Design of Self-supporting Surfaces

Abstract

TODO

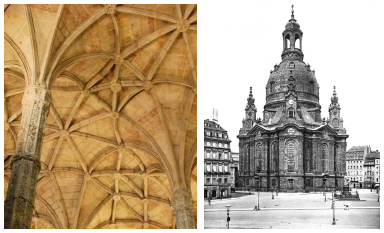

Figure 1: *Nonconvex self-supporting masonry. Left: vaults of a gothic cathedral; Right: dome of Frauenkirche, Dresden*

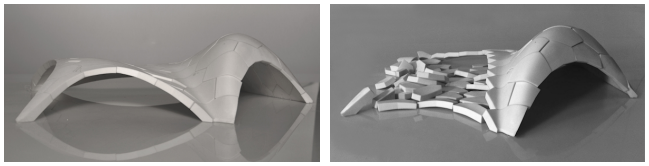

Figure 2: *Freeform masonry vault (left) and failure of the same (right). These are mock-up models from the Block Research Group at ETH Zürich.*

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

Keywords: TODO

Links: DL PDF

1 Introduction

TODO gentle introduction to self-supporting surfaces, assumptions about the model, importance to architecture etc

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, and why they continue to be an active topic of research in today's engineering community. The shapes range from convex to non-convex to freeform (see Figures 1 and 2).

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

Assumption 1: Masonry has no tensile strength, but the individual building blocks do not slip against each other (because of friction or mortar). On the other hand, their compressive strength is sufficiently high so that failure of the structure is by a sudden change in geometry, such as shown by Figure 2, and not by material failure.

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

Our approach is twofold: We first investigate the continuous case of a smooth surface under stress which turns out to be governed 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., the Airy function is convex).

For computational purposes, stresses are discretized as a fictitious *thrust network* 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 continuous case, and it turns out to have very interesting geometry: we have a polyhedral Airy stress function and a reciprocal force diagram, both of which are present in earlier work. Our own contributions are the following:

Contributions. TODO: unchanged

- We connect the smooth theory of self-supporting surfaces with vertical loads to the geometry of isotropic 3-space, whose isotropic direction is the direction of gravity (§2.3). In this view the stress surface $\phi(x, y)$ can be viewed as a relative sphere of a dual relative geometry, and one can express the characterizing equations of a self-supporting surface (1) in terms of curvatures.

- We discretize the smooth theory to polyhedral thrust networks (§2.4), and derive an analogous equivalence between thrust networks in static equilibrium and those that satisfy a condition on

their *discrete* isotropic and relative curvatures.

- We present an optimization algorithm for 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 relationship between a self-supporting surface and its relative sphere to find planar quadrilateral (PQ) representations of thrust networks (§5), and construct particularly nice families of self-supporting surfaces (§6.3).
- We demonstrate with examples the versatility and applicability of our approach to the design and analysis of large-scale masonry and steel-glass structure (§6).
- TODO graph Laplacian.

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]. Computational solutions for finding compressive force networks entirely contained within the boundary of a masonry structure have studied by various authors. See e.g. [Livesley 1992], [O'Dwyer 1998], and the many papers by Philippe Block and coauthors, from which we point only to [Block and Ochsendorf 2007]. The thrust-network method was shaped into a tool to work with.

Contributions in Computer Graphics include [Whiting et al. 2009], where structural feasibility of masonry is incorporated into procedural modeling of buildings, and [Kilian and Ochsendorf 2005] who employ particle-spring systems for form finding.

As to Mathematics and in particular polyhedral stress surfaces, 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 nonconforming finite element discretization of the former. A unifying view on polyhedral surfaces, compressive forces and corresponding 'convex' force diagrams is presented by [Ash et al. 1988].

