



Doctoral Thesis

**QUAD MESHES AS OPTIMIZED ARCHITECTURAL  
FREEFORM STRUCTURES**

submitted in satisfaction of the requirements for the degree of  
Doctor of Science  
of the Vienna University of Technology, Faculty of Civil Engineering

---

Dissertation

**VIERECKSNETZE ALS OPTIMIERTE ARCHITEKTONISCHE  
FREIFORMSTRUKTUREN**

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines  
Doktors der technischen Wissenschaften  
eingereicht an der Technischen Universität Wien, Fakultät für Bauingenieurwesen  
von

Dipl.-Ing. Davide Pellis  
Matrikelnummer 11728982  
Buchengasse 34/1/19  
1100 Wien, Austria

Betreuer: Univ. Prof. Dr. Helmut Pottmann  
Inst. of Discrete Mathematics and Geometry, TU Wien  
Wiedner Hauptstraße 8–10/104  
A-1040 Wien, Austria

Betreuer: Assoc. Prof. Dr. Josef Füssl  
Inst. for Mechanics of Materials and Structures, TU Wien  
Karlsplatz 13/202  
A-1040 Wien, Austria

Wien, October 2019

---

# Quad Meshes as Optimized Architectural Freeform Structures

by

Davide Pellis

Copyright © 2019

Davide Pellis

All rights reserved



# Abstract

This thesis tackles the design of freeform surface-like and load-bearing structures realized with cladding panels and supported by a framework substructure, often called *gridshells*. The actual fabrication of freeform gridshells is a challenging task, and easily leads to unsustainable costs. A well known strategy to realize a gridshell is to use as layout a so-called *principal mesh*. This is a quadrilateral mesh whose edges follow the principal curvature directions of a continuous surface. We achieve in this way flat cladding panels and a substructure with simplified connections.

This thesis shows that quadrilateral meshes, besides allowing manufacturing simplification, are also optimal solutions both for static performance and smooth visual appearance. In particular, we show that the best static performance is achieved for quad meshes discretizing membranes along principal stress lines, and we get an absolute minimum on such membranes where the integral of absolute principal stresses is minimal. We also show that the best smooth visual appearance is achieved for principal meshes; the absolute minimum is now reached for principal meshes discretizing surfaces where the integral of absolute principal curvatures is minimal. Therefore, from membranes where stress and curvature directions are aligned, and where the total absolute stress is minimal, we can extract principal meshes with the best static performance and with optimal visual appearance. We present then computational tools for the design of such highly efficient gridshells.

# Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
1.1	Motivation . . . . .	9
1.2	Previous work . . . . .	10
1.3	Overview and contribution . . . . .	11
<b>2</b>	<b>The geometry of meshes</b>	<b>13</b>
2.1	Meshes in architecture . . . . .	13
2.1.1	What is a mesh . . . . .	14
2.1.2	Triangular, quadrilateral and hexagonal meshes . . . . .	15
2.1.3	Offset meshes . . . . .	17
2.2	Meshes at the limit of refinement . . . . .	19
2.2.1	Parametric surfaces . . . . .	19
2.2.2	Curvature of surfaces . . . . .	21
2.2.3	Height field parametrization . . . . .	22
2.2.4	Conjugate and principal meshes . . . . .	23
<b>3</b>	<b>The geometry of equilibrium</b>	<b>26</b>
3.1	Equilibrium of gridshells . . . . .	27
3.1.1	The mechanical model . . . . .	27
3.1.2	Nodal equilibrium . . . . .	27
3.1.3	The force polyhedron . . . . .	28
3.2	Equilibrium at the limit of refinement . . . . .	29
3.2.1	The stress tensor . . . . .	30
3.2.2	Membrane stress . . . . .	30
3.2.3	Equilibrium under vertical load . . . . .	31
3.2.4	The Airy stress surface . . . . .	32

<b>4</b>	<b>Principal meshes in equilibrium</b>	<b>34</b>
4.1	Overview and contribution . . . . .	34
4.2	Previous work . . . . .	35
4.3	Aligning principal stress and curvature directions . . . . .	36
4.3.1	Principal meshes at the limit of refinement . . . . .	37
4.3.2	Estimating stress and curvature . . . . .	38
4.3.3	Design workflow . . . . .	40
4.4	Implementation . . . . .	41
4.4.1	Main variables and constraints . . . . .	41
4.4.2	Solver . . . . .	43
4.5	Results and discussion . . . . .	44
<b>5</b>	<b>Material-minimizing gridshells</b>	<b>47</b>
5.1	Overview and contribution . . . . .	48
5.2	Previous work . . . . .	49
5.3	Michell trusses . . . . .	50
5.3.1	Problem statement . . . . .	50
5.3.2	Equilibrium and displacement . . . . .	51
5.3.3	Volume and stress limit . . . . .	51
5.3.4	Maxwell lemma . . . . .	52
5.3.5	Dual formulation . . . . .	53
5.3.6	Continuum formulation . . . . .	55
5.3.7	Back to the primal problem . . . . .	57
5.4	Volume-optimal trusses in 2D . . . . .	59
5.4.1	Connection between kinks and volumes . . . . .	60
5.4.2	Total isotropic curvature . . . . .	61
5.4.3	Computing optimal trusses in 2D . . . . .	63
5.5	Volume-optimal gridshells . . . . .	65
5.5.1	Properties of optimal gridshells . . . . .	65
5.5.2	Properties of optimal truss-like continua . . . . .	66
5.5.3	Computing optimal structures – the workflow . . . . .	67
5.6	Implementation . . . . .	68
5.6.1	Variables and constraints for optimal gridshells . . . . .	69
5.6.2	Counting degrees of freedom . . . . .	72

<i>CONTENTS</i>	5
5.6.3 Target functional for optimization . . . . .	73
5.6.4 Further constraints . . . . .	74
5.6.5 Solver . . . . .	75
5.6.6 Variables and constraints for optimal 2D trusses . . . . .	75
5.7 Discussion . . . . .	75
5.7.1 Verification of results . . . . .	76
5.7.2 Implementation details . . . . .	77
5.7.3 Robustness . . . . .	78
<b>6 Visual smoothness of meshes</b>	<b>81</b>
6.1 Overview and contribution . . . . .	82
6.2 Previous work . . . . .	83
6.3 Representation of saddles in meshes . . . . .	85
6.3.1 Saddle-shaped meshing of saddle-shaped surfaces . . . . .	86
6.3.2 The normal pyramid . . . . .	87
6.4 Energies of polyhedral surfaces . . . . .	89
6.4.1 $\mathcal{E}$ -minimal meshes . . . . .	90
6.4.2 Differential-geometric interpretation of the energy $\mathcal{E}$ . . . . .	94
6.4.3 Total absolute curvature of surfaces . . . . .	96
6.5 Implementation . . . . .	99
6.6 Results and Discussion . . . . .	100
<b>7 Conclusion</b>	<b>102</b>
7.1 Limitations . . . . .	103
7.2 Further research . . . . .	104
<b>Bibliography</b>	<b>105</b>

# Acknowledgements

I would like first to thank my supervisor Helmut Pottmann for guiding me in this fascinating field, and my former supervisors Carlo Biagini and Gianni Bartoli for letting me the freedom to follow my passions. A special thanks goes to Johannes Wallner and Martin Kilian for their invaluable work and advice. Many thanks also go to Heinz Schmiedhofer for his contribution to some of the images and to Felix Dellinger for his master thesis work.

I would also like to give warm thanks to Doris Hotz, for her precious help to organize my defense, and to all my colleagues here at TU Wien that made me feeling at home — Sara Andreussi, Martin Bauer, Andy Fuksas, Konstantinos Gavriil, Ronald Haidvogel, Michael Jimenez, Kurt Leimer, Maria Lara Mirò, Christian Müller, Klara Mundilova, Przemyslaw Musialski, Mason Pember, Martin Peternell, Stefan Pillwein, Arvin Rasoulzadeh, Dino Rossinger, Luca Francesco San Mauro, Birgit Slama, Gudrun Szewieczek, and Hui Wang. Moreover, a big thanks to my colleagues and friends at University of Florence — Antonio Annis, Giulia Buffi, Vincenzo Donato, Laura Ierimonti, Silvia Monchetti, Laura Nardi, Tommaso Pacetti, Lorenzo Piscitelli, Giovanna Ramaccini, Luca Taglialegne, and Sara Venturi. Finally, I heartily thank Feray Bayar for the wonderful time we spent here in Vienna together.

This research was supported by SFB-Transregio programme *Discretization in Geometry and Dynamics* (Austrian Science Fund grant no. I 2978) and by the project “Geometry and Computational Design for Architecture and Fabrication” at TU Wien.



# Chapter 1

## Introduction

The development of computational tools and manufacturing techniques is continuously pushing forward the limit of feasible shapes in architecture. Along with new creative freedoms, new challenging issues arise on structural design, economic sustainability and, more generally, on architectural style. Using the words of the engineer Pier Luigi Nervi — structural architecture does not admit arbitrary or formal solutions even if, from a technical point of view, the ever increasing mechanical qualities of materials and the precision of computational methods make possible, even if with a guilty economic sacrifice, artificial or over-intellectual solutions. Creative action must remain today, as in the past centuries, a pure intuitive act guided by static sense [37].

This thesis is inspired by the idea, shared by Nervi and always chased by many architects, of an architectural design where form is driven by statics, manufacturing, material economy, and other aspects which have implications on construction and cost. Pursuing this idea, this work tackles the design of freeform structures that optimally perform the load-bearing function by virtue of their own shape, and aware of their buildability.

A long-term goal of computational design are tools which embed these aspects while modeling geometric shapes, assisting architects since the earlier creative process. Even small steps towards this goal can shorten the design loop and increase the creative control of designers on the final result.

## 1.1 Motivation

This thesis revolves around a prominent class of structures in freeform architecture, namely surface-like load-bearing shells, subdivided into cladding panels and supported by a framework substructure. We refer to these structures as *gridshells*. Due to the high geometric complexity and low form repetitiveness of their building components, together with high static demand, the manufacturing of such structures can lead to unsustainable costs.

A cost reduction strategy is the subdivision of the surface with planar cladding panels together with the use of a framework made up of prismatic beams. Building components can then be easily cut out from flat sheets and straight elements. The manufacturing can be further simplified if the beams of the substructure meet in the nodes along a common axis. We have in this case a *torsion-free substructure*. It is well known [31, 45] that all these goals can be achieved through the subdivision of a surface with a *principal mesh*, that is a quadrilateral mesh whose edges follow the principal curvature directions of a continuous surface. Several tools for the extraction of principal meshes from a given surface are currently available, e.g. [13].

On the statics side, a common strategy for saving structural material is to ensure equilibrium in the framework through strictly axial forces. This allows the material to be stressed in the most efficient manner, while increasing the stiffness of the structure. At this purpose, several design methods have been developed, e.g. [1, 55, 56].

Problems arise when we ask for principal meshes that are also in axial force equilibrium. Proper design tools are here currently missing: the available methods work only on meshes with a given connectivity; due to an intrinsic conflict, the optimization often fails. Furthermore, among all frameworks in axial force equilibrium over a given boundary, the achievement of the most efficient form and framework layout could significantly reduce the cost of the load-bearing structure. A mesh-tailored tool for such a task is also currently missing. Moreover, the aforementioned manufacturing simplifications come at the price of sacrificing the smooth appearance of the surface. Indeed, when dealing with reflective materials such as metal or glass, widely used in architecture, the kinks between cladding panels are strongly enhanced. A



strategy for improving the visual appearance of a polyhedral surface could be beneficial for architectural design.

This thesis aims at filling these gaps in the computational design process of load-bearing freeform structures.

## 1.2 Previous work

The properties of quadrilateral meshes relevant for architecture have been extensively studied in the context of a wider research field called *architectural geometry*. The main results on this topic can be found in the works of Bobenko et al. [11], Liu et al. [31], and Pottmann et al. [45]. These theoretical results have also been applied in actual building constructions. A torsion-free layout has been used for the skin of the Yas hotel in Abu Dhabi, by Asymptote Architecture. Examples of principal quad meshes in architecture are the Roppongi Canopy in Tokyo, realized in 2005 by Buro Happold, and the more recent roof of the Chadstone shopping centre in Melbourne. A comprehensive survey on the achievements of architectural geometry can be found in [43].

On the side of equilibrium of meshes, this thesis relies on the seminal work on *thrust network analysis* of Block and Ochsendorf [10] and the geometric approach to equilibrium of Vouga et al. [56]. One of the main achievements on self-supporting freeform structures is the Armadillo Vault exhibit in 2016 Biennale.

Regarding structural material economy, this thesis is based on the groundbreaking paper *The Limits of Economy of Material in Frame-structures* of 1904 by Michell [34], and was strongly motivated by some of the outcomes of the recent work *A Limit of Economy of Material in Shell Structures* by Mitchell [36]. A survey on the available methods for structural design and optimization of architectural shapes can be found in [1].

On the computational side, this work is strongly based on the *guided projection* of Tang et al. [55]. Besides giving a constraint solver framework, this method represents also the state of the art for the optimization of meshes with given connectivity for architecture. Among many constraints relevant for building construction, it includes planarity of faces, static equilibrium,

and mesh polylines fairness for aesthetic quality control.

More detailed states of the art of the specific topics covered in this thesis are given in chapters 4 to 6.

### 1.3 Overview and contribution

The thesis starts with an overview on known material. Chapter 2 introduces meshes and continuous surfaces. In particular, it shows how a quad mesh with planar faces and a torsion-free substructure, at the limit of refinement, approaches the principal curvature network of a continuous surface. Chapter 3 introduces the equilibrium of meshes under vertical loads and the equilibrium of their limit surface, that mechanically is a membrane.

The main contributions start in chapter 4, where the design of principal meshes in equilibrium is addressed. It is shown that these meshes are discretizations of special surfaces in membrane equilibrium where principal stress and curvature directions coincide. A tool for the design of such surfaces is then presented.

Chapter 5 seeks for the layout and the form of gridshells that minimize the demand of structural material. It turns out that quad meshes following principal stress lines of surfaces in membrane equilibrium are the minimizing layouts, and the global minimizer for given boundary conditions is reached for such layouts discretizing membranes where the sum of the absolute principal stresses is minimal. We introduce then a tool for the design of material-minimizing gridshells, together with other constraints relevant for manufacturing and cost reduction such as principality. We achieve in this way a highly efficient design solution, namely a gridshell with flat cladding panels, a torsion-free substructure, and which requires the minimal amount of structural material to span over a given boundary.

Finally, chapter 6 tackles the reflective appearance of polyhedral surfaces, looking this time for the layout and form with the best visual behavior. It turns out here that principal meshes are the “smoothest” layouts discretizing a given surface, while the polyhedral surfaces with the smoothest possible visual appearance are those principal meshes discretizing surfaces where, for a given boundary, the sum of absolute principal curvatures is minimal. A

method for the design of such surfaces is then outlined. This result further strengthens the already proven optimality of principal meshes in architecture.

Methodologically, this work moves on a common thread, namely an iterative refinement process where, at the limit, a mesh approaches a continuous surface. Here, we use tools belonging to differential geometry and continuum mechanics, such as curvature and stress, we get theoretical insights at the limit of refinement, and then we go back to meshes with relevant properties through a discretization of optimal continuous solutions — the geometry and the connectivity of the mesh arise from the continuous formulation. We overcome in this way the main limitation of current methods, i.e. working on predetermined mesh connectivities.

When coming at computation, all problems are set up in the following way. A continuous surface is modeled through a triangular mesh. The solution of the continuous problem relies on discrete differential operators. From the resulting mesh, we extract optimized vector fields that will guide a quadrilateral remeshing. The development of theoretical insights and the implementation of this procedure for the aforementioned problems is the main contribution of this work. Quad mesh extraction from the resulting triangular meshes can be performed with existing methods such as *mixed integer quadrangulation* [13]. A final round of post-optimization of the resulting quad mesh, based for instance on *guided projection* [55], gives then the final result.

The results presented throughout this thesis have been first published in [28, 40, 41].

# Chapter 2

## The geometry of meshes

This thesis tackles freeform surface-like architectural skins, subdivided into panels supported by a frame substructure. We refer to these structures as *gridshells*. The geometry of a gridshell can be represented by a mesh, that is, broadly speaking, a discrete representation of a continuous surface.

This chapter introduces meshes in architecture. In section 2.1, it is shown how some relevant properties for building construction, such as planar cladding panels and a torsion-free substructure, can be achieved thanks to meshes owning special properties. In section 2.2, it is then shown how these properties are intrinsically connected to the continuous surface that the mesh discretizes. This connection is found at the limit of a refinement process that increases the density of the mesh, until it converges to the continuous surface itself.

### 2.1 Meshes in architecture

The geometric properties of meshes relevant for architecture have been extensively studied by *architectural geometry*, a research field born around 2005 to solve the problems arising from the design and manufacturing of freeform architectures. This area of research combines results belonging to different fields such as discrete differential geometry, computer graphics, and numerical optimization. An extended introduction to architectural geometry can be found in the book of Pottmann et al. [42], while a comprehensive survey is

given in [43]. After a short introduction to meshes, this section summarizes the main results of architectural geometry concerning meshes relevant for gridshell design.

### 2.1.1 What is a mesh

From a geometric point of view, a *gridshell* is a framework of beams connected together to form a surface-like structure. If the axes of the beams are straight lines, a gridshell can be properly represented by a polygonal mesh. Roughly speaking, a *polygonal mesh* is a collection of points, called *vertices*, arranged together into *faces*. Faces are bounded by polygons, with sides given by straight lines connecting the vertices. The sides are called *edges*.

More formally, a polygonal mesh is a tuple  $M = (V, E, F)$ , where  $V$ ,  $E$  and  $F$  are, respectively, the sets of vertices, edges and faces. Vertices are points  $\mathbf{v}_i \in \mathbb{R}^3$  (or  $\mathbb{R}^2$ ),  $i \in (1, 2, \dots, |V|)$ . Edges are sets of two connected vertices  $e_k = \{\mathbf{v}_i, \mathbf{v}_j\}$ ,  $k \in (1, 2, \dots, |E|)$ . Faces are ordered sets of vertices  $f_l = (\mathbf{v}_i, \mathbf{v}_j, \dots, \mathbf{v}_n)$ ,  $l \in (1, 2, \dots, |F|)$ , where each vertex is connected by an edge with the next one, and the last with the first one.

The way in which the vertices are connected in edges and faces concerns the *connectivity* of the mesh, while the position of the vertices deals with its *geometry*. Regarding its geometry, a mesh can be seen as a discrete approximation of a two dimensional surface, referred to as *reference surface*. For a wide introduction to meshes, see [14].

From an architectural point of view, the elements of a mesh can properly represent the main structural components of a gridshell: the faces can correspond to cladding panels, the edges to the axes of the substructure, and the vertices to the substructure joints. The axes, extruded along a direction approximately normal to the reference surface, can represent the symmetry planes of the beams. The geometry of a mesh can then embed the geometry of the building components of a gridshell. Meshes with planar faces play here a significant role: indeed, it is convenient to realize the cladding panels out of planar elements, since the production cost of double curved elements is often prohibitive for architectural applications. Moreover, if dealing with a substructure made of prismatic beams, it is desirable that their symmetry



**Figure 2.1** – The three types of gridshells. Triangular (left), British museum courtyard by Foster and Partners. Quadrilateral (center), Yas Hotel Abu Dhabi by Asymptote Architecture. Hexagonal (right), Eden project by Nicholas Grimshaw.

planes meet in the vertices along a common axis. This simplifies the manufacturing of the nodes, besides improving their aesthetic. We talk in this case of a *torsion-free substructure*. In the following of this section, we will see how these useful properties are connected with the connectivity and the geometry of a mesh.

### 2.1.2 Triangular, quadrilateral and hexagonal meshes

Regarding its connectivity, a mesh is said to be *regular* if all its face polygons have the same number of sides. There are only three ways to tile the plane with equal regular polygons: with equilateral triangles, with squares, and with regular hexagons<sup>1</sup>. According to this fact, we can classify regular meshes into three main groups, depending on whether their faces are all triangles, quadrilaterals, or hexagons. We talk, respectively, about *triangular*, *quadrilateral*, and *hexagonal meshes*. For architectural applications, each of these meshes has its strengths and weaknesses.

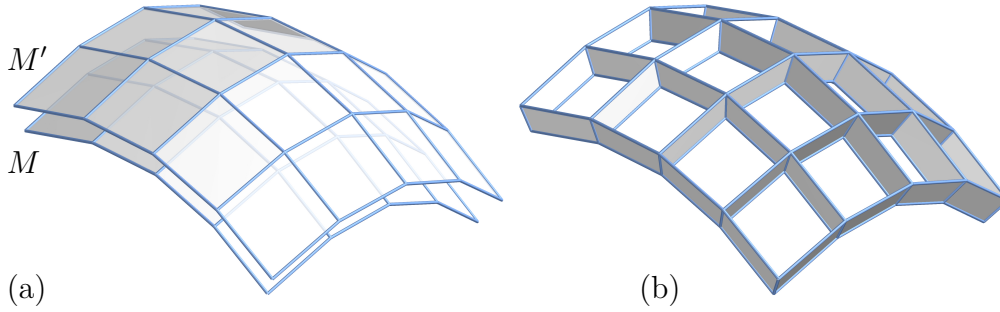
- *Triangular meshes* have been extensively used in architecture, in particular because their faces are planar. Indeed, the vertices of a non degenerate triangle always define a plane. The main drawback of triangular gridshells is in the substructure. In a triangular mesh, generally, six edges meet together at a vertex: the manufacturing of such

<sup>1</sup>This can be easily seen considering that the angles of polygons meeting at each vertex must sum up to  $360^\circ$ . With three polygons, we get  $120^\circ$  angles, corresponding to the hexagon. Four polygons give  $90^\circ$  (a square), while five give  $72^\circ$ , impossible to fill by a regular polygon. Six polygons gives  $60^\circ$ , an equilateral triangle. No other polygons exist with less than three sides.

a node is awkward. Moreover, let us imagine we want a torsion-free substructure, and consider a single face and its bounding edges. The three symmetry planes, corresponding to the three edges, will intersect in a single point (possibly at infinity), and so do the three node axes, given by the pairwise intersections of the three planes. For each other face sharing an edge with this face, two node axes are already given. The third one must pass through their intersection point, and then it is already determined. Therefore, in a triangle mesh, all node axes of a torsion-free substructure must intersect in a single point. This gives a high restriction to feasible shapes; if we want all node axes exactly orthogonal to the reference shape, this is only possible for spheres and planes. This makes triangular gridshells not so relevant in architectural geometry.

- *Quadrilateral meshes* are an interesting solution for gridshells. In a quadrilateral mesh, faces are not necessarily planar; nevertheless, on every shape, it is possible to design quadrilateral meshes with planar faces, called *PQ meshes*. Regarding the substructure, in general, only four beams are joined together at a vertex. This simplifies the manufacturing of connections. Moreover, it is also possible to approximate every reference surface with PQ meshes whose substructure is torsion-free, as will be shown in section 2.1.3. However, the design of such meshes requires a deeper geometric understanding, as will be shown in section 2.2. For this reason, these meshes have been the most studied by architectural geometry.
- *Hexagonal meshes* are interesting especially because of their nodes where only three beams are connected together. Also in this case, it is possible to design meshes with planar faces and a torsion-free substructure, but for anticlastic shapes, planar faces cannot be convex.

In the following of this thesis, the focus will be restricted to quadrilateral meshes. This choice will be justified later, in chapters 5 and 6, where we will see that a quadrilateral connectivity arises spontaneously both from mechanical and smoothness optimization of gridshells.



**Figure 2.2** – Two parallel PQ meshes  $M$  and  $M'$  (a), and the corresponding torsion-free substructure (b).

### 2.1.3 Offset meshes

Planar faces and a torsion-free substructure are desirable properties for grid-shells. While on triangular meshes this is possible only for special geometries, these goals are achievable for quad meshes approximating arbitrary shapes. This section introduces PQ meshes which have a torsion-free substructure, in particular it shows how this property is closely connected to the concepts of *parallel* and *offset meshes*.

Let  $M$  be a PQ mesh, with vertices  $\mathbf{v}_i$  and edges  $e_k = \{\mathbf{v}_i, \mathbf{v}_j\}$ . Let us consider now a mesh  $M'$ , with vertices  $\mathbf{v}'_i$  and edges  $e'_k = \{\mathbf{v}'_j, \mathbf{v}'_i\}$ , that has the same combinatorics of  $M$  (one to one correspondence among vertices, edges, and faces), and which is positioned such that corresponding edges  $e_k$  and  $e'_k$  are parallel.  $M'$  is then called a *parallel mesh* of  $M$ . A parallel mesh can be used to represent the geometry of the substructure of  $M$  letting the planar quad given by  $\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}'_j$  and  $\mathbf{v}'_i$  be the symmetry plane for the beam at the edge  $e_k$ . At each vertex  $\mathbf{v}_i$ , each symmetry plane meet along the axis passing through  $\mathbf{v}_i$  and  $\mathbf{v}'_i$ . This gives a torsion-free node. There are infinitely many parallel meshes, but to realize a substructure one wants to have node axes that are approximately orthogonal to  $M$ . This is the case when  $M$  and  $M'$  are parallel and lie approximately at constant distance  $d$ .

The distance  $d$  between two parallel meshes is not uniquely defined: it can be measured between corresponding vertices, faces or edges. We talk then respectively about *vertex*, *face*, and *edge offsets*. Let  $M$  and  $M'$  be two offset meshes at a distance  $d$ . Let us imagine to shrink the mesh  $M$  with a



uniform scaling until it collapses into a point, while accordingly keeping the mesh  $M'$  parallel at distance  $d$ . We have now three possible situations.

- *In case of vertex offset*, the vertices of  $M'$  will lie on a sphere of radius  $d$ , and therefore all faces of  $M'$  are now inscribed in a circle. A quadrilateral has a circumcircle if its opposite angles sum up to  $\pi/2$ . Since angles are preserved during the aforementioned transformation, and since for parallel meshes corresponding face angles are equal, a mesh  $M$  has a vertex offset if all its faces have a circumcircle. These special meshes, first studied by Martin et al. [32], are called *circular meshes*.
- *In case of face offset*, the faces of  $M'$  will be tangent to a sphere of radius  $d$ . This condition can be alternatively expressed at each vertex asking to its incident faces to be tangent to the same cone of revolution. Such a vertex is called *conical*. A regular vertex (with four incident faces) is conical if the sum of opposite face angles is equal. Again, this property is preserved for all parallel meshes, and then a mesh has a face offset if all its vertices are conical. These meshes, introduced by Liu et al. [31], are called *conical meshes*. Because of their face offset, conical meshes are particularly interesting for piecewise surfaces assembled from planar elements of constant thickness.
- *In case of edge offset*, the edges of  $M'$  will be tangent to a sphere of radius  $d$ . Each face has then an incircle. Such a mesh is known as a *Koebe mesh*. Because of edge offset property, Koebe meshes would be the most suitable for gridshell design, since the beams could perfectly align on top and bottom of each node. However, while circular and conical meshes are capable of approximating arbitrary shapes, this is no longer the case with Koebe meshes. Further details can be found in [45].

For the actual design of gridshells, an exact offset mesh is in general not necessary. Instead, a reasonable offset approximation is sufficient, so that one can construct a torsion-free substructure with axes almost normal to a

reference surface. In the next section, we will see how to approximate a given surface with such a mesh.

## 2.2 Meshes at the limit of refinement

Let us imagine now to refine a mesh again and again, and let us think this refinement in a way that the connectivity type of the mesh is preserved, as well as other properties such as planar or circular faces and conical vertices. At the limit of this process, the mesh will approach a continuous surface, called *limit surface*. The limit surface can be seen as the reference surface that a mesh is approximating.

Let us now imagine to refine a quadrilateral mesh. On regular quadrilateral meshes (or at least on parts of it), we can collect the edges into two groups of non intersecting polylines. At the limit of refinement, these two groups of polylines will converge to two families of curves on the limit surface. Conversely, we can think to extract a regular quadrilateral mesh by letting the edges of the mesh follow two families of curves on a reference surface. It turns out that quad meshes relevant for architecture can be extracted from families of curves owning special properties on the limit surface. In particular, Liu et al. [31] show that PQ meshes are a discretization of *conjugate networks*, while Bobenko et al. [11] and Liu et al. [31] show, respectively, that circular and conical meshes are discretizations of the so called *principal curvature network*.

In this section, after a short introduction to parametric surfaces, the properties of conjugate and principal curvature networks are described. Finally, their connection with PQ, circular, and conical meshes at the limit of refinement will be shown. For a complete treatment of differential geometry of curves and surfaces, one can refer to [18].

### 2.2.1 Parametric surfaces

Some properties of continuous surfaces, relevant for gridshell design, can be derived by their *parametric representation*. The parametric representation

of a surface  $\mathcal{M}$  is a mapping  $\mathbf{x}: \Omega \subseteq \mathbb{R}^2 \rightarrow \mathcal{M} \subset \mathbb{R}^3$ , given by

$$\mathbf{x}(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix}, \quad (u, v) \in \Omega.$$

At each point  $p \in \Omega$ , the component-wise partial derivatives

$$\mathbf{x}_{,u}(p) := \frac{\partial \mathbf{x}}{\partial u}(p) \quad \text{and} \quad \mathbf{x}_{,v}(p) := \frac{\partial \mathbf{x}}{\partial v}(p),$$

are, respectively, the tangent vectors of the two *iso-parameter curves*  $u = \text{const.}$  and  $v = \text{const.}$  on the surface at the point  $\mathbf{x}(p)$ .

The vectors  $\mathbf{x}_{,u}(p)$  and  $\mathbf{x}_{,v}(p)$  define the tangent plane of  $\mathcal{M}$  at the point  $\mathbf{x}(p)$ , denoted as  $\mathcal{T}_{\mathcal{M}}(p)$ , and can be used as a basis. A vector  $\mathbf{a} \in \mathbb{R}^3$ , tangent to  $\mathcal{M}$  at  $\mathbf{x}(p)$ , can be expressed as linear combination of the tangent basis vectors as

$$\mathbf{a} = a_u \mathbf{x}_{,u}(p) + a_v \mathbf{x}_{,v}(p). \quad (2.1)$$

The vector  $\bar{\mathbf{a}} = (a_u, a_v)^T$  is the corresponding vector in tangent coordinates. In matrix form, we can write eq. (2.1) as  $\mathbf{a} = J \bar{\mathbf{a}}$ , where  $J(p) = (\mathbf{x}_{,u}(p) \ \mathbf{x}_{,v}(p))$  is the *Jacobian matrix*. From now on, the dependency on  $p$  will be omitted.

