

Design of Self-supporting Surfaces

Abstract

Self-supporting masonry is one of the most ancient and elegant techniques for building curved shapes. Because of the very geometric nature of their failure, analyzing and modeling such structures is more a geometry processing problem than one of classical continuum mechanics. 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. The rich geometry of thrust networks leads us to close connections between different topics of discrete differential geometry, such as a finite-element discretization of the Airy stress potential, perfect graph Laplacians, and the problem of computing admissible loads via curvatures of polyhedral surfaces. This geometric viewpoint allows us, in particular, to remesh self-supporting shapes by self-supporting quad meshes 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. For this reason they have been an object of interest since antiquity; large, non-convex examples of such structures include 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, with additional support 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 actual forces present 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 locally by the so-called Airy stress function. 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.,

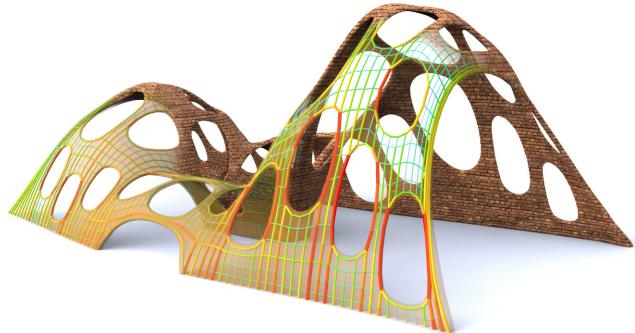


Figure 1: Surfaces with irregularly placed holes almost never stand by themselves when built from bricks; for those that do, stability is not obvious by inspection. The surface shown is produced by finding the nearest self-supporting shape from a given freeform geometry. The image also illustrates the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

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 network is a system of forces in equilibrium with the structure's deadload. 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).

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

147 $M(x, y)$ of 2×2 symmetric positive semidefinite matrices satisfying
 148

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

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

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

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

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

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

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

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

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

169 2.2 Discrete Theory: Thrust Networks

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

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

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

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

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

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

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

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

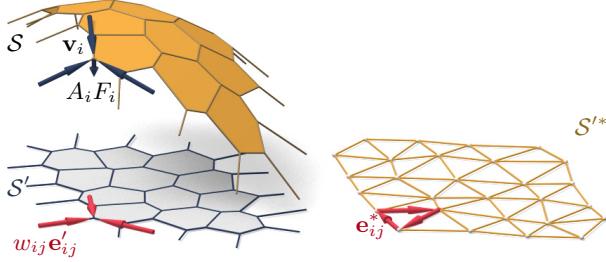


Figure 2: A thrust network \mathcal{S} with dangling edges indicating external forces (left). This network together with compressive forces which balance vertical loads $A_i F_i$ projects onto a planar mesh \mathcal{S}' with equilibrium compressive forces $w_{ij} \mathbf{e}'_{ij}$ in its edges. Rotating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

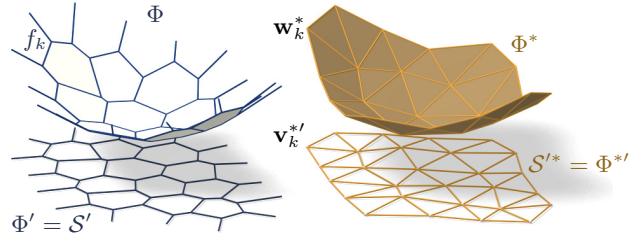


Figure 3: Airy stress potential Φ and its polar dual Φ^* . Φ projects onto the same planar mesh as \mathcal{S} does, while Φ^* projects onto the reciprocal force diagram. A primal face f_k lies in the plane $z = \alpha x + \beta y + \gamma \iff$ the corresponding dual vertex is $\mathbf{w}_k^* = (\alpha,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

187 Equation (2) asserts that vectors $w_{ij} \mathbf{e}'_{ij}$ form a closed cycle. Rotating them by 90 degrees, we see that likewise
188

