

Design of Self-supporting Surfaces

Abstract

Self-supporting masonry is one of the most ancient and at the same time most elegant ways of building curved shapes. Their analysis and modeling is a topic of geometry processing rather than classical continuum mechanics, because of the very geometric nature of failure of such structures. In this paper we use the thrust network method of analysis and present an iterative nonlinear optimization algorithm for efficiently approximating freeform shapes by self-supporting ones. This provides an interactive modeling tool for such shapes. The rich geometry of thrust networks which was first studied by Maxwell in the 1860s leads us to new viewpoints of discrete differential geometry: We find close connections between different objects such as a finite-element discretization of the Airy stress potential, perfect graph Laplacians, and computing admissible loads via curvatures of polyhedral surfaces. This geometric viewpoint shows us how to perform remeshing of a self-supporting shape by a self-supporting quad mesh with planar faces.



Figure 1: A surface with many, irregularly placed holes almost never stands by itself; those that do are surprising and their stability is not obvious by inspection. The surface shown is produced by our algorithm which finds, for a given freeform shape, the nearest self-supporting surface.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations;

Keywords: Discrete differential geometry, architectural geometry, self-supporting masonry, thrust networks, reciprocal force diagrams, discrete Laplace operators, isotropic geometry, mean curvature

1 Introduction

Vaulted masonry structures are among the simplest and at the same time most elegant solutions for creating curved shapes in building construction. This is the reason why they have been an object of interest since antiquity, large non-convex examples being provided by gothic cathedrals. They continue to be an active topic of research in today's engineering community.

Our paper is concerned with a combined geometry+statics analysis of *self-supporting* masonry and with tools for the interactive modeling of freeform self-supporting structures. Here “self-supporting” means that the structure, considered as an arrangement of blocks (bricks, stones), holds together by itself, and additional support, additional chains and similar are present only during construction. Our analysis is based on the following assumptions, which follow the classic [Heyman 1966]:

Assumption 1: Masonry has no tensile strength, but the individual building blocks do not slip against each other (because of friction

or mortar). On the other hand, their compressive strength is sufficiently high so that failure of the structure is by a sudden change in geometry, such as shown by Figure 2, and not by material failure.

Assumption 2 (The Safe Theorem): If a system of forces can be found which is in equilibrium with the load on the structure and which is contained within the masonry envelope then the structure will carry the loads, although the actually occurring forces may not be those postulated.

Our approach is twofold: We first give an overview of the continuous case of a smooth surface under stress which turns out to be governed by the so-called Airy stress function, at least locally. This mathematical model is called a membrane in the engineering literature and has been applied to the analysis of masonry before. The surface is self-supporting if and only if stresses are entirely compressive (i.e., the Airy function is convex). For computational purposes, stresses are discretized as a fictitious *thrust network* [Block and Ochsendorf 2007] contained in the masonry structure. This is a system of forces which together with the structure's deadload is in equilibrium. It can be interpreted as a finite element discretization of the continuous case, and it turns out to have very interesting geometry dating back to the work of J. C. Maxwell [1864], with the Airy stress function becoming a polyhedral surface directly related to a reciprocal force diagram.

While previous work in architectural geometry was mostly concerned with aspects of rationalization and purely geometric side-conditions which occur in freeform architecture, the focus of this paper is design with *statics* constraints. In particular, our contributions are the following:

Contributions.

- We connect the physics of self-supporting surfaces with vertical loads to the geometry of isotropic 3-space, with the direction of gravity as the distinguished direction (§2.3). Taking the convex Airy potential as unit sphere, one can express the equations governing self-supporting surfaces in terms of curvatures.
- We employ Maxwell's construction of polyhedral thrust networks and their reciprocal diagrams (§2.4), and give an interpre-

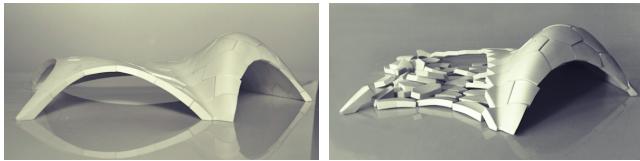


Figure 2: Masonry fails via geometric catastrophe rather than material failure (models by Block Research Group, ETH Zürich).

78 tation of the equilibrium conditions in terms of discrete curvatures

79 • The graph Laplacian derived from a thrust network with
80 compressive forces is a “perfect” one (§2.2). We show how it appears in
81 the analysis and establish a connection with mean curvatures which
82 are otherwise defined for polyhedral surfaces.

83 • We present an optimization algorithm for efficiently finding a
84 thrust network near a given arbitrary reference surface (§3), and
85 build a tool for interactive design of self-supporting surfaces based
86 on this algorithm (§4).

87 • We exploit the geometric relationships between a self-supporting
88 surface and the stress potential in order to find particularly nice
89 families of self-supporting surfaces, especially planar quadrilateral
90 representations of thrust networks (§5).

91 • We demonstrate the versatility and applicability of our approach
92 to the design and analysis of large-scale masonry and steel-glass
93 structures.

94 **Related Work.** Unsupported masonry has been an active topic of
95 research in the engineering community. The foundations for the
96 modern approach were laid by Jacques Heyman [1966] and are
97 available as the textbook [Heyman 1995]. A unifying view on
98 polyhedral surfaces, compressive forces and corresponding “con-
99 vex” force diagrams is presented by [Ash et al. 1988]. F. Frater-
100 nali [2002], [2010] established a connection between the continuous
101 theory of stresses in membranes and the discrete theory of
102 forces in thrust networks, by interpreting the latter as a certain non-
103 conforming finite element discretization of the former.

104 Several authors have studied the problem of finding discrete com-
105 pressive force networks contained within the boundary of masonry
106 structures; early work in this area includes [Schek 1974], [Livesley
107 1992], and [O’Dwyer 1998]. Fraternali [2010] proposed solving
108 for the structure’s discrete stress surface, and examining its convex
109 hull to study the structure’s stability and susceptibility to cracking.
110 Philippe Block’s seminal thesis introduced the method of *Thrust
111 Network Analysis*, which linearizes the form-finding problem by
112 first seeking a reciprocal diagram of the top view, which guarantees
113 equilibrium of horizontal forces, then solving for the heights that
114 balance the vertical loads (see e.g. [Block and Ochsendorf 2007;
115 Block 2009]). Recent work by Block and coauthors extends this
116 method in the case where the reciprocal diagram is not unique;
117 for different choices of reciprocal diagram, the optimal heights can
118 be found using the method of least squares [Van Mele and Block
119 2011], and the search for the best such reciprocal diagram can be
120 automated using a genetic algorithm [Block and Lachauer 2011].

121 Other approaches to the interactive design of self-supporting struc-
122 tures include modeling these structures as damped particle-spring
123 systems [Kilian and Ochsendorf 2005; Barnes 2009], and mirroring
124 the rich tradition in architecture of designing self-supporting
125 surfaces using hanging chain models [Heyman 1998]. Alterna-
126 tively, masonry structures can be represented by networks of rigid

127 blocks [Whiting et al. 2009], whose conditions on the structural fea-
128 sibility were incorporated into procedural modeling of buildings.

129 Algorithmic and mathematical methods relevant to this paper are
130 work on the geometry of quad meshes with planar faces [Glymph
131 et al. 2004; Liu et al. 2006], discrete curvatures for such meshes
132 [Pottmann et al. 2007; Bobenko et al. 2010], in particular curva-
133 tures in isotropic geometry [Pottmann and Liu 2007]. Schiftner and
134 Balzer [2010] discuss approximating a reference surface by quad
135 mesh with planar faces, whose layout is guided by statics proper-
136 ties of that surface.

