

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 initiated 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 in particular shows us how to perform remeshing of a self-supporting shape by a self-supporting quad mesh with planar faces.

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 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

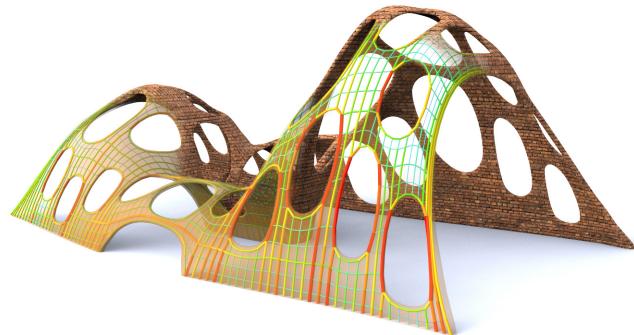


Figure 1: A surface with irregularly placed holes almost never stands by itself when built from bricks; for those that do, stability is not obvious by inspection. The surface shown is produced by finding the nearest self-supporting shape from a given freeform geometry. The image also illustrates the fictitious thrust network used in our algorithm, with edges' cross-section and coloring visualizing the magnitude of forces.

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, 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 interpretation of the equilibrium conditions in terms of discrete curvatures.
- The graph Laplacian derived from a thrust network with compressive forces is a “perfect” one (§2.2). We show how it appears in the analysis and establish a connection with mean curvatures which are otherwise defined for polyhedral surfaces.
- We present an optimization algorithm for efficiently finding a thrust network near a given arbitrary reference surface (§3), and build a tool for interactive design of self-supporting surfaces based on this algorithm (§4).

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

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

Related Work. Unsupported masonry has been an active topic of research in the engineering community. The foundations for the modern approach were laid by Jacques Heyman [1966] and are available as the textbook [Heyman 1995]. The theory of reciprocal force diagrams in the planar case was studied already by Maxwell [Maxwell 1864]; a unifying view on polyhedral surfaces, compressive forces and corresponding “convex” force diagrams is presented by [Ash et al. 1988]. F. Fraternali [2002], [2010] established a connection between the continuous theory of stresses in membranes and the discrete theory of forces in thrust networks, by interpreting the latter as a certain non-conforming finite element discretization of the former.

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

Other approaches to the interactive design of self-supporting structures include modeling these structures as damped particle-spring systems [Kilian and Ochsendorf 2005; Barnes 2009], and mirroring the rich tradition in architecture of designing self-supporting surfaces using hanging chain models [Heyman 1998]. Alternatively, masonry structures can be represented by networks of rigid blocks [Whiting et al. 2009], whose conditions on the structural feasibility were incorporated into procedural modeling of buildings.

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

2 Self-supporting Surfaces

2.1 The Continuous Theory

We are here modeling masonry as a surface given by a height field $s(x, y)$ defined in some planar domain Ω . We assume that there are vertical loads $F(x, y)$ — usually F represents the structure’s own weight. By definition this surface is self-supporting, if and only if

there exists a field of compressive stresses which are in equilibrium with the acting forces. This is equivalent to existence of a field $M(x, y)$ of 2×2 symmetric positive semidefinite matrices satisfying

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

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

The condition $\operatorname{div} M = 0$ says that M is essentially the Hessian of a real-valued function ϕ (the *Airy stress potential*): With the notation

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

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

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

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

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

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

2.2 Discrete Theory: Thrust Networks

We are discretizing a self-supporting surface by a mesh $\mathcal{S} = (V, E, F)$ (see Figure 2). Loads are again vertical, and we discretize them as force densities F_i associated with vertices \mathbf{v}_i . The load acting on this vertex is then given by $F_i A_i$, where A_i is an area of influence (using a prime to indicate projection onto the xy plane, A_i is the area of the Voronoi cell of \mathbf{v}'_i w.r.t. V'). We assume that stresses are carried by the edges of the mesh: the force exerted on the vertex \mathbf{v}_i by the edge connecting $\mathbf{v}_i, \mathbf{v}_j$ is given by

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

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

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

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

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

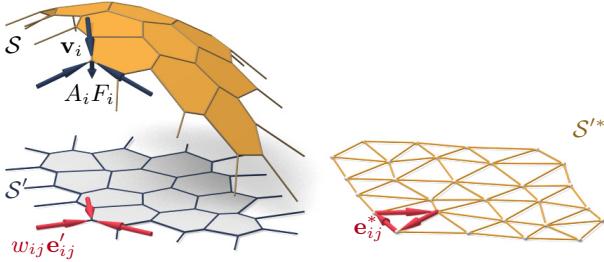


Figure 2: A thrust network \mathcal{S} , with dangling edges indicating external forces (left). This network together with compressive forces which balance vertical loads $A_i F_i$ projects onto a planar mesh \mathcal{S}' with equilibrium compressive forces $w_{ij} \mathbf{e}'_{ij}$ in its edges. Rotating forces by 90° leads to the reciprocal force diagram \mathcal{S}'^* (right).