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

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

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

195 **Polyhedral Stress Potential.** We can go further and construct a
196 convex polyhedral “Airy stress potential” surface Φ with vertices
197 $\mathbf{w}_i = (x_i, y_i, \phi_i)$ combinatorially equivalent to \mathcal{S} by requiring that
198 a primal face of Φ lies in the plane $z = \alpha x + \beta y + \gamma$ if and only if
199 (α, β) is the corresponding dual vertex of \mathcal{S}'^* (see Figure 3). Ob-
200 viously this condition determines Φ up to vertical translation. For
201 existence see [Ash et al. 1988]. The inverse procedure constructs
202 a reciprocal diagram from Φ . This procedure works also if forces
203 are not compressive: we can construct an Airy mesh Φ which has
204 planar faces, but it will no longer be a convex polyhedron.

205 The vertices of Φ can be interpolated by a piecewise-linear function
206 $\phi(x, y)$. It is easy to see that the derivative of $\phi(x, y)$ jumps by the
207 amount $\|\mathbf{e}'_{ij}^*\| = w_{ij} \|\mathbf{e}'_{ij}\|$ when crossing over the edge \mathbf{e}'_{ij} at right
208 angle, with unit speed. This identifies Φ as the Airy polyhedron in-
209 troduced by [Fraternali et al. 2002] as a finite element discretization
210 of the continuous Airy function (see also [Fraternali 2010]).

211 If the mesh is not simply connected, the reciprocal diagram and
212 the Airy polyhedron exist only locally. Our computations do not
213 require global existence.

214 **Polarity.** Polarity with respect to the *Maxwell paraboloid* $z =$
215 $\frac{1}{2}(x^2 + y^2)$ maps the plane $z = \alpha x + \beta y + \gamma$ to the point $(\alpha, \beta, -\gamma)$.
216 Thus, applying polarity to Φ and projecting the result Φ^* into the xy
217 plane reconstructs the reciprocal diagram $\Phi'^* = \mathcal{S}'^*$ (see Fig. 3).

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

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

220 This operator is a perfect discrete Laplacian in the sense of [War-
221 detzky et al. 2007], since it is symmetric by construction, Equa-
222 tion (2) implies linear precision for the planar “top view mesh” \mathcal{S}'
223 (i.e., $\Delta_\phi f = 0$ if f is a linear function), and $w_{ij} \geq 0$ ensures

224 semidefiniteness and a maximum principle for Δ_ϕ -harmonic func-
225 tions. Equation (3) can be written as $\Delta_\phi s = AF$.

226 Note that Δ_ϕ is well defined even when the underlying meshes are
227 not simply connected.

2.3 Surfaces in Isotropic Geometry

229 It is worthwhile to reconsider the basics of self-supporting surfaces
230 in the language of dual-isotropic geometry, which takes place in \mathbb{R}^3
231 with the z axis as a distinguished vertical direction. The basic ele-
232 ments of this geometry are planes, having equation $z = f(x, y) =$
233 $\alpha x + \beta y + \gamma$. The gradient vector $\nabla f = (\alpha, \beta)$ determines the
234 plane up to translation. A plane tangent to the graph of the function
235 $s(x, y)$ has gradient vector ∇s .

236 There is the notion of *parallel points*: $(x, y, z) \parallel (x', y', z') \iff$
237 $x = x', y = y'$.

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

243 **Curvatures.** Generally speaking, in the differential geometry of
244 surfaces one considers the *Gauss map* σ from a surface S to a con-
245 vex unit sphere Φ by requiring that corresponding points have par-
246 allel tangent planes. Subsequently mean curvature H^{rel} and Gaus-
247 sian curvature K^{rel} relative to Φ are computed from the derivative
248 $d\sigma$. Classically Φ is the ordinary unit sphere $x^2 + y^2 + z^2 = 1$, so
249 that σ maps each point to its unit normal vector.

