

# Interactive Design of Curved- Crease-Folding

by  
**Shajay Bhooshan**

Supervised by  
**Dr. Paul Shepherd**  
**Prof. Paul Richens**

A thesis submitted for the degree of Master of Philosophy  
University of Bath  
Department of Architecture and Civil Engineering  
September 2015



Attention is drawn to the fact that copyright of this thesis rests with its author. A copy of this thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and they must not copy it or use material from it except as permitted by law or with the consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

A handwritten signature in black ink, appearing to be "Shajay Bhooshan".

**Shajay Bhooshan**

## ABSTRACT

Building on the historic work of Huffman (Demaine et al. 2011), there has been increasing recent interest in the digital design and architectural application of curved-crease folded (CCF) geometries. This is particularly timely, given the new possibilities of producing curved surfaces from flat sheet material afforded by developments in robotic technology. However there are difficulties in interactively modelling such geometries, which stem from the lack of both appropriate geometric descriptions and constructive tools available in commercial CAD software. The author's initial survey of methods included both the iterative optimization-based methods and simple constructive methods. Most methods presented difficulties for incorporation within an intuitive, real-time, edit-and-observe exploratory method. This research overcomes attempts to overcome these difficulties through the use of Dynamic Relaxation (DR) (Day 1965) for the interactive modelling of CCF geometries. It applies discrete differential operators and their gradients (Meyer et al. 2003), within a DR framework, to perturb meshes to satisfy the geometric criteria of CCF geometries outlined in Kilian et al. (2008). This research also outlines procedural strategies for generating appropriate topologies of an initial mesh, and a novel method for applying boundary conditions. The dissertation also includes a broad overview of existing methods to model developable surfaces, simulate elasto-plastic behaviour of thin (inextensible) shells and 2D parameterization of 3D meshes.

## ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and thanks to the following people without whom this research and dissertation would not have been possible.

My supervisors, Paul Richens and Paul Shepherd, for their exhaustive support in the development of my computational skills, research methods and formal writing abilities. Their contributions to the research by way of critical questioning, suggestive guidance, specific solutions and general conversation has been invaluable and most enjoyable. I thank them for my personal development as researcher in the past two years.

My research mentors, Philippe Block and Tom Van Mele, for very valuable insight in discrete differential calculus, optimisation methods etc., apart from very enjoyable discussions related to mesh processing, invaluable company and support. Specific thanks also to their contributions to the follow-up research to this research regarding the incorporation of curve folding constraints within discrete funicular form-finding.

My colleagues at Zaha Hadid Architects Computation and Design group (ZHACODE) and collaborators on various endeavours related to curve-crease-folding – Vishu Bhooshan, David Reeves and Henry Louth – for their contributions to designing and building prototypes, exchange of ideas related to origami and computational problem solving, youthful energy and boundless curiosity. Particular thanks to Vishu for help with logistics of prototyping and code implementation. Thanks also to David for his intuition with geometric ideas, and to Henry for his expertise in physical design with paper and metal.

My former colleagues at ZHACODE - Mostafa El Sayed and Suryansh Chandra – for their contributions to the project that sparked the initial interest – *Arum* sculpture for the Venice Biennale, other prototyping efforts, formal submissions to journals, and engaging and sometimes heated conversations regarding design and computation.

Chikara Inamura and John Klein for their continued friendship and support, participation in my formative years as a designer, and researcher, inspiration with their own novel, rigorous and diligent work and research, and contribution to the built prototype at Bangalore.

Sam Saffarian for being part of my formative years as a friend, colleague and collaborator and his inspiring and obsessive attention to detail especially to design procedures with subdivision surfaces and contribution to the second built prototype at Bangalore.

Gregory Epps, for collaborative contributions to the design of *Arum* sculpture and the many subsequent years of collaboration - sharing of ideas related to curved-folding, robotic automation, design methods, intuition and prototyping.

Theodore Spyropoulos, for life long support as a mentor, tutor and friend, opportunities to run and evolve design-based, curved-folding workshops with students of the AADRL.

Students of the Architectural Association Design Research Lab for developing several seemingly improbable and intriguing, curve-crease-folded geometries that contributed significantly to my interest to pursue this avenue of research.

Patrik Schumacher and Zaha Hadid, for providing the environment and the opportunities to do the research whilst being employed, and more significantly to enable me to grow as a designer, researcher, student and teacher.

Last but most emphatically not the least, my wife Alicia Nahmad, for being the most avid, unconditional supporter of my endeavours, understanding about the long hours of toil and silence, and contributions to the research with her left-field intuition about material behaviour and algorithms, help in organizing the logistics of prototyping and making models which I wouldn't have managed otherwise, and encouragement to develop the software application.

### **CONTRIBUTIONS FROM COLLABORATIVE WORK**

Some aspects of the research presented in this dissertation were a collaborative effort between the author and other collaborators, as highlighted below.

The author was part of the collaborative design and build team that executed the prototype described in Chapter 4. Both the computational methods and the production of manufacturing information is described in a co-authored paper (S Bhooshan, Bhooshan, et al. 2015). The contributions of this prototyping exercise to the research of the dissertation in general, and the special case of the proposed method of computing CCF geometries described in (Section 3.5.1) in noted in Section 4.4.

The author was part of the collaborative design and build team that executed the prototype described in Chapter 5. Both the computational methods and the production of manufacturing information is described in a co-authored paper(Chandra et al. 2014). The contributions of this prototyping exercise to the research of the dissertation are noted at the end of Chapter 5.

Some of the diagrams and some of the text are also part of co-authored papers (Shajay Bhooshan et al. 2015; S Bhooshan, Bhooshan, et al. 2015; S Bhooshan, Van Mele, et al. 2015; Chandra et al. 2014). These are noted in-text as appropriate.

Lastly, both the construction prototype was funded by an educational workshop organized by the author, under the Visiting School Program of the Architectural Association, London.

# Contents

1	Introduction.....	13
1.1	Brief history: Artists, designers, mathematicians .....	14
1.2	Goals and contributions: Architectural motivation and focus of current work.....	16
1.2.1	Developable & Minimal mean curvature surfaces.....	17
1.2.2	Sub-Division surfaces .....	18
1.3	Summary .....	19
2	Prior Work .....	20
2.1	Prior computational geometry.....	20
2.1.1	Useful theorems and terminology for CCF design .....	20
2.1.2	Rulings and developable surfaces .....	21
2.2	Mathematical models and geometric measures.....	21
2.2.1	Optimisation and search methods .....	22
2.2.2	Analytical methods .....	24
2.3	Constructive geometries.....	25
2.3.1	Geometric techniques for CCF .....	26
2.4	Approximating approaches .....	27
2.4.1	Discrete approximations .....	27
2.4.2	Continuous approximations .....	28
2.5	Modelling and simulation of thin shells.....	28
2.5.1	Isometric deformation.....	29
2.5.2	Curve-crease-folding.....	33
2.6	Planar development.....	34
2.7	Summary of Computational methods.....	35
2.7.1	Interactive methods .....	35
2.7.2	Choice of representation .....	36
2.8	Relevance to current research .....	36
3	Computational method.....	38
3.1	Discrete representation.....	38
3.2	Necessary conditions of curve-crease-foldability .....	38
3.2.1	Discrete measures and constraints .....	39
3.3	Method Overview .....	40
3.3.1	Iterative and local methods .....	41
3.4	Perturbation.....	42
3.4.1	Gradient of planarity .....	43
3.4.2	Gradient of developability .....	45

3.5	Boundary conditions and additional degrees of freedom .....	47
3.5.1	Gradient of developability at a boundary vertex .....	47
3.5.2	Extra row of faces.....	49
3.6	Planar Development .....	50
3.7	Results .....	52
3.7.1	Stiffness of ODE.....	52
3.7.2	Step size.....	53
3.7.3	Degrees of freedom .....	53
3.7.4	Table of results .....	54
4	Case Study 1 .....	57
4.1	Context .....	57
4.2	Design process.....	58
4.2.1	Discrete compressive mesh .....	59
4.2.2	Deriving geometries for curve-foldable moulds.....	60
4.2.3	Perturbation to CCF geometry.....	60
4.2.4	Input mesh discretization.....	64
4.3	Manufacturing information.....	65
4.3.1	Planar development / Unfold layout.....	65
4.3.2	Constructing smooth boundary curves .....	65
4.4	Summary.....	67
5	Case Study 2.....	69
5.1	Method.....	69
5.1.1	Conway operators and the method of reflection.....	69
5.1.2	Independent surface.....	70
5.1.3	Dependent surface .....	71
5.2	Quasi-developable solutions.....	73
5.3	Manufacturing information.....	73
5.4	Summary.....	74
6	Procedural generation of input geometries .....	76
6.1	Pleated geometries.....	76
6.2	Mixed polygon topologies .....	77
7	Software implementation.....	79
7.1	Discrete Data structure .....	79
7.2	Minimisation of functions .....	79
7.2.1	Runge Kutta Integration scheme .....	79
7.2.2	Data-structures suitable for iterative and local methods.....	80
7.3	Implementation.....	80

7.4	Existing software .....	81
7.4.1	Continuous representation.....	81
7.4.2	Discrete representation.....	82
8	Conclusions and Future Work.....	83
8.1	Summary .....	83
8.2	Conclusions.....	83
8.3	Limitations and Future work.....	84
8.3.1	A specific, input dependent method.....	84
8.3.2	Penalty functions.....	85
8.3.3	Parameter dependency .....	85
8.3.4	Numerical stability.....	85
8.3.5	Incorporation of 2D-3D workflow .....	85
8.3.6	Materially-based simulation.....	85
9	References.....	86

## List of figures

Figure 1 Patent for method for bending sheet material, bent sheet material and system for bending sheet material through attachment devices (Epps 2010) .....	13
Figure 2 Left: David Huffman with his famous cusp folding (Demaine et al. 1999). Middle: Top- Computer generated rendering. Bottom -physical model (Resch 1974). Right: Ron resch with his famous ‘Yellow kissing cones’ model (Yellow Cones Kissing with Ron Resch n.d.)..	14
Figure 3 Left: A photograph of a model by Irene Schawinsky, a early proponent of CCF as a student at Bauhaus Shawinsky(McPharlin 1944) Middle and Right: (Demaine n.d.) Artistic expressions of the historic and popular folding pattern of concentric circles by Erik and Martin Demaine.....	15
Figure 4 Left : Sculpture by Richard Sweeney(Modular_sculpture#0 2006). Middle : geometric arrangements of delicate curved folds by Yuko Nishimura (Yosuke Otomo n.d.). Right: Tiger mask by the late Roy iwaki (Tandem n.d.) .....	15
Figure 5 Left : Concept car resulting from a collaboration between Royal College of Arts student Ko Kyugeun, Robofold and Bentley Automobiles® (Kyungeun n.d.). Middle: Columns from the AlgoRhythms collection of Metallic architectural elements produced by Harish Lalvani in collaboration with Milgo-Bufkin (Lobell 2006)Right : Commercially available lamp by Le Klint® (Le Klint n.d.).....	16
Figure 6 Left: Flux Chair™ from Flux furniture® (Flux chair 2009). Right: Folding Boat by Max Frommeld and Arno Mathies (Folding Boat n.d.).....	16
Figure 7 Patent for Foldable shelter structure with zig-zag roof profile (Yates 1965).....	17
Figure 8 Iteratively perturbing vertices of a non-developable mesh(Left) towards developability (cylinder Right) along gradients defined in Desbrun et al. ( 2002) .....	18
Figure 9 A edit-friendly modelling paradigms, whereby a user-specified coarse mesh is subdivided and perturbed to states of minimal mean or Gaussian curvature.....	18
Figure 10 A subdivision algorithm for computer display of curved surfaces(Catmull 1974). All images taken from Catmull (1974).....	19
Figure 11 Relationship between original local curvature of the planar curved crease line, and its curvature after folding across the crease by angle $\alpha$ . Figure adapted from Tachi & Epps (2011). .....	21
Figure 12 Left: 2D example of an envelope with a parabola shown as envelope of straight lines in a plane. Middle: A spatial curve as a geodesic of developable surface guaranteed to exist. Right Using the control points of the spatial curve to edit the surface. All images taken from Bo & Wang (2007) .....	22
Figure 13 Left and right sequences : Results from (Wang & Tang 2004) showing Original mesh, and the results after applying their global and local perturbation methods. Bottom row of images show Gaussian curvature of corresponding mesh. All images taken from Wang & Tang (2004) .....	22
Figure 14 Left: Searching the convex hull of a given spatial curve, for a developable surface spanning the curve. Right Example of modelling application in garment design All images taken from Rose (2007).....	23
Figure 15: An algorithm that, given a boundary curve, analytically computes the location of buckling points in the sheet metal spanning the curve. Left to Right: cases showing one, two and four buckling points and their corresponding boundary curve. All images taken from Frey (2004) .....	24
Figure 16 A mathematical model capable of predicting the occurrence of creases on folds as a consequence of the shaped of the boundary curve. Top Left: quad based discretization. Top	



Right and Bottom Left: Occurrence conical vertices during deformation. Right: Occurrence of creases and folds. All images taken from Kergosien et al. (1994).....	24
Figure 17 Differences between the control polygon and the number of control points of a regular Bezier surface and those of Developable Bezier surface.....	25
Figure 18 Interactive modelling of developable surfaces using continuous NURBS representation. All images Tang et al. (2015).....	25
Figure 19 Methods to geometrically construct CCF geometries. Left and Right: Method of refraction and Method of reflection, as rediscovered by Koschitz (2014) from David Huffman’s original notes. Images taken from Koschitz (2014).....	26
Figure 20 Methods to geometrically construct CCF geometries Left: Method of invert reflection based on reflective principle of CCF geometry (Useful theorems and terminology for CCF design). Images taken from Mitani & Igarashi (2011).....	26
Figure 21 Procedural generation of curve-creased geometries from their straight-line or prismatic counter-parts. Images taken from Gattas & You (2014).....	27
Figure 22 Discrete approximation of given triangular meshes via partitioning of meshes and subsequent simplification. Images taken from (Mitani & Suzuki 2004).....	27
Figure 23. An iterative and perturbation algorithm to approximate developable surfaces, developed for use in the design of garments where material tolerances allow for soft-constraints of developability. Images from Decaudin et al. (2006). .....	28
Figure 24 Examples from the iterative method of as-rigid-as-possible deformation (ARAP). Incorporation of user-inputs of fixed regions (Red) and editing handles (yellow) can be noted. Images from (Sorkine & Alexa 2007).....	29
Figure 25 Examples from modelling method called <i>soft-fold</i> that allows exploration of soft and hard crease lines on flat sheets, and its implication on the folded state. Images from Zhu et al. (2013).....	30
Figure 26 A method based on isometric bending used to simulate thin, sheet and inextensible materials. Left: Results for paper, cloth and rubber. Right: comparison of isometric method to popular, non-linear model. Images from Bergou et al. (2006).....	30
Figure 27 Method to simulate origami development and kinetic transformations. The iterative method is solves derives and solves rotational constraints and rigid transformations. Images from Tachi (2009). .....	31
Figure 28. An interactive modelling method based that subdivides a user-defined coarse mesh before generating a smooth developable surface. Images from Solomon et al. (2012).....	31
Figure 29 A materially realistic method to simulate folding and crumpling of sheet material that dynamically aligns mesh discretization along fold-lines to improve accuracy. Images from Narain et al. (2013).....	32
Figure 30 A physically-accurate bending model used to simulate the particular phenomena of a (closed) crease itself buckling. Images from Dias et al. (2012).....	32
Figure 31 A mathematical treatment to find the shape of a Developable Mobius strip. Images from (Starostin & van der Heijden 2007).....	33
Figure 32. A modelling method for CCF geometries based on pre-dominantly quad-based discretization and minimization of associated bending energy. Smooth rulings as limits of discrete rulings, and the smooth developable surfaces assembled by combining patches of torsal developable surfaces can be noted. Images from Kilian et al. (2008). .....	33
Figure 33 A intuitive and digital-physical hybrid strategy to design CCF geometries. Images from Tachi & Epps (2011). .....	34
Figure 34 Planar quad strips as discrete representation of developable surfaces .....	39
Figure 35 Classification of vertices based on their Guassian curvature : Left to Right : Hyperbolic,Spherical and Euclidean vertices .....	39
Figure 36 Quad strips and associated bending energy (Eqn 1).....	40

Figure 37 Use of Conway operators (Hart n.d.) to describe a coarse and predominantly quad-faced mesh.....	41
Figure 38 Overview of proposed algorithm .....	43
Figure 39 Quantities involved in the computation of gradients of planarity and Gaussian curvature.....	44
Figure 40 Gradient of planarity of a quad-face .....	45
Figure 41 Geometric understanding of the gradient of Gaussian curvature.....	46
Figure 42 Gradient of Gaussian curvature - $gi$ , and Mean curvature (Laplacian vector) - $li$ .....	47
Figure 43 Gradient of Gaussian curvature of a (adjacent) interior vertex measured at a boundary vertex.....	48
Figure 44 The effects of using various combination of forces and their respective unrolled results.....	49
Figure 45 Effect on the resultant perturbed mesh, of adding extra-row of faces in plane with original boundary faces (Top) and normal to them (Bottom).....	50
Figure 46 Left to Right : Differences between Authalic, Conformal and isometric parametrization of an example 3-crease mesh.....	51
Figure 47 Examples of misalignment errors produced by Pepakura in the unfolding process used to establish stopping criterion for the perturbation process. ....	51
Figure 48 Convergence graph of developability metric when only Gaussian force is applied: Left – two-crease example mesh, Right four-crease mesh.....	52
Figure 49 Convergence graph of planarity metric when only planairty force is applied: Left – two-crease example mesh, Right 4-crease mesh .....	52
Figure 50 - Convergence graph of developability & planarity metrics when Planarity and Gaussian forces are applied simultaneously: Left – two-crease example mesh, Right 4-crease mesh. Planarity graph scaled by 100. ....	53
Figure 51 Convergence graph of developability & planarity metrics when Planarity and Gaussian forces are applied sequentially: Left – two-crease example mesh, Right 4-crease mesh. Planarity graph scaled by 100.....	53
Figure 52 Left : Resultant positions of boundary vertices that make the interior vertices developable. Right : Original positions of boundary vertices. (Example two-crease mesh).....	54
Figure 53 Left : Resultant positions of boundary vertices that make the interior vertices developable. Right : Original positions of boundary vertices. (Example three-crease mesh).....	54
Figure 54 Left to Right: original single-crease mesh, perturbed result and unrolled layout ..	55
Figure 55 Left to Right: original two-crease mesh, perturbed result and unrolled layout.....	55
Figure 56 Left to Right: original multi-crease mesh, perturbed result and unrolled layout. Top to bottom: 3-crease mesh, 4-crease mesh, mesh with closed creases, mesh used in prototype 2 (Chapter 4) and prototype 1 (Chapter 5). ....	56
Figure 57 Construction of a funicular skeleton structure using CCF moulds. Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	57
Figure 58: Typical form work for shells. Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	58
Figure 59: Showing a mesh in close-up with mean-curvature direction (red lines), mesh-edges (grey) and a y-shaped node (bold). Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	58
Figure 60: Form finding of funicular mesh. Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	59
Figure 61: Design Iterations, the total number of Y-panels was a driving factor in the demonstrative prototype. Image courtesy (S Bhooshan, Bhooshan, et al. 2015) .....	59

Figure 62: Left to right: Bevel operator on mesh-edges, extrude and scale operator on mesh-faces, Catmull-Clark smoothing, deletion of some of the edges to produce a predominantly quad CCF topology. Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	60
Figure 63: Formulation of the planarity force. Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	61
Figure 64: Formulation of the developability force. Image courtesy (S Bhooshan, Bhooshan, et al. 2015) .....	62
Figure 65: Vertex classification. Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	62
Figure 66: A single Y component before perturbation (a) and after perturbation (b). Green faces are planar and white vertices are developable. Image courtesy (S Bhooshan, Bhooshan, et al. 2015) .....	63
Figure 67: Relaxation sequence of a pair of Y components. Image courtesy (S Bhooshan, Bhooshan, et al. 2015) .....	64
Figure 68 Separation of mesh into CCF moulds, subsequent to perturbation. Image courtesy (S Bhooshan, Bhooshan, et al. 2015).....	65
Figure 69: (A) Input mesh unrolled, (B) Extracted curves, (C) Typical panel fabrication drawing. Image courtesy (S Bhooshan, Bhooshan, et al. 2015) .....	65
Figure 70 Procedure for generating smooth boundary curves in the planar domain .....	66
Figure 71: Shows the smooth interpolation of the crease lines using various NURBS schemes.....	67
Figure 72 Relation between CCF moulds and element depths. Image courtesy (S Bhooshan, Van Mele, et al. 2015).....	67
Figure 73 Skeleton representation of a closed polyhedron, developed as CCF geometry and assembled on-site. Image courtesy (Chandra et al. 2014).....	69
Figure 74 Independent surface in white, dependent surface in dark grey. Image courtesy (Chandra et al. 2014).....	70
Figure 75 Geometric operations to construct the independent surface from a user-defined polyhedron. Image courtesy (Chandra et al. 2014).....	71
Figure 76. The method of reflection produces intersecting surfaces. Image courtesy (Chandra et al. 2014) .....	71
Figure 77. Computing the dependent surface. Image courtesy (Chandra et al. 2014).....	72
Figure 78. Key results from unfold error measurements. Image courtesy (Chandra et al. 2014).....	73
Figure 79. Process of CAD/ CAM info generation. Image courtesy (Chandra et al. 2014) ...	74
Figure 80. Parametric variations of the independent and dependent surface computed from the same base polyhedron. Image courtesy (Chandra et al. 2014).....	74
Figure 81. 2m long aluminium polyhedron, folded and assembled in 6 hours by a group of 15 students. Image courtesy (Chandra et al. 2014).....	75
Figure 82 Use of <i>Extrude</i> operator on crease edges, Catmull-clark smoothing and edge-deletion to produce CCF topology .....	76
Figure 83 Successive use of <i>Extrude</i> operator on crease edges to produce pleated topologies .....	77
Figure 84 Use of Chamfer and Bevel operators, Catmull-Clark smoothing and edge-deletion to produce 3-crease CCF topology.....	77
Figure 85 Application of Chamfer and Bevel operators from Figure 84 on mesh used to construct prototype 1 (Chapter 4) .....	78
Figure 86 showing the implementation of the Model-View-Controller pattern of software design and its interaction with the proposed algorithm and data-structures described in sections 7.1 & 7.2 .....	81

Figure 87 Folding sequence for a three-crease mesh, as produced by *Freeform Origami* software ..... 82

# 1 INTRODUCTION

Building on the historic work of Huffman and Ron Resch in the 1970s (Demaine et al. 2011), there has been increasing interest in the digital design and architectural application of Curved-Crease Folded (CCF) geometries. Designers have previously been inspired by and studied straight-crease folding, or more formally known as prismatic folding or *origami*. In comparison, CCF offers certain advantages and imposes certain other constraints: CCF is better able to represent smooth surfaces unlike the prismatic version which requires a much higher discretization to do so. The curving of the crease-line on other hand implies that the numbers of types of surfaces that can be formed are restricted. However, despite the advantages and opportunities for discovery of novel shapes, CCF has been under-explored. This has changed over the past few years, given the advancement of the mathematical and physical understanding of CCF and the new possibilities of producing curved surfaces from flat sheet material afforded by developments in robotic technology (Epps 2010; Balkcom & Mason 2008) (Figure 1).

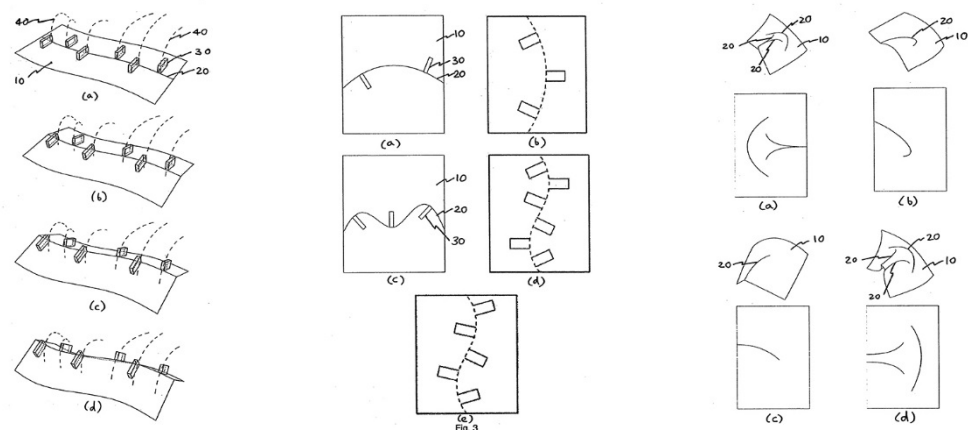


Figure 1 Patent for method for bending sheet material, bent sheet material and system for bending sheet material through attachment devices (Epps 2010)

The study of prismatic origami – its mechanisms, behaviours and simulation - has been shown to be significantly useful in the production of transformable structures (Resch 1973; Tachi 2010). The curved-crease variant also has similar benefits (Resch 1974; Vergauwen et al. 2014). Additionally, CCF surfaces improve upon the structural benefits of the prismatically folded surfaces, as noted by the research being carried forth in EPLF (Robeller et al. 2014). The understanding of structural behaviour and performance of CCF, although only beginning to be studied systematically (Rohim et al. 2013; Vergauwen et al. 2014) can significantly aid the use of CCF by designers and thus promote novel applications of this intriguing set of geometries.

The focus of this research however is the difficulty in modelling such geometries digitally and interactively. This difficulty stems from the lack of both appropriate geometric descriptions and constructive tools available in commercial CAD software. As noted in the survey paper by Demaine et al. (2011), there are two main algorithmic themes in producing CCF geometries: Constructive geometry, and the application of Discrete Differential Geometry (DDG) to model CCF shapes. The latter approach has been found to provide the most generalised solutions. Most current methods however present difficulties when incorporated within an intuitive, real-time, edit-and-observe exploratory method of design. This research then, aims to overcome these difficulties.

The research presented here was motivated by a case study of the design and fabrication of a self-supporting, multi-CCF-panel installation for the Venice Biennale 2012 by Zaha Hadid Architects (Shajay Bhooshan et al. 2014). It operates against the backdrop of the exciting potentials that the field of curved-crease folding offers in the development of curved surfaces that can be manufactured from sheet material.

## 1.1 Brief history: Artists, designers, mathematicians

A comprehensive history of prismatic and curve-crease folding can be found in the recently completed doctoral dissertation of Koschitz (2014). These histories and seminal figures in the relevant areas of art, design and mathematics are briefly recounted here to highlight the design and application focus of the current research.

Historic accounts of the origins of paper-folding have been difficult to trace. However it is accepted to have become generally known in Europe by the 18<sup>th</sup> century. Likewise prismatic folding or Origami was also generally known by the 18<sup>th</sup> century in Asia and its modern day practise, including the accepted notational system of mountains and valleys (Yoshizawa–Randlett system), is widely credited to Akira Yoshizawa (Randlett 1961). The thesis of Koschitz (2014) also lists several known documents and craft manuals pertaining to its study and application in design -napkin folding, toys, maps etc.

The historic development of CCF inherits the episodic structure and multi-disciplinary influence from its prismatic variant. Modern history of CCF - its systematic study and design application - is known to have started with the work of Joseph Albers at Bauhaus in 1927 (Esther Dora Adler 2004), subsequently formally cemented and artistically expanded by the work mathematician David Huffman and Artist Ron Resch in the 1970s (Demaine et al. 2011)(Figure 2).

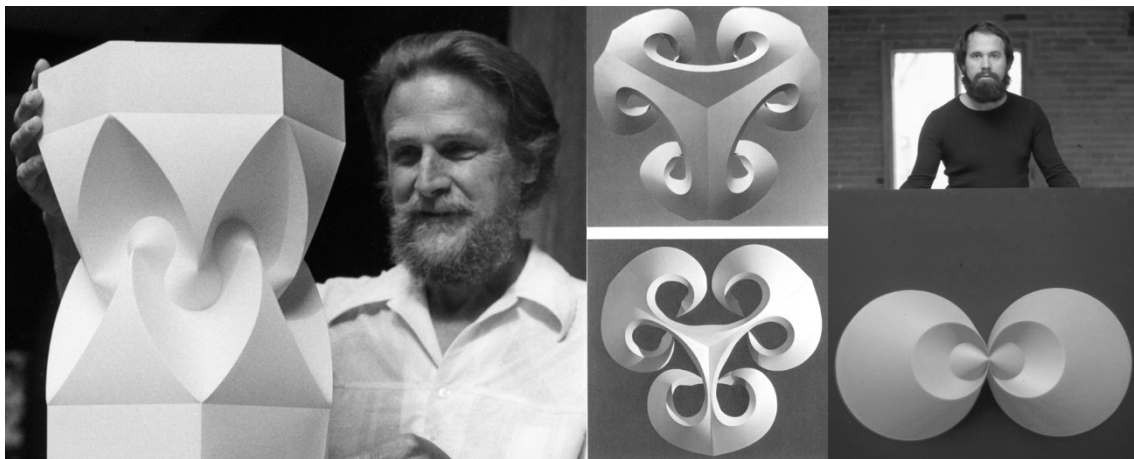


Figure 2 Left: David Huffman with his famous cusp folding (Demaine et al. 1999). Middle: Top- Computer generated rendering. Bottom -physical model (Resch 1974). Right: Ron resch with his famous ‘Yellow kissing cones’ model (Yellow Cones Kissing with Ron Resch n.d.)

Another group of prolific researchers from MIT – Professor Erik Demaine, Martin Demaine, and Richard Duks Koschitz - have contributed immensely to comprehensively tracing the history of CCF (Demaine et al. 2011; Koschitz 2014; Demaine et al. 1999), the systemic treatment of the mathematics of folding, the dissemination of related knowledge (Demaine & O’Rourke 2007) and also in the production of artistic expression using CCF (Figure 3).

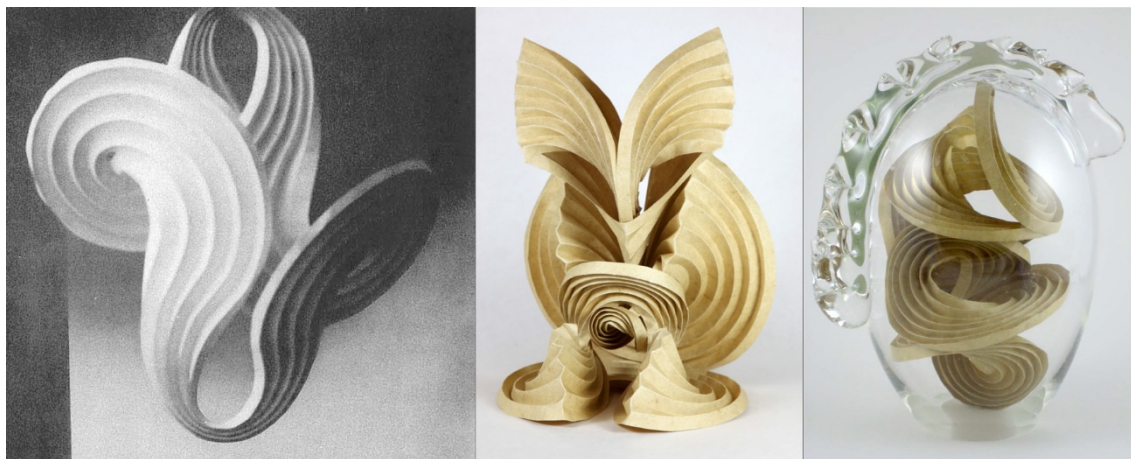


Figure 3 Left: A photograph of a model by Irene Schawinsky, a early proponent of CCF as a student at Bauhaus Shawinsky(McPharlin 1944) Middle and Right: (Demaine n.d.) Artistic expressions of the historic and popular folding pattern of concentric circles by Erik and Martin Demaine.

Recent artistic proponents of curved-crease folding include the designer Richard Sweeney, Yuko Nishimura and the late Roy Iwaki (Figure 4). Sweeney's CCF large-scale paper sculptures have been famously exhibited at the Selfridges departmental store in London (2008), Lincoln Cathedral in the UK (2012) etc. Nishimura is known for her exploration of geometric patterns and Iwaki for his use of CCF to create paper-masks of animals.

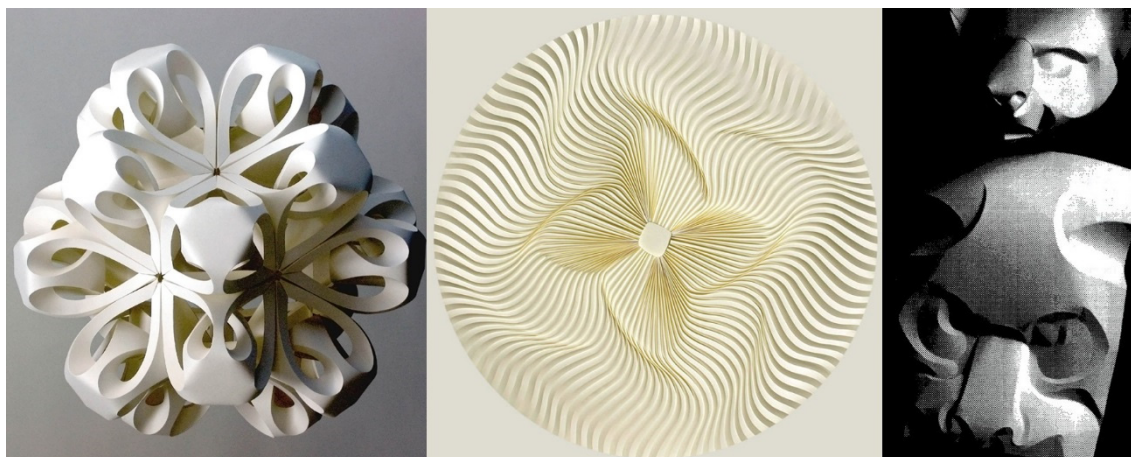


Figure 4 Left : Sculpture by Richard Sweeney(Modular\_sculpture#0 2006). Middle : geometric arrangements of delicate curved folds by Yuko Nishimura (Yosuke Otomo n.d.). Right: Tiger mask by the late Roy iwaki (Tandem n.d.)

In the realm of design and commercial products, several collaborations of Gregory Epps and his company Robofold, computationally created sculptural columns by Harish Lalvani, and the lamp from the Le Klint Company stand out (Figure 5). The lamp from Le Klint, has been in production since 1971(Le Klint n.d.), whilst Lalvani's columns were part of a pioneering collaboration between the architect and an architectural-metal fabricator – Milgo Bufkin in 2003(Lobell 2006). Robofold has been instrumental in making industrial production of CCF geometries accessible to digitally minded designers since its inception in 2008.

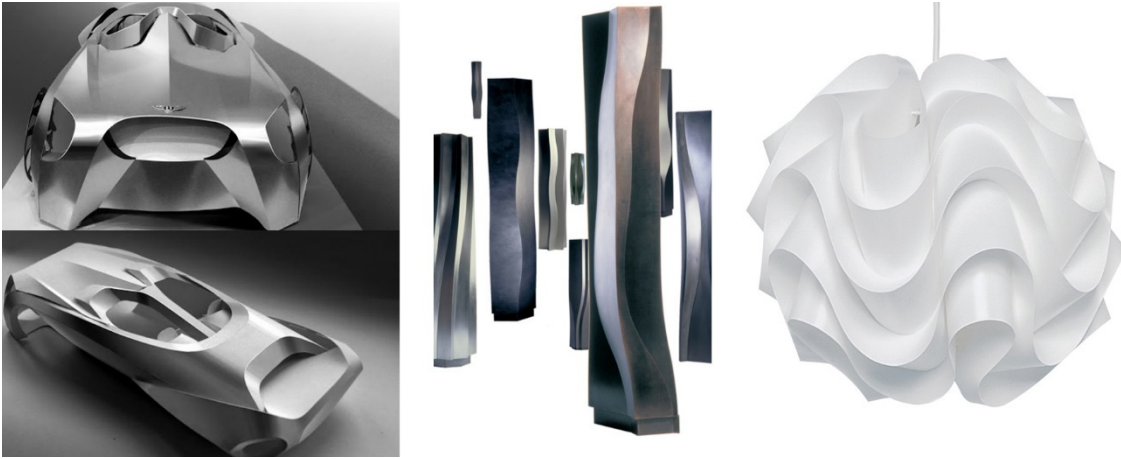


Figure 5 Left : Concept car resulting from a collaboration between Royal College of Arts student Ko Kyugeun, Robofold and Bentley Automobiles® (Kyungeun n.d.). Middle: Columns from the AlgoRhythms collection of Metallic architectural elements produced by Harish Lalvani in collaboration with Milgo-Bufkin (Lobell 2006) Right : Commercially available lamp by Le Klint® (Le Klint n.d.)

