

Parametric Self-supporting Surfaces via Direct Computation of Airy Stress Functions

Masaaki Miki*
The University of Tokyo

Takeo Igarashi†
The University of Tokyo

Philippe Block‡
ETH Zurich

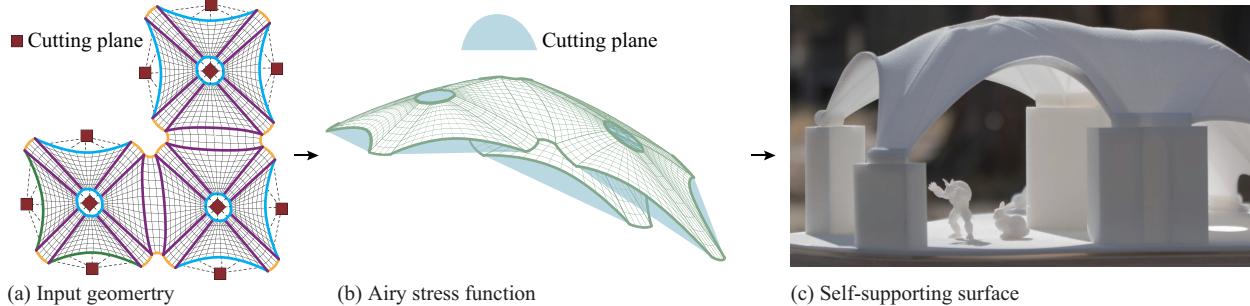


Figure 1: An Airy stress function is directly computed by the proposed computational approach to self-supporting surfaces.

Abstract

This paper presents a method that employs parametric surfaces as surface geometry representations at any stage of a computational process to compute self-supporting surfaces. This approach can be differentiated from existing relevant methods because such methods represent surfaces by a triangulated mesh surface or a network consisting of lines. The proposed method is based on the theory of Airy stress functions. Although some existing methods are also based on this theory, they apply its discrete version to discrete geometries. The proposed method simultaneously applies the theory to parametric surfaces directly and the discrete theory to the edges of parametric patches. The discontinuous boundary between continuous patches naturally corresponds to ribs seen in traditional vault masonry buildings. We use nonuniform rational B-spline surfaces in this study; however, the basic idea can be applied to other parametric surfaces. A variety of self-supporting surfaces obtained by the proposed computational scheme is presented.

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

Keywords: NURBS surface, Airy stress function, self-supporting surface

*e-mail: masaakim@acm.org

†e-mail: takeo@acm.org

‡e-mail: pblock@ethz.ch

ACM Reference Format

Miki, M., Igarashi, T., Block, P. 2015. Parametric Self-Supporting Surfaces via Direct Computation of Airy Stress Functions. *ACM Trans. Graph.* 34, 4, Article 89 (August 2015), 12 pages. DOI = [10.1145/2766888](https://doi.acm.org/10.1145/2766888)

Copyright Notice

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH '15 Technical Paper, August 09 – 13, 2015, Los Angeles, CA.
Copyright 2015 ACM 978-1-4503-3331-3/15/08 ... \$15.00.
DOI: [http://doi.acm.org/10.1145/2766888](https://doi.acm.org/10.1145/2766888)

1 Introduction

A self-supporting surface can support its self-weight by a purely compressive stress field. A structure of this class does not exhibit bending when design loading (typically the surface structure's self-weight plus any other dead loads) is acting on it; therefore, employing self-supporting surfaces is advantageous for construction of compressive structures, such as masonry and reinforced concrete shells. Once a self-supporting surface has been obtained, the shape will then be relayed to a structural engineering stage, typically using a series of finite element analyses, deformation, and the computation of strain and stress fields, in which both dead (design) and live loads are considered. Snow, wind, and seismic loads are typical examples of live loads.

In recent studies related to graphics, several authors have presented computational methods for self-supporting surfaces [Vouga et al. 2012; Panizzo et al. 2013; Liu et al. 2013; de Goes et al. 2013; Tang et al. 2014]. To some extent, those methods are based on Thrust Network Analysis (TNA), which was first presented by Block et al. [2007], or on discrete Airy potential polyhedral, which was first presented by Fraternali et al. [2002; 2014]. Note that Fraternali later mentioned Block's TNA [2010], implying a strong connection between TNA and discrete Airy potential polyhedral.

However, all those methods model surfaces with a triangulated mesh surface or a line network. Although applications of interesting concepts such as dual/reciprocal diagrams and a discrete Hodge star have been posed, these methods cannot handle continuous parametric surfaces without discretization. This is unfortunate because architects directly manipulate nonuniform rational B-spline (NURBS) surfaces in CAD software, such as Rhinoceros® and Blender®, and it is desirable to represent self-supporting surfaces in the same parametric representation without discretization. Another approach is to directly determine stress functions through elementary mathematical functions [Williams 1990]. However, there is only limited design freedom of stress functions in this approach.

Thus, we have developed a method that employs NURBS surfaces as surface representations at any stage of the computational process to compute parametric self-supporting surfaces. The proposed method is based on the theory of Airy stress functions, similar to some existing methods. However, while existing methods are based

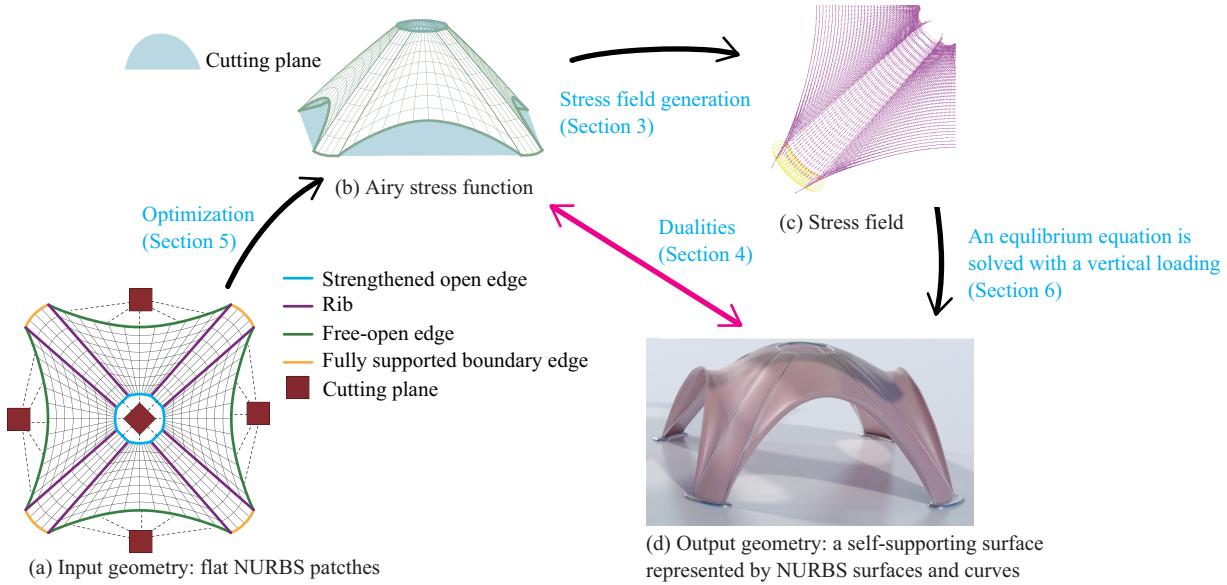


Figure 2: Overview of the proposed computational scheme.

on a discrete version of the theory of Airy stress functions or the equivalent reciprocal diagram, we first interpret the theory in curvilinear coordinate system and then apply it to NURBS patches directly. Note that the discrete version of the Airy stress function is applied to the edges of NURBS patches. As a result of combining discrete and continuous theories, plenty of beautiful dualities are observed in the proposed method. These dualities uniquely transform geometric conditions to mechanical conditions and, from a technical perspective, satisfying such geometric conditions is more realistic than struggling with mechanical conditions directly.