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

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

256 By moving along a curve $\mathbf{u}(t) = (x(t), y(t))$ in the parameter
257 domain we get the first variation of tangent planes: $\frac{d}{dt} \nabla \phi|_{\mathbf{u}(t)} =$
258 $(\nabla^2 \phi) \dot{\mathbf{u}}$. This yields the derivative $(\nabla^2 \phi) \dot{\mathbf{u}} \xrightarrow{d\sigma} (\nabla^2 s) \dot{\mathbf{u}}$, for all
259 $\dot{\mathbf{u}}$, and the matrix of $d\sigma$ is found as $(\nabla^2 \phi)^{-1} (\nabla^2 s)$. By definition,

260 curvatures of the surface s relative to ϕ are found as

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

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

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

265 together with the formulas above implies

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

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

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

268 Let us draw some conclusions:

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

2.4 Meshes in Isotropic Geometry

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

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

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

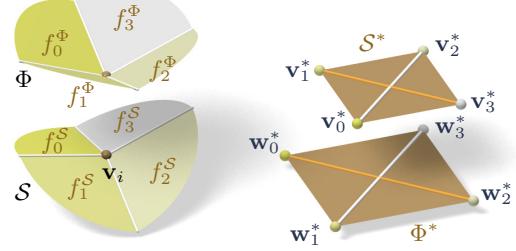
286 The prime denotes the projection into the xy plane, and summation
287 is over those dual vertices which are adjacent to \mathbf{v}_i . Replacing \mathbf{v}'_k
288 by \mathbf{w}'_k yields $A_i(\Phi, \Phi) = \frac{1}{2} \sum \det(\mathbf{w}'_k, \mathbf{w}'_{k+1})$.

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

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

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

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



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

Without loss of generality $\mathbf{v}_i = 0$, so the vertex \mathbf{v}'_{j+1}^* by construction equals the gradient of the linear function $\mathbf{x} \mapsto \langle \mathbf{v}'_{j+1}^*, \mathbf{x} \rangle$ 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 $2A_i(\mathcal{S}, \Phi)$:

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

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, 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 them planar, and the theory is applicable. In the interest of space, we refrain from elaborating further.

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

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

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

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

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

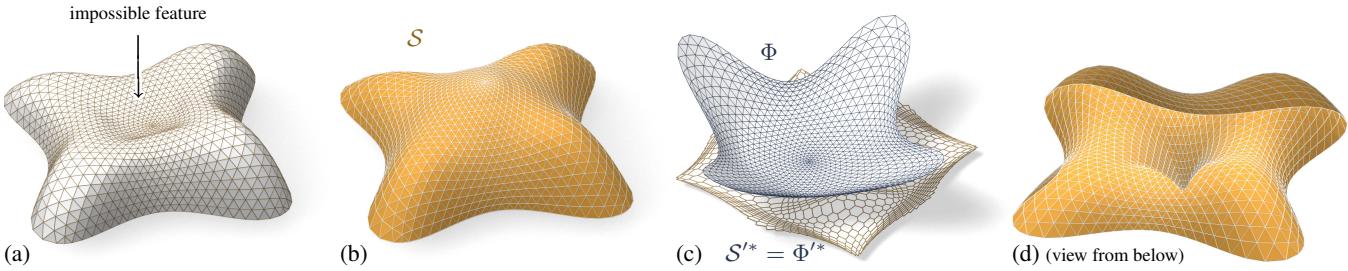


Figure 5: The top of the Lilium Tower (a) cannot stand as a masonry structure, because its central part is concave. Our algorithm finds a nearby self-supporting mesh (b) without this impossible feature. (c) shows the corresponding Airy mesh Φ and reciprocal force diagram $S'^* = \Phi'^*$. (d) The user 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 , locally fit an associated stress surface Φ .
3. Alter positions \mathbf{v}_i to improve the fit.
4. Repeat from Step 1 until convergence.

This staggered approach has several advantages: a nearby self-supporting surface is found given only a suggested reference shape, without needing to single one of the many possible top view reciprocal diagrams or needing to specify boundary tractions – these are found automatically during optimization. Although providing an initial top view graph with good combinatorics remains important, our approach allows the thrust network to slide both vertically and tangentially to the ground, essential to finding faithful thrust networks for surfaces with free boundary conditions. The surface of Fig. 1, for example, relies heavily on such sliding.

