz, M., Künstler, A. (2009). Deployable folded plate structures – folding patterns based on 4-fold-mechanism using stiff plates. In: *Proceedings of the IASS Symposium 2009, Valencia*, Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures. Valencia: Editorial Universitat Politècnica de València, pp. 2306-2317.
- WolframResearch. (n.d.). Mathematica. Accessed January 10, 2019, from <http://www.wolfram.com/mathematica-home-edition/>.
- Yin, S. (2009). *The Mathematics of Origami*. Accessed January 16, 2018, from https://sites.math.washington.edu/~morrow/336_09/papers/Sheri.pdf.
- Zirbel, S. A., Lang, R. J., Thomson, M. W., Sigel, D. A., Walkemeyer, P. E., Trease, B. P., Magleby, S. P., Howell, L. L. (2013). Accommodating Thickness in Origami-Based Deployable Arrays. *Journal of Mechanical Design*, 135(11): 111005. <https://doi.org/10.1115/1.4025372>.
- Zwierzycki M. (n.d.) Anemone. Accessed June 27,2018, from <http://www.food4rhino.com/app/anemone>.

GLOSSARY

Accordion:	A sequence of alternated mountain and valley creases.
Algorithm:	A process or a set of rules to be followed to reach an expected result.
Array:	Collection of elements values or variables identified by an index.
Asymmetric reverse fold:	A reverse fold which is not flat-foldable.
Base:	A folded geometrical shape that has a structure which simplifies the desired subject.
Bifurcated motion:	When there are two possible paths into the motion of a mechanism.
Blocking crease:	The crease that hits 180° first in a rigid-origami pattern and arrests the motion of the whole mechanism. They can be more than one (e.g. in flat-foldable patterns all the creases hit 180° at the same time).
Boolean value:	A value which is “1” or “0” or reciprocally “True” or “False”.
Box pleating:	A folding technique which allows only the use of creases multiple of 45° , they are usually built on a grid.
Branched list	A list of lists.
B-rep:	B-rep stands for “Boundary representation”. In solid modelling and computer-aided design, it represents a collection of connected surfaces which defines the boundary between solid and non-solid.

Chaotic type:	It is a family of folded surfaces which has a crease pattern characterized by an irregular mesh of creases.
Circle packing:	Placing circles on a surface so that they do not overlap.
Circle river method:	A folding technique that constructs the crease pattern by packing non-overlapping circles and rivers into the surface which is usually a square.
Closed sink fold:	A sink fold which locks after folding. It usually cannot be performed rigidly, it needs to exploit the paper flexibility of ten crumpling or forcing the point while pushing it inside.
Cluster:	A group of items nested into a single new item.
Collapse:	This term describes the action of folding a crease pattern all at once to form the folded base.
Corrugation:	A particular type of tessellation that has no triple or more layers overlapped. The entire original surface of the paper is usually visible. The most common corrugations are in form of a wave with alternated peaks and valley.
CP:	Crease Pattern.
Crease:	The mark that appears on the paper after folding and unfolding it.
Crease assignment:	Determination if a crease is mountain or valley.
Crease pattern:	The scheme of creases on a flat sheet that is necessary to fold a particular base.
Crimp fold:	A sequence of symmetric valley and mountain creases with respects of a central pre-existing crease.
Curved fold:	A fold that starts from a curved crease and exploits the flexibility of the material to configure the surface into a curved shape. It can be performed only with a flexible sheet of material unless the ruling is predetermined, thus the curve has to be discretized into a polygonal chain.
Degree-4 vertex:	A point inside a crease pattern where only four creases meet.

Degree >4 vertex:	A point inside a crease pattern where more than four creases meet.
Developable:	That can be unrolled/unfolded on a plane.
Dihedral angle:	The angle between two faces adjacent to the same crease, it is defined as the angle between the normal vectors of the faces.
DOF:	Degree of freedom.
Edge:	A single linear segment which is on the perimeter of a face or a sheet of paper.
Flap:	A region paper which is usually attached to the rest of the base by a single edge. It can be composed of one or more layer of folded paper.
Flat-foldable:	A pattern that can be folded in the plane.
Fold:	Acronym of “Flexible Origami List Data-structure” is a file format (with extension .fold) for describing origami models with meshes. Developed by E. D. Demaine, J. S. Ku and R. J. Lang.
Fold angle:	The angle between the limbs of a fold. Usually, it is measured by measuring the angle between the normal vectors of the faces adjacent to the crease.
Folding mode:	Way to rigid-fold a pattern with a specific mountain/valley assignment. Any pattern usually has more than one folding mode.
Generative algorithm:	A sequence of operations that generates a particular result.
Generatrix:	A moving point, line, or surface forming a line, surface, or solid.
Grafting:	Modifying a crease pattern by slicing it along existing creases and adding a strip of new paper in order to add new features.
Hex pleating:	A design technique similar to box pleating but that uses only angles multiple of 30° . It usually starts from a grid made by equilateral triangles.