Our contributions are summarized as follows.

- We present a scheme to represent a vault masonry surface as a collection of continuous parametric patches. The discontinuous boundary between these continuous patches naturally corresponds to ribs in traditional vault masonry buildings (Section 2).
- We provide an interpretation of the Airy stress functions (continuous) in a curvilinear coordinate system (Sections 3.1 and 3.2), apply a discrete version of Airy stress functions to the edges of parametric surfaces (Sections 3.3 and 3.4), and introduce concepts of cutting planes (Section 3.4) that are attached to the unsupported (i.e., open) boundary edges.
- We enumerate and visually explain all dualities found between Airy stress functions and self-supporting surfaces (Section 4).
- We present an optimization scheme to obtain an Airy stress function that satisfies all conditions necessary for compression-only structure (Section 5).
- We present a method to form and solve equilibrium problems after an appropriate stress field for self-supporting surfaces is obtained from an Airy stress function (Section 6).
- We examine variations of self-supporting surfaces that can be obtained by the proposed computational scheme (Section 7). We also indicate that specifying lower and upper bounds of some parameters in computations of Airy stress functions allows the user more control of the design of self-supporting

surfaces.

This paper requires that the readers have existing knowledge of the following geometrical/algebraic concepts: covariant derivatives, Hodge star operator, differential forms, and Einstein summation convention including raising and lowering indices laws [Eisenhart 1947; Aris 1962; Darling 1994; Eisenhart 1997; Ciarlet 2006; Arnold et al. 2006; Arnold et al. 2010].

Note that our NURBS-based computational scheme connects well to recent trends in computational mechanics that are directed toward NURBS-based finite element methods. This new trend is called isogeometric analysis [Hughes et al. 2005; Cottrell et al. 2009; Simpson et al. 2012]. Such state-of-the-art analysis methods primarily focus on the consistent use of NURBS from design to structural analysis, similar to our work. Our computational scheme contains geometric restrictions as input geometries that must be represented by watertight NURBS patches; however, NURBS-compatible surface representations that potentially alleviate such geometric restrictions are investigated in both graphics and computational mechanics (such as T-splines, PHT-splines and LR B-splines) [Sederberg et al. 2003; Li et al. 2007; Sederberg et al. 2008; Bazilevs et al. 2010; Dörfel et al. 2010; Scott et al. 2013; Buffa et al. 2014; Johannessen et al. 2014] and extending our computational scheme to those NURBS-compatible surface representations would be possible. For example, it is difficult to extend our computational scheme to trimmed NURBS surfaces; however, such a trimmed shape might be well represented by T-splines [Sederberg et al. 2008].

2 NURBS Patch Representation for a Masonry Surface

To represent a masonry surface as a parametric surface, we present a method to represent a masonry surface as a collection of NURBS patches and a computational scheme to compute its three-dimensional geometry. A NURBS patch is a surface region bounded by four boundary curves, each of which is connected to one or two NURBS patches. A boundary curve represents discontinuity between continuous smooth surface patches. The boundary curve naturally corresponds to ribs seen in traditional vault masonry

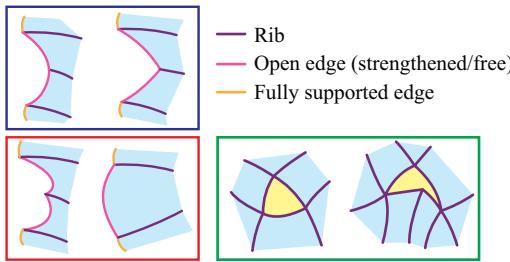


Figure 3: Input shapes of the proposed computational scheme are not completely arbitrary. Examples in the blue frame show good illustrations of open edges, whereas those in the red frame represent bad illustrations. Examples in the green frame show holes that cannot be filled by a NURBS patch.

buildings.

As shown in Fig. 2(a), inputs to the proposed computational scheme are watertight flat NURBS patches and link information of boundary edges with cutting planes. Then, an Airy stress function represented by the same NURBS interpolation functions as the input is computed using an optimization solver (Fig. 2(b), Section 5). An Airy stress function must satisfy a number of conditions; therefore, it is not possible to place control points at arbitrary locations. In Section 5.4, we present two ideas to address this problem. From the obtained Airy stress function, a stress field is computed (Fig. 2(c)) on the basis of the theories described in Section 3. When a stress field is obtained, the equilibrium problem is formed and solved (Fig. 2(d), Section 6). As a result, a self-supporting surface represented by NURBS is obtained (Section 7). As discussed in Section 7, a resultant self-supporting surface has the same planar projection as the Airy stress function and input geometry.

Note that ribs play a critical role in our formulation. They can be explained as one-dimensional concentrations of stress in the self-supporting surface, because they are yielded from discontinuities in the Airy stress function. Along these creases, a thicker section is locally needed when translating the theoretical surface into a real structure. Thus, these ribs are structurally self-supporting spatial arches that follow the creases of the self-supporting solution, and carry the self-supporting continuous patches between them. The horizontal equilibrium between ribs and patches is supplemented in Appendix A. The ribs are essential components of our representation; it is impossible to remove them completely. However, as discussed in Section 7, it is technically possible to make the magnitude of stress (i.e., axial forces) in the ribs nearly zero. Using this technique, one can obtain an approximately no-ribbed surface. Similar to ribs, open (i.e., unsupported boundary) edges are also strengthened by adding self-supporting arches. Free-open edges (i.e., unsupported boundary edges that are not strengthened by arches) are also achieved approximately using the same technique that is applied to the ribs.

Input geometry is not fully arbitrary and a user must be sufficiently trained beforehand. First, the geometry must be represented by watertight NURBS patches. Each patch must be four-sided. Thus, if a hole that cannot be filled by a NURBS patch appears (such as the cases shown in the green frame in Fig. 3), the design must be performed again. Second, the shape must ensure the existence of Airy stress functions that satisfy all necessary conditions. Particularly, as indicated by Fig. 3, open edges must be perfectly concave while fully supported edges can take arbitrary shapes.

3 Interpretations of Existing Continuum and Discrete Airy Stress Functions

An Airy stress function is a height-field defined on a domain that is a planar projection of a structure. A unique stress distribution on the same domain can be obtained from an Airy stress function. Such a stress field automatically satisfies some of the conditions that a stress field should satisfy. We refer interested readers to [Gurtin 1963]. Gurtin discusses a three-dimensional case of stress functions but mentions Airy's contribution briefly in its introduction.