TODO [Giaquinta and Giusti 1985]; isotropic geometry Strubecker, [Koenderink and van Doorn 2002], etc PQ meshes [Schiftner and Balzer 2010], [Glymph et al. 2004], [Pottmann et al. 2007], [Zadravec et al. 2010], [Wardetzky et al. 2007], etc

2 Self-supporting Surfaces

2.1 The Continuous Theory

We are here modeling masonry as a surface S 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

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

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

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

$$M = \left(\begin{smallmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{smallmatrix}\right) \iff \widehat{M} = \left(\begin{smallmatrix} m_{22} & -m_{12} \\ -m_{12} & m_{11} \end{smallmatrix}\right)$$

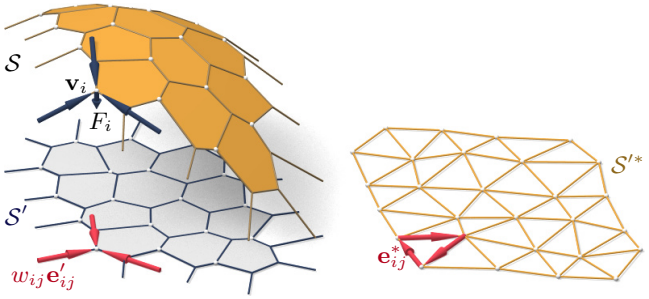

Figure 3: *Self-supporting thrust network $\mathcal{S}$, dangling edges indicating the directions of external forces (left). This network together with compressive forces which balance vertical loads F_i projects onto a planar mesh $\mathcal{S}'$ with equilibrium compressive forces "$w_{ij}\mathbf{e}'_{ij}$"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

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 $\widehat{M}$) then characterizes *convexity* of the Airy potential ϕ.

Remark: 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, and 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 $\Delta_\phi s = F$.

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 $\langle M\nabla s, \mathbf{n}\rangle = 0$.

2.2 Discrete Theory: Thrust Networks

We are now going to discretize a self-supporting surface by a polyhedral mesh $\mathcal{S} = (V, E, F)$. 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

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

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

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.

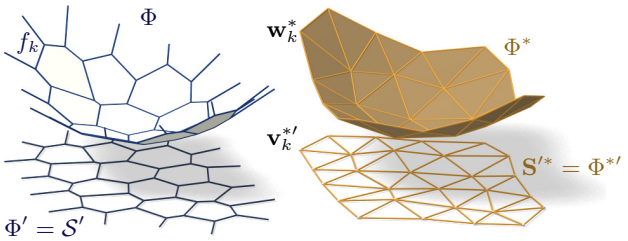

Figure 4: *Airy stress potential Φ and its polar dual Φ^*. Φ projects onto the same planar mesh as $\mathcal{S}$ does, while Φ^* projects onto the reciprocal force diagram. A primal face f_k lies in the plane $z = \alpha x + \beta y + \gamma \iff$ the corresponding dual vertex is $\mathbf{w}_k^* = (\alpha,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

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

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

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

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

form a closed cycle (see Figure 3). 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 $\mathbf{v}'^*_i$.

Polyhedral Stress Potential. We can go further and construct a convex polyhedral 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 $\mathcal{S}'^*$. 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.

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]).

Even if forces are not compressive, we can construct an Airy mesh Φ which has planar faces, but will no longer be a convex polyhedron.

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 $\Phi^{*\prime} = \mathcal{S}'^*$ (see Fig. 4).

Remark: The edge weights w_{ij} may be used to define a graph Laplacian Δ_ϕ which acts on a vertex-based function s $(\Delta_\phi s)_i = \sum_{\mathbf{v}_j \sim \mathbf{v}_i} w_{ij}(s_j - s_i)$. It is symmetric and semidefinite. Equation (2) directly implies linear precision for the planar 'top view mesh' $\mathcal{S}'$ (i.e., $\Delta_\phi f = 0$ if f is a linear function). Furthermore, Δ_ϕ-harmonic functions enjoy a maximum principle. Equation (3) can be written as $\Delta_\phi s = AF$.

2.3 Surfaces in Isotropic Geometry

It is worth wile 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*: $(x, y, z) \parallel (x', y', z') \iff x = x',\ y = y'$.

In the differential geometry of surfaces one considers the *Gauss map* σ from a surface S to a convex gauge body 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 unit sphere (so that σ maps each point its unit normal vector), leading to the ordinary curvatures H and K.