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 surface area on each of its incident faces (note that this surface area is distinct from A_i , the vertex’s Voronoi area on the top view), 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. We define 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, and 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

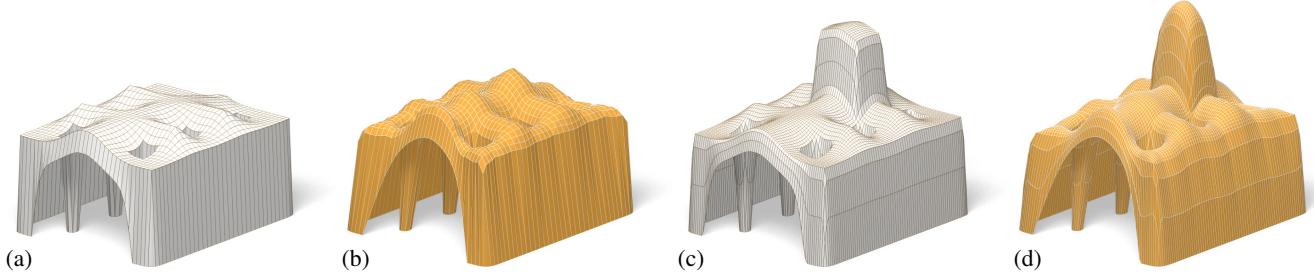


Figure 6: The user-designed reference mesh (a) is not self-supporting, but our algorithm finds a nearby perturbation of the reference surface (b) that is in equilibrium. As the user makes edits to the reference surface (c), the thrust network automatically adjusts (d).

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. We show the -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 over interior vertices \mathbf{v}_i .

407 locally integrate to a stress surface satisfying (6), and this Φ certifies
 408 that \mathcal{S} is self-supporting – we are done. Otherwise, \mathcal{S} is not self-
 409 supporting and its vertices must be moved.

437 system. While the MINRES algorithm [Paige and Saunders 1975]
 438 is likely the most robust algorithm for solving this system, in prac-
 439 tice we have observed that the method of conjugate gradients works
 440 well despite the potential ill-conditioning of the objective matrix.

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

441 **Limitations.** This algorithm is not guaranteed to always con-
 442 verge; this fact is not surprising from the physics of the problem
 443 (if the boundary of the reference mesh encloses too large of a re-
 444 gion, w_{\max} is set too low, and the density of the surface too high,
 445 a thrust network in equilibrium simply does not exist – the vault is
 446 too ambitious and cannot be built to stand; pillars are needed.)

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

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

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

447 We can, however, make a few remarks. Step 2 always decreases the
 448 equilibrium energy E of Equation (7) and Step 3 does as well as
 449 $\beta \rightarrow 0$. Moreover, as $\alpha \rightarrow 0$ and $\beta \rightarrow 0$, Step 3 approaches a lin-
 450 ear system with as many equations as unknowns; if this system has
 451 full rank, its solution sets $E = 0$. These facts suggest that the algo-
 452 rithm should generally converge to a thrust network in equilibrium,
 453 provided that Step 1 does not increase the loads by too much at ev-
 454 ery iteration, and this is indeed what we observe in practice. One
 455 case where this assumption is guaranteed to hold is if the thickness
 456 of the surface is allowed to freely vary, so that it can be chosen so
 457 that the surface has uniform density over the top view.

458 **Implementation Details.** Solving the weighted least-squares
 459 problem of Step 3 amounts to solving a sparse, symmetric linear
 460

461 If the linear system in Step 3 is singular and infeasible, the algo-
 462 rithm can stall at $E > 0$. This failure occurs, for instance, when
 463 an interior vertex has height z_i lower than all of its neighbors, and
 464 Step 2 assigns all incident edges to that vertex a weight of zero:
 465 clearly no amount of moving the vertex or its neighbors can bring
 466 the vertex into equilibrium. We avoid such degenerate configura-
 467 tions by bounding weights slightly away from zero in (8), trading
 468 increased robustness for slight smoothing of the resulting surface.
 469 Attempting to optimize meshes that have self-intersecting top views
 470 (i.e., aren’t height fields), have too many impossible features, or are
 471 insufficiently supported by fixed boundary points can also result in
 472 errors and instability.