In Section 3.1, we briefly describe continuum Airy stress functions in a Cartesian coordinate system. In Section 3.2, we interpret the theory in a curvilinear coordinate system to apply the theory to parametric surfaces. The highlight of this paper is found in the transition from Section 3.3 to Section 3.4. In Section 3.3, we briefly describe the discrete version of the Airy stress function, which was first presented by Fraternali et al. [2002]. Then, in Section 3.4, we apply the discrete version to curved parametric surface edges and introduce ideas of cutting planes that trace the boundary edges.

3.1 Continuum Airy stress functions in a Cartesian coordinate system

The theory of Airy stress functions is typically described with a Cartesian coordinate system. We denote the Cartesian coordinates by (x_1, x_2) . Note that we use capital letters for quantities measured on the Cartesian coordinate system. In addition, we use α, β, γ for the indices of Cartesian coordinates. An Airy stress function $\phi(x_1, x_2)$ can be any scalar function that has second derivatives in the given domain. For a given ϕ , a stress tensor $S_{\alpha\beta}$ is computed as follows:

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} H_{22} & -H_{21} \\ -H_{12} & H_{11} \end{bmatrix}, \quad (1)$$

where $H_{\alpha\beta}$ is the Hessian of ϕ , i.e., $H_{\alpha\beta} = \partial_\alpha \partial_\beta \phi$. $\partial_\alpha \partial_\beta \partial_\gamma \phi$ is symmetric in (α, β) ; thus, we observe the following:

$$\frac{\partial S_{\alpha\beta}}{\partial x_\alpha} = 0.$$

This means that the obtained tensor field $S_{\alpha\beta}$ is free of divergence. Another understanding of this formula is that any stress field $S_{\alpha\beta}$ that is yielded from any ϕ is horizontally self-equilibrated. If one applies the obtained tensor field to the second Piola–Kirchhoff (PK) stress field of a structure, a vertical loading does not affect horizontal equilibrium. Thus, finding such a nice horizontal stress distribution is a good starting point for designing a self-supporting surface.

If we explicitly represent the basis of $H_{\alpha\beta}$ in equations, we obtain $H = H_{\alpha\beta} dx_\alpha \otimes dx_\beta$. Similarly, $S = S_{\alpha\beta} dx_\alpha \otimes dx_\beta$. Using the Hodge star operator defined in exterior calculus, Equation (1) can be rewritten as follows:

$$S = *H*,$$

where, $*$ is the Hodge star operator that acts on the basis as follows:

$$*dx_1 = dx_2, *dx_2 = -dx_1.$$

These relations are referred to as Hodge's dual. We refer readers to [de Goes et al. 2014], which states that a divergence free tensor field is given by a Hodge's dual of a Hessian.

3.2 Extension to curvilinear coordinate system

To apply the above theory to NURBS patches directly, we interpret the existing theory in a curvilinear coordinate system. We denote the curvilinear coordinates as (θ^1, θ^2) . We use lower case letters for quantities measured on the curvilinear coordinate system. In addition, we use i, j, k for indices of the curvilinear coordinates.

Due to the geometric aspect of the Hodge star operator, this interpretation can be performed quickly. First, we assume that functions $x_1 = x_1(\theta^1, \theta^2)$ and $x_2 = x_2(\theta^1, \theta^2)$ are given. Metric g_{ij} of the surface is then calculated by $g_{ij} = \partial_i x_\gamma \partial_j x_\gamma$. We define $\theta_i = g_{ij} d\theta^j$ as the dual basis of $d\theta^i$. The Hodge's duals on this setup are given as follows:

$$*d\theta^1 = \sqrt{g^{-1}} \theta_2, *d\theta^2 = -\sqrt{g^{-1}} \theta_1.$$

Here $g = \det g_{ij}$. We use $h = h_{ij} d\theta^i \otimes d\theta^j$ and $s = s^{ij} \theta_i \otimes \theta_j$ as counterparts of H and S , respectively. The indices of the stress tensor s^{ij} are raised only because the following equations become simpler. For h_{ij} , we must not use the normal Hessian $\partial_i \partial_j \phi$ but the covariant Hessian, $\nabla_i \nabla_j \phi = \partial_i \partial_j \phi - \Gamma_{ij}^k \partial_k \phi$, where Γ_{ij}^k are the connection coefficients (i.e., functions of the first and second derivatives of x_1 and x_2). As tensors, $h = H$ holds. Likewise, $*h* = *H*$ holds. Finally, we determine the following:

$$s = *h*.$$

As a result, $s = S$ holds. This means that, for an implementation, we can use the following as a counterpart of Equation (1) if a curvilinear coordinate system is used:

$$\begin{bmatrix} s^{11} & s^{12} \\ s^{21} & s^{22} \end{bmatrix} = g^{-1} \begin{bmatrix} h_{22} & -h_{21} \\ -h_{12} & h_{11} \end{bmatrix}. \quad (2)$$

According to the theory of Airy stress functions, the tensor field s^{ij} must be divergence free. This is expressed as follows:

$$\nabla_i s^{ij} = 0, \quad (3)$$

where $\nabla_i s^{ij}$ represents the i -th covariant derivative of s^{ij} . The k -th covariant derivative of s^{ij} is given by $\nabla_k s^{ij} = \partial_k s^{ij} + \Gamma_{ik}^l s^{lj} + \Gamma_{lk}^j s^{il}$. The reader may think that $\nabla_i s^{11} = \nabla_i (g^{-1} h_{22})$ represents a correct relation and so on for other combinations. However, the correct relations between two different types of covariant derivatives are expressed as follows:

$$\nabla_i s^{11} = g^{-1} \nabla_i h_{22}, \nabla_i s^{12} = -g^{-1} \nabla_i h_{21},$$

$$\nabla_i s^{21} = -g^{-1} \nabla_i h_{12}, \nabla_i s^{22} = g^{-1} \nabla_i h_{11}.$$

With these four relations, we derive the followings:

$$\nabla_1 s^{11} + \nabla_2 s^{21} = g^{-1} (\nabla_1 \nabla_2 \nabla_2 \phi - \nabla_2 \nabla_1 \nabla_2 \phi),$$

$$\nabla_1 s^{12} + \nabla_2 s^{22} = g^{-1} (-\nabla_1 \nabla_2 \nabla_1 \phi + \nabla_2 \nabla_1 \nabla_1 \phi).$$

In this setup, Riemannian curvature vanishes; thus, $\nabla_i \nabla_j \nabla_k \phi$ is symmetric in (i, j) . Therefore, s^{ij} is free of divergence.

3.3 Discrete Airy stress functions

In this work, input geometries are modeled by flat NURBS patches plus the link information of boundary open edges with cutting planes, as shown in Fig. 2(a). All boundary edges except those that will be fully supported must be connected with a cutting plane. We refer to boundary edges that will not be fully supported as open edges. Two neighboring patches must have overlapping edges with

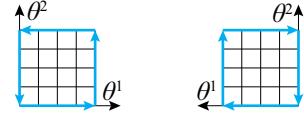


Figure 4: Boundary curves setup: having θ^2 axis upward, $\psi(t)$ goes anticlockwise when θ^1 axis points to the right and clockwise when it points to the left.

exactly the same control points, orders of interpolating functions, and knot vectors, i.e., they are G^0 continuous at the intersecting edges. If an Airy stress function is connected to the cutting planes appropriately, i.e., if all control points lie on each linked cutting plane, one can cut the resulting self-supporting surfaces at the exact position that the boundary edges are placed because the cutting planes generate exactly zero stress distributions.