137 2 Self-supporting Surfaces

138 2.1 The Continuous Theory

139 We are here modeling masonry as a surface given by a height field
140 $s(x, y)$ defined in some planar domain Ω . We assume that there are
141 vertical loads $F(x, y)$ — usually F represents the structure’s own
142 weight. By definition this surface is self-supporting, if and only if
143 there exists a field of compressive stresses which are in equilibrium
144 with the acting forces. This is equivalent to existence of a field
145 $M(x, y)$ of 2×2 symmetric positive semidefinite matrices satisfy-
146 ing

$$\text{div}(M\nabla s) = F, \quad \text{div } M = 0, \quad (1)$$

147 where the divergence operator $\text{div} \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = u_x + v_y$ is under-
148 stood to act on the columns of a matrix (see e.g. [Fraternali 2010],
149 [Giaquinta and Giusti 1985]).

150 The condition $\text{div } M = 0$ says that M is essentially the Hessian of
151 a real-valued function ϕ (the *Airy stress potential*): With the nota-
152 tion

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{pmatrix} \iff \widehat{M} = \begin{pmatrix} m_{22} & -m_{12} \\ -m_{12} & m_{11} \end{pmatrix}$$

153 it is clear that $\text{div } M = 0$ is an integrability condition for \widehat{M} , so
154 locally there is a potential ϕ with

$$\widehat{M} = \nabla^2 \phi, \quad \text{i.e.,} \quad M = \widehat{\nabla^2 \phi}.$$

155 If the domain Ω is simply connected, this relation holds globally.
156 Positive semidefiniteness of M (or equivalently of \widehat{M}) character-
157 izes *convexity* of the Airy potential ϕ . The Airy function enters
158 computations only by way of its derivatives, so global existence is
159 not an issue.

160 *Remark:* Stresses at boundary points depend on the way the sur-
161 face is anchored: A fixed anchor means no condition, but a free
162 boundary with outer normal vector \mathbf{n} means $\langle M\nabla s, \mathbf{n} \rangle = 0$.

163 **Stress Laplacian.** Note that $\text{div } M = 0$ yields $\text{div}(M\nabla s) =$
164 $\text{tr}(M\nabla^2 s)$, which we like to call $\Delta_\phi s$. The operator Δ_ϕ is sym-
165 metric. It is elliptic (as a Laplace operator should be) if and only if
166 M is positive definite, i.e., ϕ is strictly convex. The balance condition
167 (1) may be written as $\Delta_\phi s = F$.

168 2.2 Discrete Theory: Thrust Networks

169 We are discretizing a self-supporting surface by a mesh \mathcal{S} (see Fig-
170 ure 3). Loads are again vertical, and we discretize them as force
171 densities F_i associated with vertices \mathbf{v}_i . The load acting on this
172 vertex is then given by $F_i A_i$, where A_i is an area of influence (us-
173 ing a prime to indicate projection onto the xy plane, A_i is the area
174 of the Voronoi cell of \mathbf{v}'_i w.r.t. V'). We assume that stresses are

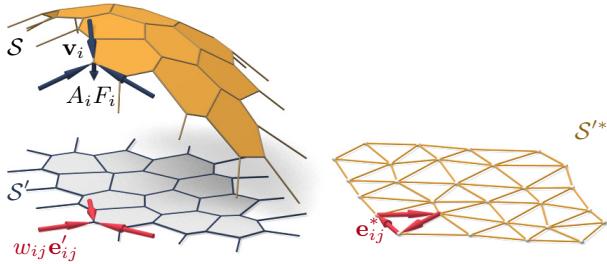


Figure 3: A thrust network \mathcal{S} , with dangling edges indicating external forces (left). This network together with compressive forces which balance vertical loads $A_i F_i$ projects onto a planar mesh \mathcal{S}' with equilibrium compressive forces $w_{ij} \mathbf{e}'_{ij}$ in its edges. Rotating forces by 90° leads to the reciprocal force diagram \mathcal{S}'^* (right).

175 carried by the edges of the mesh: the force exerted on the vertex \mathbf{v}_i
176 by the edge connecting $\mathbf{v}_i, \mathbf{v}_j$ is given by

$$w_{ij}(\mathbf{v}_j - \mathbf{v}_i), \quad \text{where } w_{ij} = w_{ji} \geq 0.$$

177 The nonnegativity of the individual weights w_{ij} expresses the
178 compressive nature of forces. The balance conditions at vertices then
179 read as follows: With $\mathbf{v}_i = (x_i, y_i, s_i)$ we have

$$\sum_{j \sim i} w_{ij}(x_j - x_i) = \sum_{j \sim i} w_{ij}(y_j - y_i) = 0, \quad (2)$$

$$\sum_{j \sim i} w_{ij}(s_j - s_i) = A_i F_i. \quad (3)$$

180 A mesh equipped with edge weights in this way is a discrete *thrust*
181 network. Invoking the safe theorem, we can state that a masonry
182 structure is self-supporting, if we can find a thrust network with
183 compressive forces which is entirely contained within the structure.

184 **Reciprocal Diagram.** Equations (2) have a geometric interpreta-
185 tion: With edge vectors

$$\mathbf{e}'_{ij} = \mathbf{v}'_j - \mathbf{v}'_i = (x_j, y_j) - (x_i, y_i),$$

186 Equation (2) asserts that vectors $w_{ij} \mathbf{e}'_{ij}$ form a closed cycle. Rotat-
187 ing them by 90 degrees, we see that likewise

$$\mathbf{e}'_{ij}^* = w_{ij} J \mathbf{e}'_{ij}, \quad \text{with } J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

188 form a closed cycle (see Figure 3). If the mesh \mathcal{S} is simply con-
189 nected, there exists an entire *reciprocal diagram* \mathcal{S}'^* which is a
190 combinatorial dual of \mathcal{S} , and which has edge vectors \mathbf{e}'_{ij}^* . Its ver-
191 tices are denoted by \mathbf{v}'_i^* .

192 *Remark:* If \mathcal{S}' is a Delaunay triangulation, then the corresponding
193 Voronoi diagram is an example of a reciprocal diagram.

194 **Polyhedral Stress Potential.** We can go further and construct
195 a convex polyhedral ‘Airy stress potential’ surface Φ with vertices
196 $\mathbf{w}_i = (x_i, y_i, \phi_i)$ combinatorially equivalent to \mathcal{S} by requiring that
197 a primal face of Φ lies in the plane $z = \alpha x + \beta y + \gamma$ if and only if
198 (α, β) is the corresponding dual vertex of \mathcal{S}'^* (see Figure 4). Ob-
199 viously this condition determines Φ up to vertical translation. For
200 existence see [Ash et al. 1988]. The inverse procedure constructs a
201 reciprocal diagram from Φ . This procedure obviously works also if
202 forces are not compressive: we can construct an Airy mesh Φ which
203 has planar faces, but it will no longer be a convex polyhedron.

204 The vertices of Φ can be interpolated by a piecewise-linear function
205 $\phi(x, y)$. It is easy to see that the derivative of $\phi(x, y)$ jumps by the
206 amount $\|\mathbf{e}'_{ij}^*\| = w_{ij} \|\mathbf{e}'_{ij}\|$, when crossing over the edge \mathbf{e}'_{ij} at right

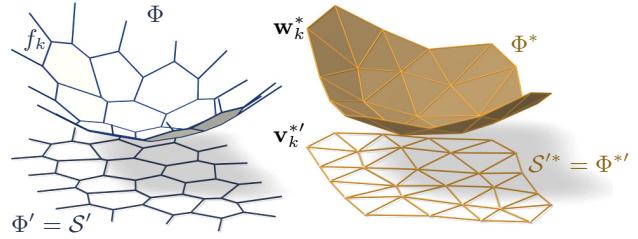


Figure 4: Airy stress potential Φ and its polar dual Φ^* . Φ projects onto the same planar mesh as \mathcal{S} does, while Φ^* projects onto the reciprocal force diagram. A primal face f_k lies in the plane $z = \alpha x + \beta y + \gamma \iff$ the corresponding dual vertex is $\mathbf{w}_k^* = (\alpha, \beta, -\gamma)$.