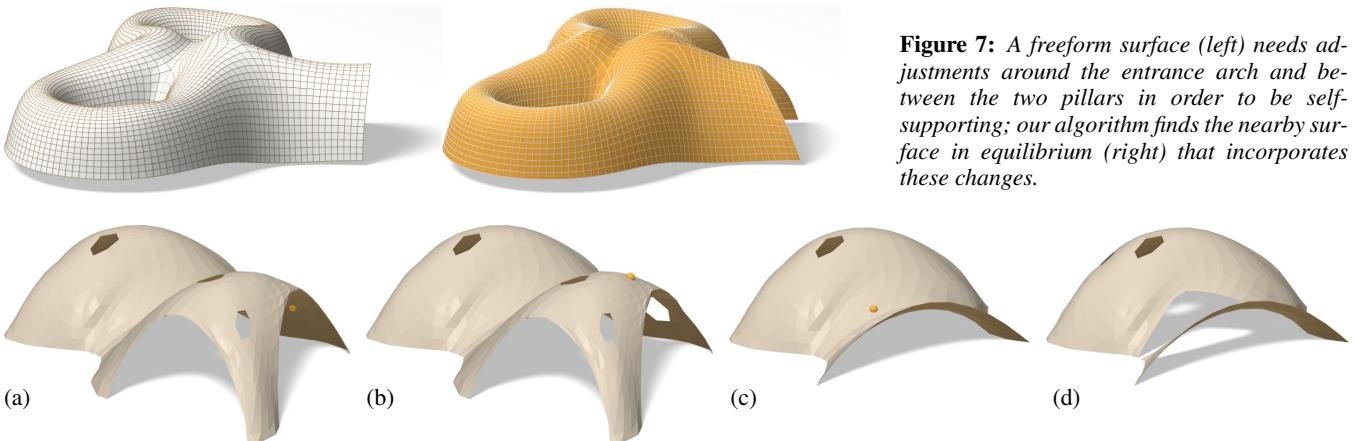


Figure 8: Destruction sequence. We simulate removing small parts of masonry (their location is shown by a yellow ball) and the falling off of further pieces which are no longer supported after removal. For this example, removing a certain small number of single bricks does not affect stability (a,b). Removal of material at a certain point (yellow ball in (b)) 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).

4 Results

Interactive Design of Self-Supporting Surfaces. The optimization algorithm described in the previous section forms the basis of an interactive design tool for self-supporting surfaces. Users manipulate a mesh representing a reference surface, and the computer searches for a nearby thrust network in equilibrium (see e.g. Figure 6). Features of the design tool include:

- 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 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 dual to the thrust network and corresponding reciprocal diagram.

Examples. *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 6). 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 5). 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 of a nearly self-supporting freeform surface (Figure 7). Our algorithm reveals those edits needed to make the structure sound – principally around the entrance arch, and the area between the two pillars.

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

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. Figure 9 shows such an implausible and

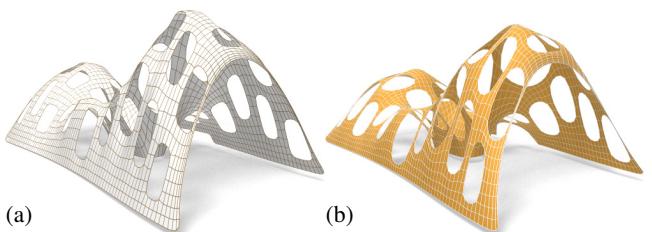


Figure 9: A mesh with holes (a) requires large deformations to both the top view and heights to render it self-supporting (b)