Two recent products of note, that use the flat-packing potential of CCF are the Flux-Chair® by the Dutch furniture company Flux® and the Folding Boat by designers by Max Frommled and Arno Mathies (Figure 6). The furniture company has since expanded its products to other furniture that also use CCF geometries.



Figure 6 Left: Flux Chair™ from Flux furniture® (Flux chair 2009). Right: Folding Boat by Max Frommled and Arno Mathies (Folding Boat n.d.).

## 1.2 Goals and contributions: Architectural motivation and focus of current work

There have been numerous applications of prismatic folding in architecture – both for the discovery of novel shape, and its transformative and structural properties, as briefly summarised in Chapter 2: Prior Work. This is perhaps best exemplified by the 1966 patent for a foldable house by Yates (1965) (Figure 7).



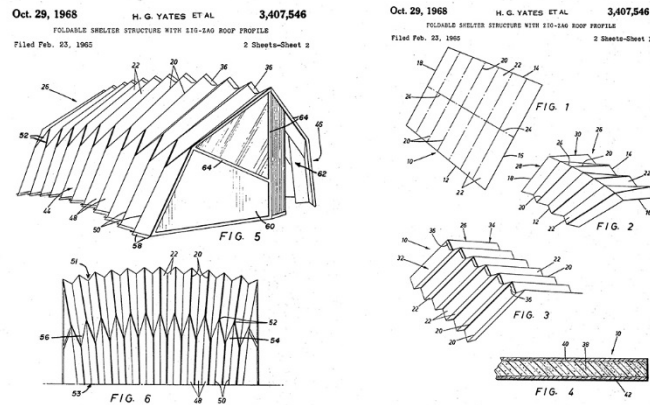


Figure 7 Patent for Foldable shelter structure with zig-zag roof profile (Yates 1965)

CCF on the other hand, has not found as many architectural-scale applications yet. Notable recent exceptions to this are Zaha Hadid Architects and their Arum sculpture (Bhooshan et al. 2014a), and Rebecca Braun and Tyler Smith from University of Michigan (Braun 2015a; Braun 2015b). Architect Harish Lalvani was an early pioneer in the use of CCF at architectural scale as previously mentioned (Lobell 2006). The installation of ZHACODE at the Venice biennale was the subject of a preceding case-study to this research and provides the motivation of developing methods for architectural application of CCF. Thus, the primary objectives of the research include:

1. Incorporation CCF design within established contemporary digital design workflows
2. Compatibility with an interactive and exploratory design process for CCF geometries.
3. Retaining the possibility of discovering novel shapes that physical methods of CCF design afford. One such physical method of design promoted and employed by Ron Resch (Paper & Stick Film 1992), included crumpling pieces of paper and subsequently systematically investigating the resulting (curved) creases. This then leads to developing and describing precise flat curve-crease layouts, that when folded result in three-dimensional shapes.

In view of the above objectives, two increasingly common techniques in computational (architectural) design are surmised below.

### 1.2.1 Developable & Minimal mean curvature surfaces

*Dynamic relaxation* as originally developed by Day (1965) and extended by Barnes (1999), is a computational method used to find equilibrium shapes of geometries subjected to (axial) forces. It has been extensively used find the shapes of cable-nets, and fabric membranes subjected to tensile forces (Wakefield 1999). The method treats nodes of a given net as lumped masses and its edges as springs, and proceeds to iteratively move the nodes that are subjected to gravity and spring forces until equilibrium is reached. There are several examples of modifications to the method, especially in the formulation of the forces (Harding & Shepherd 2011; Bak et al. 2012), that have been proposed to solve other problems of statics, simulating cloth-like material etc. Famously, Dr Williams from University of Bath used a modification of the method to design the roof of the British Museum, London (Shepherd & Williams 2010). The method shares similarities with the particle-spring method of simulating various deformable shells (Baraff et al. 1997; Bhooshan et al. 2014b), commonly used in computer graphics applications.

Particularly relevant to the context of the current research is that it has been employed to find the shapes of so called minimal-mean-curvature-nets (M-surfaces) under given boundary conditions (Wakefield 1999). Also relevant is that the method has previously been modified to solve problems of energy minimisation and thus employed in a geometric setting as opposed to

its original setting of non-linear problems of static equilibrium. For example, subjecting the nodes of the graph to virtual or non-physically based forces, such as gradients of planarity of associated mesh-faces (Gauss 2014), can perturb the nodes of an originally non-polyhedral mesh to produce a polyhedral mesh. Similarly subjecting the nodes to forces along gradients of Gaussian curvature can produce developable surfaces or minimal-Gaussian-curvature surfaces (D-surfaces) (Figure 8).

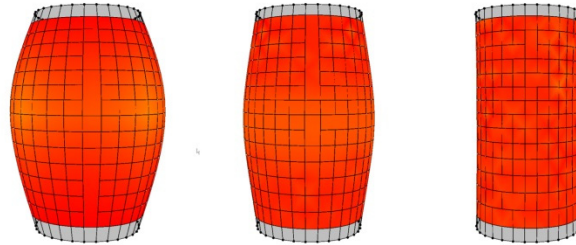


Figure 8 Iteratively perturbing vertices of a non-developable mesh(Left) towards developability (cylinder Right) along gradients defined in Desbrun et al. ( 2002)

This aspect of using DR as a framework to variably produce M surfaces or D surfaces has particular architectural benefits: M surfaces can be produced by stretching or tailoring sheet material such as fabric, where-as D-surfaces can be produced by forming sheet material such as metal (Figure 9).

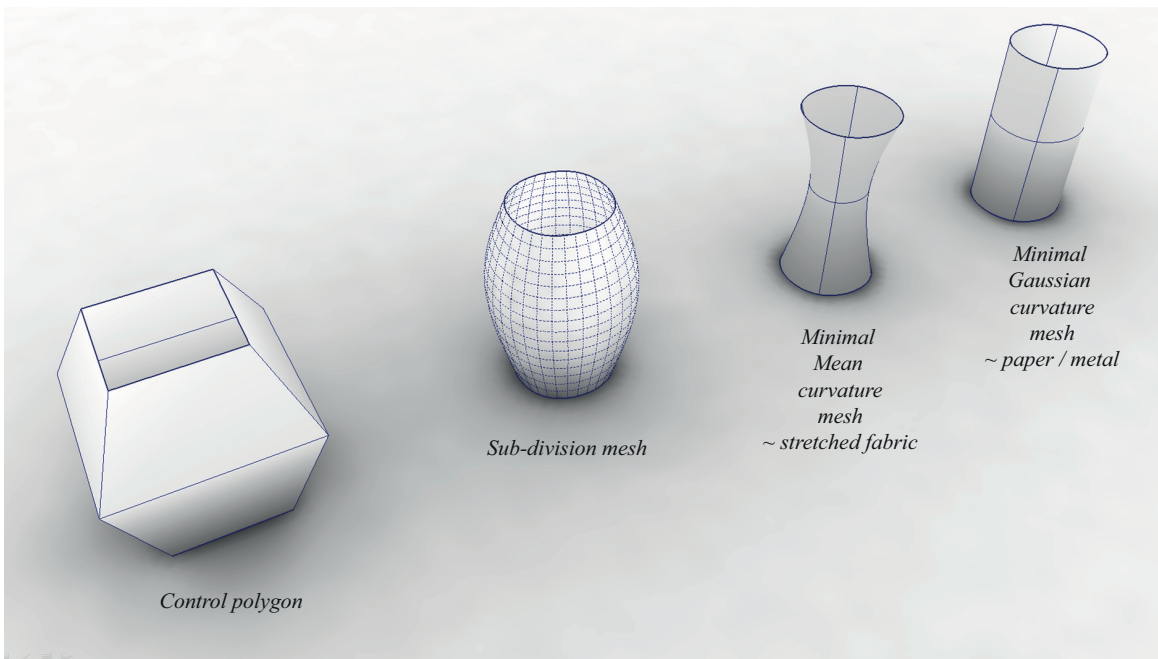


Figure 9 A edit-friendly modelling paradigms, whereby a user-specified coarse mesh is sub-divided and perturbed to states of minimal mean or Gaussian curvature.

### 1.2.2 Sub-Division surfaces

One of the widely used geometric descriptions and technologies in the computer graphics and animation industry is the so-called subdivision surfaces (Catmull 1974). This essentially involves the procedural generation of smooth geometries via the subdivision of low-resolution input geometry (Figure 10) (Catmull 1974). The benefits of subdivision surface based modelling in architectural form-finding have been previously established (Shepherd & Richens 2010; Bhooshan & El Sayed 2011). Further, the benefits of application of DR techniques combined with subdivision surfaces to explore, design and fabricate minimal mean curvature surfaces has also been established (Bhooshan & El Sayed 2012). Thus, one of the specific goals of this

research is to extend the use of the established mesh modelling techniques and tools, in combination with the DR method, to find geometries that can be curved-crease folded.

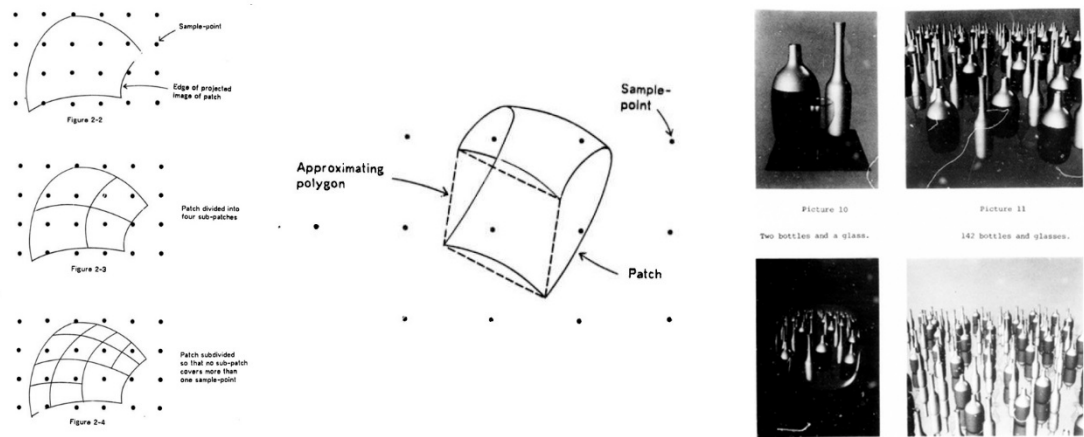


Figure 10 A subdivision algorithm for computer display of curved surfaces(Catmull 1974). All images taken from Catmull (1974)

### 1.3 Summary

In summary, the research operates against the backdrop of the exciting potentials that the field of curved-crease folding offers to architectural design: its rich history of design strategies and precedents, mathematical under-pinning and the possibility of manufacture of curved surfaces from sheet material. The main challenges are developing an intuitive design strategy, production of information adhering to manufacturing constraints and incorporation within established digital design work-flows. The research intends to overcome these.

The essential contribution of the research is a proposed computational method to find curve-crease foldable geometries, including novel strategies to deal with boundary conditions. The proposed method could negotiate the dual objectives of ease of use in exploratory design, and the physical production of the geometries by folding sheet material. The research was applied in the design and production of an architectural scale prototype where CCF geometries were used as lost-formwork for concrete casting and additionally to produce two artistic installations in aluminium. The documentation and exposition of the details of these built work, form additional contributions of the research. Lastly, the research also outlines a few procedural operations that could be employed to produce various input topologies that can then be perturbed towards being CCF geometries. This represents the design contribution of the research, in a vein similar to the repository of crease-patterns for Origami (Lang 2004) by mathematician Robert Lang.

## 2 PRIOR WORK

CCF has interested researchers and practitioners in a myriad of disciplines. Principal among them, in the context of this research, are (discrete) differential geometry and Origami.

### *Differential geometry*

Fundamental aspects of differential geometry as applicable to developable surfaces - surfaces embedded in 3D space that can be transformed to their planar equivalents without stretching or tearing - and Curve Crease Folding can be studied in Do Carmo (1976); Toponogov & Rovenski n.d.; Fuchs & Tabachnikov (1999); Huffman (1976); Duncan & Duncan (1982).

### *Origami*

The art and practice of origami, and its increasing use in engineering applications is another vast field of research and knowledge that can contribute to understanding CCF. An early interest, from the 1980s, in the applicability of the origami to engineering problems can be found in the recently translated Haga et al. (2008). This particular contribution was a result of a fusion of interests in the design of cranes and origami. Such an interest has only proliferated since, as can be understood by the spate of papers in engineering journals: The benefits of Origami configurations for shell structures was studied in the doctoral dissertation of (Schenk 2011); the use of Origami water-bomb structure in compliant design is described in Hanna et al. (2015), their use in deployable structures in Saito et al. (2013) and Tachi (2010).

Similarly CCF is also beginning to be used in deployable structures particularly because of their properties of isometric (distortion free) mapping from 3D embedding to planar configurations (Resch 1974; Tachi & Epps 2011; Vergauwen et al. 2014). They are also being considered for use as kinetic structures because of their property of (almost) rigid transformation from one state to another (J Lienhard et al. 2011; Julian Lienhard et al. 2011).

### 2.1 Prior computational geometry

Curve-crease folding (CCF) could be viewed as a special extension of developable surfaces. In particular CCF can be viewed as intersection of two or more developable surfaces, subject to additional geometric constraints (as further explained in Section 3.1 on discrete representation). Further, CCF shares similarities with origami in the sense of being produced from scoring and bending sheet materials. Thus the survey of prior work attempts to surmise the long history of research and contributions in the field of Computational Developable Surfaces, and Origami. The survey also describes prior work specifically within CCF where applicable, and will highlight geometric properties that underpin the surveyed methods. It will also highlight aspects relevant to the primary objective of this research (Section 1.2): compatibility with interactive, edit-friendly, exploratory, and well-established CAD work-flows.

#### 2.1.1 Useful theorems and terminology for CCF design

To situate the survey in the context above, some fundamental theorems from differential geometry, is recounted here from Vergauwen et al. (2014). These might be known to the CCF design community as intuitive geometric relations. Additionally, a few definitions of developable surfaces and its *rulings* are also mentioned.

1. At every point on the curved crease, the tangent planes of the two surfaces make equal angles with the osculating plane of the curved crease in that point (Huffman 1976) and the process of curved line folding produces 2 surfaces having equal and opposite surface normal curvatures along the fold line (Duncan & Duncan 1982).

2. Throughout the deformation the geodesic curvature remains equal to the ordinary curvature of the crease in the plane development (Fuchs & Tabachnikov 1999)(Figure 11). As a result, the following equation is obtained:

$$k = k_0 / \cos \beta$$

With  $k_0$  the curvature of the plane curve,  $\beta$  the angle between the osculating plane and the surface and  $k$  the curvature of the space curve (Fuchs & Tabachnikov 1999).

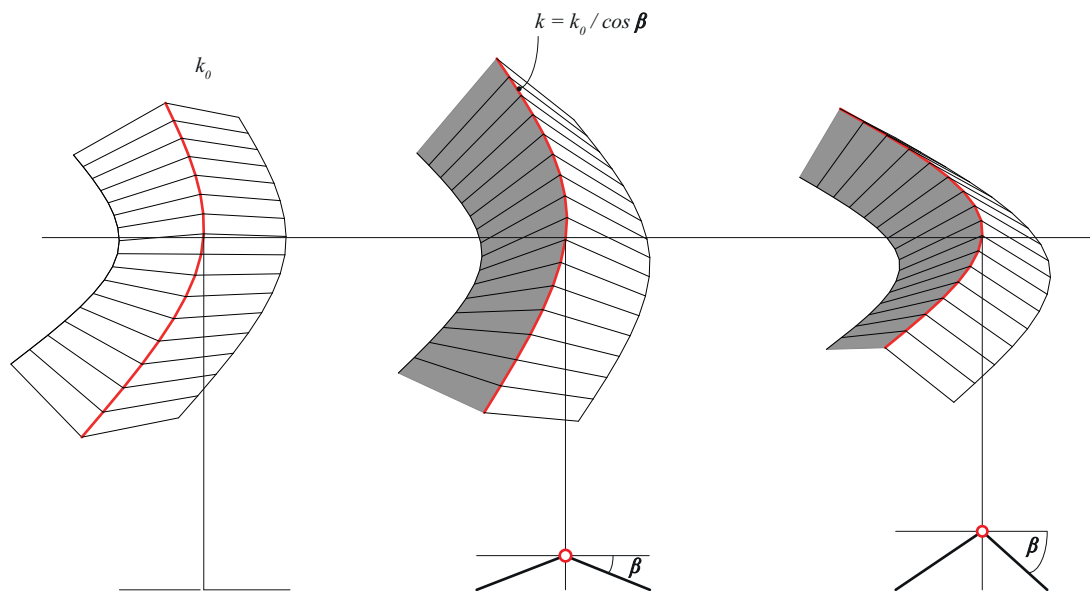


Figure 11 Relationship between original local curvature of the planar curved crease line, and its curvature after folding across the crease by angle  $\alpha$ . Figure adapted from Tachi & Epps (2011).

### 2.1.2 Rulings and developable surfaces

Generally speaking, a ruled surface is a surface generated by a straight line moving along a curve. The straight lines themselves are called *rulings*. A *ruling* is *torsal* if the tangent plane of the surface is the same for all points on the ruling line. A ruled surface is developable if all its rulings are *torsal* (Dolgachev 2012). Such patches can combine to form more complex developable surfaces.

Another often used property of developable surfaces stems from the *Gauss-bonnet theorem* (Eric W. Weisstein n.d.), which links the integral of the Gaussian curvature over the entire surface (Riemann Manifold) to the Euler characteristic of the surface. It then gives the property that surfaces isomorphic to a plane i.e. developable surfaces will have vanishing Gaussian curvature. The discrete equivalent which links the integral to the total angular defect of all the vertices of a polyhedron comes from the Descartes *theorem of total angular defect* (Eric W Weisstein n.d.).

## 2.2 Mathematical models and geometric measures

A class of approaches to computationally represent and model developable surfaces, aim to appropriate their geometric properties such as vanishing Gaussian curvature, properties of its

Gauss or normal maps, tangent spaces etc. These methods usually aim to satisfy manufacturing constraints and/or interactive design of geometries. As a consequence, they abstract material properties of thin-shells as mathematical models incorporating geometric constraints such as planarity, minimal Gaussian curvature, and smoothness of curvature. Such methods are of particular interest to this research, due to their amenability both to intuitive understanding and interactive editing.

### 2.2.1 Optimisation and search methods

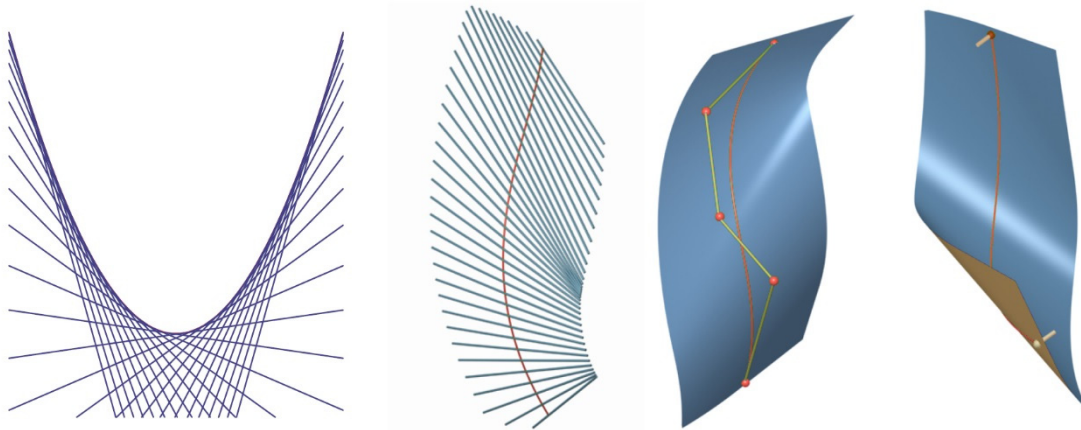


Figure 12 Left: 2D example of an envelope with a parabola shown as envelope of straight lines in a plane. Middle: A spatial curve as a geodesic of developable surface guaranteed to exist. Right Using the control points of the spatial curve to edit the surface. All images taken from Bo & Wang (2007)

In the search for edit-friendly methods, Bo & Wang (2007) provide one of the earliest examples of interactive editing of bending of paper-like surfaces. They exploit the fact that developable surfaces are the envelopes of *rectifying planes* of spatial curves – planes that span the bi-normal and tangent of a spatial curve at each point of the curve. They also show that this property essentially means that any given curve is a geodesic on a unique developable surface, called the *rectifying developable*, that is guaranteed to exist. They then proceed to develop a method to procedurally generate the surface. The rulings of such a surface are the intersections of the *rectifying planes*. They further utilise this property to use the control points of the spatial curve or geodesic to interactively manipulate the curve and consequently the surface (Figure 12). They use a Bezier representation for the curve and assume that at least third order derivatives are defined everywhere. They then minimize a quadratic function to interpolate the (normals of) *ruling lines* from given normals at the end-points of the curve.

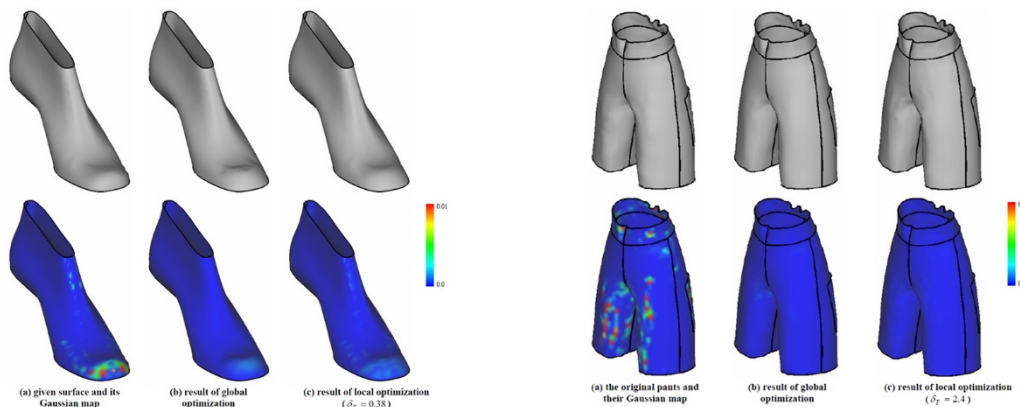


Figure 13 Left and right sequences : Results from (Wang & Tang 2004) showing Original mesh, and the results after applying their global and local perturbation methods. Bottom row of images show Gaussian curvature of corresponding mesh. All images taken from Wang & Tang (2004)

This dissertation is explicitly interested in perturbing a user-defined mesh towards a nearest CCF solution. Wang & Tang (2004), provide an understanding of the mechanisms of such a perturbation process. They propose a method to perturb the vertices of a given triangular mesh to make them *developable*. Their global optimisation method operates on the geometric property of developable surfaces of being locally homoeomorphic to a disk i.e. to have zero Gaussian curvature everywhere. Their optimisation routine utilises a regularising function to penalise deviation from the input meshes. Their optimisation method is iterative, and based on the gradients of a composite *error function* with analytical gradients for the penalty function w.r.t to vertex positions and numerically computed gradients for the Gaussian curvature. Their global approach, by their own account, provides high accuracy results at the expense of interactivity. Interestingly they propose another local approach that provides interactivity at the expense of accuracy (Figure 13). It can also be noted that their attempt was one of the earliest to convert the geometric feature of developable surfaces of zero Gaussian curvature into an optimisation requirement that the sum of angles around each vertex is equal to  $2\pi$ .

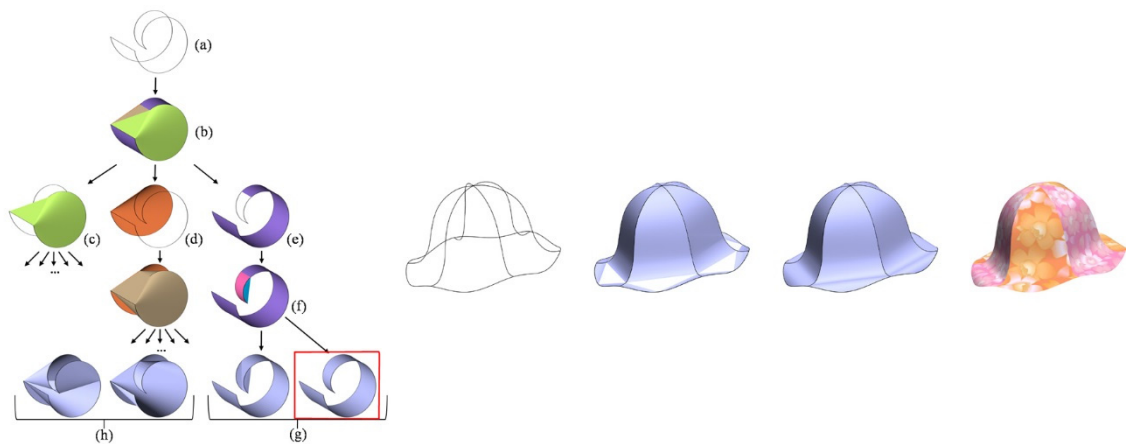


Figure 14 Left: Searching the convex hull of a given spatial curve, for a developable surface spanning the curve. Right Example of modelling application in garment design All images taken from Rose (2007)

The work of Rose (2007) provides an intuitive understanding of the existence and neighbourhood of the previously mentioned *nearest* (developable) solution. It appropriates the link between the convex hull of *arbitrary* spatial curves – a wrapping envelope of its constituent points, and the developable surfaces that they enclose (Figure 14). The proposed method exploits the observation that most edges of developable triangulations should be locally convex, and consequently, convex hulls of (closed) spatial curves are a good place to search within. The method uses a fairly elaborate branch-and-bound machinery to search through the space of possible, interpolating developable surfaces, embedded within the convex hull. The search is able to accommodate additional constraints of desirable shape properties such as *fairness* of resulting surface, quality of the triangulation etc. It may be noted that the stated computation times, are not compatible with interactive modelling. However, given that this seminal work is from a decade ago, we might expect the algorithm to be reasonably interactive now.

## 2.2.2 Analytical methods

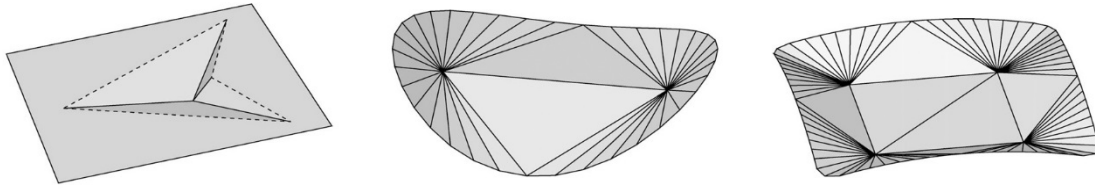


Figure 15: An algorithm that, given a boundary curve, analytically computes the location of buckling points in the sheet metal spanning the curve. Left to Right: cases showing one, two and four buckling points and their corresponding boundary curve. All images taken from Frey (2004)

Frey (2004) is of interest in terms of the application of computational methods in a manufacturing setting. His method also uses the zero Gaussian curvature property of developable surfaces, in particular to the applicability of the angle sum of  $2\pi$  to the special case of interior points being points of singularity (w.r.t Gaussian curvature). These points, referred to as *d-vertices*, are encountered in the design of sheet metal parts in the automotive industry and additionally they are also locations where the metal physically exhibits *buckling* (Figure 15). The method uses an analytical solution for finding the position of such points for a given (usually planar) triangulation of an input curve. The solution is derived from moving the points in the *z direction* to a location where the *angle-defect* is corrected. Evidently, the analytical solution gets more complex as the number of input points increase. Interestingly, potential incorporation of a surface area constraint – an important manufacturing consideration - is suggested. It is however not explicitly included in the solution.

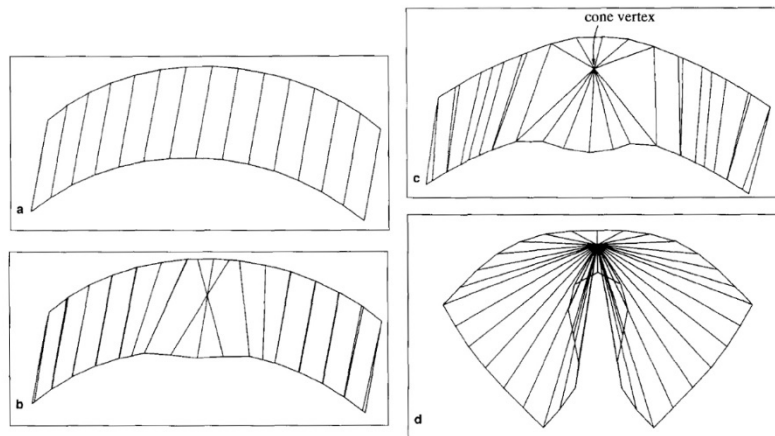


Figure 16 A mathematical model capable of predicting the occurrence of creases on folds as a consequence of the shaped of the boundary curve. Top Left: quad based discretization. Top Right and Bottom Left: Occurrence conical vertices during deformation. Right: Occurrence of creases and folds. All images taken from Kergosien et al. (1994)

The mathematical model of Kergosien et al. (1994) provides insights into computational representation of developable surfaces and also the mechanisms of simulating the bending and creasing of paper-like sheets. They derive their model and its constraints from the inextensible and zero-Gaussian curvature properties of paper. Their simulation, based on this model, enables the study of isometric 3D deformation of a planar, rectangular piece of paper and importantly predicts the occurrence of creases and folds caused by certain shapes of boundaries (Figure 16). Interestingly, their computer simulation utilises simple node-based dynamics with user-interaction abstracted as ‘forces’ that are counter-acted by internal bending forces of the sheet, as computed by their mathematical model.



### 2.3 Constructive geometries

This class of techniques to model developable surfaces, in contrast to the previous approaches, aim to integrate within the well-known CAD paradigm of allowing users to directly draw and manipulate smooth surfaces using a small set of so-called *control points*. It is quite common for this paradigm of surface editing to employ parametric surface representations –Bezier and NURBS - to allow for such interactive and yet precise control over geometries. However, all attempts surveyed in this category of techniques, have used Bezier parametric representations, as briefly described below:

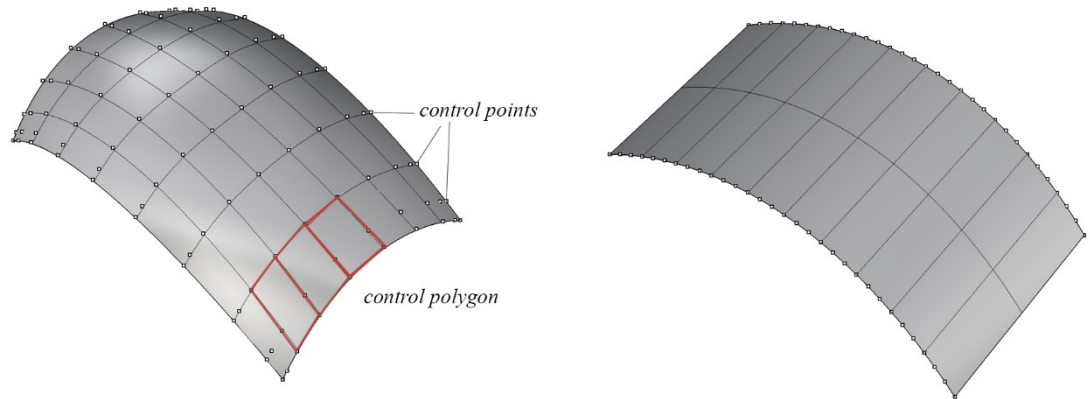


Figure 17 Differences between the control polygon and the number of control points of a regular Bezier surface and those of Developable Bezier surface.

In Bezier surfaces, every point of the surface is defined by two parameters –  $u$  and  $v$  and the Cartesian coordinates of such a point in 3D is a function of the positions of the grid of control-points (*control-net*) i.e. every point is a weighted interpolation of the positions of the control points. If the number of control-points in any of the two grid-directions is restricted to 2, the resultant surface will be *ruled* (Figure 17).

Lang & Röschel (1992) developed the necessary conditions and constraints on the locations of the *control-net* of a *ruled* Bezier surface, for it to become developable. They derive these constraints from the geometric requirements of developable ruled surfaces – namely that all points on each *ruling* line, must lie on a unique tangent plane. The non-linearity of these constraints, have in part contributed to the fact that the interactive manipulation of such developable Bezier surfaces have been elusive since the early 1990s (Tang et al. 2015). Since the current research is focussed on discrete geometric representations, the reader is referred to a recent paper by Tang et al. (2015). They provide a complete treatment of the topic and a novel method that overcomes several difficulties, including interactively modelling CCF geometries using Bezier representations (Figure 18).

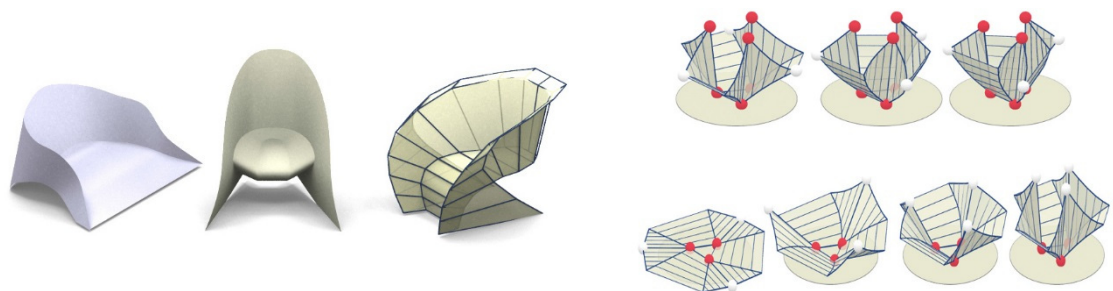


Figure 18 Interactive modelling of developable surfaces using continuous NURBS representation. All images Tang et al. (2015)

### 2.3.1 Geometric techniques for CCF

These techniques are derived from the properties of developable surfaces collated from aforementioned pioneering efforts in differential geometry (Section 2.1) and/or the long history of origami. Such methods are of interest to the current research because they provide useful means of verification and aid geometric intuition for the design of CCF surfaces. Two of the often used and cited techniques are summarised here.