Computing Curvatures. In our setting, parallelity is a property of *points* rather than lines, and the Gauss map σ goes the other way, mapping the tangent planes of the gauge body $z = \phi(x, y)$ 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 $\dot{\mathbf{u}}$, and the matrix of $d\sigma$ is found as $(\nabla^2\phi)^{-1}(\nabla^2 s)$. By definition, curvatures of the surface s *relative* to ϕ are found as

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

The Maxwell paraboloid $\phi_0(x, y) = \frac{1}{2}(x^2 + y^2)$ is called the *unit sphere* of isotropic geometry, its Hessian equals E_2. Curvatures relative to that gauge body are not called 'relative'. We get

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

(for the last formula we have used $\operatorname{tr}(M\nabla^2\phi) = \operatorname{tr}(E_2) = 2$).

Relation to Self-supporting Surfaces. Applying the definitions above to the convex Airy stress potential ϕ of a self-supporting surface, we rewrite the balance conditions (1) as

$$2K_\phi H_s^{\text{rel}} = F. \tag{4}$$

Let us draw some conclusions:

- Since $H_\phi^{\text{rel}} = 1$ we see that the load $F_\phi = 2K_\phi$ is admissible for the stress surface $\phi(x, y)$, which is hereby shown as self-supporting. The quotient of admissible loads yields

$$H_s^{\text{rel}} = F/F_\phi. \tag{5}$$

- If the stress surface coincides with the Maxwell paraboloid, then *constant loads characterize constant mean curvature surfaces*, because we get $K_\phi = 1$ and $H_s = F/2$.
- If s_1, s_2 have the same stress potential ϕ, then $H_{s_1-s_2}^{\text{rel}} = H_{s_1}^{\text{rel}} - H_{s_2}^{\text{rel}} = 0$, so $s_1 - s_2$ is a 'relative' minimal surface.

2.4 Meshes in Isotropic Geometry

A general theory of curvatures of polyhedral surfaces with respect to a gauge body was proposed by [Pottmann et al. 2007], and its dual complement in isotropic geometry was elaborated by [Pottmann and Liu 2007]. The mean curvature of a self-supporting surface $\mathcal{S}$ *relative* to its discrete Airy stress potential is associated with the vertices of $\mathcal{S}$. It is computed from areas and mixed areas of faces in the polar polyhedra $\mathcal{S}^*$ and Φ^* which correspond to the vertex $\mathbf{v}_i$:

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

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

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

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

This is an immediate consequence of the following

Lemma. $2A_i(\mathcal{S},\Phi) = \sum_{j\sim i} w_{ij}(s_j - s_i)$.

Proof. Consider edges $\mathbf{e}_1', \ldots, \mathbf{e}_n'$ emanating from $\mathbf{v}_i'$, and the dual cycles in $\Phi^{*\prime}$ and $\mathcal{S}^{*\prime}$ which without loss of generality are given by vertices $(\mathbf{v}_1^{*\prime}, \ldots, \mathbf{v}_n^{*\prime})$ and $(\mathbf{w}_1^{*\prime}, \ldots, \mathbf{w}_n^{*\prime})$, respectively. The former has edges $\mathbf{w}_{j+1}'^* - \mathbf{w}_j'^* = w_{ij} J\mathbf{e}_j'$ (indices modulo n).

Without loss of generality $\mathbf{v}_i = 0$, so the vertex $\mathbf{v}_j'^*$ equals the gradient of the linear function $\mathbf{x} \mapsto \langle \mathbf{v}_j'^*, \mathbf{x}\rangle$ defined by the properties $\mathbf{e}_{j-1}' \mapsto s_{j-1} - s_i$, $\mathbf{e}_j' \mapsto s_j - s_i$.

The two polygons have parallel edges, because $\langle \mathbf{v}_{j+1}' - \mathbf{v}_j', \mathbf{e}_j'\rangle = (s_j - s_i) - (s_j - s_i) = 0$. Now 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 $\det(\mathbf{a}, J\mathbf{b}) = \langle \mathbf{a}, \mathbf{b}\rangle$. □

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

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

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

Discrete Balance Equation. We now prove the discrete analogue to Equation (4).