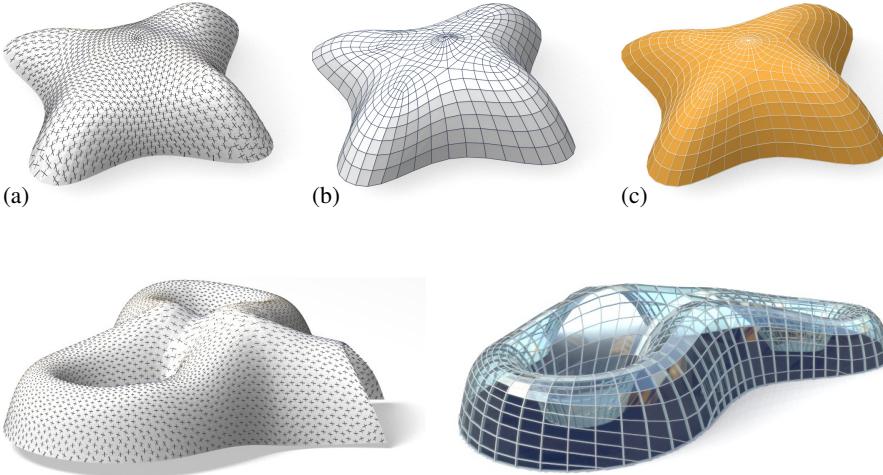


Figure 11: Planar quad remeshing of the surface of Figure 7. Left: Principal directions. Center: The result of optimization is a self-supporting PQ mesh, which guides a moment-free steel/glass construction. Right: Interior view.

unstable surface; our optimization finds a nearby, equally implausible but stable surface without difficulty (see Figures 1 and 9).

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.

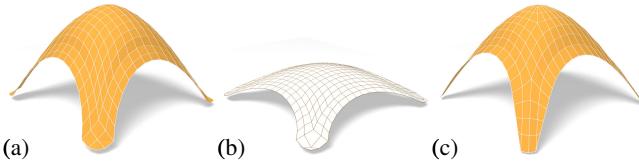


Figure 12: Directly enforcing planarity of the faces of even a very simple self-supporting quad-mesh vault (a) results in a surface far removed from the original design (b). Starting instead from a remeshing of the surface with edges following relative principal curvature directions yields a self-supporting, 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 – either by staggering a planarity optimization step every outer iteration, or adding a planarity penalty term to the position update – does not yield good results, as shown in Figure 12. Indeed, as we will see later in this section, such a planar perturbation of a thrust network is not expected to generally exist.

Consider 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} ,

538 $\mathbf{v}_{i+1,j}, \mathbf{v}_{i+1,j+1}, \mathbf{v}_{i,j+1}$, assuming the vertices lie exactly on the
539 surface $s(x, y)$. This results in

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

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

540 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

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

541 $\mathbf{a}_1, \mathbf{a}_2$ are therefore eigenvectors of $(\nabla^2 \phi)^{-1} \nabla^2 s$. In view of §2.3,
542 $\mathbf{a}_1, \mathbf{a}_2$ indicate the principal directions of the surface $s(x, y)$ relative
543 to $\phi(x, y)$.

544 In the discrete case, where s, ϕ are not given as continuous surfaces,
545 but are represented by a mesh in equilibrium and its Airy mesh, we
546 use the techniques of Schiftner [2007] and Cohen-Steiner and Mor-
547 van [2003] to approximate the Hessians $\nabla^2 s, \nabla^2 \phi$, compute prin-
548 cipal directions as eigenvectors of $(\nabla^2 \phi)^{-1} \nabla^2 s$, and subsequently
549 find meshes \mathcal{S}, Φ approximating s, ϕ which follow those directions.
550 Global optimization can now polish \mathcal{S}, Φ to a valid thrust network
551 with discrete stress potential, where before it failed: we do so by
552 taking the planarity energy $\sum_f (2\pi - \theta_f)^2$, where the sum runs
553 over faces and θ_f is the sum of the interior angles of face f , lin-
554 earizing it at every iterations, and adding it to the objective function
555 of the position update (Step 3). Convexity of Φ ensures that \mathcal{S} is
556 self-supporting.

557 Note that for each Φ , the relative principal curvature directions give
558 the *unique* curve network along which a planar quad discretization
559 of a self-supporting surface is possible. Other networks lead to re-
560 sults like the one shown by Figure 12. Figures 10 and 11 further
561 illustrate the result of applying this procedure to self-supporting
562 surfaces.