Let  $\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2$  be two vectors at a point  $p \in \Omega$ . The scalar product between the corresponding vectors  $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^3$  is

$$\langle \mathbf{a}_1, \mathbf{a}_2 \rangle = (J \bar{\mathbf{a}}_1)^T (J \bar{\mathbf{a}}_2) = \bar{\mathbf{a}}_1^T J^T J \bar{\mathbf{a}}_2 = \bar{\mathbf{a}}_1^T \mathbb{I} \bar{\mathbf{a}}_2,$$

where the  $2 \times 2$  matrix

$$\mathbb{I} = \begin{pmatrix} \mathbf{x}_{,u}^T \mathbf{x}_{,u} & \mathbf{x}_{,u}^T \mathbf{x}_{,v} \\ \mathbf{x}_{,u}^T \mathbf{x}_{,v} & \mathbf{x}_{,v}^T \mathbf{x}_{,v} \end{pmatrix}$$

is called *first fundamental form* of  $\mathcal{M}$ . If two vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are orthogonal, we have then

$$\bar{\mathbf{a}}_1^T \mathbb{I} \bar{\mathbf{a}}_2 = 0, \quad (2.2)$$

while the length of a tangent vector  $\mathbf{a}$  is computed as

$$\|\mathbf{a}\|^2 = \bar{\mathbf{a}}^T \mathbb{I} \bar{\mathbf{a}}.$$

With the first fundamental form we can also measure the infinitesimal area element  $dA$  on  $\mathcal{M}$  at a point  $p$ , considering that the area of the parallelogram spanned by the vectors  $\mathbf{x}_{,u}$  and  $\mathbf{x}_{,v}$  is given by  $\|\mathbf{x}_{,u} \times \mathbf{x}_{,v}\|$ . For Lagrange identity we have:

$$\|\mathbf{x}_{,u} \times \mathbf{x}_{,v}\|^2 = \|\mathbf{x}_{,u}\|^2 \|\mathbf{x}_{,v}\|^2 - (\mathbf{x}_{,u}^T \mathbf{x}_{,v})^2 = \det(\mathbb{I}),$$

and then

$$dA = \sqrt{\Delta}, \quad \Delta = \det(\mathbb{I}).$$

## 2.2.2 Curvature of surfaces

The curvature of a surface, in a similar way to curves, can be derived by the variation of its normal vector. At each point  $\mathbf{x}(p)$  of  $\mathcal{M}$ , the *normal vector*  $\mathbf{n}$  is defined as:

$$\mathbf{n} = \frac{\mathbf{x}_{,u} \times \mathbf{x}_{,v}}{\|\mathbf{x}_{,u} \times \mathbf{x}_{,v}\|}.$$

Let  $\mathbf{n}_{,\bar{\mathbf{a}}}$  be the directional derivative of the vector  $\mathbf{n}$  along the vector  $\bar{\mathbf{a}}$ , given in matrix notation by  $N\bar{\mathbf{a}}$ , with  $N = (\mathbf{n}_{,u} \ \mathbf{n}_{,v})$  and where

$$\mathbf{n}_{,u} := \frac{\partial \mathbf{n}}{\partial u} \quad \text{and} \quad \mathbf{n}_{,v} := \frac{\partial \mathbf{n}}{\partial v}.$$

Given two tangent vectors  $\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2$  at a point  $p \in \Omega$ , the bilinear form

$$\langle -\mathbf{n}_{,\bar{\mathbf{a}}_1}, \bar{\mathbf{a}}_2 \rangle = (-N\bar{\mathbf{a}}_1)^T (J\bar{\mathbf{a}}_2) = -\bar{\mathbf{a}}_1^T N^T J \bar{\mathbf{a}}_2 = \bar{\mathbf{a}}_1^T \mathbb{III} \bar{\mathbf{a}}_2, \quad (2.3)$$

where

$$\mathbb{III} = \begin{pmatrix} -\mathbf{n}_{,u}^T \mathbf{x}_{,u} & -\mathbf{n}_{,u}^T \mathbf{x}_{,v} \\ -\mathbf{n}_{,v}^T \mathbf{x}_{,u} & -\mathbf{n}_{,v}^T \mathbf{x}_{,v} \end{pmatrix},$$

is called *second fundamental form* of  $\mathcal{M}$  at the point  $p$ . Observe that  $\mathbf{n}^T \mathbf{x}_{,u} = \mathbf{n}^T \mathbf{x}_{,v} = 0$ , by definition of the normal vector. We have then  $(\mathbf{n}^T \mathbf{x}_{,i})_{,j} = 0$ , with  $i, j \in (u, v)$ . This implies  $\mathbf{n}^T \mathbf{x}_{,ij} = -\mathbf{n}_{,i}^T \mathbf{x}_{,j}$ . The second fundamental

form can be then written as

$$\mathbb{III} = \begin{pmatrix} \mathbf{n}^T \mathbf{x}_{,uu} & \mathbf{n}^T \mathbf{x}_{,uv} \\ \mathbf{n}^T \mathbf{x}_{,uv} & \mathbf{n}^T \mathbf{x}_{,vv} \end{pmatrix}.$$

**Definition 1** (conjugate directions). Two directions  $\bar{\mathbf{a}}_1$  and  $\bar{\mathbf{a}}_2$  are called *conjugate* if

$$\bar{\mathbf{a}}_1^T \mathbb{III} \bar{\mathbf{a}}_2 = 0. \quad (2.4)$$

Two families of curves on  $\mathcal{M}$  are called a *conjugate network* if their tangent vectors at each point  $p$  are conjugate.

We can define now the *shape operator*  $\mathbb{S}$ , such that

$$J(\mathbb{S} \bar{\mathbf{a}}) = -\mathbf{n}_{,\bar{\mathbf{a}}}. \quad (2.5)$$

Substituting eq. (2.5) in eq. (2.3), we get

$$\mathbb{S} = \mathbb{I}^{-1} \mathbb{III}. \quad (2.6)$$

**Definition 2** (principal curvatures). The eigenvalues of  $\mathbb{S}$  at a point  $p$  are called *principal curvatures*  $\kappa_1$  and  $\kappa_2$  of  $\mathcal{M}$  at  $p$ . The corresponding eigenvectors  $\bar{\mathbf{e}}_1$  and  $\bar{\mathbf{e}}_2$  are called *principal curvature directions*. Two families of curves on  $\mathcal{M}$  are called *principal curvature network* if their tangent vectors at each point  $p$  are pointing along the principal curvature directions.

Note that the corresponding tangent vectors  $\mathbf{e}_1 = J \bar{\mathbf{e}}_1$  and  $\mathbf{e}_2 = J \bar{\mathbf{e}}_2$  are orthogonal.

### 2.2.3 Height field parametrization

Computations can be often simplified with the special parametrization

$$\mathbf{x}(x, y) = \begin{pmatrix} x \\ y \\ z(x, y) \end{pmatrix}, \quad (x, y) \in \Omega \subseteq \mathbb{R}^2, \quad (2.7)$$

referred to as *height field parametrization*, since the surface corresponds to the graph of the height function  $z(x, y)$ . It can be shown that every surface can be represented, at least locally, by such a parametrization (see [18]).

The first fundamental form is here given by

$$\mathbb{I} = \begin{pmatrix} 1 + z_{,x}^2 & z_{,x}z_{,y} \\ z_{,x}z_{,y} & 1 + z_{,y}^2 \end{pmatrix}, \quad (2.8)$$

where with comma are denoted partial derivatives, and then

$$\Delta = \det(\mathbb{I}) = 1 + z_{,x}^2 + z_{,y}^2. \quad (2.9)$$

The second fundamental form is simply given by

$$\mathbb{III} = \Delta^{-\frac{1}{2}} \begin{pmatrix} z_{,xx} & z_{,xy} \\ z_{,xy} & z_{,yy} \end{pmatrix} = \Delta^{-\frac{1}{2}} \nabla^2 z, \quad (2.10)$$

and conjugacy of directions  $\bar{\mathbf{a}}_1$  and  $\bar{\mathbf{a}}_2$  can be expressed by

$$\bar{\mathbf{a}}_1^T \nabla^2 z \bar{\mathbf{a}}_2 = 0. \quad (2.11)$$

## 2.2.4 Conjugate and principal meshes

We see now how to extract a PQ mesh  $M$  from two families of curves on a limit surface  $\mathcal{M}$ . To simplify the next computations, let us consider a local height field parametrization of  $\mathcal{M}$  at a point  $p$ , with the  $xy$  plane coinciding with  $\mathcal{T}_{\mathcal{M}}$ , and origin in  $\mathbf{x}(p)$ . Let  $\mathbf{a}_1 = J\bar{\mathbf{a}}_1$  and  $\mathbf{a}_2 = J\bar{\mathbf{a}}_2$  be the tangent vectors to the two curves passing through the point  $\mathbf{x}(p)$ . Let us extract a face of  $M$  from the quadrilateral defined by the surface points  $\mathbf{x}(p)$ ,  $\mathbf{x}(p + \bar{\mathbf{a}}_1)$ ,  $\mathbf{x}(p + \bar{\mathbf{a}}_2)$  and  $\mathbf{x}(p + \bar{\mathbf{a}}_1 + \bar{\mathbf{a}}_2)$ . In the aforementioned parametrization, the quadrilateral is planar if  $z(p + \bar{\mathbf{a}}_1 + \bar{\mathbf{a}}_2) = z(p + \bar{\mathbf{a}}_1) + z(p + \bar{\mathbf{a}}_2)$ . With a Taylor expansion, we get

$$z(p + \bar{\mathbf{a}}_1 + \bar{\mathbf{a}}_2) - z(p + \bar{\mathbf{a}}_1) - z(p + \bar{\mathbf{a}}_2) = 2\bar{\mathbf{a}}_1^T \nabla^2 z \bar{\mathbf{a}}_2 + \dots,$$

where with dots are denoted higher order terms. Planarity, at first order approximation, requires  $\bar{\mathbf{a}}_1^T \nabla^2 z \bar{\mathbf{a}}_2 = 0$ . According to eq. (2.10), this implies  $\langle \mathbf{n}_{,\bar{\mathbf{a}}_1}, \mathbf{a}_2 \rangle = 0$ , therefore  $\bar{\mathbf{a}}_1$  and  $\bar{\mathbf{a}}_2$  are conjugate directions. If we ask the directions  $\mathbf{a}_1$  and  $\mathbf{a}_2$  to be orthogonal as well, according to eq. (2.2), we have  $\bar{\mathbf{a}}_1 \mathbb{I} \bar{\mathbf{a}}_2 = 0$ . Together with conjugacy, this implies that  $\bar{\mathbf{a}}_1$  and  $\bar{\mathbf{a}}_2$  are pointing along the eigenvectors of  $\mathbb{I}^{-1} \mathbb{III}$ . Equation (2.6) tells us that  $\bar{\mathbf{a}}_1$  and  $\bar{\mathbf{a}}_2$  are principal curvature directions, and that principal curvature directions are conjugate. We can then state the following

**Proposition 1.** *At the limit of refinement, a quad mesh with infinitesimal planar faces converges to a network of conjugate curves on its limit surface. The only orthogonal and conjugate network is the principal curvature network.*

Principal curvature directions, besides spanning infinitesimal planar quads, possess another interesting property relevant for architecture. Along principal curvature directions  $\mathbf{e}_i$ , the change of the normal vector is given by  $\mathbf{n}_{,\mathbf{e}_i} = -k_i \mathbf{e}_i$ . Therefore, when moving infinitesimally along the direction  $\mathbf{e}_i$  on the surface, the normal vector span an infinitesimal planar quad. This property is exactly what we want for the symmetry planes of the substructure. We can then state

**Proposition 2.** *At the limit of refinement, a principal mesh has a torsion-free substructure orthogonal to the limit surface.*

From proposition 2, we can see that circular and conical meshes are both discretizations of a network of principal curvature, depending on whether we refer to face or vertex normals. For circular meshes, we can consider the normals at each face given by the axes of the face circumcircles. Across each edge, the neighboring face normals intersect in a common point; therefore, these lie on a common plane [53]. For conical meshes, we can consider the normals at each vertex, given by the axes of the tangent cones. The axes of the cones at the two vertices of each edge intersect again in a point, as shown in [31].

We can then say that principal meshes are a convenient geometry layout for gridshells, since we can achieve planarity of cladding panels and a torsion-free substructure. Moreover, from principal networks one can easily extract

conical and circular meshes, in case one needs a face or vertex offset for further construction requirements.



## Chapter 3

# The geometry of equilibrium

As seen in chapter 2, the geometry of a gridshell can be represented by a mesh. From a mechanical point of view, the edges of this mesh are charged to bear the loads within the structure. It is well known that the most efficient manner of bearing loads in a framework is through strictly axial forces. Our focus will be then restricted to frameworks in axial force equilibrium.

In this thesis, the static behavior of gridshells plays the role of a form finding criterion. If gravitational loads are dominant, it is then reasonable to neglect wind, seismic acceleration, and other lateral loads, leaving these effects to a subsequent structural verification. The restriction to vertical loads allows us to split the equilibrium in a vertical and a horizontal component. It will be shown that the horizontal equilibrium of a gridshell implies the existence of a mesh with planar faces, corresponding in vertical projection to the gridshell, called *force polyhedron*. At the limit of refinement, through the process described in section 2.2, the gridshell converges to a *membrane*, that is a surface-like continuum with no bending stiffness. Accordingly, the force polyhedron converges to a continuous surface, called *Airy stress surface*, representing the horizontal equilibrium of the membrane.

This chapter introduces the computational setting of discrete and continuous equilibrium, namely gridshells (section 3.1) and membranes (section 3.2).

## 3.1 Equilibrium of gridshells

This section introduces the equilibrium of gridshells under vertical loads. The horizontal equilibrium of these structures can be described through the existence of a force polyhedron, introduced by Maxwell [33].

### 3.1.1 The mechanical model

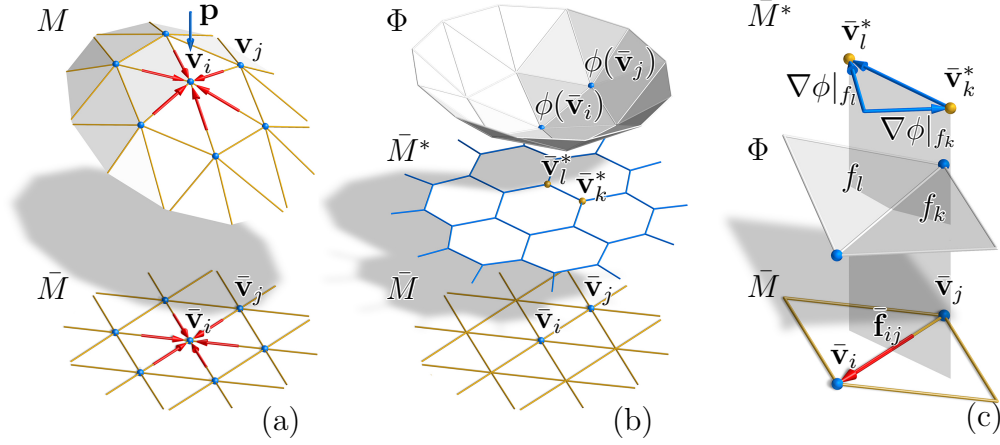
This thesis tackles gridshells in axial force equilibrium under vertical loads only. Self-weight, dead, and static live loads are lumped in forces and applied in the vertices. The resulting framework structure is a *gridshell truss*: a system of straight beams, with axes corresponding to the edges of a mesh, connected together and to the supports with frictionless pin-joints. This kind of structure, depending on its geometry, connectivity, and support conditions, might be a mechanism in equilibrium. However, even if in an actual gridshell the nodes are manufactured as rigid joints for stability and safety reasons, the use of a truss model in the design stage is strongly beneficial for minimizing bending effects.

### 3.1.2 Nodal equilibrium

Let  $M$  be a three dimensional gridshell truss, with members corresponding to the edges of a mesh. Loads are applied in the vertices  $\mathbf{v}_i = (x_i, y_i, z_i)$ , and support conditions are given along the boundary. The force  $\mathbf{f}_{ij}$  exerted by the oriented bar  $\mathbf{e}_{ij} = \mathbf{v}_i - \mathbf{v}_j$  on the vertex  $\mathbf{v}_i$  can be expressed as  $w_{ij}(\mathbf{v}_i - \mathbf{v}_j)$ , where  $w_{ij}$  is the axial force per unit bar length or *force density*, and where positive values of  $w_{ij}$  indicate compression. If the system is in equilibrium, at each unsupported vertex  $\mathbf{v}_i$  we have

$$\sum_{j \sim i} w_{ij}(\mathbf{v}_i - \mathbf{v}_j) + \mathbf{p}_i = 0, \quad (3.1)$$

where with  $j \sim i$  we denote all the vertices  $j$  connected with the vertex  $i$ , and where  $\mathbf{p}_i = (p_x^i, p_y^i, p_z^i)$  is the load applied at the node  $\mathbf{v}_i$ . Since we consider only vertical loads, we can split the horizontal and vertical equilibrium



**Figure 3.1** – Gridshell equilibrium. (a) A portion of a gridshell  $M$  under vertical load and its projection into the  $xy$  plane  $\bar{M}$ . (b) The horizontal equilibrium yields a force dual mesh  $\bar{M}^*$  with edges given by the forces acting in the correspondent primal edges of  $\bar{M}$  rotated by 90 degrees, as shown by Maxwell [33]. We can construct an Airy polyhedron  $\Phi$  with face gradients given by the coordinates of the corresponding dual vertex of  $\bar{M}^*$ . (c) By construction, the magnitudes of the forces in the bars  $\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j$  are given by the isotropic angles between the adjacent faces on  $\Phi$ . The isotropic angle can be seen as the change in slope between two faces of  $\Phi$  when traversed orthogonally to  $\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j$ .

respectively as

$$\sum_{j \sim i} w_{ij}(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j) = \mathbf{0}, \quad \sum_{j \sim i} w_{ij}(z_i - z_j) = p_z^i, \quad (3.2)$$

where  $\bar{\mathbf{v}}_i, \bar{\mathbf{v}}_j$  are the  $xy$  projections of the points  $\mathbf{v}_i, \mathbf{v}_j$ .

### 3.1.3 The force polyhedron

Let us now consider the projection of the structure in the  $xy$  plane, denoted as  $\bar{M}$ . Let  $\bar{\mathbf{f}}_{ij}$  be the  $xy$  projections of the forces  $\mathbf{f}_{ij}$ . Since  $\bar{M}$  is a 2D system in horizontal equilibrium under boundary loads (given by the  $xy$  projections of the support reactions), the forces  $\bar{\mathbf{f}}_{ij}$  acting on each vertex  $\bar{\mathbf{v}}_i$  can be arranged in a planar closed cycle. We can build thus a reciprocal diagram  $\bar{M}^*$ , combinatorially dual to  $\bar{M}$ , whose edges are given by the forces acting in the corresponding primal edge. For convenience, we represent this dual diagram rotated by  $90^\circ$  clockwise in the  $xy$  plane, as shown in fig. 3.1b.

We can now construct a *force polyhedron*  $\Phi = (x, y, \phi(x, y))$ , whose edges and vertices coincide in the  $xy$  projection to the primal truss  $\bar{M}$ , in the following way. Let us denote as  $f_k$  the faces of  $\Phi$ , and let  $\bar{\mathbf{v}}_k^* = (x_k^*, y_k^*)$  be the corresponding dual vertices of  $\bar{M}$ . Hence, each face  $f_k$  of  $\Phi$  lies on a plane with gradient  $\nabla\phi|_{f_k} = (x_k^*, y_k^*)$ . The closure of each face of  $\bar{M}^*$  ensures the closure of the polyhedron  $\Phi$  when turning around the corresponding primal vertex. This construction is uniquely defined up to vertical translations and shearing. For further details see [20] and [56].

Let  $f_k, f_l$  be the faces of  $\Phi$  meeting at the oriented edge with projection  $\bar{\mathbf{e}}_{ij}$ , as shown in fig. 3.1c. The force  $\bar{\mathbf{f}}_{ij}$ , by construction, is given by  $\bar{R}(\nabla\phi|_{f_l} - \nabla\phi|_{f_k})$ , where  $\bar{R} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$  is the  $90^\circ$  counterclockwise rotation matrix in the  $xy$  plane. Denoting the  $xy$  unit edge vector as  $\hat{\mathbf{e}}_{ij} = (\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j)/\|\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j\|$ , the quantity

$$\beta^{is}(\bar{\mathbf{e}}_{ij}) = \bar{R}(\nabla\phi|_{f_l} - \nabla\phi|_{f_k}) \cdot \hat{\mathbf{e}}_{ij} \quad (3.3)$$

is the *signed isotropic angle* between the faces  $f_l$  and  $f_k$ . Positive values of  $\beta^{is}(\bar{\mathbf{e}}_{ij})$  indicate compression in the bar  $\bar{\mathbf{e}}_{ij}$ . Note that  $\beta^{is}(\bar{\mathbf{e}}_{ij}) = \beta^{is}(\bar{\mathbf{e}}_{ji})$ .

## 3.2 Equilibrium at the limit of refinement

Let us consider now a refinement process that increases the density of a gridshell truss. From a mechanical point of view, at the limit of refinement the gridshell will tend to a *membrane*: a surface-like continuum that cannot support out of plane bending, and with mechanical properties derived at each point from the thickness in the normal direction. At the same time, the force polyhedron representing the horizontal equilibrium of the structure will tend to a continuous surface representing the stresses of a 2D body, called *Airy stress surface*. After an introduction to the stress tensor of a continuous body, this section describes the equilibrium of membranes and their Airy stress surface. For a detailed description of gridshells approaching membranes, see [36].

### 3.2.1 The stress tensor

When dealing with continuous bodies  $\mathcal{D}$ , the stress state at a point can be described by the *Cauchy stress tensor*  $\boldsymbol{\sigma}$ . For 3D bodies, this is a symmetric  $3 \times 3$  tensor with the following meaning. Given an infinitesimal surface at a point of  $\mathcal{D}$ , with normal vector  $\mathbf{a}$ , the stress vector  $\mathbf{f}$  (force per unit area) acting across the surface is given by

$$\mathbf{f} = \frac{\boldsymbol{\sigma} \mathbf{a}}{\|\mathbf{a}\|}. \quad (3.4)$$

The component of the stress vector  $\mathbf{f}$  along the direction  $\mathbf{a}$  gives the *normal stress* acting on the infinitesimal surface, and it is computed as

$$\sigma_n(\mathbf{a}) = \frac{\langle \mathbf{f}, \mathbf{a} \rangle}{\|\mathbf{a}\|} = \frac{\mathbf{a}^T \boldsymbol{\sigma} \mathbf{a}}{\mathbf{a}^T \mathbf{a}}. \quad (3.5)$$

**Definition 3** (principal stresses). At a point of a body  $\mathcal{D}$ , the extrema of  $\sigma_n(\mathbf{a})$  are given by the eigenvalues of  $\boldsymbol{\sigma}$ . These are called *principal stresses*  $\sigma_1$ ,  $\sigma_2$ , and  $\sigma_3$ . The corresponding eigenvectors, mutually orthogonal, gives the *principal stress directions*.

Let  $\mathbf{p}$  be the volume loads acting on  $\mathcal{D}$  (force per unit volume). The body equilibrium asks at each interior point

$$\operatorname{div}(\boldsymbol{\sigma}) + \mathbf{p} = 0. \quad (3.6)$$

On the boundary  $\partial\mathcal{D}$ , the stress vector must balance the applied boundary tractions  $\mathbf{b}$  (force per unit surface area), we have then

$$\boldsymbol{\sigma} \mathbf{n} + \mathbf{b} = 0, \quad (3.7)$$

where  $\mathbf{n}$  is the outer normal of the surface  $\partial\mathcal{D}$ .

### 3.2.2 Membrane stress

A membrane is a surface-like body, described by a surface  $\mathcal{M}$ . A membrane is characterized by supporting stresses only in its tangent plane, and these

are described by the  $2 \times 2$  *membrane stress tensor*  $S \in \mathcal{T}_{\mathcal{M}}$ . The tensor  $S$  represents the stresses integrated over the membrane normal thickness, and is symmetric for balance of angular momentum. The membrane stress vector  $\mathbf{f}$  (force per unit length) acting across an infinitesimal line orthogonal to a tangent direction  $\mathbf{a}$  is computed, in tangent coordinates, as

$$\bar{\mathbf{f}} = \frac{S \mathbb{I} \bar{\mathbf{a}}}{\|\mathbf{a}\|},$$

where  $\mathbb{I}$  is the first fundamental form of the surface defined in section 2.2.1. The normal stress is now given by

$$\sigma_n(\mathbf{a}) = \frac{\langle \mathbf{f}, \mathbf{a} \rangle}{\|\mathbf{a}\|} = \frac{(S \mathbb{I} \bar{\mathbf{a}})^T \mathbb{I} \bar{\mathbf{a}}}{\bar{\mathbf{a}}^T \mathbb{I} \bar{\mathbf{a}}} = \frac{\bar{\mathbf{a}}^T \mathbb{I} S \mathbb{I} \bar{\mathbf{a}}}{\bar{\mathbf{a}}^T \mathbb{I} \bar{\mathbf{a}}}. \quad (3.8)$$

Note that the matrices  $S$  and  $\mathbb{I}$  are symmetric. The extrema of  $\sigma_n(\mathbf{a})$  are the *membrane principal stresses* and are given by the eigenvalues of  $S \mathbb{I}$ . If  $\bar{\mathbf{s}}_1$  and  $\bar{\mathbf{s}}_2$  are the corresponding eigenvectors, eq. (3.8) implies also  $\bar{\mathbf{s}}_1^T \mathbb{I} \bar{\mathbf{s}}_2 = 0$ , therefore the corresponding principal stress directions are orthogonal on the surface.

### 3.2.3 Equilibrium under vertical load

Away from points with a vertical tangent plane, we can parametrize the surface  $\mathcal{M}$  locally as a height function  $\mathcal{M}(x, y) = (x, y, z(x, y))$ . Let  $\bar{S}$  be the tensor representing the  $xy$  projection of the membrane stresses given by

$$\bar{S} = \sqrt{\Delta} S, \quad (3.9)$$

where  $\Delta = \det(\mathbb{I})$ . If we consider only vertical loads, it is convenient to express equilibrium in the global coordinate system  $(x, y, z)$ , with a vertical  $z$  axis. The horizontal and vertical equilibrium, respectively, are then expressed by

$$\operatorname{div}(\bar{S}) = \mathbf{0}, \quad \operatorname{div}(\bar{S} \nabla z) = \bar{\rho}. \quad (3.10)$$

Here divergence of a matrix is applied to its columns, and  $\bar{\rho}(x, y)$  is the vertical load per unit  $xy$  area. Further details can be found in [4, 56].

On boundaries, a membrane can support only tractions lying in its tangent plane. Along supported boundaries, we can assume no conditions. Along unsupported boundaries, equilibrium asks

$$S \mathbb{I} \bar{\mathbf{b}} = \mathbf{0}, \quad (3.11)$$

where  $\bar{\mathbf{b}}$  is the surface boundary normal in tangent coordinates.

### 3.2.4 The Airy stress surface

The horizontal equilibrium of membranes under vertical loads implies the existence of a divergence-free tensor  $\bar{S}$ . Every divergence-free tensor can be given by the adjoint Hessian of a continuous function  $\phi(x, y)$ , because of commutativity of third order derivatives with respect to  $x$  and  $y$ . The stress tensor  $\bar{S}$  is then

$$\bar{S} = \tilde{\nabla}^2 \phi = \begin{pmatrix} \phi_{,yy} & -\phi_{,xy} \\ -\phi_{,xy} & \phi_{,xx} \end{pmatrix}, \quad (3.12)$$

where with comma are denoted partial derivatives, and where with over-tilde is denoted the adjoint matrix operation. The function  $\phi(x, y)$  is called *Airy stress function*. The vertical equilibrium component of eq. (3.10) expands now to

$$z_{,xx} \phi_{,yy} - 2 z_{,xy} \phi_{,xy} + z_{,yy} \phi_{,xx} = \bar{\rho}. \quad (3.13)$$

For more details, see [48].

Considering eqs. (3.9) and (3.12), and recalling that the adjoint matrix operation, by definition, asks  $\tilde{S} \mathbb{I} = \tilde{\mathbb{I}} \tilde{S}$  and  $\tilde{\mathbb{I}} = \Delta \mathbb{I}^{-1}$ , we find

$$\tilde{S} \mathbb{I} = \sqrt{\Delta} \mathbb{I}^{-1} \nabla^2 \phi. \quad (3.14)$$

Since the eigenvalues of  $S \mathbb{I}$  are pointing along the principal stress directions of the membrane, and since adjoint matrices have swapped principal directions, the principal stresses are the eigenvalues of  $\sqrt{\Delta} \mathbb{I}^{-1} \nabla^2 \phi$ , and the corresponding eigenvectors are the swapped principal stress directions.

We can consider the Airy stress potential as a surface in *isotropic space*. Isotropic geometry is a linearization of Euclidean geometry with a distin-

guished vertical  $z$  axis and a horizontal  $xy$  plane. Surfaces are described as graphs  $z = \phi(x, y)$ , and the role of the second fundamental form is played by the Hessian of  $\phi$ . Its eigenvalues are  $i$ -principal curvatures  $\kappa_1^i$  and  $\kappa_2^i$ , and its eigenvectors define the  $i$ -principal directions and the network of  $i$ -principal curves. This network of curves, corresponding to the principal stress lines, is  $i$ -orthogonal, meaning that its projection onto the  $xy$  plane is an orthogonal network of curves. Like the classical Euclidean principal curves, the  $i$ -curves form a conjugate network. For an introduction to isotropic geometry, see [44].



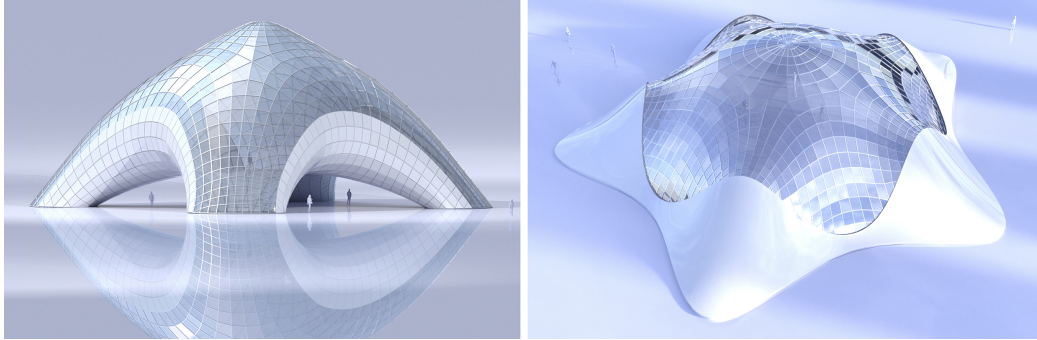
# Chapter 4

## Principal meshes in equilibrium

Designing freeform gridshells with due regard to economic and feasibility factors is a challenging task. As seen in chapter 2, rationalizing such surfaces by means of principal meshes is beneficial for manufacturing reasons, such as planar cladding panels and torsion-free substructure connections. On the other hand, as seen in chapter 3, it is convenient to ensure static equilibrium in the load bearing structure through axial forces only. It turns out that both of these goals can be achieved only for meshes that discretize surfaces in membrane equilibrium where principal stress and curvature directions coincide. In this chapter, a method for the optimization of a given shape towards stress and curvature alignment is presented, within a workflow for the design of principal meshes in equilibrium.

### 4.1 Overview and contribution

To generate a principal mesh in equilibrium, one can start generating a principal mesh that approximates a given shape; then, optimize this mesh for equilibrium, planarity of faces and, eventually, for circularity or conical vertices. For that purpose, existing tools such as [49, 52, 55] can be used. However, due to an intrinsic conflict, this optimization often fails. It turns out, indeed, that principal meshes in axial equilibrium can approximate only special surfaces in membrane equilibrium, where stress and curvature directions coincide.