The key point is that Airy stress functions are represented using the same interpolation functions as the input parametric surfaces. Therefore, the Airy stress functions do not have second derivatives on the edges where two NURBS patches intersect. In addition, other than Equation (3), a static stress distribution must also satisfy boundary equilibrium conditions on the boundary edges, with the exception of those that are fully supported. To solve these two problems with a single approach, we have focused on the theory of discrete Airy potential polyhedral [Fraternali et al. 2002].

Several authors, particularly Fraternali et al., have suggested that when an Airy stress function is represented by a polyhedral mesh, of which all faces are planar, the “jump” (Fraternali’s term) of the quantity

$$D_n = \frac{\partial \phi}{\partial n} \quad (4)$$

at each edge will appear as the horizontal component of the axial force acting in the same edge, where n represents the direction on the $x_1 - x_2$ plane that is perpendicular to the edge and $\partial/\partial n$ represents a directional differentiation with respect to n .

For our input geometries, this “jump” can also be computed on the edges shared by two patches. However, differing from polyhedral cases, this “jump” is not constant along an edge; thus, it should be computed per point of each edge. In the following, we refer to this “jump” as a kink angle. Such dihedral kink angles will appear as the horizontal component of the axial forces of ribs. In our computational scheme, ribs are added automatically, as a strengthened crease, to the edges where two patches intersect to solve discontinuity of stress along that edge (see Appendix A). In addition, two options can be chosen for boundary edges: fully supported and open (i.e., unsupported) edges. Similar to the shared edges, arches are added automatically to the open edges. The height of fully supported edges can be moved in the vertical direction per each fully supported edge.

3.4 Extension to curved edges

For convenience, we construct a closed boundary curve $\psi(t) = [\theta^1(t) \ \theta^2(t)]$ on the boundary edges of each NURBS patch. Note that we consistently construct boundary curves as shown in Fig. 4; otherwise, the consistency of the plus and minus signs in the following equations may be lost. In the following, we represent df/dt by f' , where f is an arbitrary function of t . In our implementation, only $\dot{\psi}$ is coded explicitly; however, ψ is never referenced. Denoting the metric of these curves as $\gamma = \dot{\theta}^i \dot{\theta}^j g_{ij}$, a length element is represented by $dl = \sqrt{\gamma} dt$. The quantity denoted D_n ,

represented by Equation (4), can be obtained by solving the following:

$$D_L \mathrm{d}l = \psi^* \left(\frac{\partial \phi}{\partial \theta^i} (*d\theta^i) \right),$$

where D_L represents the same quantity as D_n , ψ^* is a pullback operator that maps $d\theta^i$ to $\dot{\theta}^i dt$, and $*d\theta^i$ represents Hodge's duals of $d\theta^i$. The following can be used for an implementation:

$$D_L = \frac{\sqrt{g}}{\sqrt{\gamma}} \begin{bmatrix} \frac{\partial \phi}{\partial \theta^1} & \frac{\partial \phi}{\partial \theta^2} \end{bmatrix} \begin{bmatrix} g^{ij} \\ -\dot{\theta}^1 \end{bmatrix}, \quad (5)$$

where g^{ij} is the inverse of g_{ij} . Similarly, assuming that each cutting plane is expressed by $ax + by + c\phi + d = 0$, on an edge, the quantity denoted D_n , measured on the cutting plane side can be computed by solving the following:

$$D_C \mathrm{d}l = \psi^* \left(\mathbf{a} \cdot \mathbf{g}_i (*d\theta^i) \right),$$

where $\mathbf{a} = \frac{1}{c} \begin{bmatrix} -a & -b \end{bmatrix}$ and $\mathbf{g}_i = \begin{bmatrix} \frac{\partial x_1}{\partial \theta^i} & \frac{\partial x_2}{\partial \theta^i} \end{bmatrix}^T$. Thus, the following can be used for an implementation,

$$D_C = \frac{\sqrt{g}}{\sqrt{\gamma}} \begin{bmatrix} \mathbf{a} \cdot \mathbf{g}_1 & \mathbf{a} \cdot \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} g^{ij} \\ -\dot{\theta}^1 \end{bmatrix}. \quad (6)$$

The kink angle distribution between a NURBS patch and a cutting plane at an edge is computed as follows:

$$D = D_L - D_C.$$

Here, it is assumed that the same ψ is used in the computations of D_L and D_C . If an edge is shared by two NURBS patches, the kink angle at the edge is computed as follows:

$$D = D_L + D_R,$$

where D_R is the same quantity as D_L but measured on the other patch side. We assumed that different ψ is used in the computations of D_L and D_R .

Careful attention should be paid to the transformation of D to the axial forces in the ribs/arches. On a curve with parameter t , the dual basis of dt is γdt . Thus, on a curve, a stress tensor should have the form $n = n^{11}(\gamma dt) \otimes (\gamma dt)$. Therefore, we assumed that $n^{11} = \kappa D$, where κ is an unknown coefficient that must be identified. The candidates for κ include $1, \gamma, \gamma^{-1}, \sqrt{\gamma}, \sqrt{\gamma^{-1}}$. We tested these candidates and found that γ^{-1} always gives an expected numerical result. A comparison of these coefficients is presented in Appendix B.

Respecting the fact that reciprocal/dual diagrams and discrete Hodge stars appear frequently in discussions of self-supporting surfaces, one possible way to explain this coefficient choice is to assume that the transformation between the kink angle and axial force is carried by the Hodge star. On a curve, Hodge's duals are given as follows:

$$*dt = \frac{1}{\sqrt{\gamma}}, *1 = \sqrt{\gamma} dt.$$

If we assume that $n = * (D (1 \otimes 1)) *$, we obtain the following:

$$n^{11} = \frac{1}{\gamma} D. \quad (7)$$

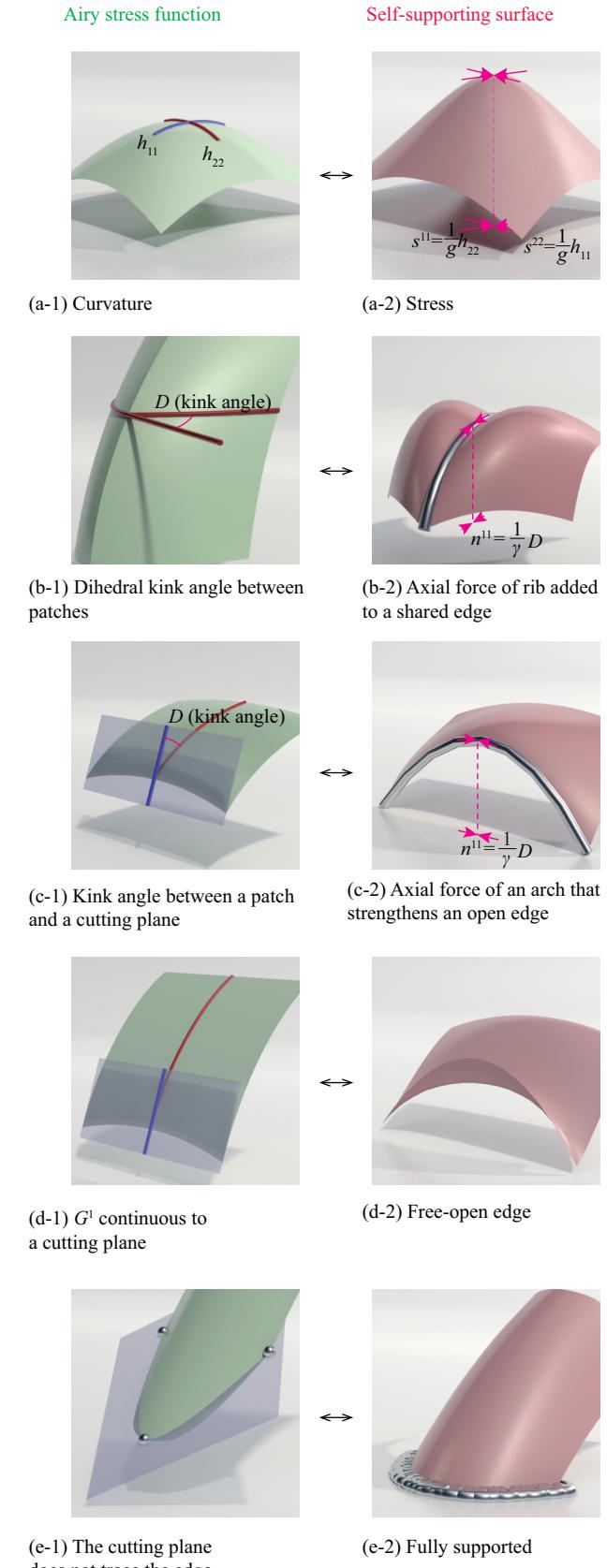


Figure 5: Dualities between Airy stress functions (left) and self-supporting surfaces (right). Note that geometric conditions are transformed to mechanical conditions uniquely.

4 Dualities

Here we present a visual way to comprehend the equations presented in the previous section, i.e., the dualities. This concept can help design a stress distribution on a self-supporting surface. Such dualities uniquely transform geometric conditions to mechanical conditions and from a technical perspective, satisfying such geometric conditions may be easier than struggling with mechanical conditions directly. Thus, Airy stress functions are demonstrated to have practical relevance.

As shown in Fig. 5(a), a curvature in one direction in an Airy stress function appears as the stress in the orthogonal direction in the self-supporting surface. We measure a curvature that bulges upward as negative (convex) in this work. If one wants a compression-only structure, curvatures must be negative in all directions at every point. As shown in Fig. 5(b), if an edge is shared by two surfaces in an Airy stress function, the kink angle distribution between the two surfaces along the edge appears as the axial force distribution of a rib added to that edge in the self-supporting surface. We always measure this kink angle as negative when convex. If one wants a compression-only structure, this kink should be convex. As shown in Fig. 5(c), if a cutting plane that intersects an Airy stress function along an edge exists and the kink angle between the Airy stress function and the cutting plane has a value, this value appears as the axial force in the arch that strengthens that open (i.e., unsupported boundary) edge in the self-supporting surface. This kink should also be convex, i.e., negative, if one wants a compression-only structure. As shown in Fig. 5(d), if the kink angle between the Airy stress function and the cutting plane along the edge is exactly 0 everywhere, i.e., if they are G^1 continuous, the horizontal thrust (perpendicular to the edge) vanishes. Thus, no force attraction appears on the edge and the edge will appear as a free-open edge in the self-supporting surface. If the cutting plane cannot be found on an edge, the force attraction at that edge cannot be equilibrated. Therefore, as shown in Fig. 5(e), that edge should be fully supported on the ground such that reaction forces can be received from the ground.

5 Computation of Airy Stress Functions

The theories described in Section 3 can only be used when an appropriate Airy stress function is given. Thus, in an implementation, computation of an Airy stress function plays a key role. By respecting all dualities described in the previous section, all conditions of Airy stress functions necessary for compression-only structures can be enumerated. As a result, an Airy stress function can be obtained using a standard optimization solver. When an Airy stress function is obtained, a stress field is computed on the basis of theories described in Section 3. Then, the equilibrium problem is formed and solved, which is described in the next section.

5.1 Compression-only conditions

In structural engineering, minus signs are usually assigned to compressive forces. To achieve a compression-only stress field, the eigenvalues of s^{ij} must be zero or negative. For the computation of such eigenvalues, we must consider the metric g_{ij} , i.e., we must solve a generalized eigenvalue problem. However, g_{ij} is supposed to be positive definite; thus, two conditions, $\text{tr} s^{ij} \leq 0$ and $\det s^{ij} \geq 0$, are necessary and sufficient for a compression-only condition. Note that $\text{tr} h_{ij} \leq 0$ and $\det h_{ij} \geq 0$ are respectively equivalent to these two conditions. By defining additional variables $\tilde{h}_{ij} = -h_{ij}$, those two conditions are equivalent to the following:

$$\tilde{h}_{11}\tilde{h}_{22} + \tilde{h}_{12}\tilde{h}_{21} \geq 0, \quad \tilde{h}_{11} \geq 0, \quad \tilde{h}_{22} \geq 0. \quad (8)$$

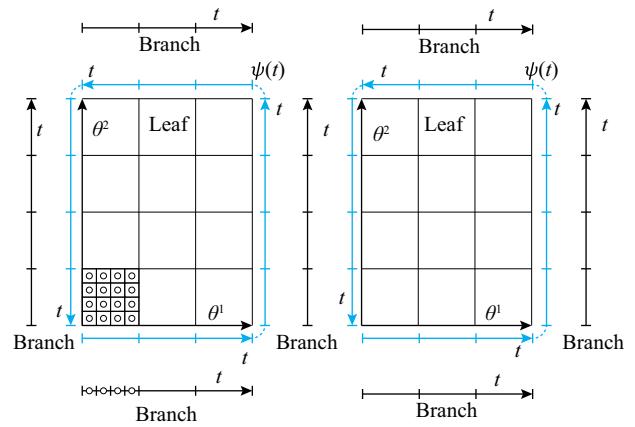


Figure 6: Leaf (NURBS patches), their boundary edges and branches (NURBS curves). Evaluation points distributed on leaves and branches are shown by white dots.

This is a second-order rotated conic inequality constraint condition. Geometrically, as mentioned in the previous section, this condition requires that the Airy stress function be convex.