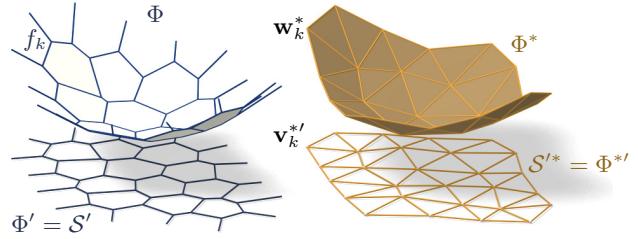


Figure 3: Airy stress potential Φ and its polar dual Φ^* . Φ projects onto the same planar mesh as \mathcal{S} does, while Φ^* projects onto the reciprocal force diagram. A primal face f_k lies in the plane $z = \alpha x + \beta y + \gamma \iff$ the corresponding dual vertex is $\mathbf{w}_k^* = (\alpha, \beta, -\gamma)$.

Reciprocal Diagram. Equations (2) have a geometric interpretation: With edge vectors

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

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

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

form a closed cycle (see Figure 2). If the mesh \mathcal{S} is simply connected, there exists an entire reciprocal diagram \mathcal{S}'^* which is a combinatorial dual of \mathcal{S} , and which has edge vectors \mathbf{e}'^*_{ij} . Its vertices are denoted by \mathbf{v}'^*_i .

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

Polyhedral Stress Potential. We can go further and construct a convex polyhedral “Airy stress potential” surface Φ with vertices $\mathbf{w}_i = (x_i, y_i, \phi_i)$ combinatorially equivalent to \mathcal{S} by requiring that a primal face of Φ lies in the plane $z = \alpha x + \beta y + \gamma$ if and only if (α, β) is the corresponding dual vertex of \mathcal{S}'^* (see Figure 3). Obviously this condition determines Φ up to vertical translation. For existence see [Ash et al. 1988]. The inverse procedure constructs a reciprocal diagram from Φ . This procedure works also if forces are not compressive: we can construct an Airy mesh Φ which has planar faces, but it will no longer be a convex polyhedron.

The vertices of Φ can be interpolated by a piecewise-linear function $\phi(x, y)$. It is easy to see that the derivative of $\phi(x, y)$ jumps by the amount $\|\mathbf{e}'^*_{ij}\| = w_{ij} \|\mathbf{e}'_{ij}\|$, when crossing over the edge \mathbf{e}'_{ij} at right angle, with unit speed. This identifies Φ as the Airy polyhedron introduced by [Fraternali et al. 2002] as a finite element discretization of the continuous Airy function (see also [Fraternali 2010]).

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

Polarity. Polarity with respect to the Maxwell paraboloid $z = \frac{1}{2}(x^2 + y^2)$ maps the plane $z = \alpha x + \beta y + \gamma$ to the point $(\alpha, \beta, -\gamma)$. Thus, applying polarity to Φ and projecting the result Φ^* into the xy plane reconstructs the reciprocal diagram $\Phi'^* = \mathcal{S}'^*$ (see Fig. 3).

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

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

This operator is a perfect discrete Laplacian in the sense of [Wardetzky et al. 2007], since it is symmetric by construction, Equation (2) implies linear precision for the planar “top view mesh” \mathcal{S}' (i.e., $\Delta_\phi f = 0$ if f is a linear function), and $w_{ij} \geq 0$ ensures semidefiniteness and a maximum principle for Δ_ϕ -harmonic functions. Equation (3) can be written as $\Delta_\phi s = AF$.

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

2.3 Surfaces in Isotropic Geometry

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

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

Remark: The Maxwell paraboloid is considered the unit sphere of isotropic geometry, and the geometric quantities considered above are assigned specific meanings: The forces $\|\mathbf{e}'^*_{ij}\| = w_{ij} \|\mathbf{e}'_{ij}\|$ are dihedral angles of the Airy polyhedron Φ , and also “lengths” of edges of Φ^* . We do not use this terminology in the sequel.

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

$$\nabla \phi \overset{\sigma}{\longmapsto} \nabla s.$$

By moving along a curve $\mathbf{u}(t) = (x(t), y(t))$ in the parameter domain we get the first variation of tangent planes: $\frac{d}{dt} \nabla \phi|_{\mathbf{u}(t)} = (\nabla^2 \phi) \dot{\mathbf{u}}$. This yields the derivative $(\nabla^2 \phi) \dot{\mathbf{u}} \overset{d\sigma}{\longmapsto} (\nabla^2 s) \dot{\mathbf{u}}$, for all

261 $\dot{\mathbf{u}}$, and the matrix of $d\sigma$ is found as $(\nabla^2\phi)^{-1}(\nabla^2s)$. By definition,
 262 curvatures of the surface s relative to ϕ are found as

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

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

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

267 together with the formulas above implies

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

268 **Relation to Self-supporting Surfaces.** Summarizing the for-
 269 mulas above, we rewrite the balance condition (1) as

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

270 Let us draw some conclusions:

- 271 Since $H_\phi^{\text{rel}} = 1$ we see that the load $F_\phi = 2K_\phi$ is admissible
 272 for the stress surface $\phi(x, y)$, which is hereby shown as self-
 273 supporting. The quotient of loads yields $H_s^{\text{rel}} = F/F_\phi$.
- 274 If the stress surface coincides with the Maxwell paraboloid,
 275 then *constant loads characterize constant mean curvature
 276 surfaces*, because we get $K_\phi = 1$ and $H_s = F/2$.
- 277 If s_1, s_2 have the same stress potential ϕ , then $H_{s_1-s_2}^{\text{rel}} =$
 278 $H_{s_1}^{\text{rel}} - H_{s_2}^{\text{rel}} = 0$, so $s_1 - s_2$ is a (relative) minimal surface.