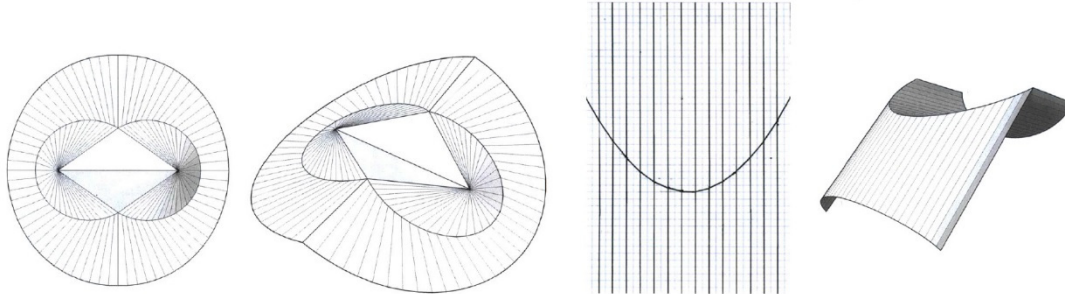


Figure 19 Methods to geometrically construct CCF geometries. Left and Right: Method of refraction and Method of reflection, as rediscovered by Koschitz (2014) from David Huffman's original notes. Images taken from Koschitz (2014)

Doctoral thesis of Koschitz (2014) studied the historic work of David Huffman and subsequently produced a taxonomy of design techniques that could be applied to produce planar layouts of fold-curves and *rulings*. Such tools are therein referred to as *design gadgets*. These techniques are based on the observations and proofs of David Huffman with regard to the relationship between rulings on one side of a curve line to those on the other. Koschitz categorizes these methods using an optical terminology of *gadgets of reflection* and *gadgets of refraction* (Figure 19). Such a terminology is used because David Huffman himself used optical analogies to study and predict the layout of *rulings* and the resultant 3D surface upon folding. These analogies provide an intuitive understanding and method whilst remaining consistent with many of the more formal proofs developed by Huffman and others (Section 2.1). As such this effort from Koschitz is one of the most exhaustive and valuable documents for designers and researchers alike.

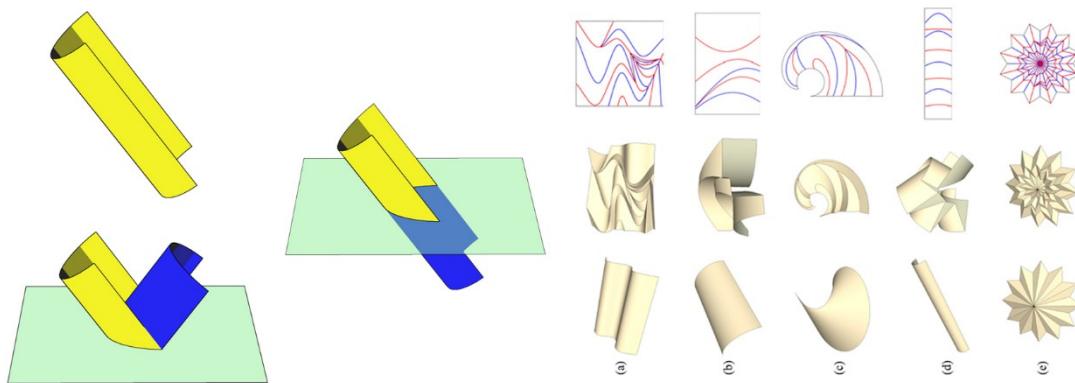


Figure 20 Methods to geometrically construct CCF geometries Left: Method of invert reflection based on reflective principle of CCF geometry (Useful theorems and terminology for CCF design). Images taken from Mitani & Igarashi (2011)

Mitani & Igarashi (2011), build on the reflective principle of rulings on either side of the curve-created-fold (Useful theorems and terminology for CCF design), and propose an interactive system to develop multi-crease geometries (Figure 20). It can be noted that their method is restricted to the use of planar curves as fold-curves and this stems from observations from differential geometry that a planar curve is a stable physical configuration (Fuchs & Tabachnikov 1999)

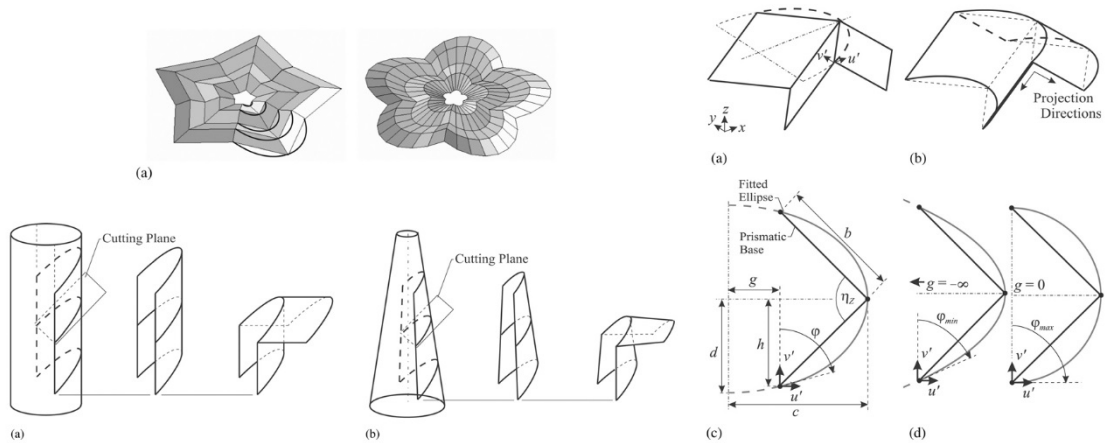


Figure 21 Procedural generation of curve-creased geometries from their straight-line or prismatic counter-parts. Images taken from Gattas & You (2014)

In context of methods that aid the derivation smooth and parametric CCF surfaces from a user-specified coarse input, it is worth mentioning the very recent contribution from Gattas & You (2014). They propose simple procedural generation of curve-creased geometries from their straight-line or prismatic counter-parts (Figure 21). Their method also builds on the method of reflection in that instead of the invert-reflection of the rulings, they first fit an (analytical) ellipse through the rulings and subsequently reflect the elliptic surface. Thus their method, although currently restricted to certain *Miura*-typologies, has the benefit of simple parameterization of CCF geometries in addition to the intuitive-ease of use of the geometric techniques. The parameterizations lend themselves to carry forth more analysis of CCF geometries, such as structural simulations and also simulating CCF geometries by building on the simpler-to-simulate prismatic variants.

## 2.4 Approximating approaches

Prior to the development of exact methods of representing developable surfaces, several methods have been proposed to approximate input triangular meshes with developable surfaces. Such methods have been usually applied in paper-craft, clothing design and origami. Some of the prominent attempts are described below, to provide an overview of the algorithmic themes used in such a design and physically based setting.

### 2.4.1 Discrete approximations

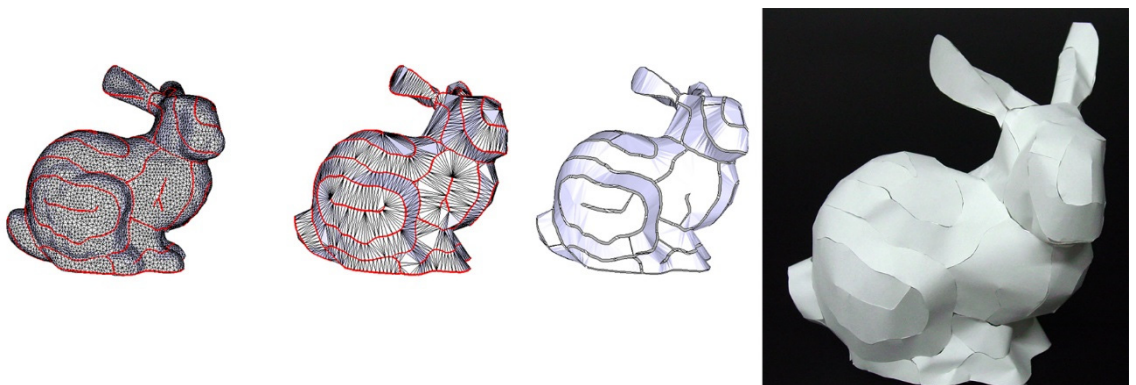


Figure 22 Discrete approximation of given triangular meshes via partitioning of meshes and subsequent simplification. Images taken from (Mitani & Suzuki 2004)

Mitani & Suzuki (2004) approximate input triangular meshes with multiple strips of developable triangulations. Their algorithm is fairly involved, starting with the identification of

geometric features such as sharp creases and ridges, the subsequent partitioning of the mesh into zones with boundary curves, constrained re-meshing of the zones and the use of edge-collapse and other mesh-editing techniques to recover *ruled* meshes for each of the zones (Figure 22). It can be noted that their method is not reliant on the actual (Gaussian) curvature of the input mesh. They cite the difficulty in partitioning meshes based on curvature as the reason to avoid such an approach.

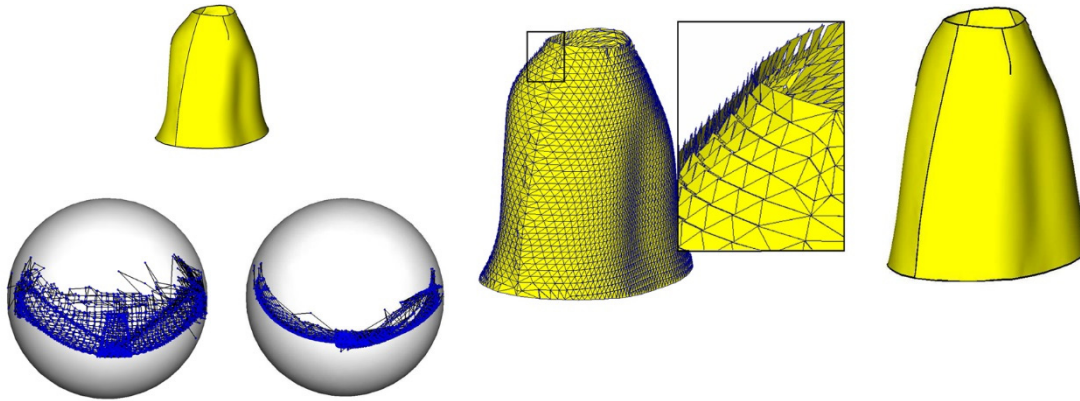


Figure 23. An iterative and perturbation algorithm to approximate developable surfaces, developed for use in the design of garments where material tolerances allow for soft-constraints of developability. Images from Decaudin et al. (2006).

Decaudin et al. (2006) propose an alternative method that exploits a particular property of developable surfaces - that of their Gauss map of unit normals being one-dimensional. Their method is inspired by the moving-least-squares method to locally approximate scattered data with a polynomial curve, which nonetheless guarantees global continuity. They thus locally approximate an analytical developable surface and subsequently (best-fit) transform each of the triangles of their initial collection of triangles onto these patches. They then proceed to glue all the triangles together subject to continuity constraints to produce a mesh (Figure 23). The algorithm is iterative and the process is repeated until certain measures of the unfolded layout of the mesh fall within acceptable tolerance. It can be noted that their application was focussed towards the design of garments, where such tolerances are indeed allowed due to the slight stretching capacities of fabric.

## 2.4.2 Continuous approximations

As with the direct modelling approaches using parametric surface presentations described previously, approximating input surfaces with developable equivalents amounts to defining constraints on the basis polynomials of (Bezier) surfaces and the subsequent constrained optimisation-solutions for the positions of the *control-net*. For further information, the reader is referred to Pottmann & Wallner (1999) who describe a linear approximation algorithm for developable NURBS surfaces, Wang et al. (2004) who apply their previously described discrete algorithm to the continuous case, and Chen et al. (1999) who fit a NURBS surface to scattered data derived either from scans or a mesh.

## 2.5 Modelling and simulation of thin shells

Physically, curve-crease folding is an example of elastic and plastic deformation since the act of folding sheet material along pre-defined curved-crease-lines causes the material to also bend. It is therefore, natural to include computational modelling of deformation of thin sheets, in the survey of prior work.

Approaches to elastic-plastic deformation of surfaces are related to the idea of *elastic energy* of thin shells – an energy that measures deformation of initially curved surfaces. The discretized formulation of this energy that is adopted is based on the requirement of the application. Such requirements, in the context of the current research, could be broadly categorised into materially realistic *simulation* of deformation of geometry and physically plausible, *modelling* of geometry. The various formulations used in computer applications of either kind, stems from the seminal work of Terzopoulou et al. (1987). They describe the derivation of such an elastic deformation energy from the fundamental forms of surfaces. More importantly they describe its incorporation into a particle or node based model to simulate deformation of thin surfaces. In essence, such a model consists of two terms to capture stretching and bending of the surface respectively:

$$E_{shell} = E_{membrane} + E_{bending} \text{ (Grinspun 2003; Botsch \& Sorkine 2008).}$$

Discretization of the elastic energy in time-dependent, *simulation* applications tend to be non-linear and include terms that capture various physical aspects of a material deformation. Approaches in *modelling* applications, especially interactive modelling, on the other hand, tend to focus on minimising the elastic energy and solving for the rest-state of the deformation process (Botsch & Sorkine 2008). Additionally, such applications tend to prefer simplified and linearized versions of the energy, in order to achieve the fast computation times required in modelling applications. This distinction is useful, since the research of this dissertation is a hybrid between the two in that it uses a time-dependent simulation framework whilst also incorporating discrete, geometry based measures. Thus, a synopsis of prior works is presented under algorithmic themes in addition to this distinction. The section includes pertinent aspects of developable surfaces and origami, and ends with seminal application in curve-crease folding.

### 2.5.1 Isometric deformation

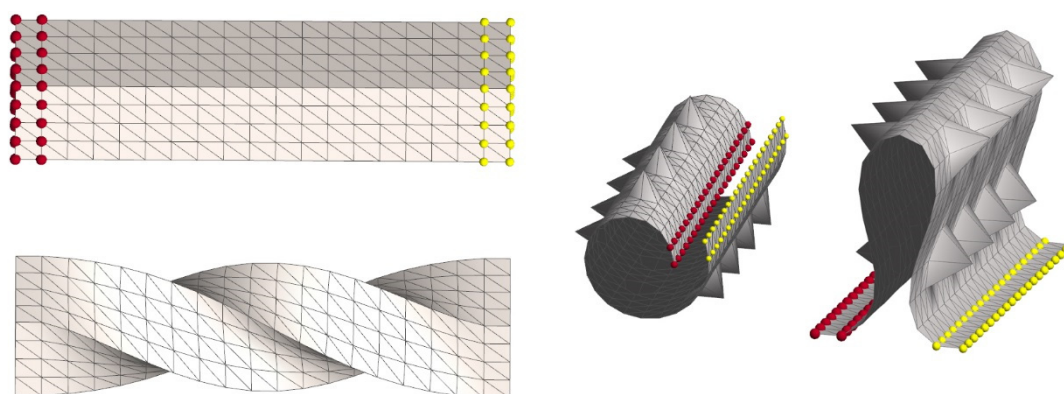


Figure 24 Examples from the iterative method of as-rigid-as-possible deformation (ARAP). Incorporation of user-inputs of fixed regions (Red) and editing handles (yellow) can be noted. Images from (Sorkine & Alexa 2007)

Sorkine & Alexa (2007) present a method of as-rigid-as-possible (ARAP) deformation on input meshes that exemplifies the usage of the *elastic energy* as described above. Their modelling-focused method aims to minimise the elastic energy between the rest state of the input mesh and the deformed state that the user can direct (Figure 24). An iteration of their algorithm consists of two steps: A first localised step aims to minimise the bending term by finding the best rigid (translation and rotation) transformation of each cell in the input mesh to the corresponding cell in the deformed state. The second, global step aims to minimise the stretching energy by performing a Laplacian smoothing of the vertex positions of the deformed mesh. This method, as stated, above produces physically plausible deformations which are not necessarily physically exact. However, it could be used to intuitively explore the bending deformation of planar rest states of sheets.

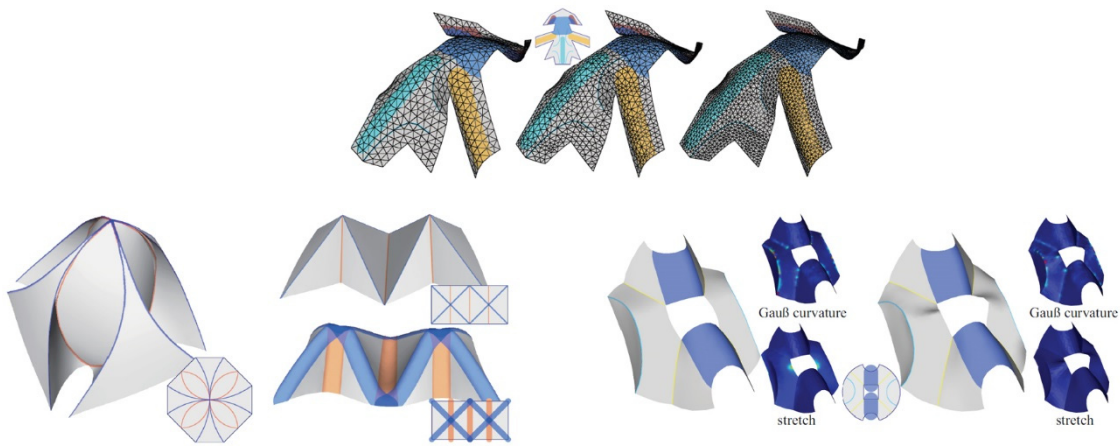


Figure 25 Examples from modelling method called *soft-fold* that allows exploration of soft and hard crease lines on flat sheets, and its implication on the folded state. Images from Zhu et al. (2013)

A recent contribution by (Zhu et al. 2013) called the *soft-fold* method, proposes an interactive editing system to compute multiple user-directed folds (Figure 25). Whilst the overall algorithm is vastly different, it has some aspects common with the previous method. Their method assumes isometric folding, i.e. they assume that the stretching term of the Elastic energy (Section 2.5) is absent. Subsequently the problem mutates to finding the best-fit rigid transformation from a flat rest state to a deformed state inferred from user-input.

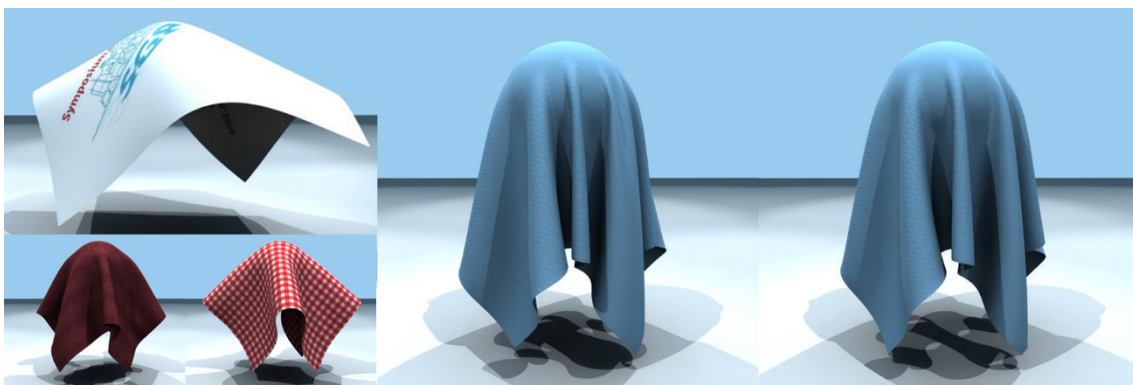


Figure 26 A method based on isometric bending used to simulate thin, sheet and inextensible materials. Left: Results for paper, cloth and rubber. Right: comparison of isometric method to popular, non-linear model. Images from Bergou et al. (2006)

Bergou et al. (2006), have previously proposed a similar isometric bending model for the purposes of simulating sheet materials such as cloth and paper (Figure 26). They also propose a quadratic bending energy formulation involving the Laplacian matrix that is similar to the one summarized in the survey by Botsch & Sorkine (2008).

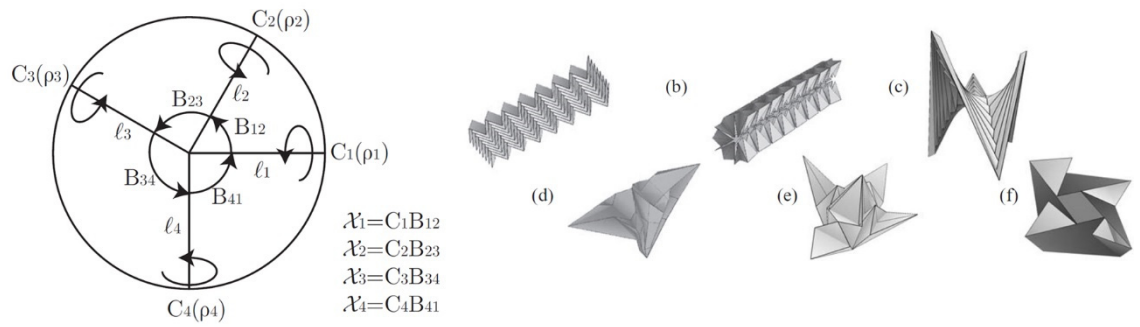


Figure 27 Method to simulate origami development and kinetic transformations. The iterative method is solves derives and solves rotational constraints and rigid transformations. Images from Tachi (2009).

Tachi (2009) proposed one of the earliest complete methods to simulate origami development and deployment. His method shares some similarity with the methods described here, even though he does not explicitly mention or use an energy based formulation. His simulation method attempts to derive rotational constraints (rigid transformations) from user-defined or assumed mountain-valley edges and fold-angles. The system of equations is iteratively updated until the corresponding rigid transformations and constraints are solved (Figure 27).

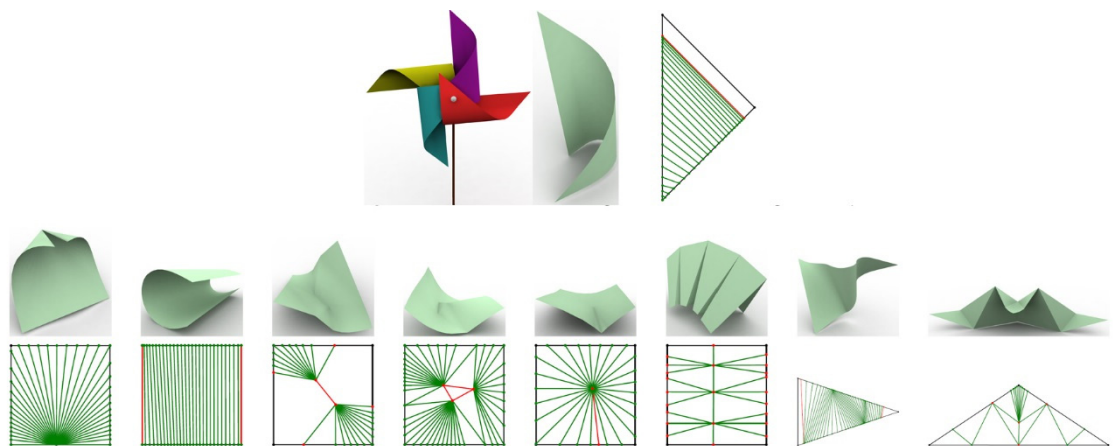


Figure 28. An interactive modelling method based that subdivides a user-defined coarse mesh before generating a smooth developable surface. Images from Solomon et al. (2012)

Solomon et al. (2012) propose an interactive modelling method to describe smooth developable surfaces. Their method receives a coarse mesh input from the user and proceeds to sub-divide it before minimising the discrete bending energy (Figure 28). They derive their own, rather elaborate bending energy to suit their discretization.

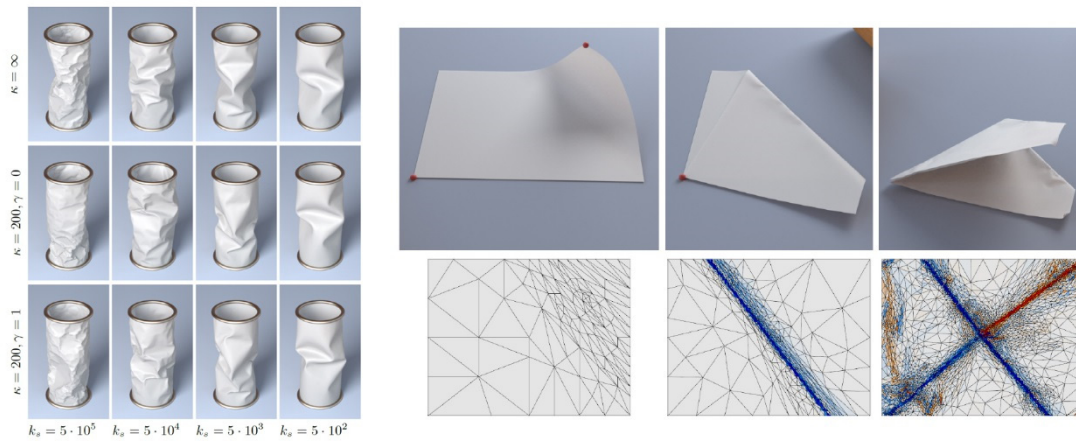


Figure 29 A materially realistic method to simulate folding and crumpling of sheet material that dynamically aligns mesh discretization along fold-lines to improve accuracy. Images from Narain et al. (2013)

Narain et al. (2013), use a fully materially-realistic, and inextensible (isometric) model of *elastic energy* to simulate the folding and crumpling of sheet material. They also use adaptive re-meshing of the sheet to align with the directions of creases and folds (Figure 29). It can be noted that their method is computationally expensive, but produces very realistic behaviours. (Schreck et al. 2015) interleave a similar physically-based simulation with a procedurally generated discrete developable surface to achieve interactive frame rates whilst simulating crumpling of paper.

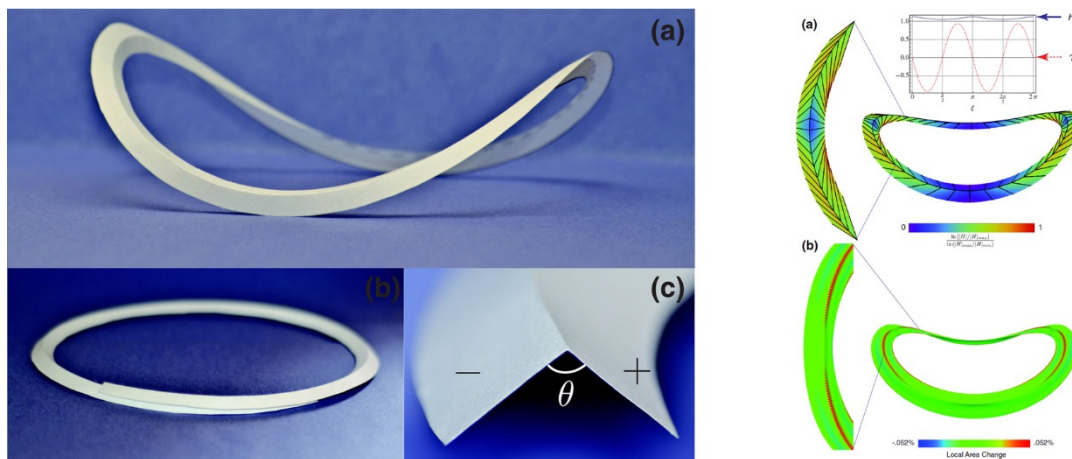


Figure 30 A physically-accurate bending model used to simulate the particular phenomena of a (closed) crease itself buckling. Images from Dias et al. (2012)

In this context, it is worth mentioning the recent (2012) contributions of Dias et al. (2012) who model a very specific example of CCF –where all the crease curves are closed as in the case of a annular strip. In addition they model, through this example, the peculiar phenomena of the crease itself buckling (Figure 30). They also derive the necessary bending and creasing energy formulations to simulate this.

Lastly, a mathematical treatment that combines several of the aspects of simulation from above, with the continuous representation of rectifiable developable from Bo & Wang (2007), and energy minimization from the *modelling* approaches is the work of Starostin & van der Heijden (2007). They develop and treat a one-dimensional variational problem to find the shape of a elastic Mobius strip and show extensions of the their approach to predict crumpling of paper and suggest applicability to similar phenomena of draping the inextensible material of fabric.





Figure 31 A mathematical treatment to find the shape of a Developable Mobius strip. Images from (Starostin & van der Heijden 2007)

### 2.5.2 Curve-crease-folding

The prior works summarised in this section, are the current state-of-the art in terms of the discrete modelling and simulation of curve-crease-folded geometries. They synthesize the various algorithmic themes and geometric properties previously described, into methods for use on CCF geometries. They have thus, made the most direct contributions to the research presented in this dissertation.

$$E_i = w_i \|\mathbf{n}_i - \mathbf{n}_{i-1}\|^2.$$

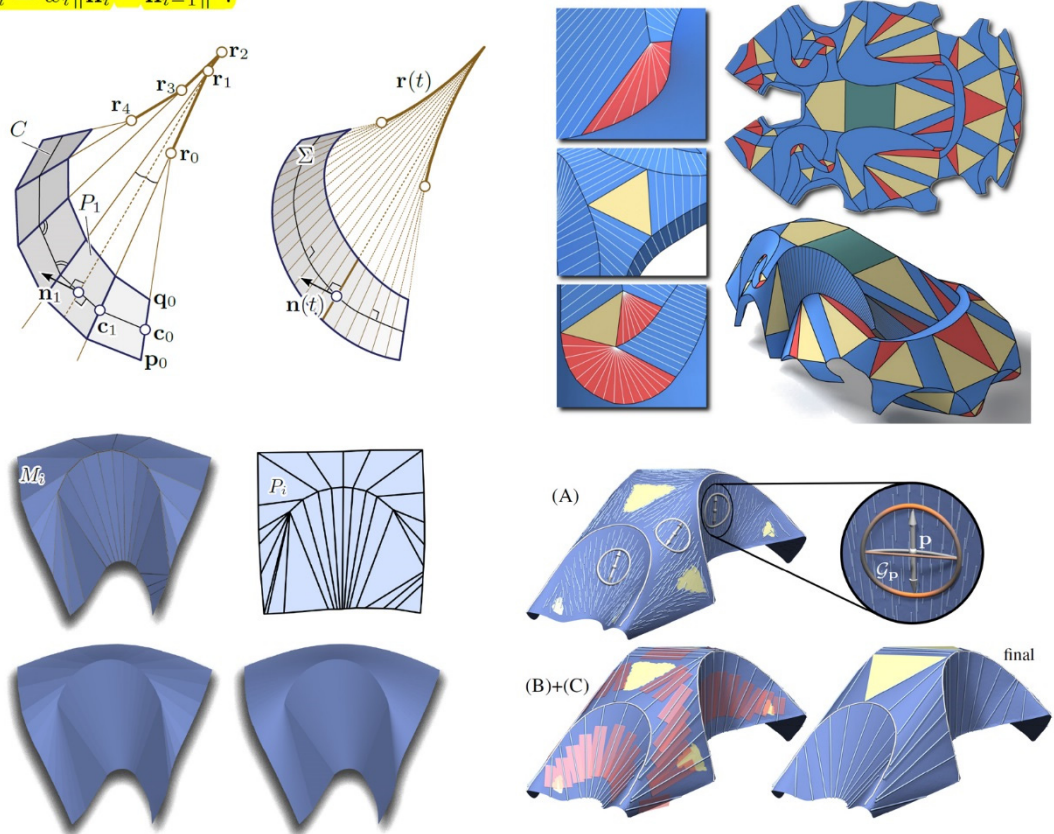


Figure 32. A modelling method for CCF geometries based on pre-dominantly quad-based discretization and minimization of associated bending energy. Smooth rulings as limits of discrete rulings, and the smooth developable surfaces assembled by combining patches of torsal developable surfaces can be noted. Images from Kilian et al. (2008).

Kilian et al. (2008) proposed a method to model CCF geometries using a discretization composed of quadrangular facets. They, as others previously (Liu et al. 2006; Weiss & Furtner 1988), propose the use of planar-quad strip (PQ strips) as discrete representation of a *torsal ruled surface*, which are known to be developable. They formulate a quadratic bending energy (Figure 32 Top Left) to suit their quad-based discretisation and propose optimisation methods to model isometric bending of paper. They also show that the limits of the *rulings* in the discrete representation (PQ mesh) are the *rulings* of the developable surfaces (Figure 32 Top Left). They

extend the discrete strip representation to composite meshes made up of PQ strips and planar triangular facets to represent curve-crease-folded geometries. They derive the necessary planarity and developability conditions for such a mesh to become curve-crease-foldable. They then propose an algorithm, albeit elaborate, to accurately recreate scanned models of physical CCF geometries. This optimisation algorithm starts with a collection of triangles (Figure 32 Bottom Left) and proceeds to iteratively perturb the vertices towards achieving curve-crease-foldable meshes.

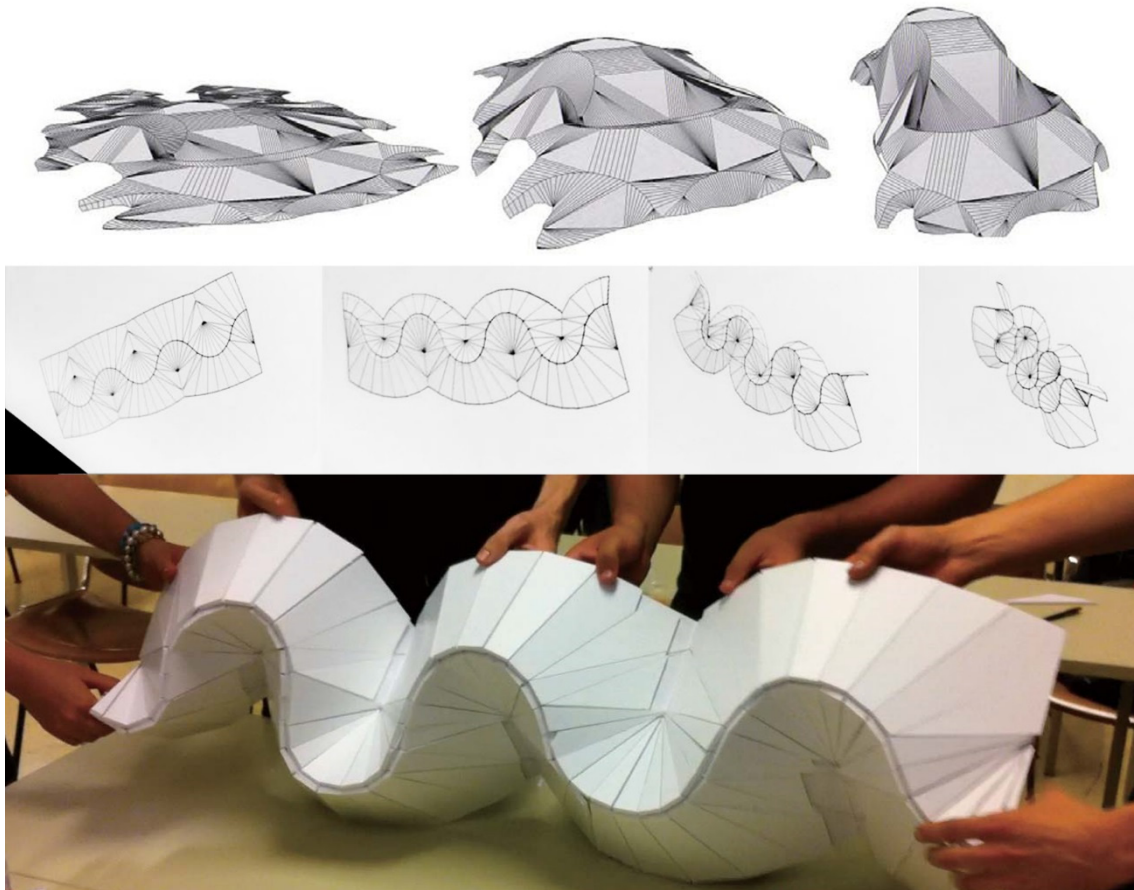


Figure 33 A intuitive and digital-physical hybrid strategy to design CCF geometries. Images from Tachi & Epps (2011).

Tachi's contributions to the simulation of prismatic origami have been mentioned previously. Tachi & Epps (2011) have subsequently shown that CCF geometries can be intuitively designed by applying simulation methods from prismatic origami to the PQ discretization proposed by Kilian et al. (2008). They also propose an intuitive hybrid physical – digital design technique to design CCF geometries (Figure 33): they generate the planar layout of PQ meshes by observing the physical model, and subsequently digitally extend the constraints and parameters to explore the space of three dimensional possibilities.