Hinge:	A movable joint which connects adjacent faces or flaps.
Huzita-Justin axioms:	A set of rules related to the mathematical principles of paper folding. They explain the operations that can be made while folding a piece of paper in the plane.
Inside reverse fold:	A type of reverse fold which changes the direction of the tip of a flap keeping its layers inside the rest of the flap.
List:	Several connected items that are written consecutively one below the other.
Macro-molecule:	A group of molecules.
Miura pattern or Miura-ori:	It is a pattern made by rhomboid faces. It is famous to be one of the easiest one-DOF corrugations. It has been used by the engineer Koryo Miura to optimize the packing of solar panels for space travels.
Molecule:	Part of a crease pattern that can be attached to another molecule by matching the outer edges and vertices of the crease pattern with corresponding edges and vertices of a different molecule.
Mountain fold:	A crease that is convex from the observer point of view. It is usually drawn with a red dot-dot-dash line or dash-dotted line.
Nodal definition:	A group of linked nodes that make an algorithm in Grasshopper. Once set off, these definitions perform specific operations in a digital environment.
Node:	The endpoint of a line in a scheme structured as a tree. In Grasshopper, the nodes are the components which perform specific operations that can be connected one to each other with wires.
Non-developable:	That cannot be unrolled/unfolded on a plane.
Non-flat-foldable:	A pattern that can not be folded in the plane.
Nurbs:	Acronym for “Non-uniform rational basis spline”. It is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces.

Offset base:	A base with a shifted crease pattern compared with the traditional. The shifting preserves the angle between the creases and it creates space in some location of the crease pattern which can be used to add features.
Origami:	Japanese word formed by “Ori” meaning “Folding” and “Kami” meaning “Paper”. It is the art of folding paper usually performed without cutting or glueing. It is usually associated with Japanese culture.
Open sink fold:	A sink fold which does not lock after folding. It can usually be performed without flexing or crumpling the paper.
Outside reverse fold:	A type of reverse fold which changes the direction of the tip of a flap keeping its layers outside the rest of the flap.
Petal fold:	A combination of two squash folds narrowed to form a rhomboid shape. It is used to fold the petals of the traditional iris flower.
Planar curved fold:	A curved fold that lies on a plane. It is usually performed by reflecting a developable ruled surface with respect of a slicing plane.
Pleat fold:	A sequence of alternated mountain and valley creases through one or more layers of paper.
Pre-creasing:	Folding and unfolding a crease before collapsing it. It is preferred to perform a pre-creasing before complex steps.
Rabbit-ear fold:	A type of folding that makes a flap from a triangular face by folding along the three bisectors of the triangle.
Reciprocal diagram:	A graphical tool for understanding and designing structural systems. In origami is used to investigate the first order approximation of rigid origami and other kinetic properties.
Recursive tessellations:	Is a particular type of tessellation where the same folds are repeated in a smaller scale following the principles of fractals figures. (The hydrangea of Fujimoto is one of the most famous).

Reverse fold:	A type of fold that changes the direction of a flap by partially inverting the direction of the central mountain or valley crease.
Rigid-foldable:	Something that is foldable without flexing or stretching the faces.
River:	A curved or rectangular constant-width region of paper that space the flaps in a crease pattern.
Self-arrests:	When a rigid origami structure reaches a state where at least one crease hit a fold angle of 180° , thus the two adjacent faces are colliding and co-planar.
Self-blocks:	See “Self-arrests”.
Semi-pre-folded:	It is said about origami patterns which are configured in an intermediate folding state which is not unfolded nor completely folded.
Shape-oriented type:	It is a family of folded surfaces which has a crease pattern characterized by creases arranged specifically to make a particularly shaped figure.
Shaping:	The act of sculpting the abstract geometric base to form the finished model. In shaping sometimes, the paper is stretched or folded with free-form or curved creases.
Single linear crease:	A non-curving crease that does not intersect any other crease.
Sink fold:	A fold performed on an internal vertex which forms a pointy flap. It consists in mirroring the point inside the model, by inverting the mountain valley assignment of the last tip of the point and pushing it inside while collapsing it.
Squash fold:	A type of fold where a single multi-layered-flap is opened and its layers are spread and flattened (usually) symmetrically.
Structured type:	It is a family of folded surfaces which has a crease pattern characterized by groups of equal tiles.

Swivel fold:	The asymmetric version of the squash fold, where usually there is a pivot point on one of the creases, around which the whole flap rotates while being spread and flattened.
Tessellation:	It is a particular type of origami made by equal molecules that can be spread in all the directions, the limit is only the dimension of the paper used. Tessellations are usually exhibited showing the front and back side or with backlight.
Tile:	A portion of a crease pattern that can be assembled into crease patterns by matching circles and river boundaries.
Tree structure:	The branched structure that contains lists or items at different hierarchy levels.
Triangulated accordion:	An accordion where all the quadrangular faces are divided along the diagonals and all the creases are redefined with an alternated mountain/valley assignment.
Unfold:	Open a folded model obtaining, as a result, a creased sheet of paper.
Unfoldable:	Something that cannot be folded.
Unsink:	Removing a sink fold.
Valley fold:	A crease that is concave from the observer point of view. It is usually drawn with a blue dashed line.
Vertex:	A point in a crease pattern where more than one crease converges.
Wire:	A line, thread or string that connects two nodes. In Grasshopper, wires are used to input into one or more nodes the outputs of other nodes. They move data in form of single items, lists, or trees.

Finito di stampare nel mese di giugno 2022
per i tipi di Bologna University Press



alphabet **18**



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



www.buonline.com