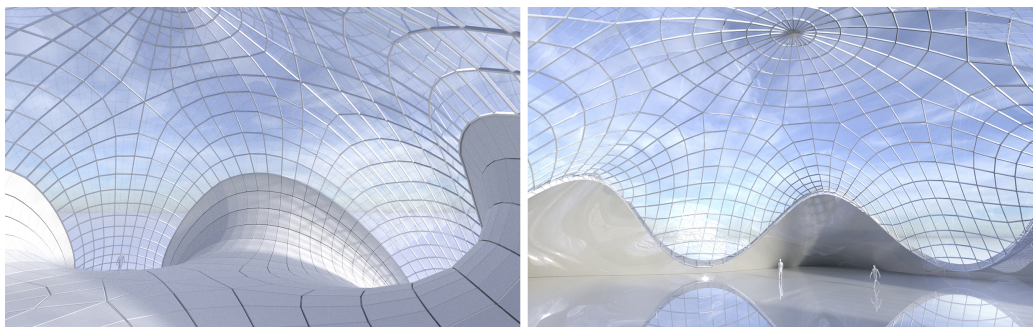
**Figure 4.1** – Architectural surfaces discretized with principal meshes in equilibrium, achieved thanks to stress and curvature alignment. Cladding can be realized with flat panels, and the substructure with prismatic beams and torsion-free nodes. At the same time, structural bending effects are minimized. The mesh on the left discretizes a non-height field shape.

In this chapter, the design of principal meshes in equilibrium is addressed. Thanks to the alignment of principal curvature and stress directions on triangulated surfaces and a subsequent principal remeshing, the shape and the connectivity of the mesh are both part of the solution. We overcome in this way the main limitation of existing methods, namely working with given connectivities.

In section 4.3, it is first shown that principal meshes in equilibrium are discretizations of membrane surfaces with coincident stress and curvature directions. A method for the estimation of stress directions on a triangulated surface is then described. This method is based on the normal cycle approach of Cohen-Steiner and Morvan [15], applied in this case to the  $i$ -principal curvatures of the Airy stress surface. A design workflow is then outlined. In section 4.4, a computational method for the optimization of a given shape towards stress and curvature alignment is presented. Results are discussed in section 4.5. This work has been published in [41].

## 4.2 Previous work

A significant step in the optimization of meshes for equilibrium comes from *thrust network analysis*, introduced by Block and Ochsendorf [10]. Vouga et al. [56] provide a differential geometric understanding of this approach



**Figure 4.2** – Architectural applications. Interior views of the meshes shown in Figures 4.5a (on the left) and 4.3 (on the right). Exterior views are shown in fig. 4.1.

and use it for the design of planar quad meshes in equilibrium. An efficient optimization of quad meshes for equilibrium and face planarity is provided by Tang et al. [55], but the success of this method is strongly dependent on the initial mesh connectivity. Schiffner and Balzer [49] propose a method for planar quad-remeshing of given surfaces, initialized by principal stress lines. However, the effectiveness of this method is limited, since for a general surface, planarity of quads and the alignment with principal stress directions are often conflicting goals. A first attempt to directly design principal meshes in equilibrium was made by Sun [52], fixing the mesh combinatorics in advance. Unfortunately, this approach rarely yields good convergence of optimization.

### 4.3 Aligning principal stress and curvature directions

This section introduces principal meshes in force equilibrium. In section 4.3.1, using the results of chapters 2 and 3, we show that at the limit of refinement a principal mesh in axial force equilibrium must be aligned with the principal stress directions of a surface in membrane equilibrium. In section 4.3.2, it is shown how it is possible to model a membrane with a triangular mesh, enforcing the axial force equilibrium on its edges. It is then shown how to estimate on this mesh equivalent stress and curvature directions. Finally, in section 4.3.3, a workflow for the design of principal meshes in equilibrium is

outlined.

### 4.3.1 Principal meshes at the limit of refinement

Principal meshes are a discretization of the principal curvature network of a continuous surface. A principal mesh in equilibrium, from a mechanical point of view, is a gridshell truss with a quad combinatorics. At the limit of refinement, this gridshell will tend to a principal network of curves on a continuous surface in membrane equilibrium.

Let  $\mathcal{M}(x, y)$  be this membrane under vertical load, parametrized as a height field surface over the  $xy$  plane, as described in section 3.2. Let us then consider the principal network of curves of  $\mathcal{M}(x, y)$ , defined at each point by two tangent vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$ , and let  $\bar{\mathbf{a}}_1$  and  $\bar{\mathbf{a}}_2$  be their  $xy$  projections. We are now looking for simple conditions which express that the principal network is in equilibrium.

First, for principal curve networks, the vectors  $\mathbf{a}_1$ ,  $\mathbf{a}_2$  follow principal curvature directions. These directions are orthogonal on the surface. With  $\mathbb{I}$  as first fundamental form of  $\mathcal{M}(x, y)$ , according to eq. (2.2), we can express the orthogonality condition of  $\mathbf{a}_1$ ,  $\mathbf{a}_2$  as

$$\bar{\mathbf{a}}_1^T \mathbb{I} \bar{\mathbf{a}}_2 = 0. \quad (4.1)$$

Secondly, as seen in section 3.1, if a gridshell is in equilibrium under vertical loads, its  $xy$  projection must admit a force polyhedron  $\Phi$  with planar faces. At the limit of refinement, the polyhedron  $\Phi$  will tend to a continuous Airy surface  $z = \phi(x, y)$ . For a quadrilateral gridshell, the corresponding force polyhedron is a quad mesh with planar faces. As stated in proposition 1, a planar quad mesh at the limit of refinement will converge to a network of conjugate curves on a surface. We can then state the following condition: *a quad network on a surface is in horizontal equilibrium under vertical load if it is vertically projected onto a conjugate curve network on the corresponding Airy stress surface.* According to eq. (2.11), the condition for the directions  $\mathbf{a}_1$ ,  $\mathbf{a}_2$  to be vertically projected onto conjugate directions of  $\phi(x, y)$  is expressed by

$$\bar{\mathbf{a}}_1^T \nabla^2 \phi \bar{\mathbf{a}}_2 = 0. \quad (4.2)$$

Equations (4.1) and (4.2) imply that the vectors  $\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2$  are eigenvectors of  $\mathbb{I}^{-1}\nabla^2\phi$ . Since the principal stress directions on  $\mathcal{M}(x, y)$  are given by the eigenvectors of  $\sqrt{\Delta}\mathbb{I}^{-1}\nabla^2\phi$ , as shown in eq. (3.14), we can see that the only directions in horizontal equilibrium and orthogonal on the membrane are the principal stress directions. We can then state the following important fact:

**Proposition 3.** *Principal meshes in equilibrium under vertical loads are discrete representations of membrane surfaces where principal stress and principal curvature directions agree. There, they follow these principal directions.*

### 4.3.2 Estimating stress and curvature

As seen in section 3.2, at each point of a membrane we find three unknown stress components and three equilibrium equations. Membranes are then statically determinate in the sense that, given the loads and the boundary tractions, the stress tensor is uniquely determined; the existence of a solution depends only on the membrane geometry. Let us now consider a triangular gridshell forming a closed polyhedron  $\Gamma$  of genus zero, and with loads applied in its nodes. Denoting by  $|V|$  its number of vertices and  $|E|$  its number of edges, Euler's formula shows that  $3|V| = |E| + 6$ . Since we have one unknown axial force per edge and three equilibrium equations per vertex, the solution is uniquely determined up to rigid body motion; the existence of the solution depends on the geometry of the polyhedron. The same is true for a portion of  $\Gamma$ , given the force reactions of the remaining part. Triangular gridshells can therefore reproduce the statical determinacy of membranes, see [39]. In the following, the membrane behavior of a surface is expressed through the equilibrium of a gridshell triangulation.

In a continuous membrane, the projected stress tensor  $\bar{S}$  and the isotropic shape operator  $\nabla^2\phi$  are related by  $\bar{S} = \tilde{\nabla}^2\phi$ . We are now searching for a discrete analog of the isotropic shape operator defined for triangle meshes, and at first look at the Euclidean counterpart. For that, we use the normal cycle approach by [15]. One computes an extended shape operator  $W$  ( $3 \times 3$  matrix with two eigenvectors in principal curvature direction and the third eigenvector, with eigenvalue close to zero, orthogonal to the surface) as follows. Selecting a vertex  $\mathbf{v}_i$  and a surrounding region  $R_i$  of area  $A_i$ ,  $W$  is

found by

$$W(\mathbf{v}_i) = \frac{1}{A_i} \sum_{j \sim i} \beta(\mathbf{e}_{ij}) \|\mathbf{e}_{ij} \cap R_i\| \hat{\mathbf{e}}_{ij} \hat{\mathbf{e}}_{ij}^T. \quad (4.3)$$

Here  $\beta(\mathbf{e}_{ij})$  is the signed Euclidean angle between the two normals of the faces adjacent to the edge  $\mathbf{e}_{ij}$ ,  $\mathbf{e}_{ij} \cap R_i$  is the portion of the edge  $\mathbf{e}_{ij}$  intersecting the region  $R_i$ , and  $\hat{\mathbf{e}}_{ij}$  is the unit edge vector, given by  $(\mathbf{v}_i - \mathbf{v}_j) / \|\mathbf{v}_i - \mathbf{v}_j\|$ . The eigenvalues of  $W(\mathbf{v}_i)$ , associated with the two eigenvectors lying in the tangent plane at  $\mathbf{v}_i$ , will give an estimation of principal curvatures along the swapped tangent eigenvectors. To obtain a discrete isotropic shape operator, we have to replace Euclidean quantities by isotropic ones. This means that lengths and areas are measured in the  $xy$  plane and the Euclidean angle  $\beta(\mathbf{e}_{ij})$  is replaced by the signed isotropic angle  $\beta^{is}(\bar{\mathbf{e}}_{ij})$ , given by eq. (3.3). Setting  $\bar{\mathbf{e}}_{ij} \cap \bar{R}_i = \bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j$ , we can estimate the  $2 \times 2$  adjoint Hessian of  $\Phi$  at  $\bar{\mathbf{v}}_i$  as

$$\tilde{\nabla}^2 \phi(\bar{\mathbf{v}}_i) = \bar{S}(\bar{\mathbf{v}}_i) = \frac{1}{A_i} \sum_{j \sim i} J(\nabla \phi|_{f_l} - \nabla \phi|_{f_k})(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j)^T.$$

Observing that  $J(\nabla \phi|_{f_l} - \nabla \phi|_{f_k}) = \bar{\mathbf{f}}_{ij} = w_{ij}(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j)$ , we can estimate the stress tensor directly through force densities as

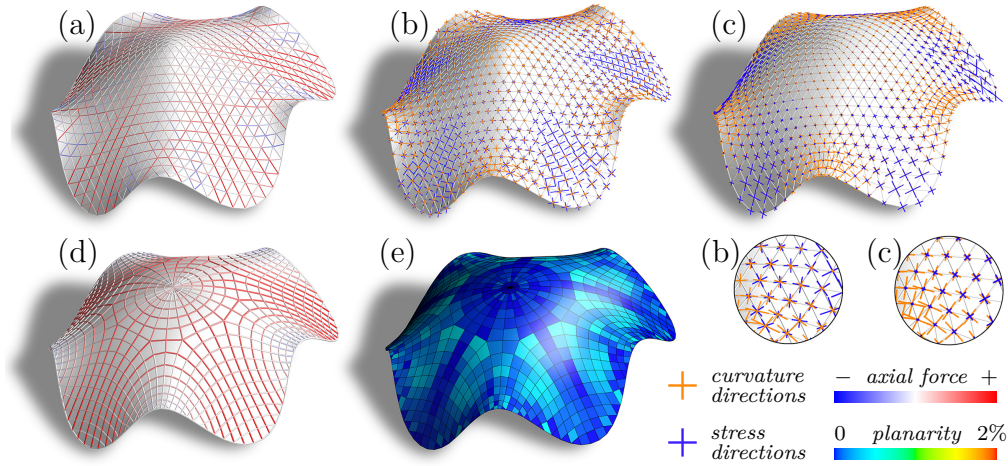
$$\bar{S}(\bar{\mathbf{v}}_i) = \frac{1}{A_i} \sum_{j \sim i} w_{ij}(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j)(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j)^T. \quad (4.4)$$

To estimate the principal curvature directions on the triangulated surface, it is possible to use again Cohen-Steiner eq. (4.3). For sufficiently smooth meshes, we can make the approximation  $\beta \approx \sin \beta$ . With  $\mathbf{n}_{f_k}$  and  $\mathbf{n}_{f_l}$  as the unit normals of the left and right faces of the edge  $\mathbf{e}_{ij}$ , we can then estimate the  $3 \times 3$  extended shape operator as

$$W(\mathbf{v}_i) = \frac{1}{A_i} \sum_{j \sim i} (\mathbf{n}_{f_l} \times \mathbf{n}_{f_k})(\mathbf{v}_i - \mathbf{v}_j)^T. \quad (4.5)$$

Let  $\kappa_1$  and  $\kappa_2$  be the eigenvalues of  $W$  corresponding to the two eigenvectors in the tangent plane of  $M$ . It is possible to ensure the alignment of two vectors  $\mathbf{a}_1, \mathbf{a}_2$  with principal directions at each vertex  $\mathbf{v}_i$  by requiring

$$W\mathbf{a}_1 = \kappa_1\mathbf{a}_1, \quad W\mathbf{a}_2 = \kappa_2\mathbf{a}_2.$$



**Figure 4.3** – Design workflow. (a) An initial shape is given as triangular mesh and the equilibrium is enforced on the edges. (b) The estimated curvature and stress directions, in general, are not aligned. (c) After our optimization, we reach the alignment with a change in the shape. (d) We remesh the resulting shape with mixed integer quadrangulation along the computed directions. After a post-optimization, the structure is in equilibrium under axial forces, (e) and panels are close to planar. According to a finite element analysis, the ratio of internal elastic work  $w_a$  due to axial forces in the final structure is 0.95. The stress and curvature directions are scaled according to their anisotropy, given by the difference between the two eigenvalues. A possible application of this design is depicted in figs. 4.1 and 4.2 (on the right).

### 4.3.3 Design workflow

We have now the elements to design principal meshes in equilibrium. In particular, we solve the following problem: *given an initial surface subject to gravitational load and its support conditions, find a quadrilateral mesh in force equilibrium with edges aligned along principal curvature directions that is close to the initial design surface.* The procedure can be summarized in the following steps:

- *Step 1.* Given an input surface as a triangular mesh and the support conditions, the mesh geometry is optimized in order to align the equivalent stress and curvature directions as described in section 4.3.2, while keeping the vertices as close as possible to the input shape.
- *Step 2.* The resulting directions are used as guide for a quadrilateral

remeshing of the optimized mesh. For this purpose, we use *mixed integer quadrangulation* proposed by Bommès et al. [13]. In this step, the density of the mesh can be chosen according to fabrication and design considerations.

- *Step 3.* The obtained quadrilateral mesh is subject to post-optimization for equilibrium and planarity of faces, while applying some fairness to the network curves to guarantee aesthetic quality. For this purpose, we use the method of [55]. Thanks to *step 1*, we can expect convergence with minimized conflict between planarity and equilibrium.

## 4.4 Implementation

In this section we briefly describe the implementation of *step 1*, described in the workflow section 4.3.3. Starting from a given triangular mesh  $M^0$  with specified support conditions, we find a mesh  $M$  where principal stresses and principal curvature directions are aligned, as close as possible to  $M^0$ .

### 4.4.1 Main variables and constraints

For a mesh  $M^0 = (V, E, F)$ , being  $s$  and  $c$  respectively the number of vertices that are mechanically supported, and fixed during the optimization, the main variables of the problem are:

- the position of the vertices  $\mathbf{v}_i$  of  $M$  ( $3(|V| - c)$  variables)
- the force densities  $w_{ij} = w_{ji}$  ( $|E|$  variables)
- the components of the stress tensor  $\bar{S}_{11}$ ,  $\bar{S}_{22}$  and  $\bar{S}_{12}$  ( $3|V|$  variables)
- the components of the extended shape operator  $W_{11}$ ,  $W_{22}$ ,  $W_{33}$ ,  $W_{12}$ ,  $W_{23}$  and  $W_{13}$  ( $6|V|$  variables)
- the tangent eigenvalues  $\lambda_1$  and  $\lambda_2$  of the extended shape operator ( $2|V|$  variables)
- the directions  $\mathbf{a}_1$  and  $\mathbf{a}_2$  at  $\mathbf{v}_i$  ( $6|V|$  variables).

The main constraints are:



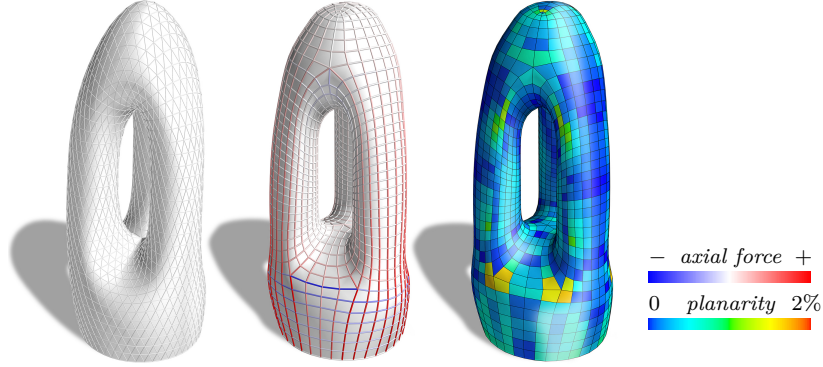
- the equilibrium at unsupported vertices  $\mathbf{v}_i$  through eq. (3.2) ( $3(|V| - s)$  equations)
- the connection of the stress tensor components with force densities through eq. (4.4) ( $3V$  equations). Since we are interested only in principal directions, we can omit  $\bar{A}_i$  from the equations.
- the connection of the curvature components with face normals through eq. (4.5) ( $6|V|$  equations). As for the previous point, we omit  $A_i$ .
- the normalization of directions:  $\mathbf{a}_1^T \mathbf{a}_1 = 1$  and  $\mathbf{a}_2^T \mathbf{a}_2 = 1$  ( $2|V|$  equations)
- the tangency of directions (tangency is guaranteed together with principal direction alignment, see below):  $(\mathbf{a}_1 + \mathbf{a}_2)^T \mathbf{n}_i = 0$  ( $|V|$  equations)

where  $\mathbf{n}_i$  is the vertex normal at  $\mathbf{v}_i$ . The target functions are given by the alignment equations of the vectors  $\mathbf{a}_1, \mathbf{a}_2$  with stress and curvature directions, as seen in sections 4.3.1 and 4.3.2. We have then:

- conjugacy on the Airy surface:  $\bar{\mathbf{a}}_1^T \tilde{\bar{S}} \bar{\mathbf{a}}_2 = 0$  ( $|V|$  equations)
- principal direction alignment:  $W\mathbf{a}_1 = \lambda_1 \mathbf{a}_1$  and  $W\mathbf{a}_2 = \lambda_2 \mathbf{a}_2$  ( $6|V|$  equations).

For proximity to the starting surface, we minimize the distance between the points  $\mathbf{v}_i$  and the tangent plane of their closest vertex  $\mathbf{v}_j^0$  of  $M^0$ . We point out here that the projected stress tensor  $\bar{S}$  is not properly defined for surface points with a vertical tangent plane. To avoid noise in the solution, we remove the target functions of Airy conjugacy on vertices  $\mathbf{v}_i$  where the  $z$  coordinate of the normal  $\mathbf{n}_i$  is in the range  $\pm 10^{-2}$ .

Subtracting the number of constraints from the number of variables, and keeping fixed during the optimization the supported vertices (then  $s = c$ ), we find  $5|V| + |E|$  degrees of freedom. The target functions of alignment yield  $7V$  equations. Considering that on a triangle mesh we have  $|E| \approx 3|V|$ , we are left with approximately  $|V|$  degrees of freedom. This allows us to ask for closeness to the reference shape as a soft constraint.



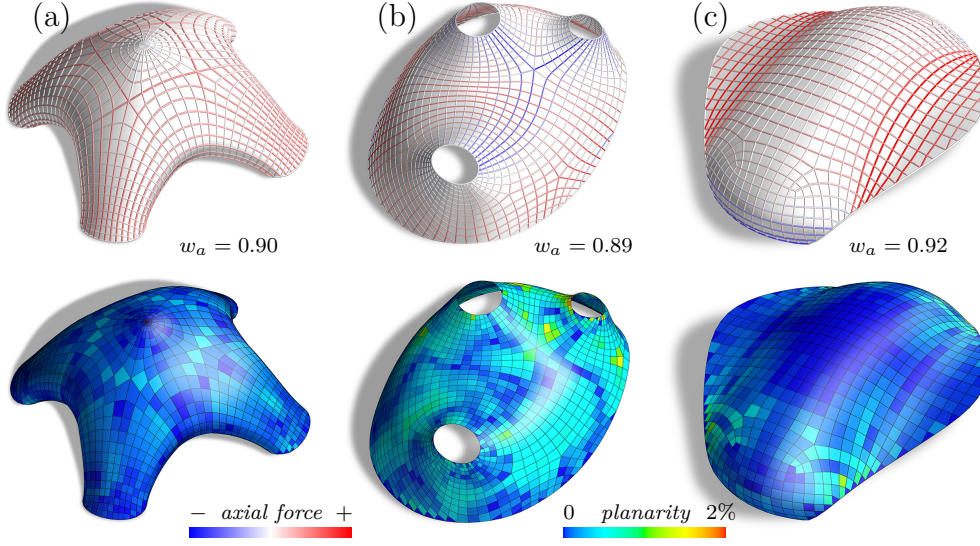
**Figure 4.4** – Results. A high-genus principal mesh in equilibrium. Finite element analysis showed an axial work ratio  $w_a$  of 0.89. On the left, the starting mesh  $M_0$  is shown.

#### 4.4.2 Solver

For the optimization, we use the *guided projection* method of Tang et al. [55]. This method works best for systems of quadratic constraints. To reduce the degree of the main constraints when higher than two, we introduce secondary variables that are quadratic functions of the main ones; then, these functions are added as constraints. Let us rearrange all the variables, in number of  $m$ , in the vector  $\mathbf{x} \in \mathbb{R}^m$ . Let then  $\varphi_l(\mathbf{x}) = 0$ ,  $l = \{1, \dots, n\}$ , be the equations given by the constraints and the target functions. It is possible to add more or less importance to a specific constraint or target function by multiplying the corresponding equations by a weight  $\omega_l$ . The system is solved iteratively. At each iteration  $k$ , given the current variable vector  $\mathbf{x}^k$ , each equation is linearized with a 1st order Taylor expansion:

$$\varphi_l(\mathbf{x}) \approx \varphi_l(\mathbf{x}^k) + \nabla \varphi_l(\mathbf{x}^k)^T (\mathbf{x} - \mathbf{x}^k) = 0.$$

The linearized system of weighted equations can be rearranged in matrix form as  $H\mathbf{x} = \mathbf{r}$ , with  $H \in \mathbb{R}^{n \times m}$  and  $\mathbf{r} \in \mathbb{R}^n$ . To guarantee mesh quality and smoothness during the optimization, we add a fairness energy; we define it at each vertex  $\mathbf{v}_i$  as the squared norm of the distance between  $\mathbf{v}_i$  and the barycenter of its connected vertices  $\mathbf{v}_{j \sim i}$ . The total fairness energy can be written in matrix form as  $\|K\mathbf{x} - \mathbf{s}\|^2$ . Additionally, the distance from  $\mathbf{x}^k$  is



**Figure 4.5** – Results. Principal meshes in equilibrium achieved with our method. Meshes (a) and (c) discretize non-height field shapes. All boundaries are supported. The gridshell structures are in axial equilibrium under a homogeneous vertical load per unit surface area. Axial forces, planarity error and axial work ratios  $w_a$  are shown.

used as a regularizer. The successive variable vector  $\mathbf{x}^{k+1}$  is found by solving

$$\|H\mathbf{x} - \mathbf{r}\|^2 + \delta^2\|K\mathbf{x} - \mathbf{s}\|^2 + \epsilon^2\|\mathbf{x} - \mathbf{x}^k\|^2 \rightarrow \min, \quad (4.6)$$

with  $\delta, \epsilon \in (0, 1)$  as weights. The iteration stops when a desired accuracy is achieved, or when no more improvement is gained. For further details on *guided projection*, see [55].

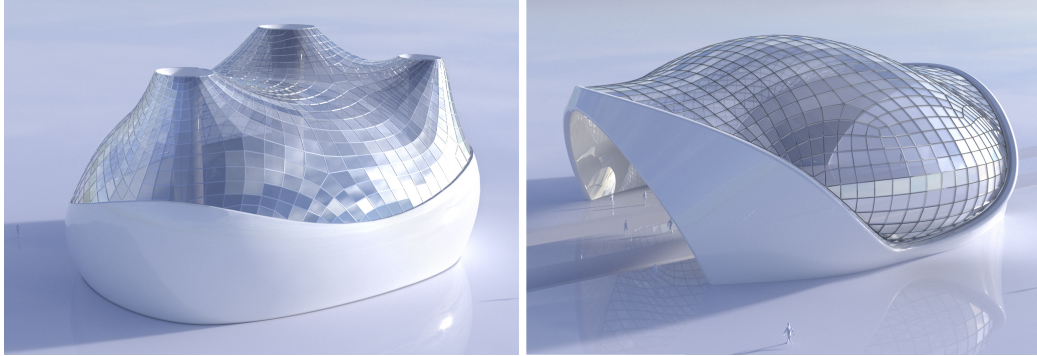
## 4.5 Results and discussion

The proposed workflow has been tested on some sample architectural surfaces. Results are shown in figs. 4.3 to 4.5. The presented examples were subject to a uniform load per unit surface area and supported along the boundary. Optimization times of *step 1* are given in table 4.1. To evaluate the quality of the result, we used the following two criteria.

- *Convergence of post-optimization.* As seen in section 4.3.3, the quad mesh is post-optimized for equilibrium and face planarity with the

Fig.	$ V $	$ E $	iter.	time (s)
4.3	681	1960	12	13.8
4.4	1941	5784	15	78.4
4.5a	606	1760	15	16.7
4.5b	1140	3302	14	31.9
4.5c	1089	3136	13	28.3

**Table 4.1** – Optimization times and corresponding number of iterations for stress and curvature alignment, relative to the presented results. Values refer to triangular meshes with  $v$  vertices and  $e$  edges. The algorithm has been implemented in Python and tested with an Intel Core i7-6700HQ CPU with 2.60 GHz and a 15.9 GB RAM memory.



**Figure 4.6** – Architectural applications. Steel-glass gridshells achievable with the meshes shown in Figures 4.5b (on the left) and 4.5c (on the right). Face planarity errors below 2% are compatible with cladding through flat glass panels.

method of [55]. In this step, the supported vertices are let to glide along the corresponding boundary. The planarity error of quadrilateral faces is estimated as the distance between the two face diagonals divided by their mean length. Regarding equilibrium, the error per vertex is estimated as the norm of equilibrium eq. (3.2) divided by the mean vertex load magnitude. Convergence of post-optimization was considered successful when it reached a maximum planarity error below 2% and a mean equilibrium error below 1%. In the test samples, convergence was achieved in less than ten iterations, noticing small changes in the mesh.

- *Finite element analysis.* In actual gridshells the structure is dimen-

sioned according to finite element analysis. It is of interest to evaluate the effectiveness of the optimization in this way as well. For this purpose, the final mesh was modeled as a frame with steel S235 Timoshenko beam elements, connected together with rigid joints. Area loads were lumped in the nodes. The size of the cross section was chosen constant, according to resistance verification. To evaluate the equilibrium hypothesis, we computed the ratio of internal elastic work due to axial forces in the beams over the total elastic work made by external loads. Axial work ratios  $w_a$ , found for the results, are shown beside figs. 4.3 to 4.5.

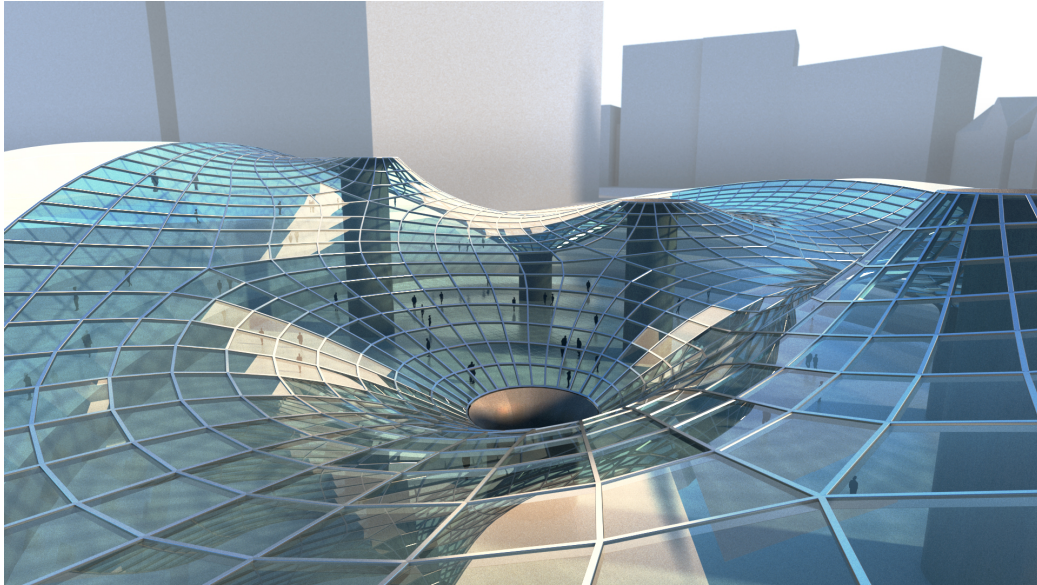
# Chapter 5

## Material-minimizing gridshells

Principal meshes in equilibrium, shown in chapter 4, are an appealing layout for freeform gridshells. In this way we achieve flat panels, a torsion-free substructure, and axial force equilibrium. We now want to go further and look for the gridshells with the best statics performance. According to a groundbreaking result of the Australian engineer A. G. M. Michell, the most efficient trusses, at the limit of refinement, possess an orthogonal quad combinatorics, therefore compatible with principal meshes.

This chapter combines the classical work of Maxwell, Michell, and Airy with differential-geometric considerations and obtain a geometric understanding of statics optimality of gridshell structures. It turns out that the gridshells that use the smallest amount of structural material are orthogonal quad meshes discretizing the principal stress lines of membranes. The absolute minimum, for some given boundary conditions, is achieved on such membranes where the sum of absolute principal stresses is minimal.

We enable then the modeling of structures of minimal weight through the minimization of absolute stresses on triangulated surfaces. Even if statics optimality does not require principality, this can be added on top of our optimization, together with other properties relevant for building construction like alignment with prescribed boundaries. It is then possible to design principal meshes in equilibrium where, for given boundary conditions, the volume of the structure is minimized.



**Figure 5.1** – This chapter provides a tool for freeform architectural design that performs a combined form and stress optimization with the goal to create structures of minimal weight. It incorporates features relevant to statics and to architectural design like alignment of stress and curvature directions, similarly to chapter 4, and the alignment of principal curves with the boundary. The workflow generating this example is shown by fig. 5.6.

## 5.1 Overview and contribution

After a recap of known material in section 5.3 about Maxwell lemma and Michell theorem, in section 5.4 a new result on 2D optimal trusses is presented, starting with relations between discrete curvatures of an Airy potential on the one hand, and the total volume of a truss on the other hand. This topic is interesting because of its connection to differential geometry and because it is relevant to applications, despite the restriction to 2D. An interesting point here is that the combinatorics of optimal structures is part of the solution. The chapter continues with optimal gridshells in section 5.5. This topic is a bit more involved than the 2D case, but we are able to exploit analogies. We derive a procedure for optimization, described in section 5.6. Results are shown and discussed in section 5.7. This chapter presents results published in [28].