If one wants a rib between two patches to be compression-only, i.e., for the case shown in Fig. 5(b), $D \leq 0$ must be satisfied on the edge, i.e.,

$$D = D_L + D_R \leq 0. \quad (9)$$

Similarly, for the case shown in Fig. 5(c),

$$D = D_L - D_C \leq 0. \quad (10)$$

For the case shown in Fig. 5(d), $D = 0$ must be satisfied, i.e.,

$$D = D_L - D_C = 0. \quad (11)$$

Unfortunately, we found that except for a few simple cases (such as the one shown in Fig. 2), it is often difficult to satisfy this strong condition. To weaken this condition, we allow the user to specify the upper and lower bounds of D so that they can manually lead D to nearly zero. Regarding the required precision for architectural construction, there is no necessity to constrain D to precisely zero even though a free-open edge is desired. The same technique can be applied to the kink angle between two NURBS patches to achieve a no-ribbed surface. Moreover, more user control can be added by making upper and lower bounds of mean curvatures defined by $H = 1/2(\tilde{h}_{11} + \tilde{h}_{22})$ controllable. In our implementation, the user is asked to perform grouping of curves and surfaces before the computation of an Airy stress function is performed. In the first computation of the Airy stress function, only the minimum necessary conditions such as $D \leq 0$ and conic inequality are taken into consideration. After the first computation, the system shows the user the minimum and maximum values of D and H per group. Then, the user may specify the lower and upper bounds of D and H . In fact, repetition of this operation allows more indirect but dynamic control over the shapes of self-supporting surfaces, which is discussed in Section 7.

5.2 Optimization

The goal of our approach is to not allow a user to design the Airy stress function manually but obtain an Airy stress function that satisfies all conditions necessary for compression-only using a general optimization solver.

In our implementation, we input to the algorithm the NURBS patches (leafs in the following) and NURBS curves (branches in the following), as shown in Fig. 6. Our algorithm can be implemented without introducing the concept of branches. However, due to the introduction of branches, one can explicitly divide the computations of curvatures and kink angles to leafs and branches. Every edge of a leaf must be tied to one of the given branches. This can only be achieved if the dimension, order, knot vector, and coordinate domains between the leaf edge and target branch are aligned appropriately. This is not considered by the proposed algorithm; thus, the input geometry must be prepared accordingly. Leafs will appropriately share their edges with neighborhoods via the links with branches. Both curvatures and kink angles are not computed per control point but evaluated at points distributed according to the procedure described below (white dots in Fig. 6). We discuss issues about cutting planes in Section 5.3.

On each leaf, $x_1(\theta^1, \theta^2)$ and $x_2(\theta^1, \theta^2)$ are given in the following form:

$$\mathbf{r}(\theta^1, \theta^2) = \sum_{i=1}^{n_1 \times n_2} N_i(\theta^1, \theta^2) \mathbf{P}_i,$$

where $\mathbf{r} = (x_1, x_2)$, n_1 and n_2 are the number of control points in θ^1 and θ^2 directions, respectively, N_i is the i -th interpolating function, and \mathbf{P}_i contains the (x, y) coordinates of the i -th control point of the leaf. As mentioned previously, we use the same interpolation function for the Airy stress function $\phi(\theta^1, \theta^2)$:

$$\phi(\theta^1, \theta^2) = \sum_{i=1}^{n_1 \times n_2} N_i(\theta^1, \theta^2) z_i,$$

where z_i is the height of the i -th control point, which is managed as an independent unknown variable in the optimization engine. Note that (x, y) coordinates of the control points are managed as known parameters and z_i are not used in the computations of metric g_{ij} or connection coefficients Γ_{ij}^k . Thus, on each leaf, the degrees of freedom of the Airy stress function are reduced to $n_1 \times n_2$. Similarly, on each branch, $x_1(t)$ and $x_2(t)$ are given in the following form:

$$\mathbf{r}(t) = \sum_{i=1}^{n_t} N_i(t) \mathbf{P}_i,$$

where n_t denotes the number of control points. Note that the kink angles are computed on the branches. As a result, the Airy stress function is not evaluated on branches.

Interpolation functions $N_i(\theta^1, \theta^2)$ and $N_i(t)$ are given by the polynomial form of (θ^1, θ^2) or t . A leaf or branch can be divided into segments such that this type of polynomial representation does not change in one segment. Neighboring segments share a certain number of control points such that a leaf or a branch becomes continuous to some degree but the polynomial representation varies from segment to segment. In the following, we denote the order of NURBS patches and curves odr_1, odr_2 , and odr_t . The highest degree of polynomial representations are usually $dm_1 = odr_1 - 1$, $dm_2 = odr_2 - 1$, and $dm_t = odr_t - 1$, respectively. Thus, leafs and branches are segmented to $(n_1 - dm_1) \times (n_2 - dm_2)$ or $(n_t - dm_t)$ segments. Without loss of generality, one can choose a coordinate system such that each segment becomes a unit square in the two-dimensional domain or a unit line in the one-dimensional domain. Thus, we consistently assume for the domains that $\Omega_2 = \{(\theta^1, \theta^2) \mid 0 \leq \theta^1 \leq (n_1 - dm_1), 0 \leq \theta^2 \leq (n_2 - dm_2)\}$ and $\Omega_1 = \{(t) \mid 0 \leq t \leq (n_t - dm_t)\}$. To satisfy convex conditions everywhere, we found that it is sufficient to distribute $(dm_1 - 1) \times (dm_2 - 1)$ evaluation points to each segment in a leaf or $(dm_t - 1)$

evaluation points to each segment in a branch. However, this number of evaluation points is inadequate to achieve sufficient precision when solving the equilibrium problems described in Section 6. Therefore, we chose greater numbers such as 4×4 for order four patches and 6×6 for order five patches. NURBS patches with orders greater than five sometimes cause undesirable high frequency waves in the solutions of equilibrium problems; therefore, we prefer order four or five NURBS patches. Specifically, we have primarily used order four patches with 8×8 control points and order five patches with 12×12 control points.

If the (x, y) coordinates of the control points are given as inputs and the heights z_i of the control points are managed as design variables, $h_{11}, h_{22}, h_{12}, h_{21}, D_L$, and D_R become linear in the design variables. Thus, an Airy stress function that satisfies all the conditions posed in the previous section can be obtained using a general optimization solver that supports conic quadratic programming. Note that we have used Mosek® for this purpose.

5.3 Cutting planes

To be in equilibrium, every boundary edge of the Airy stress function, except for those that are fully supported, must be connected with a cutting plane. A cutting plane is represented by the form $ax + by + cz + d = 0$. The degrees of freedom of a plane are always three; thus, the independent parameters are $a/c, b/c$, and d/c unless the plane becomes vertical. In this work, we assume that cutting planes never become vertical.

Thus, the link is expressed as follows:

$$z_i = -\frac{d}{c} - \frac{b}{c}y_i - \frac{a}{c}x_i, \quad (12)$$

where x_i, y_i and z_i are the coordinates of the i -th control point of an edge, which are linked to the cutting plane. Generally, a NURBS curve does not pass through the control points. However, if all control points lie on a common plane, the curve also lies on the same plane. There is no constraint condition for cutting planes; therefore, $a/c, b/c$, and d/c of each cutting plane can be simply added to the list of free design variables in the scheme. This makes D_C (Equation (6)) linear in the design variables. In addition, it makes Equation (12) linear in the design variables. Thus, cutting planes can be appropriately handled by an optimization solver.

5.4 Additional terms

The conditions discussed in Sections 5.1 and 5.3 are only the minimum conditions necessary for compression-only structures; therefore, an optimization solver may return a trivial solution (e.g., all z_i of control points are set to 0). Thus, to allow for more user control, additional terms should be added to the optimization problem. We present two approaches to address this issue.