## 2.6 Planar development

### *Flattening*

Several of the algorithms surveyed are iterative. It follows that successive iterations are stopped when the error is within numerical tolerance. This then implies that the resulting mesh is only developable within tolerance. In other words, the resultant mesh is quasi-developable. As with the previously indicated applications of quasi-developable surfaces (Section 2.4), they are subject to the tolerances afforded by the method and material of physical reproduction of the digital surface. The computing of the planar development of a mesh or *flattening* of the mesh

has in itself, a vast amount of literature and prior work. Only a brief synopsis, as applicable to current research is described here.

Flattening of a mesh amounts to computing two-dimensional coordinates for every vertex in 3D. Any function that maps the three-dimensional Cartesian coordinates of a mesh-vertex to a two-dimensional (planar) domain can thus be considered. It is also desirable, for such mapping-functions to produce unique mappings or to be *bijective*. Such mappings are often used in computer-graphics in the reverse sense i.e. to apply (2D) images or textures unto (3D) meshes (Vallet & Lévy 2009). Similarly, mapping easy-to-draw 2D quadrangular meshes, unto the 3D-triangulated mesh is an often used technique in (quad) *re-meshing* applications (Hormann & Greiner 2000). Further, such mappings are used in a more true-sense, in the production of maps or cartography (van Wijk 2008; Hurdal & Stephenson 2004). Lastly, the bijective property is often useful in reducing three-coordinates to two and thus finds use in mesh-data compression and transmission (Gu et al. 2002).

### *Parameterization*

By virtue of assigning two real values (parameters) to each vertex on the 3D mesh, such methods are often referred to as *parameterization* in computer-graphics literature. There is extensive literature related to this process of computing a parametric mapping between a discrete mesh and its isomorphic and planar counterpart. The reader is referred to Desbrun et al. (2002) for various methods used and their limitations. Additionally, Sheffer et al. (2006) provide a more recent survey on mesh parameterization. Most parameterization algorithms could be classified in accordance to the nature of input meshes that they are designed to operate on – single-patch or multi-patch with seams between patches (Panozzo & Jacobson 2014). This research restricted itself single patch cases.

### *Isometry*

In general, a completely accurate *parameterization* can exist only in the case of *exact* developable surfaces, in which case the flattening can be achieved by merely re-orientating triangles from 3D unto the plane. In the all other cases, methods aim to *minimize* the deviations between the 3D mesh and its planar counterpart. The common steps in such a process are to *pin* a few vertices (usually the border) unto the plane and subsequently to formulate and minimize a measure of distortion. Some methods (*conformal* mapping), aim to minimize angular deviations (Lévy et al. 2002), others to preserve areas (*authalic* mapping) (Floater 1997). The pinning being essential to prevent the trivial case, where the vertices collapse to a point. To physically reproduce CCF geometries then, the mapping would need to be *isometric* i.e. both *conformal* and *authalic*. In keeping with the DR scheme, *conformal* or *authalic* parameterizations could be achieved by applying forces to vertices along the direction of respective gradients. These gradients at each vertex can be computed from the position vectors of the 1-ring neighbourhood of vertices, as formulated in Desbrun et al. (2002). It may be worth noting that the methods that Desbrun et al. (2002), Pinkall & Polthier (1993) propose suffer from arbitrary boundary parameterization. Alternatively, the iterative methods of as-rigid-as-possible (ARAP) parameterization (Liu et al. 2008) and most isometric parameterizations (MIPS) method proposed by (Hormann & Greiner 1999) produce more ‘natural’ boundaries albeit at the expense of computational time. Both methods are recent, robust and relatively elaborate to implement.

## **2.7 Summary of Computational methods**

### **2.7.1 Interactive methods**

Most of the precedent projects and available literature on design methods highlight the difficulty of developing an intuitive, exploratory digital-design method to generate feasible 3D

geometries. The survey of methods included both the simple and common method – the method of reflection (Mitani & Igarashi 2011) and the involved Planar-Quad-meshes and optimization-based method (Kilian et al. 2008). Most methods, including the two above, present difficulties of incorporation within an intuitive, edit-and-observe method of design; The first one proving difficult to explore a variety of generalized solutions free of prior assumptions and the second one being elaborate involving scanning of physical paper models, proprietary optimization algorithms.

### 2.7.2 Choice of representation

The survey also showed particularly interesting recent developments of interactive tools that operate on user-specified coarse linear piecewise complexes (Solomon et al. 2012) and spatial curves(Bo & Wang 2007) to produce smooth developable surfaces. In both cases the user-specified input geometry also provides the handles to manipulate the generated geometry. Perturbation based methods(Wang & Tang 2004; Decaudin et al. 2006) were reviewed and their algorithmic mechanisms surmised. Several energy minimization methods used to find developable or curve-crease foldable rest-states were summarised (Kilian et al. 2008; Solomon et al. 2012; Bo & Wang 2007; Sorkine & Alexa 2007); These algorithmic themes taken together, are in alignment with established benefits of subdivision surface based modelling paradigm(Section 1.2.2) in architectural form-finding (Shepherd & Richens 2010; Bhooshan & El Sayed 2011) and the application of dynamic relaxation (Barnes 1999) techniques on subdivision surfaces to design and fabricate minimal mean curvature surfaces(Section 1.2.1) - so called minimal surfaces (Bhooshan & El Sayed 2012).

The survey also highlighted a consolidating trend in computational geometry and design of thin-shells based on *elastic deformation energy* (Section 2.5). Simulation-minded applications use physically realistic deformation models, whilst modelling applications utilise a physically plausible and approximate model. In the specific case of this research, both are relevant due to the objectives of interactive and exploratory design as also incorporating realistic manufacturing constraints such that the designed geometries can be re-produced physically.

Additionally, the popular and seminal constructive-geometry methods were also described. These can provide an easy means of verification. Importantly they provide insight and intuition for generating and manipulating appropriate topologies of rulings during design.

## 2.8 Relevance to current research

The synopsis above (Section 2.7), which was expanded in the preceding survey (Section 2.1), explains the following choices that were made for the research presented in this document.

The natural choice for a computational representation is a discrete one, as reinforced in Section (2.7.2). The research also sets out an explicit intention to perturb, input 3D geometries to find feasible CCF geometry similar to Wang & Tang (2004). This is in contrast to finding the folded state of a 2D input mesh. It may be noted that the optimization-based method proposed by Kilian et al. (2008)( Section 2.5.2) does in fact solve this problem, albeit it is considerably more difficult to implement. The method developed here is intended to be easier to implement and extend. However, unlike their method, would require the user to provide an initial mesh with appropriate topology. The research will use a pre-dominantly quad based discretization similar to their method and utilize the geometric implications suggested by their formulation of bending energy. The intention subsequently, is to follow energy minimisation strategies that are amenable to the modelling paradigm (Section 2.5). This is further described in the next Chapter (Sections 3.1 and 3.2). Specifically, computational techniques based on Dynamic relaxation (DR) (Section 1.2.1) will be used to iteratively perturb the surface towards minimal Gaussian

curvature and local planarity, which form the necessary conditions for CCF geometries (Section 3.2).

The research will develop simple procedural methods involving known mesh-operations that can be used to produce an initial mesh (Chapter 6). This will constitute the exploratory mechanism for CCF geometries i.e. a known CAD and edit-and-observe paradigm based on subdivision surfaces (Section 1.2.2) is augmented to aid exploration of CCF geometries.

### 3 COMPUTATIONAL METHOD

#### 3.1 Discrete representation

As can be discerned from the previous survey of methods for the computational description of origami, developable surfaces and curve-crease-folding, there are two commonly used representations for the corresponding geometries : continuous and discrete ( triangular or quadrangular meshes). As explained in the preceding section (Section 2.7.2 & 2.8), a choice was made to follow a discrete approach in the current research.

Focusing on developable surfaces, the survey could be summarily viewed along the lines of distinction made by Solomon et al. (2012) : approaches that *synthesize* developable surfaces to suit given constraints, *approximate* given (scattered) data with developable surfaces or smoothly and isometrically *deform* known developable surfaces towards other states. *Synthetic* approaches either exploit geometric properties of developable surfaces (*Constructive* methods: Section 2.3) or solve for the rest state of a deformation process (*Modelling* methods: Section 2.5). In the latter, subsequent to a choice of representation, a suitable (discrete) formulation of elastic energy is derived, and optimization techniques are employed to minimize such an energy to arrive at the desired rest and developable state of the surface. Such rest-states can further be classified as *Quasi* and almost developable or *exact* and fully developable. *Quasi-developable* surfaces are usually employed in applications that are sufficiently tolerant on their requirements of developability. These include the design of clothing and soft-toys due to the slight stretching capacities of the fabric (Julius et al. 2005), application of texture maps on meshes due to the requirements of avoiding only visually perceptible stretching of textures (Lévy et al. 2002), in the design of ship-hulls due to possibility of heat-forming marginally non-developable surfaces(Pérez & Suárez 2007) etc.

*Exact* discretization on the other hand are usually required if the digital models are to be faithfully physically reproduced such as in CAD applications, cartography etc. This is also an objective of the current research. Thus there are several discrete representations - *exact* and *inexact* - for developable surfaces and by extension, for curve-crease folded geometries. For a comprehensive list and summary, the reader is referred to Solomon et al. (2012).The use of planar-quad strips as discrete representations for developable surfaces has been previously noted in the survey(Liu et al. 2006; Kilian et al. 2008). The extension of PQ meshes to suit CCF geometries by Kilian et al. (2008) has also been noted. A choice was made to utilize this particular discrete representation in the current research due its *exact*-ness, compatibility with subdivision surfaces (Section 1.2.2) and well-established mesh-editing paradigms (Section 2.5), easy recovery of planar development or flattening of a 3D surface etc. The necessary conditions of such meshes to become curve-crease foldable is detailed in Kilian et al. (2008) and is briefly surmised below.

#### 3.2 Necessary conditions of curve-crease-foldability

A simple, single curved-crease fold can be seen as two developable surfaces on either-side of the fold, intersecting to form the crease. Thus, if such a folded surface is discretely represented as a *manifold* mesh with two quad strips with equal number of rulings and merged coincident vertices, then the (interior) vertices that lie along the crease have to be developable. Additionally, the two quad strips themselves have to be developable. Taken together, they form the necessary conditions for such a mesh to be curve-crease-foldable. This implies that the *faces* of the mesh have to be planar (PQ mesh) and the crease vertices have to be developable or the Gaussian curvature measured at such vertices has to be zero. This reasoning can be applied to more complex assembly of quads-strips, including hybrid meshes with triangles and quads (Figure 34).

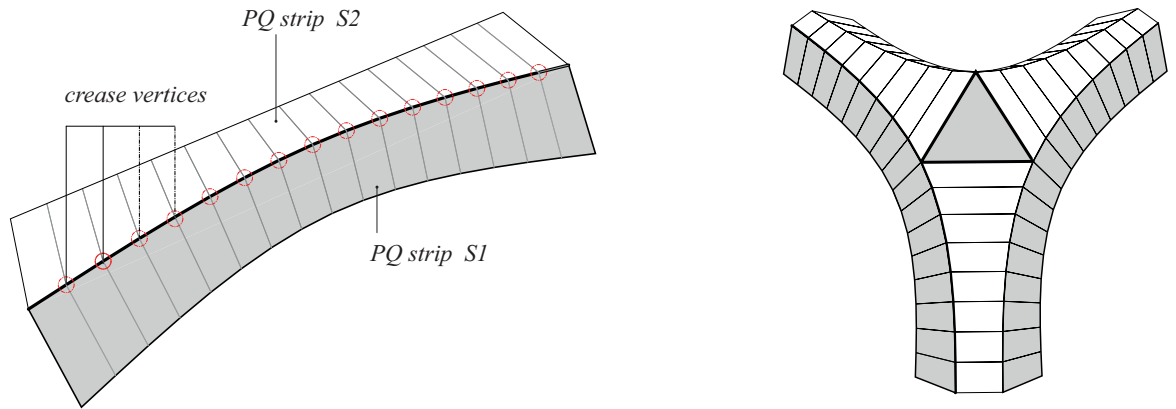


Figure 34 Planar quad strips as discrete representation of developable surfaces

### 3.2.1 Discrete measures and constraints

#### Gaussian curvature

One of the well-established discrete measures of Gaussian curvature is proportional to the difference between  $2\pi$  and the angle-sum around a vertex. It is also proportional to the area of the incident faces (Aleksandrov & Zalgaller 1967; Meyer et al. 2003). Angle-sum is computed by adding the angles between cyclical pairs of edges meeting at the vertex. Thus the Gaussian curvature will be zero if the angle-sum is  $2\pi$ , since the case where the face-areas collapse to zero indicates a *degenerate* mesh. Such a measure of the *angular defect* makes intuitive sense if we consider the case when the vertex and its edges are *flattened* unto a flat sheet. The angles, in such a case would sum up to  $2\pi$  by virtue of being on a plane. Stemming from such an intuition, vertices where the angle-sum is in excess of  $2\pi$  or the angular-defect is negative, are called *Hyperbolic* vertices. Such vertices and their edges cannot be isometrically flattened to a plane without having to tear the *disk* (Figure 35). Similarly, vertices where the angular defect is positive are called *Spherical* vertices and they would leave a gap when flattened or the edges would have to stretch to overcome the gap. Lastly, perfectly developable vertices are called *Euclidean* vertices.

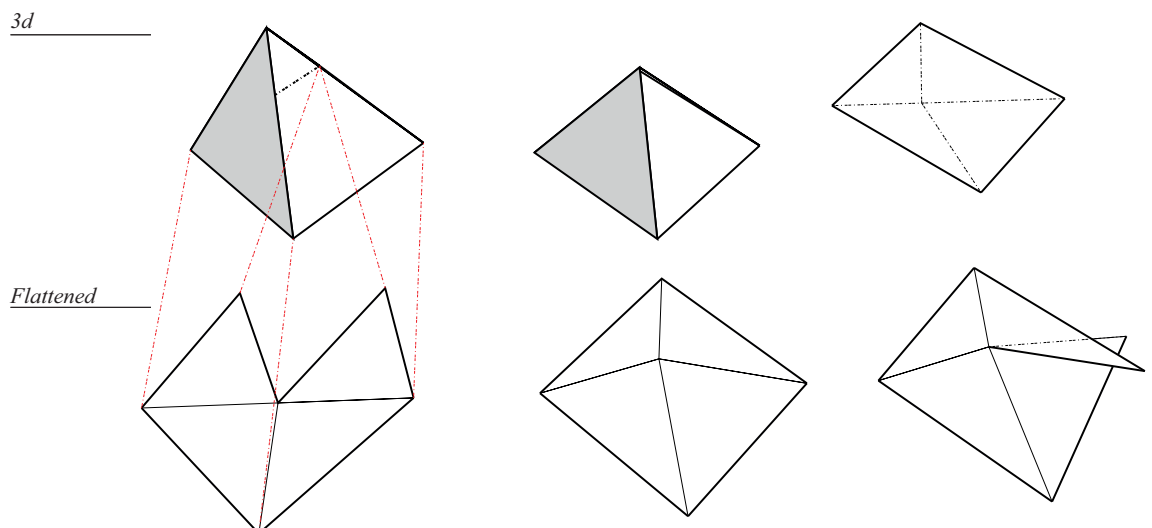


Figure 35 Classification of vertices based on their Gaussian curvature : Left to Right : Hyperbolic, Spherical and Euclidean vertices

## Planarity

Kilian et al. (2008) also formulate a discrete bending energy to suit their PQ discretization. This energy, is locally proportional to magnitude of the vector difference between the unit-normal associated with each the two faces incident on an edge (Eqn 1). These normals associated with each face are the normals of the planes spanned by the two ruling-edges of the face (Figure 36). The total energy is the sum of such magnitudes across all edges of a mesh. It can be noted that this formulation is different from that for triangular meshes (Grinspun 2003) which is proportional to the dihedral angle across the edge. It can be seen that, locally this energy is minimal when the normal of the two faces are co-linear or in other words the two faces are co-planar. Thus the planarity requirement is also in alignment with developable surfaces having minimal bending energy or that they have the same bending energy as a planar sheet.

$$E_i = k_i \| \mathbf{n}_1 - \mathbf{n}_2 \|^2 \quad (1)$$

Where  $E_i$  is the bending energy across an edge  $i$ ,  $k_i$  is a constant, and  $\mathbf{n}_1$  and  $\mathbf{n}_2$  are normals associated with the two faces incident on edge  $i$  (Figure 36)

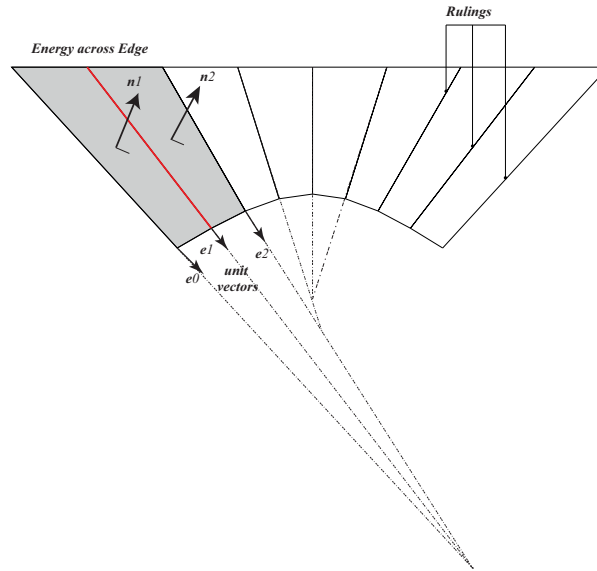


Figure 36 Quad strips and associated bending energy (Eqn 1)

Thus the two conditions that a pre-dominantly quad mesh needs to satisfy for it to become curve-crease-folded are that the Gaussian curvature or angular-defect of interior and crease vertices is zero and that the mesh faces are planar.

### 3.3 Method Overview

Given the representation and the constraints, the proposed design-friendly method essentially involves the use of various mesh operations to describe a coarse and predominantly quad-faced mesh (*low-poly*) with an appropriate topology (Figure 37) and subsequently finding the *nearest* mesh that satisfies the previously described conditions. In other words, given an input mesh of appropriate topology of rulings (Chapter 6), the method distils to a minimization problem, stated as below:



$$\left\{ \mathbf{x} \mid x_i \in \mathbf{R}^3, \operatorname{argmin} f(\mathbf{x}), f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}) = \sum_{i=0}^n g_i + \sum_{j=0}^m \sum_{k=0}^o d_k \right\} \quad (2)$$

Where the mesh has  $n$  vertices and  $m$  faces each made up of  $o$  vertices,  $\mathbf{x}$  is an  $n \times 3$  matrix of vertex positions,  $g(\mathbf{x})$  measures the sum of angle defects  $g_i$  at each vertex and  $h(\mathbf{x})$  measures sum of the distances  $d_k$  of each vertex of a face to the corresponding best-fit plane.

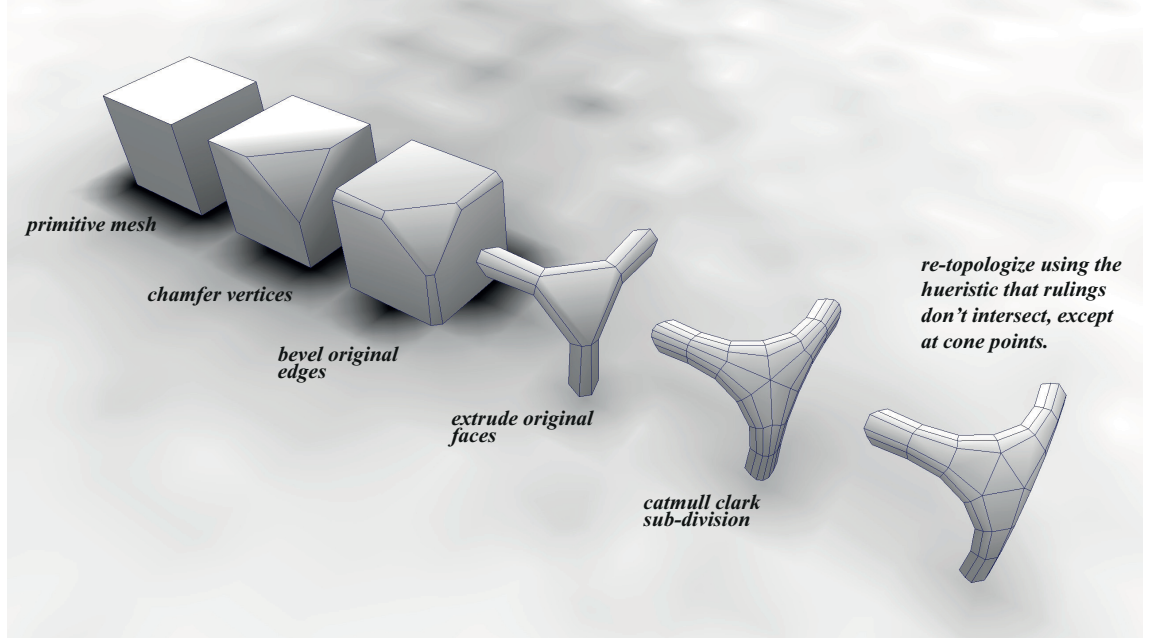


Figure 37 Use of Conway operators (Hart n.d.) to describe a coarse and predominantly quad-faced mesh

### 3.3.1 Iterative and local methods

The minimization problem stated above could in theory be solved to its *minima* or the rest-state of curve crease foldability. However common optimization techniques prefer to use quadratic functions, because their minimization involves solving a linear system of equations, which are computationally inexpensive to solve. Often then, higher order functions are approximated with quadratic terms to suit this paradigm. Hence, in popular texts on the topic, the notion of *linearized* formulation of functions can be found being applied to even quadratic functions. The amenability of Eqn (2), to such linear techniques and computational expense of the two functions stated above is described next.

#### *Planarity*

The non-planarity of quadrangular and triangular mesh faces can be geometrically visualized as the volume of the tetrahedron defined by its vertices (Poranne et al. 2013). Consequently the planarity of the face is ensured by the volume being zero. There are no additional constraints and the volume function is quadratic. Thus, this lends itself to straight-forward minimization techniques. N-sided faces on the other hand are more difficult to minimize due to higher order functions. Poranne et al. (2013) and additionally Deng et al. (2011), also show that the planarization problem for n-sided faces can be cast as a least-squares optimization subject to linear constraints (that they call distance-from-plane constraints). However they claim that such optimization problems require the use of sophisticated computational techniques, especially for high-resolution meshes. In the context of this research however, since predominantly-quad meshes are a representational requirement, the unconstrained and intuitive constraint of the volume of the tetrahedron is used. Thus, the *planarization* or minimizing the first term of (Eqn 2) is indeed linear and straight-forward.

## Gaussian curvature

Similarly, Wang & Tang (2004), have shown that the requirement of zero Gaussian curvature can be cast as a quadratic minimization problem. The quadratic function that they aim to minimize is in fact the square of the difference in lengths of the edges of the initial mesh and those of the rest-state mesh to be found. The constraints that they impose on such a minimization is that all the vertices are developable i.e. that their angular defect is zero. In other words, the constraints of their method are exactly the bending terms in the elastic energy (Section 2.5). The angular-defect constraints are actually linear in nature, since they only involve the computing of sums of angles. Thus the developability constraints are also straightforward, in terms of its minimization.

Thus, independently both planarity and developability constraints have been shown to be minimization problems, upon which known optimization techniques could be applied. However, two important features should be noted:

1. Both Poranne et al. (2013) and Wang & Tang (2004), employ *iterative* procedures to minimize the functions i.e. they start with an initial guess of the solution and incrementally improve the solution *until* it is within prescribed tolerance. If each iterative state of the mesh were to be visualized, the nodes of the mesh would appear to be *perturbed* incrementally towards their final state. This is contrast to *direct* methods that solve for the final rest in a finite number of steps. A well-known example of a direct method in mesh-processing, is the computing of minimal surfaces via minimization of mean curvature (Botsch et al. 2010) and the historically seminal work of Schek (1974) in finding minimal-nets.
2. The computation of normals in the case of Poranne et al. (2013) et al and angular defects in the case of Wang & Tang (2004) is *local*. In each iteration, these quantities are updated by traversing the faces and the nodes of the mesh respectively. It is normally preferred in optimization procedures for such quantities to be defined as manipulations of a linear system of equations i.e. as matrix operations such that all quantities can be computed at once or *globally*. For instance volumes of the tetrahedrons defined by vertices of faces can be computed by such operations whilst face-normals and angular defects are not easy to define as such.

## Dynamic relaxation

These two aspects stated above, along with those described in (Section 1.2.1) were important considerations for the choice to use a Dynamic Relaxation framework to minimize the function.

### 3.4 Perturbation

It follows from Eqn 2, that vertices could be iteratively perturbed along the gradient of the two functions -  $g(\mathbf{x})$  and  $h(\mathbf{x})$  - to minimize the error function  $f(\mathbf{x})$ ,

$$\nabla f(\mathbf{x}) = \nabla (g(\mathbf{x}) + h(\mathbf{x})) = \nabla g(\mathbf{x}) + \nabla h(\mathbf{x})$$

The perturbation of the vertices of the mesh follows a *dynamic relaxation* schema – treating the vertices of the mesh as lumped masses, applying virtual-forces along the respective gradients at the vertices, and subsequently updating the positions of the vertices by integrating the resulting Ordinary Differential Equation (ODE), using an appropriate numerical method. This process is continued until the resultant force at each vertex is zero. More details of the dynamic relaxation method and numerical aspects are described in the *Software Implementation* chapter (Section 7.2).

Thus, the DR-based method can be summarized (Figure 38) as below:

1. (Procedural) generation of input mesh.
2. Applying virtual forces of planarity and developability, and solving for the equilibrium positions of the input mesh subjected to such forces.
3. Computing a planar development (2D mesh).
4. Optionally, rectify the residual ‘errors’ by minimizing the strain energy between the 3d and 2d meshes.

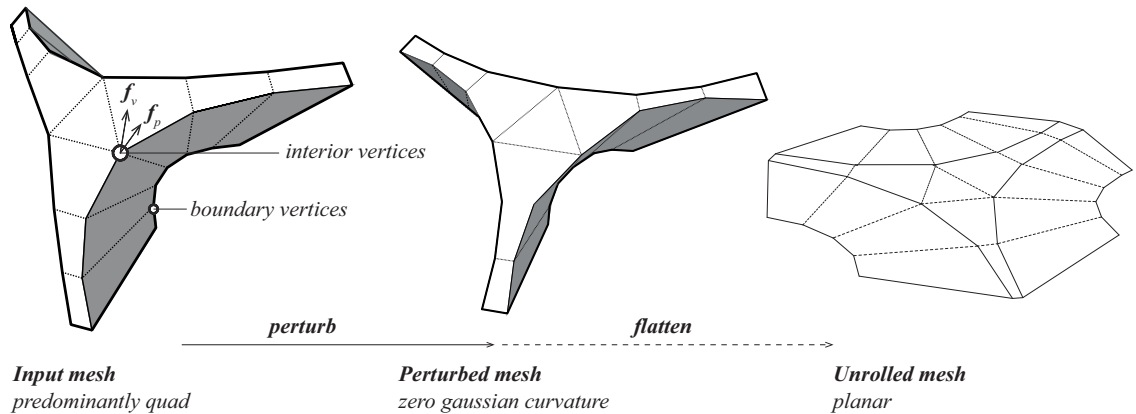


Figure 38 Overview of proposed algorithm

### 3.4.1 Gradient of planarity

#### *N-sided faces*

The planarity of the faces of a mesh is ensured by accumulating forces on the vertices of each face (Figure 39). The direction of the force is towards the best-fit plane and its magnitude is proportional to distance  $d$  of the vertex from the best-fit plane. The normal of this plane is computed as the weighted sum of the cross products of cyclical vector pairs from the node to each of its neighbours. The centroid of the 1-ring is considered the origin of the plane. An Eigen decomposition of the covariance matrix of the nodal positions of the neighbours may also be employed for this purpose (Poranne et al. 2013). However, the first approximation was found to be more compatible with the developability force (Section 3.4.2) in achieving convergence.

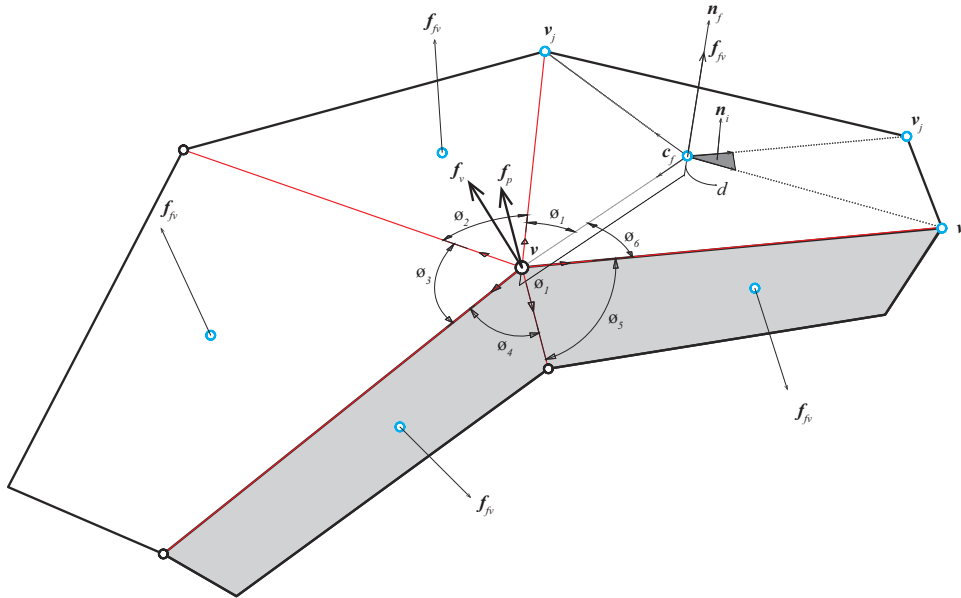


Figure 39 Quantities involved in the computation of gradients of planarity and Gaussian curvature

The normal  $N_f$  and origin  $C_f$  of the best – fit plane of each face is given by

$$N_f = \sum_{i=0}^{f_e} \mathbf{n}_i / f, \quad C_f = \sum_{i=0}^f \mathbf{v}_i / f$$

$\mathbf{n}_i$  is the normal of each triangle in the face, and  $f$  is the number of vertices of the face

Then distance  $d$  of each vertex from the best-fit plane is

$$d = (\mathbf{V} - C_f) \cdot N_f$$

And the corresponding force

$$F_{f_v} = d * N_f$$

Lastly, the accumulated planarity force at each vertex is

$$F_p = \sum_{f=0}^{vf} F_{f_v}$$

where  $vf$  is the number of faces that the vertex belongs to.

### Quadrangular faces

In the case of quadrangular faces, the gradient could also be formulated as the gradient of the volume of tetrahedron formed by the four vertices as previously mentioned (Poranne et al. 2013). Additionally, it can also be thought of as laying along the line of shortest approach between the two diagonal skew-lines of the face. This direction is the cross product of the two vectors along each diagonal (Figure 40). Both of these are easy to implement as they depend on the positions of the vertices of the face. Additionally, since pre-dominantly quad meshes are a pre-requisite for the method, both methods are particularly relevant.

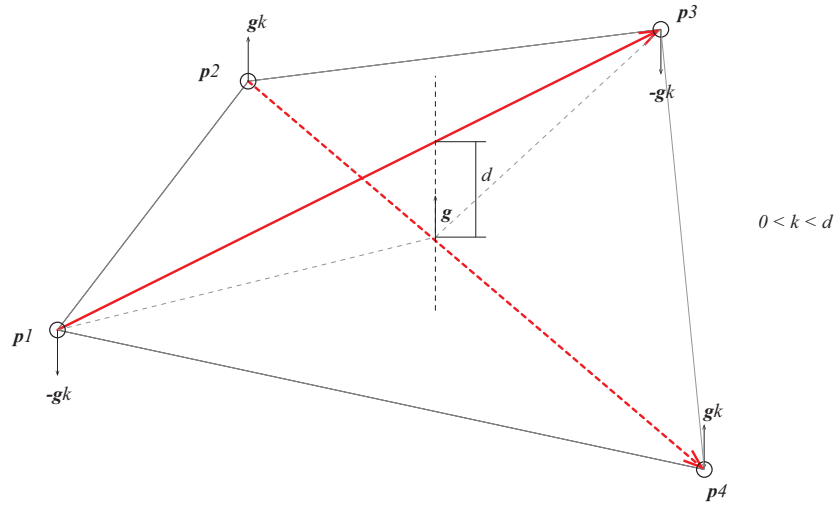


Figure 40 Gradient of planarity of a quad-face

### 3.4.2 Gradient of developability

Developability can be ensured by the presence of uniform and zero Gaussian curvature throughout the mesh. Thus the solver accumulates a force along the direction of gradient of Gaussian curvature at each vertex, with a magnitude proportional to the angle deficit (from  $2\pi$ ) at the vertex. Both an analytically computed (Desbrun et al. 2002) and a numerically computed gradient were tested, with the analytical gradient predictably converging 2-3 times faster.

$$\mathbf{F}_g = \nabla\phi * (2\pi - \sum\phi) \quad (\text{Refer to Figure 39})$$

where  $\nabla\phi$  is the gradient of gaussian curvature

The derivation of the analytical gradient is described in Desbrun et al. (2002). A geometric understanding of the gradient of an angle is described below. The setting is simplified.

Gaussian curvature is a linear function of the sum of angles at a vertex (Figure 41). Thus the gradient of Gaussian curvature is gradient of the sum of the angles, which in turn is a sum of gradients of the individual angles.

$$\nabla K = \nabla \left( \frac{2\pi - \sum\theta}{Area} \right) = \nabla (\sum\theta) = \sum_{j \in N(i)} \nabla\theta_j$$

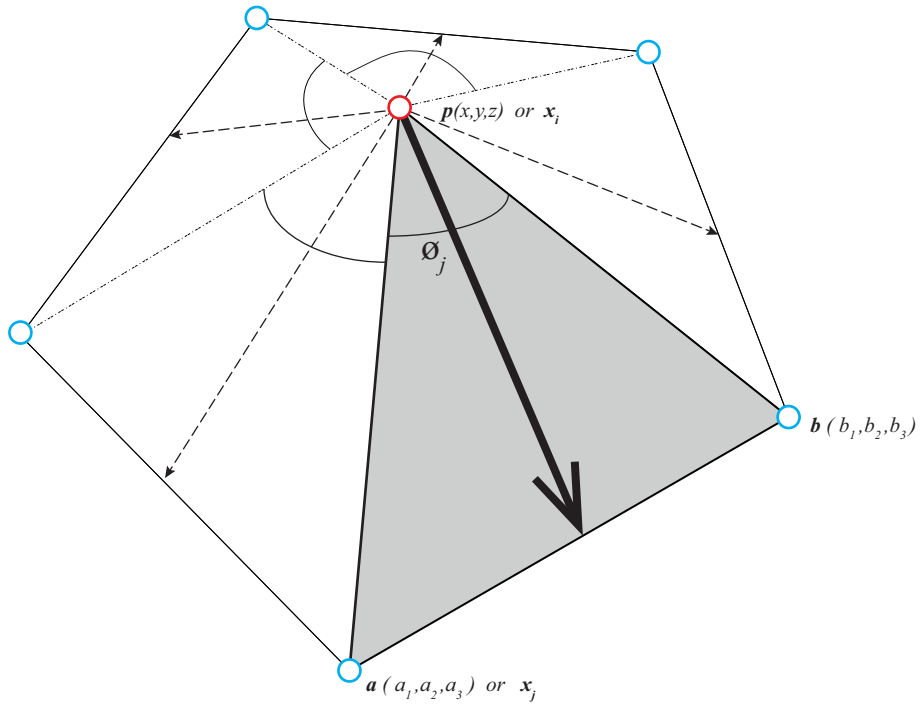


Figure 41 Geometric understanding of the gradient of Gaussian curvature

Therefore looking specifically at one of the individual angles -  $\text{angle } \widehat{APB}$ , and assuming that the 1-ring edges are all unitised, it can be shown that the gradient of  $\phi_j$  w.r.t position  $\mathbf{p}$  is,

$$\nabla(\phi_j) = \left( \frac{(\mathbf{a}+\mathbf{b})}{2} - \mathbf{p} \right) * k, \text{ where } k \text{ is any constant}$$

This is a vector in the direction from the vertex to the mid-point of the edge AB. This makes intuitive sense, in that the angle between the two edges meeting at the point in question, will be the largest if the point is co-linear with the other two i.e. they all lay on a straight line. The quickest way to get towards the maximum possible is thus move along the shortest path to the mid-point between the two fixed points.