## 5.2 Previous work

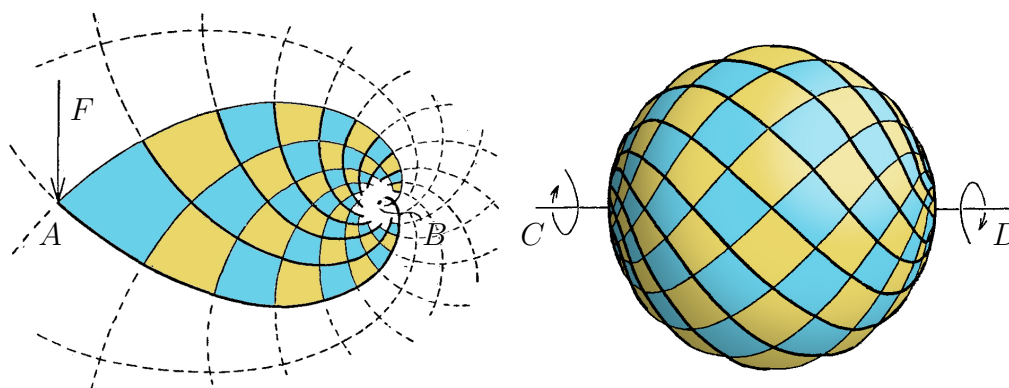
In 1904 A. G. M. Michell, in his seminal paper *The Limits of Economy of Material in Frame-structures* [34], solves a problem far ahead of his time: finding the optimal truss layout that can safely carry some given loads to prescribed support positions. One of the main intuitions of Michell is to turn the problem of optimal frames into a continuum problem, where beams become a fibrous microstructure; a finite truss can be then seen as an approximation, at an appropriate scale, of an optimal continuum solution.

The pioneering work of Michell has been discussed and further investigated by several authors. The exact solutions to specific problems have been developed around the seventies and can be found in the books of Cox [17] and Hemp [23]. Rozvany [47] revises the original formulation for different stress limits in tension and compression. Prager [46] derives a discrete “Hencky-Prandtl” property for turning angles between beams in optimal 2D trusses. Baker et al. [6] studies optimality in connection with Maxwells reciprocal force diagrams and discuss primal/dual pairs of optimal trusses. Recently, the analytical formulation has been embedded into the systematic theory of optimization by Whittle [58] and Lewiński and Sokół [29]. An extended treatment of Michell structures can be found in the recent book of Lewiński et al. [30].

Recently Michells work has been extended to shell-like structures by Mitchell [36]. This chapter is also concerned with this topic. It should be emphasized that in this search for optimal structures, the combinatorics of the structure is part of the solution. This aspect seems to have been neglected in the geometry processing community so far. E.g. Jiang et al. [27] optimize space frames (not gridshells), keeping the combinatorics unchanged.

Imposing optimality properties on structures may not only influence the layout and combinatorics of the structure, but also the shape of the surface which the structure follows. This leads to *computational optimization as form-finding*. This principle is not new, cf. [9]. Here it is applied to gridshells, which was first done by Mitchell [36].





**Figure 5.2** – Minimal volume trusses shown in the paper of Michell [34]. On the left, the solution for a single load  $F$  applied in  $A$ , acting orthogonal to  $AB$ , balanced by opposite force and torque in  $B$ . The minimum frame is formed by two families of similar equiangular spirals with origin in  $B$ , intersecting orthogonally. On the right, the solution to equal and opposite couples applied at points  $C$  and  $D$  on the straight line  $CD$ . Up to now, this is the only known analytical solution in three dimensional space.

## 5.3 Michell trusses

This section shows the main results on Michell trusses, based on the method of Lagrange duality as in the formulation of Lewiński and Sokół [29]. It turns out that the primal and dual versions of the problem statement correspond to minimizing volume under equilibrium constraints, and maximizing work under deformation constraints, respectively. The primal version of the problem will be further developed in the following sections towards the optimization of gridshells.

### 5.3.1 Problem statement

Michell trusses are solutions to the following problem: *given a set of loads with their application points, support positions, and a domain  $\mathcal{D} \subseteq \mathbb{R}^3$ , find the framework structure  $M \subseteq \mathcal{D}$  of minimal total volume that can support the given loads.* The problem is solved under the following assumptions:

- *assumption (i).* The optimal solution is a truss, with axial stresses homogeneously distributed over the cross section.

- *assumption (ii)*. The self-weight of the structure is neglected.
- *assumption (iii)*. The truss experiences small displacements (geometric linearity).

### 5.3.2 Equilibrium and displacement

Let us consider a truss  $M \subseteq \mathcal{D}$  with nodes  $\mathbf{v}_i$ , and let  $\mathbf{p}_i$  be the corresponding nodal loads, collected respectively in the vectors  $\mathbf{v}, \mathbf{p} \in \mathbb{R}^{3|V|}$ . As a consequence of the applied loads, each  $e$ -th beam is stressed by an axial force of magnitude  $f_e$ . All bar force magnitudes are collected in the vector  $\mathbf{f} \in \mathbb{R}^{|E|}$ . Let us now imagine the truss undergoing to a consistent displacement, and let  $\boldsymbol{\lambda}_i$  be the displacements of nodes  $\mathbf{v}_i$ , collected as  $\mathbf{v}$  in the vector  $\boldsymbol{\lambda} \in \mathbb{R}^{3|V|}$ . Accordingly, each beam  $e$  experiences an elongation  $\Delta_e(\boldsymbol{\lambda})$ . According to *assumption (iii)*, the relation  $\Delta_e(\boldsymbol{\lambda})$  is linear in  $\boldsymbol{\lambda}$ . The beam *axial strain* is defined as

$$\varepsilon_e(\boldsymbol{\lambda}) = \frac{\Delta_e(\boldsymbol{\lambda})}{\ell_e}, \quad (5.1)$$

where  $\ell_e$  is the length of the undeformed bar.

If the truss is in static equilibrium under the given loads  $\mathbf{p}$ , the *principle of virtual work* asks that, under every displacement  $\boldsymbol{\lambda}$  compatible with the constraints, the corresponding virtual work made by external forces  $\mathbf{p}$  equals the internal virtual work made by axial forces  $\mathbf{f}$  in the beams:

$$\mathbf{p}^T \boldsymbol{\lambda} - \sum_e f_e \Delta_e(\boldsymbol{\lambda}) = 0, \quad \forall \boldsymbol{\lambda}. \quad (5.2)$$

### 5.3.3 Volume and stress limit

Let  $A_e$  be the cross section area of the  $e$ -th beam. The total volume  $\mathcal{V}$  of the structure can be computed as

$$\mathcal{V} = \sum_e A_e \ell_e.$$

To be safe, a structure must respect the stress limits allowed by its material. On a truss, if we assume a homogeneous distribution of stresses among

the cross section area, the stress  $\sigma_e$  acting at each cross section of the  $e$ -th beam is

$$\sigma_e = \frac{f_e}{A_e}.$$

Given a material, let  $\sigma_c$  and  $\sigma_T$  be its limit allowable stresses in compression and in tension respectively. Let us label the beams in compression and tension with the indices  $c$  and  $t$  respectively. The beam cross sections are then bounded by the following limits:

$$A_t \geq \frac{f_t}{\sigma_T}; \quad A_c \geq \frac{f_c}{\sigma_c},$$

and consequently

$$\mathcal{V} \geq \sum_t \frac{1}{\sigma_T} f_t \ell_t + \sum_c \frac{1}{\sigma_c} f_c \ell_c = \mathcal{V}^*. \quad (5.3)$$

Here  $\mathcal{V}^*$  is a lower bound for the volume  $\mathcal{V}$ , and we have  $\mathcal{V} = \mathcal{V}^*$  when the truss is *fully stressed*, i.e. all beams are stressed up to the limit in tension or in compression. Since the elastic equilibrium depends on the structure stiffness, depending in turn on beam cross sections, for a given set of equilibrated beam forces  $\mathbf{f}$ , the volume  $\mathcal{V}^*$  could be unreachable. A special case are statically determinate trusses, where  $\mathbf{f}$  does not depend on the cross section areas. In these cases, the fully stressed volume  $\mathcal{V}^*$  can always be attained. It will turn out that when  $\mathcal{V}^*$  is a minimum, the truss is statically determinate. Therefore, this is also a minimum for the actual volume  $\mathcal{V}$ .

Let us define now the *normalized fully stressed volume*  $\hat{\mathcal{V}}^*$  as

$$\hat{\mathcal{V}}^* = \sigma_0 \mathcal{V}^*, \quad \text{with} \quad \sigma_0 = \frac{\sigma_T - \sigma_c}{2}.$$

In the following, we will seek the minimum of  $\hat{\mathcal{V}}^*$ , that dimensionally is equivalent to work.

### 5.3.4 Maxwell lemma

Michell bases his formulation on a previous result shown by Maxwell in 1872 [33]. Maxwell considers a truss under given nodal forces  $\mathbf{p}$ , corresponding

to applied loads or support reactions. He imagines such a truss undergoing to a uniform contraction until it collapses into a single point. Consequently to this virtual displacement, each bar undergoes to an elongation  $\Delta_e = -\ell_e$ , while the external virtual work made by the forces  $\mathbf{p}$  is a constant  $C$  for all trusses supporting the loads  $\mathbf{p}$ . The equilibrium yields

$$\sum_c f_c \ell_c + \sum_t f_t \ell_t = C.$$

In case the truss has no members in tension or no members in compression, the internal virtual work is constant, and so is the fully stressed volume. It follows this first important result on truss optimization.

**Proposition 4** (Maxwell lemma). *If in a truss all members are only in tension or only in compression while supporting some given loads, the fully stressed volume  $\mathcal{V}^*$  is a minimum.*

### 5.3.5 Dual formulation

Let us now consider all possible trusses supporting the given loads  $\mathbf{p}$ , defined by the node vector  $\mathbf{v} \subseteq \mathcal{D}$  and the beams force vector  $\mathbf{f}$ . We now want to solve, for  $\mathbf{v}$  and  $\mathbf{f}$ ,

$$\hat{\mathcal{V}}^*(\mathbf{v}, \mathbf{f}) \rightarrow \min,$$

subject to

$$\mathbf{p}^T \boldsymbol{\lambda} - \sum_t f_t \ell_t \varepsilon_t(\boldsymbol{\lambda}) - \sum_c f_c \ell_c \varepsilon_c(\boldsymbol{\lambda}) = 0,$$

with

$$f_t \geq 0, \quad f_c \leq 0,$$

where we split the internal virtual work of eq. (5.2) in its compression and tension components, and we write out the elongations  $\Delta_e$  through strains  $\varepsilon_e$  with eq. (5.1).

According to the linearity of elongations  $\Delta_e$  with respect to the displacements  $\boldsymbol{\lambda}$ , the virtual work eq. (5.2) is linear in  $\boldsymbol{\lambda}$  as well. This implies that the derivatives of the virtual work equation with respect to  $\boldsymbol{\lambda}$  are again equilibrium equations (these gives indeed the nodal equilibrium eq. (3.1)). We

can then introduce the nodal displacement vector  $\boldsymbol{\lambda}$  as Lagrange multiplier, and write the Lagrangian equation

$$\mathcal{L}(\mathbf{v}, \mathbf{f}, \boldsymbol{\lambda}) = \hat{\mathcal{V}}^* - \sum_t f_t l_t \varepsilon_t(\boldsymbol{\lambda}) - \sum_c f_c l_c \varepsilon_c(\boldsymbol{\lambda}) + \mathbf{p}^T \boldsymbol{\lambda}.$$

We take now the dual function

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{v}} \min_{\substack{\mathbf{f} \\ f_t \geq 0, f_c \leq 0}} \mathcal{L}(\mathbf{v}, \mathbf{f}, \boldsymbol{\lambda}).$$

Since the external work  $\mathbf{p}^T \boldsymbol{\lambda}$  does not depend on  $\mathbf{f}$ , exchanging the max operations on  $\mathbf{v}$  and  $\boldsymbol{\lambda}$ , we can solve the dual problem

$$\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) = \min_{\mathbf{v}} \max_{\boldsymbol{\lambda}} (\mathbf{p}^T \boldsymbol{\lambda} + K(\boldsymbol{\lambda}))$$

with

$$K(\boldsymbol{\lambda}) = \min_{\substack{\mathbf{f} \\ f_t \geq 0, f_c \leq 0}} \left[ \sum_t \left( \frac{\sigma_0}{\sigma_T} - \varepsilon_t(\boldsymbol{\lambda}) \right) f_t l_t + \sum_c \left( \frac{\sigma_0}{\sigma_C} - \varepsilon_c(\boldsymbol{\lambda}) \right) f_c l_c \right],$$

where the fully stressed volume  $\mathcal{V}^*$  has been written with eq. (5.3). Let us rename the normalized limit strains as

$$\frac{\sigma_0}{\sigma_T} = \varepsilon_+; \quad \frac{\sigma_0}{\sigma_C} = \varepsilon_-.$$

The minimum over  $\mathbf{f}$  is attained when

$$K(\boldsymbol{\lambda}) = \begin{cases} 0, & \text{if } \varepsilon_t = \varepsilon_+ \text{ and } \varepsilon_c = \varepsilon_- \text{ with } f_t \geq 0, f_c \leq 0 \\ 0, & \text{if } \varepsilon_t < \varepsilon_+ \text{ and } \varepsilon_c > \varepsilon_- \text{ with } f_t = 0, f_c = 0 \\ 0, & \text{if } \varepsilon_t = \varepsilon_+ \text{ and } \varepsilon_c > \varepsilon_- \text{ with } f_t \geq 0, f_c = 0 \\ 0, & \text{if } \varepsilon_t < \varepsilon_+ \text{ and } \varepsilon_c = \varepsilon_- \text{ with } f_t = 0, f_c \leq 0 \\ -\infty, & \text{if } \varepsilon_t > \varepsilon_+ \text{ with } f_t = +\infty \\ -\infty, & \text{if } \varepsilon_c < \varepsilon_- \text{ with } f_c = -\infty \end{cases} \quad (5.4)$$

The last two cases are excluded by the max operation. We have then

$$\min_{\mathbf{v}} \min_{\substack{\mathbf{f} \\ \text{s.t. eq. (5.2)}}} \hat{\mathcal{V}}^* = \min_{\mathbf{v}} \max_{\substack{\boldsymbol{\lambda} \\ \varepsilon_e \in [\varepsilon_-, \varepsilon_+]}} \mathbf{p}^T \boldsymbol{\lambda}. \quad (5.5)$$

### 5.3.6 Continuum formulation

If we increase the number of nodes  $\mathbf{v}_i$ , since we add more degrees of freedom, it is reasonable to expect solutions with decreasing minimal volumes  $\mathcal{V}^*$ . We can keep adding nodes  $\mathbf{v}_i$  until we reach a limit formulation where  $\mathbf{v} \equiv \mathcal{D}$ , obtaining in this way a continuous fibrous material, referred to as *truss-like continuum*. Passing through a phase of continuous formulation is a common practice in discrete optimization, see for instance [58]. The main advantage of this approach is that, at the continuous limit, the dependency of eq. (5.5) on  $\mathbf{v}$  becomes redundant.

Let us now restrict our attention to a subclass of solutions where the nodal displacements  $\boldsymbol{\lambda}$  can be derived from a continuous displacement field  $\boldsymbol{\lambda}_{\mathcal{D}}$  of the domain  $\mathcal{D}$ . As a consequence of a displacement  $\boldsymbol{\lambda}_{\mathcal{D}}$ , the domain  $\mathcal{D}$  undergoes to a strain field  $\boldsymbol{\varepsilon}$ . For small displacements, the first order strain is given by

$$\boldsymbol{\varepsilon}(\boldsymbol{\lambda}_{\mathcal{D}}) = \frac{1}{2}(\nabla \boldsymbol{\lambda}_{\mathcal{D}} + \nabla \boldsymbol{\lambda}_{\mathcal{D}}^T).$$

Consider now a direction  $\mathbf{a}$  emanating from a point of  $\mathcal{D}$ . The axial strain along the direction  $\mathbf{a}$  is given by

$$\varepsilon_{\mathbf{a}} = \frac{\mathbf{a}^T \boldsymbol{\varepsilon} \mathbf{a}}{\mathbf{a}^T \mathbf{a}}.$$

According to the limits of  $\varepsilon_e$  from eq. (5.4), an infinitesimal beam in direction  $\mathbf{a}$  will be active if  $\varepsilon_{\mathbf{a}} = \varepsilon_+$  or  $\varepsilon_{\mathbf{a}} = \varepsilon_-$ . Since we have the condition  $\varepsilon_- \leq \varepsilon_{\mathbf{a}} \leq \varepsilon_+$ , the values  $\varepsilon_+$  and  $\varepsilon_-$  must be eigenvalues of  $\boldsymbol{\varepsilon}$ , and then  $\mathbf{a}$  is aligned with one of the eigenvectors  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  or  $\mathbf{e}_3$  of  $\boldsymbol{\varepsilon}$ .

Let us then consider a displacement field  $\boldsymbol{\varepsilon}$  of  $\mathcal{D}$  with eigenvalues  $\varepsilon_i \in [\varepsilon_-, \varepsilon_+]$ . Let us denote all the eigenvalues that do not attain the bounds as  $\varepsilon_{\sim}$ . According to the associated triplet of eigenvalues  $\text{spec}(\boldsymbol{\varepsilon})$ , we can therefore distinguish the following cases for an infinitesimal volume at each

point of  $\mathcal{D}$ :

1.  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_+, \varepsilon_+, \varepsilon_+\}$  or  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_-, \varepsilon_-, \varepsilon_-\}$ : *Maxwell volume*, the infinitesimal volume is isotropically strained, any layout of fibers is optimal.
2.  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_+, \varepsilon_+, \varepsilon_-\}$  or  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_+, \varepsilon_-, \varepsilon_-\}$ : *Michell volume*, the infinitesimal volume is isotropically strained on a plane, with opposite strain along the orthogonal direction. The fiber layout consists of a surface, with no preferred layout, and a fan of fibers orthogonal to it.
3.  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_+, \varepsilon_+, \varepsilon_\sim\}$  or  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_-, \varepsilon_-, \varepsilon_\sim\}$ : *Maxwell surface*, where only fibers on a plane are active, with no preferred layout.
4.  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_+, \varepsilon_-, \varepsilon_\sim\}$ : *Michell surface*, where only fibers on a plane are active and disposed along the two extremal eigenvectors, no other fibers are active.
5.  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_+, \varepsilon_\sim, \varepsilon_\sim\}$  or  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_-, \varepsilon_\sim, \varepsilon_\sim\}$ : *Fan*, the fibers are straight lines disposed as a fan.
6.  $\text{spec}(\boldsymbol{\varepsilon}) = \{\varepsilon_\sim, \varepsilon_\sim, \varepsilon_\sim\}$ : *Void*, no fibers are active.

Michell volumes can be seen as a family of Maxwell surfaces, stressed as an isotropic membrane under normal load and connected with one family of orthogonal lines, or as two families of Michell surfaces. According to these considerations, we have the following result.

**Proposition 5.** *In a truss-like continuum of minimum volume  $\mathcal{V}^*$ , if a tension member meets a compression member, they must do so at right angles.*

In the truss-like continuum, instead of forces  $f_e$ , we can define a stress tensor  $\boldsymbol{\sigma}$  (see section 3.2.1) such that the axial stress along a fiber in direction  $\mathbf{a}$  is given by

$$\sigma_{\mathbf{a}} = \frac{\mathbf{a}^T \boldsymbol{\sigma} \mathbf{a}}{\mathbf{a}^T \mathbf{a}}. \quad (5.6)$$

This quantity is a force per unit area. Divided by the limit stresses  $\sigma_C$  or  $\sigma_T$ , it gives us the density of the fully stressed fibers in direction  $\mathbf{a}$ .

Let us now consider the Michell problem into a design domain  $\mathcal{D}$ , with interior applied loads  $\mathbf{p}$ , boundary applied loads  $\mathbf{b}$ , and constraints  $\lambda_i = 0$ , undergoing to a virtual strain  $\boldsymbol{\varepsilon}$ . The virtual work is then given by a linear function  $\mathcal{W}(\boldsymbol{\varepsilon})$ . Let  $\mathcal{S} = \text{Sym}^2(\mathcal{D})$  be the space of second order symmetric tensors on  $\mathcal{D}$ , and  $\mathcal{S}_{[\varepsilon_-, \varepsilon_+]}$  the tensors of  $\mathcal{S}$  whose eigenvalues are in the interval  $[\varepsilon_-, \varepsilon_+]$ . Let  $\mathcal{S}_{\text{div}\mathbf{p}}$  the symmetric tensor space whose divergence equals  $-\mathbf{p}$ . Recalling the equilibrium eq. (3.6), we can write the continuum version of eq. (5.5) as:

$$\min_{\substack{\boldsymbol{\sigma} \in \mathcal{S}_{\text{div}\mathbf{p}} \\ \boldsymbol{\sigma}\mathbf{n} + \mathbf{b} = 0}} \hat{\mathcal{V}}^* = \max_{\substack{\boldsymbol{\varepsilon} \in \mathcal{S}_{[\varepsilon_-, \varepsilon_+]} \\ \lambda_i = 0}} \mathcal{W}, \quad (5.7)$$

where  $\mathbf{n}$  is the outer normal of the boundary  $\partial\mathcal{D}$ . Here, on the left side, we have the primal problem: the minimization of volume subject to stress equilibrium. On the right side, we have the dual formulation: the maximization of work under strain constraints. This latter was the version given by Michell in 1904.

### 5.3.7 Back to the primal problem

Let us now go back to the primal problem. According to proposition 4 and 5, we can suppose that, in an optimal truss-like continuum, the fibers are aligned with the principal strain directions  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  and  $\mathbf{e}_3$  of a certain strain field  $\boldsymbol{\varepsilon} \in \mathcal{S}_{[\varepsilon_-, \varepsilon_+]}$ . Let  $\sigma_{\mathbf{e}_1}$ ,  $\sigma_{\mathbf{e}_2}$  and  $\sigma_{\mathbf{e}_3}$  be the corresponding stresses, as from eq. (5.6), and let us define the function  $\mu(\sigma_{\mathbf{e}_i})$ , with  $i = 1, 2, 3$ , as

$$\mu(\sigma_{\mathbf{e}_i}) = \begin{cases} 1/\sigma_T, & \text{if } \sigma_{\mathbf{e}_i} \geq 0 \\ 1/\sigma_C, & \text{if } \sigma_{\mathbf{e}_i} < 0 \end{cases}.$$

Consider now an infinitesimal element  $d\xi_1, d\xi_2, d\xi_3$ , aligned respectively with the directions  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  and  $\mathbf{e}_3$ . The cross section area of the beam along  $\mathbf{e}_i$  is then given by

$$dA_{\mathbf{e}_i} = \mu(\sigma_{\mathbf{e}_i}) |\sigma_{\mathbf{e}_i}| d\xi_j d\xi_k.$$



Its length equals  $d\xi_i$ . Its infinitesimal fully stressed volume is then:

$$d\mathcal{V}_{\mathbf{e}_i}^* = \mu(\sigma_{\mathbf{e}_i})|\sigma_{\mathbf{e}_i}| d\xi_i d\xi_j d\xi_k.$$

Therefore, in an optimal truss-like continuum, the total normalized fully stressed volume can be computed as

$$\hat{\mathcal{V}}^* = \sigma_0 \int_{\mathcal{D}} \mu(\sigma_{\mathbf{e}_1})|\sigma_{\mathbf{e}_1}| + \mu(\sigma_{\mathbf{e}_2})|\sigma_{\mathbf{e}_2}| + \mu(\sigma_{\mathbf{e}_3})|\sigma_{\mathbf{e}_3}|.$$

The principle of virtual work asks that  $\mathcal{W}(\boldsymbol{\varepsilon})$  equals the internal virtual work made by the stress tensor:

$$\mathcal{W}(\boldsymbol{\varepsilon}) = \int_{\mathcal{D}} \langle \boldsymbol{\sigma}, \boldsymbol{\varepsilon} \rangle.$$

Here the inner product between tensors is intended as sum of element-wise multiplications. We can write now the Lagrangian equation

$$\mathcal{L}(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}) = \hat{\mathcal{V}}^*(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}) + \mathcal{W}(\boldsymbol{\varepsilon}) - \int_{\mathcal{D}} \langle \boldsymbol{\sigma}, \boldsymbol{\varepsilon} \rangle$$

with  $\boldsymbol{\varepsilon}(\boldsymbol{\lambda}_{\mathcal{D}})$  as Lagrange multiplier, and then solve the dual problem

$$\max_{\boldsymbol{\varepsilon}} g(\boldsymbol{\varepsilon}) = \max_{\boldsymbol{\varepsilon}} (\mathcal{W}(\boldsymbol{\varepsilon}) + H(\boldsymbol{\varepsilon})),$$

where min and max operation has been exchanged, and where

$$H(\boldsymbol{\varepsilon}) = \min_{\boldsymbol{\sigma}} \left( \hat{\mathcal{V}}^*(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}) - \int_{\mathcal{D}} \langle \boldsymbol{\sigma}, \boldsymbol{\varepsilon} \rangle \right).$$

For every  $\sigma_{\mathbf{e}_1}, \sigma_{\mathbf{e}_2}, \sigma_{\mathbf{e}_3}$ , it asks for

$$\max_{\boldsymbol{\sigma}} \int_{\mathcal{D}} \langle \boldsymbol{\sigma}, \boldsymbol{\varepsilon} \rangle.$$

Due to the CauchySchwarz inequality, we have

$$\langle \boldsymbol{\sigma}, \boldsymbol{\varepsilon} \rangle^2 \leq \langle \boldsymbol{\sigma}, \boldsymbol{\sigma} \rangle \langle \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \rangle$$

and the equality holds if the eigenvectors of  $\boldsymbol{\sigma}$  are pointing in the same direction of the eigenvectors of  $\boldsymbol{\varepsilon}$ . Therefore, we can see that in a truss-like continuum with minimal fully stressed volume  $\mathcal{V}^*$ , the fibers are aligned along principal stress directions.

Let suppose now that a certain optimal truss is a solution of eq. (5.7). At each point of  $\mathcal{D}$  we have then a given orientation for the stress tensor eigenvectors; there are only three remaining degrees of freedom. For given boundary conditions, the three differential equations 3.6 of equilibrium are uniquely determined. We can then state the following:

**Proposition 6.** *A truss-like continuum of minimal volume  $\mathcal{V}^*$  under given loads is statically determinate.*

In case some loads are undetermined, being the reactions of constraints, the solution is not unique. In these cases, different values of stress limits  $\sigma_C$  and  $\sigma_T$  will yield different solutions. The missing loads will undergo to minimization; once found, the stress state in the structure is determined.

We have now all the elements to set up the *primal formulation* of the volume minimization problem. Being  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$  the eigenvalues of  $\boldsymbol{\sigma}$ , the optimal truss-like continuum is found as

$$\min \mathcal{V} = \min_{\substack{\boldsymbol{\sigma} \in \mathcal{S}_{\text{divp}} \\ \boldsymbol{\sigma} \mathbf{n} = \mathbf{b}}} \int_{\mathcal{D}} \mu(\sigma_1)|\sigma_1| + \mu(\sigma_2)|\sigma_2| + \mu(\sigma_3)|\sigma_3|, \quad (5.8)$$

where some undetermined components of  $\mathbf{p}$  and  $\mathbf{b}$  can possibly take part to the minimization.

## 5.4 Volume-optimal trusses in 2D

Before dealing with gridshells, we will focus on 2D planar structures. In case of materials with symmetric tensile and compressive stress limit, it turns out a relation between discrete curvatures of an Airy potential on the one hand, and the total volume of a truss on the other hand.

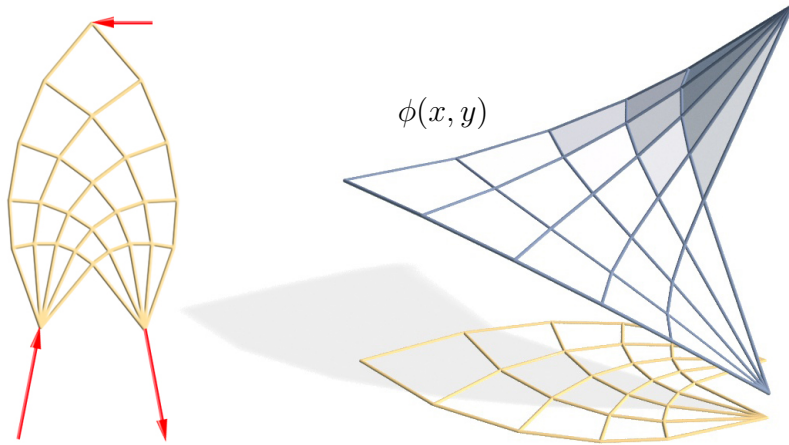
### 5.4.1 Connection between kinks and volumes

Let us consider structures made out of materials with  $\sigma_T = -\sigma_C = \sigma$ , as is the case for steel. The fully stressed volume of a truss with elements  $e$  is then

$$\mathcal{V}^* = \frac{1}{\sigma} \sum_e |f_e| \ell_e. \quad (5.9)$$

The discrete Airy potential can be used here to establish a first new relation between volumes and discrete curvatures. Having in mind the properties of continuous optimal structures, we consider *circular quad meshes*. The geometric setting appropriate to our situation is 3D *isotropic geometry*, where the slope of a line w.r.t. the horizontal  $xy$  plane plays the role of an angle, and change in slope (such as a 2nd derivative, or a kink divided by horizontal length) plays the role of curvature, as seen in section 3.2.4. Let  $\mathbf{e}_{ij} = \mathbf{v}_i - \mathbf{v}_j$  be the edge vector, and let  $\mathbf{e}_{ij}^* = \mathbf{v}_l^* - \mathbf{v}_k^*$  be the corresponding dual edge as in the construction of section 3.1.3, where  $l$  and  $k$  are the indices of the left and right faces incident at the edge such that  $e = f_k \cap f_l$ . We need the following ingredients:

- For a member  $e$  connecting vertices  $\mathbf{v}_i, \mathbf{v}_j$ , the product  $|f_e| \ell_e$  (force



**Figure 5.3** – The mesh yielding a 2D volume-optimal truss (shown in yellow) for the three force problem on the left, together with the Airy polyhedron projecting onto it (blue). The angle of transverse mesh polylines enjoys a discrete Hencky-Prandtl property.

times length) is expressed as  $\|\mathbf{e}_{ij}\| \|\mathbf{e}_{ij}^*\|$ , by construction of the dual truss.

- We can define a region of influence of this member, namely the quadrilateral  $\mathbf{v}_i \mathbf{c}_k \mathbf{v}_j \mathbf{c}_l$ , where  $\mathbf{c}_k, \mathbf{c}_l$  are circumcenters of faces adjacent to the member. Its area is  $area_e$ .
- The isotropic angle in the Airy potential along a line which crosses this member at right angles equals  $\pm \|\nabla \phi|_{f_k} - \nabla \phi|_{f_l}\|$  by definition of  $\nabla$ . It equals  $\pm \|\mathbf{e}_{ij}^*\|$  by construction of the potential.
- The isotropic curvature of the Airy potential surface across the edge  $e$  is the isotropic angle divided by the distance of centers  $\mathbf{c}_k, \mathbf{c}_l$ . Let us temporarily call this value  $curvature_e$ .