The first approach is to control the Airy stress function partially by specifying the exact configurations of the cutting planes. This can be done by constraining $a/c, b/c$, and d/c to specific values. This may make optimization fail unless the cutting planes are placed appropriately. Therefore, this approach is not good for designing free-form self-supporting surfaces. However, because this approach has been very useful for researching the effects of cutting planes, we include it as an option.

The second approach is to specify a target function of an Airy stress function roughly. We used surface interpolation using multiquadric functions, which were first presented by Hardy et al. [1971]. A unique multiquadric surface can be computed from a finite number of point sets; thus, it is very easy to manage. Here, let $\tilde{\phi}$ be a target

value of the Airy stress function at each evaluation point. We then compute the following:

$$\tilde{\phi} = F(x_1, y_2),$$

where F is a multiquadric function computed from a point set, and (x_1, x_2) is the Cartesian coordinates of the evaluation point. The distance between the multiquadric function and an Airy stress function can be computed as follows:

$$\epsilon = \sqrt{\sum_{\text{evaluation points}} (\phi - \tilde{\phi})^2},$$

where ϕ is the value of the Airy stress function at each evaluation point. This can be transformed to a second-order conic constraint condition; thus, it can be added to an optimization problem. The simplest idea to minimize this distance would be to minimize ϵ directly by setting it as the objective in an optimization problem. However, this usually does not work. Specifically, although an obtained Airy stress function appears sufficient visually, the solutions of equilibrium problems described in the next section become too wavy. Particularly, the heights of boundary edges become unrealistically large. This indicates that an as close as possible Airy stress function to a target function can potentially yield an unexpected result. We assumed that this occurs because the resultant Airy stress function contains nearly flat regions. To avoid this problem, we found that constraining ϵ to a preferable number gives desirable results as the obtained surface is much smoother than the as close as possible one. If one wants to minimize ϵ , it would be sufficient to decrease this number manually, and when the optimization solver returns “FAILED” (UNKNOWN SOLUTION STATUS in Mosek®), the last result would be taken. The second approach constantly gives an appropriate Airy stress function; thus, it is a better approach for designing free-form self-supporting surfaces.

6 Equilibrium Problem

When an Airy stress function is obtained, a stress field denoted by s^{ij} and n^{11} is computed on the basis of the theories described in Section 3. From the obtained stress field, the equilibrium problem is formed and solved as follows. When a vertical loading is applied, a self-supporting surface is obtained.

In the following, we put $\bar{\cdot}$ to all quantities computed in the former computational stages, such as $s^{ij} \rightarrow \bar{s}^{ij}$, $n^{11} \rightarrow \bar{n}^{11}$, $g_{ij} \rightarrow \bar{g}_{ij}$, and $\gamma \rightarrow \bar{\gamma}$. We then consider them constants at each set of curvilinear coordinates (θ^1, θ^2) . In the previous sections, we managed (x, y) coordinates of the control points as known parameters and only z coordinates of the control points were managed as independent unknown variables. In the equilibrium problem, we do not separate these and manage all (x, y, z) coordinates of the control points as unknown variables, except those for fully supported edges. We collect all the Cartesian coordinate parameters of control points to unknown variable vector \mathbf{x} . In addition, we introduce g_{ij} and γ , which are metrics of leafs and branches. These differ from \bar{g}_{ij} and $\bar{\gamma}$ and the functions of \mathbf{x} . When we apply \bar{s}^{ij} and \bar{n}^{11} to second Piola-Kirchhoff stress tensors of a self-supporting surface, and with the input (flat) geometry as its reference configuration, the strain energies of leafs and branches take the following forms:

$$E_{leaf} = \frac{1}{2} \int_{\Omega_2} \bar{s}^{ij} g_{ij} \sqrt{\det \bar{g}_{ij}} d\theta^1 d\theta^2,$$

$$E_{branch} = \frac{1}{2} \int_{\Omega_1} \bar{n}^{11} \gamma \sqrt{\bar{\gamma}} dt.$$

Note that E_{leaf} yields the well-known cotangent formula when $\bar{s}^{ij} = \bar{g}^{ij}$ and the surface is represented by a triangulated mesh

surface. Note also that g_{ij} and γ are quadratic in \mathbf{x} and the remaining quantities are all constants; thus, those energies are quadratic in \mathbf{x} . Therefore, the total energy takes the following form:

$$E_{total} = \frac{1}{2} \mathbf{x}^T \mathbf{D} \mathbf{x},$$

where \mathbf{D} is a constant matrix. A solution of $\operatorname{argmax}_{\mathbf{x}} E_{total}$ gives a shape that is in self-equilibrium, and, when fully supported edges are constrained to the same height, it gives a shape in horizontal self-equilibrium. It is difficult to compute those integrals analytically; therefore, we compute the integrals approximately by taking summations of the integrands over the evaluation points, i.e., we assumed that the integrands have a constant value in each small square or line that each evaluation point governs. Because \bar{s}^{ij} and \bar{n}^{11} are supposed to be negative semi-definite, \mathbf{D} must be negative semi-definite theoretically. Thus, E_{total} can be maximized using a general optimization solver that is specialized in convex problems only (such as Mosek®). Another option is to use a general linear algebraic package such as Lapack, because the problem can be solved simply by a linear system of equations.

One important characteristics of E_{total} is that it is independent of x , y , or z directions. Thus, if one moves the fully supported edges to arbitrary heights, the planar projection of the equilibrium shape does not change. Similarly, if one applies a vertical loading to the self-supporting surface, each control point moves only along the vertical axis and its planar projection does not change. A vertical loading can be applied by solving the following:

$$\operatorname{argmax}_{\mathbf{x}} E_{total} + \mathbf{p}^T \mathbf{x}, \quad (13)$$

where \mathbf{p} is a collection of nodal forces that approximates the design loading. Methods to approximate a continuous loading by equivalent nodal forces can be found in various textbooks describing finite element methods; thus, we do not describe such methods. If \mathbf{p} only has components of the z coordinate, it represents a vertical loading. If one wants a precise vertical loading that approximates the self-weight of a structure, Equation (13) is solved iteratively, and \mathbf{p} is updated until the solution converges. However, this is also just a technical issue; thus, we only distribute a common vertical vector to every control point and perform a single computation.

The theory of Airy stress functions tells us that the planar projection (top view) of the solution of the equilibrium problem must be very close to the input geometry. This is not obvious because (x, y) coordinates of the control points are managed as free (movable) variables. Thus, x and y variables can be used for verification of the validity of the proposed method and for investigating a sufficient number of evaluation points to provide sufficient precision when solving equilibrium problems. The number of evaluation points indicated in Section 5.2 was determined as such.

7 Results

To investigate the effect of cutting planes, we performed a series of computations on the basis of the input geometry shown in Fig. 2(a). The results are shown in Fig. 7. Figures 7(a–c) show how the shapes of self-supporting surfaces change due to different cutting plane configurations. Note that their planar projections stay unchanged from the input geometry although horizontal movements of control points is allowed. In addition, Fig. 7(d) shows a computational result that was obtained with exactly the same configurations of the cutting planes shown in Fig. 7(a) but with larger kink angles (achieved by increasing lower bounds of kink angles). Note that subtle differences of Airy stress functions result in large differences in self-supporting surfaces.