2.4 Meshes in Isotropic Geometry

280 A general theory of curvatures of polyhedral surfaces with respect
 281 to a polyhedral unit sphere was proposed by [Pottmann et al. 2007;
 282 Bobenko et al. 2010], and its dual complement in isotropic geom-
 283 etry was elaborated by [Pottmann and Liu 2007]. As illustrated by
 284 Figure 4, the mean curvature of a self-supporting surface \mathcal{S} relative
 285 to its discrete Airy stress potential is associated with the vertices of
 286 \mathcal{S} . It is computed from areas and mixed areas of faces in the polar
 287 polyhedra \mathcal{S}^* and Φ^* :

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

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

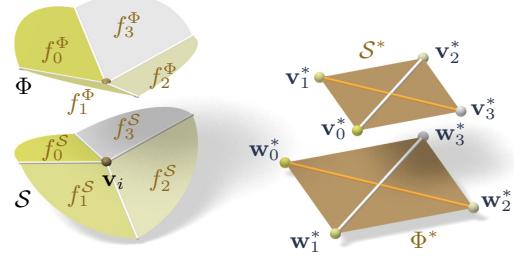
288 The prime denotes the projection into the xy plane, and summation
 289 is over those dual vertices which are adjacent to \mathbf{v}_i . Replacing \mathbf{v}'_k
 290 by \mathbf{w}'_k yields $A_i(\Phi, \Phi) = \frac{1}{2} \sum \det(\mathbf{w}'_k, \mathbf{w}'_{k+1})$.

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

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

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

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



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

269 Without loss of generality $\mathbf{v}_i = 0$, so the vertex \mathbf{v}'_{j+1}^* by construction
 270 equals the gradient of the linear function $\mathbf{x} \mapsto \langle \mathbf{v}'_{j+1}^*, \mathbf{x} \rangle$ defined by
 271 the properties $\mathbf{e}'_{j-1} \mapsto s_{j-1} - s_i$, $\mathbf{e}'_j \mapsto s_j - s_i$. Corresponding
 272 edge vectors $\mathbf{v}'_{j+1} - \mathbf{v}'_j$ and $\mathbf{w}'_{j+1} - \mathbf{w}'_j$ are parallel, because
 273 $\langle \mathbf{v}'_{j+1} - \mathbf{v}'_j, \mathbf{e}'_j \rangle = (s_j - s_i) - (s_j - s_i) = 0$. Expand $2A_i(\mathcal{S}, \Phi)$:

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

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

275 In order to discretize (4), we also need a discrete Gaussian curva-
 276 ture, which is usually defined as a quotient of areas which corre-
 277 spond under the Gauss mapping. We define

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

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

280 **Remark:** If the faces of the thrust network \mathcal{S} are not planar, the sim-
 281 ple trick of introducing additional edges with zero forces in them
 282 makes them planar, and the theory is applicable. We refrain from
 283 elaborating this further.

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

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

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

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

294 **Proof.** The relation between equilibrium forces $w_{ij} \mathbf{e}_{ij}$ in \mathcal{S} and
 295 the polyhedral stress potential Φ has been discussed above, and
 296 so has the equivalence “ $w_{ij} \geq 0 \iff \Phi$ convex” (see e.g.

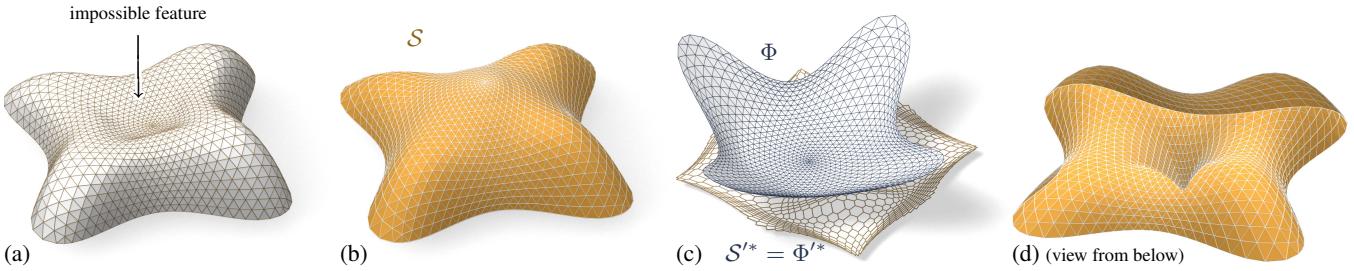


Figure 5: The top of the Lilium Tower (a) cannot stand as a masonry structure, because its central part is concave. Our algorithm finds a nearby self-supporting mesh (b) without this impossible feature. (c) shows the corresponding Airy mesh Φ and reciprocal force diagram S'^* . (d) The user can edit 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.