Theorem. *A simply-connected mesh $\mathcal{S} = (V,E,F)$ with vertices (x_i, y_i, s_i) can be put into static equilibrium with vertical forces $\{F_i\}_{\mathbf{v}_i \in V}$ 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*

$$2K(\mathbf{v}_i)H^{\text{rel}}(\mathbf{v}_i) = F_i \tag{6}$$

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 of $w_{ij} \geq 0$ with convexity of Φ. It remains to show that Equations (2) and (6) are equivalent. This is the case because the proposition above implies $2K(\mathbf{v}_i)H^{\text{rel}}(\mathbf{v}_i) = 2\frac{A_i(\Phi,\Phi)}{A_i}\frac{A_i(\Phi,\mathcal{S})}{A_i(\Phi,\Phi)} = \frac{1}{A_i}(\sum_{j\sim i} w_{ij}(s_j - s_i))$. □

3 Thrust Networks from Reference Meshes

Consider now the problem of taking a given reference mesh, $\mathcal{R} = (\tilde{V}, \tilde{E}, \tilde{F})$ with vertices $\tilde{\mathbf{v}}_i$, and finding a combinatorically equivalent mesh $\mathcal{S}$ in static equilibrium approximating $\mathcal{R}$. The loads on $\mathcal{S}$ include user-prescribed loads as well as the self-weight of the mesh (which depends on the vertices $\mathbf{v}_i$ of $\mathcal{S}$.) Conceptually, finding $\mathcal{S}$ amounts to minimizing some formulation of distance between $\mathcal{R}$ and $\mathcal{S}$, subject to constraints (2), (3), and $w_{ij} \geq 0$. For any choice of distance this minimization will be a nonlinear, nonconvex, inequality-constrained variational problem that cannot be efficiently solved in practice. Instead we propose a staggered optimization algorithm:

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

Estimating Self-load. Since we need to include the self-weight of the surface as a component of the load on $\mathcal{S}$, F_i depends not only on the top view of $\mathcal{S}$ but also on the surface area of its faces. To avoid adding nonlinearity to the algorithm, we estimate F_i at the beginning of each iteration, and assume it remains constant until the next iteration. We estimate the mass at each vertex by calculating its Voronoi area on each of its incident faces, and then multiplying by a user-specified surface density ρ.

Fit a Stress Surface. In this step, we fix $\mathcal{S}$ and try to fit a stress surface Φ subordinate to the top view $\mathcal{S}'$ of the primal mesh. We do so by searching for convex face normals for 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:

$$\min_{w_{ij}} \sum_i \left\| \left[\sum_{j\sim i} w_{ij}(\mathbf{v}_j - \mathbf{v}_i)\right] - (0,0,A_iF_i)\right\|^2$$
$$\text{s.t.} \quad 0 \leq w_{ij} \leq w_{\max},$$

where the outer sum is over the interior and free boundary vertices, and $w_{\max}$ 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 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 $\mathcal{S}$ is self-supporting – we are done. Otherwise, $\mathcal{S}$ is not self-supporting and its vertices must be moved. Reference the least-squares implementation I actually use in the code. Right now I'm using some BCLS code from UBC that I'm not very happy with.

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

While this approach would yield a self-supporting $\mathcal{S}$, this mesh is often far from the reference mesh $\mathcal{R}$, since any local errors in the stress surface from step 3 amplify into global errors in $\mathcal{S}$. We propose instead to look for new positions that decrease the imbalance in the stresses and loads, while also penalizing drift away from the reference mesh:

$$\min_{\mathbf{v}} \sum_i \left\| \left[\sum_{j\sim i} w_{ij}(\mathbf{v}_j - \mathbf{v}_i) \right] - (0,0,A_iF_i) \right\|^2 + \alpha \sum_i \left(\mathbf{n}_i \cdot \left(\mathbf{v}_i - \mathbf{v}_i^0\right)\right)^2 + \beta \left\|\mathbf{v} - \mathbf{v}_P^0\right\|^2,$$

where $\mathbf{v}_i^0$ is the position of the i-th vertex at the start of this step of the optimization, $\mathbf{n}_i$ is the starting vertex normal (computed as the average of the incident face normals), $\mathbf{v}_P^0$ is the projection of $\mathbf{v}^0$ onto the reference mesh, and $\alpha > \beta$ are penalty coefficients that are decreased every iteration of steps 2-4 of the algorithm. The second term allows $\mathcal{S}$ to slide over itself (if doing so improves equilibrium) but penalizes drift in the normal direction. The third term, weaker than the second, regularizes the optimization by preventing large drift away from the reference surface or excessive tangetial sliding.