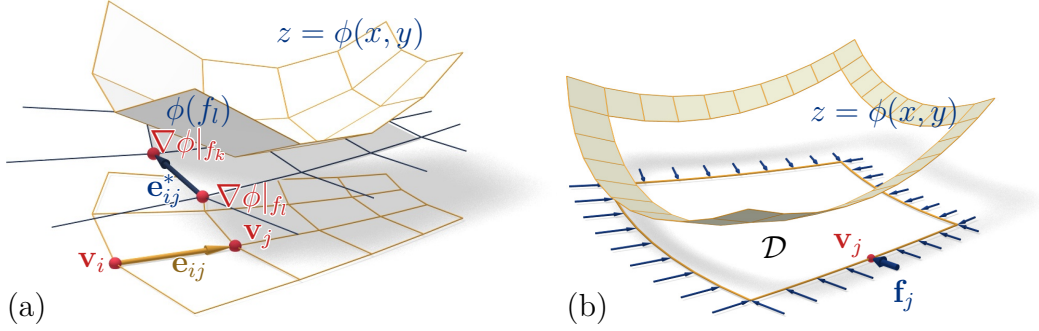
Putting everything together, we get

$$\begin{aligned} \sum_e |f_e| \ell_e &= \sum_e \|\mathbf{e}_{ij}^*\| \|\mathbf{e}_{ij}\| \\ &= \sum_e \|\mathbf{e}_{ij}\| \|\mathbf{c}_k - \mathbf{c}_l\| \frac{\|\mathbf{e}_{ij}^*\|}{\|\mathbf{c}_k - \mathbf{c}_l\|} \\ &= \sum_e 2 |curvature_e| \cdot area_e. \end{aligned} \quad (5.10)$$

Except for the boundary, the areas of influence cover the mesh, so we have converted the volume optimization problem “ $\sum_e |f_e| \ell_e \rightarrow \min$ ” into a total curvature minimization problem. It remains to interpret the formula above and give it a meaning in terms of classical differential geometry.

### 5.4.2 Total isotropic curvature

As seen in section 2.2.4, discretizations of conjugate networks are planar quad meshes [31]. A principal curve network, characterized by conjugacy+orthogonality, requires that a discrete version of orthogonality is imposed on top of conjugacy; such a net thus is discretized as a circular quad mesh in the classical case, and as an  $i$ -circular mesh in the isotropic case (see section 3.2.4): an  $i$ -circular mesh is a quad mesh with planar faces whose projection onto the  $xy$  plane is a 2D circular net [44]. The Airy polyhedra



**Figure 5.4** – *Airy polyhedra of trusses.* (a) An Airy polyhedron  $z = \phi(x, y)$  which projects onto the primal truss. The gradients of the piecewise-linear function  $\phi$  are the vertices of a dual-reciprocal truss – compare with fig. 3.1. (b) Given forces along the polygonal boundary of a planar domain, we incrementally construct the boundary strip of an Airy polyhedron by relating its face gradients with the given forces. Equilibrium ensures the construction closes up. (c) The Airy potential  $\phi(\mathbf{x}(t))$  experiences a kink a point  $\mathbf{x}(t)$  moves across the edge  $\mathbf{v}_i \mathbf{v}_j$  at right angles. The kink is of magnitude  $\|\mathbf{f}_{ij}\| = \|\mathbf{e}_{ij}^*\|$ . This figure also illustrates the quadrilateral  $\mathbf{v}_i \mathbf{c}_l \mathbf{v}_j \mathbf{c}_k$  which serves as region of influence of the member  $\mathbf{v}_i \mathbf{v}_j$ .

erected over 2D circular nets are therefore discretizations of  $i$ -principal curve networks.

Let us now think of a sequence of finer and finer Airy polyhedra which converge to a principal parametrization of a continuous Airy potential  $\phi(x, y)$ , as seen in section 3.2, and let us investigate the limit of eq. (5.10): in isotropic geometry, curvature measures the change in slope w.r.t. progress in the  $xy$  plane (i.e.,  $i$ -curvature is a second derivative of the  $z$  coordinate w.r.t. arc length in the  $xy$  plane). Thus, a discrete version of  $i$ -curvature is the value *curvature<sub>e</sub>* used above (it is a kink divided by a horizontal length). Curvatures along the  $i$ -principal directions are the  $i$ -principal curvatures  $\kappa_1^i, \kappa_2^i$ . Since our limit of circular meshes is principal parametrization, we see that eq. (5.10) discretizes the integral  $2 \int_{\mathcal{D}} |\kappa_1^i| + |\kappa_2^i|$ . Let us summarize this result:

**Proposition 7.** *The infinitesimal members of a volume-minimizing optimal truss-like continuum are aligned with the isotropic-principal directions of the Airy potential surface  $z = \phi(x, y)$ , which minimizes isotropic total absolute curvature, i.e.,*

$$\int_{\mathcal{D}} |\kappa_1^i| + |\kappa_2^i| \rightarrow \min, \quad (5.11)$$

under the given boundary and load conditions. A volume-minimizing discrete truss — if orthogonality is enforced by means of the circular property — are found as projections of a polyhedral Airy potential minimizing a discrete version of isotropic total absolute curvature (again, under the given boundary and load conditions):

$$\sum_e |\text{curvature}_e| \cdot \text{area}_e \rightarrow \min.$$

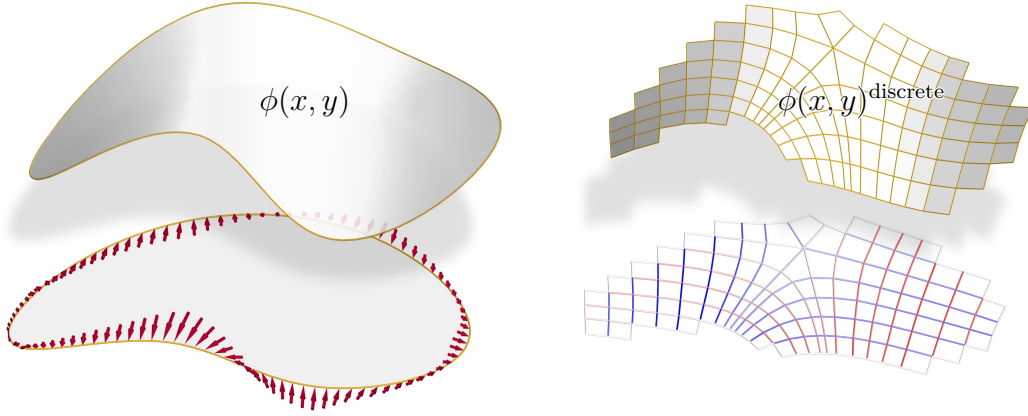
It is not difficult to see that  $\kappa_1^i, \kappa_2^i$  equal the principal stresses, which has already been noticed by Strubecker [51]. It is also easy to see that the minimization of isotropic total absolute curvature of eq. (5.11) is equivalent to the Michell's primal formulation of eq. (5.8) in 2D, for materials with equal tensile and compressive stress limit. We will see that in the 3D (gridshell) case, an analogous result holds, see eq. (5.17).

### 5.4.3 Computing optimal trusses in 2D

We are now able to solve the following problem: *given is a polygonal domain  $\mathcal{D}$  in  $\mathbb{R}^2$  with vertices  $\mathbf{v}_1, \dots, \mathbf{v}_n$ . Further we are given forces  $\mathbf{f}_j$  acting on  $\mathbf{v}_j$  which are in equilibrium (i.e., there is zero net force and zero net torque). Connect the given vertices by a truss in the interior of  $\mathcal{D}$  which balances the given forces in a volume-optimizing manner. A four-step procedure determines the combinatorics and geometry of the solution:*

- *Step 1.* The given loads define an Airy potential  $\phi(x, y)$  outside  $\mathcal{D}$ .
- *Step 2.* Extend  $\phi(x, y)$  to the interior of  $\mathcal{D}$ , minimizing  $\int |\kappa_1^i| + |\kappa_2^i|$ , see fig. 5.5.
- *Step 3.* Find an  $i$ -circular net approximating the surface  $z = \phi(x, y)$ .
- *Step 4.* In theory, this net projects onto an optimal truss. In order to account for discretization errors, apply a final round of direct optimization to this truss.

*Step 1* is illustrated by fig. 5.4b. A piecewise-linear Airy potential  $\phi(x, y)$  in a neighbourhood of  $\mathcal{D}$  just outside the boundary  $\partial\mathcal{D}$  is composed of linear functions  $\phi^j(\mathbf{x}) = \langle \nabla\phi^j, \mathbf{x} \rangle + \gamma_j$ , whose domain is bounded by the edge



**Figure 5.5** – *Computing an optimal truss in 2D.* Left: Given external forces, we compute an Airy potential  $\phi(x, y)$  which minimizes total absolute “isotropic” curvature. Equivalently, we compute a stress state where the integral  $\int |\sigma_1| + |\sigma_2|$  of absolute principal stresses is minimal. *Right:* an optimal circular truss is derived from the principal stress directions; it has a corresponding discrete Airy potential. The misalignment with the boundary shows that constraining the vertices of this circular mesh onto the given boundary will impose additional constraints. Later images in this chapter show the 3D situation, where these constraints have tacitly been taken into account. Color coding indicates tensile (blue) and compressive (red) axial forces.

$\mathbf{v}_j \mathbf{v}_{j+1}$ , and the lines of action of the forces  $\mathbf{f}_j$  and  $\mathbf{f}_{j+1}$  (indices modulo  $n$ ). The Airy potential is unique only up to adding a linear function, so we let  $\nabla \phi^1 = \mathbf{0}$ ,  $\gamma_1 = 0$ . Since  $\phi^{j-1}, \phi^j$  have the same value for the vertex  $\mathbf{v}_j$ , and equilibrium implies that  $\nabla \phi^j = \nabla \phi^{j-1} + \bar{R} \mathbf{f}_j$  (where  $R$  is the 2D 90° rotation), we can recursively define  $\phi^2, \phi^3, \dots$ . Our equilibrium assumption ensures that  $\phi^{n+1} = \phi^1$ , i.e., the construction closes up and  $\phi$  is indeed well defined [5].

*Step 2.* The Airy potential of a volume-optimal truss minimizes a discrete version of  $\int |\kappa_1^i| + |\kappa_2^i|$ . We can therefore switch from discrete to continuous and extend  $\phi$  to the interior of  $\mathcal{D}$  by minimizing this integral. This optimization problem is solved numerically, using a suitable triangulation of  $\mathcal{D}$  which is unrelated to the optimal truss, see section 5.6.6.

*Step 3.* The eigenvectors of the Hessian of  $\phi$  yields the cross field of  $i$ -principal directions on the Airy potential surface. We find a quad mesh aligned with it, using the method of Bommès et al. [13].

*Step 4.* The mesh extracted in step 3 is approximately circular, and at the same time approximately a volume-optimizing truss. Depending on the property we wish to establish in an exact manner, a final round of optimization is applied, using the method of Tang et al. [55]. An example is shown by fig. 5.5.

## 5.5 Volume-optimal gridshells

We want now to extend the results on Michell trusses to our 3D gridshell structures. Note that Maxwell's lemma and Michell's theorem no longer apply here, even if some of their conclusions are still true. This is because on gridshells the self-load amounts to non-constant forces acting in all interior vertices of the structure.

The main aim of this chapter is to design volume-optimal gridshells. It will be argued why such structures should be based on quadrilateral meshes. Like in the 2D case, the combinatorics of the structure is part of the solution.

An essential ingredient of this approach is the separation of horizontal forces from vertical ones (see section 3.2), so that it is possible to treat a projection of our structure as an ordinary 2D truss loaded at the boundary, using the methods shown in section 3.1.

### 5.5.1 Properties of optimal gridshells

When applied to architectural gridshells, the assumptions of section 5.3.1 must be further discussed.

Regarding *assumption (i)*, the restriction to axial only forces here implies that the truss-like continuum is a surface in membrane equilibrium. We call it *truss-like membrane*. The membrane equilibrium of section 3.2 then applies.

Concerning *assumption (ii)*, in actual gridshell architectures the self-weight of the load bearing structure could be not negligible in some cases. However, we can identify two scenarios where nevertheless the structure is fully stressed.

- *Scenario 1* is quad mesh combinatorics. In a quad mesh, we have roughly  $3/2$  times as many linear equilibrium equations as there are



forces. Thus the gridshell is statically determinate. Each member can be made volume-optimal independent of the others, and is therefore fully stressed.

- *Scenario 2* occurs if the load which is proportional to surface area dominates the unknown weight of the structure, which is subject to optimization. This is the case of steel-glass structures. Indeed we might think of panel weights of, say, 30 kg/m<sup>2</sup> and maintenance loads of 50 kg/m<sup>2</sup>. Snow load might increase the total up to 300 kg/m<sup>2</sup>. Generally a steel structure has only 10–30 kg/m<sup>2</sup>, and the variation of this weight experienced during optimization is even less. By neglecting this variation we assume the total load is constant during optimization. Then optimal structures are fully stressed, and the structure is fully stressed without assuming quad mesh combinatorics.

With these assumptions, and in one of the two scenarios, the volume minimization amounts then to

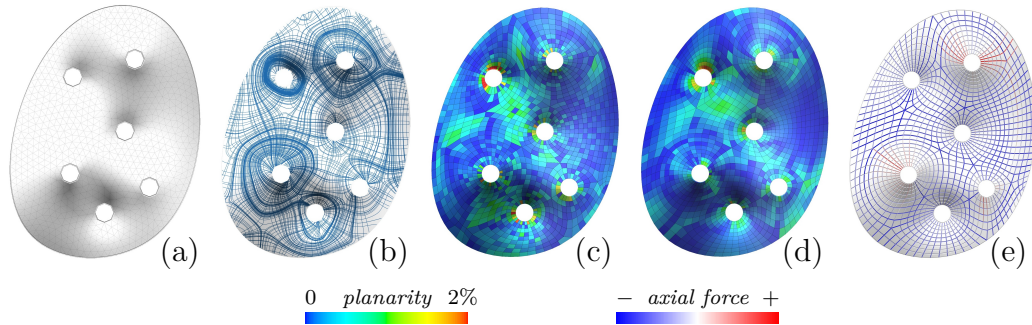
$$\mathcal{V}^* = \sum_t \frac{1}{\sigma_T} f_t \ell_t + \sum_c \frac{1}{\sigma_C} f_c \ell_c \rightarrow \min. \quad (5.12)$$

### 5.5.2 Properties of optimal truss-like continua

With the previous assumptions, we can use eq. (5.8). In a truss-like membrane  $\mathcal{M}$ , the stress tensor  $S$  is defined in the tangent plane and represents the stresses integrated over the normal direction, as seen in section 3.2.3. For simplicity, let us assume again a symmetric material with  $\sigma_T = -\sigma_C$ . Then eq. (5.8) became

$$\int_{\mathcal{M}} |\sigma_1| + |\sigma_2| \rightarrow \min, \quad (5.13)$$

where  $\sigma_1$  and  $\sigma_2$  are the principal stresses of  $S$ . If we parametrize the membrane as a height field (see section 2.2.3), the principal stresses are given by the eigenvalues of  $\sqrt{\Delta} \mathbb{I}^{-1} \nabla^2 \phi$ , as shown in eq. (3.14). The surface element in  $xy$  coordinates is  $\sqrt{\Delta} dx dy$ . We therefore have the following version of



**Figure 5.6** – *Workflow for optimizing structures.* Given a boundary curve, we compute the optimal stress potential  $\phi(x, y)$  and design surface  $\mathcal{M}(x, y)$  shown in (a). Optimization in particular makes the principal stress directions coincide with the principal curvature directions (b), so a quad mesh which follows these directions has approximately planar faces and is optimally placed to carry the flow of forces (c). The measure of planarity of individual quads given here is the ratio of distance of diagonals, over average edge length. Optimization towards planarity and equilibrium does not change the mesh much (d), verifying that the “continuum” version of the optimization has been accurate. Finally we construct a structure following the mesh, connecting members with rigid joints. Finite element analysis shows both tension and compression in its members (e).

eq. (5.13) in height field parametrization:

$$\iint (|\lambda_1| + |\lambda_2|) \Delta \, dx \, dy \rightarrow \min, \quad (5.14)$$

where  $\lambda_1, \lambda_2$  are the eigenvalues of  $\mathbb{I}^{-1} \nabla^2 \phi$ .

*Remark.* If eq. (5.14) had  $\sqrt{\Delta}$  instead of  $\Delta$ , then it would express minimization of a certain total absolute curvature, similar to the 2D case. This is because  $\lambda_1, \lambda_2$  are the  $i$ -curvatures of the Airy surface  $z = \phi(x, y)$  w.r.t. the first fundamental form of the design surface.

### 5.5.3 Computing optimal structures – the workflow

The preparations above enable us to compute optimal discrete structures which have the shape of shells, see fig. 5.6. This procedure is similar to the one presented in section 5.4.3 for the 2D case. Note that the combinatorics of the structure is part of the solution.

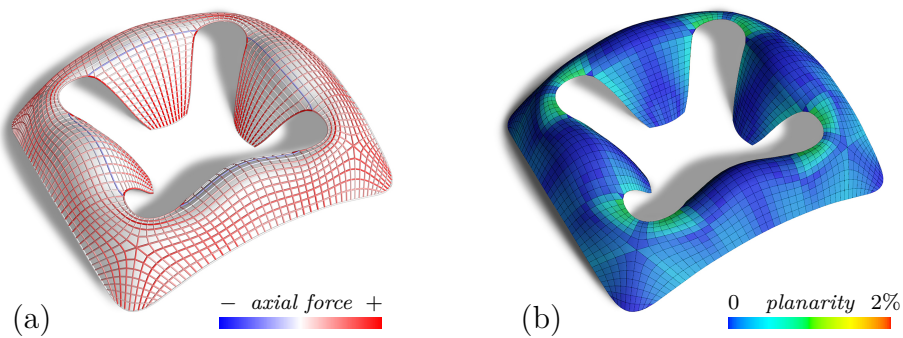
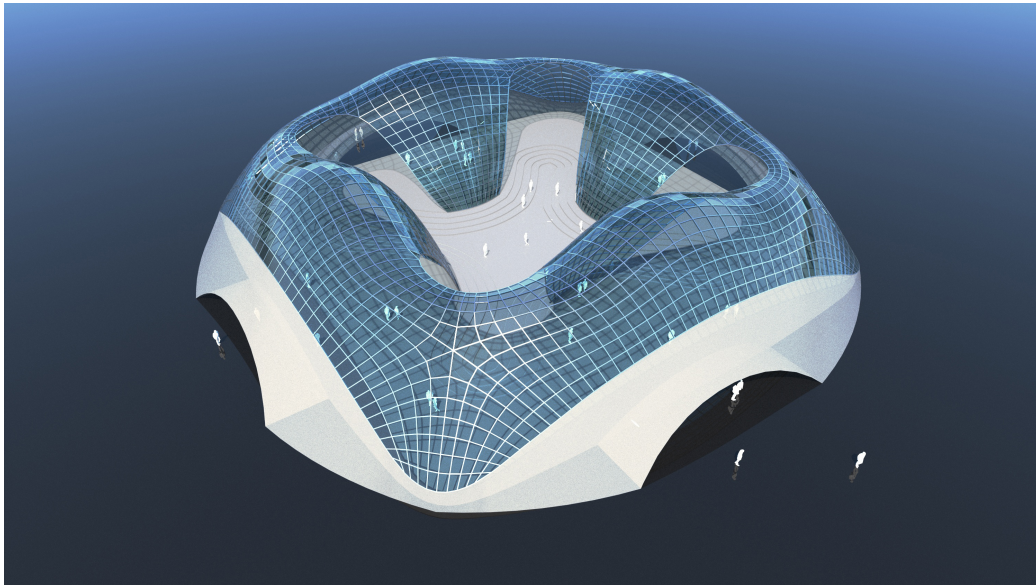
We solve the following problem: *given boundary conditions, find a shell-*

like truss which (i) balances self-loads and exterior boundary forces; (ii) observes geometric boundary conditions; and (iii) is volume-optimal. The procedure consists of the following steps:

- *Step 1: Form and force optimization.* We set up an optimization problem for an Airy potential function  $\phi(x, y)$  and a design surface  $z = s(x, y)$ . The optimization target is eq. (5.14). Constraints include the user-defined ones, and the ones arising from the relations between  $\phi$  and  $s$ . In addition we require that the principal stress directions are conjugate w.r.t. the design surface — otherwise the faces of the final truss cannot be planar. Numerical optimization is performed over a suitable triangulation which is unrelated to the optimal truss. It requires an elaborate setup of variables and constraints, see section 5.6.1.
- *Step 2: Meshing.* Using the *mixed integer quadrangulation* method of Bommès et al. [13], we extract a quad mesh which follows the cross field of principal directions in the design surface.
- *Step 3: Post-Processing.* A round of optimization makes faces planar and puts forces into equilibrium, which is exactly the problem solved by Tang et al. [55]. Postprocessing may of course vary from example to example. Finally, an engineering task is performed: an actual structure is planned on basis of that mesh. The cross-sections of members are chosen on basis of the forces we computed, such that the member has minimal weight (taking a safety margin into account). Joints and foundations are modeled, and the structure can undergo to FEM testing for standard load cases (not treated in this thesis). Note that the members' weight influences the load, so these members must be implicitly present already during Step 1.

## 5.6 Implementation

This section describes in detail the first stage in the workflow for computing optimal structures, namely finding an optimal design surface  $\mathcal{M} = z(x, y)$



**Figure 5.7** – *Boundary conditions.* Parts of the boundary of this steel-glass structure are physically supported, others are free. During optimization, however, all boundaries have prescribed  $x, y$  coordinates in order to preserve the design intent. (a) shows tension and compression in the final structure, and (b) planarity of faces.

together with its stress potential  $\phi(x, y)$ . The 2D case is treated briefly afterwards.

### 5.6.1 Variables and constraints for optimal gridshells

The design surface  $\mathcal{M}$  is parametrized as a height field  $z(x, y)$  (see section 2.2.3) over a planar domain which is given as a triangle mesh. Each

vertex is associated with a coordinate vector  $\bar{\mathbf{v}} = (v_1, v_2) \in \mathbb{R}^2$  (fixed), the design surface's  $z$  coordinate  $v_3 = z(\bar{\mathbf{v}})$  which is subject to optimization, and the following additional variables.

- gradient  $\nabla z$  and Hessian  $\nabla^2 z$  (5 variables)
- 2nd derivatives  $\phi_{,xx}, \phi_{,xy}, \phi_{,yy}$  of the Airy potential, and their derivatives  $\phi_{,xxx}, \phi_{,xxy}, \dots$  (9 variables)
- matrix  $\mathbb{I}$ , values  $\Delta = \det(\mathbb{I})$ ,  $\delta = \sqrt{\Delta}$ ,  $\omega = \sqrt{\delta}$  (6 variables)
- eigenvalues  $\lambda_1, \lambda_2$  of  $\mathbb{I}^{-1}\nabla^2\phi$ , their product  $\lambda$ , and principal stresses  $\sigma_1, \sigma_2$  (5 variables)
- eigenvectors  $\bar{\mathbf{a}}_j$  and vectors  $\bar{\mathbf{b}}_j, \bar{\mathbf{c}}_j$  described below (12 variables).

We also have the following constants:

- panel weight  $\rho$  in  $\text{kg/m}^3$  and panel thickness  $h$ .
- The future members of the truss are fully stressed, so their weight can already be computed while we are still optimizing the surface, using eq. (5.14): The weight per  $xy$  area element is  $\mu(|\lambda_1| + |\lambda_2|)\Delta$ , where  $\mu$  is a constant of dimension  $\text{m}^{-1}$ .

The variables introduced above obey constraints, starting with relations between derivatives and function values. For any of the functions  $\psi \in \{\phi_{,xx}, \phi_{,xy}, \phi_{,yy}\}$  the first derivatives are variables in the optimization, and for  $z$  we even involve second derivatives. For each pair  $\bar{\mathbf{v}}\bar{\mathbf{w}}$  of adjacent vertices, we require

$$\begin{aligned} z(\bar{\mathbf{w}}) - z(\bar{\mathbf{v}}) &= \nabla z(\bar{\mathbf{v}})^T \cdot (\bar{\mathbf{w}} - \bar{\mathbf{v}}) + \frac{1}{2}(\bar{\mathbf{w}} - \bar{\mathbf{v}})^T \cdot \nabla^2 z(\bar{\mathbf{v}}) \cdot (\bar{\mathbf{w}} - \bar{\mathbf{v}}), \\ \psi(\bar{\mathbf{w}}) - \psi(\bar{\mathbf{v}}) &= \nabla\psi(\bar{\mathbf{v}})^T \cdot (\bar{\mathbf{w}} - \bar{\mathbf{v}}) \quad , \quad \text{for } \psi \in \{\phi_{,xx}, \phi_{,xy}, \phi_{,yy}\}. \end{aligned}$$

As  $\bar{\mathbf{w}}$  runs through the 1-ring or 2-ring neighbourhood of  $\bar{\mathbf{v}}$  (depending on whether we want to compute 1st or 1st + 2nd derivatives) the above equations constitute an over-determined linear system of the form  $H\mathbf{x} = \mathbf{r}$  for the vector  $\mathbf{x}$  of derivatives at  $\bar{\mathbf{v}}$ . In guided projection, it is solved by minimizing  $\|H\mathbf{x} - \mathbf{r}\|^2$ , see eq. (4.6). The derivatives of this expression w.r.t. both  $\mathbf{x}$  and  $\mathbf{r}$  must vanish, since both  $\mathbf{x}$ ,  $\mathbf{r}$  are variables in our optimization. This leads

to

$$H^T H \mathbf{x} = H^T \mathbf{r}, \quad H \mathbf{x} = \mathbf{r}. \quad (5.15)$$

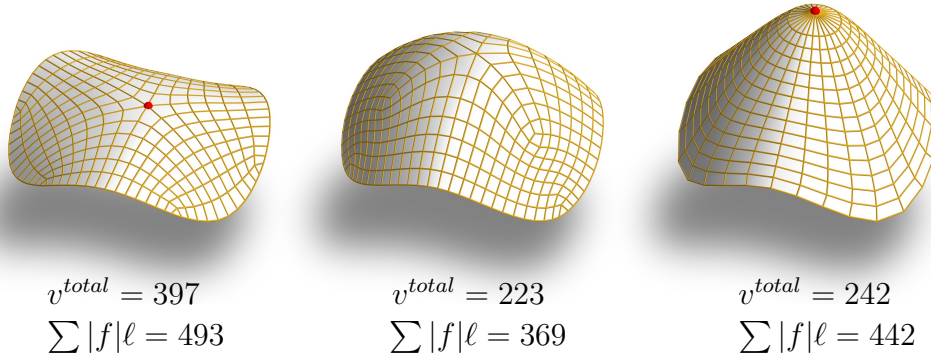
The size of  $H$  depends on the size of the neighbourhood, but for the purpose of counting the number of constraints vs. the number of variables, we consider the balance even. The constraint sets  $C_1$  and  $C_2$  obtained in this way refer to derivatives of design surface and stress surface, resp. Further, for each vertex we have the following sets  $C_3, \dots, C_8$  of linear or quadratic constraints:

- $C_3$  Derivatives obey  $\phi_{,xxy} = \phi_{,xyx}$  and  $\phi_{,xyy} = \phi_{,yyx}$ .
- $C_4$  Equations (2.8), (2.9) define  $\mathbb{I}$ ,  $\Delta$ . We require  $\delta^2 = \Delta$ ,  $\omega^2 = \delta$  in order to ensure a positive square root of  $\Delta$  (6 constraints).
- $C_5$  The defining relations  $\lambda_1 + \lambda_2 = \text{tr}(\mathbb{I}^{-1} \nabla^2 \phi)$ ,  $\lambda_1 \lambda_2 = \det \nabla^2 \phi / \Delta$  are made quadratic by multiplication of both l.h.s. and r.h.s. with  $\Delta$ , and the substitution  $\lambda = \lambda_1 \lambda_2$ . Stresses are defined by  $\delta \sigma_i = \lambda_i$ ,  $i = 1, 2$  (5 constraints).
- $C_6$  The defining relations  $\nabla^2 \phi \cdot \bar{\mathbf{a}}_i = \lambda_i \mathbb{I} \bar{\mathbf{a}}_i$  of the principal stress directions are made quadratic by the substitution  $\mathbb{I} \bar{\mathbf{a}}_i = \bar{\mathbf{b}}_i$ . We further require  $\bar{\mathbf{a}}_i^T \bar{\mathbf{a}}_i = 1$  and  $\bar{\mathbf{a}}_1^T \bar{\mathbf{b}}_2 = \bar{\mathbf{a}}_2^T \bar{\mathbf{b}}_1 = 0$  to be sure to pick an orthonormal basis of eigenvectors.
- $C_7$  Equation (2.11), substituting  $\bar{\mathbf{c}}_i = \nabla^2 z \bar{\mathbf{a}}_i$ , makes 5 constraints.

Note that  $|C_6| = 5$ : Computing eigenvectors is 2 equations, normalization amounts to another 2, and orthogonality constraints for eigenvectors can be omitted in a d.o.f. count. Constraint set  $C_8$  consists of the vertical equilibrium equation given by eq. (3.13). The load per unit  $xy$  area  $\bar{\rho}$  is the structure's weight plus the panel weight, leading to  $\bar{\rho} = \mu(|\lambda_1| + |\lambda_2|)\Delta + \sqrt{\Delta}\rho h$ . This equation involves absolute values and is approached by the method of iteratively reweighted least squares: at iteration level  $k$ ,  $|\lambda_i|$  is replaced by

$$|\lambda_i| = w_{i,k} \lambda_i^2, \quad \text{where } w_{i,k} = 1 / \sqrt{(\lambda_{i,k-1})^2 + \epsilon}. \quad (5.16)$$

Here  $\epsilon > 0$  is a small regularizer, and  $\lambda_{1,k-1}, \lambda_{2,k-1}$  are constants equal to the value of  $\lambda_1, \lambda_2$  in the previous iteration. With this substitution, the



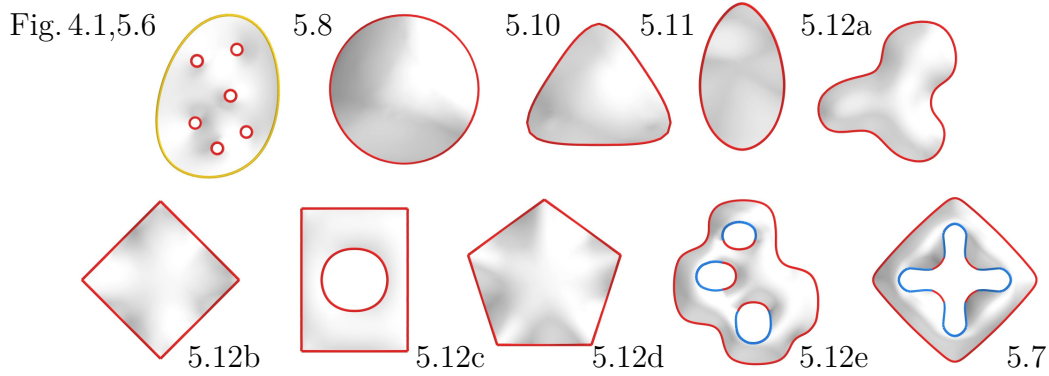
**Figure 5.8** – *User interaction.* We show optimal quad meshes derived without additional constraints (center), and under the constraint that a handle point lies lower (left), resp. higher (right). The effect is demonstrated by means of the value  $v^{total}$  of the target functional, and the value  $\sum |f|_\ell$  of the final structure. In all three cases, FEM analysis confirms that more than 90% of the elastic energy (i.e., the potential energy stored in the structure as the deadload is applied, in the standard load case, with the same boundary conditions as used during optimization) is due to axial forces.

equilibrium equation becomes quadratic. This concludes the enumeration of constraints per vertex.