By extension then, the gradient of the sum of angles around a vertex is a weighted average of all such vectors pointing to the mid-points of edges. This implies the gradient should be pointed towards a weighted average position of the 1-ring vertices. This is very similar to the Laplacian vector (Figure 42) which points towards a weighted average and is in alignment with the normal of the surface.

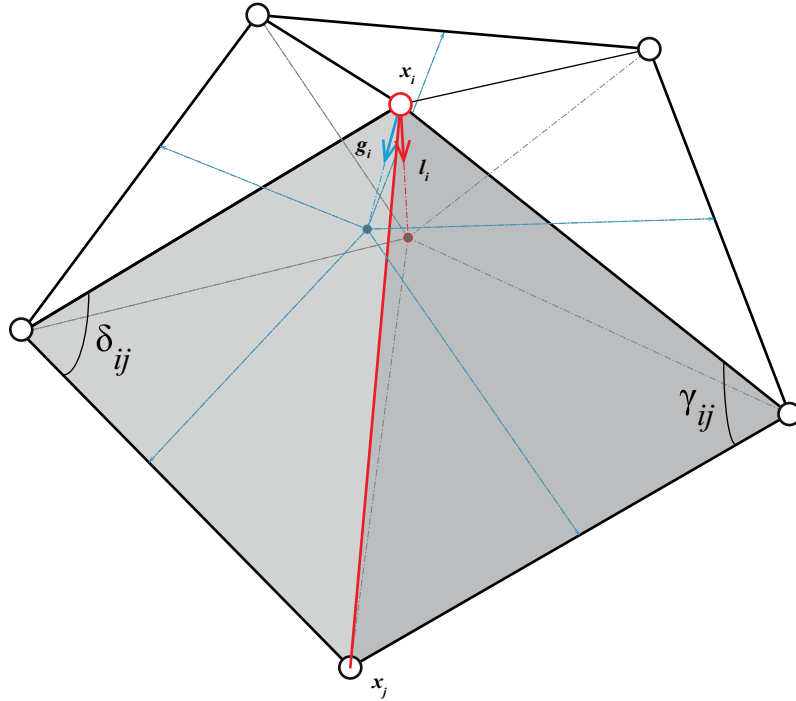


Figure 42 Gradient of Gaussian curvature -  $g_i$ , and Mean curvature (Laplacian vector) -  $l_i$

Thus the gradient of Gaussian curvature can be expected to have a formulation similar to the Laplacian vector, which is known to minimise mean curvature. This explains the analytical gradient that Desbrun et al. (2002) formulate and used in this research:

$$\nabla(\phi) = \sum_{j \in N(i)} \frac{\cot(\gamma_{ij}) + \cot(\delta_{ij})}{\|x_i - x_j\|} (x_i - x_j)$$

### 3.5 Boundary conditions and additional degrees of freedom

Typically, when DR is used to *form-find* minimal (mean curvature) surfaces, some or all of the boundary vertices of the mesh are held fixed. However in the case of *finding* curve-creased surfaces, the boundaries are required to find their equilibrium positions, since physically, any folding across a crease causes the boundaries to rearrange itself to compensate for the induced stretching of the material. As such, planarity forces are applied to the boundary vertices as well. However, since the boundary vertices do not have the full set of vertices to compute the Gaussian curvature correctly, developability forces are not applied to those vertices.

#### 3.5.1 Gradient of developability at a boundary vertex

Additionally, the boundary vertices could be considered as extra degrees of freedom that can be utilized towards finding the equilibrium solution, i.e. they can be moved to affect the Gaussian curvature of the adjacent interior vertex. Thus a gradient of Gaussian curvature of the adjacent interior vertex, measured at the boundary vertex, is computed (Figure 43). A force is then applied to the boundary vertex along this gradient direction. The difference that this additional force makes to the final solution can be seen in (Figure 44).

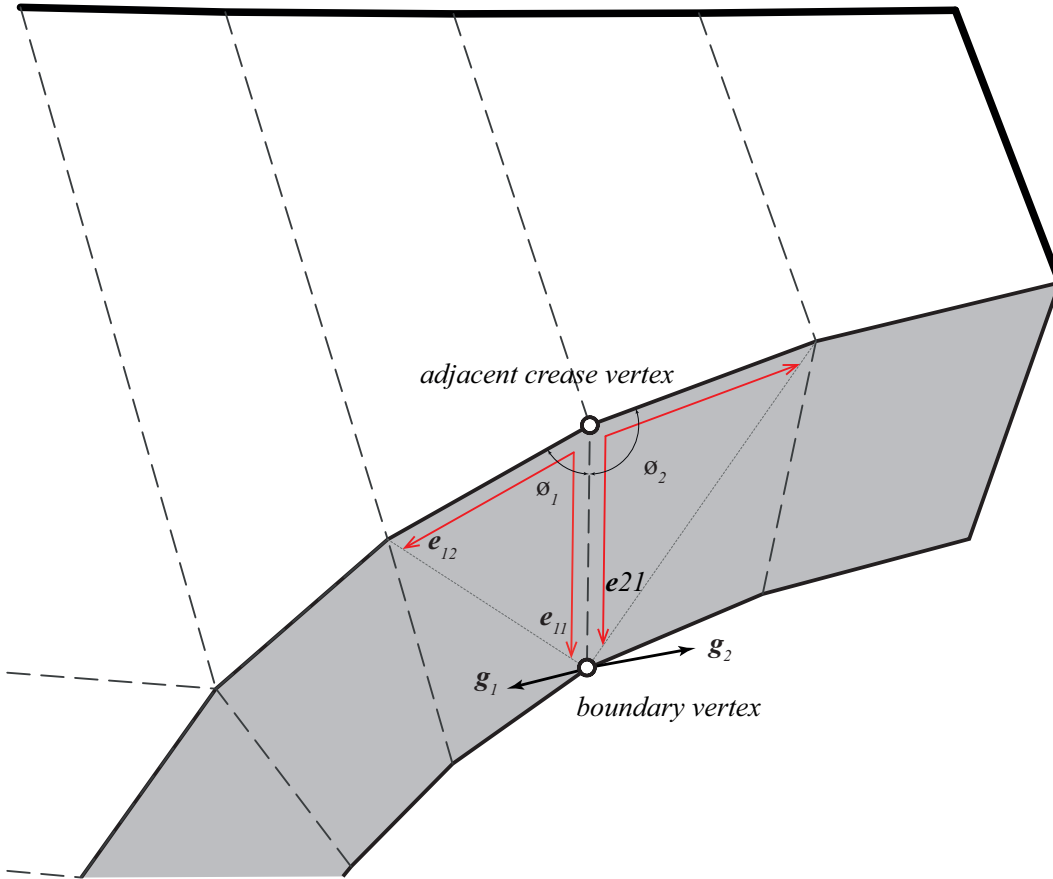


Figure 43 Gradient of Gaussian curvature of a (adjacent) interior vertex measured at a boundary vertex

$$F_{bg} = \nabla\phi_{bg} * (2\pi - \sum\phi_g)$$

Of the three or more angles contributing to the Gaussian curvature of the adjacent interior vertex, moving the boundary vertex can affect only two of the angles as shown in (Figure 43). Inspecting any of those two angles, it is easy to see that the angle at the interior vertex can be changed the most rapidly, if the boundary vertex is moved along a circle centred at the interior vertex, with radius equal to the length of the edge. In other words and as the formulation below shows, the gradient of each of the angles is ortho-normal to the edge connecting the boundary vertex to the interior vertex, and the normal of the triangle. Thus the gradient of sum of the two angles is the sum of the two such vectors. This is stated formally below:

$$\nabla\phi_{bg} = \nabla\phi_1 + \nabla\phi_2 = \mathbf{g}_1 + \mathbf{g}_2 = (\mathbf{e}_{11} \otimes \mathbf{e}_{12}) \otimes \mathbf{e}_{12} + (\mathbf{e}_{21} \otimes \mathbf{e}_{22}) \otimes \mathbf{e}_{21}$$

$bg$  is the boundary vertex and its connected interior vertex is  $g$ , and  $\mathbf{e}_{11}$  and  $\mathbf{e}_{12}$  and  $\mathbf{e}_{21}$  and  $\mathbf{e}_{22}$  are unitized vectors along the edges of each of the two triangles.

In practice, the vertex could also be moved along its normal, which has a similar global effect.



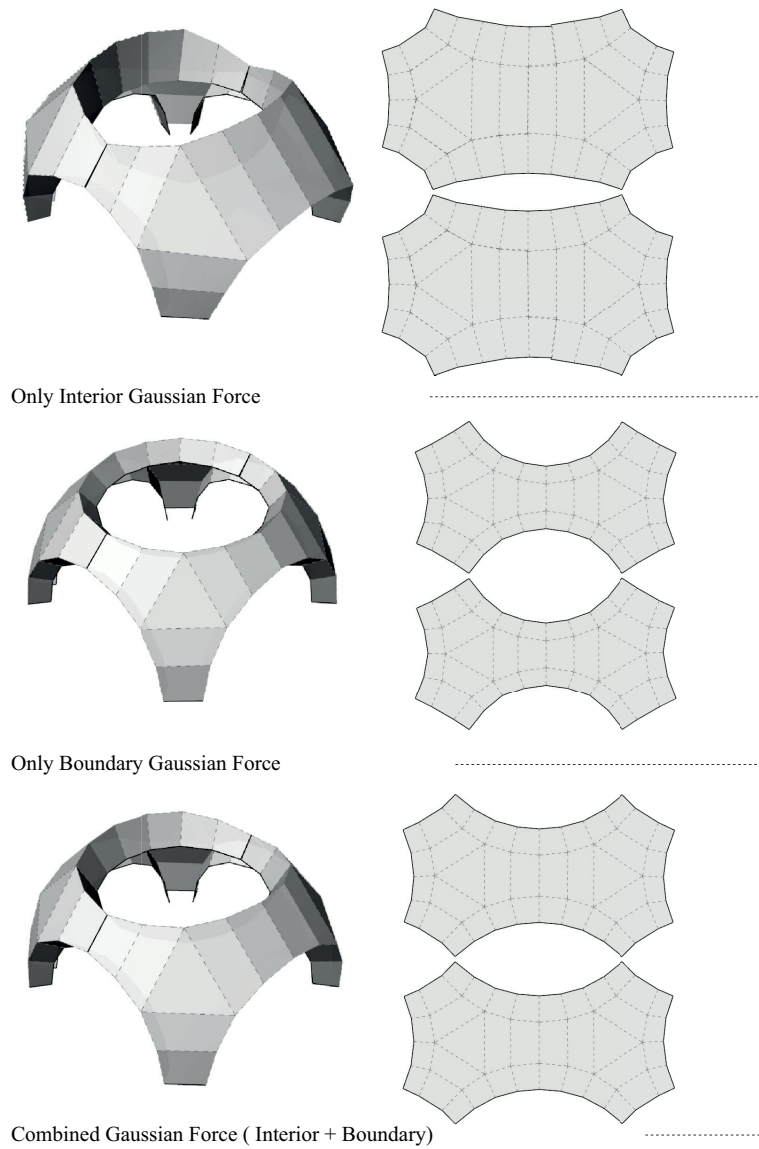


Figure 44 The effects of using various combination of forces and their respective unrolled results.

### 3.5.2 Extra row of faces

Alternatively, an extra row of boundary faces can be added to the mesh, thus converting the original boundary vertices to interior ones. The extra vertices can then be held fixed. These extra faces can be used to direct the solution of the solver, since the additional row controls half the number of angles subtended at a vertex, thereby limiting the scope of movement of the vertex in order to ensure that the total sum is  $2\pi$ . The effect of this user-defined addition can be seen in (Figure 45).

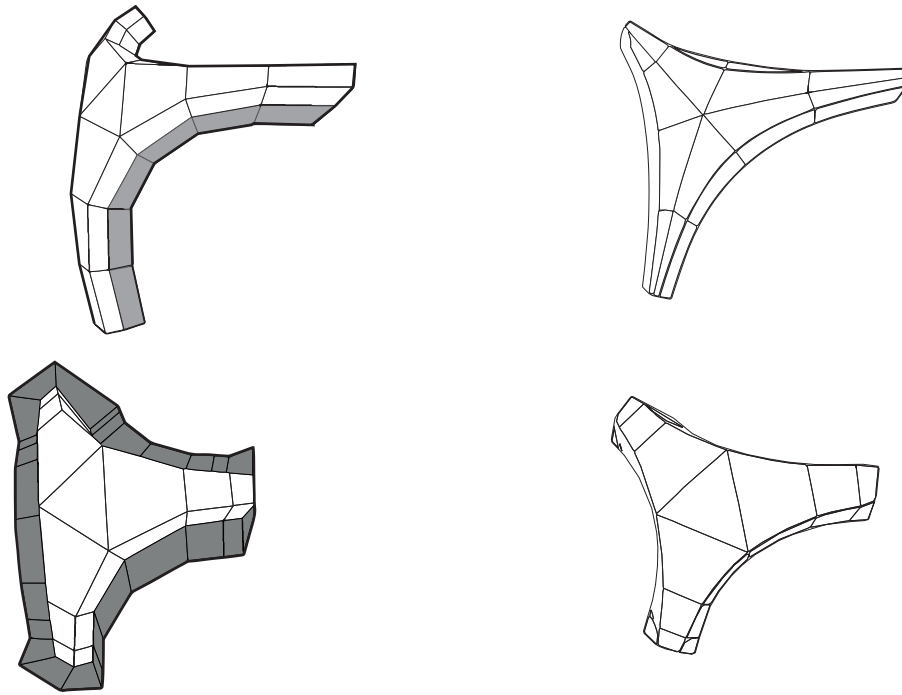


Figure 45 Effect on the resultant perturbed mesh, of adding extra-row of faces in plane with original boundary faces (Top) and normal to them (Bottom)

### 3.6 Planar Development

The algorithm proposed in the previous section (Section 3.3), employs an iterative and local method to solve the minimization problem stated in Eqn (2). It follows that successive iterations are stopped when the error is within numerical tolerance. This then implies that the resulting mesh is only developable within tolerance. In other words, the resultant, perturbed mesh is a quasi-developable. As with the previously indicated applications of quasi-developable surfaces (Sections 2.4 & 3.1), they are subject to the tolerances afforded by the method and material of physical reproduction of the digital surface. In this research, such tolerances were measured by the extent of isometry between the 3d surface and its planar counterpart or development.

#### *Spring-networks and Convex mappings*

Section 2.6 provides an overview of the various aspects of developing a planar mapping of a 3D mesh. An intuitive approach to the problem on the other hand, is to minimize the curvature of the mesh since flat objects do not have curvature. *Harmonic* mapping achieves this by applying the Laplacian operator unto the 3d mesh (Hélein & Wood 2008). This amounts to trying to minimize the energy of a spring-network or equalize the lengths of all the edges (Eck et al. 1995).

Apart from the *conformal* and *authalic* mapping methods (Desbrun et al. 2002) described previously (Section 2.6), this research implemented a modification to the harmonic mapping to achieve nearly *isometric* planar developments: rather than equalize the edge-lengths of the planar mesh, the difference in edge-lengths between the 3D and 2D meshes were equalized. This is in line with one of prominent algorithmic themes to ensure developability of meshes: minimize the *strain* between the 3D mesh and its corresponding planar-development (Kilian et al. 2008; McCartney et al. 1999; Wang et al. 2002). Similar to Wang et al. (2002), this was achieved this by accumulating *spring* forces on vertices, proportional to difference in lengths of the corresponding edges in the 3D and 2D meshes. The differences between the three are shown in (Figure 46). For more on parameterizations with spring-like energy minimizations, the reader is referred to Zhong & Xu (2006).

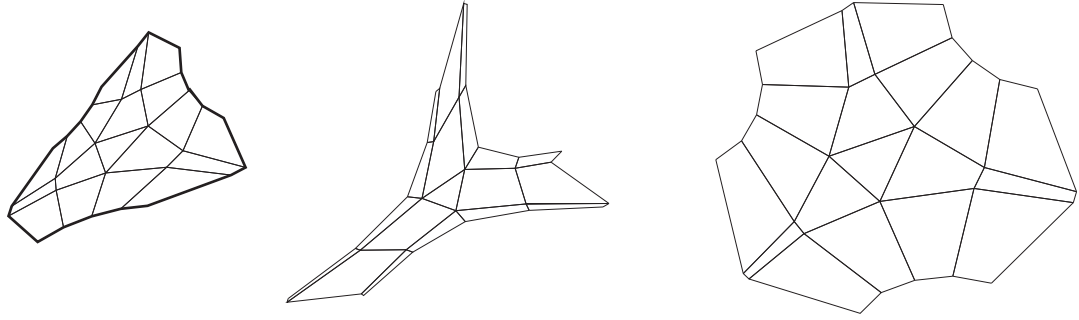


Figure 46 Left to Right : Differences between Authalic, Conformal and isometric parametrization of an example 3-crease mesh

It is useful to note an important theorem on parameterization, particularly in the context of use of spring-like methods. Implementation of such methods often runs into problems of overlap of edges, folding-over of triangles etc. In general, a *bijective* parameterization is guaranteed only if the spring constants that are used are non-negative and if the pinned boundaries are convex (Tutte 1963; Floater 2002).

#### *Implementation and commercially available methods*

Computing an isometric planar development of the resultant mesh is important in the process of physical reproduction of the computed CCF geometries. In other words, it is not integral to the proposed method. As such, the implementations described previously were attempts at a complete design tool and to verify the accuracy of proposed algorithm. In practice, more robust and commercially available implementations were used – the well-known CAD software of Rhinoceros™ and Pepakura™ that is often used by paper-craft artists. The tolerances that were used to establish the stopping criteria for the algorithm subsequently were a function of the error-thresholds afforded by these software (Figure 47) and the acceptable difference in surface area of the 3D and 2D meshes. This limitation is discussed in the *conclusions and future work* (Chapter 8)

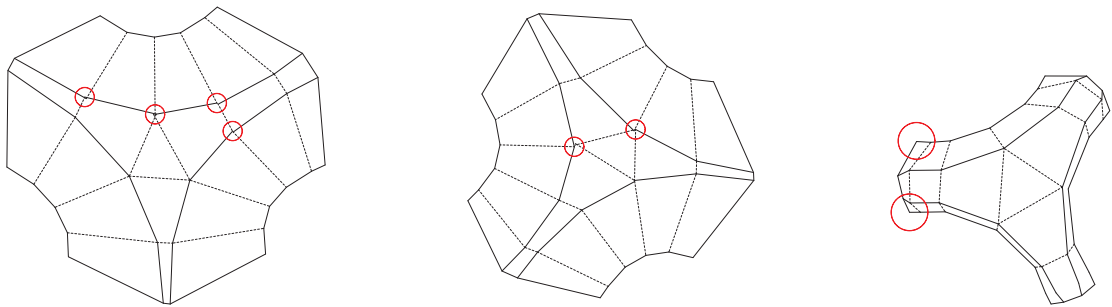


Figure 47 Examples of misalignment errors produced by Pepakura in the unfolding process used to establish stopping criterion for the perturbation process.

### 3.7 Results

#### 3.7.1 Stiffness of ODE

The *solving* for the equilibrium position of the vertices that are subjected to virtual forces of planarity and developability as described previously, is a so called ‘stiff’ problem i.e. it requires that the step sizes are very small. If the mesh is subjected to only one of the forces, the solver converges rapidly, and within acceptable numerical error (Figure 48 and Figure 49). If both forces are applied, the convergence behaviour changes (Figure 50), requiring smaller step sizes and thus more iterations. A possible method to overcome the stiffness is minimizing one of the functions beyond a threshold before applying the forces corresponding to the other. In practice, it was found that solving first for developability and subsequently solving for planarity speeds up the search (Figure 51).

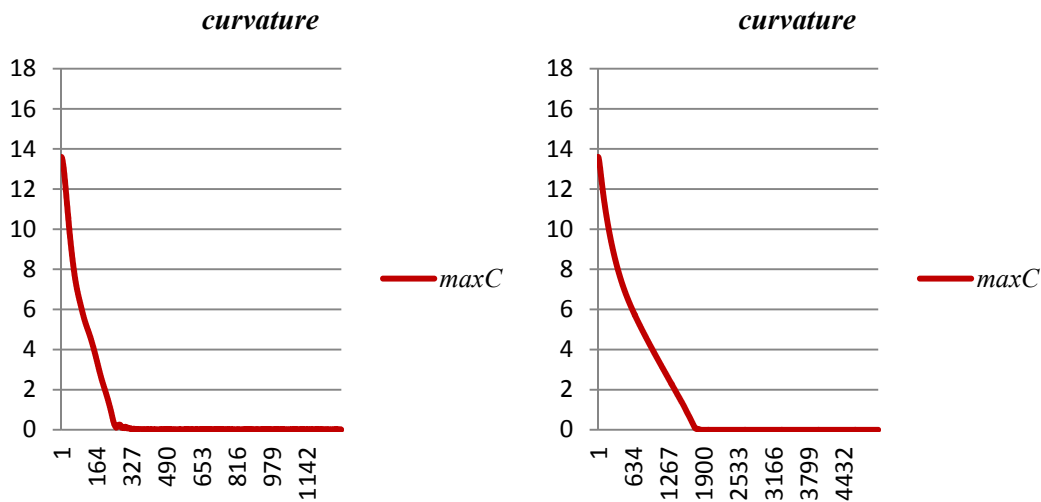


Figure 48 Convergence graph of developability metric when only Gaussian force is applied: Left – two-crease example mesh, Right four-crease mesh

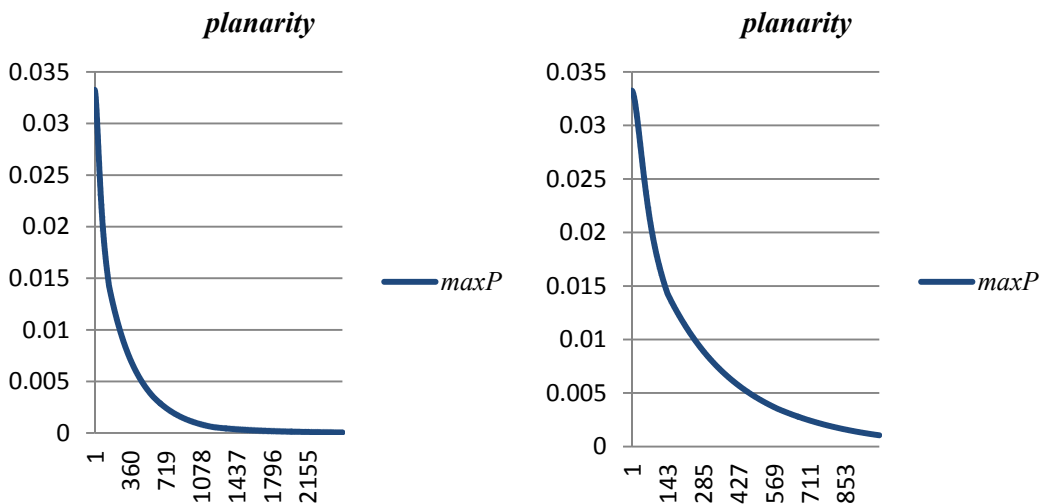


Figure 49 Convergence graph of planarity metric when only planarity force is applied: Left – two-crease example mesh, Right 4-crease mesh

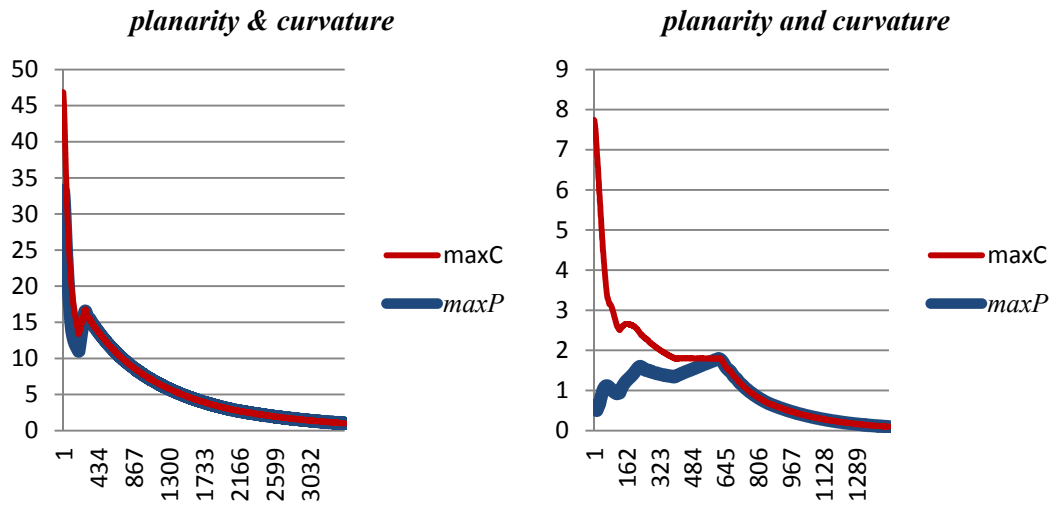


Figure 50 - Convergence graph of developability & planarity metrics when Planarity and Gaussian forces are applied simultaneously: Left – two-crease example mesh, Right 4-crease mesh. Planarity graph scaled by 100.

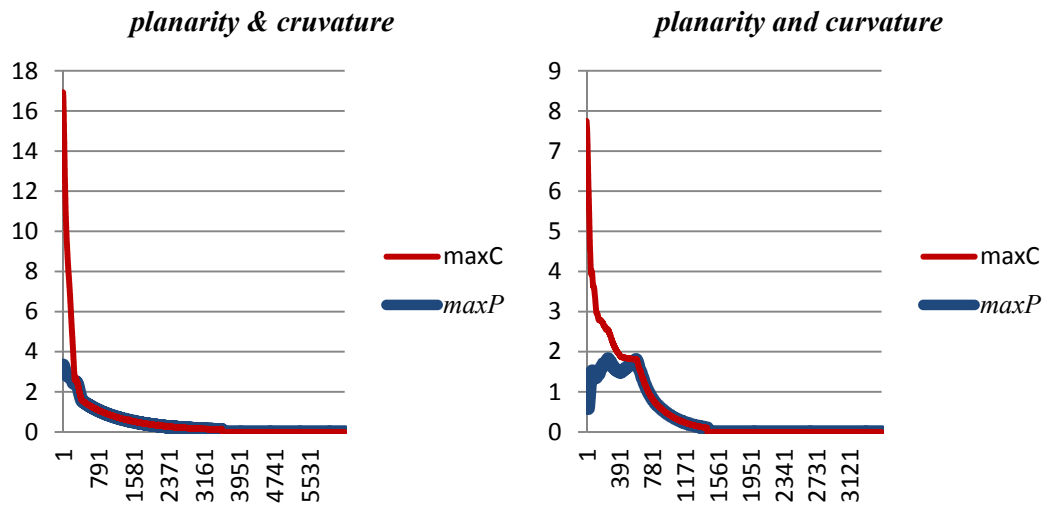


Figure 51 Convergence graph of developability & planarity metrics when Planarity and Gaussian forces are applied sequentially: Left – two-crease example mesh, Right 4-crease mesh. Planarity graph scaled by 100.

### 3.7.2 Step size

Additionally, the method used to integrate the ODE – 4<sup>th</sup> order Runge Kutta (RK4) - is an explicit scheme. Explicit integration schemes are known to have problems with numerical stability, especially when partial differential quantities or gradients are involved (Shampine & Thompson 2007). Practically speaking, this was overcome using relatively small step sizes (0.05 – 0.1 units). Implicit schemes are known to be unconditionally stable i.e. irrespective of step size. This short-coming in the software implementation is noted in *Conclusions and Future work* (Section 8.3.4).

### 3.7.3 Degrees of freedom

Section 3.5 mentions the potential use of boundary vertices as extra degrees of freedom in the perturbation process. This implied the application of an additional force to the boundary vertices, along the gradient of Gaussian curvature. Figure 44 already showed the effective use of this force. This force however, is not always compatible with the other forces at play. Figure 52 and Figure 53 show the effect of the independent application of this force on two test-cases. As

can be discerned from the figures, this *force* can perturb the (boundary) vertices quite far from their original positions and therefore better to omit it. It was found that the method does indeed converge on a solution with such a force included, albeit a solution distant from the input mesh.

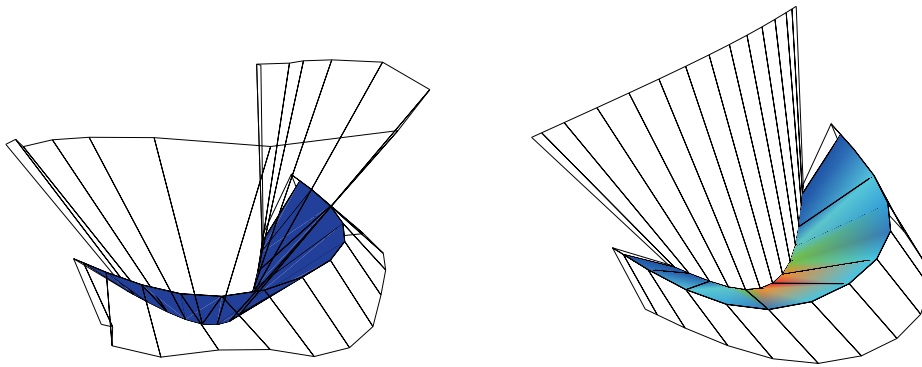


Figure 52 Left : Resultant positions of boundary vertices that make the interior vertices developable. Right : Original positions of boundary vertices. (Example two-crease mesh)

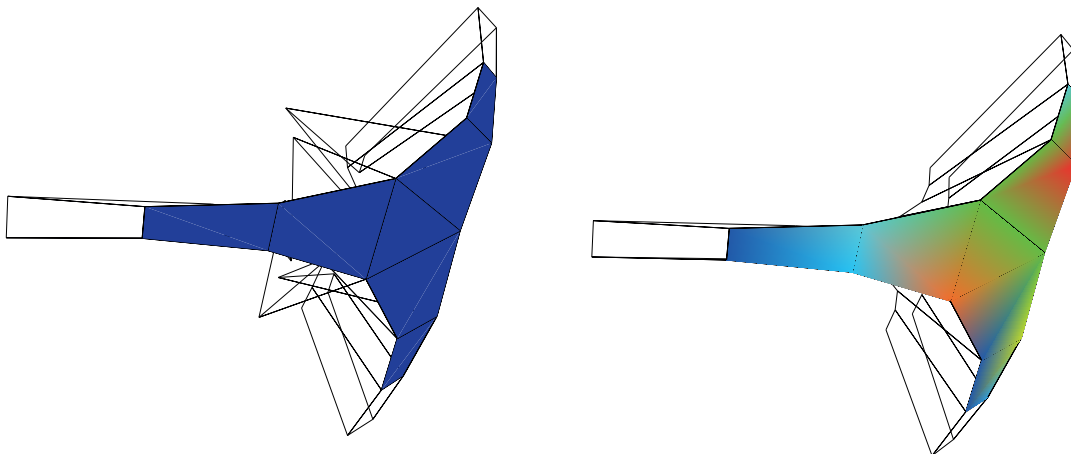


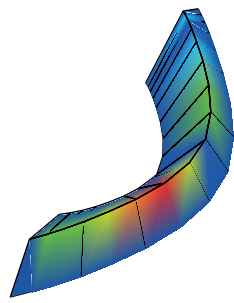
Figure 53 Left : Resultant positions of boundary vertices that make the interior vertices developable. Right : Original positions of boundary vertices. (Example three-crease mesh)

### 3.7.4 Table of results

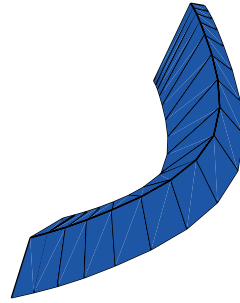
Perturbed CCF results from input meshes with increasing number of creases are shown below: Figure 54 and Figure 55 show simple geometries with one and two crease-folds respectively. Figure 56 shows a primitive type with three-crease folds and other compound assemblies of the basic 3-crease type. The last two rows of Figure 56 shows that the proposed method does in fact fully *solve* geometries that were used to construct the physical prototypes described in Chapter (4) & Chapter (5). These chapters give more details regarding the extent to which, the method that is proposed here, was used in the case studies.