[Ash et al. 1988] for a survey of this and related results). It remains to show that Equations (2) and (6) are equivalent. This is the case because the proposition above implies $2K(\mathbf{v}_i)H^{\text{rel}}(\mathbf{v}_i) = 2\frac{A_i(\Phi, \Phi)}{A_i} \frac{A_i(\Phi, S)}{A_i(\Phi, \Phi)} = \frac{1}{A_i} (\sum_{j \sim i} w_{ij}(s_j - s_i)) = \frac{1}{A_i} A_i F_i$. \square

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

- Does S' have a reciprocal diagram whose corresponding Airy polyhedron Φ approximates the continuous Airy potential ϕ ? (if the surfaces involved are not simply connected, these objects are defined locally).
- Does S' possess a “perfect” discrete Laplace-Beltrami operator Δ_ϕ in the sense of Wardetzky et al. [2007] whose weights are the edge length scalars of such a reciprocal diagram?

From [Wardetzky et al. 2007] we know that perfect Laplacians exist only on regular triangulations which are projections of convex polyhedra. On the other hand, previous sections show how to appropriately re-triangulate: Let Φ be a triangle mesh convex hull of the vertices $(x_i, y_i, \phi(x_i, y_i))$, where (x_i, y_i) are vertices of S' . Then its polar dual Φ^* projects onto a reciprocal diagram with positive edge weights, so Δ_ϕ has positive weights, and the vertices (x_i, y_i, s_i) of S can be found by solving the discrete Poisson problem $(\Delta_\phi s)_i = A_i F_i$.

Assuming the discrete Δ_ϕ approximates its continuous counterpart, this yields a mesh approximating $s(x, y)$, and we conclude: A smooth self-supporting surface can be approximated by a discrete self-supporting triangular mesh for any sampling of the surface.

3 Thrust Networks from Reference Meshes

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

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

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

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

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

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

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

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

Step 3: Alter Positions. In the previous step we fit as best as possible a stress surface Φ to 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 S , respectively. It is possible to simply solve for new vertex positions

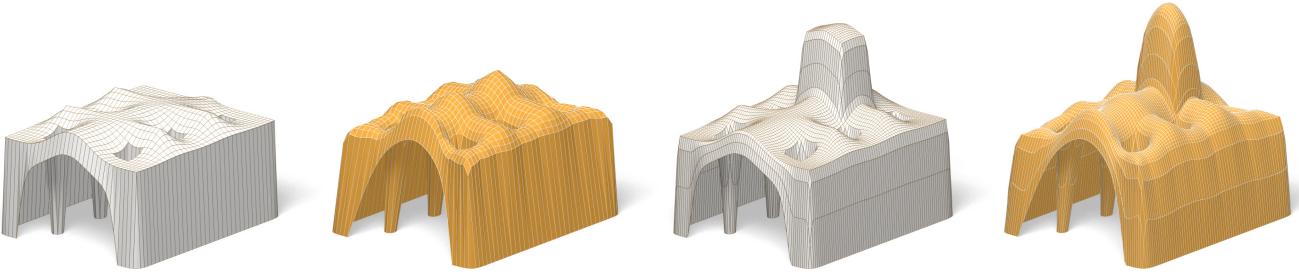


Figure 6: The user-designed reference mesh (left) is not self-supporting, but our algorithm finds a nearby perturbation of the reference surface (middle-left) that is in equilibrium. As the user makes edits to the reference surface (middle-right), the thrust network automatically adjusts (right).

Example	Figure	Vertices	Edges	Time (s)	Iterations	Max Rel Error
Top of Lilium Tower	Fig. 5b	1201	3504	21.6	9	4.2×10^{-5}
Top of Lilium Tower (with pillar)	Fig. 5d	1200	3500	26.5	10	8.5×10^{-5}
Freeform Structure with Two Pillars	Fig. 7	1535	2976	17.0	21	2.7×10^{-5}
Swiss Cheese	Fig. 9	2358	4302	19.5	9	3.0×10^{-4}
Brick Domes	Fig. 8	752	2165	8.0	9	5.8×10^{-5}
Structural Glass	Fig. 15	527	998	5.7	25	2.4×10^{-5}

Table 1: Numerical details about the examples throughout this paper. Time: The wall-clock time needed by an Intel Xeon 2.3GHz desktop PC with 4 GB of RAM to find a self-supporting thrust network and associated stress surface from the example’s reference mesh; we also give the number of outer iterations of the four steps in (§3). The maximum relative error is the dimensionless relative error in force equilibrium defined by $\max_i \|A_i F_i - \sum_{j \sim i} w_{ij}(\mathbf{v}_j - \mathbf{v}_i)\| / \|A_i F_i\|$, where the maximum is taken runs over interior vertices \mathbf{v}_i .

407 that put \mathcal{S} in static equilibrium, since Equations (2) and (3) with
408 w_{ij} fixed form a square linear system that is typically nonsingular.