563 **Remark:** When remeshing a given shape by planar quad meshes, we
564 know that the circular and conical properties require that the mesh
565 follows the ordinary, Euclidean principal curvature directions [Liu
566 et al. 2006]. It is remarkable that the self-supporting property in a
567 similar manner requires us to follow certain *relative* principal direc-
568 tions. 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 $H_{\mathcal{S}}^{\text{rel}} = H_{\mathcal{S} + \mathcal{R}}^{\text{rel}}$ in terms of mean curvature, and is equivalent to

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

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

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

6 Conclusion and Future Work

Conclusion. This paper builds on relations between statics and geometry, some of which have been known for a long time, and connects them with newer methods of discrete differential geometry, such as discrete Laplace operators and curvatures of polyhedral surfaces. We were able to find efficient ways of modeling self-supporting freeform shapes, and provide architects and engineers with an interactive tool for evaluating the statics of freeform geometries. The self-supporting property of a shape is directly relevant for freeform masonry. The actual thrust networks we use for computation are relevant e.g. for steel constructions, where equilibrium of deadload forces implies absence of moments. This theory and accompanying algorithms thus constitute a new contribution to architectural geometry, connecting statics and geometric design.

Future Work. There are several directions of future research. One is to incorporate non-manifold meshes, which occur 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.

Acknowledgements. This work was very much inspired by Philippe Block’s plenary lecture at the 2011 Symposium on Geometry Processing in Lausanne. Several illustrations (the maximum load example of Figure 14 and the destruction sequence of Figure 8) have real-world analogues on his web page [Block 2011].

References

- ANDREU, A., GIL, L., AND ROCA, P. 2007. Computational analysis of masonry structures with a funicular model. *J. Engrg. Mechanics* 133, 473–480.
- ASH, P., BOLKER, E., CRAPO, H., AND WHITELEY, W. 1988. Convex polyhedra, Dirichlet tessellations, and spider webs. In *Shaping space (Northampton 1984)*. Birkhäuser, 231–250.
- BARNES, M. R. 2009. Form finding and analysis of tension structures by dynamic relaxation. *Int. J. 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.
- BLOCK, P., 2011. Project webpage at <http://block.arch.ethz.ch/projects/freeform-catalan-thin-tile-vaulting>.
- BOBENKO, A., AND SURIS, YU. 2008. *Discrete differential geometry: Integrable Structure*. No. 98 in Graduate Studies in Math. 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.

- 694 GLYMPH, J., SHELDEN, D., CECCATO, C., MUSSEL, J., AND
 695 SCHOBER, H. 2004. A parametric strategy for free-form glass
 696 structures using quadrilateral planar facets. *Automation in Con-*
 697 *struction* 13, 2, 187 – 202.
- 698 HEYMAN, J. 1966. The stone skeleton. *Int. J. Solids Structures* 2,
 699 249–279.
- 700 HEYMAN, J. 1995. *The Stone Skeleton: Structural Engineering of*
 701 *Masonry Architecture*. Cambridge University Press.
- 702 HEYMAN, J. 1998. *Structural Analysis: A Historical Approach*.
 703 Cambridge University Press.
- 704 KILIAN, A., AND OCHSENDORF, J. 2005. Particle-spring sys-
 705 tems for structural form finding. *J. Int. Assoc. Shell and Spatial*
 706 *Structures* 46, 77–84.
- 707 LIPMAN, Y., SORKINE, O., COHEN-OR, D., LEVIN, D., ROSSI,
 708 C., AND SEIDEL, H. 2004. Differential coordinates for interac-
 709 tive mesh editing. In *Proc. SMI. IEEE*, 181–190.
- 710 LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND
 711 WANG, W. 2006. Geometric modeling with conical meshes
 712 and developable surfaces. *ACM Trans. Graph.* 25, 3, 681–689.
- 713 LIVESLEY, R. K. 1992. A computational model for the limit anal-
 714 ysis of three-dimensional masonry structures. *Meccanica* 27,
 715 161–172.
- 716 MAXWELL, J. 1864. On reciprocal diagrams and diagrams of
 717 forces. *Philosophical Magazine* 4, 27, 250–261.
- 718 O'Dwyer, D. 1998. Funicular analysis of masonry vaults. *Com-*
 719 *puters and Structures* 73, 187–197.
- 720 PAIGE, C. C., AND SAUNDERS, M. A. 1975. Solution of sparse
 721 indefinite systems of linear equations. *SIAM J. Num. Analysis*
 722 12, 617–629.
- 723 POTTMANN, H., AND LIU, Y. 2007. Discrete surfaces in isotropic
 724 geometry. In *Mathematics of Surfaces XII*, M. Sabin and J. Win-
 725 kler, Eds., vol. 4647 of *LNCS*. Springer-Verlag, 341–363.
- 726 POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND
 727 WANG, W. 2007. Geometry of multi-layer freeform structures
 728 for architecture. *ACM Trans. Graphics* 26, 3, #65,1–11.
- 729 SCHIFTNER, A., AND BALZER, J. 2010. Statics-sensitive layout
 730 of planar quadrilateral meshes. In *Advances in Architectural Ge-
 731 ometry 2010*, C. Ceccato et al., Eds. Springer, Vienna, 221–236.
- 732 SCHIFTNER, A. 2007. *Planar quad meshes from relative principal
 733 curvature lines*. Master's thesis, TU Wien.
- 734 SORKINE, O., COHEN-OR, D., AND TOLEDO, S. 2003. High-
 735 pass quantization for mesh encoding. In *Symposium Geometry
 736 processing*. Eurographics Assoc., 42–51.
- 737 VAN MELE, T., AND BLOCK, P. 2011. A novel form finding
 738 method for fabric formwork for concrete shells. *J. Int. Assoc.*
 739 *Shell and Spatial Structures* 52, 217–224.
- 740 WARDETZKY, M., MATHUR, S., KÄLBERER, F., AND GRIN-
 741 SPUN, E. 2007. Discrete Laplace operators: No free lunch.
 742 In *Symposium on Geometry Processing*. 33–37.
- 743 WHITING, E., OCHSENDORF, J., AND DURAND, F. 2009. Pro-
 744 cedural modeling of structurally-sound masonry buildings. *ACM
 745 Trans. Graph.* 28, 5, #112,1–9.