The  $x, y$  coordinates of all vertices are fixed. The coordinates  $z(\bar{\mathbf{v}})$  of vertices are typically free. At the boundary, we can opt for fixed or for free  $z$  coordinates. A different property of a boundary vertex is if it is physically supported or free, see e.g. fig. 5.7. An unsupported part of the membrane's boundary experiences no external forces except the deadload. The corresponding boundary part of the Airy potential lies in a plane [35], but in our optimization we compute with eq. (3.11). Figure 5.9 gives an overview of the different boundary conditions we use.

### 5.6.2 Counting degrees of freedom

Let us first discuss the continuous case in a naive way. If the design surface  $z(x, y)$  is given, the stress potential  $\phi(x, y)$  is found by the equilibrium condition (3.10), which is a 2nd order PDE. It is elliptic in case stresses are all compressive [56], so  $\phi$  is uniquely determined by boundary values. Even if we have a mixture of tensile and compressive stresses we expect the same behaviour. The requirement that principal stress directions agree with prin-



**Figure 5.9** – Boundary conditions imposed on the examples contained in this chapter. The manner of condition along the boundary is indicated by colors: red is physically supported with fixed  $z$  coordinate, yellow is supported with free  $z$  coordinate, blue is unsupported with free  $z$  coordinate.

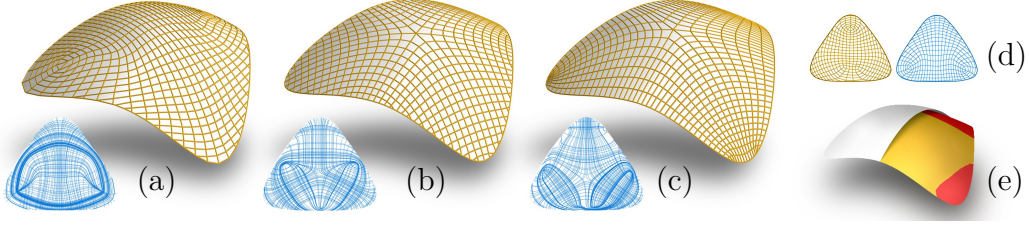
principal curvature directions is again a 2nd order PDE, so we can expect that both  $z$  and  $\phi$  are determined by boundary values. Any further constraint, like requiring that a boundary is a principal curve, leads to loss of freedom for the boundary of the design surface. For volume optimization not much freedom is left now – we basically only have the boundary values of  $\phi$  at our disposal.

In the discrete setting we have one constraint per vertex less than there are variables, if we disregard eq. (2.11) which enforces alignment of principal stress directions with principal curvature directions. This is in accordance with the fact that for a given design surface, the stress potential is determined. Enforcing eq. (2.11) leads to as many equations as there are variables, implying that no degrees of freedom are left. This is in accordance with the shape restriction we already saw in the continuous case, and it means that enforcing this condition makes our optimization procedure a tool for form-finding.

### 5.6.3 Target functional for optimization

It is not difficult to express the target functional of eq. (5.14) in terms of our variables. Recall that we work with a triangulation of a planar domain.





**Figure 5.10** – *Comparing optimization with and without alignment constraints.* Optimization and quad meshing guided by principal stress directions (inset figures) yields different meshes, if optimization is performed subject to different constraints: (a) without alignment of principal stresses and principal curvatures, (b) with this alignment, using eq. (2.11) (c) in addition, alignment of principal curves with the boundary. Subfigure (d) shows the discrepancy between principal stress directions and principal curvature directions in the first case, and subfigure (e) shows the 3 design surfaces superimposed (a = white, b = red, c = yellow), from which we conclude that the difference in these three meshes is mainly the combinatorics, not the geometric shape.

Each vertex  $\bar{\mathbf{v}}$  has an area of influence  $A(\bar{\mathbf{v}})$ , which is computed as one third of the area of its vertex star. In the  $k$ -th round of our iterative optimization procedure, volume minimization is then expressed by

$$\begin{aligned} \iint (|\lambda_1| + |\lambda_2|) \Delta \, dx \, dy &\approx \sum_{\text{vertices}} (w_1^k \lambda_1^2 + w_2^k \lambda_2^2) \Delta A \\ &= \sum_{\text{vertices}} (w_1^k \sigma_1^2 + w_2^k \sigma_2^2) A \rightarrow \min. \end{aligned} \quad (5.17)$$

All expressions are values associated with vertices; recall the weights  $w_i^k$  from eq. (5.16). This is a quadratic objective function.

#### 5.6.4 Further constraints

Several of the results shown in this paper add more constraints to the optimization, like handles operated by the designer (fig. 5.8), or proximity to a reference surface, which can be treated as quadratic constraints [55]. A particular constraint highly relevant to architectural design is alignment of the mesh with the given boundary (see examples in figs. 5.6 to 5.8 and the comparison in fig. 5.10). With  $\bar{\mathbf{t}}$  as a (projected) tangent vector at the boundary, and  $\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2$  as the principal stress vectors, constraint set  $C_7$  is augmented

by  $(\bar{\mathbf{t}}^T \mathbb{I} \bar{\mathbf{a}}_1) \cdot (\bar{\mathbf{t}}^T \mathbb{I} \bar{\mathbf{a}}_2) = 0$ . Unfortunately this constraint can almost never be fulfilled exactly — a closed curve (like the boundary) can be a principal curvature line only if the integral of torsion is an integer multiple of  $2\pi$ . This can be achieved if the  $z$  coordinates of boundary vertices are free variables in our optimization, but otherwise this obstruction has the effect that near the boundary, the angles in our mesh deviate from  $90^\circ$ , see figs. 5.7 and 5.12b, and 5.12e.

### 5.6.5 Solver

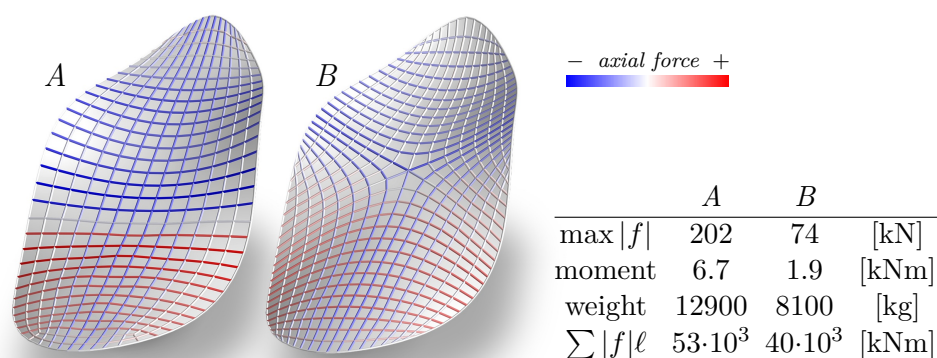
The target functional eq. (5.14) resp. eq. (5.17) is optimized by the constraint solver by Tang et al. [55], as for section 4.4.2. The method tries to move the vector of variables towards the constraint manifold, such that the path is guided by a quadratic energy. In this case, the energy is provided by the quadratic objective function of eq. (5.17). Statistics are shown by table 5.1.

### 5.6.6 Variables and constraints for optimal 2D trusses

The procedure for optimizing shell-like trusses described in section 5.6.1 specializes to the 2D case if we let  $s$  and its derivatives equal zero, and let the self-weight be zero. Then  $\mathbb{I}$  becomes the unit matrix, and  $\Delta = \delta = \omega = 1$ . The eigenvalues  $\lambda_i$  become the curvatures  $\kappa_i$  needed for the target functional. In this way, many variables and constraints disappear. Further we can drop the conjugacy condition of eq. (2.11). The rest is unchanged, and we do not elaborate further.

## 5.7 Discussion

We start the discussion with several different instances of the optimization procedure proposed in this chapter. Figure 5.10 compares optimization with and without enforcing eq. (2.11), i.e., disregarding alignment of principal stress directions with principal curvature directions. It is encouraging to see that the design surface is almost unchanged, see superimposed surfaces in fig. 5.10e. We see that eq. (2.11) does not prevent optimization for minimal

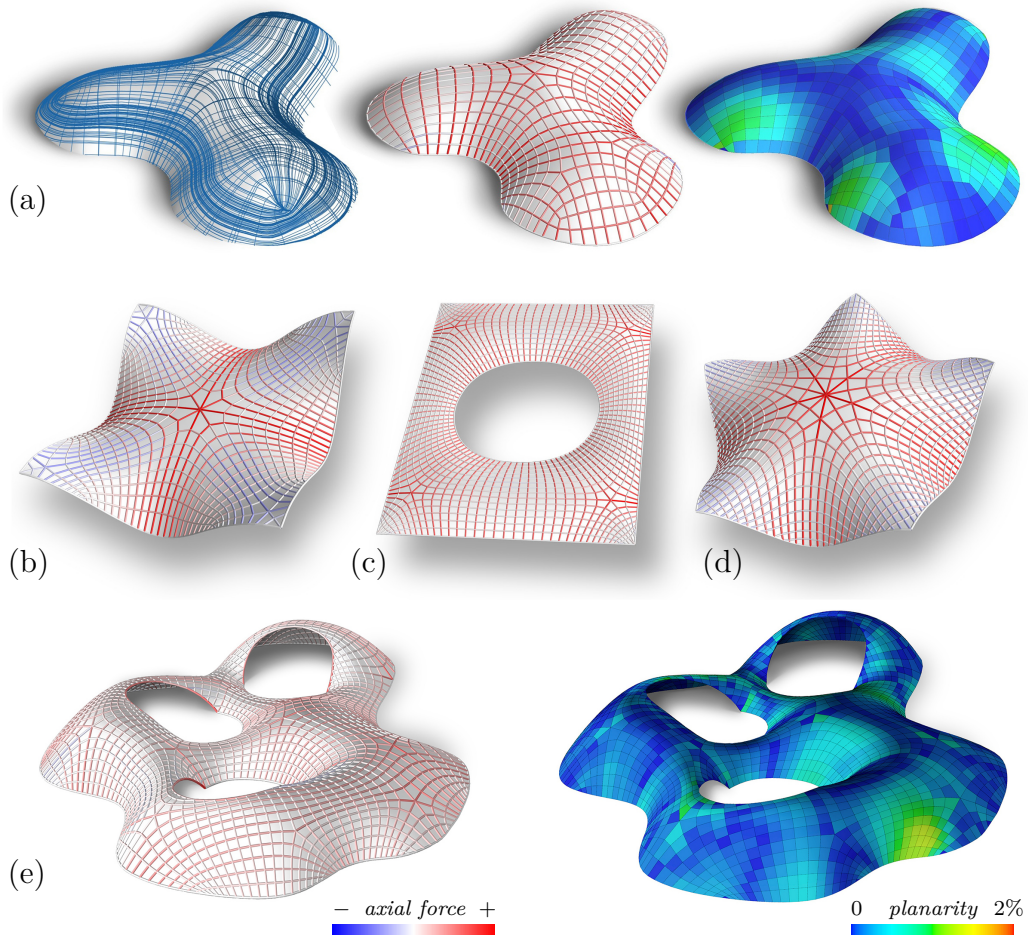


**Figure 5.11 – Effectiveness.** We compare meshes with the same boundary: Mesh A is optimized according to section 5.6.1, while mesh B is computed by minimizing a fairness energy and subsequent quad meshing. They undergo postprocessing by optimization for equilibrium and planar faces according to [55], and subsequent modeling of a structure with rigid joints. The members’ cross-sections are selected according to maximum allowable stresses and maximum allowable displacement. Then structure B is 60% heavier than structure A, and the effect is even bigger if we skip the maximum displacement requirement. The underlying surfaces are very similar, confirming the big influence of the structure’s combinatorics (and the positioning of members) on the weight.

weight.

### 5.7.1 Verification of results

The success of our optimization procedures is measured in different ways. Firstly, measuring planarity of faces is straightforward (see individual images). Secondly, our working assumption to treat shells as membranes is verified a posteriori by FEM analysis of structures with rigid joints based on the meshes we generate: in our examples, more than 90% of the elastic energy in the basic load case is due to axial forces. Thirdly, we compare structures derived from optimized meshes with structures derived from meshes obtained in another way, see fig. 5.11. Numerical experiments confirm that in non-optimized meshes, axial forces typically are 2–3 times larger, and the weight of the structure is about 50% bigger. These experiments measure the effect of our optimization. Another comparison is shown by Figure 5.8, where we demonstrate how a user’s interference with free optimization causes the



**Figure 5.12** – Examples of optimal structures, (a)–(e). We show tensile and compressive forces in structures derived from optimized meshes. For example (a) also the optimal shape with principal stress lines is shown (left). For examples (a) and (e) the planarity of the quad mesh derived from is depicted (right). All structures except (e) have been optimized with the additional constraint of alignment with the fixed boundary. We hit the “integral of torsion” obstruction mentioned in section 5.6.1, so the angle between edges visibly deviates from 90 degrees.

weight of the final structure to increase.

## 5.7.2 Implementation details

Table 5.1 gives details on computation times for the optimization process described by section 5.6.1. For meshing and post-meshing optimization, which

are no contributions of this thesis, we refer to [13] and [55], respectively. The times given are for an Intel Core i5-6500 CPU with 3.20GHz and 15.6 GB memory.

A word about the nature of our optimization procedure is in order. The guided projection method of Tang et al. [55] does not exactly *minimize* the target functional, but use this functional to guide a mesh towards the solution manifold defined by the constraints (one could achieve minimization by adjusting weights as the iteration progresses). Table 5.2 shows comparisons with a constrained solver. Numerical experiments show that apart from some outliers, already after 10 iterations our method minimizes the target within a margin of 10–12%, but computation is faster by orders of magnitude. We stayed with our method because we expect that in an architectural design situation, exact minimization does not have priority. The reason for this is not only the desire for design freedom, but also the additional constraints and safety requirements an actual structure is required to conform to, and which cause some discrepancy between the optimized mesh and the structure based on it (see e.g. maximum displacement constraint in fig. 5.11).

### 5.7.3 Robustness

When computing with curvatures (and generally, higher derivatives), robustness is always an issue. We should add that we did not experience problems here, since the principal directions of our shapes are variables in our optimization procedure and are thus automatically subject to its regularizing effect.

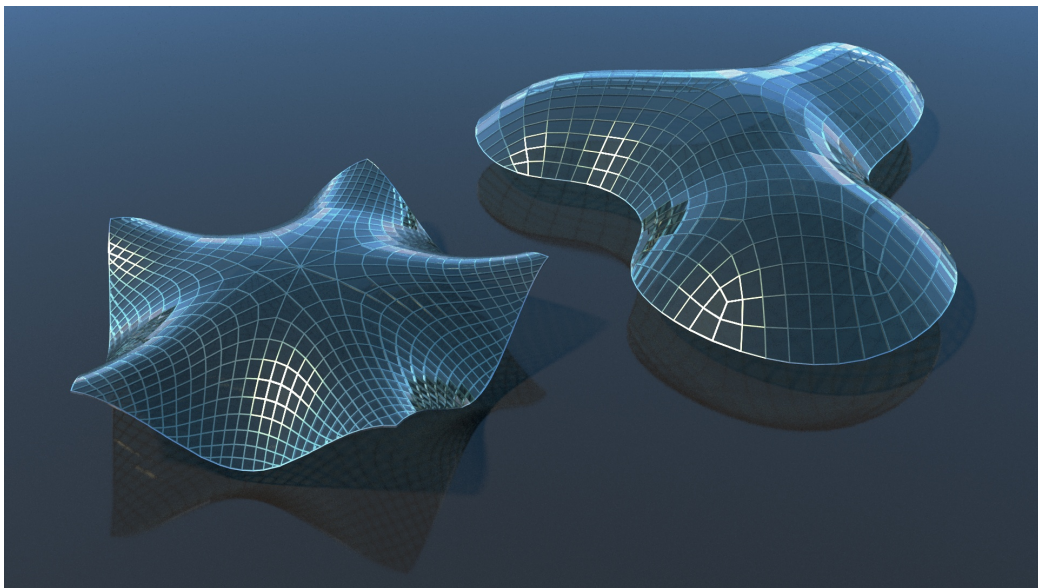
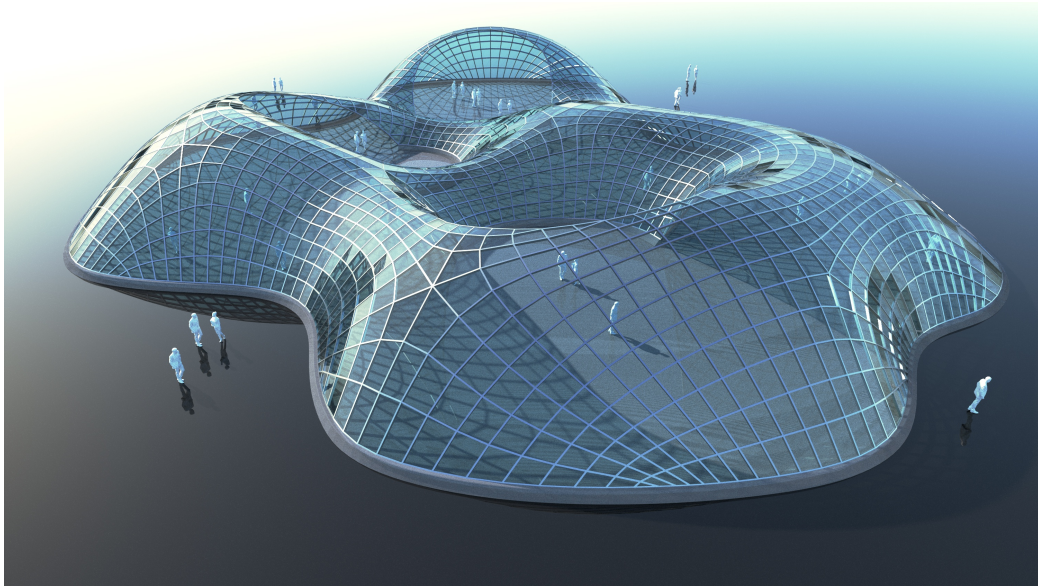
Fig.	5.6	5.7	5.11	5.12a	5.12b	5.12c	5.12d	5.12e
$ V $	1082	1865	882	678	673	973	1133	2003
$ F $	2026	3403	1662	1254	1252	1772	2149	3712
# of variables	41 k	89 k	33 k	25 k	25k	36 k	16k	96 k
# of constraints	73 k	228 k	68 k	51 k	51k	73 k	32k	238 k
time [sec]	325	265	16	35	19	60	42	338
<i>weights given to individual constraints, ordered by constraint set</i>								
target functional	$10^{-5}$	$10^{-8}$	$10^{-4}$	$10^{-4}$	$10^{-4}$	$10^{-2}$	$10^{-3}$	$10^{-6}$
$C_1, C_2, C_3$	400,10 <sup>2</sup> ,40	1,1,1	2,1,1	2,1,1	1,1,1	2,2,2	1,1,1	1,1,1.4
$C_4, C_5, C_8$	10 <sup>2</sup> ,10 <sup>3</sup> ,200	1,1,2.4	2,1,1	2,1,1	1,1,1	2,2,2	1,1,1	1,1,1.4
$C_6, C_7$	200,1	1.4,.01	1,1	1,1	1,1	2,1	1,1	1.4,.01
<i>RMS residuals of constraints, ordered by constraint set</i>								
$C_1$ (derivatives $s$ )	.017	.081	.005	.006	.014	.004	.007	.081
$C_2$ (derivatives $\phi$ )	.019	.015	.009	.010	.013	.002	.005	.017
$C_3$ ( $\partial_{xy} = \partial_{yx}$ )	.021	.005	.001	.003	.002	.000	.001	.005
$C_4$ ( $\mathbb{I}, \Delta, \dots$ )	.006	.006	.001	.001	.004	.001	.001	.001
$C_5$ (eigenvalues)	.012	.001	.000	.010	.011	.000	.000	.012
$C_6$ (eigenvectors)	.002	.730	.002	.004	.003	.001	.001	.005
$C_7$ (alignment)	.039	.012	.004	.009	.003	.003	.001	.416
$C_8$ (equilibrium)	.016	.005	.003	.008	.005	.000	.001	.019

**Table 5.1** – Details concerning form+stress optimization on a triangle mesh  $(V, E, F)$ . The large number of constraints appears to contradict section 5.6.1, but is due to the many equations of type eq. (5.15). The weights given to constraints and the target functional refer to the method of Tang et al. [55]

Random experiment	GP <sub>1</sub>	I <sub>1</sub>	GP <sub>2</sub>	I <sub>2</sub>	GP <sub>3</sub>	I <sub>3</sub>	GP <sub>4</sub>	I <sub>4</sub>
time [sec]	0.18	145	0.14	101	0.15	160	0.14	139
target functional	10.8	9.75	12.7	11.4	13.3	11.8	6.62	5.90
distance	0.03		0.03		0.04		0.02	

**Table 5.2** – We compare our “guided projection” method with the constrained solver of [57]. Columns “GP” and “I” refer to 10 rounds of either method. We perform optimization according to section 5.6.6 on four meshes with 1600 vertices and randomly generated boundary (9444 variables, 8000 constraints). We show runtime, the value of the target functional, and the distance between the resulting surfaces. The final designs achieved with the two methods are very different.





**Figure 5.13** – Architectural applications of the examples of fig. 5.12d,a (bottom) and e (bottom). Planarity of panels allows the use of flat glass panels.

# Chapter 6

## Visual smoothness of meshes

The discretization of architectural freeforms with flat panels is particularly convenient for manufacturing. However, a polyhedral surface is not the best solution when coming to visual appearance, especially when dealing with reflective materials such as glass or metal. In this chapter, it is shown how the reflective quality of a mesh can to some extent be expressed by the requirement that its dihedral angles are small: small angles yield “softer” reflections, enhancing the effect of continuity. It turns out that, among all possible meshes discretizing a smooth surface, principal meshes are the ones with the smallest angles and then with the best visual appearance. This result is fortunate since principal meshes, besides having flat panels and torsion-free joints, can also reduce the use of structural material, as shown in chapter 5.

Aside from discretization of a given surface, the minimization of dihedral angles can be used to find meshes with the best visual appearance inside a given boundary, turning into a form-finding tool. It turns out that such meshes are discretizations of principal curvature networks of surfaces where the sum of absolute principal curvatures is minimal. Curvature plays here the role that stress had for volume-optimal gridshells in chapter 5.



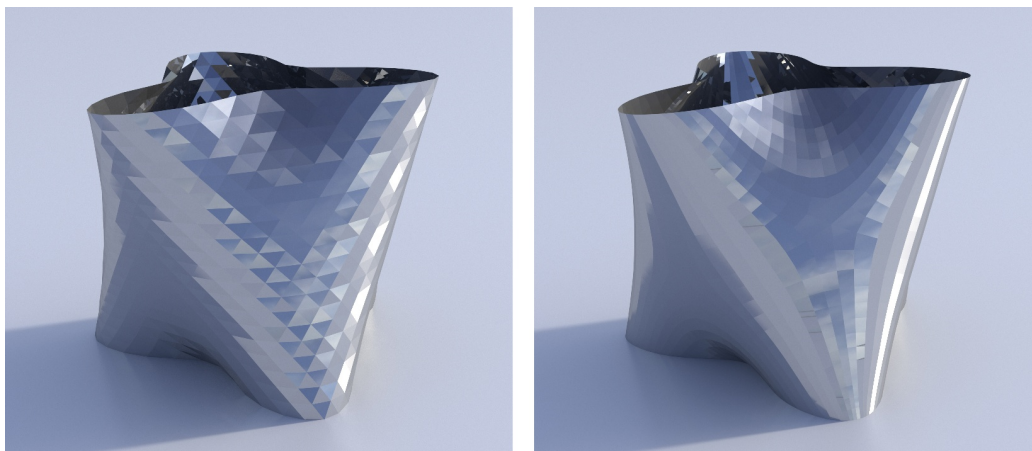
## 6.1 Overview and contribution

After an overview on previous work, this chapter starts in section 6.3 with a discussion of the inevitable deficiencies of discrete surfaces which are meant to approximate smooth surfaces. We identify the shape of the normal pyramid as a key difference between the discrete case and the smooth case. We propose that a mesh faithfully models a smooth surface if, in negatively curved areas, it exhibits so-called *good saddles*. We go on to show that then normal pyramids are free of self-intersections. This is a criterion for visual smoothness of the mesh (in particular it implies small dihedral angles).

To quantify the visual smoothness of a mesh, in section 6.4 we turn to a mesh energy measuring the total variation of the normal vectors. It essentially is the sum of edge lengths times dihedral angles. A small energy implies that normal pyramids cannot be too badly behaved. A thorough discussion of properties of this energy reveals that for a given geometric shape, the mesh with smallest energy is no triangle mesh, but a quad mesh aligned with principal curvature lines. We prove this fact in a similar way as we did in section 5.3. This result is fortunate, because principal meshes are relevant for freeform architecture for several reasons, as seen in chapters 2 and 5. It is interesting to know that they have good properties also from the viewpoint of visual appearance.

It turns out that among all meshes approximating a given smooth reference surface, the lowest achievable energy equals a certain curvature measure which we call *total absolute curvature*. It has an interpretation as total variation of the surface's normal vector field. If that quantity is small, the surfaces have especially good visual smoothness when represented by a mesh.

Finally, in section 6.5 we show the computational framework for finding surfaces of minimal total absolute curvature for given boundary. Thanks to a principal quad remeshing, on such surfaces we can achieve the shape and connectivity of the “smoothest” mesh over a given boundary. This chapter presents results published in [40].



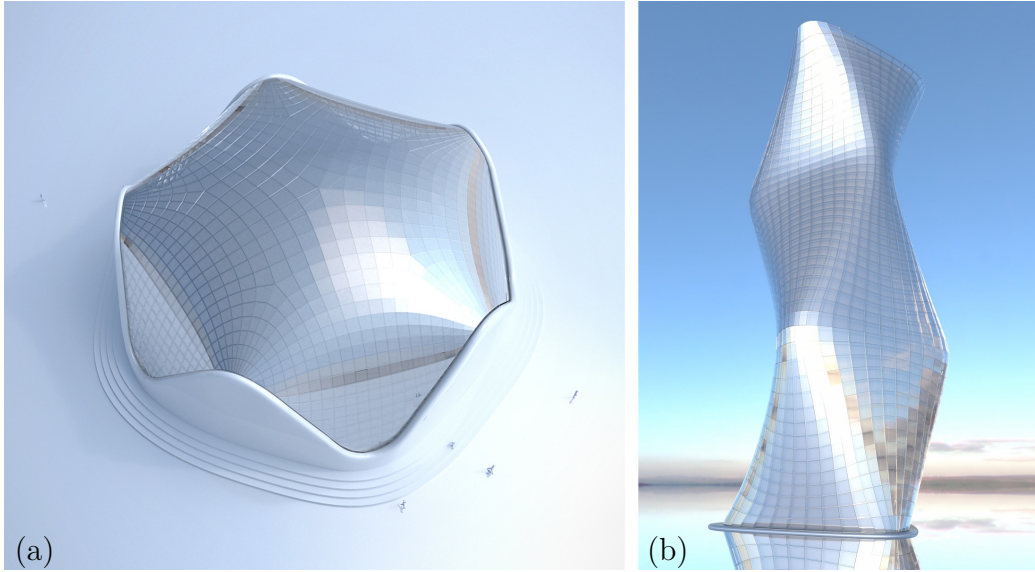
**Figure 6.1** – A reflective surface reveals the deficiencies in mesh fairness which we quantify by means of a discrete fairness energy based on edge lengths and dihedral angles. This energy is minimal for principal quad meshes (right). In this special case both meshes approximate a reference surface of small total variation of the normal vector field, so all visible deficiencies are not caused by the reference shape, but by the way this shape is meshed.

## 6.2 Previous work

There has been systematic work on meshes which approximate smooth surfaces, and criteria which ensure that discrete-differential quantities derived from meshes approximate their smooth counterparts, see [7, 24]. This question is also going to be relevant in this work.

Smoothness of polyhedral surfaces in our narrower sense revolves around the behaviour of the *Gauss image* of the mesh, i.e., the face normals. Günther et al. [22] investigate in detail properties of vertex stars and Gauss images, establishing relations between the local shape of the mesh and the question of self-intersections of the Gauss image. On that basis, Jiang et al. [26] discuss optimization of meshes towards star-shaped Gauss images (which implies absence of self-intersections). Good representation of saddles is already implicitly present in their work, but is only expressed in terms of relative position of edges to a smooth reference surface.

It is worth noting that the relation between mesh and Gauss image employed in this chapter has a 2-dimensional analogy, namely the relation between a 2D triangulation and a reciprocal-dual mesh. Here Orden et al. [38]



**Figure 6.2** – Two-step architectural designs. Firstly we find surfaces which minimize total absolute curvature, secondly we mesh these surfaces along principal curves, making the energy minimal.

characterize pseudo-triangulations whose reciprocals have no self intersections.

Imposing conditions on the normal pyramid has been successfully done before: the “hinge condition” of Stein et al. [50] corresponds to developability of surfaces.

To achieve fairness (in our special sense) of meshes, we work with a mesh fairness energy. It falls into a broader class of energies of the form  $\sum f(\ell_e, \alpha_e)$ , where the sum is taken over all edges of a mesh and  $f$  is a function of edge length  $\ell_e$  and dihedral angle  $\alpha_e$ , see [54] for an overview. We employ the energy  $\mathcal{E} = 2 \sum \ell_e |\sin \frac{\alpha_e}{2}|$  which approximates the energy  $\mathcal{E}' = \sum \ell_e |\alpha_e|$ . Both are a kind of bending energy.  $\mathcal{E}'$  has been used for the purpose of optimizing triangulations of surfaces (see [2] and follow-up papers), and also of volumes [19]. Its usage for formfinding was proposed in [21].

This work also involves fairness functionals operating on surfaces. These are frequently defined in terms of curvatures. We employ the *total absolute curvature* defined as the surface integral  $\int |\kappa_1| + |\kappa_2|$ , where  $\kappa_1, \kappa_2$  refers to principal curvatures. Several deep theorems concern the gradient flow of such functionals. Well-studied examples include the surface area functional,

and also the total mean curvature  $\int \frac{\kappa_1 + \kappa_2}{2}$ . Moving the surface with speed proportional to mean curvature  $\frac{\kappa_1 + \kappa_2}{2}$  resp. Gauss curvature  $\kappa_1 \kappa_2$  corresponds to  $L^2$ -steepest descent of surface area resp. total mean curvature. It is known that in both cases, convex surfaces flow to round points [3, 25]. Flows used for geometry processing purposes include the gradient flow of Willmore energy  $\int (\kappa_1 - \kappa_2)^2$ , see [12] for a mesh version. A sign-corrected Gauss curvature flow has been used by [59].

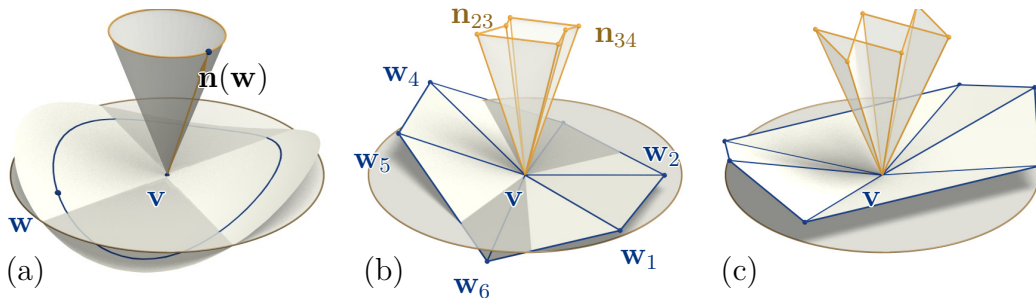
In the following, we do not use the gradient flow for minimizing total absolute curvature, but a method very similar to the one of [28], presented in chapter 5.