207 angle, with unit speed. This identifies Φ as the Airy polyhedron in-
208 troduced by [Fraterno et al. 2002] as a finite element discretization
209 of the continuous Airy function (see also [Fraterno 2010]).

210 If the mesh is not simply connected, the reciprocal diagram and the
211 Airy polyhedron exist only locally. Global existence is not an issue
212 for our computations.

213 **Polarity.** Polarity with respect to the *Maxwell paraboloid* $z =$
214 $\frac{1}{2}(x^2 + y^2)$ maps the plane $z = \alpha x + \beta y + \gamma$ to the point $(\alpha, \beta, -\gamma)$.
215 Thus, applying polarity to Φ and projecting the result Φ^* into the xy
216 plane reconstructs the reciprocal diagram $\Phi'^* = \mathcal{S}'^*$ (see Fig. 4).

217 **Discrete Stress Laplacian.** The weights w_{ij} may be used to define
218 a graph Laplacian Δ_ϕ which on vertex-based functions acts as

$$\Delta_\phi s(\mathbf{v}_i) = \sum_{j \sim i} w_{ij}(s_j - s_i).$$

219 This operator is a perfect discrete Laplacian in the sense of [War-
220 detzky et al. 2007], since it is symmetric by construction, Equa-
221 tion (2) implies linear precision for the planar ‘top view mesh’ \mathcal{S}'
222 (i.e., $\Delta_\phi f = 0$ if f is a linear function), and $w_{ij} \geq 0$ ensures
223 semidefiniteness and a maximum principle for Δ_ϕ -harmonic func-
224 tions. Equation (3) can be written as $\Delta_\phi s = AF$.

225 Note that Δ_ϕ is well defined also in case the underlying meshes are
226 not simply connected.

227 2.3 Surfaces in Isotropic Geometry

228 It is worth while to reconsider the basics of self-supporting surfaces
229 in the language of dual-isotropic geometry, which takes place in \mathbb{R}^3
230 with the z axis as a distinguished vertical direction. The basic ele-
231 ments of this geometry are planes, having equation $z = f(x, y) =$
232 $\alpha x + \beta y + \gamma$. The gradient vector $\nabla f = (\alpha, \beta)$ determines the
233 plane up to translation. A plane tangent to the graph of the function
234 $s(x, y)$ has gradient vector ∇s .

235 There is the notion of *parallel points*: $(x, y, z) \parallel (x', y', z') \iff$
236 $x = x', y = y'$.

237 *Remark:* The Maxwell paraboloid is considered the unit sphere of
238 isotropic geometry, and the geometric quantities considered above
239 are assigned a specific meaning: The forces $\|\mathbf{e}'_{ij}^*\| = w_{ij} \|\mathbf{e}'_{ij}\|$ are
240 dihedral angles of the Airy polyhedron Φ , and ‘lengths’ of edges
241 of Φ^* . We do not make use of this terminology in the sequel.

242 **Curvatures.** Generally speaking, in the differential geometry of
243 surfaces one considers the *Gauss map* σ from a surface S to a con-

vex unit sphere Φ by requiring that corresponding points have parallel tangent planes. Subsequently mean curvature H^{rel} and Gaussian curvature K^{rel} relative to Φ are computed from the derivative $d\sigma$. Classically Φ is the ordinary unit sphere $x^2 + y^2 + z^2 = 1$, so that σ maps each point its unit normal vector.

In our setting, parallelity is a property of *points* rather than planes, and the Gauss map σ goes the other way, mapping the tangent planes of the unit sphere $z = \phi(x, y)$ to the corresponding tangent plane of the surface $z = s(x, y)$. If we know which point a plane is attached to, then it is determined by its gradient. So we simply write

$$\nabla\phi \xrightarrow{\sigma} \nabla s.$$

By moving along a curve $\mathbf{u}(t) = (x(t), y(t))$ in the parameter domain we get the first variation of tangent planes: $\frac{d}{dt}\nabla\phi|_{\mathbf{u}(t)} = (\nabla^2\phi)\dot{\mathbf{u}} \xrightarrow{d\sigma} (\nabla^2s)\dot{\mathbf{u}}$. This yields the derivative $(\nabla^2\phi)\dot{\mathbf{u}} \xrightarrow{d\sigma} (\nabla^2s)\dot{\mathbf{u}}$, for all $\dot{\mathbf{u}}$, and the matrix of $d\sigma$ is found as $(\nabla^2\phi)^{-1}(\nabla^2s)$. By definition, curvatures of the surface s relative to ϕ are found as

$$K_s^{\text{rel}} = \det(d\sigma) = \frac{\det\nabla^2s}{\det\nabla^2\phi},$$

$$H_s^{\text{rel}} = \frac{1}{2}\text{tr}(d\sigma) = \frac{1}{2}\text{tr}\left(\frac{M}{\det\nabla^2\phi}\nabla^2s\right) = \frac{\Delta_\phi s}{2\det\nabla^2\phi}.$$

The Maxwell paraboloid $\phi_0(x, y) = \frac{1}{2}(x^2 + y^2)$ is the canonical unit sphere of isotropic geometry, its Hessian equals E_2 . Curvatures relative to ϕ_0 are not called “relative” and are denoted by the symbols H, K instead of $H^{\text{rel}}, K^{\text{rel}}$. The observation

$$\Delta_\phi\phi = \text{tr}(M\nabla^2\phi) = \text{tr}(\widehat{\nabla^2\phi}\nabla^2\phi) = 2\det\nabla^2\phi$$

together with the formulas above imply

$$K_s = \det\nabla^2s, K_\phi = \det\nabla^2\phi \implies H_s^{\text{rel}} = \frac{\Delta_\phi s}{2K_\phi} = \frac{\Delta_\phi s}{\Delta_\phi\phi}.$$

Relation to Self-supporting Surfaces. Summarizing the formulas above, we rewrite the balance condition (1) as

$$2K_\phi H_s^{\text{rel}} = \Delta_\phi s = F. \quad (4)$$

Let us draw some conclusions:

- Since $H_\phi^{\text{rel}} = 1$ we see that the load $F_\phi = 2K_\phi$ is admissible for the stress surface $\phi(x, y)$, which is hereby shown as self-supporting. The quotient of loads yields $H_s^{\text{rel}} = F/F_\phi$.
- If the stress surface coincides with the Maxwell paraboloid, then *constant loads characterize constant mean curvature surfaces*, because we get $K_\phi = 1$ and $H_s = F/2$.
- If s_1, s_2 have the same stress potential ϕ , then $H_{s_1-s_2}^{\text{rel}} = H_{s_1}^{\text{rel}} - H_{s_2}^{\text{rel}} = 0$, so $s_1 - s_2$ is a (relative) minimal surface.

2.4 Meshes in Isotropic Geometry

A general theory of curvatures of polyhedral surfaces with respect to a polyhedral unit sphere was proposed by [Pottmann et al. 2007; Bobenko et al. 2010], and its dual complement in isotropic geometry was elaborated by [Pottmann and Liu 2007]. As illustrated by Figure 5, the mean curvature of a self-supporting surface \mathcal{S} relative to its discrete Airy stress potential is associated with the vertices of \mathcal{S} . It is computed from areas and mixed areas of faces in the polar polyhedra \mathcal{S}^* and Φ^* :

$$H^{\text{rel}}(\mathbf{v}_i) = \frac{A_i(\mathcal{S}, \Phi)}{A_i(\Phi, \Phi)}, \quad \text{where}$$

$$A_i(\mathcal{S}, \Phi) = \frac{1}{4} \sum_{k: f_k \in 1\text{-ring}(\mathbf{v}_i)} \det(\mathbf{v}'_k, \mathbf{w}'_{k+1}) + \det(\mathbf{w}'_k, \mathbf{v}'_{k+1}).$$

The prime denotes the projection into the xy plane, and summation is over those dual vertices which are adjacent to \mathbf{v}_i . Replacing \mathbf{v}'_k by \mathbf{w}'_k yields $A_i(\Phi, \Phi) = \frac{1}{2} \sum \det(\mathbf{w}'_k, \mathbf{w}'_{k+1})$.