Geometric interpretation of this step. It is *almost* fixing the dihedral angles of the stress surface, and adjusting the top view to improve the integrability of the stress surface – this would be the case if the edges of the top view were constrained to have fixed lengths. (We don't want to do this, of course, since that would make the top view uselessly rigid (not to mention make the optimization non-linear.) I need to reflect more on this.

Solving this weighted least-squares problem amounts to solving a sparse, symmetric linear system. While the MINRES algorithm [**?**] is likely the most robust method for solving this system, in practice we have observed that the method of conjugate gradients works well despite the potential singularity of the objective matrix.

3.1 Convergence

This algorithm is not guaranteed to always converge; this fact is not surprising from the physics of the problem (if the boundary of the reference mesh encloses too large of a region, $w_{\max}$ is set too low, and the density of the surface too high, a thrust network in equilibrium simply does not exist – the vault is too ambitious and cannot be built to stand; pillars are needed.) We can, however, make a few remarks. Step 3 always decreases the equilibrium energy

$$E = \sum_i \left\| \left[\sum_{j\sim i} w_{ij}(\mathbf{q}_j - \mathbf{q}_i) \right] - (0,0,A_iF_i) \right\|^2$$

and step 4 does as well as $\beta \to 0$. Moreover, as $\alpha \to 0$ and $\beta \to 0$, step 4 approaches a linear system with as many equations as unknowns; if this system has full rank, its solution sets $E = 0$. These facts suggest that the algorithm should generally converge to a thrust network in equilibrium, provided that step 2 does not increase the loads by too much at every iteration, and this is what we observe in practice. It would be nice if I could prove that, for sufficiently large $w_{\max}$, step 2 cannot cause the algorithm to fail to converge, but I don't yet know how to approach this problem.

4 Design of Self-Supporting Surface

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. Fitting this thrust network does not require that the user specify boundary tractions, and although the top view of the reference mesh is used as an initial guess for the top view of the thrust network, the search is not restricted to this top view.

Some 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 her edits on whether or not the surface can stand.

- Specification of boundary conditions. Points of contact between the reference surface and the ground or environment are specified by "pinning" vertices of the surface, specifying that the thrust network must coincide with the reference mesh at this point, and relaxing the condition that forces must be in equilibrium there.

- Interactive adjustment of surface density ρ, external loads, and maximum permissible stress per edge $w_{\max}$, with visual feedback of how these parameters affect the fitted thrust network.

- Upsampling of the thrust network through Catmull-Clark subdivision [**?**] and polishing of the resulting refined thrust network using optimization (§3).

- Visualization of the stress surface R dual to the thrust network.

TODO more to come?

4.1 Limitations

TODO Limits on the size of the mesh due to performance; cases where the optimization breaks down (bad reference geometry), etc

5 Self-supporting Quad Meshes with Planar Faces

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

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

$$\text{vol} = \tfrac{1}{6} \det(\mathbf{a}_1, \mathbf{a}_2, (\mathbf{a}_1)^T \nabla^2 s\, \mathbf{a}_2) + \cdots,$$
$$\text{where } \mathbf{a}_1 = \begin{pmatrix} x_{i+1,j} - x_{ij} \\ y_{i+1,j} - y_{ij} \end{pmatrix}, \quad \mathbf{a}_2 = \begin{pmatrix} x_{i,j+1} - x_{ij} \\ y_{i,j+1} - y_{ij} \end{pmatrix},$$

and the dots indicate higher order terms. Not only must the mesh $\mathcal{S}$ approximate the surface $s(x, y)$, also the corresponding polyhedral Airy surface Φ must approximate $\phi(x, y)$. Since meshes $\mathcal{S}, \Phi$ have the same top view edges $\mathbf{a}_1, \mathbf{a}_2$, and both are required to have planar faces, we get the conditions

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

Thus $\mathbf{a}_1, \mathbf{a}_2$ are eigenvectors of $(\nabla^2\phi)^{-1}\nabla^2 s$. In view of (§2.3), $\mathbf{a}_1, \mathbf{a}_2$ indicate the principal directions of the surface $s(x, y)$ relative to $\phi(x, y)$ (see Figure 5).