A mesh energy may or may not be a discretization of a surface fairness functional. It is well known that discrete mean curvature  $\frac{1}{2} \sum \ell_e \alpha_e$  is a discretization of total mean curvature  $\int \frac{\kappa_1 + \kappa_2}{2}$ , see [16]. It must be emphasized that our energies  $\mathcal{E}, \mathcal{E}'$  do not enjoy such a property for general meshes, despite being called total “absolute mean curvature” in the literature.

Returning to curvature-based functionals, both our total absolute curvature  $\int |\kappa_1| + |\kappa_2|$  and the functional  $\int (\kappa_1^2 + \kappa_2^2)^{1/2}$  have an interpretation as *total variation* of the normal vector field. The latter functional is studied in detail in a recent preprint by Bergmann et al. [8]. They also discuss the energies  $\mathcal{E}, \mathcal{E}'$  and touch upon several of the presented topics. Energy-minimal meshes (considered in this chapter) are mentioned as an unsolved problem.

### 6.3 Representation of saddles in meshes

Experience shows that visible deficiencies in fairness of polyhedral meshes occur mainly in negatively curved regions, i.e., parts of the mesh where the underlying smooth reference surface is locally saddle-shaped. The visual appearance of the mesh depends heavily on the normal vectors of faces – fig. 6.1 shows how a reflective surface reveals the normal vectors high variation. For this reason we have a closer look at the immediate neighbourhood of a vertex and visualize the normal vectors. fig. 6.3 illustrates a fact which we feel has not been sufficiently appreciated so far: there is a big difference between the local shape of a smooth surface and the local shape of a mesh. In particular the normal vectors of faces in the immediate neighbourhood of a vertex



**Figure 6.3** – *Discrepancy between smooth surfaces and discrete surfaces.* (a) When circling a point  $v$  of a smooth surface  $\Phi$ , the normal vector describes a convex cone. (b) In a polyhedral surface, the normal vectors of a vertex star form a pyramid. Deviating from the smooth situation, this pyramid is typically non-convex in case of negative curvature. (c) Normal pyramids can even self-intersect. Here we show the worst possible representation of a saddle-shaped surface by a vertex star which itself is not even saddle-shaped.

behave in a much more irregular way than they do in the smooth surface case.

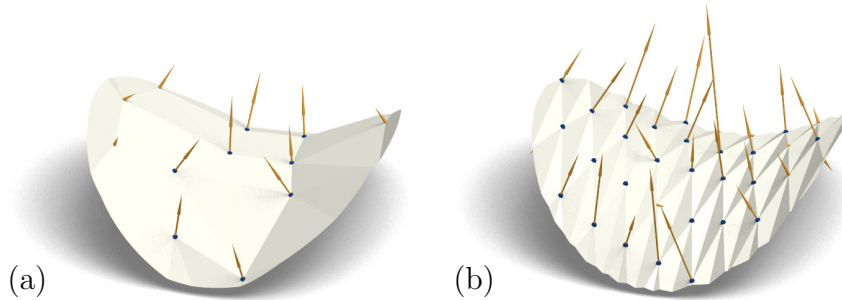
### 6.3.1 Saddle-shaped meshing of saddle-shaped surfaces

Even if a mesh is a regular sampling of a smooth saddle-shaped surface, vertex stars are not guaranteed to have proper saddle form.

Here saddle-shapedness means that there is a test plane (which can be considered a tangent plane) through the central vertex, intersecting the vertex star in precisely 4 line segments. We require that these four segments are not contained in a common half-plane, and that the vertex star projects onto the tangent plane in a 1-1 manner.

Examples of proper saddles are shown by Figures 6.3b and 6.5, while the vertex star of fig. 6.3c is neither convex nor a saddle.

Figure 6.4 demonstrates how evaluation of discrete differential quantities can go wrong even for a mesh precisely inscribed in a smooth surface, if the local geometry of the mesh does not reflect the local geometry of the surface. This experiment supports the claim that visually smooth meshes are better behaved with respect to the numerical differential geometry, even if the mesh and normals of faces approximate a smooth reference geometry to a similar extent.

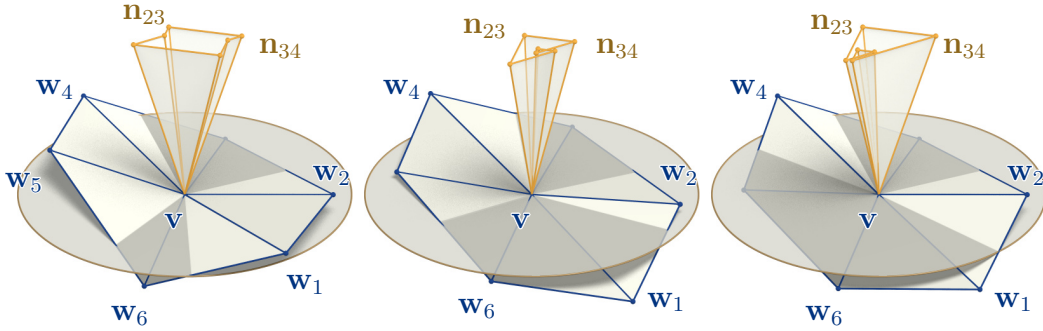


**Figure 6.4** – *Saddles and numerical differential geometry.* We display the “cotan” mean curvature field  $\vec{H}$  for a mesh inscribed in the surface  $z = x^2 - y^2/2$ . The  $x, y$  coordinates of vertices are taken from a regular lattice, slightly perturbed. Vertices and face normals approximate the smooth reference geometry to roughly the same extent in both (a) and (b), but  $\vec{H}$  is much better behaved in (a), where vertex stars are saddle-shaped. In (b), vertex stars are no saddles, like in fig. 6.3c. Compared to (a), a much smaller deviation from a precisely regular sampling makes  $\vec{H}$  break down.

### 6.3.2 The normal pyramid

Consider a vertex  $\mathbf{v}$  and neighbours  $\mathbf{w}_1, \mathbf{w}_2$  etc., in that order – see fig. 6.5. The normal vectors of faces  $\{\mathbf{v}, \mathbf{w}_i, \mathbf{w}_{i+1}\}$  form the *normal pyramid*. Such normal pyramids have been studied by [22, 26] who were in particular interested in cases where it has no self-intersections. This is a criterion for smoothness of the mesh. We attach a local  $xyz$  coordinate system with origin in  $\mathbf{v}$ , such that the tangent plane is the  $xy$  plane.

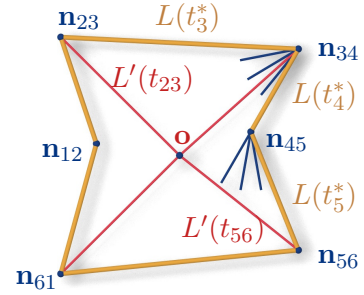
The edges  $\{\mathbf{v}, \mathbf{w}_j\}$  are partitioned into four sets of edges lying below resp. above the tangent plane. The number of elements of these sets can be 1, 2, 1, 2 (fig. 6.5a) or 2, 2, 1, 1 (fig. 6.5b) or 3, 1, 1, 1 (fig. 6.5c). We want to give a criterion which is easily checked and which ensures that the normal pyramid is free of self-intersections. We say a face  $(\mathbf{v}, \mathbf{w}_i, \mathbf{w}_{i+1})$  is an *inflection face*, see [22], if neighbours  $(\mathbf{v}, \mathbf{w}_{i-1}, \mathbf{w}_i)$  and  $(\mathbf{v}, \mathbf{w}_{i+1}, \mathbf{w}_{i+2})$  lie on different sides of the plane which carries the face  $f$ . In the simplest case of valence 4, every face of a saddle is an inflection face. We now say that a valence 6 saddle is a *good saddle* if the numbers of edges above and below the tangent plane is 1, 2, 1, 2 (fig. 6.5a) and the two faces not intersecting the tangent plane are no inflection faces.



**Figure 6.5** – (a) Good saddle, where  $\{\mathbf{v}, \mathbf{w}_1, \mathbf{w}_2\}$  and  $\{\mathbf{v}, \mathbf{w}_4, \mathbf{w}_5\}$  are no inflection faces. (b), (c): Bad saddles where the number of edges above and below the tangent plane is 2, 2, 1, 1 resp. 3, 1, 1, 1. Besides, there are inflection faces not intersecting the tangent plane.

**Proposition 8.** *In a “good” saddle point, the pyramid of vertex normals is free of self-intersections.*

**PROOF.** We cut off the normal pyramid by the plane  $\eta$  which has equation  $z = 1$ . Vertices  $\mathbf{w}_1, \dots, \mathbf{w}_6$  are neighbours of  $\mathbf{v}$  (indices modulo 6). In the plane  $\eta$  we mark the points  $\mathbf{n}_{i,i+1}$ , which are defined as intersection with  $\eta$  of the normal of the plane  $[\mathbf{v}\mathbf{w}_i\mathbf{w}_{i+1}]$ . Consider a ray  $L(t)$  rotating around  $v$  but always contained inside the surface of the mesh. Here  $t$  is a time parameter. Consider the plane  $L(t)^\perp$  orthogonal to  $L(t)$  and trace the intersection  $L'(t) = L(t)^\perp \cap \eta$ .



There are six time instances  $t_1^*, \dots, t_6^*$  where  $L(t_j^*)$  coincides with an edge  $\{\mathbf{v}, \mathbf{w}_j\}$ . Then  $L'(t_j^*)$  equals the line  $\mathbf{n}_{j-1,j} \vee \mathbf{n}_{j,j+1}$  (yellow in the inset figure). In the time interval where  $L(t)$  is contained in the face  $(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_{j+1})$ ,  $L'(t)$  rotates about  $\mathbf{n}_{j,j+1}$ . The rotation is consistently in one direction. Four times the line  $L(t)$  lies in the tangent plane. We assume the vertices are numbered so that these four lines lie in faces  $(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_{j+1})$  for  $(j, j+1) = (2, 3), (3, 4), (5, 6), (6, 1)$  (which are inflection faces). This happens at time instances  $t = t_{j,j+1}$ . The corresponding line  $L'(t_{j,j+1})$  passes through the point  $o$  where the  $z$  axis intersects  $\eta$  (shown in red).

Observe the ray  $L(t)$  rotating from edge  $\{\mathbf{v}, \mathbf{w}_4\}$  to edge  $\{\mathbf{v}, \mathbf{w}_5\}$ . The edge  $\{\mathbf{v}, \mathbf{w}_5\}$  is reached at time  $t = t_5^*$ , before the plane  $[\mathbf{v}\mathbf{w}_5\mathbf{w}_6]$  is crossed

(this is the non-inflection property of  $(\mathbf{v}, \mathbf{w}_4, \mathbf{w}_5)$ ). The line  $L'(t)$  during this rotation has not crossed  $\mathbf{n}_{56}$ . Similarly, during the rotation in the time interval  $[t_5^*, t_6^*]$ , the line  $L'(t)$  does not cross  $\mathbf{n}_{34}$ . Therefore  $\mathbf{n}_{45}$  lies in the interior of the triangle  $\mathbf{o}, \mathbf{n}_{34}, \mathbf{n}_{56}$ . Likewise,  $\mathbf{n}_{12}$  lies in the interior of the triangle  $\mathbf{o}, \mathbf{n}_{61}, \mathbf{n}_{23}$ .  $\square$

*Remark.* If a triangle mesh of regular combinatorics is generated by sampling a smooth surface in a geometrically regular way, it is hard to create saddle points with more than 4 inflection faces, and typically also the asymmetric situations of fig. 6.5b,c will not occur. There is however a nonzero probability that the mesh will have vertices which are neither convex nor proper saddles like in fig. 6.3.

## 6.4 Energies of polyhedral surfaces

The local shape properties of meshes discussed in section 6.3 are in direct relation to the *variation of normal vectors* of meshes. E.g. the total variation of normal vectors around a vertex, which equals the sum of unsigned dihedral angles of the vertex star, can be seen as a kind of spherical perimeter of the normal pyramid. A smaller total variation corresponds to a smaller perimeter which is obviously achieved in cases where the normal pyramid is free of self-intersections. We therefore propose to construct a discrete mesh energy  $\mathcal{E}$  as explained below, which penalizes big dihedral angles and in turn, a higher total variation of normals.

Consider a polyhedral surface  $M = (V, E, F)$  with vertex set  $V$ , edge set  $E$  and face set  $F$ . Each face  $f \in F$  has a normal vector  $\mathbf{n}_f$ . Any edge  $e \in E$  has a length  $\ell_e$ , a dihedral angle  $\alpha_e$ , and a dual edge length  $\ell_e^*$ , which is defined by the relations

$$e = f_k \cap f_l \implies |\alpha_e| = \angle(\mathbf{n}_f, \mathbf{n}'_f), \quad \ell_e^* = \|\mathbf{n}_{f_l} - \mathbf{n}_{f_k}\| = 2 \sin \frac{\alpha_e}{2}.$$

The energy

$$\mathcal{E}'(M) = \sum_e \ell_e |\alpha_e|$$



is built similar to discrete mean curvature and penalizes big dihedral angles (i.e., a big contribution to the total variation of normal vectors). A slight modification leads to the energy

$$\mathcal{E}(M) = \sum_e \ell_e \ell_e^* = 2 \sum_e \ell_e \left| \sin \frac{\alpha_e}{2} \right|,$$

which is easier to handle from the computational viewpoint.

With  $\sin x \approx x$  for small values of  $x$ , the energies  $\mathcal{E}, \mathcal{E}'$  assume similar values; they act identically for all practical purposes.

### 6.4.1 $\mathcal{E}$ -minimal meshes

We observe a strong analogy between the energy  $\mathcal{E}$  and the volume  $\mathcal{V}^*$  of a Michell truss, where instead of absolute values of axial forces we have now the absolute value of a dihedral angle measure — the length  $\ell_e^*$ . Arguing in a similar way as we did in section 5.3, we come to the following conclusion.

**Proposition 9.** *At the limit of refinement, among all meshes discretizing a given surface of negative Gaussian curvature, principal meshes minimize the energy  $\mathcal{E}$ . On surfaces of positive Gaussian curvature, at the limit of refinement, all discretizations yield a constant energy  $\mathcal{E}$ .*

PROOF. Let  $M = (V, E, F)$  be a mesh discretizing a surface  $\mathcal{M}$ , we look then for

$$\mathcal{E}(M) = \sum_e \ell_e^* \ell_e \rightarrow \min.$$

At a vertex  $\mathbf{v}_i$ , with ring vertices  $\mathbf{v}_j$ , let us consider the oriented edges  $\mathbf{e}_{ij} = \mathbf{v}_j - \mathbf{v}_i$ , and let  $f_k$  and  $f_l$  be, respectively, the left and right faces of the edge  $\mathbf{e}_{ij}$  according to a consistent surface orientation. Consider now, at each vertex  $\mathbf{v}_i$ , the base of the normal pyramid given by vectors  $\mathbf{n}_{f_l} - \mathbf{n}_{f_k}$ , and let  $\hat{\mathbf{b}}_{ij}$  be the corresponding unitized vectors oriented consistently when turning counterclockwise around the base. We define the signed dihedral difference  $\gamma_{ij}$ , corresponding to the edge  $\mathbf{e}_{ij}$ , as

$$\gamma_{ij} = \hat{\mathbf{b}}_{ij}^T (\mathbf{n}_{f_l} - \mathbf{n}_{f_k}).$$

The closure condition of the base of the normal pyramid asks

$$\sum_{j \sim i} \gamma_{ij} \hat{\mathbf{b}}_{ij} = 0. \quad (6.1)$$

Note that  $\gamma_{ij} = \gamma_{ji}$ , we can then assign this value to non oriented edges as  $\gamma_e$ . We have then  $|\gamma_e| = \ell_e^*$ . Let us collect the signed dihedral differences  $\gamma_e$  in the vector  $\boldsymbol{\gamma} \in \mathbb{R}^{|E|}$ . The closure conditions can be written in matrix form as

$$B\boldsymbol{\gamma} = \mathbf{0},$$

where in the matrix  $B \in \mathbb{R}^{3|V| \times |E|}$  are properly rearranged the vectors  $\hat{\mathbf{b}}_{ij}$ .

Let us now define the Lagrange multipliers  $\boldsymbol{\lambda}_i \in \mathbb{R}^3$  for each vertex  $\mathbf{v}_i$ , and the functions

$$\varepsilon_{ij} = \frac{\hat{\mathbf{b}}_{ij}^T (\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_i)}{\ell_{ij}}, \quad (6.2)$$

where  $\ell_{ij} = \ell_{ji} = \ell_e$ . Observing that  $\varepsilon_{ij} = \varepsilon_{ji}$ , we can again consider the functions  $\varepsilon_e$  defined per edge. Collecting the multipliers  $\boldsymbol{\lambda}_i$  in the vector  $\boldsymbol{\lambda} \in \mathbb{R}^{3|V|}$  and the values  $\varepsilon_e, \ell_e$  respectively in the vectors  $\boldsymbol{\varepsilon}, \boldsymbol{\ell} \in \mathbb{R}^{|E|}$ , we can write

$$\boldsymbol{\varepsilon} * \boldsymbol{\ell} = B^T \boldsymbol{\lambda}, \quad (6.3)$$

where with  $*$  is denoted the element-wise vector multiplication.

We can now write the Lagrangian equation

$$\mathcal{L}(\mathbf{v}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) = \mathcal{E}(\mathbf{v}, \boldsymbol{\gamma}) - \boldsymbol{\lambda}^T B \boldsymbol{\gamma},$$

Where the vertices  $\mathbf{v}_i$  are collected in the vector  $\mathbf{v} \in \mathbb{R}^{3|V|}$ . Substituting eq. (6.3), we get  $\boldsymbol{\lambda}^T B \boldsymbol{\gamma} = (\boldsymbol{\varepsilon} * \boldsymbol{\ell})^T \boldsymbol{\gamma}$ . Let then  $p$  and  $n$  be the indices for the edges where  $\gamma_e$  is, respectively, positive and negative. Recalling that  $\ell_p^* = \gamma_p$  and  $\ell_n^* = -\gamma_n$ , the Lagrangian equation can be rewritten in sum notation as

$$\mathcal{L}(\mathbf{v}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) = \sum_p \ell_p \gamma_p - \sum_n \ell_n \gamma_n - \sum_p \varepsilon_p \ell_p \gamma_p - \sum_n \varepsilon_n \ell_n \gamma_n.$$

It's dual is

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{v}} \min_{\gamma_p \geq 0, \gamma_n \leq 0} \mathcal{L}(\mathbf{v}, \boldsymbol{\gamma}, \boldsymbol{\lambda}),$$

and then, interchanging the min and max operations on  $\mathbf{v}$  and  $\boldsymbol{\lambda}$ , we can solve

$$\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) = \min_{\mathbf{v}} \max_{\boldsymbol{\lambda}} K(\boldsymbol{\lambda}),$$

with

$$K(\boldsymbol{\lambda}) = \min_{\gamma_p \geq 0, \gamma_n \leq 0} \left[ \sum_t (1 - \varepsilon_p) \gamma_p \ell_t + \sum_n (-1 - \varepsilon_n) \gamma_n \ell_n \right].$$

The minimum over  $\boldsymbol{\gamma}$  is attained when

$$K(\boldsymbol{\lambda}) = \begin{cases} 0, & \text{if } \varepsilon_p = 1 \text{ and } \varepsilon_n = -1 \text{ with } \gamma_p \geq 0, \gamma_n \leq 0 \\ 0, & \text{if } \varepsilon_p < 1 \text{ and } \varepsilon_n > -1 \text{ with } \gamma_p = 0, \gamma_n = 0 \\ -\infty, & \text{if } \varepsilon_p > 1 \text{ with } \lambda_p = +\infty \\ -\infty, & \text{if } \varepsilon_n < -1 \text{ with } \lambda_n = -\infty \end{cases}. \quad (6.4)$$

The last two cases are excluded by the max operation over  $\boldsymbol{\lambda}$ .

We now consider the problem at the limit of refinement, as in section 5.3.6. We have then  $\mathbf{v} \equiv \mathcal{M}$ , making the dependency on  $\mathbf{v}$  redundant. All quantities are now defined in the tangent plane  $\mathcal{T}_{\mathcal{M}}(\mathbf{v}_i)$ . At each point  $\mathbf{v}_i$  of  $\mathcal{M}$ , we consider the Lagrange multipliers as a tangent vector field  $\boldsymbol{\lambda}_{\mathcal{M}}(\mathbf{v}_i)$  over the surface  $\mathcal{M}$ . Let us consider a unit orthogonal local basis for  $\mathcal{T}_{\mathcal{M}}(\mathbf{v}_i)$ , and let  $\mathbf{a} \in \mathcal{T}_{\mathcal{M}}(\mathbf{v}_i)$  be a direction in the tangent plane corresponding to the infinitesimal edge pointing to the vertex  $\mathbf{v}_j(\mathbf{a})$ . We now need a continuous version of eq. (6.2) for the edges  $\mathbf{e}_{ij}(\mathbf{a})$ . We first can see that

$$\lim_{\mathbf{v}_j \rightarrow \mathbf{v}_i} \frac{\boldsymbol{\lambda}_{\mathcal{M}}(\mathbf{v}_j) - \boldsymbol{\lambda}_{\mathcal{M}}(\mathbf{v}_i)}{\|\mathbf{v}_j - \mathbf{v}_i\|} = \frac{\nabla \boldsymbol{\lambda}_{\mathcal{M}}(\mathbf{v}_i) \mathbf{a}}{\|\mathbf{a}\|} + o(\|\mathbf{v}_j - \mathbf{v}_i\|),$$

while the vector  $\hat{\mathbf{b}}(\mathbf{a})$  is now given by  $R \mathbf{a} / \|\mathbf{a}\|$ , where  $R$  is the 90° counter-clockwise rotation in the tangent plane. We can then write the continuous

version of eq. (6.2) as

$$\varepsilon_{\mathcal{M}}(\mathbf{a}) = \frac{(R\mathbf{a})^T[\nabla\lambda_{\mathcal{M}}(\mathbf{v}_i)\mathbf{a}]}{\mathbf{a}^T\mathbf{a}}.$$

This can be written as a function

$$\varepsilon_{\mathcal{M}}(\mathbf{a}) = \frac{\mathbf{a}^T E \mathbf{a}}{\mathbf{a}^T \mathbf{a}}, \quad (6.5)$$

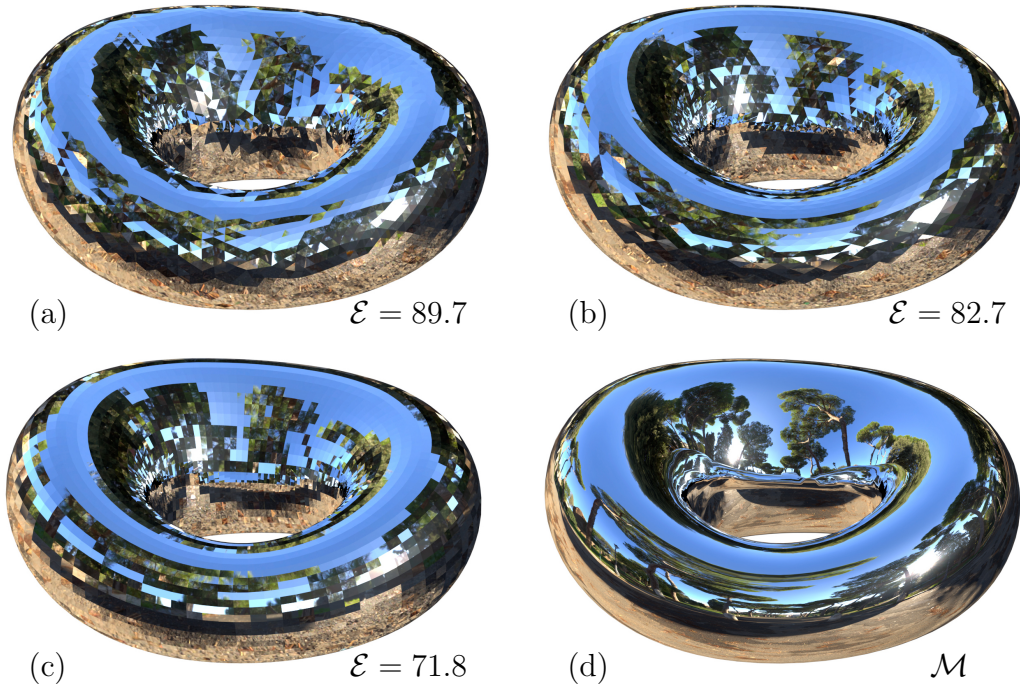
where  $E$  is a symmetric matrix representing the quadratic form on the numerator, given by

$$E = \frac{1}{2}(R^T \nabla \lambda_{\mathcal{M}} + \nabla \lambda_{\mathcal{M}}^T R),$$

where we omitted the dependency on  $\mathbf{v}_i$ . The extrema of  $\varepsilon_{\mathcal{M}}(\mathbf{a})$  occur then along the two eigenvectors of  $E$ , corresponding to two directions orthogonal in the tangent plane. Equation (6.4) asks that the corresponding eigenvalues are in the interval  $[-1, 1]$ , and requires that for  $-1 < \varepsilon_{\mathcal{M}} < 1$  the dihedral angle vanishes. Therefore, in a  $\mathcal{E}$ -minimal mesh, infinitesimal edges have non vanishing dihedral angle only when oriented along the eigenvectors of  $E$  whose eigenvalues attain one of the limits  $\pm 1$ . We have then the following possibilities:

- $\text{spec}(E) = \{-1, 1\}$ , *negatively curved regions of  $\mathcal{M}$* . The  $\mathcal{E}$ -minimal mesh can have non vanishing dihedral angles only along two orthogonal directions. This implies orthogonality of edges and planarity of faces. Therefore, according to proposition 1, at the limit of refinement it is a principal mesh of  $\mathcal{M}$ .
- $\text{spec}(E) = \{1, 1\}$  or  $\{-1, -1\}$ , *positively curved regions of  $\mathcal{M}$* . All directions of edges are allowed. At the limit of refinement all meshes are  $\mathcal{E}$ -minimal.

□

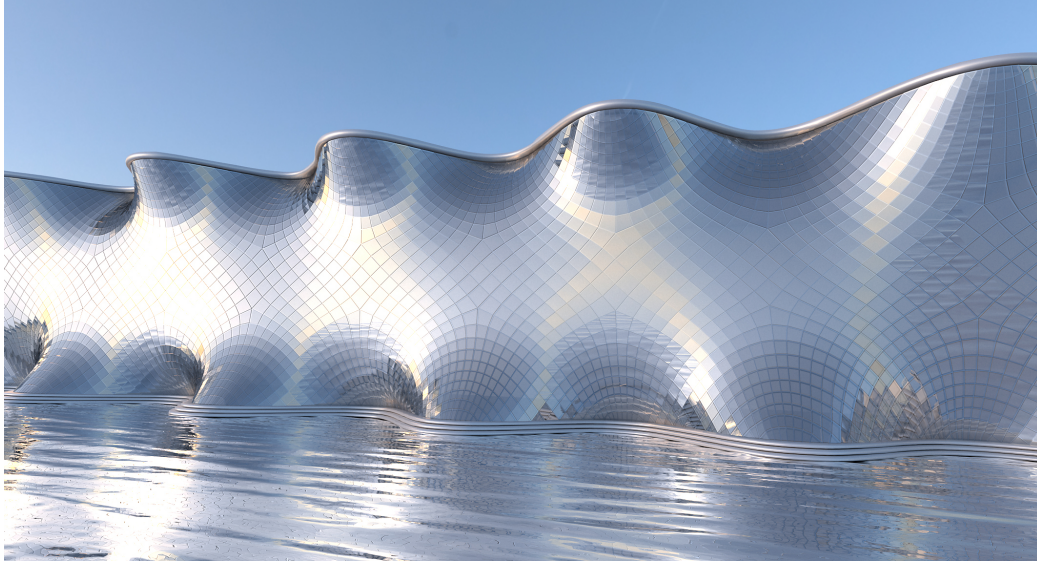


**Figure 6.6** – Meshes with similar number of faces approximating the same smooth surface  $\mathcal{M}$  (d). From top left, a generic triangular mesh (a), a mesh with good saddles and highly symmetric (b), and a principal mesh (c). We observe decreasing values of the energy  $\mathcal{E}$  and corresponding increase of the quality of reflection, especially in negatively curved regions.

### 6.4.2 Differential-geometric interpretation of the energy $\mathcal{E}$

Recall the definition of the shape operator  $\mathbb{S}$  of a surface  $\mathcal{M}$ , given in eq. (2.5). When sitting in a point  $\mathbf{x} \in \mathcal{M}$  and moving by the small amount  $\delta\mathbf{x}$ , the unit normal vector  $\mathbf{n}$  is incremented by  $\delta\mathbf{n}$ . In tangent coordinates we have  $\delta\mathbf{n} \approx -\mathbb{S}\delta\bar{\mathbf{x}}$ . The shape operator  $\mathbb{S}$  is linear. Its eigenvectors are orthogonal and indicate the principal directions: if the increment  $\delta\bar{\mathbf{x}}$  is in the  $j$ -th *principal* direction ( $j = 1, 2$ ), then the normal vector increment can be computed as  $\delta\mathbf{n} = -\kappa_j \delta\bar{\mathbf{x}}$ , where  $\kappa_j$  is the  $j$ -th principal curvature. Therefore

$$\delta\mathbf{n} = -\kappa_j \delta\bar{\mathbf{x}}.$$

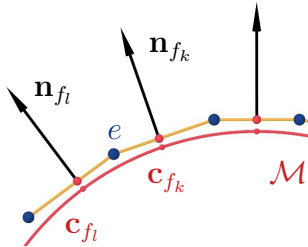


**Figure 6.7** – Here a surface is endowed with a principal quad mesh. In addition to the meshing, the reference surface itself has minimal total absolute curvature, for prescribed boundaries. The rendering is that of a freeform glass facade.

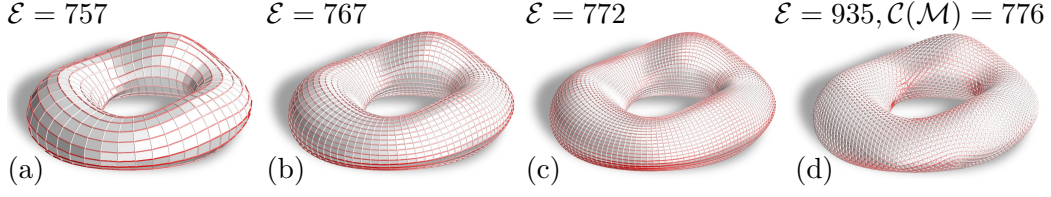
In an orthonormal coordinate system aligned with the principal directions,  $\mathbb{S}$  is described by the diagonal matrix  $(\begin{smallmatrix} \kappa_1 & \\ & \kappa_2 \end{smallmatrix})$ . For a principal quad mesh  $M = (V, E, F)$ , the edge polylines approximate the principal curvature lines of a smooth surface  $\mathcal{M}$ . In such cases the faces of the mesh (away from umbilics) are near-flat, and can in fact be assumed flat, as shown in section 2.2.4.

For any face  $f$  of the mesh, the normal vector  $\mathbf{n}_f$  is also a normal vector of the underlying smooth reference surface  $\mathcal{M}$  in an appropriate point. This location, when projected back onto the face  $f$ , yields a point  $\mathbf{c}_f$ , see fig. 6.8.