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

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

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

425 **Implementation Details.** Solving the weighted least-squares
426 problem of Step 3 amounts to solving a sparse, symmetric linear
427 system. While the MINRES algorithm [Paige and Saunders 1975]
428 is likely the most robust algorithm for solving this system, in prac-
429 tice we have observed that the method of conjugate gradients works
430 well despite the potential ill-conditioning of the objective matrix.

431 **Limitations.** This algorithm is not guaranteed to always con-
432 verge; this fact is not surprising from the physics of the problem
433 (if the boundary of the reference mesh encloses too large of a re-
434 gion, w_{\max} is set too low, and the density of the surface too high,

435 a thrust network in equilibrium simply does not exist – the vault is
436 too ambitious and cannot be built to stand; pillars are needed.)

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

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

4 Results

457 **Interactive Design of Self-Supporting Surfaces.** The opti-
458 mization algorithm described in the previous section forms the ba-
459 sis of an interactive design tool for self-supporting surfaces. Users
460 manipulate a mesh representing a reference surface, and the com-
461 puter searches for a nearby thrust network in equilibrium (see e.g.
462 Figure 6). Fitting this thrust network does not require that the user
463 specify boundary tractions, and although the top view of the refer-
464 ence mesh is used as an initial guess for the top view of the thrust
465 network, the search is not restricted to this top view. The features

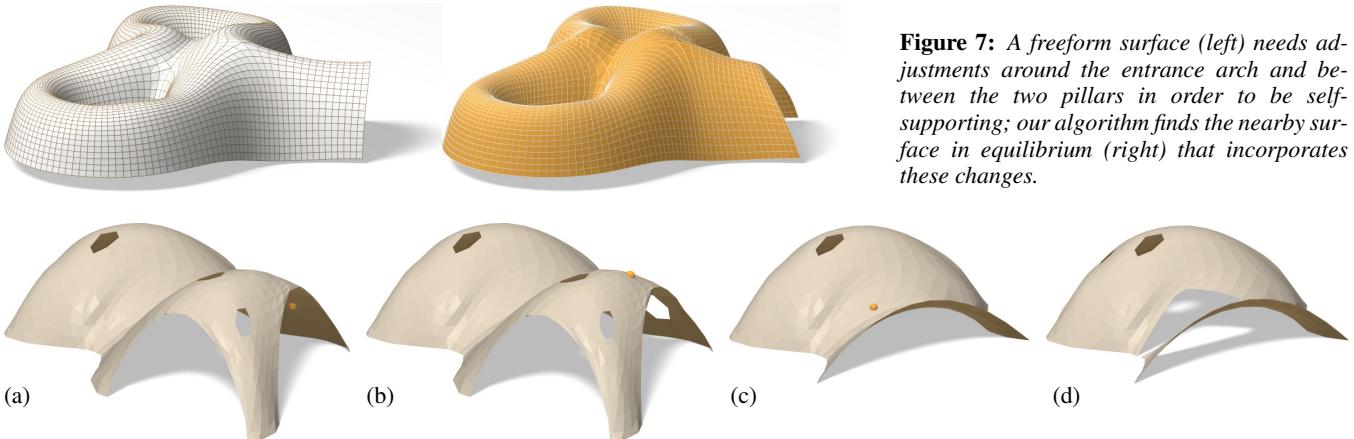


Figure 8: Destruction sequence. We simulate removing small parts of masonry (their location is shown by a yellow ball) and the falling off of further pieces which are no longer supported after removal. For this example, removing a certain small number of single bricks does not affect stability (a,b). Removal of material at a certain point (yellow) will cause a greater part of the structure to collapse, as seen in (c). (d) shows the result after one more removal (all images show the respective thrust networks, not the reference surface).

466 of the design tool include:

- 467 Handle-based 3D editing of the reference mesh using Lapla-
468 cian coordinates [Lipman et al. 2004; Sorkine et al. 2003] to
469 extrude vaults, insert pillars, and apply other deformations to
470 the reference mesh. Handle-based adjustments of the heights,
471 keeping the top view fixed, and deformation of the top view,
472 keeping the heights fixed, are also supported. The thrust net-
473 work adjusts interactively to fit the deformed positions, giving
474 the usual visual feedback about the effects of edits on whether
475 or not the surface can stand.
- 476 • Specification of boundary conditions. Points of contact be-
477 tween the reference surface and the ground or environment
478 are specified by “pinning” vertices of the surface, specifying
479 that the thrust network must coincide with the reference mesh
480 at this point, and relaxing the condition that forces must be in
481 equilibrium there.
- 482 • Interactive adjustment of surface density ρ , external loads,
483 and maximum permissible stress per edge w_{\max} , with visual
484 feedback of how these parameters affect the fitted thrust net-
485 work.
- 486 • Upsampling of the thrust network through Catmull-Clark sub-
487 division and polishing of the resulting refined thrust network
488 using optimization (§3).
- 489 • Visualization of the stress surface dual to the thrust network
490 and corresponding reciprocal diagram.