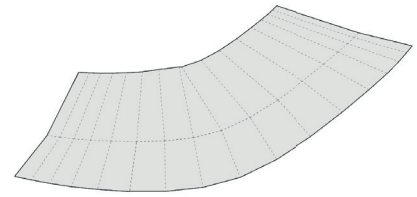
Re-Mapped Gaussian Curvature



**Input mesh**  
surface area: 106.550019 sq.units



**Perturbed mesh**  
surface area: 106.550232 sq.units

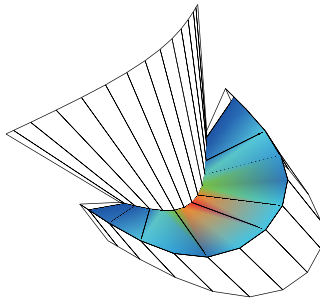


**Unrolled mesh**  
Unrolled surface area difference:  
0.000213 sq.units

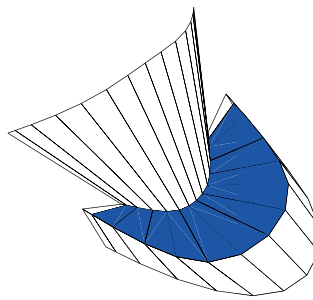
Figure 54 Left to Right: original single-crease mesh, perturbed result and unrolled layout



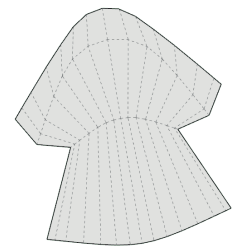
Re-Mapped Gaussian Curvature



**Input mesh**  
surface area: 56.6851003 sq.units



**Perturbed mesh**  
surface area: 56.6853824 sq.units



**Unrolled mesh**  
Unrolled surface area difference:  
0.0002821 sq.units

Figure 55 Left to Right: original two-crease mesh, perturbed result and unrolled layout

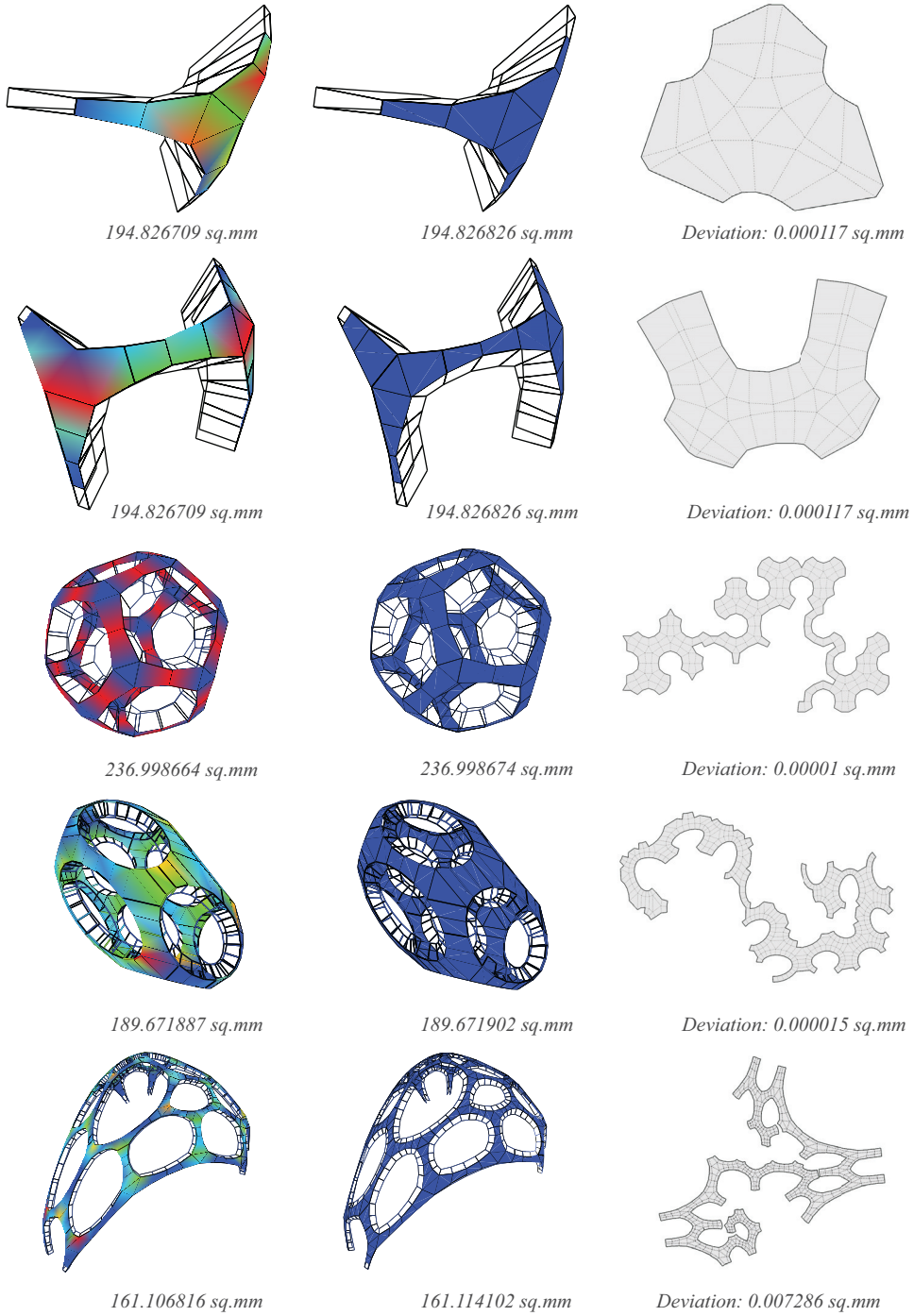
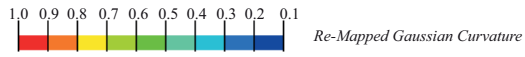


Figure 56 Left to Right: original multi-crease mesh, perturbed result and unrolled layout. Top to bottom: 3-crease mesh, 4-crease mesh, mesh with closed creases, mesh used in prototype 2 (Chapter 4) and prototype 1 (Chapter 5).



## 4 CASE STUDY 1

The prototype described in this chapter, was an attempt to explore the applicability of curve-crease-folded geometries for producing moulds for a specific class of skeletal structures. The prototyping exercise was also attempt to employ the computational method described in Chapter (3), in a realistic design and construction scenario (Figure 57). It should be noted that the research described in this dissertation was under development at the time this exercise, and the author was part of the collaborative design and build team that executed the prototype. Both the computational methods and the production of manufacturing information is described in a co-authored paper (S Bhooshan, Bhooshan, et al. 2015). The contributions of this prototyping exercise to the research of the dissertation in general, and the special case of the proposed method of computing CCF geometries described in Section (3.5.1), is noted at the end of this chapter and acknowledged in the statement of co-authorship. Lastly, the construction prototype was funded by an educational workshop organized by the author, under the Visiting School Program of the Architectural Association, London.



Figure 57 Construction of a funicular skeleton structure using CCF moulds. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

### 4.1 Context

Physical *form-finding* using hanging chains and associated architectural design methods is common among architects. There are several digital methods to find the equilibrium shape of shell structures, including TNA-Thrust Network Analysis (Block & Ochsendorf 2007), Mass-spring methods ( Bhooshan et al. 2014), Dynamic relaxation (Barnes 1999) ,Force Density Method (Schek 1974) etc. For a comparative analysis of the methods the reader is referred to Veenendaal & Block (2012). In this case-study, Force Density Method was used due to its ease of implementation and amenability to parameter-free, interactive *modelling* paradigm of solving directly for the rest, equilibrium state ( Section 2.5).

The construction of *form-found* geometries with concrete requires the use of form-work, into which concrete is poured. Traditional form-work is often straight and difficult to form curved surfaces, an essential feature of shell structures (Figure 58).

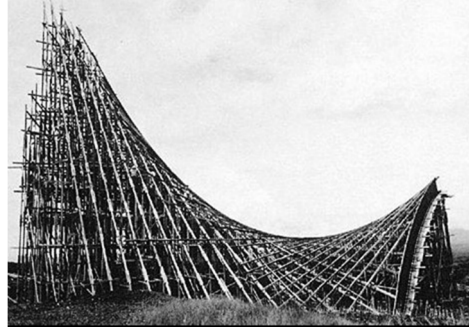


Figure 58: Typical form work for shells. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

These difficulties extend to the construction of curved skeletal geometries. In this context the proposed use of curve-folded moulds is relevant, in that curved-moulds can be economically and efficiently formed from sheet material. It is important to note that the curve-folding technique can be used only in cases where the underlying mesh has consistent positive mean-curvature. Hence, there is compatibility between compressive skeletal structures and curve-folded moulds (Figure 59). This is elaborated at the end of this chapter. Lastly, a precedent project to note is Pedersen et al. (2015) for their use of straight folded plastic as form-work for a skeletal shell computed using TNA.

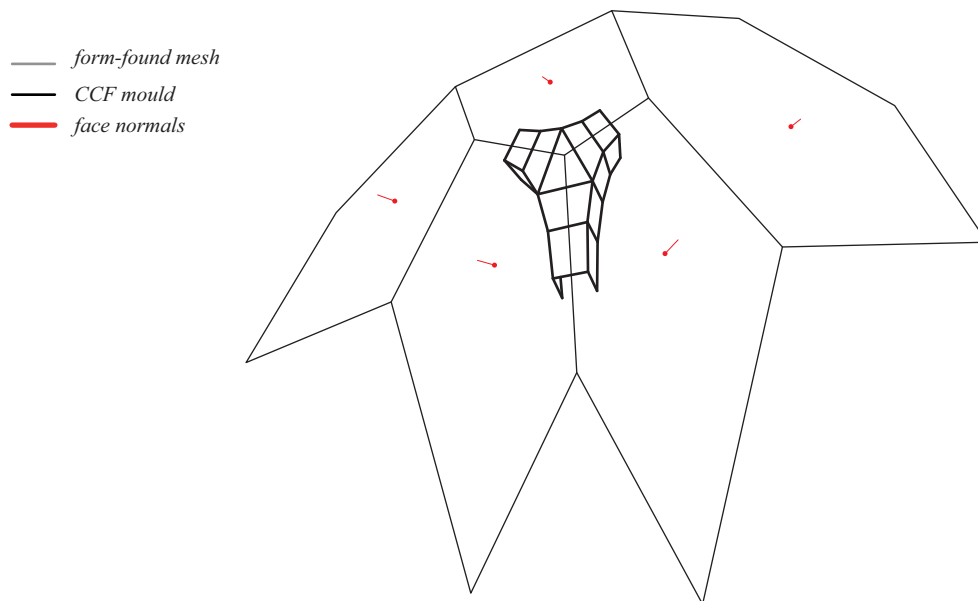


Figure 59: Showing a mesh in close-up with mean-curvature direction (red lines), mesh-edges (grey) and a y-shaped node (bold). Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

## 4.2 Design process

The design pipeline consists of four main steps, each described in subsequent sections.

1. Generate a discrete compressive mesh.
2. Derive geometries for curve-foldable moulds from 1.
3. Apply the previously described method (Chapter 3) to perturb such an input mesh to form CCF geometries.
4. Generate manufacturing information.

### 4.2.1 Discrete compressive mesh

A user-specified, typically planar, mesh, is transformed into a topologically identical compression-only surface-mesh, using the Force-density method (Schek 1974)(Figure 60).

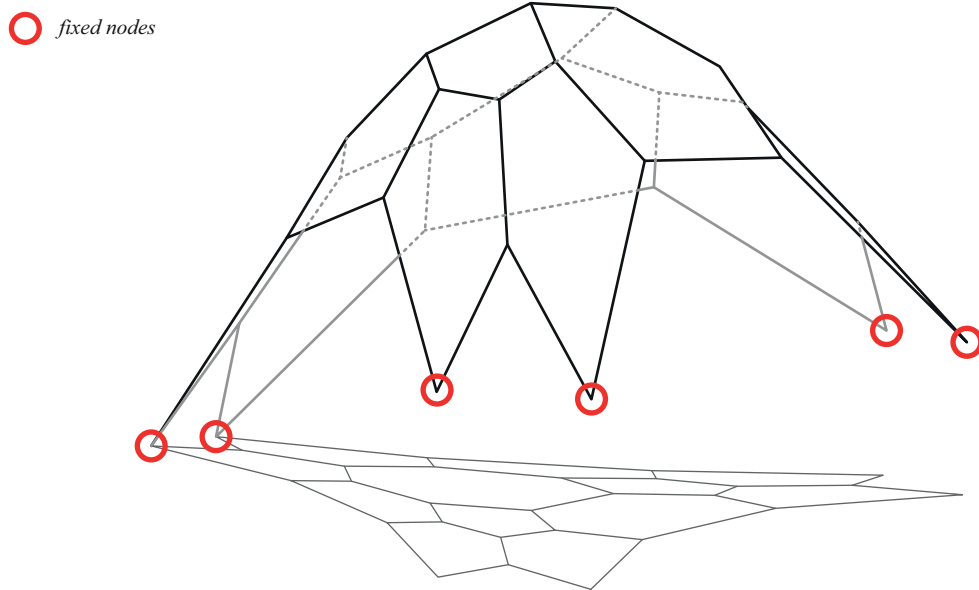


Figure 60: Form finding of funicular mesh. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

Of relevance to the current research is that for the built prototype, all the internal vertices of the user-specified, input mesh were tri-valent. This was to constrain the construction of the prototype to the use of only Y-shaped moulds (Figure 61), and thus reducing complexity in on-site production.

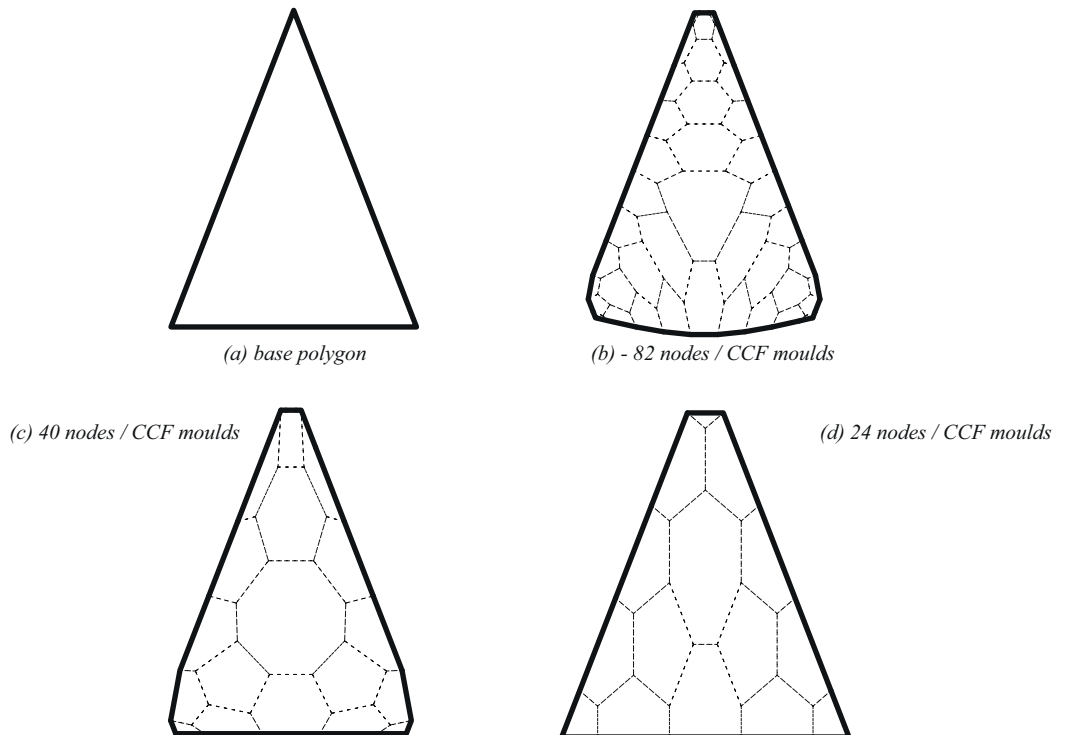


Figure 61: Design Iterations, the total number of Y-panels was a driving factor in the demonstrative prototype. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

#### 4.2.2 Deriving geometries for curve-foldable moulds

This process involves the conversion of the *form-found* compressive geometry to a predominantly quad topology suitable for perturbation towards becoming curve-crease-foldable i.e. suitable for applying the method described in Chapter (3). The process is mostly procedural in that it involves the sequential use of well-known mesh manipulation operations known as Conway operators (Conway et al. 2008). User input is restricted to choosing the parameters of each operation. However, it can additionally involve arbitrarily moving some vertices to correct *degeneracy*, acute angles etc. The specific sequence of operators that were used in the exercise is shown in (Figure 62). Detailed description and other applications are shown in (Chapter 6).

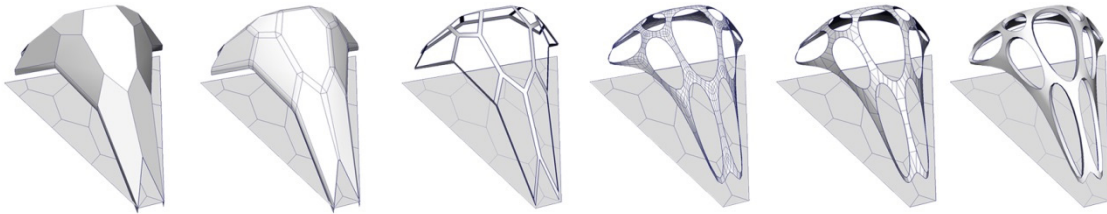


Figure 62: Left to right: Bevel operator on mesh-edges, extrude and scale operator on mesh-faces, Catmull-Clark smoothing, deletion of some of the edges to produce a predominantly quad CCF topology. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

#### 4.2.3 Perturbation to CCF geometry

Subsequent to the generation of a suitable topology for the input mesh, the methods described in Chapter (3) can be directly applied as shown in (Section 3.6). The application of the method produces a planar-quad mesh that is additionally developable i.e. CCF geometry. In this exercise however, certain changes were made in consideration of time and production constraints.

##### *Planarity Force*

The choice of mesh operations to produce the input geometries, guarantees that the input geometry consists of only triangles and quads. Thus, the simpler, diagonal-distance formulation of the gradient of planarity (Section 3.4.1) was employed as shown in ( Figure 63).

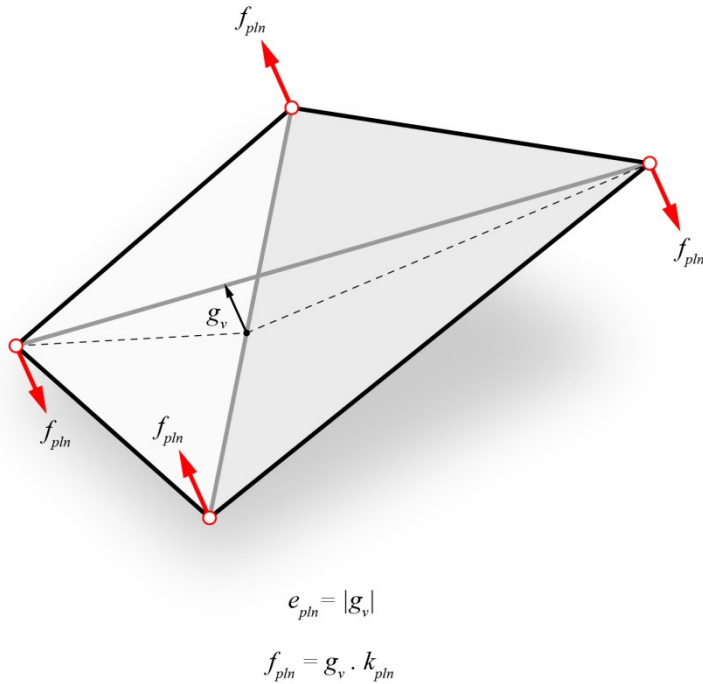
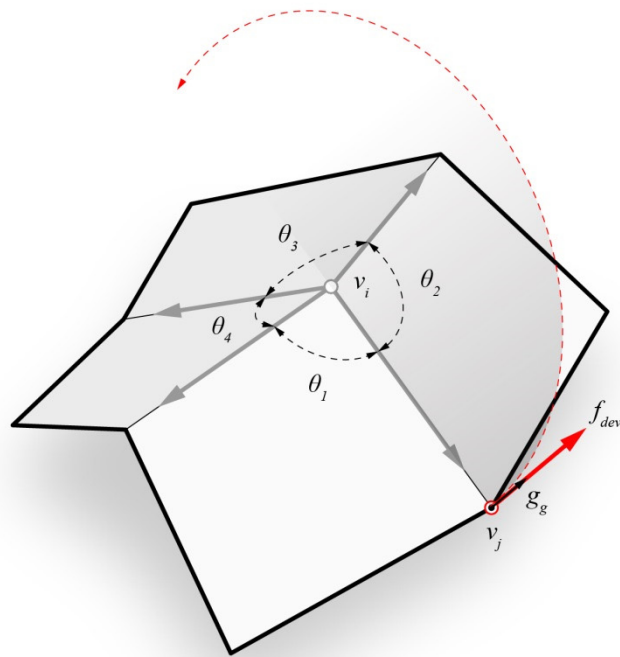


Figure 63: Formulation of the planarity force. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

### Developability Force – Boundary vertices

The application of developability force as described previously (Section 3.4.1), presented a peculiar constraint in that, moving the interior crease vertex would often cause the compressive graph to ‘stick-through’ the mould geometry. In other words, the crease-vertices of the mould-geometry were required to remain relatively close to the input geometry. This constraint meant that additional degrees of freedom afforded by boundary vertices (Section 3.5.1) were indeed useful in this case. Additionally, the ‘regular’ developability force was not applied to the *crease* vertices. It can be noted that, in this exercise the boundary vertices were only moved along their normal by an amount proportional to the angular defect at the adjacent interior vertex (Figure 64) as opposed to the moving them along the gradient as described in (Section 3.5.1). This was because vertex normals are more readily available in mesh data structures, whilst the gradient would imply additional computation. Iterative schemes are usually tolerant of such deviations since the local error is distributed across the network, and the solution still converges to within specified tolerances.



$$e_{dev} = 2\pi - (\theta_1 + \theta_2 + \theta_3 + \theta_4)$$

$$f_{dev} = \hat{g}_g \cdot e_{dev} \cdot k_{dev}$$

Figure 64: Formulation of the developability force. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

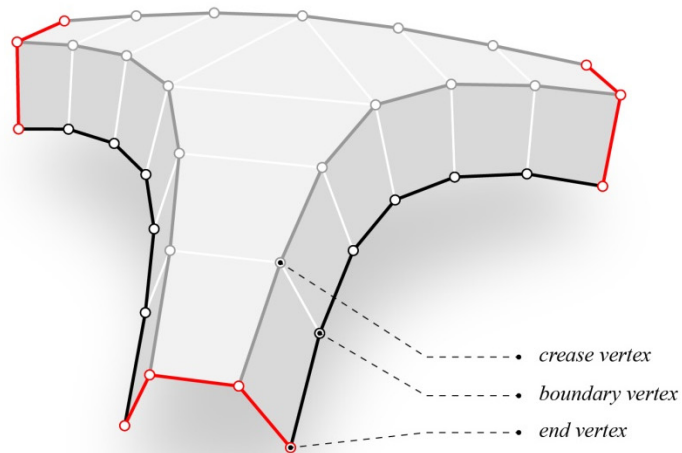


Figure 65: Vertex classification. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

### Developability Force – End-vertices

The input mesh also contained special cases of boundary vertices (Figure 65), referred to as *end* vertices. An end vertex is a boundary vertex which is located on an edge loop spanning across

multiple fold creases. These were omitted from developability forces since unlike standard boundary vertices their adjacent edge loop does not define a fold crease.

The results of the method on a Y-shaped node are shown below (Figure 66) and a few states at intermediate iterations are noted (Figure 67).

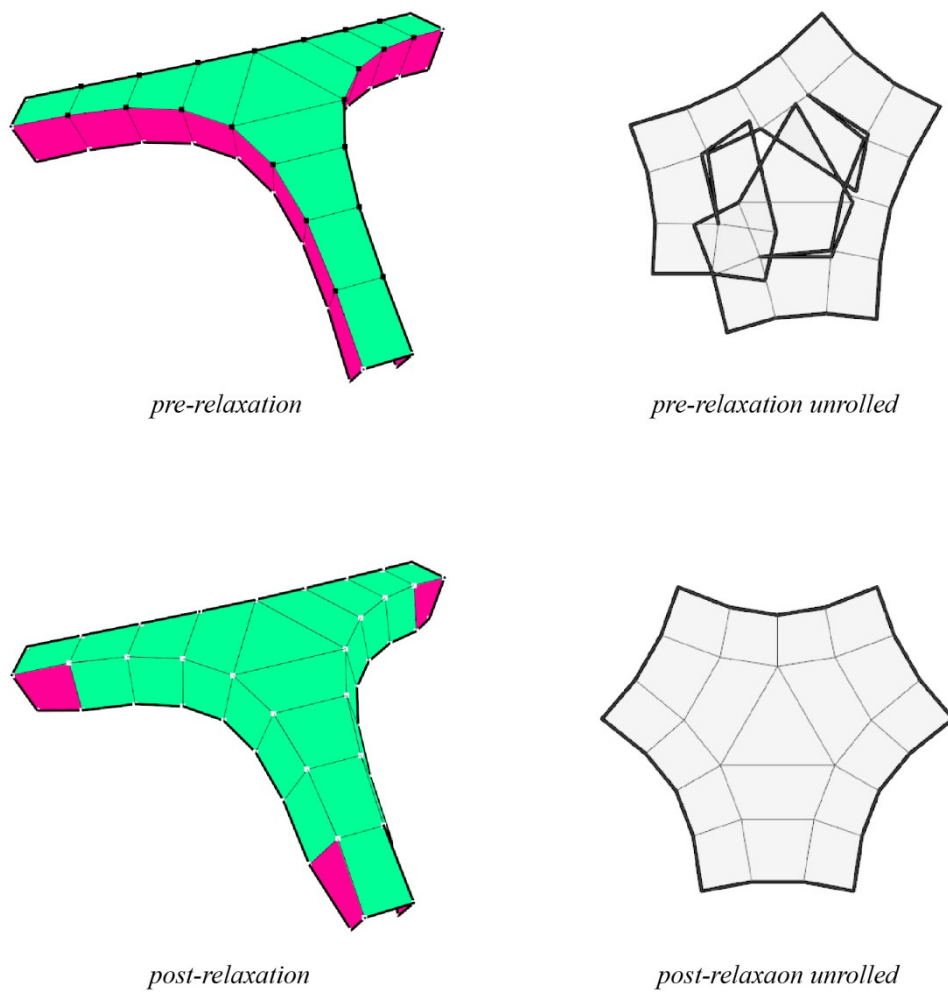


Figure 66: A single Y component before perturbation (a) and after perturbation (b). Green faces are planar and white vertices are developable. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

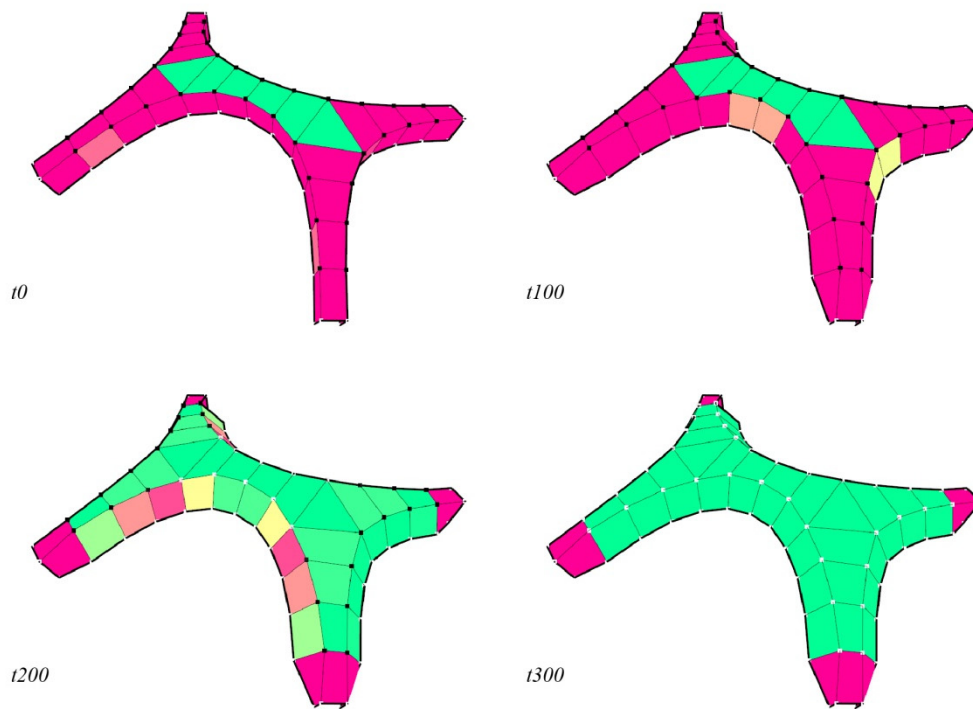


Figure 67: Relaxation sequence of a pair of Y components. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

#### 4.2.4 Input mesh discretization

The discretization of the input mesh presented some problems with regard to the numerical stability of the perturbation process. The mesh had many-creases ( $\sim 85$ ), and certain areas needed to be manually adjusted to achieve convergent solutions.



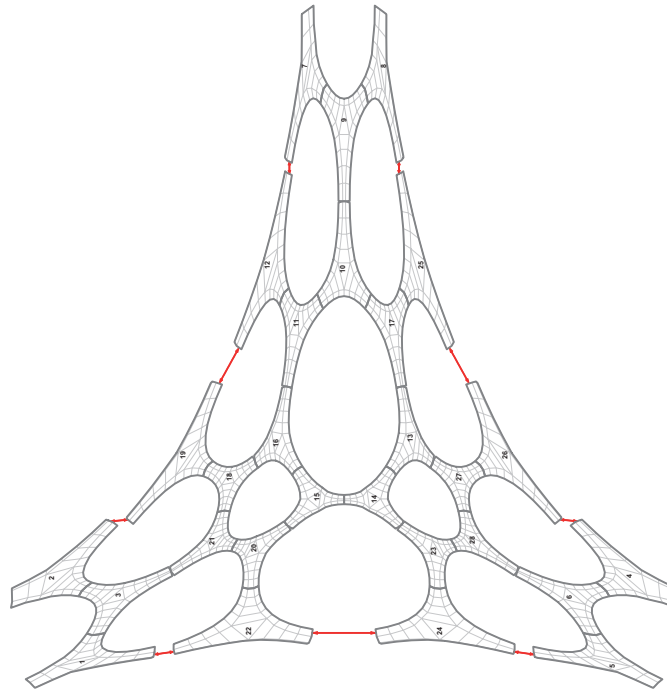


Figure 68 Separation of mesh into CCF moulds, subsequent to perturbation. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

### 4.3 Manufacturing information

#### 4.3.1 Planar development / Unfold layout

These discrete CCF meshes were first isometrically flattened into 2D meshes (Figure 69) using in-built tools in Rhinoceros. Subsequently smooth boundary curves were derived for laser-cutting. This ensured minimum deviation during the unfolding process, as opposed to constructing a refined mesh in 3D and flattening it.

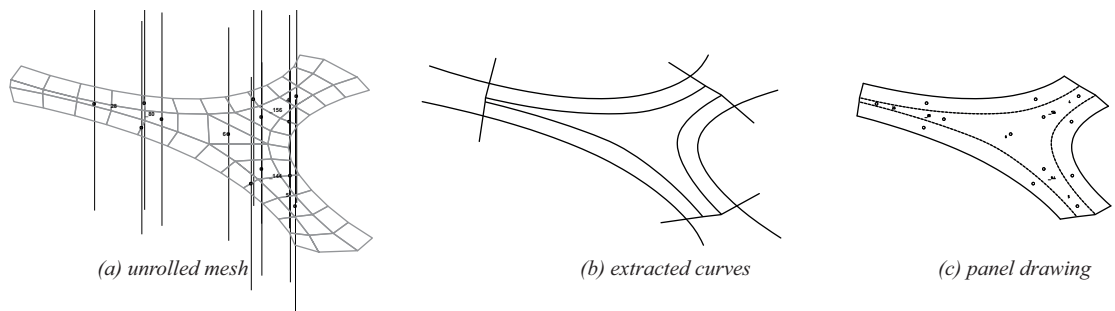


Figure 69: (A) Input mesh unrolled, (B) Extracted curves, (C) Typical panel fabrication drawing. Image courtesy (S Bhooshan, Bhooshan, et al. 2015)

#### 4.3.2 Constructing smooth boundary curves

A procedure to interpolate between discrete flattened rulings to their *exact* smooth (limit) rulings is described in Kilian et al. (2008). This procedure is constructed from the fact that the *rulings* of a smooth developable surface are the limits of *ruling* edges of the discrete mesh. Thus a smooth boundary curve can be constructed by interpolating additional rulings as shown in

Figure 70, and subsequently drawing a curve that starts on any given ruling and meets intersects subsequent rulings.

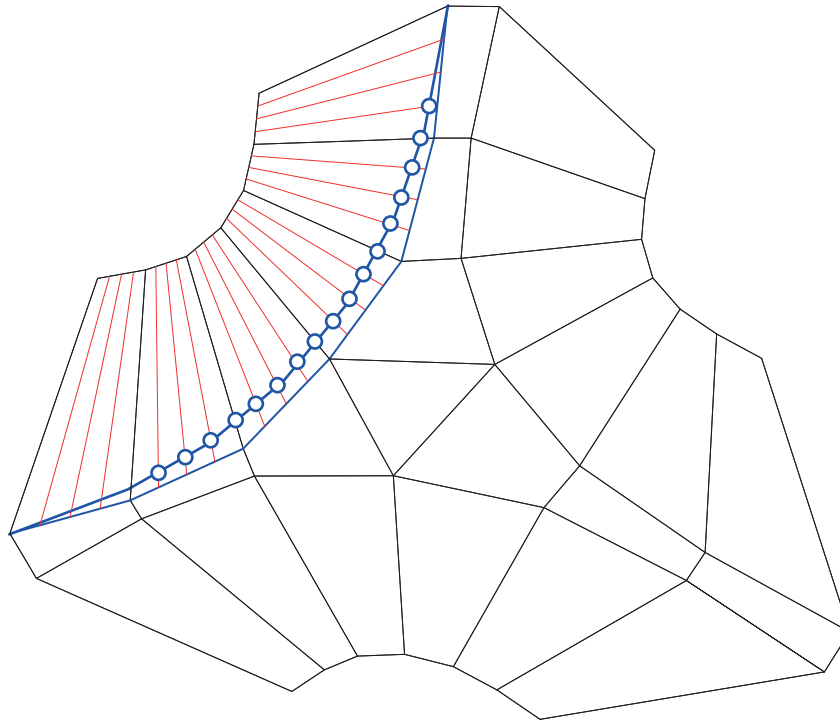


Figure 70 Procedure for generating smooth boundary curves in the planar domain

In the production of the cutting-layouts for the moulds however, readily available interpolation schemes in Rhinoceros were used to construct smooth curves passing through the discrete *flattened* vertices. This caused some of the curves to become over-constrained and have rapidly changing curvatures (Figure 71). Additionally, the interpolated curves were not an *exact* fit to the CCF geometries in the sense that if the smooth curves were to be folded back to 3D, they wouldn't align with the computed 3D geometry. Such instances were few, and were manually corrected in the planar domain.

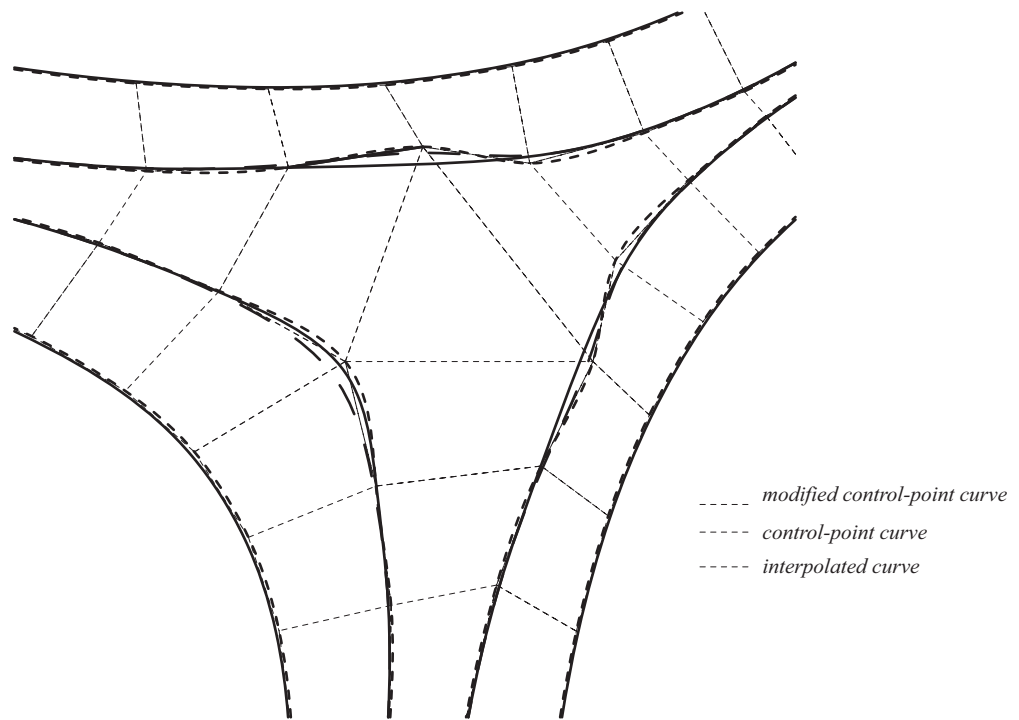


Figure 71: Shows the smooth interpolation of the crease lines using various NURBS schemes

#### 4.4 Summary

This prototyping exercise in a real-world design scenario greatly helped to:

- i. Formalize the computational method described previously,
- ii. Reveal the synergy between the discrete funicular structures and moulds for casting concrete that are curve-folded from sheet material.

The CCF technique is most amenable in cases where the compressive mesh has consistently signed (positive or negative) mean curvature (S Bhooshan, Bhooshan, et al. 2015). This is usually the case in the case of funicular meshes. Additionally, the cross-sectional depth of the mould, and thus the cast element, is dependent on the curvature of the curved creases. These, in turn, are dependent on the volume of the tetrahedron of the 1-ring neighbourhood of the vertex, or the distance of the vertex to the plane formed by its 1-ring vertices (Figure 72).