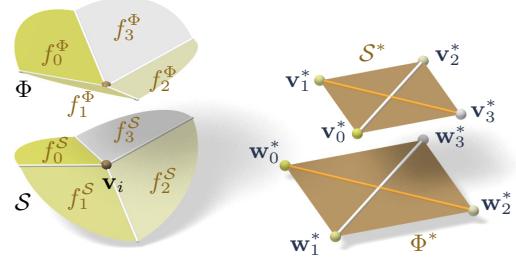


Figure 5: Mean curvature of a vertex \mathbf{v}_i of \mathcal{S} : Corresponding edges of the polar duals \mathcal{S}^* , Φ^* are parallel, and mean curvature according to [Pottmann et al. 2007] is computed from the vertices polar to faces adjacent to \mathbf{v}_i . For valence 4 vertices the case of zero mean curvature shown here is characterized by parallelity of non-corresponding diagonals of corresponding quads in \mathcal{S}^* , Φ^* .

Proposition. If Φ is the Airy surface of a thrust network \mathcal{S} , then the mean curvature of \mathcal{S} relative to Φ is computable as

$$H^{\text{rel}}(\mathbf{v}_i) = \frac{\sum_{j \sim i} w_{ij}(s_j - s_i)}{\sum_{j \sim i} w_{ij}(\phi_j - \phi_i)} = \frac{\Delta_\phi s}{\Delta_\phi\phi}|_{\mathbf{v}_i}. \quad (5)$$

Proof. It is sufficient to show $2A_i(\mathcal{S}, \Phi) = \sum_{j \sim i} w_{ij}(s_j - s_i)$.

For that, consider edges $\mathbf{e}'_1, \dots, \mathbf{e}'_n$ emanating from \mathbf{v}'_i . The dual cycles in Φ^* and \mathcal{S}^* without loss of generality are given by vertices $(\mathbf{v}'_1, \dots, \mathbf{v}'_n)$ and $(\mathbf{w}'_1, \dots, \mathbf{w}'_n)$, respectively. The latter has edges $\mathbf{w}'_{j+1} - \mathbf{w}'_j = w_{ij}\mathbf{J}\mathbf{e}'_j$ (indices modulo n).

Without loss of generality $\mathbf{v}_i = 0$, so the vertex \mathbf{v}'_i by construction equals the gradient of the linear function $\mathbf{x} \mapsto \langle \mathbf{v}'_i, \mathbf{x} \rangle$ defined by the properties $\mathbf{e}'_{j-1} \mapsto s_{j-1} - s_i$, $\mathbf{e}'_j \mapsto s_j - s_i$. Corresponding edge vectors $\mathbf{v}'_{j+1} - \mathbf{v}'_j$ and $\mathbf{w}'_{j+1} - \mathbf{w}'_j$ are parallel, because $\langle \mathbf{v}'_{j+1} - \mathbf{v}'_j, \mathbf{e}'_j \rangle = (s_j - s_i) - (s_j - s_i) = 0$. Expand $2A_i(\mathcal{S}, \Phi)$:

$$\begin{aligned} & \frac{1}{2} \sum \det(\mathbf{w}'_j, \mathbf{v}'_{j+1}) + \det(\mathbf{v}'_j, \mathbf{w}'_{j+1}) \\ &= \frac{1}{2} \sum \det(\mathbf{w}'_j - \mathbf{w}'_{j+1}, \mathbf{v}'_{j+1}) + \det(\mathbf{v}'_j, \mathbf{w}'_{j+1} - \mathbf{w}'_j) \\ &= \frac{1}{2} \sum \det(-w_{ij}\mathbf{J}\mathbf{e}'_j, \mathbf{v}'_{j+1}) + \det(\mathbf{v}'_j, w_{ij}\mathbf{J}\mathbf{e}'_j) \\ &= \sum \det(\mathbf{v}'_j, w_{ij}\mathbf{J}\mathbf{e}'_j) = \sum w_{ij}\langle \mathbf{v}'_j, \mathbf{e}'_j \rangle = \sum w_{ij}(s_j - s_i). \end{aligned}$$

Here we have used $\det(\mathbf{a}, J\mathbf{b}) = \langle \mathbf{a}, \mathbf{b} \rangle$. \square

In order to discretize (4), we also need a discrete Gaussian curvature, which is usually defined as a quotient of areas which correspond under the Gauss mapping. We define

$$K_\Phi(\mathbf{v}_i) = \frac{A_i(\Phi, \Phi)}{A_i},$$

where A_i is the Voronoi area of vertex \mathbf{v}'_i in the projected mesh \mathcal{S}' used in (3).

Remark: If the faces of the thrust network \mathcal{S} are not planar, the simple trick of introducing additional edges with zero forces in them makes it planar, and the theory is applicable. We refrain from elaborating this further.

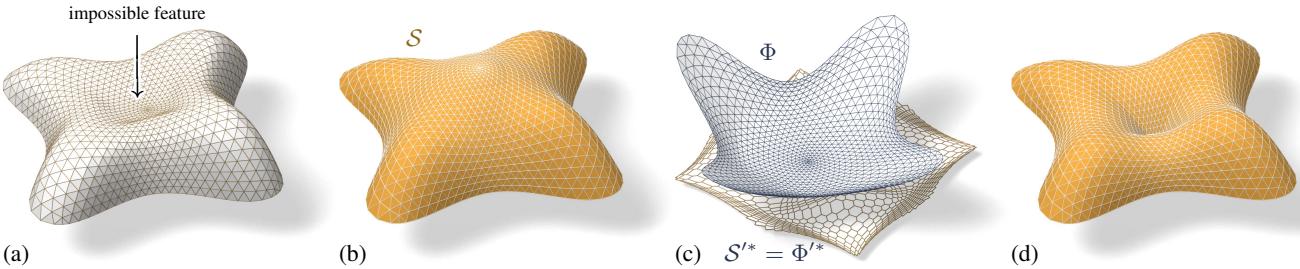


Figure 6: The top of the Lilium Tower (a) cannot stand as a masonry structure, because its central part is concave. Our algorithm finds a nearby self-supporting mesh (b) without this impossible feature. (c) shows the corresponding Airy mesh Φ and reciprocal force diagram S'^* . (d) The user can edit the original surface, such as by specifying that the center of the surface is supported by a vertical pillar; and the self-supporting network adjusts accordingly

310 **Discrete Balance Equation.** The discrete version of the balance
311 equation (4) reads as follows:

312 **Theorem.** A simply-connected mesh \mathcal{S} with vertices $\mathbf{v}_i =$
313 (x_i, y_i, s_i) can be put into static equilibrium with vertical forces
314 “ $A_i F_i$ ” if and only if there exists a combinatorially equivalent
315 mesh Φ with planar faces and vertices (x_i, y_i, ϕ_i) , such that cur-
316 vatures of \mathcal{S} relative to Φ obey

$$2K_\Phi(\mathbf{v}_i)H^{\text{rel}}(\mathbf{v}_i) = F_i \quad (6)$$

317 at every interior vertex and every free boundary vertex \mathbf{v}_i . \mathcal{S} can
318 be put into compressive static equilibrium if and only if there exists
319 a convex such Φ .