491 **Example: Vault with Pillars.** As an example of the design and
492 optimization workflow, consider a rectangular vault with six pillars,

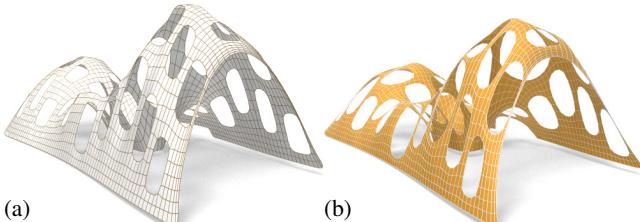


Figure 9: A mesh with holes (a) requires large deformations to both the top view and heights to render it self-supporting (b)

Figure 7: A freeform surface (left) needs ad-
justments around the entrance arch and be-
tween the two pillars in order to be self-
supporting; our algorithm finds the nearby sur-
face in equilibrium (right) that incorporates
these changes.

493 free boundary conditions along one edge, fixed boundary conditions
494 along the others, and a tower extruded from the top of the surface
495 (see Figure 6). This surface is neither convex nor simply connected,
496 and exhibits a mix of boundary conditions, none of which cause
497 our algorithm any difficulty; it finds a self-supporting thrust net-
498 work near the designed reference mesh. The user is now free to
499 make edits to the reference mesh, and the thrust network adapts to
500 these edits, providing the user feedback on whether these designs
501 are physically realizable.

Example: Top of the Lilium Tower. Consider the top portion of
502 the steel-glass exterior surface of the Lilium Tower, which is cur-
503 rently being built in Warszaw (see Figure 5). This surface contains
504 a concave part with local minimum in its interior and so cannot pos-
505 sibly be self-supporting. Given this surface as a reference mesh, our
506 algorithm constructs a nearby thrust network in equilibrium with-
507 out the impossible feature. The user can then explore how editing
508 the reference mesh – adding a pillar, for example – affects the thrust
509 network and its deviation from the reference surface.

Example: Freeform Structure with Two Pillars. Suppose an
511 architect’s experience and intuition has permitted the design of a
512 nearly self-supporting freeform surface (Figure 7). Our algorithm
513 reveals those edits needed to make the structure sound – principally
514 around the entrance arch, and the area between the two pillars.

Example: Destruction Sequence. In Figure 8 we simulate re-
516 moving parts of masonry and the falling off of further pieces which
517 are no longer supported after removal. This is done by deleting the
518 1-neighborhood of a vertex and solving for a new thrust network in
519 compressive equilibrium close to the original reference surface. We
520 delete those parts of the network which deviate too much and are
521 no longer contained in the masonry hull, and iterate.

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

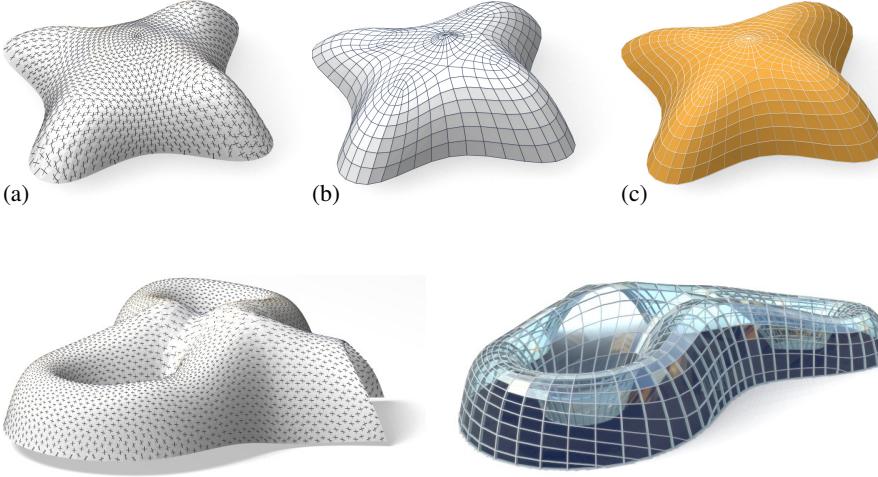


Figure 11: Planar quad remeshing of the surface of Figure 7. Left: Principal directions. Center: The result of optimization is a self-supporting PQ mesh, which guides a moment-free steel/glass construction. Right: Interior view.

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.

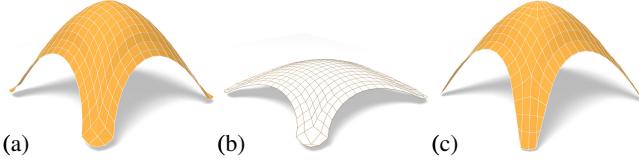


Figure 12: Directly enforcing planarity of the faces of even a very simple self-supporting quad-mesh vault (a) results in a surface far removed from the original design (b). Starting instead from a remeshing of the surface with edges following relative principal curvature directions yields a self-supporting, PQ mesh far more faithful to the original (c).

Taking an arbitrary non-planar quad mesh and attempting naive, simultaneous enforcement of planarity and static equilibrium does not yield good results, as shown in Figure 12. To identify the necessary conditions, we first discuss a planar quad mesh \mathcal{S} with vertices $\mathbf{v}_{ij} = (x_{ij}, y_{ij}, s_{ij})$ which approximates a given continuous surface $s(x, y)$. 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

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

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

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 $\phi(x, y)$.