For any edge  $e = f_k \cap f_l$ , we now have the following correspondences



**Figure 6.8** – To interpret the energy  $\mathcal{E}$ , we look at faces  $f, f'$  intersecting in an edge  $e$ . Stepping from  $f$  to  $f'$  (actually, from point  $\mathbf{c}_f$  to point  $\mathbf{c}_{f'}$ ), we increment position by  $\delta\mathbf{x} = \mathbf{c}_{f'} - \mathbf{c}_f$  and the normal vector by  $\delta\mathbf{n} = \mathbf{n}_{f'} - \mathbf{n}_f$ .



**Figure 6.9** – Convergence of mesh energy to total absolute curvature  $\mathcal{C}(\mathcal{M})$ . (a)–(c) show finer and finer principal meshes, each approximating the same reference shape  $\mathcal{M}$ . The energies approach  $\mathcal{C}(\mathcal{M})$ . Every edge  $e$  is color-coded according to increasing values of  $\ell_e|\alpha_e|$ , from white to red. The reference surface  $\mathcal{M}$  is represented by the triangle mesh (d).

between the discrete mesh situation and the smooth surface situation:

$$\begin{aligned} \mathbf{c}_{f_l} - \mathbf{c}_{f_k} &\approx \delta \mathbf{x}, & \mathbf{n}_{f_l} - \mathbf{n}_{f_k} &\approx \delta \mathbf{n}, & \ell_e^* &\approx \|\delta \mathbf{n}\| \approx |\kappa_e| \|\delta \mathbf{x}\| \\ \implies \mathcal{E}(M) &= \sum_e \ell_e \ell_e^* \approx \sum_e |\kappa_e| \ell_e \|\mathbf{c}_{f_l} - \mathbf{c}_{f_k}\|. \end{aligned}$$

Here  $\kappa_e$  is the principal curvature in the direction orthogonal to  $e$ . The quadrangle spanned by  $\mathbf{c}_{f_k}$ ,  $\mathbf{c}_{f_l}$  and  $e$  has area approximately  $\frac{1}{2} \ell_e \|\mathbf{c}_{f_l} - \mathbf{c}_{f_k}\|$ . One half of the edges corresponds to the first principal direction, the other half to the 2nd principal direction, so the term  $\ell_e \|\mathbf{c}_{f_l} - \mathbf{c}_{f_k}\|$  occurring in the formula above is the area of influence of the edge  $e$ . Summing up,

$$\mathcal{E}(P) \approx \int_{\mathcal{M}} (|\kappa_1| + |\kappa_2|) = \mathcal{C}(\Phi). \quad (6.6)$$

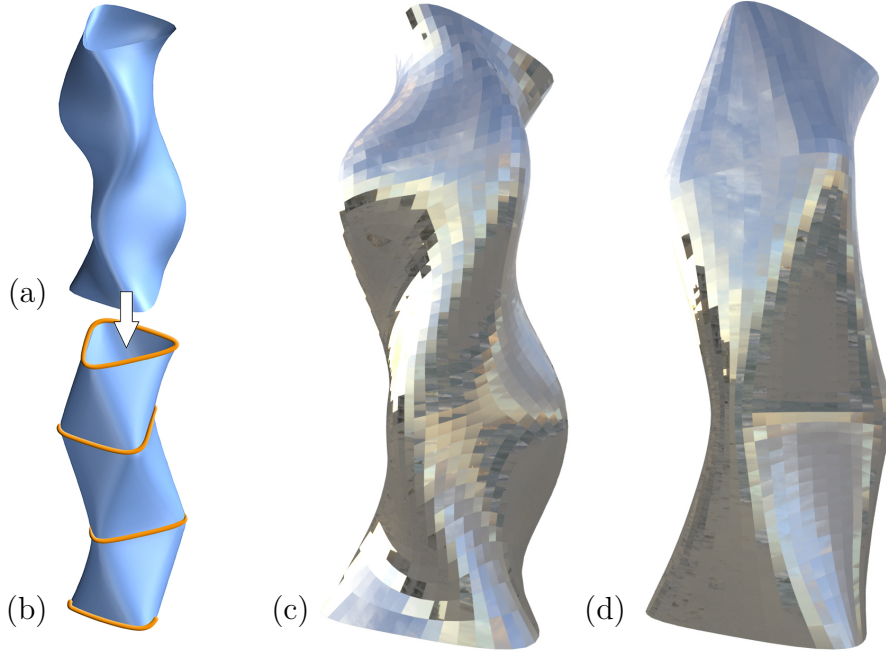
Numerical experiments confirm this, see fig. 6.9. We will return to the functional  $\mathcal{C}(\mathcal{M})$  in section 6.4.3. We call this quantity *total absolute curvature*. Note the analogy of eq. (6.6) with eq. (5.11) of total isotropic absolute curvature, arising from volume minimization of 2D trussed shown in section 5.4.

### 6.4.3 Total absolute curvature of surfaces

We already saw that the total absolute curvature

$$\mathcal{C}(\mathcal{M}) = \int_{\mathcal{M}} (|\kappa_1| + |\kappa_2|)$$





**Figure 6.10** – A given surface (a) undergoes to minimization of  $\int |\kappa_1| + |\kappa_2|$ . (b) for fixed boundaries shown in orange. The resulting surface is shown in (c) a principal mesh approximating (a), while in (d) a principal mesh approximating (b). The mesh (d) is then approximately the smoothest mesh over the fixed boundaries. Observe that minimizers have features not seen in minimizers of e.g. Willmore energy.

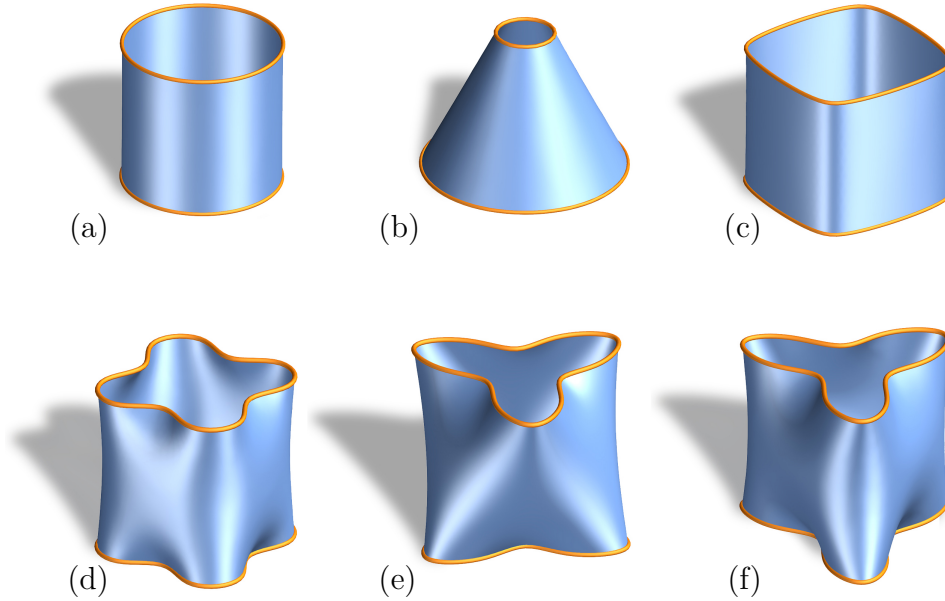
of a smooth surface  $\mathcal{M}$  coincides with the smallest achievable value of the discrete energy  $\mathcal{E}$  for polyhedral meshes approximating  $\mathcal{M}$ . It therefore makes sense to ask the question which surface  $\mathcal{M}$ , under given boundary conditions, achieves the lowest value of  $\mathcal{C}(\mathcal{M})$ .

The shape operator  $\mathbb{S}$  in a principal coordinate frame is described by the diagonal matrix  $(\begin{smallmatrix} \kappa_1 & \\ & \kappa_2 \end{smallmatrix})$ , so total absolute curvature equals

$$\mathcal{C}(\mathcal{M}) = \int \|\mathbb{S}\|_1, \quad \text{where} \quad \|\mathbb{S}\|_1 = |\kappa_1| + |\kappa_2|$$

is the 1-norm (trace norm) of the shape operator. It can be used to bound the infinitesimal normal vector increment  $\delta\bar{\mathbf{n}} \approx -\mathbb{S}\delta\bar{\mathbf{x}}$  caused by the position increment  $\delta\bar{\mathbf{x}}$  (cf. the discussion in the beginning of section 6.4). Like any





**Figure 6.11** – *Surfaces minimizing total absolute curvature.* Here we show surfaces with prescribed boundary curves (in orange) which minimize  $\int |\kappa_1| + |\kappa_2|$ . Among rotational surfaces, minimizers are cylinders and cones. Observe that minimizers have features not seen in minimizers of e.g. Willmore energy.

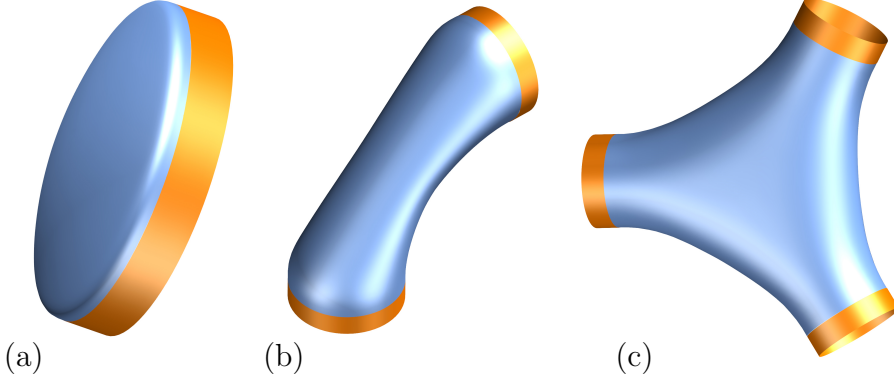
proper matrix norm,  $\|\mathbb{S}\|_1$  obeys

$$\|\delta\bar{\mathbf{n}}\| \approx \|\mathbb{S} \delta\bar{\mathbf{x}}\| \leq \|\mathbb{S}\|_1 \cdot \|\delta\bar{\mathbf{x}}\|. \quad (6.7)$$

Because of this formula, the total absolute curvature  $\mathcal{C}(\mathcal{M})$  can be interpreted as the total variation  $\text{TV}(\mathbf{n})$  of the normal vector field, if the size of derivatives is measured using  $\|\cdot\|_1$ .

Only in special situations we are able to describe the minimizers of  $\mathcal{C}(\mathcal{M})$ . Consider e.g. a surface with rotational symmetry about the  $z$  axis described by a radius function  $r(z) > 0$  and boundary values  $r(z_0) = r_0$ ,  $r(z_1) = r_1$ . It is not difficult to compute the area element  $dA = 2\pi r \sqrt{1 + r'^2} dz$  and curvatures  $\kappa_1 = 1/(r\sqrt{1 + r'^2})$ ,  $\kappa_2 = r''(1 + r'^2)^{-3/2}$ . Thus,  $\int |\kappa_1| dA = \int 2\pi dz = 2\pi(z_1 - z_0)$  depends only on the boundary conditions, whereas  $\int |\kappa_2| dA = \int r''(\dots) dz$  vanishes if and only if  $r(z)$  is linear. Thus, among rotational surfaces, minimizers are cones and cylinders (see fig. 6.11).

The computation of surfaces which minimize energy under different bound-



**Figure 6.12** – Minimizers of total absolute curvature where boundaries plus tangent planes along boundaries are prescribed (in orange).

ary conditions is treated below. Examples are shown throughout this chapter figs. 6.1, 6.2, 6.7 and 6.11 to 6.14.

## 6.5 Implementation

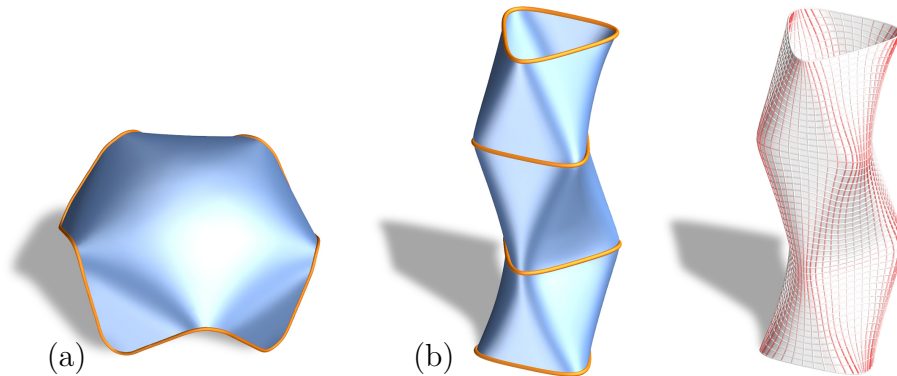
We are solving the following problem: *Find surfaces of small total absolute curvature under given boundary conditions.*

We show how to compute a surface  $\mathcal{M}$  which minimizes total absolute curvature under boundary conditions. The surface is represented by a mesh  $M = (V, E, F)$  of sufficiently high resolution, which have to be optimized w.r.t. a nonlinear target functional. For that purpose, we employ again *guided projection* as proposed by [55] described in section 4.4.2.

Variables and constraints are built on top of the implementation of curvature directions shown in section 4.4.1. For target functional  $\mathcal{C}(\mathcal{M})$  we use a weighted iteration approach. In the  $k$ -th round of iteration we do not wish to minimize  $\sum |\lambda_{1,\mathbf{v}}| + |\lambda_{2,\mathbf{v}}|$  directly, but rather the quadratic function

$$E_{\text{target}} = \sum_{\mathbf{v}} \frac{\lambda_{1,\mathbf{v}}^2}{|\lambda_{1,\mathbf{v}}^{(k-1)}| + \epsilon} + \frac{\lambda_{2,\mathbf{v}}^2}{|\lambda_{2,\mathbf{v}}^{(k-1)}| + \epsilon}.$$

Here  $\lambda_{j,\mathbf{v}}^{(k-1)}$  is the value of the variable  $\lambda_{j,\mathbf{v}}$  assumed in the previous round of iteration.  $\epsilon$  is a small regularizer for the division.  $E_{\text{target}}$  is added to guided

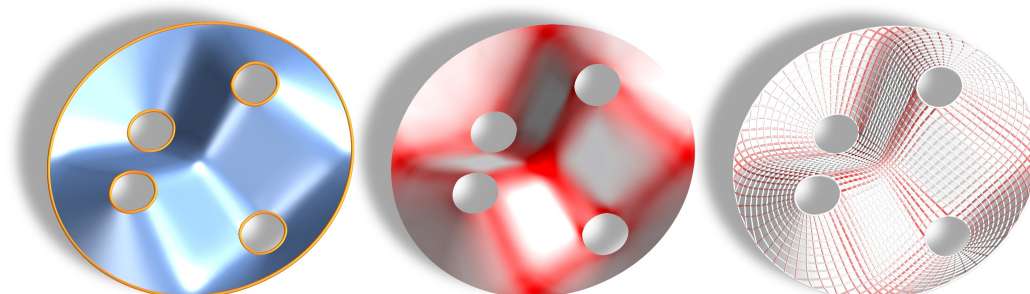


**Figure 6.13** – *Surfaces minimizing total absolute curvature.* Here we show other surfaces including prescribed curves (in orange) which minimize  $\int |\kappa_1| + |\kappa_2|$ . On the right, a principal mesh approximating shape (b), with edges color-coded according to increasing values of  $\ell_e|\alpha_e|$ , from white to red.

projection eq. (4.6), as regularizer. Its weight  $w_{\text{target}}$  has to be chosen such that the overall regularizer is still positive definite.

## 6.6 Results and Discussion

We have at our disposal the procedure to find a surface minimizing total absolute curvature for given boundary conditions. In addition to that, a mesh approximating that surface has minimal energy, if its faces are planar quads following the principal curvature lines. Meshing and postprocessing, however, are not contributions of this thesis. We refer to [13] for meshing with edges aligned with principal curvatures, and to [55] for postprocessing for planarity of faces. Examples are given by figures 6.1, 6.7, 6.11, 6.12, 6.2, 6.14.



**Figure 6.14** – Surface of minimal total absolute curvature. We also show the value of absolute curvature (center; white = zero) and a principal remeshing with edges color coded according to dihedral angle (right).

Fig.	$ V $	var.	$\epsilon$	$\delta$	$w_{\text{target}}$	$\mathcal{C}(\mathcal{M})$	iter.	time (s)
6.1	2784	87k	$10^{-3}$	.30	.05	94.3	18	58.3
6.7	2606	82k	$10^{-3}$	.20	.03	179.3	11	32.8
6.11a,6.2b	1056	33k	$10^{-3}$	.30	.05	124.4	14	16.1
6.11b	4224	133k	$10^{-3}$	.20	.10	180.2	11	57.4
6.11c	555	17k	$10^{-3}$	.20	.05	45.5	7	3.8
6.12a	6084	193k	$10^{-3}$	.05	.01	36.9	8	69.4
6.12c	2166	68k	$10^{-3}$	.10	.05	171.6	17	45.3
6.14	2118	66k	$10^{-3}$	.20	.05	143.8	17	39.2

**Table 6.1** – Statistics for minimizing  $\mathcal{C}(\mathcal{M})$ . We give the number of vertices of the meshes we ran the optimization on (not necessarily the meshes shown in figures), the number of variables as well as the weights used for regularizing functionals in the *guided projection* method. Here  $\epsilon$  is the weight of previous iteration closeness and  $\delta$  the weight of fairness, see eq. (4.6). Experience shows that after about half the number of iterations shown, the minimizer shape is more or less defined. The algorithm has been implemented in Python and tested with an Intel Core i7-6700HQ CPU with 2.60 GHz and a 15.9 GB RAM memory. We show total computation time in seconds.

# Chapter 7

## Conclusion

Throughout this thesis, we have seen how to realize conveniently a freeform load-bearing structure spanning over a given boundary. Not only for well known manufacturing reasons, but also for mechanical and visual performance, among all meshes, quadrilateral ones turn out to be optimal solutions.

In chapter 2 we have seen that principal meshes are a convenient solution for manufacturing reasons, namely planar panels and a torsion-free substructure. In chapter 4, we have seen how a principal mesh can be in axial force equilibrium as well. This is not possible for all shapes, but only for those in membrane equilibrium where principal stress and curvature directions coincide. In chapter 5, we have shown that, from a mechanical viewpoint, the most efficient gridshells are those following the principal stress directions of a membrane — these are then quad meshes. Fixing some boundary conditions, the absolute minimum is then achieved on membranes that minimize the total absolute stress. Finally, in chapter 6, we have shown that principal meshes are also the “smoothest” discretizations of surfaces. The smoothest meshes for a given boundary are then principal meshes discretizing surfaces with minimal total absolute curvature.

From the presented results, we can see that mechanical optimality does not require principal meshes, but just orthogonal quad meshes in equilibrium. Orthogonality is strictly required only for tension-compression regions, but for practical reasons can be assumed for the whole structure. The choice

of a mechanically sub-optimal solution, where it is minimized the total absolute stress under the additional additional constraint of stress and curvature alignment, appears to be a very efficient solution. With a principal remeshing of the resulting surface, we get then the principal mesh that uses the minimal structural volume, and at the same time the discretization of that shape with the smoothest visual appearance. The minimization of total absolute curvature could be considered for structures where the mechanical performance is less demanding, or just used as a slight fairness energy over total absolute stress to slightly improve visual smoothness.

## 7.1 Limitations

The main limitation of this approach is due to the lack of design control over the mesh layout. Not all shapes own a stress-curvature network suitable for the extraction of architectural meshes. Indeed, the network layout may yield a mesh with a large variation of cell size, numerous or bad positioned singularities, or more generally, the resulting mesh may not possess the desired aesthetic qualities. Nevertheless, the principal curvature network layout is highly sensible to shape changes; small modifications in the shape can then solve the issue. For that, an interactive procedure can help. However, computational times of our implementations are surely borderline for interactive modeling, but still acceptable for many kind of users.

Moreover, some of the constraints and optimization targets can be conflicting. Not all designer's wishes can be fulfilled, for instance when a fixed boundary curve is expected to be aligned with a principal direction. Another limitation of the methods presented in this thesis therefore lies in understanding the effect of side conditions. They must be balanced by weighting, which is a designer's choice. So far we have an academic implementation of our procedures which requires some experience in correctly setting the weights.

On the statics side, the presented methods do not consider buckling effects in the form finding criteria. Nevertheless, the buckling of a gridshell can be considered as a phenomenon connected with out-of-plane stiffness. While with our method we minimize the cross sectional area of beams, the out-of-plane stiffness can be controlled, up to a certain extent, with an appropriate

distribution of this area among the cross section to maximize the moment of inertia.

## 7.2 Further research

It is easy to see a strong connection between total absolute curvature and total absolute stress in, respectively, material and kinks minimization on discrete structures. More generally, the systematic study of optimization problems which exhibit both a discrete and a continuous version can be a research direction. Likewise, the total absolute curvature of surfaces deserves a thorough investigation, also from the mathematical point of view.

Regarding static optimization, the presented methods look for structures in axial force equilibrium only. An interesting problem is to find the optimal layout on given surfaces that, under the given loads, cannot be in membrane equilibrium, considering then the effect of out-of-plane bending. It is also of interest to develop tools for the design of fully 3D optimal frameworks.

# Bibliography

- [1] Adriaenssens, S., P. Block, D. Veenendaal, and C. Williams (Eds.) (2014). *Shell Structures for Architecture*. Taylor & Francis.
- [2] Alboul, L. and R. van Damme (1996). Polyhedral metrics in surface reconstruction. In G. Mullineux (Ed.), *The Mathematics of Surfaces VI*, pp. 171–200. Clarendon Press.
- [3] Andrews, B. (1999). Gauss curvature flow: the fate of the rolling stones. *Inventiones mathematicae* 138, 151–161.
- [4] Angelillo, M. and A. Fortunato (2004). Equilibrium of masonry vaults. In M. Fremond and F. Maceri (Eds.), *Novel Approaches in Civil Engineering*, pp. 106–111. Oxford: Springer.
- [5] Ash, P., E. Bolker, H. Crapo, and W. Whiteley (1988). Convex polyhedra, Dirichlet tessellations, and spider webs. In *Shaping space*, pp. 231–250. Birkhäuser.
- [6] Baker, W. F., L. L. Beghini, A. Mazurek, J. Carrion, and A. Beghini (2013). Maxwells reciprocal diagrams and discrete Michell frames. *Structural and Multidisciplinary Optimization* 40(2), 267–277.
- [7] Bauer, U., K. Polthier, and M. Wardetzky (2010). Uniform convergence of discrete curvatures from nets of curvature lines. *Discrete & Computational Geometry* 43, 798–823.
- [8] Bergmann, R., M. Herrmann, R. Herzog, S. Schmidt, and J. Vidal Núñez (2019). Total variation of the normal vector field as



- shape prior with applications in geometric inverse problems. preprint. <http://arxiv.org/abs/1902.07240>.
- [9] Bletzinger, K.-U. and E. Ramm (1993). Form finding of shells by structural optimization. *Engineering with Computers* 9(1), 27–35.
- [10] Block, P. and J. Ochsendorf (2007). Thrust network analysis: A new methodology for three-dimensional equilibrium. *Journal of International Association for Shell and Spatial Structures* 48(3), 167–173.
- [11] Bobenko, A., D. Matthes, and Y. Suris (2003). Discrete and smooth orthogonal systems:  $C^\infty$ -approximation. *International Mathematics Research Notices* (45), 2415–2459.
- [12] Bobenko, A. I. and P. Schröder (2005). Discrete Willmore flow. In *Geometry Processing 2005 (Symposium Proceedings)*, pp. 101–110.
- [13] Bommes, D., H. Zimmer, and L. Kobbelt (2009). Mixed-integer quadrangulation. *ACM Transactions on Graphics* 28(3), 77:1–77:10.
- [14] Botsch, M., L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy (2010). *Polygon mesh processing*. AK Peters/CRC Press.
- [15] Cohen-Steiner, D. and J. M. Morvan (2003a). Restricted Delaunay triangulations and normal cycle. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, New York, pp. 312–321. ACM.
- [16] Cohen-Steiner, D. and J.-M. Morvan (2003b). Restricted Delaunay triangulations and normal cycle. In *Proc. 19th SoCG*, pp. 237–246.
- [17] Cox, H. L. (1965). *The design of structures of least weight*. Oxford: Pergamon Press.
- [18] do Carmo, M. P. (1976). *Differential Geometry of Curves and Surfaces*. Prentice-Hall.
- [19] Dyn, N., K. Hormann, S.-J. Kim, and D. Levin (2001). Optimizing 3D triangulations using discrete curvature analysis. In *Mathematical Methods for Curves and Surfaces: Oslo 2000*, pp. 135–146. Vanderbilt University Press.

- [20] Fraternali, F. (2010). A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions. *Mechanics Research Communications* 37(2), 198–204.
- [21] Garanzha, V. A. (2010). Discrete extrinsic curvatures and approximation of surfaces by polar polyhedra. *Comp. Math. & Math. Physics* 50, 65–92.
- [22] Günther, F., C. Jiang, and H. Pottmann (2017). Smooth polyhedral surfaces. preprint. <http://arxiv.org/abs/1703.05318>.
- [23] Hemp, W. S. (1973). *Optimum Structures*. Oxford: Clarendon Press.
- [24] Hildebrandt, K., K. Polthier, and M. Wardetzky (2006). On the convergence of metric and geometric properties of polyhedral surfaces. *Geometriae Dedicata* 123, 89–112.
- [25] Huisken, G. (1984). Flow by mean curvature of convex surfaces into spheres. *Journal of Differential Geometry* 20, 237–266.
- [26] Jiang, C., F. Günther, J. Wallner, and H. Pottmann (2016). Measuring and controlling fairness of triangulations. In *Advances in Architectural Geometry 2016*, pp. 24–39. Zürich: vdf Hochschulverlag.
- [27] Jiang, C., C. Tang, H.-P. Seidel, and P. Wonka (2017). Design and volume optimization of space structures. *ACM Transactions on Graphics* 36(4), 159:1–159:14.
- [28] Kilian, M., D. Pellis, J. Wallner, and H. Pottmann (2017). Material-minimizing forms and structures. *ACM Transactions on Graphics* 36(6), 173:1–173:12. Proc. SIGGRAPH Asia.
- [29] Lewiński, T. and T. Sokół (2014). On basic properties of michell’s structures. In G. Rozvani and T. Lewiński (Eds.), *Topology Optimization in Structural and Continuum Mechanics*, pp. 87–128. Udine: Springer.
- [30] Lewiński, T., T. Sokół, and C. Graczykowski (2019). *Michell Structures*. Springer.

- [31] Liu, Y., H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang (2006). Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics* 25(3), 681–689. Proc. SIGGRAPH.
- [32] Martin, R. R., J. De Pont, and T. J. Sharrock (1986). Cyclide surfaces in computer aided design. *The mathematics of surfaces*, 253–268.
- [33] Maxwell, J. C. (1872). On reciprocal figures, frames, and diagrams of forces. *Transactions of the Royal Society of Edinburgh* 26, 1–40.
- [34] Michell, A. G. M. (1904). The limits of economy of material in frame-structures. *Philosophical Magazine Series 6* 8(47), 589–597.
- [35] Miki, M., T. Igarashi, and P. Block (2015). Parametric self-supporting surfaces via direct computation of Airy stress functions. *ACM Transactions on Graphics* 34(4), 89:1–89:12.
- [36] Mitchell, T. (2013). *A Limit of economy of material in shell structures*. Ph. D. thesis, University of California, Berkeley.
- [37] Nervi, P. L. (1945). *Scienza o arte del costruire? Caratteristiche e possibilità del cemento armato*. Edizioni della Bussola.
- [38] Orden, D., G. Rote, F. Santos, B. Servatius, H. Servatius, and W. Whiteley (2004). Non-crossing frameworks with non-crossing reciprocals. *Discrete & Computational Geometry* 32(4), 567–600.
- [39] Pavlovi, M. (1984). A statically determinate truss model for thin shells: One-surface analysis (membrane hypothesis). *International Journal for Numerical Methods in Engineering* 20(10), 1841–1861.
- [40] Pellis, D., M. Kilian, F. Dellinger, J. Wallner, and H. Pottmann (2019). Visual smoothness of polyhedral surfaces. *ACM Transactions on Graphics* 38(4), 260:1–260:11. Proc. SIGGRAPH.
- [41] Pellis, D. and H. Pottmann (2018). Aligning principal stress and curvature directions. In L. Hesselgren et al. (Eds.), *Advances in Architectural Geometry 2018*. Klein Publishing.

- [42] Pottmann, H., A. Asperl, M. Hofer, and A. Kilian (2007). *Architectural geometry*. Exton: Bentley Institute Press.
- [43] Pottmann, H., M. Eigensatz, A. Vaxman, and J. Wallner (2015). Architectural geometry. *Computers & Graphics* 47, 145–164.
- [44] Pottmann, H. and Y. Liu (2007). Discrete surfaces in isotropic geometry. In R. Martin, M. Sabin, and J. Winkler (Eds.), *Mathematics of Surface XII*, Volume XII, pp. 431–363. Springer.
- [45] Pottmann, H., Y. Liu, J. Wallner, A. Bobenko, and W. Wang (2007). Geometry of multi-layer freeform structures for architecture. *ACM Transactions on Graphics* 26(3), 65:1–65:11. Proc. SIGGRAPH.
- [46] Prager, W. (1978). Nearly optimal design of trusses. *Computers & Structures* (8), 451–454.
- [47] Rozvany, G. I. N. (1996). Some shortcomings in Michells truss theory. *Structural Optimization* 12(4), 244–250.
- [48] Sadd, M. H. (2009). *Elasticity: theory, applications, and numerics*. Oxford: Academic Press.
- [49] Schiffner, A. and J. Balzer (2010). Statics-sensitive layout of planar quadrilateral meshes. In C. Ceccato et al. (Eds.), *Advances in Architectural Geometry 2010*, pp. 221–236. Springer.
- [50] Stein, O., E. Grinspun, and K. Crane (2018). Developability of triangle meshes. *ACM Transactions on Graphics* 37(4), 77:1–77:14.
- [51] Strubecker, K. (1962). Airy’sche Spannungsfunktion und isotrope Differentialgeometrie. *Mathematische Zeitschrift* 78, 189–198.
- [52] Sun, X. (2016). *Discrete Curvature Theories and Applications*. Ph. D. thesis, King Abdullah University of Science and Technology, Thuwal.
- [53] Suris, Y. B. and A. I. Bobenko (2008). *Discrete differential geometry: Integrable structure*. American Mathematical Society.

- [54] Tamstorf, R. and E. Grinspun (2013). Discrete bending forces and their Jacobians. *Graph. Models* 75, 362–370.
- [55] Tang, C., X. Sun, A. Gomes, J. Wallner, and H. Pottmann (2014). Form-finding with polyhedral meshes made simple. *ACM Transactions on Graphics* 33(4), 70:1–70:9.
- [56] Vouga, E., M. Höbinger, J. Wallner, and H. Pottmann (2012). Design of self-supporting surfaces. *ACM Transactions on Graphics* 31(4), 87:1–87:11.
- [57] Wächter, A. and L. T. Biegler (2006). On the implementation of a primal-dual interior point filter line search algorithm for large-scale non-linear programming. *Mathematical Programming* 106, 25–57.
- [58] Whittle, P. (2007). *Networks: Optimisation and Evolution*. Cambridge University Press.
- [59] Zhao, H. and G. Xu (2006). Triangular surface mesh fairing via gaussian curvature flow. *Journal of Computational and Applied Mathematics* 195, 300–311.