320 *Proof.* The relation between equilibrium forces $w_{ij}\mathbf{e}_{ij}$ in \mathcal{S} and
321 the polyhedral stress potential Φ has been discussed above, and
322 so has the equivalence “ $w_{ij} \geq 0 \iff \Phi$ convex” (see e.g.
323 [Ash et al. 1988] for a survey of this and related results). It re-
324 mains to show that Equations (2) and (6) are equivalent. This is
325 the case because the proposition above implies $2K(\mathbf{v}_i)H^{\text{rel}}(\mathbf{v}_i) =$
326 $2\frac{A_i(\Phi, \Phi)}{A_i} \frac{A_i(\Phi, \mathcal{S})}{A_i(\Phi, \Phi)} = \frac{1}{A_i} (\sum_{j \sim i} w_{ij}(s_j - s_i)) = \frac{1}{A_i} A_i F_i$. \square

327 **Existence of Discretizations.** When considering discrete thrust
328 networks as discretizations of continuous self-supporting surfaces,
329 the following question is important: For a given smooth surface
330 $s(x, y)$ with Airy stress function ϕ , does there exist a polyhedral
331 surface \mathcal{S} in equilibrium approximating $s(x, y)$, whose top view
332 is a given planar mesh \mathcal{S}' ? We restrict our attention to triangle
333 meshes, where planarity of the faces of the discrete stress surface Φ
334 is not an issue. This question has several equivalent reformulations:

- 335 • Does \mathcal{S}' have a reciprocal diagram whose corresponding Airy
336 polyhedron Φ approximates the continuous Airy potential ϕ ?
337 (if the surfaces involved are not simply connected, these ob-
338 jects are defined locally).
- 339 • Does \mathcal{S}' possess a “perfect” discrete Laplace-Beltrami operator
340 Δ_ϕ in the sense of Wardetzky et al. [2007] whose weights
341 are the edge length scalars of such a reciprocal diagram?

342 From [Wardetzky et al. 2007] we know that perfect Laplacians ex-
343 ist only on regular triangulations which are projections of convex
344 polyhedra. On the other hand, previous sections show how to ap-
345 propriately re-triangulate: Let Φ be a triangle mesh convex hull of
346 the vertices $(x_i, y_i, \phi(x_i, y_i))$, where (x_i, y_i) are vertices of \mathcal{S}' .
347 Then its polar dual Φ^* projects onto a reciprocal diagram with pos-
348 itive edge weights, so Δ_ϕ has positive weights, and the vertices
349 (x_i, y_i, s_i) of \mathcal{S} can be found by solving the discrete Poisson prob-
350 lem $(\Delta_\phi s)_i = A_i F_i$, which yields a mesh approximating $s(x, y)$.

351 Assuming the discrete operator Δ_ϕ approximates its continuous
352 counterpart, we conclude: A smooth self-supporting surface can

353 be approximated by a discrete self-supporting triangular mesh for
354 any sampling of the surface.

3 Thrust Networks from Reference Meshes

356 Consider now the problem of taking a given reference mesh, say
357 \mathcal{R} , and finding a combinatorially equivalent mesh \mathcal{S} in static equi-
358 librium approximating \mathcal{R} . The loads on \mathcal{S} include user-prescribed
359 loads as well as the dead load caused by the mesh’s own weight.
360 Conceptually, finding \mathcal{S} amounts to minimizing some formulation
361 of distance between \mathcal{R} and \mathcal{S} , subject to constraints (2), (3), and
362 $w_{ij} \geq 0$. For any choice of distance this minimization will be a
363 nonlinear, non-convex, inequality-constrained variational problem
364 that cannot be efficiently solved in practice. Instead we propose a
365 staggered optimization algorithm:

- 366 0. Start with an initial guess $\mathcal{S} = \mathcal{R}$.
- 367 1. Estimate the self-load on the vertices of \mathcal{S} , using their current
368 positions.
- 369 2. Fixing \mathcal{S} , fit an associated stress surface Φ .
- 370 3. Alter positions \mathbf{v}_i to improve the fit.
- 371 4. Repeat from Step 1 until convergence.

372 **Step 1: Estimating Self-Load.** The dead load due to the sur-
373 face’s own weight depends not only on the top view of \mathcal{S} , but also
374 on the surface area of its faces. To avoid adding nonlinearity to the
375 algorithm, we estimate the load coefficients F_i at the beginning of
376 each iteration, and assume they remain constant until the next iter-
377 ation. We estimate the load “ $A_i F_i$ ” associated with each vertex by
378 calculating its Voronoi area on each of its incident faces, and then
379 multiplying by a user-specified surface density ρ .

380 **Step 2: Fit a Stress Surface.** In this step, we fix \mathcal{S} and try to
381 fit a stress surface Φ subordinate to the top view \mathcal{S}' of the primal
382 mesh. We do so by searching for dihedral angles between the faces
383 of Φ which minimize, in the least-squares sense, the error in force
384 equilibrium (6) and local integrability of Φ . Doing so is equivalent
385 to minimizing the squared residuals of Equations (3) and (2), re-
386 spectively, with the positions held fixed. Defining the *equilibrium*
387 energy

$$E = \sum_i \left\| \begin{pmatrix} 0 \\ 0 \\ A_i F_i \end{pmatrix} - \sum_{j \sim i} w_{ij}(\mathbf{v}_j - \mathbf{v}_i) \right\|^2 \quad (7)$$

388 where the outer sum is over the interior and free boundary vertices,
389 we solve

$$\min_{w_{ij}} E, \quad \text{s.t. } 0 \leq w_{ij} \leq w_{\max}. \quad (8)$$

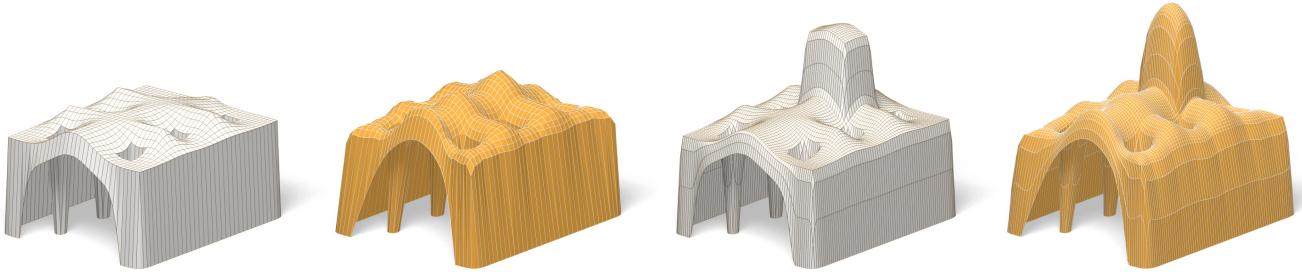


Figure 7: The user-designed reference mesh (left) is not self-supporting, but our algorithm finds a nearby perturbation of the reference surface (middle-left) that is in equilibrium. As the user makes edits to the reference surface (middle-right), the thrust network automatically adjusts (right).

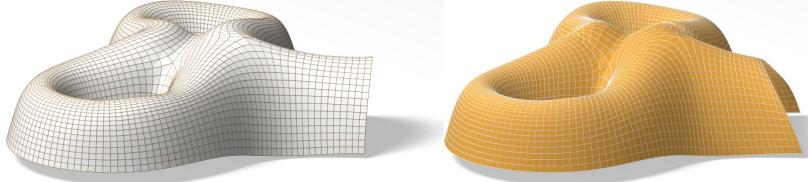


Figure 8: A freeform surface (left) needs adjustments around the entrance arch and between the two pillars in order to be self-supporting; our algorithm finds the nearby surface in equilibrium (right) that incorporates these changes.