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

Note that the relative principal curvature directions give the *unique* curve network along which a planar quad discretization of a self-supporting surface is possible. Other networks lead to results like the one shown by Figure 12. Figures 10 and 11 further illustrate the result of applying this procedure to self-supporting surfaces.

Remark: When remeshing a given shape by planar quad meshes, we know that the circular and conical properties require that the mesh follows the ordinary, Euclidean principal curvature directions [Liu et al. 2006]. It is remarkable that the self-supporting property in a similar manner requires us to follow certain *relative* principal directions. Practitioners' observations regarding the beneficial statics properties of principal directions can be explained by this analogy, because the relative principal directions are close to the Euclidean ones, if the stress distribution is uniform and $\|\nabla s\|$ is small.

Koenigs Meshes. Given a self-supporting thrust network \mathcal{S} with stress surface Φ , we ask the question: Which vertical perturbation $\mathcal{S} + \mathcal{R}$ is self-supporting, with the same loads as \mathcal{S} ? As to notation, all involved meshes $\mathcal{S}, \mathcal{R}, \Phi$ have the same top view, and arithmetic operations refer to the respective z coordinates s_i, r_i, ϕ_i of vertices.

The condition of equal loads then is expressed as $\Delta_\phi(s + r) = \Delta_\phi s$ in terms of Laplacians or as $H_{\mathcal{S} + \mathcal{R}}^{\text{rel}} = H_{\mathcal{S}}^{\text{rel}}$ in terms of mean



Figure 10: Planar quad remeshing of the “Lilium tower” surface of Figure 5. (a) Principal directions which are found as eigenvectors of $(\nabla^2 \phi)^{-1} \nabla^2 s$. (b) Quad mesh guided by principal directions is almost planar and almost self-supporting. (c) Small changes achieve both properties.

588 curvature, and is equivalent to

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

589 So \mathcal{R} is a *minimal surface* relative to Φ . While in the triangle mesh
590 case there are enough degrees of freedom for nontrivial solutions,
591 the case of planar quad meshes is more intricate: Polar polyhedra
592 \mathcal{R}^* , Φ^* have to be Christoffel duals of each other [Pottmann and
593 Liu 2007], as illustrated by Figure 4. Unfortunately not all quad
594 meshes have such a dual; the condition is that the mesh is *Koenigs*,
595 i.e., the derived mesh formed by the intersection points of diagonals
596 of faces again has planar faces [Bobenko and Suris 2008].



Figure 13: 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.

597 **Koebe meshes.** An interesting special case occurs if Φ is a
598 *Koebe mesh* of isotropic geometry, i.e., a PQ mesh whose edges
599 touch the Maxwell paraboloid. Since Φ approximates the Maxwell
600 paraboloid, we get $2K(\mathbf{v}_i)H^{\text{rel}}(\mathbf{v}_i) \approx 1$ and Φ consequently is
601 self-supporting for unit load. Applying the Christoffel dual con-
602 struction described above yields a minimal mesh \mathcal{R} and a family of
603 meshes $\Phi + \alpha\mathcal{R}$ which are self-supporting for unit load (see Fig-
604 ure 13).

6 Conclusion and Future Work

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

620 **Future Work.** There are several directions of future research. One
621 is to incorporate non-manifold meshes, which occur naturally when
622 e.g. supporting walls are introduced. It is also obvious that non-ver-
623 tical loads, e.g. wind load, play a role. There are also some direc-
624 tions to pursue in improving the algorithms, for instance adaptive
625 remeshing in problem areas. Probably the interesting connections
626 between statics properties and geometry are not yet exhausted, and
627 we would like to propose the *geometrization* of problems as a strat-
628 egy for their solution.

629 **Acknowledgements.** This work was very much inspired by
630 Philippe Block’s plenary lecture at the 2011 Symposium Geom-
631 etry Processing in Lausanne. Several illustrations (the maximum

632 load example of Figure 14 and the destruction sequence of Figure
633 8) have real-world analogues on his web page [Block 2011].