Figure 14: This self-supporting surface of diameter 14 m is imagined as masonry of thickness 0.1 m. It still possesses a thrust network inside the masonry hull if 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.

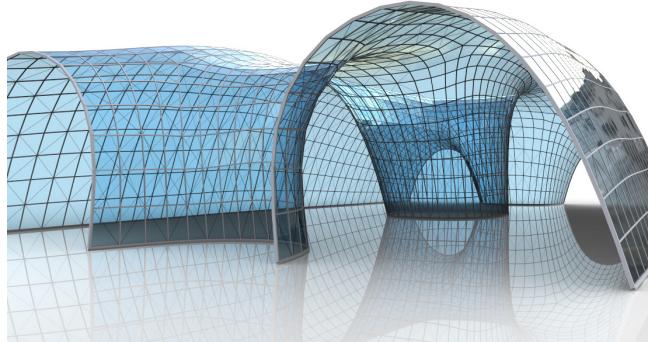
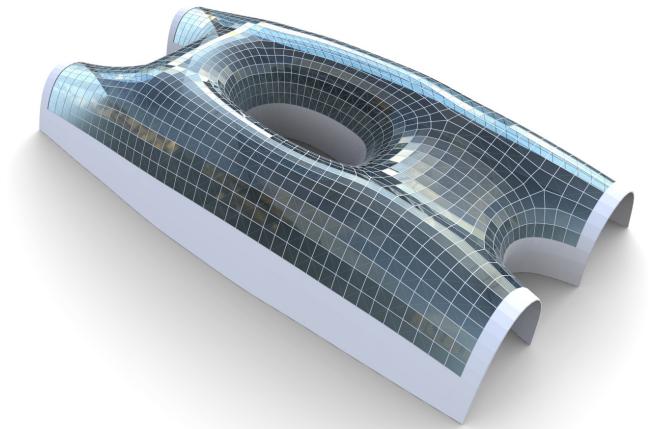


Figure 15: Glass as a structural element can support stresses up to, say, 30 MPa. We propose steel/glass constructions which utilize the structural properties of glass by first solving for a self-supporting thrust network such that forces do not exceed the maximum values, and subsequent remeshing of 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.