390 Here w_{\max} is an optional maximum weight we are willing to assign
 391 (to limit the amount of stress in the surface). This convex, sparse,
 392 box-constrained least-squares problem [Friedlander 2007] always
 393 has a solution. If the objective is 0 at this solution, the faces of Φ
 394 locally integrate to a stress surface satisfying (6), and so Φ certifies
 395 that \mathcal{S} is self-supporting – we are done. Otherwise, \mathcal{S} is not self-
 396 supporting and its vertices must be moved.
 397

Step 3: Alter Positions. In the previous step we fit as best as possible a stress surface Φ to \mathcal{S} . There are two possible kinds of error with this fit: the faces around a vertex (equivalently, the reciprocal diagram) might not close up; and the resulting stress forces might not be exactly in equilibrium with the loads. These errors can be decreased by modifying the top view and heights of \mathcal{S} , respectively. It is possible to simply solve for new vertex positions that put \mathcal{S} in static equilibrium, since Equations (2) and (3) with w_{ij} fixed form a square linear system that is typically nonsingular.

406 While this approach would yield a self-supporting \mathcal{S} , this mesh is often far from the reference mesh \mathcal{R} , since any local errors in the 407 stress surface from Step 2 amplify into global errors in \mathcal{S} . We propose instead to look for new positions that decrease the imbalance 408 in the stresses and loads, while also penalizing drift away from the 409 reference mesh:
 410

$$\min_{\mathbf{v}} E + \alpha \sum_i \langle \mathbf{n}_i, \mathbf{v}_i - \mathbf{v}_i^0 \rangle^2 + \beta \|\mathbf{v} - \mathbf{v}_P^0\|^2,$$

412 where \mathbf{v}_i^0 is the position of the i -th vertex at the start of this step
 413 of the optimization, \mathbf{n}_i is the starting vertex normal (computed as
 414 the average of the incident face normals), \mathbf{v}_P^0 is the projection of \mathbf{v}^0
 415 onto the reference mesh, and $\alpha > \beta$ are penalty coefficients that are
 416 decreased every iteration of Steps 1–3 of the algorithm. The second
 417 term allows \mathcal{S} to slide over itself (if doing so improves equilibrium)
 418 but penalizes drift in the normal direction. The third term, weaker
 419 than the second, regularizes the optimization by preventing large
 420 drift away from the reference surface or excessive tangential sliding.
 421

422 **Implementation Details.** Solving the weighted least-squares
 423 problem of Step 3 amounts to solving a sparse, symmetric linear
 424 system. While the MINRES algorithm [Paige and Saunders 1975]
 425 is likely the most robust algorithm for solving this system, in prac-

426 tice we have observed that the method of conjugate gradients works
 427 well despite the potential ill-conditioning of the objective matrix.

428 **Limitations.** This algorithm is not guaranteed to always converge;
 429 this fact is not surprising from the physics of the problem
 430 (if the boundary of the reference mesh encloses too large of a region,
 431 w_{\max} is set too low, and the density of the surface too high,
 432 a thrust network in equilibrium simply does not exist – the vault is
 433 too ambitious and cannot be built to stand; pillars are needed.)

434 We can, however, make a few remarks. Step 2 always decreases the
 435 equilibrium energy E of Equation (7) and Step 3 does as well as
 436 $\beta \rightarrow 0$. Moreover, as $\alpha \rightarrow 0$ and $\beta \rightarrow 0$, Step 3 approaches a lin-
 437 ear system with as many equations as unknowns; if this system has
 438 full rank, its solution sets $E = 0$. These facts suggest that the algo-
 439 rithm should generally converge to a thrust network in equilibrium,
 440 provided that Step 1 does not increase the loads by too much at ev-
 441 ery iteration, and this is indeed what we observe in practice. One
 442 case where this assumption is guaranteed to hold is if the thickness
 443 of the surface is allowed to freely vary, so that it can be chosen so
 444 that the surface has uniform density over the top view.

445 If the linear system in Step 3 is singular and infeasible, the algo-
 446 rithm can stall at $E > 0$. This failure occurs, for instance, when
 447 an interior vertex has height z_i lower than all of its neighbors, and
 448 Step 2 assigns all incident edges to that vertex a weight of zero:
 449 clearly no amount of moving the vertex or its neighbors can bring
 450 the vertex into equilibrium. We avoid such degenerate configura-
 451 tions by bounding weights slightly away from zero in (8), trading
 452 increased robustness for slight smoothing of the resulting surface.

4 Results

454 **Interactive Design of Self-Supporting Surfaces.** The opti-
 455 mization algorithm described in the previous section forms the ba-
 456 sis of an interactive design tool for self-supporting surfaces. Users
 457 manipulate a mesh representing a reference surface, and the com-
 458 puter searches for a nearby thrust network in equilibrium (see e.g.
 459 Figure 7). Fitting this thrust network does not require that the user
 460 specify boundary tractions, and although the top view of the refer-
 461 ence mesh is used as an initial guess for the top view of the thrust
 462 network, the search is not restricted to this top view. The features
 463 of the design tool include:

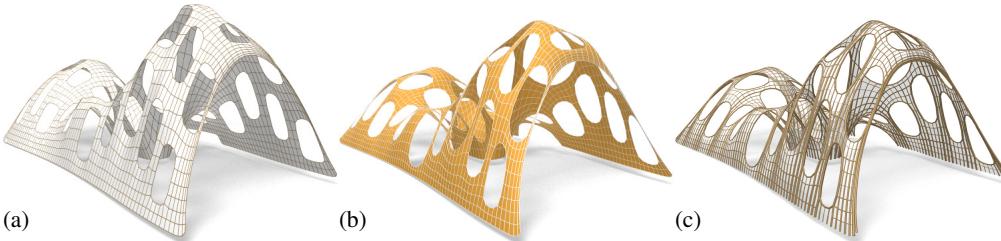


Figure 9: A mesh with holes (a) requires large deformations to both the top view and heights to render it self-supporting (b). The magnitude of forces in the edges of this thrust-network is visualized by the cross-section of edges in (c).

- Handle-based 3D editing of the reference mesh using Laplacian coordinates [Lipman et al. 2004; Sorkine et al. 2003] to extrude vaults, insert pillars, and apply other deformations to the reference mesh. Handle-based adjustments of the heights, keeping the top view fixed, and deformation of the top view, keeping the heights fixed, are also supported. The thrust network adjusts interactively to fit the deformed positions, giving the usual visual feedback about the effects of her edits on whether or not the surface can stand.
- Specification of boundary conditions. Points of contact between the reference surface and the ground or environment are specified by “pinning” vertices of the surface, specifying that the thrust network must coincide with the reference mesh at this point, and relaxing the condition that forces must be in equilibrium there.
- Interactive adjustment of surface density ρ , external loads, and maximum permissible stress per edge w_{\max} , with visual feedback of how these parameters affect the fitted thrust network.
- Upsampling of the thrust network through Catmull-Clark subdivision and polishing of the resulting refined thrust network using optimization (§3).
- Visualization of the stress surface \mathcal{R} dual to the thrust network and corresponding reciprocal diagram.

Example: Vault with Pillars. As an example of the design and optimization workflow, consider a rectangular vault with six pillars, free boundary conditions along one edge, fixed boundary conditions along the others, and a tower extruded from the top of the surface (see Figure 7). This surface is neither convex nor simply connected, and exhibits a mix of boundary conditions, none of which cause our algorithm any difficulty; it finds a self-supporting thrust network near the designed reference mesh. The user is now free to make edits to the reference mesh, and the thrust network adapts to these edits, providing the user feedback on whether these designs are physically realizable.

Example: Top of the Lilium Tower. Consider the top portion of the steel-glass exterior surface of the Lilium Tower, which is currently being built in Warszaw (see Figure 6). This surface contains a concave part with local minimum in its interior and so cannot possibly be self-supporting. Given this surface as a reference mesh, our algorithm constructs a nearby thrust network in equilibrium without the impossible feature. The user can then explore how editing the reference mesh – adding a pillar, for example – affects the thrust network and its deviation from the reference surface.