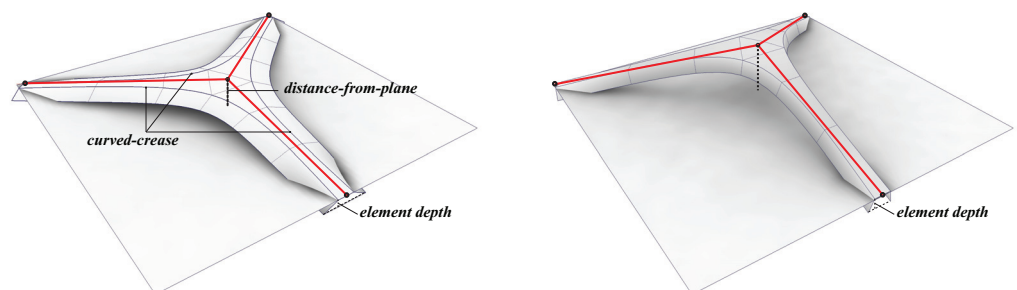


Figure 72 Relation between CCF moulds and element depths. Image courtesy (S Bhooshan, Van Mele, et al. 2015)

This correlation was subsequently stated as a constrained optimization problem in the use of the Force-density method that was employed to find the funicular shape in this exercise. This is described in Bhooshan, Van Mele, et al. (2015)

- iii. Reveal the potential use of the boundary vertices as additional degrees of freedom as described previously. This additionally prompted the exploration of the analytical solution for the gradient of developability at such vertices (Section 3.5.1).
- iv. Identification of potential fabrication-related problems with regard to generating smooth, cutting curves from discrete parameterization of CCF geometries. This prompted the discovery of the procedure to construct smooth limit curves from PQ discretization that is described in Kilian et al. (2008).
- v. Highlight the need for penalty functions in the proposed perturbation method. Additional penalty functions to reduce deviation from the input mesh were not necessary in the case of this prototype due to use of additional freedom afforded by the boundary vertices. This was also aided by the relatively good approximation of curve-foldable geometry provided by the input mesh. For generalized use however, penalty functions could be added an additional force to minimize deviation from input mesh, as several other authors do (Kilian et al. 2008; Poranne et al. 2013; Wang & Tang 2004).
- vi. Develop the sequence of Conway operators needed to (procedurally) transform a compression mesh to one that has a topology suitable for CCF geometries.

## 5 CASE STUDY 2

The research described in this dissertation was under development at the time this exercise, and the author was part of the collaborative design-and-build team that executed the prototype. Both the computational methods and the production of manufacturing information is described in a co-authored paper (Chandra et al. 2014). The contributions of this prototyping exercise to the research of the dissertation is noted at the end of this chapter and acknowledged in the statement of co-authorship. Lastly, the construction prototype was funded by an educational workshop organized by the current author, under the Visiting School Program of the Architectural Association, London.

This exercise was an attempt to investigate simplification of the proposed perturbation method (Chapter 3) when dealing with a specific class of geometries. In effect, the results of this exercise confirmed that some of the constraints of the proposed method could be relaxed in specific cases. A key point of departure was to geometrically construct feasible geometry as opposed to an iterative and local search method. As such, the method used in this prototype benefits from speedy computation and thus an edit-and-observe exploratory strategy to design. Critically however, it is applicable to a specific class of geometries, i.e. convex polyhedral meshes and it produces quasi-developable solutions. As shown previously (Section 3.6), the geometries could indeed be solved using the more generally applicable iterative perturbation method proposed (IPM) in this dissertation. The method discovered during the course of this design-make exercise could serve as good input geometry, upon which the IPM could be further applied.



Figure 73 Skeleton representation of a closed polyhedron, developed as CCF geometry and assembled on-site. Image courtesy (Chandra et al. 2014)

### 5.1 Method

#### 5.1.1 Conway operators and the method of reflection

It is well acknowledged that planar curved folds are simpler to solve as noted in the introductory survey (Section 2.3) and Mitani & Igarashi (2011). Further, several iterations of paper models demonstrated that the creases remained nearly planar unless they were forced out of plane by external forces. Thus, whilst the user-specified polygon is modelled using well-known mesh editing tools (in-built in Autodesk Maya in this case), it is subsequently converted to a

polyhedron by applying planarization routine described in Chapter (3). Subsequently the constructive method used in this exercise, constructs, from the polyhedron, developable surfaces that are its curved-foldable, skeletal representation. We can recall that CCF geometries are two developable strips meeting along a crease. Mitani & Igarashi (2011) have shown that given any one the developable surfaces, the other can be computed as its invert-reflection (Section 2.3). In extension, the constructive method used in this exercise also constructs the first surface - the *independent surface* (white surfaces in Figure 74). The *dependent surface* (dark grey surfaces in Figure 74) is constructed as derivate of the independent surface.

The construction of the *independent surface* follows a procedure already described in (Section 4.2) and involves the sequential use of mesh operators known as the Conway operators (Conway et al. 2008). Users input, is needed in choosing the parameters of each operation. The specific sequence of operators that were used in the exercise is shown in (Figure 75). Other applications are described in (0). The *dependent surface* forms the inward extrusions from the *independent surface*, and is computed through an adaptation of the reflection method (Mitani & Igarashi 2011). Lastly, the resultant geometry is used to produce manufacturing information such as unfolded cutting layouts, and their assemblies.

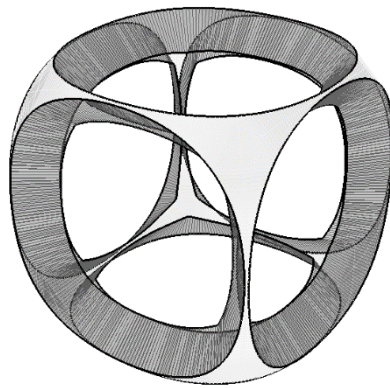


Figure 74 Independent surface in white, dependent surface in dark grey. Image courtesy (Chandra et al. 2014)

### 5.1.2 Independent surface

This step of the constructive method generates the *independent surface* by the applying the following series of operations on a convex polyhedron (Figure 75):

1. Chamfer and Bevel Conway operators (Hart, G, 1998) applied to all vertices and then edges respectively
2. Cubic Bezier curves extracted from faces of polyhedron
3. Planar polygonal surfaces generated at each vertex
4. Translational surfaces generated at each edge

In the above sequence, operations 1-2 permit a significant range of design variations and define the curves that form the edges of the designed surface. Operations 3 and 4 are surfacing operations, replacing each vertex of the original polyhedron with a planar polygonal surface and each edge with a translational surface. These translation surfaces have no Gaussian curvature by virtue of the planar faces of the initial polyhedron. In others words, the *independent surface* is developable by construction.

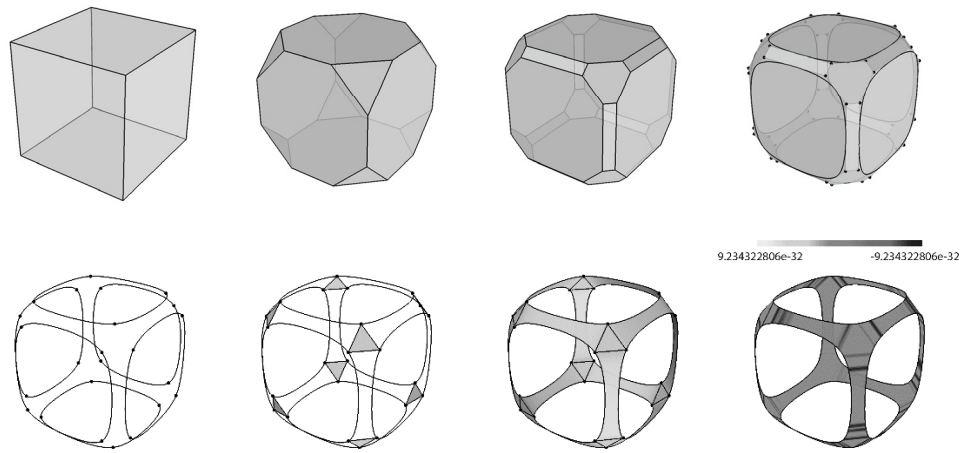


Figure 75 Geometric operations to construct the independent surface from a user-defined polyhedron. Image courtesy (Chandra et al. 2014)

### 5.1.3 Dependent surface

Early attempts to compute the *dependent* surface involved using the method of reflection (Mitani & Igarashi 2011). However the reflected surfaces tended to intersect each other (Figure 76). This prompted the search for a closed form solution specific to these types of closed convex polyhedral geometries.

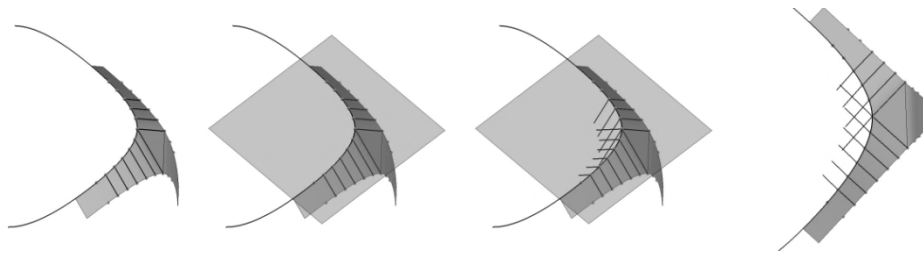


Figure 76. The method of reflection produces intersecting surfaces. Image courtesy (Chandra et al. 2014)

The solution computes an extrusion vector at each crease-vertex of the independent surface such that the sum of angles projected by the existing faces and the new faces sum up to  $2\pi$ . In other words, the extrusion vector is computed such that the crease-vertex becomes developable.

In Figure 77, consider P, A and B to be border vertices and the line PC to be an interior edge on the designed surface. A solution set  $\{Q\}$  exists such that every vertex  $Q_i$ :

$$\angle APC + \angle BPC + \angle APQ + \angle BPQ = 2\pi$$

Or:

$$\theta_1 + \theta_2 + \theta_x + \theta_y = 2\pi$$

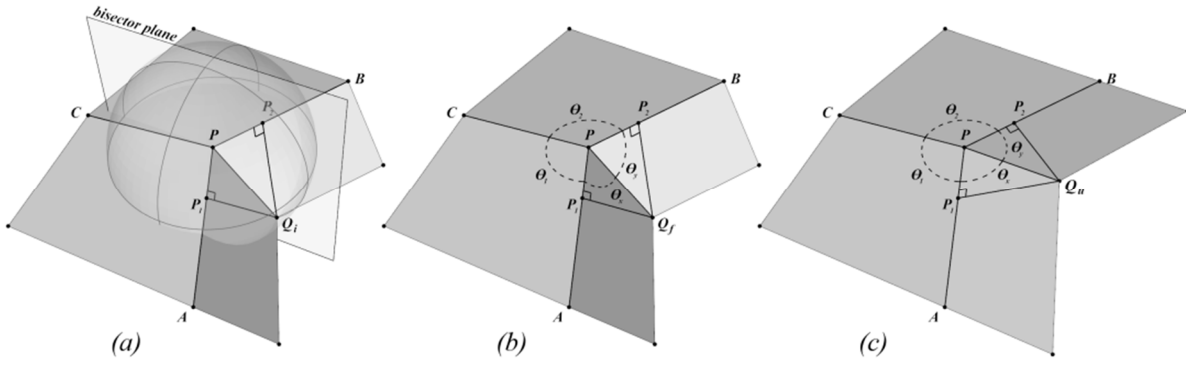


Figure 77. Computing the dependent surface. Image courtesy (Chandra et al. 2014)

As  $\{Q\}$  contains an infinite number of solutions, the solution space is constrained to the bisector plane of angle  $\angle APB$  (Figure 77a) and its intersection with a unit sphere centred at  $P$ , which limits the number of solutions to the folded-state with vertex  $Q_f$  (Figure 77b), and the unfolded-state with vertex  $Q_u$  (Figure 77c). We could also make assumptions of other linear and polynomial relations between  $\theta_x$  and  $\theta_y$ , which will result in higher order roots for  $Q_i$ . If no assumption is made regarding the relation between the two angles, the solution space  $\{Q\}$  would be the intersection of a quartic surface and the unit sphere centred at  $P$ .

If  $P_1$  and  $P_2$  are points on edges  $PA$  and  $PB$  respectively nearest to  $Q_i$ , and  $d_1$  &  $d_2$  are the distances from  $P$  to  $P_1$  and  $P_2$  respectively we can state that:

$$d_1 = d_2 = \sqrt{(\cos(2\pi - \theta_1 - \theta_2) + 1) * 0.5}$$

Thus the solution can be understood as the intersection of a unit sphere centred at  $P$ , with a line connecting  $Q_u$  and  $Q_f$ . Further, this line is the intersection of two planes, one centred at  $P_1$  with  $PA$  as its normal and the other centred at  $P_2$  and  $PB$  as normal. This results in a system with quadratic roots as shown below. This makes the length of the extrusion vector the only designer controllable parameter in the derived surface.

$$Q_u = P_0 + \lambda_u V \quad \text{and} \quad Q_f = P_0 + \lambda_f V$$

Where:

$$\lambda_u = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad \lambda_f = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$V = PA \times PB$$

$$a = V \cdot V$$

$$b = 2(V \cdot P)$$

$$c = (P_0 \cdot P_0 - 1)$$

$$P_0 = \langle x_0, y_0, 0 \rangle \quad x_0 = \frac{d_1 y_2 - y_1 d_2}{x_1 y_2 - y_1 x_2} \quad y_0 = \frac{x_1 d_2 - d_1 x_2}{x_1 y_2 - y_1 x_2}$$

$$d_1 = d_2 = \sqrt{(\cos(2\pi - \theta_1 - \theta_2) + 1) * 0.5}$$



## 5.2 Quasi-developable solutions

In summary, the *independent* surface is developable by construction. The crease vertices of the resulting geometry are also developable by virtue of the analytical solution described above. The *dependent* surface, on other hand is not developable i.e. the faces of the *dependent* quad strip are not planar. Unfolding a resultant, pre-dominantly quad mesh produces unfold error due to the non-planarity of mesh faces. This issue can be corrected in the 2D unfolded layout by merging spherical or hyperbolic vertices. This however would mean that the physical object will not match the digital 3D model. Constraints of time and the free-standing nature of the sculpture meant this issue was not addressed. However, in order to quantify the error in the geometry, several mesh resolution variants of the same base polyhedron (Figure 78), were digitally unfolded. The lower mesh resolution computed to significantly higher accuracy.

In practise however, these surfaces were almost planar or quasi-developable. For the purpose of architectural installations in paper and sheet aluminium, the resulting geometry was within acceptable tolerance for fabrication, as further illustrated by the reasonably precise edge alignment.

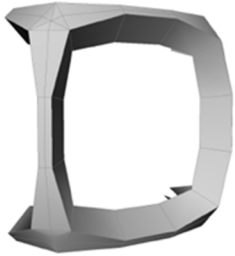

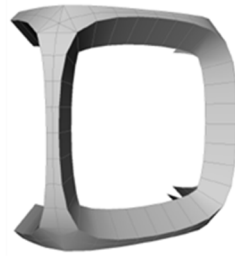
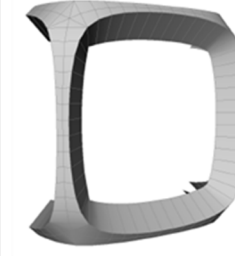
	DISCRETIZATION LEVEL 1	DISCRETIZATION LEVEL 2	DISCRETIZATION LEVEL 3	DISCRETIZATION LEVEL 4
				
AVG EDGE LENGTH	15.0	11.6	8.8	6.8
MINIMUM ERROR	0.076	0.093	0.124	0.129
MAXIMUM ERROR	0.137	0.164	1.05	1.54

Figure 78. Key results from unfold error measurements. Image courtesy (Chandra et al. 2014)

## 5.3 Manufacturing information

The resultant geometry is used to generate manufacturing information, in the form of unfolded CAD/ CAM data.

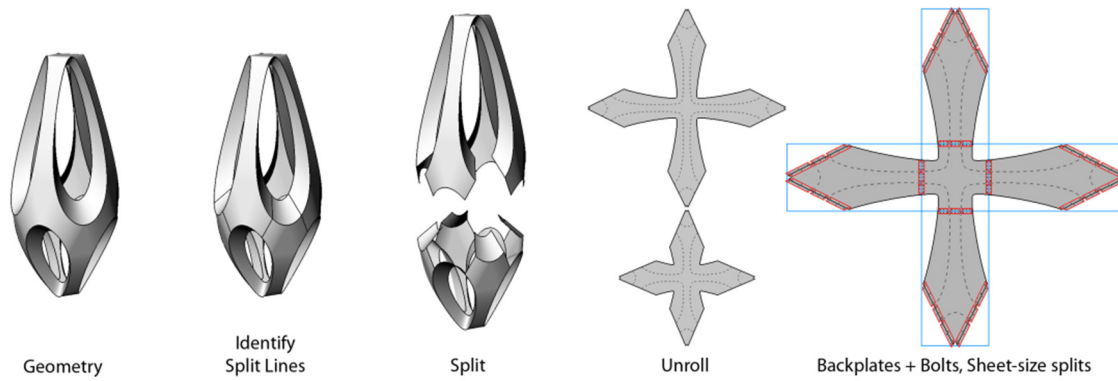


Figure 79. Process of CAD/ CAM info generation. Image courtesy (Chandra et al. 2014)

## 5.4 Summary

This prototyping exercise in a real-world design scenario greatly helped to:

- i. Develop computationally simple and manipulation friendly method that be can be used for specific low-resolution convex polyhedral geometries.

The ease of using polyhedral meshes and their manipulation using ubiquitous mesh modelling tools enabled speedy exploration of variations and is amenable with edit-and-observe strategies that designers tend to prefer (Figure 80).



Figure 80. Parametric variations of the independent and dependent surface computed from the same base polyhedron. Image courtesy (Chandra et al. 2014)

This intuitive nature of mesh editing tools lead to their incorporation, extension and formalised description in this dissertation (Chapter 6).

- ii. Demonstrate the useful-ness of CCF geometries in construction.

Apart from the obvious benefit of using sheet material to produce curved surfaces, they also ease the on-site description of geometry and assembly of parts due to their capacity

to be formed with minimal effort. In this exercise, 2 metre tall skeleton was built from sheet aluminium with teams of 10-15 students within a matter of 6-8 hours (Figure 81).



Figure 81. 2m long aluminium polyhedron, folded and assembled in 6 hours by a group of 15 students. Image courtesy (Chandra et al. 2014)

- iii. Prompted the understanding of the analytical solution for the position of boundary vertices, and revealed the potential to use the degree of freedom that they afford. This eventually was incorporated into the method described in Chapter (3).

## 6 PROCEDURAL GENERATION OF INPUT GEOMETRIES

The perturbation based method proposed in this research requires a user-defined mesh of pre-dominantly quad topology as its input. This enables the use of well-known and interaction-friendly mesh tools, to produce such an input mesh. This section describes some strategies in combining popular tools that operate locally on each of the vertices, edges or faces of an easy-to-manipulate *coarse* mesh, to produce a higher resolution mesh with topology that is appropriate as an input. However, it should be noted that some amount of experience and intuition regarding curve-crease folding is necessary. The fundamental theorems and properties of developable surfaces noted in Section (2.1.1) and Section (2.1.2), along with some of the design heuristics mentioned in Section (2.3.1) would be beneficial to know.

### 6.1 Pleated geometries

The simplest example of a curve-crease fold is one with a single crease or fold line. The discrete topology of such a mesh consists only of quads. The corresponding input geometry, therefore can be constructed by starting with one or more quads, and the use of edge-extrusion operations to form either a mountain or valley crease (Figure 82 a & b). A higher discretization can subsequently be achieved by converting the coarse mesh to its corresponding sub-divided and smoothed surface (Section 1.2.2) and deleting the edges rows parallel to the crease (Figure 82 –c & d). The recursive indexing scheme of sub-division surfaces makes it amenable to keeping track of the edges that need to be deleted. This however was not implemented in the research, and existing tools in Autodesk Maya were used to partially automate the process.

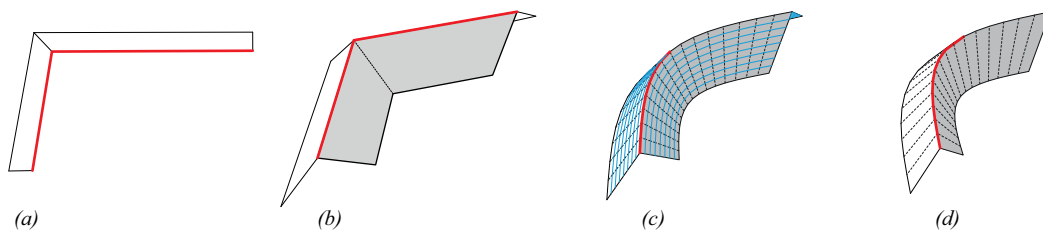


Figure 82 Use of *Extrude* operator on crease edges, Catmull-clark smoothing and edge-deletion to produce CCF topology

Successive application of extrusion operations can be combined to form alternate mountain and valley creases (Figure 83), and thus can form the basis of generating appropriate input meshes for pleated geometries. It can be noted that such operations and the use of subdivision surfaces, allow an intuitive link between prismatic origami and its CCF counterpart (Section 2.3.1 and Figure 21).

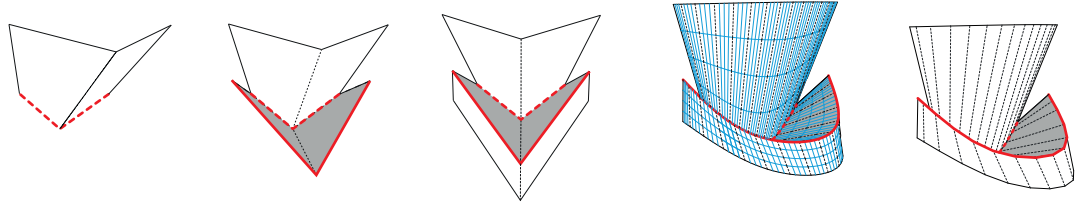


Figure 83 Successive use of *Extrude* operator on crease edges to produce pleated topologies

## 6.2 Mixed polygon topologies

The simplest example of a CCF geometry that requires its discrete representation to include a combination of triangles and quads is a 3-crease fold (Figure 84). The figure shows the use of *chamfer* and *bevel* Conway operators (Conway et al. 2008) on the vertices and edges of a cube, the corresponding subdivision surface and derived topology of rulings of such a tri-fold geometry. The steps (a-c) show:

1. The application chamfer operation on a particular vertex,
2. Bevel operation on the some edges. In terms of indexing, these edges will have indices corresponding to the edges of incident on the original vertex.
3. Deleting the original faces and extrusion of the border edges to form the outer row of faces.

Subsequent steps (e – f3) show the familiar conversion to a higher resolution mesh using the Catmull-Clark sub-division scheme, and deletion of edges parallel to the creases. In this case the deletion of highlighted edges makes additional intuitive sense in that rulings of CCF geometries cannot intersect except at conical parts (Solomon et al. 2012)( Section 3.2).

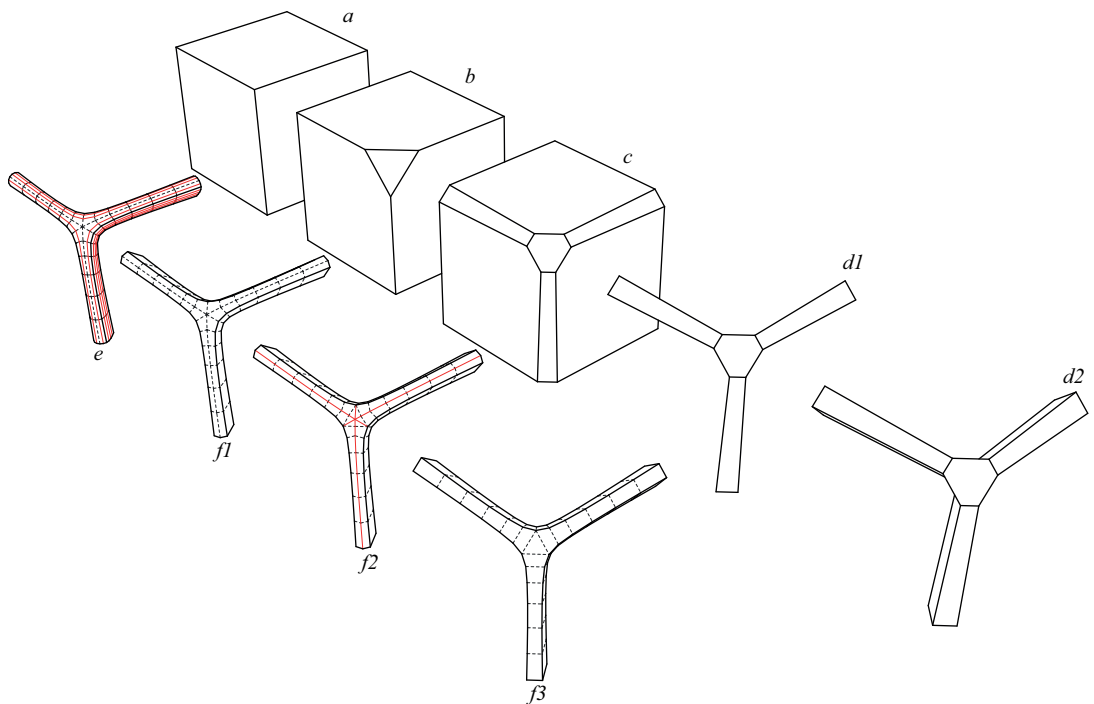


Figure 84 Use of Chamfer and Bevel operators, Catmull-Clark smoothing and edge-deletion to produce 3-crease CCF topology

Chamfer and Bevel operations as shown above, can be applied on a mesh with several vertices and edges to produce a topology of appropriate rulings. Figure 85, shows the application of such operations on a user-defined mesh with 24 vertices, to construct an appropriate CCF topology that was subsequently used to produce the physical prototype described in Chapter (4).

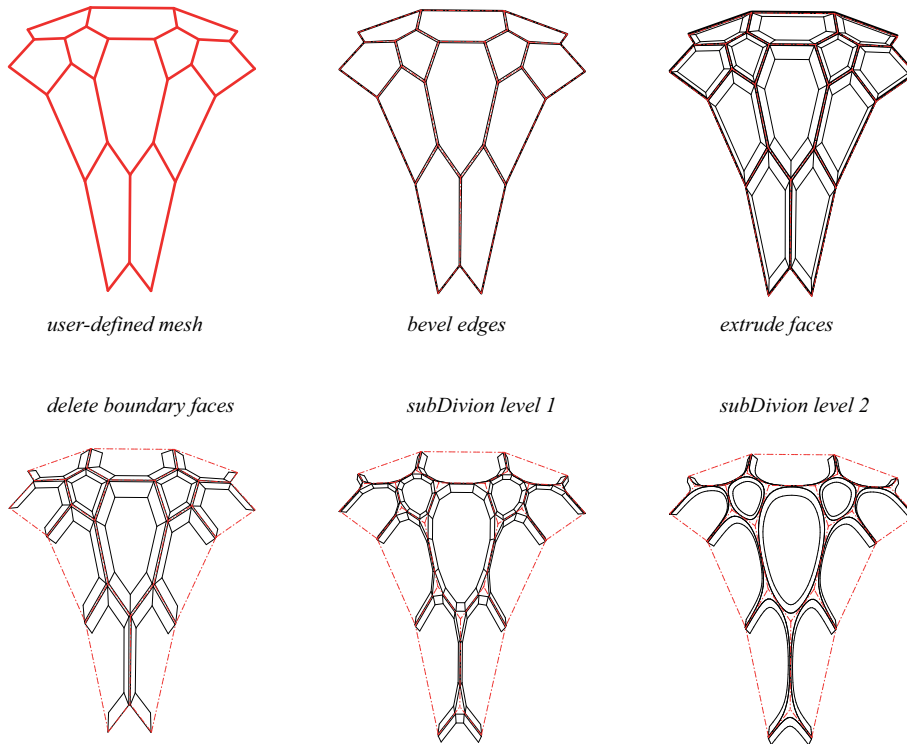


Figure 85 Application of Chamfer and Bevel operators from Figure 84 on mesh used to construct prototype 1 (Chapter 4)

All CCF topologies are combinations of quad-strips with triangles, which after perturbation, will represent discrete combinations of developable (planar-quad) strips intersecting with planar regions (triangles). Thus a combination of extrusion operations with chamfer and bevel can be used to explore a variety of input geometries.

## 7 SOFTWARE IMPLEMENTATION

The introduction and survey of the state-of-art in design and computational methods for curve-crease-folding (Chapters 1 & 2), attempted to provide context for the aspects that motivated the current dissertation: A computational method to design curve-crease-folded geometries, which is amenable to an edit-and-observe *modelling* paradigm (Section 2.5). Additionally it would operate directly on a user specified 3D geometry and find the nearest CCF geometry that would be suitable for manufacture. The literature review also highlighted major representational structures – smooth and discrete, along with the major algorithmic themes discerned in prior work: analytical methods, constructive techniques, approximating approaches, and modelling & simulation of thin-shells. The objectives mentioned above, and a proclivity towards edit-friendly mesh-based *modelling* paradigm involving minimisation of (Elastic) energy associated with CCF geometries meant that software implementation of this dissertation required the both an appropriate data structure(Section 7.1) and numerical techniques to minimise objective functions(Section 7.2).

### 7.1 Discrete Data structure

Given the context described above, a mesh data structure was a natural choice to represent the topology of CCF geometries. There are several well-established mesh-data structures used in computer graphics, simplest of them being the face-based structure which only maintains a list of vertices that form each face. However the discrete differential quantities that are an integral part of the *modelling* paradigm critically require easy computation of the 1-ring neighbourhood of every vertex i.e. a cyclically ordered list of the vertices that are connected to any given vertex. These are rather tedious to compute with a simple data structure and thus more involved data structures such as winged-edge (Baumgart 1972)and Half-edge (Kettner 1998)meshes are used. Both structures maintain, in addition to a list of vertices that form a face, additional topological information such as the two faces incident on an edge, the two vertices that make an edge including differentiating the *start* and *end* vertices, etc. Parameters that affect the choice between these representations include memory usage versus extent of processing required to compute adjacencies, such as edges emanating from a vertex, the faces incident at a vertex, the frequency of access to such information as vertex normals, face areas etc. This dissertation implemented a modified version of the simpler-and older *winged-edge* data structure. For more on mesh representation the reader is referred to Botsch et al. (2010).

### 7.2 Minimisation of functions

Minimisation of differentiable functions such as the elastic energy associated with developable and CCF geometries, involve more than solutions to linear system of equations as shown in Chapter 3). Subsequently they require numerical integration of ordinary differential equations (ODE) to find their maxima or minima.

#### 7.2.1 Runge Kutta Integration scheme

This dissertation implemented the well-known numerical approximation method called the Runge Kutta integration. Other popular choices are Euler integration and Verlet integration. For more on Numerical integration techniques and their comparative strengths, see Baraff et al. (1997). A nuance in the choice of integration schemes is deciding between implicit and explicit

schemes. Explicit schemes are easier to implement as most of them are variants of the simplest Euler method. Implicit schemes are harder to implement and are computationally more expensive since they require solutions to linear system of equations, every iteration. However they are numerically very stable. It can be noted that the explicit formulation of the integration scheme (RK4) was used in this dissertation. This is well-known to be unstable for certain types of time-dependent equations. This is particularly so for partial differential equations, such as the ones described in Chapter (3). This instability is acknowledged in Section (3.7.1) and the use of implicit schemes of integration is mentioned in the concluding chapter (Section 8.3.4).

## 7.2.2 Data-structures suitable for iterative and local methods

The data structures previously mentioned – winged and half-edge – are well suited for localised inspection of meshes. This makes them also suitable for use within *iterative* optimisation methods, which would require *localised* quantities such as curvature to be updated every iteration. The localised flexibility offered by these data structures also extends to dealing with varying number of sides in mesh faces – for example, a mix of triangles and quadrilaterals. This can be compared with *global* and non-iterative optimisation methods such as Laplacian-based mesh fairing (Botsch et al. 2010), which compute localised quantities once and solve for the *stationary points* of functions directly or non-iteratively within a finite number of steps. Additionally, such methods assume uniform discretization – triangular or quadrilateral, such that they are easier to describe within a matrix-based mathematical and data structure that is usually used in such methods.

## 7.3 Implementation

The stand-alone software application used in the dissertation was implemented using C++ 1.0, compiled to run on 64-bit Windows 7. The application follows the established Model-View-Controller paradigm (MSDN n.d.) by implementing the necessary call-back functions within *free-GLUT*, a portable and simple OpenGL software development toolkit and associated application programming interface (API). The call-back functions of *Setup* and *Update* (Figure 86) is used to initialise and modify the 3D *Model* containing the necessary topological, positional and optimisation data apart from variables and other infrastructure. The functions of *mousePress* and *KeyPress* are used to control the display or states of the model.

Following from Sections 7.1 & 7.2, a base *Mesh* class was implemented to manipulate topological data, and another class – *ActiveMesh* – was derived and extended from the base class to handle the optimisation aspects. The *Mesh* class includes methods to

- a. Compute Gaussian curvature and its gradient at each vertex
  - b. Compute best-fit-plane for each face
  - c. Compute distance from best-fit-plane and its gradient at each vertex
- Etc

The *ActiveMesh* class on the other hand is used to manage calls these methods for across all the vertices, accumulation of forces, integration of the resultant ODE and updating the vertex positions.



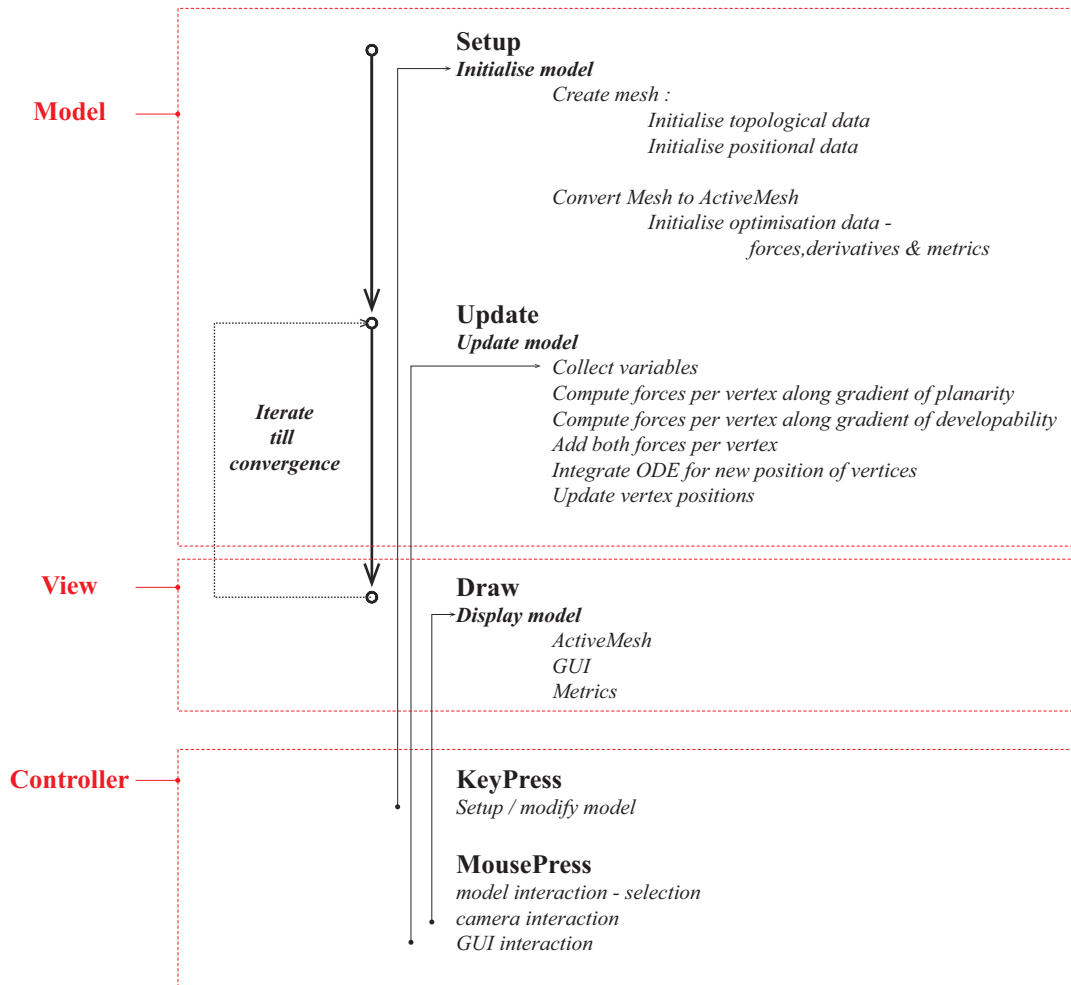


Figure 86 showing the implementation of the Model-View-Controller pattern of software design and its interaction with the proposed algorithm and data-structures described in sections 7.1 & 7.2

## 7.4 Existing software

Existing software that enables modelling of developable surfaces is limited. This extends also to curve-crease-folded geometries.