References

- 634 ANDREU, A., GIL, L., AND ROCA, P. 2007. Computational anal-
635 ysis of masonry structures with a funicular model. *J. Engrg.
636 Mechanics* 133, 473–480.
- 638 ASH, P., BOLKER, E., CRAPO, H., AND WHITELEY, W. 1988.
639 Convex polyhedra, Dirichlet tessellations, and spider webs. In
640 *Shaping space (Northampton 1984)*. Birkhäuser, 231–250.
- 641 BARNES, M. R. 2009. Form finding and analysis of tension struc-
642 tures by dynamic relaxation. *Int. J. Space Structures* 14, 2, 89–
643 104.
- 644 BLOCK, P., AND LACHAUER, L. 2011. Closest-fit, compression-
645 only solutions for free form shells. In *IABSE — IASS 2011 Lon-
646 don Symposium*, Int. Ass. Shell Spatial Structures. electronic.
- 647 BLOCK, P., AND OCHSENDORF, J. 2007. Thrust network analysis:
648 A new methodology for three-dimensional equilibrium. *J. Int.
649 Assoc. Shell and Spatial Structures* 48, 3, 167–173.
- 650 BLOCK, P. 2009. *Thrust Network Analysis: Exploring Three-
651 dimensional Equilibrium*. PhD thesis, Massachusetts Institute
652 of Technology.
- 653 BLOCK, P., 2011. Project webpage at [http://block.arch.ethz.ch/
654 projects/freeform-catalan-thin-tile-vaulting](http://block.arch.ethz.ch/projects/freeform-catalan-thin-tile-vaulting).
- 655 BOBENKO, A., AND SURIS, YU. 2008. *Discrete differential geom-
656 etry: Integrable Structure*. No. 98 in Graduate Studies in Math.
657 American Math. Soc.
- 658 BOBENKO, A., POTTMANN, H., AND WALLNER, J. 2010. A
659 curvature theory for discrete surfaces based on mesh parallelity.
660 *Math. Annalen* 348, 1–24.
- 661 COHEN-STEINER, D., AND MORVAN, J.-M. 2003. Restricted
662 Delaunay triangulations and normal cycle. In *Proc. 19th Symp.
663 Computational geometry*, ACM, 312–321.
- 664 FRATERNALI, F., ANGELILLO, M., AND FORTUNATO, A. 2002.
665 A lumped stress method for plane elastic problems and the
666 discrete-continuum approximation. *Int. J. Solids Struct.* 39,
667 6211–6240.
- 668 FRATERNALI, F. 2010. A thrust network approach to the equi-
669 librium problem of unreinforced masonry vaults via polyhedral
670 stress functions. *Mechanics Res. Comm.* 37, 2, 198 – 204.
- 671 FRIEDLANDER, M. P., 2007. BCLS: Bound constrained least
672 squares. <http://www.cs.ubc.ca/~mpf/bcls>.
- 673 GIAQUINTA, M., AND GIUSTI, E. 1985. Researches on the equi-
674 librium of masonry structures. *Archive for Rational Mechanics
675 and Analysis* 88, 4, 359–392.
- 676 GLYMPH, J., SHELDEN, D., CECCATO, C., MUSSLEM, J., AND
677 SCHOBER, H. 2004. A parametric strategy for free-form glass
678 structures using quadrilateral planar facets. *Automation in Con-
679 struction* 13, 2, 187 – 202.
- 680 HEYMAN, J. 1966. The stone skeleton. *Int. J. Solids Structures* 2,
681 249–279.
- 682 HEYMAN, J. 1995. *The Stone Skeleton: Structural Engineering of
683 Masonry Architecture*. Cambridge University Press.
- 684 HEYMAN, J. 1998. *Structural Analysis: A Historical Approach*.
685 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 J. Num. 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., vol. 4647 of *LNCS*. 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.
- 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*. Eurographics Assoc., 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.

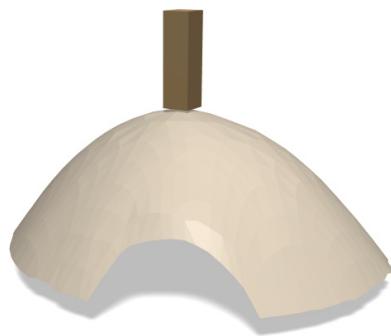


Figure 14: Testing stability. This self-supporting surface of length 24 m is imagined as masonry of thickness 0.1 m. It possesses a thrust network inside the masonry hull if – for the sake of example – a load of 11.000 kg (shown in dark) is applied to a certain vertex. This means that the surface is still stable after that load is applied (N.B. This method of testing is rather conservative).

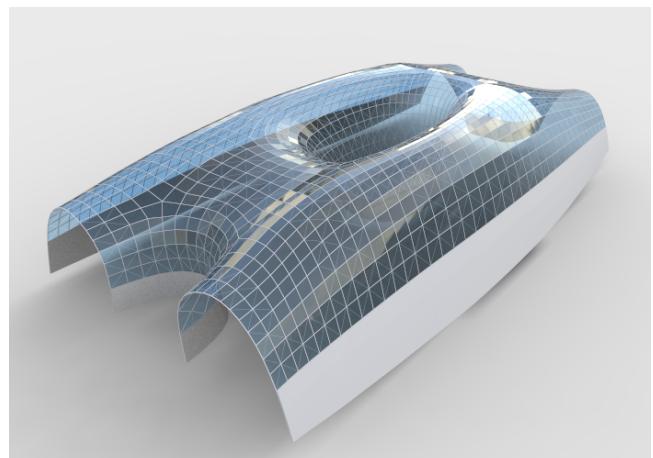


Figure 15: Glass as a structural element can support stresses up to, say, 30 MPa. We propose a steel/glass construction which utilizes the structural properties of glass by first solving for a self-supporting thrust network such that forces do not exceed the maximum values, and subsequent remeshing of this surface by a planar quad mesh (not necessarily self-supporting itself). Since this surface is very close to a self-supporting shape, joints will experience low bending and torsion moments.