Example: Freeform Structure with Two Pillars. Suppose an architect’s experience and intuition has permitted the design a freeform surface (see Figure 8) that is nearly self-supporting. Our algorithm reveals those edits needed to make the structure sound – principally around the entrance arch, and the area between the two pillars.

Example: Swiss Cheese. Cutting holes in a self-supporting surface interrupts force flow lines and causes dramatic global changes to the surface stresses, often to the point that the surface is no longer in equilibrium. Whether a given surface with many such holes can stand is far from obvious. Figures 9 show such an implausible and unstable surface; our optimization finds a nearby, equally implausible but stable surface without difficulty (see Figures 1 and 9, right).

5 Special Self-Supporting Surfaces

PQ Meshes. Meshes with *planar* faces are of particular interest in architecture, so in this section we discuss how to remesh a given thrust network in equilibrium such that it becomes a quad mesh with planar faces (again in equilibrium). If this mesh is realized as a steel-glass construction, it is self-supporting in its beams alone, with no forces exerted on the glass (this is the usual manner of using glass). The beams constitute a self-supporting structure which is in perfect force equilibrium (without moments in the nodes) if only the deadload is applied.

For this purpose we first demonstrate how to find a quad mesh \mathcal{S} with vertices $\mathbf{v}_{ij} = (x_{ij}, y_{ij}, s_{ij})$ which approximates a given continuous surface $s(x, y)$ equipped with an equilibrium stress potential $\phi(x, y)$.

It is known that \mathcal{S} must approximately follow a network of conjugate curves in the surface (see e.g. [Liu et al. 2006]). We can derive this condition in an elementary way as follows: Using a Taylor expansion, we compute the volume of the convex hull of the quadrilateral $\mathbf{v}_{ij}, \mathbf{v}_{i+1,j}, \mathbf{v}_{i+1,j+1}, \mathbf{v}_{i,j+1}$, assuming the vertices lie exactly on the surface $s(x, y)$. This results in

$$\text{vol} = \frac{1}{6} \det(\mathbf{a}_1, \mathbf{a}_2, (\mathbf{a}_1)^T \nabla^2 s \mathbf{a}_2) + \dots,$$

$$\text{where } \mathbf{a}_1 = \begin{pmatrix} x_{i+1,j} - x_{ij} \\ y_{i+1,j} - y_{ij} \end{pmatrix}, \quad \mathbf{a}_2 = \begin{pmatrix} x_{i,j+1} - x_{ij} \\ y_{i,j+1} - y_{ij} \end{pmatrix},$$

and the dots indicate higher order terms. We see that planarity requires $(\mathbf{a}_1)^T \nabla^2 s \mathbf{a}_2 = 0$. In addition to the mesh \mathcal{S} approximating the surface $s(x, y)$, the corresponding polyhedral Airy surface Φ must approximate $\phi(x, y)$; thus we get the conditions

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

$\mathbf{a}_1, \mathbf{a}_2$ are therefore eigenvectors of $(\nabla^2 \phi)^{-1} \nabla^2 s$. In view of §2.3, $\mathbf{a}_1, \mathbf{a}_2$ indicate the principal directions of the surface $s(x, y)$ relative to $\phi(x, y)$ (see Figure 10).

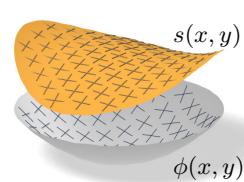


Figure 10: Planar quad remeshing of a self-supporting surface $s(x, y)$ with stress potential ϕ is guided by the principal curvature directions of s relative to ϕ (found from eigenvectors of $(\nabla^2 \phi)^{-1} \nabla^2 s$).

In the discrete case, where s, ϕ are not given as continuous surfaces, but are represented by a mesh in equilibrium and its Airy

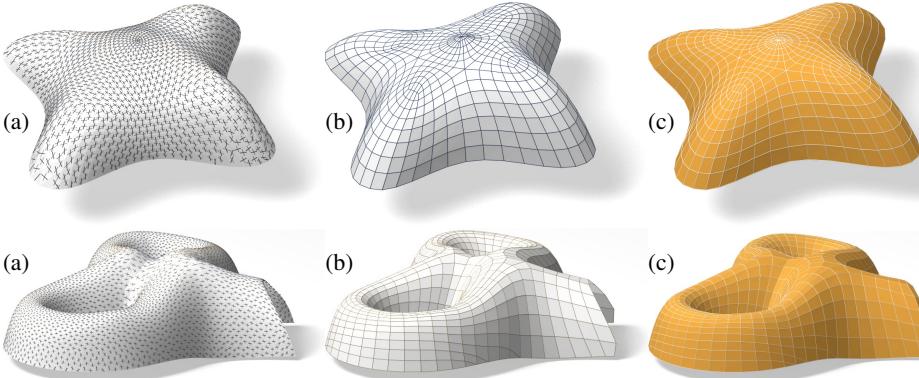


Figure 11: Planar quad remeshing of the “Lilium tower” surface of Figure 6. (b) Principal directions. (c) Quad mesh guided by principal directions is close to planar and close to self-supporting. (c) Small changes make it self-supporting.

mesh, we use the techniques of Schiftner [2007] and Cohen-Steiner and Morvan [2003] to approximate the Hessians $\nabla^2 s$, $\nabla^2 \phi$, compute principal directions as eigenvectors of $(\nabla^2 \phi)^{-1} \nabla^2 s$, and subsequently find meshes \mathcal{S} , Φ approximating s , ϕ which follow those directions. Global optimization now makes \mathcal{S} , Φ a valid thrust network with discrete stress potential. Convexity of Φ ensures that \mathcal{S} is self-supporting.

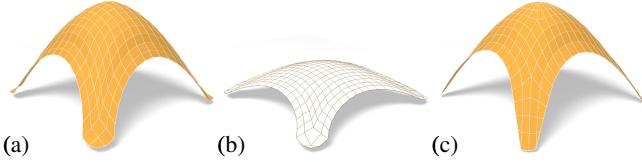


Figure 13: Directly enforcing planarity of the faces of even a very simple self-supporting quad-mesh vault (a) results in a surface far removed from the original design (b). Starting instead from a remeshing of the surface with edges following relative principal curvature directions yields a self-supporting, PQ mesh far more faithful to the original (c).

581 So \mathcal{R} is a *minimal surface* relative to Φ . While in the triangle mesh
 582 case there are enough degrees of freedom for nontrivial solutions,
 583 the case of planar quad meshes is more intricate: Polar polyhedra
 584 \mathcal{R}^* , Φ^* have to be Christoffel duals of each other [Pottmann and
 585 Liu 2007], as illustrated by Figure 5. Unfortunately not all quad
 586 meshes have such a dual; the condition is that the mesh is *Koenigs*,
 587 i.e., the derived mesh formed by the intersection points of diagonals
 588 of faces again has planar faces [Bobenko and Suris 2008].



Figure 14: A ‘Koebe’ mesh Φ is self-supporting for unit dead load. An entire family of self-supporting meshes with the same top view is defined by $\mathcal{S}_\alpha = \Phi + \alpha \mathcal{R}$, where \mathcal{R} is chosen as Φ ’s Christoffel-dual.

589 **Koebe meshes.** An interesting special case occurs if Φ is a
 590 *Koebe mesh* of isotropic geometry, i.e., a PQ mesh whose edges
 591 touch the Maxwell paraboloid. Since Φ approximates the Maxwell
 592 paraboloid, we get $2K(\mathbf{v}_i)H^{\text{rel}}(\mathbf{v}_i) \approx 1$ and Φ consequently is
 593 self-supporting for unit load. Applying the Christoffel dual con-
 594 struction described above yields a minimal mesh \mathcal{R} and a family of
 595 meshes $\Phi + \alpha \mathcal{R}$ which are self-supporting for unit load (see Fig-
 596 ure 14).