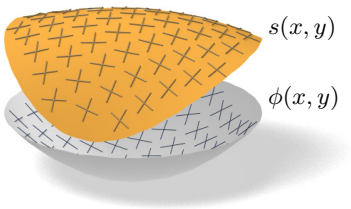

Figure 5: *A planar quad mesh approximating a self-supporting surface $s(x, y)$ with stress potential $\phi(x, y)$ is guided by the eigenvectors of $(\nabla^2\phi)^{-1}\nabla^2 s$.*

In the discrete case, where s, ϕ are not given as continuous surfaces, but are represented by a mesh in equilibrium and its Airy mesh, we use the techniques of Schiftner [2007] and Cohen-Steiner and Morvan [2003] to approximate $\nabla^2 s$, $\nabla^2\phi$, compute principal directions as eigenvectors of $(\nabla^2\phi)^{-1}\nabla^2 s$, and subsequently find meshes $\mathcal{S}, \Phi$ approximating s, ϕ which follow those directions. Subseqent global optimization makes $\mathcal{S}, \Phi$ a valid thrust network with discrete stress potential. Convexity of Φ ensures that $\mathcal{S}$ is self-supporting.

6 Results

6.1 Concrete Vaults

6.2 PQ Steel-Glass Structures

6.3 Koenig Mesh

7 Conclusion and Future Work

TODO

Acknowledgements

References

ASH, P., BOLKER, E., CRAPO, H., AND WHITELEY, W. 1988. Convex polyhedra, Dirichlet tessellations, and spider webs. In *Shaping space (Northampton, Mass., 1984)*. Birkhäuser, Boston, 231–250.

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.

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.

GIAQUINTA, M., AND GIUSTI, E. 1985. Researches on the equilibrium of masonry structures. *Archive for Rational Mechanics and Analysis 88*, 4, 359–392.

GLYMPH, J., SHELDEN, D., CECCATO, C., MUSSEL, J., AND SCHOBER, H. 2004. A parametric strategy for free-form glass structures using quadrilateral planar facets. *Automation in Construction 13*, 2, 187 – 202.

HEYMAN, J. 1966. The stone skeleton. *Int. J. Solids Structures 2*, 249–279.

HEYMAN, J. 1995. *The Stone Skeleton: Structural Engineering of Masonry Architecture*. Cambridge University Press.

KILIAN, A., AND OCHSENDORF, J. 2005. Particle-spring systems for structural form finding. *J. Int. Assoc. Shell and Spatial Structures 46*, 77–84.

KOENDERINK, J., AND VAN DOORN, A. 2002. Image processing done right. In *Computer Vision – ECCV 2002*, A. Heyden et al., Eds., no. 2350 in LNCS. Springer, 158–172.

LIPMAN, Y., SORKINE, O., COHEN-OR, D., LEVIN, D., ROSSI, C., AND SEIDEL, H. 2004. Differential coordinates for interactive mesh editing. In *Shape Modeling Applications, 2004. Proceedings*. 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.

O'DWYER, D. 1998. Funicular analysis of masonry vaults. *Computers and Structures 73*, 187–197.

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 Association, 42–51.

WARDETZKY, M., MATHUR, S., KÄLBERER, F., AND GRINSPUN, E. 2007. Discrete Laplace operators: No free lunch. In *Symposium on Geometry Processing*, A. Belyaev and M. Garland, Eds. 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.

ZADRAVEC, M., SCHIFTNER, A., AND WALLNER, J. 2010. Designing quad-dominant meshes with planar faces. *Computer Graphics Forum 29*, 5, 1671–1679. Proc. SGP.