### 7.4.1 Continuous representation

In the continuous representation, the usual and often only means to model developable surfaces is to *loft* between free-form splines such that the resulting surface is developable. Most commercial CAD packages that include NURBS based tools (McNeel® Rhinoceros, Autodesk® Maya), include this feature. Frank Gehry, whose architecture is famously known of its use of developable surfaces, initiated the development of Digital Projects software platform (using CATIA as a foundation), to enable the design and construction of his projects (Glymph et al. 2004). Gehry technologies have since been a pioneering contributor to the development of tools to represent and manipulate developable surfaces for architectural use.

## 7.4.2 Discrete representation

Options available within discrete representation were, until recently, also limited. However the following tools are now available and they allow the design of curve-crease-folded geometries in addition to origami folding. It can be noted that all of these software, simulate the folding of planar layout of rulings into a 3D shape i.e. the inverse of the problem that is tackled in this dissertation.

### *Freeform origami*

As described in the introductory survey (Chapter 2 - Figure 27), Tachi (2009) developed one of the earliest and complete simulations of multi-fold prismatic origami. He subsequently extended his prismatic origami software to another called *Freeform Origami* which includes solutions to fold (and unfold), planar meshes with curved crease lines. Figure 87, shows the folding sequence for a three-crease mesh, as produced by this freely available and stand-alone software.

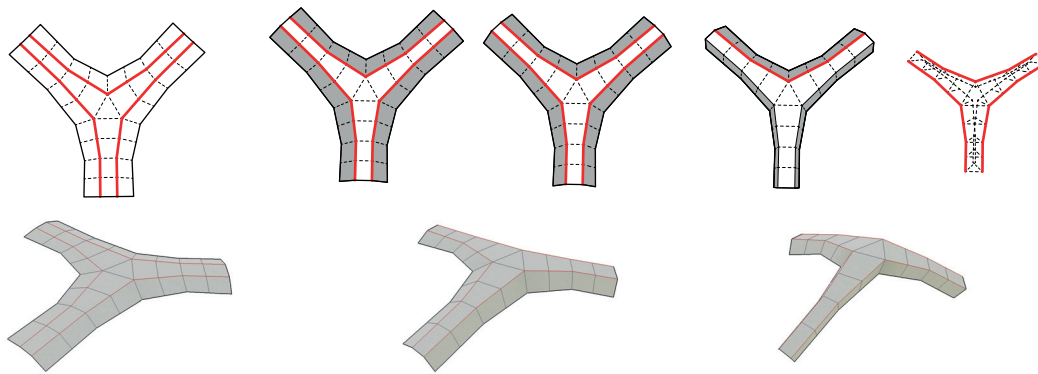


Figure 87 Folding sequence for a three-crease mesh, as produced by *Freeform Origami* software

### *Kangaroo plug-in for McNeel® Grasshopper*

Daniel Piker, previously an architect at Foster & partners, developed a plug-in called *Kangaroo*, for the visual programming interface of Grasshopper that operates within the CAD software of Rhinoceros™. This plug-in is a general purpose particle-spring simulator that allows the user to subject any given mesh to several custom physically-based and virtual *forces*. A recent extension (2013) to the plug-in enables the simulation of the folding process of a given (planar) layout of ruling. The plug-in achieves the simulation of rigid transformations by modelling rotations based on hinge-like forces across user-specified crease-edges.

### *King-kong plug-in for McNeel® Grasshopper*

The introduction of this dissertation mentions the recent developments in robotic folding of sheet metal. Gregory Epps, who holds the patent mentioned previously (Chapter 1, Figure 1), and his company roboFold®, also developed a plug-in called *King-Kong* for Grasshopper. It operates similarly to kangaroo. It is however, restricted to explore only folding simulations.

## 8 CONCLUSIONS AND FUTURE WORK

### 8.1 Summary

Curve-crease-folded geometries are a special case of intersecting and inter-dependent set of developable surfaces. As summarized in the introduction (Chapter 1), there are several disciplines contributing to the historic and continued development of the field: design, art, mathematics, computer graphics, industrial design etc. Compatibility with interactive, edit-friendly, exploratory, and well-established CAD work-flows was subsequently set-out as an explicit objective for the current research. Additionally, the resultant geometries were required to be suitable for manufacture.

Further, the survey of prior work (Chapter 2) highlighted the vast amount of prior work in the area of computational representation of developable surfaces, and their more recent extensions to curved crease folding. The survey noted several algorithmic approaches to computationally describe both developable surfaces and CCF geometries: Analytical, Geometric, approximation, and the modelling and simulation of thin shells. It also noted seminal contributions in the field of CCF – (Huffman 1976; Resch 1974; Mitani & Igarashi 2011; Demaine & O'Rourke 2007; Koschitz 2014), and the particular influence of Tachi & Epps (2011) and Kilian et al. (2008) on the current research (Section 2.5.2). Finally, it concluded by setting out the rationale for the method followed in this research (Section 2.8): a method to perturb input geometries towards their nearest CCF solution.

The research subsequently detailed an iterative and local method, based on the framework of Dynamic Relaxation (Day 1965), to minimize an objective function (Section 3.3). This perturbation method proposed in this dissertation, draws from the various approaches including the modelling paradigm of interactive computational tools (Section 2.5), the optimization approach of Kilian et al. (2008) and Wang & Tang (2004), and the physical-digital design strategies of Tachi & Epps (2011).

Two case-studies (Chapters 4 & 5) attempted to utilize the proposed method in realistic design and build scenarios. Case study 1 (Chapter 4) showed the compatibility between discrete funicular skeletal structures and the use of CCF moulds to cast concrete. The second case study (Chapter 5) similarly showed the ease of physical and on-site production and assembly of CCF panels.

The research noted the design strategies to develop planar layout of rulings and corresponding 3D folded geometries, as extensively described in Koschitz (2014) (Sections 1.1 & 2.3). The research built upon this exhaustive and intuitive compilation of David Huffman's design *gadgets* and proposed the use of Conway (mesh) operators (Hart n.d.; Conway et al. 2008) to develop equivalent (discrete) topologies in 3D (Chapter 6), that will be suitable for application of the perturbation method to produce CCF geometries. Finally, two important considerations – mesh data structures and numerical integration methods – and attendant details as required for implementation of the proposed iterative and local method are noted in Chapter (7).

### 8.2 Conclusions

Most surveyed methods presented difficulties when incorporated within an intuitive, interactive, edit-and-observe, exploratory method of design. This research aimed to overcome these

difficulties through the use of Dynamic Relaxation (DR)(Day 1965) for the (interactive) modelling of CCF geometries. It followed in the vein of discrete differential geometry methods and utilized discrete operators and their gradients (Meyer et al. 2003; Desbrun et al. 2002) within a DR framework, to perturb meshes to satisfy the geometric criteria of CCF geometries outlined in Kilian et al. (2008).

The proposed method was successfully applied to several examples with varying topological complexity (Section 3.6). It was also successfully tested in a realistic design-and-fabricate scenario (Chapter 4).

The method is implicitly dependent on the quality of the input geometries i.e. the input geometries should have an appropriate topology (Sections 3.1 & 3.2) to allow a successful *search* for a near-by CCF solution. The use of Conway operators to explore such appropriate inputs is shown in (Chapter 6). These operators are well-established in commercially available, mesh-based CAD software, and are thus, an intuitive tool-set to explore such topologies. The proposed method can be then used to visualize the *nearest* CCF solution. Such an edit-and-observe strategy is amenable to the usual CAD design workflow (Bhooshan & El Sayed 2012) (Section 1.2.1). As a positive side effect of user-defined input meshes, crease or fixed vertices need not be explicitly specified and are implicit in the input mesh. This further augments the ease of exploratory use.

Lastly, the method developed in this research, does not require the use of sophisticated numerical optimisation routines as the DR based method can easily be incorporated into a localised mesh data structure as noted in Section (7.1).

### 8.3 Limitations and Future work

#### 8.3.1 A specific, input dependent method

As noted above, the proposed method requires that the input geometries have an appropriate topology. As such it is not a generally applicable method, unlike the one proposed by Killian et al. The proposed method is based on the representation of a developable surface as single-row of quads i.e. without internal divisions. The overall CCF topology should consist of such one-row quad-strips meeting other one-row quad strips and/or triangles. The limitation on such input geometries is partially overcome by the use of well-known mesh operations(Section 6.2) and the manual application of heuristic measures such as those noted in Koschitz (2014)( Section 2.3.1). The production of a more comprehensive sequence of mesh-operation to produce appropriate 3D geometries would be a very useful compendium for designers.

Currently, these mesh-operations are almost procedural. However they require manual input in certain stages and cases (Sections 4.2.1 , 5.1.1 & 6.2). The extension of these to be fully procedural would significantly aid subsequent work such as conversion of prismatic, origami geometries to their CCF counterparts(Gattas & You 2014)(Figure 21 in Chapter 2 ).

In addition, the proposed method has been found to be sensitive to the levels of discretisation i.e. it performs best at lower mesh resolutions. Investigation to the cause and solution for such instability is thus a potential avenue of further work. The current hypothesis is that these are artefacts of the numerical integration scheme (Section 7.2.1).

### **8.3.2 Penalty functions**

Additional penalty functions to reduce deviation from the input mesh were not necessary in the case of the physical prototypes produced during this research, due to use of additional freedom afforded by the boundary vertices. This was also aided by the relatively good approximation of curve-foldable geometry provided by the input mesh. For generalized use however, penalty functions could be added as an additional virtual-force to minimize deviation from the input mesh, as several other authors do (Kilian et al. 2008; Poranne et al. 2013; Wang & Tang 2004)

### **8.3.3 Parameter dependency**

The proposed DR based method, currently requires weights for planarity and Gaussian curvature constraints to be chosen. The development of a robust mechanism to choose these weights in relation to the input geometry is an important aspect of future work. The current hypothesis is that development of unified, linear units of measurement for the two constraints would allow for such a choice.

### **8.3.4 Numerical stability**

Currently, an explicit numerical integration scheme is used to solve the iterative minimisation problem (Section 7.2.1). This is known to be unstable in certain cases, especially those involving partial derivatives or gradients. The incorporation of an implicit scheme therefore, is another aspect of future work.

### **8.3.5 Incorporation of 2D-3D workflow**

As the two case-studies in realistic design scenarios show (Chapters 4 & 5), it is often easier to correct certain errors and tolerances in the planar domain. A design work-flow that incorporates a folding simulation (Tachi & Epps 2011) and enables a 2D-3D correspondence would be very handy. This is particularly so, when the CCF geometries would have to interface with other geometry in a design context.

### **8.3.6 Materially-based simulation**

The proposed method follows a modelling paradigm (Section 2.5) i.e. an approach where a material constraint such as inextensibility of material is abstracted into geometric constraints such as planarity and vanishing Gaussian curvature. However, for full-scale industrial production of CCF geometries, the incorporation of a materially-based simulation (Section 2.5.1) would be ideal. Additionally, systemic quantification of material tolerances in the folding and unfolding process would be beneficial.

## 9 REFERENCES

- Aleksandrov, A.D. & Zalgaller, V.A., 1967. *Intrinsic geometry of surfaces*, American mathematical society Providence, RI.
- Bak, A., Shepherd, P. & Richens, P., 2012. Intuitive interactive form finding of optimised fabric-cast concrete. *Second International Conference on Flexible Formwork (icff2012)*.
- Balkcom, D.J. & Mason, M.T., 2008. Robotic origami folding. *Int. J. Rob. Res.*, 27(5), pp.613–627. Available at:  
<http://portal.acm.org/citation.cfm?id=1363297.1363299&coll=GUIDE&dl=GUIDE&CFID=70144537&CFTOKEN=65097458>.
- Baraff, D., Witkin, A. & Kass, M., 1997. Physically Based Modeling. Available at:  
[http://scholarworks.gvsu.edu/honorsprojects/82/?utm\\_source=scholarworks.gvsu.edu/honorsprojects/82&utm\\_medium=PDF&utm\\_campaign=PDFCoverPages](http://scholarworks.gvsu.edu/honorsprojects/82/?utm_source=scholarworks.gvsu.edu/honorsprojects/82&utm_medium=PDF&utm_campaign=PDFCoverPages).
- Barnes, M.R., 1999. Form Finding and Analysis of Tension Structures by Dynamic Relaxation. *International Journal of Space Structures*, 14(2), pp.89–104.
- Baumgart, B.G., 1972. *Winged edge polyhedron representation*, DTIC Document.
- Bergou, M. et al., 2006. A Quadratic Bending Model for Inextensible Surfaces. *Eurographics Symposium on Geometry Processing*, pp.227–230. Available at:  
[http://dl.acm.org/citation.cfm?id=1281987&http://www.researchgate.net/publication/221316563\\_A\\_quadratic\\_bending\\_model\\_for\\_inextensible\\_surfaces/file/d912f50a62fce5132d.pdf](http://dl.acm.org/citation.cfm?id=1281987&http://www.researchgate.net/publication/221316563_A_quadratic_bending_model_for_inextensible_surfaces/file/d912f50a62fce5132d.pdf).
- Bhooshan, S. et al., 2015. Applying dynamic relaxation techniques to form-find and manufacture curve-crease folded panels. *SIMULATION*, p.0037549715599849.
- Bhooshan, S., Bhooshan, V., et al., 2015. Curve-folded form-work for cast, compressive skeletons,. In *Proceedings of the SIMAUD 2015 Conference, Alexandria, USA*. Available at:  
[http://simaud.com/proceedings/download.php?f=SimAUD2015\\_Proceedings\\_HiRes.pdf](http://simaud.com/proceedings/download.php?f=SimAUD2015_Proceedings_HiRes.pdf).
- Bhooshan, S., El-Sayed, M. & Chandra, S., 2014. Design-friendly strategies for computational form-finding of curved-folded geometries: a case study. In *Proceedings of the Symposium on Simulation for Architecture & Urban Design*. Society for Computer Simulation International, p. 19.
- Bhooshan, S., Van Mele, T. & Block, P., 2015. Discrete funicular structures with curve-crease-folded moulds. In *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2015*. Amsterdam.
- Bhooshan, S. & El Sayed, M., 2012. Sub-division surfaces in architectural form finding and fabric forming. In *Second International Conference on Flexible Formwork, edited by JJ Orr, M. Evernden, AP Darby, and T. Ibell*. pp. 64–74.
- Bhooshan, S. & El Sayed, M., 2011. Use of sub-division surfaces in architectural form-finding and procedural modelling. In *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*. Society for Computer Simulation International, pp. 60–67.
- Bhooshan, S., Veenendaal, D. & Block, P., 2014. Particle-spring systems - Design of a cantilevering concrete canopy. In S. Adriaenssens et al., eds. *Shell Structures for Architecture: Form Finding and Optimization*. Routledge.
- Block, P. & Ochsendorf, J., 2007. Thrust network analysis\_a new methodology for three-dimensional equilibrium.pdf. *JOURNAL-INTERNATIONAL ASSOCIATION FOR SHELL AND SPATIAL STRUCTURES*, 155, p.167.
- Bo, P. & Wang, W., 2007. Geodesic-controlled developable surfaces for modeling paper bending.

- Computer Graphics Forum*, 26(3), pp.365–374.
- Botsch, M. et al., 2010. *Polygon mesh processing*, CRC press.
- Botsch, M. & Sorkine, O., 2008. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1), pp.213–230.
- Braun, R., 2015a. Foldout Couch. Available at: <https://vimeo.com/128060141>.
- Braun, R., 2015b. Foldout Couch The Making Of. Available at: [Foldout Couch The Making Of](#).
- Do Carmo, M., 1976. *Differential Geometry of Curves and Surfaces*. , p.497.
- Catmull, E., 1974. *A subdivision algorithm for computer display of curved surfaces*, DTIC Document.
- Chandra, S., Bhooshan, S. & ElSayed, M., 2014. Curved-Folding Convex Polyhedra through Smoothing. In *Proceedings of the 6th International Meeting on Origami in Science, Mathematics and Education*. Tokyo(Accepted).
- Chen, H.-Y. et al., 1999. On Surface Approximation Using Developable Surfaces. *Graphical Models and Image Processing*, 61(2), pp.110–124.
- Conway, J.H., Burgiel, H. & Goodman-Strauss, C., 2008. *The symmetry of things*. Wellesley, MA, A K Peters.
- Day, A.S., 1965. An introduction to dynamic relaxation. *The engineer*, 219, pp.218–221.
- Decaudin, P. et al., 2006. Virtual garments: A fully geometric approach for clothing design. *Computer Graphics Forum*, 25(3), pp.625–634.
- Demaine, E.D., Demaine, M.L. & Koschitz, D., 1999. Reconstructing David Huffman 's Legacy in Curved-Crease Folding.
- Demaine, E.D., Demaine, M.L. & Koschitz, R.D., 2011. Curved Crease Folding a Review on Art , Design and Mathematics Curved Creases in Art and Design. *Proceedings of the IABSE-IASS Symposium: Taller, Longer, Lighter (IABSE-IASS 2011)*.
- Demaine, E.D. & O'Rourke, J., 2007. *Geometric folding algorithms*, Cambridge university press Cambridge.
- Demaine, M., Portfolio. Available at: <http://erikdemaine.org/portfolio.pdf>.
- Deng, B., Pottmann, H. & Wallner, J., 2011. Functional webs for freeform architecture. *Computer Graphics Forum*, 30(5), pp.1369–1378. Available at: <http://doi.wiley.com/10.1111/j.1467-8659.2011.02011.x>.
- Desbrun, M., Meyer, M. & Alliez, P., 2002. Intrinsic Parametrization of Surface Meshes. *Eurographics*, 21(2).
- Dias, M. a. et al., 2012. Geometric mechanics of curved crease origami. *Physical Review Letters*, 109(11).
- Dolgachev, I. V., 2012. *Classical algebraic geometry: a modern view*, Cambridge University Press.
- Duncan, J.P. & Duncan, J.L., 1982. Folded Developables. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 383(1784), pp.191–205. Available at: <http://rspa.royalsocietypublishing.org/content/383/1784/191.abstract>.
- Eck, M. et al., 1995. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, pp. 173–182.
- Epps, G., 2010. Method for bending sheet material, bent sheet material and system for bending sheet material through attachment devices, US Patent 20100279842 A1. Available at: <https://www.google.ch/patents/US20100279842>.

- Esther Dora Adler, M. of A., 2004. *a New Unity!" the Art and Pedagogy of Josef Albers*. University of Maryland.
- Floater, M.S., 2002. One-to-one piecewise linear mappings over triangulations. *Mathematics of Computation*, 72(242), pp.685–697.
- Floater, M.S., 1997. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3), pp.231–250.
- Flux chair, 2009. Flux Chair. Available at: <http://www.fluxfurniture.com/items/chair/> [Accessed September 14, 2015].
- Folding Boat, Folding boat. Available at: <http://www.maxfrommeld.com/#/folding-boat/> [Accessed August 10, 2014].
- Frey, W.H., 2004. Modeling buckled developable surfaces by triangulation. *CAD Computer Aided Design*, 36(4), pp.299–313.
- Fuchs, D. & Tabachnikov, S., 1999. More on Paperfolding. *The American Mathematical Monthly*, 106(1), pp.27–35. Available at: <http://www.jstor.org/stable/2589583>.
- Gattas, J.M. & You, Z., 2014. Miura-Base Rigid Origami: Parametrizations of Curved-Crease Geometries. *Journal of Mechanical Design*, 136(12), p.121404. Available at: <http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4028532>.
- Gauss, F., 2014. Planar Panelization of Quad Meshes using Dynamic Relaxation Principles. In R. M.L.R.F & R. M. . Pauletti, eds. *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2014*.
- Glymph, J. et al., 2004. A parametric strategy for free-form glass structures using quadrilateral planar facets. , 13, pp.187–202.
- Grinspun, E., 2003. A discrete model of thin shells. *ACM Symposium on Computer Animation*, pp.62–67. Available at: <http://dl.acm.org/citation.cfm?id=1198555.1198663>.
- Gu, X., Gortler, S.J. & Hoppe, H., 2002. Geometry images. *ACM Transactions on Graphics*, 21(3), pp.355–361.
- Haga, K., Fonacier, J. & Isoda, M., 2008. *Origamics: mathematical explorations through paper folding*, World Scientific.
- Hanna, B.H. et al., 2015. Force–Deflection Modeling for Generalized Origami Waterbomb-Base Mechanisms. *Journal of Applied Mechanics*, 82(8), p.081001. Available at: <http://appliedmechanics.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4030659>.
- Harding, J. & Shepherd, P., 2011. Structural form finding using zero-length springs with dynamic mass. In *2011 IASS Annual Symposium: IABSE-IASS 2011: Taller, Longer, Lighter*. University of Bath.
- Hart, G., Conway Notation for Polyhedra. Available at: [http://www.georgehart.com/virtual-polyhedra/conway\\_notation.html](http://www.georgehart.com/virtual-polyhedra/conway_notation.html) [Accessed September 15, 2015].
- Hélein, F. & Wood, J.C., 2008. HARMONIC MAPS. *Handbook of global analysis*, pp.417–491.
- Hormann, K. & Greiner, G., 1999. MIPS : An Efficient Global Parametrization Method.
- Hormann, K. & Greiner, G., 2000. Quadrilateral remeshing. *Proceedings of Vision, Modeling and Visualization, 2000*, pp.153–162.
- Huffman, D. a., 1976. Curvature and Creases: A Primer on Paper. *IEEE Transactions on Computers*, C-25(10), pp.1010–1019.
- Hurdal, M.K. & Stephenson, K., 2004. Cortical cartography using the discrete conformal approach of



- circle packings. *NeuroImage*, 23, pp.S119–S128.
- Julius, D., Kraevoy, V. & Sheffer, A., 2005. D-Charts: Quasi-Developable Mesh Segmentation. *Computer Graphics Forum*, 24(3), pp.581–590. Available at: <http://doi.wiley.com/10.1111/j.1467-8659.2005.00883.x>.
- Kergosien, Y.L., Gotoda, H. & Kunii, T., 1994. Bending and creasing virtual paper. *IEEE Computer Graphics and Applications*, 14(1), pp.40–48.
- Kettner, L., 1998. Designing a data structure for polyhedral surfaces. In *Proceedings of the fourteenth annual symposium on Computational geometry*. ACM, pp. 146–154.
- Kilian, M. et al., 2008. Curved folding. *ACM Transactions on Graphics*, 27(3), p.75. Available at: <http://discovery.ucl.ac.uk/1328005/>.
- Le Klint, LE KLINT 172 A. Available at: <http://www.leklint.com/en-GB/Products/Show-Product/LE-KLINT-172-A.aspx>.
- Koschitz, R.D., 2014. *Computational Design with Curved Creases : David Huffman's Approach to Paperfolding*. Massachusetts Institute of Technology.
- Kyungeun, K., Bentley and Kyungeun KO. Available at: <http://www.robofold.com/assets/img/gallery/Karen-Bentley-WEB-6.jpg> [Accessed March 21, 2014].
- Lang, J. & Röschel, O., 1992. Developable (1, n) - Bézier surfaces. *Computer Aided Geometric Design*, 9(4), pp.291–298.
- Lang, R., 2004. Crease Patterns for Folders.
- Lévy, B. et al., 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics*, 21(3).
- Lienhard, J. et al., 2011. Flectofin: a hingeless flapping mechanism inspired by nature. *Bioinspiration & Biomimetics*, 6(4), p.045001.
- Lienhard, J., Schleicher, S. & Knippers, J., 2011. Bending-active Structures--Research Pavilion ICD/ITKE. In *Proceedings of the International Symposium of the IABSE-IASS Symposium*.
- Liu, L. et al., 2008. A local/global approach to mesh parameterization. *Eurographics Symposium on Geometry Processing*, 27(5), pp.1495–1504.
- Liu, Y. et al., 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics*, 25(3), p.681.
- Lobell, J., 2006. The Milgo Experiment : An Interview with Haresh Lalvani. *Architectural Design*, 76(4), pp.52–61.
- McCartney, J., Hinds, B.K. & Seow, B.L., 1999. The flattening of triangulated surfaces incorporating darts and gussets. *Computer-Aided Design*, 31(4), pp.249–260.
- McPharlin, P., 1944. shawinsky\_moma.jpg. *Paper Sculpture: Its Construction & Uses for Display & Decoration*, p.42. Available at: [http://erikdemaine.org/curved/history/shawinsky\\_moma.jpg](http://erikdemaine.org/curved/history/shawinsky_moma.jpg) [Accessed August 17, 2014].
- Meyer, M. et al., 2003. Discrete Differential-Geometry Operators for ' Triangulated 2-Manifolds. In H.-C. Hege & E. Al., eds. *Visualization and mathematics III*. Springer-Verlag Berlin Heidelberg, pp. 35–57.
- Mitani, J. & Igarashi, T., 2011. Interactive Design of Planar Curved Folding by Reflection. *Design*, pp.77–81.

- Mitani, J. & Suzuki, H., 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transactions on Graphics*, 23(3), p.259.
- Modular\_sculpture#0, 2006. modular\_sculpture#0. Available at: [http://www.richardsweeney.co.uk/modular\\_sculpture#0](http://www.richardsweeney.co.uk/modular_sculpture#0) [Accessed June 1, 2013].
- MSDN, Model-View-Controller. *MSDN*. Available at: <https://msdn.microsoft.com/en-us/library/ff649643.aspx> [Accessed November 10, 2014].
- Narain, R., Pfaff, T. & O'Brien, J.F., 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph.*, 32(4), pp.51:1–51:8. Available at: <http://doi.acm.org/10.1145/2461912.2462010> \n[http://dl.acm.org/ft\\_gateway.cfm?id=2462010&type=pdf](http://dl.acm.org/ft_gateway.cfm?id=2462010&type=pdf).
- Panozzo, D. & Jacobson, A., 2014. Chapter 5: Parametrization. *SGP Graduate School 2014*. Available at: [http://www.inf.ethz.ch/personal/dpanozzo/libigl\\_tutorial/tutorial.html#500](http://www.inf.ethz.ch/personal/dpanozzo/libigl_tutorial/tutorial.html#500) [Accessed September 15, 2015].
- Paper & Stick Film, 1992. The Ron Resch Paper & Stick Film. Available at: <https://vimeo.com/36122966>.
- Pedersen, O.E., Larsen, N.M. & Pigram, D., 2015. Post-tensioned Discrete Concrete Elements Developed for Free-form Construction. In *Advances in Architectural Geometry 2014*. Springer, pp. 15–28.
- Pérez, F. & Suárez, J. a., 2007. Quasi-developable B-spline surfaces in ship hull design. *CAD Computer Aided Design*, 39(10), pp.853–862.
- Pinkall, U. & Polthier, K., 1993. Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics*, 2(1), pp.15–36.
- Poranne, R., Ovreiu, E. & Gotsman, C., 2013. Interactive planarization and optimization of 3D meshes. *Computer Graphics Forum*, 32(1), pp.152–163.
- Pottmann, H. & Wallner, J., 1999. Approximation algorithms for developable surfaces. *Computer Aided Geometric Design*, 16(6), pp.539–556.
- Randlett, S., 1961. *The Art of Origami*, Faber & Faber.
- Resch, R.D., 1974. *the Space Curve As a Folded Edge*, ACADEMIC PRESS, INC. Available at: <http://linkinghub.elsevier.com/retrieve/pii/B9780120790500500175>.
- Resch, R.D., 1973. The topological design of sculptural and architectural systems. *Proceedings of the June 4-8, 1973, national computer conference and exposition on - AFIPS '73*, p.643. Available at: <http://portal.acm.org/citation.cfm?doid=1499586.1499744>.
- Robeller, C., Nabaei, S.S. & Weinand, Y., 2014. Robotic Fabrication in Architecture, Art and Design 2014. *Robotic Fabrication in Architecture, Art and Design 2014*, pp.67–81. Available at: <http://link.springer.com/10.1007/978-3-319-04663-1>.
- Rohim, R., Choong, K.K. & Jeevaratnam, T., 2013. Structural behaviour of origami inspired folded shell surface. *World Applied Sciences Journal*, 24(4), pp.497–502.
- Rose, K.L.P., 2007. *Modeling Developable Surfaces from Arbitrary Boundary Curves by*. The university of British Columbia.
- Saito, K., Tsukahara, A. & Okabe, Y., 2013. New Deployable Structures Based on an Elastic Origami Model. *Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2013 August 4-7, 2013*, 137(February), pp.1–7. Available at: <http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1830759>.

- Schek, H.-J., 1974. The force density method for form finding and computation of general networks. *Computer methods in applied mechanics and engineering*, 3(1), pp.115–134.
- Schenk, M., 2011. *Folded Shell Structures*. University of Cambridge.
- Schreck, C. et al., 2015. Interactively animating crumpling paper To cite this version : Interactively animating crumpling paper.
- Shampine, L.F. & Thompson, S., 2007. Stiff systems. *Scholarpedia*, 2, p.2855. Available at: [http://www.scholarpedia.org/article/Stiff\\_systems](http://www.scholarpedia.org/article/Stiff_systems).
- Sheffer, A., Praun, E. & Rose, K., 2006. Mesh Parameterization Methods and Their Applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2), pp.105–171.
- Shepherd, P. & Richens, P., 2010. Subdivision surface modelling for architecture.
- Shepherd, P. & Williams, C.J.K., 2010. British Museum Great Court.
- Solomon, J. et al., 2012. Flexible Developable Surfaces. *Computer Graphics Forum*, 31(5), pp.1567–1576.
- Sorkine, O. & Alexa, M., 2007. As-Rigid-As-Possible Surface Modeling. *Symposium on Geometry processing*, 4.
- Starostin, E.L. & van der Heijden, G.H.M., 2007. The shape of a Möbius strip. *Nature Materials*, 6(8), pp.563–567. Available at: <http://www.nature.com/doi/10.1038/nmat1929>.
- Tachi, T., 2010. Geometric Considerations for the Design of Rigid Origami Structures. *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2010*, 12, pp.458–460. Available at: <http://dx.doi.org/10.1016/j.mpaic.2011.07.005>.
- Tachi, T., 2009. Simulation of Rigid Origami. *Origami 4*, 4, pp.175–187.
- Tachi, T. & Epps, G., 2011. Designing One-DOF Mechanisms for Architecture by Rationalizing Curved Folding. *Proceedings of the International Symposium on Algorithmic Design for Architecture and Urban Design*.
- Tandem, T., Tiger tandem. Available at: [http://www.caroladrienne.com/roy\\_iwaki/the\\_NEW\\_FOLD/Photo\\_Graphics.html#67](http://www.caroladrienne.com/roy_iwaki/the_NEW_FOLD/Photo_Graphics.html#67).
- Tang, C. et al., 2015. Interactive design of developable surfaces. *ACM Transactions on Graphics*, VV, p.18. Available at: <http://arxiv.org/abs/1006.4903>.
- Terzopoulou, D. et al., 1987. Elastically Deformable Models. *ACM Transactions on Graphics*, 21(4), pp.205–214.
- Toponogov, V.A. & Rovenski, V.Y., *Differential Geometry of Curves and Surfaces*,
- Tutte, W., 1963. How to draw a graph. *Proc. London Math. Soc.*, 8(May 1962), pp.743–767. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.950&rep=rep1&type=pdf>.
- Vallet, B. & Lévy, B., 2009. What you seam is what you get. , (1), pp.1–9.
- Veenendaal, D. & Block, P., 2012. An overview and comparison of structural form finding methods for general networks. *International Journal of Solids and Structures*, 49(26), pp.3741–3753. Available at: <http://dx.doi.org/10.1016/j.ijsolstr.2012.08.008>.
- Vergauwen, A., Temmerman, N. De & Laet, L. de, 2014. Digital modelling of deployable structures based on curved-line folding. In R. M.L.R.F & R. M. . Pauletti, eds. *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2014*. Brasilia.
- Wakefield, D.S., 1999. Engineering analysis of tension structures: theory and practice. *Engineering Structures*, 21(8), pp.680–690.

- Wang, C.C.L., Smith, S.S.F. & Yuen, M.M.F., 2002. Surface flattening based on energy model. *Computer-Aided Design*, 34(11), pp.823–833.
- Wang, C.C.L. & Tang, K., 2004. Achieving developability of a polygonal surface by minimum deformation: A study of global and local optimization approaches. *Visual Computer*, 20(8-9), pp.521–539.
- Wang, C.C.L., Wang, Y. & Yuen, M.M.F., 2004. On increasing the developability of a trimmed NURBS surface. *Engineering with Computers*, 20(1), pp.54–64.
- Weiss, G. & Furtner, P., 1988. Computer-aided treatment of developable surfaces. *Computers & Graphics*, 12(1), pp.39–51.
- Weisstein, E.W., Descartes Total Angular Defect. *From MathWorld--A Wolfram Web Resource*. Available at: <http://mathworld.wolfram.com/DescartesTotalAngularDefect.html> [Accessed December 15, 2015].
- Weisstein, E.W., Gauss-Bonnet Formula. *From MathWorld--A Wolfram Web Resource*. Available at: <http://mathworld.wolfram.com/Gauss-BonnetFormula.html> [Accessed December 15, 2015].
- van Wijk, J.J., 2008. Unfolding the Earth: Myriahedral Projections. *The Cartographic Journal*, 45(1), pp.32–42.
- Yates, H., 1965. Foldable shelter structure with zig-zag roof profile, US patent 3407546A. Available at: <https://patents.google.com/patent/US3407546A/en?q=zig-zag+roof+yates&inventor=H+Yates>.
- Yellow Cones Kissing with Ron Resch, Yellow Cones Kissing with Ron Resch. *photograph*. Available at: [http://www.ronresch.org/ronresch/gallery/yellow-folded-cones-kissing/Yellow Cones Kissing-with Ron.1-Crop-1024.jpg/image\\_view\\_fullscreen.html](http://www.ronresch.org/ronresch/gallery/yellow-folded-cones-kissing/Yellow%20Cones%20Kissing-with%20Ron.1-Crop-1024.jpg/image_view_fullscreen.html) [Accessed August 25, 2014].
- Yosuke Otomo, stir. Available at: <http://yukonishimura.com/works-en.php> [Accessed July 15, 2014].
- Zhong, Y. & Xu, B., 2006. A physically based method for triangulated surface flattening. *Computer-Aided Design*, 38(10), pp.1062–1073.
- Zhu, L., Igarashi, T. & Mitani, J., 2013. Soft folding. *Computer Graphics Forum*, 32(7), pp.167–176.