6 Conclusion and Future Work

598 **Conclusion.** This paper builds on relations between statics and
 599 geometry, some of which have been known for a long time, and
 600 connects them with newer methods of discrete differential geom-
 601 etry, such as discrete Laplace operators and curvatures of polyhedral
 602 surfaces. We were able to find efficient ways of modeling self-sup-
 603 porting freeform shapes, and provide architects and engineers with
 604 an interactive tool which gives quick information on the statics of
 605 freeform geometries. The self-supporting property of a shape is di-
 606 rectly relevant for freeform masonry. The actual thrust networks we
 607 use for computation are relevant e.g. for steel constructions, where
 608 equilibrium of deadload forces implies absence of moments. This
 609 theory and accompanying algorithms thus constitute a new contribu-
 610 tion to architectural geometry, connecting statics and geometric
 611 design.

612 **Future Work.** There are several obvious directions of future re-
 613 search. One is to incorporate non-manifold meshes, which occur

$$\Delta_\phi r = 0, \quad \text{i.e.,} \quad H_{\mathcal{R}}^{\text{rel}} = 0.$$

naturally when e.g. supporting walls are introduced. It is also obvious that non-vertical loads, e.g. wind load, play a role. There are also some directions to pursue in improving the algorithms, for instance adaptive remeshing in problem areas. Probably the interesting connections between statics properties and geometry are not yet exhausted, and we would like to propose the *geometrization* of problems as a strategy for their solution.

References

- ASH, P., BOLKER, E., CRAPO, H., AND WHITELEY, W. 1988. Convex polyhedra, Dirichlet tessellations, and spider webs. In *Shaping space (Northampton, Mass., 1984)*. Birkhäuser, Boston, 231–250.
- BARNES, M. R. 2009. Form finding and analysis of tension structures by dynamic relaxation. *International Journal of Space Structures* 14, 2, 89–104.
- BLOCK, P., AND LACHAUER, L. 2011. Closest-fit, compression-only solutions for free form shells. In *IABSE — IASS 2011 London Symposium*, Int. Ass. Shell Spatial Structures. electronic.
- BLOCK, P., AND OCHSENDORF, J. 2007. Thrust network analysis: A new methodology for three-dimensional equilibrium. *J. Int. Assoc. Shell and Spatial Structures* 48, 3, 167–173.
- BLOCK, P. 2009. *Thrust Network Analysis: Exploring Three-dimensional Equilibrium*. PhD thesis, Massachusetts Institute of Technology.
- BOBENKO, A., AND SURIS, YU. 2008. *Discrete differential geometry: Integrable Structure*. American Math. Soc.
- BOBENKO, A., POTTMANN, H., AND WALLNER, J. 2010. A curvature theory for discrete surfaces based on mesh parallelity. *Math. Annalen* 348, 1–24.
- COHEN-STEINER, D., AND MORVAN, J.-M. 2003. Restricted Delaunay triangulations and normal cycle. In *Proc. 19th Symp. Computational geometry*, ACM, 312–321.
- FRATERNALI, F., ANGELILLO, M., AND FORTUNATO, A. 2002. A lumped stress method for plane elastic problems and the discrete-continuum approximation. *Int. J. Solids Struct.* 39, 6211–6240.
- FRATERNALI, F. 2010. A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions. *Mechanics Res. Comm.* 37, 2, 198 – 204.
- FRIEDLANDER, M. P., 2007. BCLS: Bound constrained least squares. <http://www.cs.ubc.ca/~mpf/bcls>.
- GIAQUINTA, M., AND GIUSTI, E. 1985. Researches on the equilibrium of masonry structures. *Archive for Rational Mechanics and Analysis* 88, 4, 359–392.
- GLYMPH, J., SHELDEN, D., CECCATO, C., MUSSEL, J., AND SCHOBER, H. 2004. A parametric strategy for free-form glass structures using quadrilateral planar facets. *Automation in Construction* 13, 2, 187 – 202.
- HEYMAN, J. 1966. The stone skeleton. *Int. J. Solids Structures* 2, 249–279.
- HEYMAN, J. 1995. *The Stone Skeleton: Structural Engineering of Masonry Architecture*. Cambridge University Press.
- HEYMAN, J. 1998. *Structural Analysis: A Historical Approach*. Cambridge University Press.
- KILIAN, A., AND OCHSENDORF, J. 2005. Particle-spring systems for structural form finding. *J. Int. Assoc. Shell and Spatial Structures* 46, 77–84.
- LIPMAN, Y., SORKINE, O., COHEN-OR, D., LEVIN, D., ROSSI, C., AND SEIDEL, H. 2004. Differential coordinates for interactive mesh editing. In *Proc. SMI*. IEEE, 181–190.
- LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph.* 25, 3, 681–689.
- LIVESLEY, R. K. 1992. A computational model for the limit analysis of three-dimensional masonry structures. *Meccanica* 27, 161–172.
- MAXWELL, J. 1864. On reciprocal diagrams and diagrams of forces. *Philosophical Magazine* 4, 27, 250–261.
- O'Dwyer, D. 1998. Funicular analysis of masonry vaults. *Computers and Structures* 73, 187–197.
- PAIGE, C. C., AND SAUNDERS, M. A. 1975. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis* 12, 617–629.
- POTTMANN, H., AND LIU, Y. 2007. Discrete surfaces in isotropic geometry. In *Mathematics of Surfaces XII*, M. Sabin and J. Winkler, Eds. Springer-Verlag, 341–363.
- POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graphics* 26, 3, #65,1–11.
- SCHEK, H.-J. 1974. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mathematics and Engineering* 3, 115–134.
- SCHIFTNER, A., AND BALZER, J. 2010. Statics-sensitive layout of planar quadrilateral meshes. In *Advances in Architectural Geometry 2010*, C. Ceccato et al., Eds. Springer, Vienna, 221–236.
- SCHIFTNER, A. 2007. *Planar quad meshes from relative principal curvature lines*. Master's thesis, TU Wien.
- SORKINE, O., COHEN-OR, D., AND TOLEDO, S. 2003. High-pass quantization for mesh encoding. In *Symposium Geometry processing*. 42–51.
- VAN MELE, T., AND BLOCK, P. 2011. A novel form finding method for fabric formwork for concrete shells. *J. Int. Assoc. Shell and Spatial Structures* 52, 217–224.
- WARDETZKY, M., MATHUR, S., KÄLBERER, F., AND GRINSPUN, E. 2007. Discrete Laplace operators: No free lunch. In *Symposium on Geometry Processing*. 33–37.
- WHITING, E., OCHSENDORF, J., AND DURAND, F. 2009. Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph.* 28, 5, #112,1–9.

Figure 15: Test sequence: Stability w.r.t. removal of single pieces of masonry. (a) self-supporting surface and thrust network. (b) Part of the surface has been removed, the existence of a modified thrust network shows it is self-supporting. (c) If we remove too much, our algorithm no longer finds an admissible thrust network. It is plausible that the surface is no longer self-supporting.

Figure 16: Test sequence: Stability w.r.t. increasing loads in a single point. **IMAGES SIMILAR TO PREVIOUS**

Figure 17: Structural Glass. Glass as a structural element can support stresses up to **FILL IN VALUES**. (a) Self-supporting thrust network such that forces do not exceed F_{\max} . (b) Remeshing of this surface by a planar quad mesh (not necessarily self-supporting). (c) A steel/glass construction following this quad mesh, which utilizes the structural properties of glass. Since it is very close to a self-supporting shape, joints will experience low bending and torsion moments.