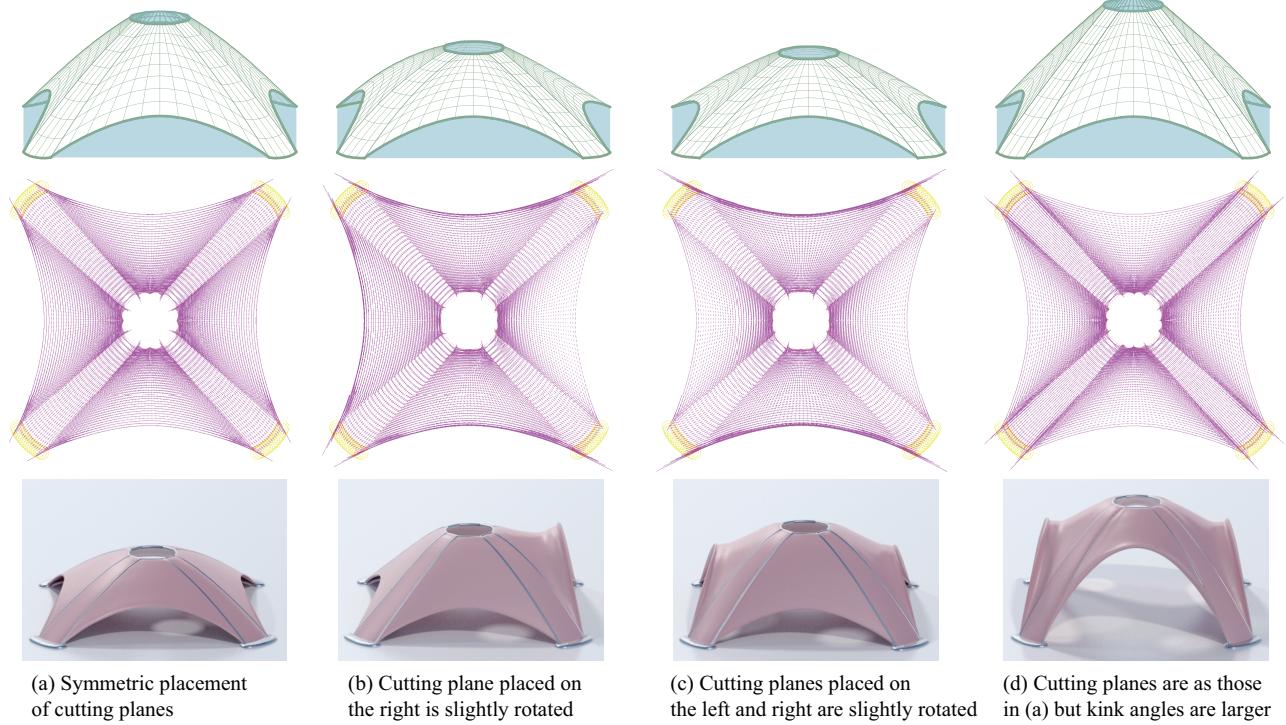


Figure 7: A variety of computational results based on the input geometry shown in Fig. 2 (a). Top: Airy stress functions. Middle: stress distributions generated from the Airy stress functions. Bottom: resultant self-supporting surfaces.

- Rib
- Open edge (strengthened/free)
- Fully supported edge
- Cutting plane
- Point set for the multiquadric function
- Target function represented by a multiquadric function

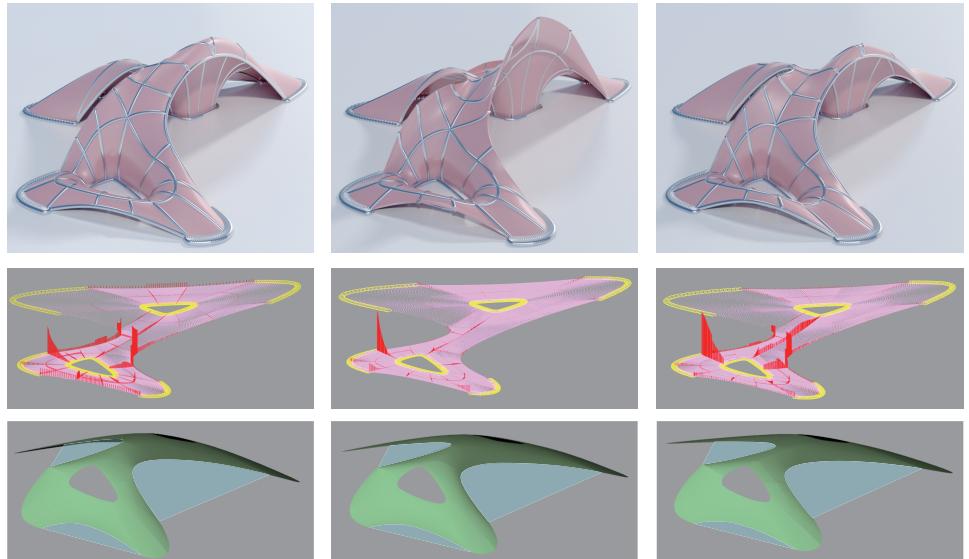
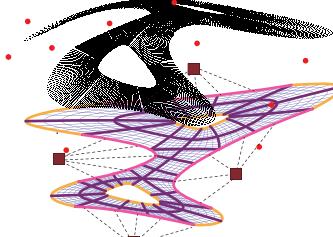


Figure 11: Numerical examples to demonstrate the effects of setting lower and upper bounds of kink angles and mean curvatures. Left: a reference result (first computation) in which only minimum necessary lower and upper bounds are set. Middle: upper bounds of kink angles for open edges are set to small values (e.g., 0.001) to approximately achieve free-open edges. Right: mean curvatures are set to smaller values so that a more “bumpy” shape is obtained.

Although it was possible to set the kink angles for free-open edges to precisely zero for all computational results shown in Fig. 2, in more complex cases such as those discussed in the following, we observed that it sometimes made computations of Airy stress functions fail. As an alternative, we made the upper bounds of kink angles controllable parameters and allow the user to manually lead them to nearly zero. In addition, the mean curvatures of NURBS

patches in Airy stress functions are managed as controllable parameters, as mentioned in Section 5.1. The curves and surfaces were grouped into several groups, and the user can specify the lower and upper bounds per group. This additional management of lower and upper bounds of those parameters makes it possible interactively control the shapes of self-supporting surfaces with a right-size design space.

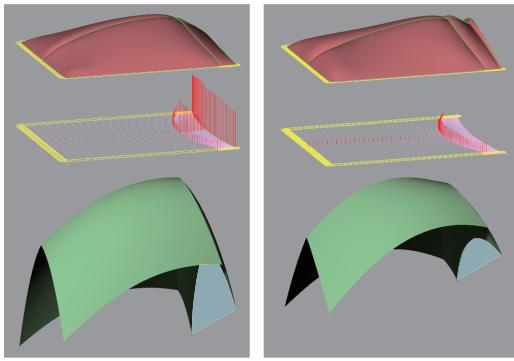


Figure 8: Computational results based on an input shape from [Panizzo et al. 2013, Fig. 4]. Top: self-supporting surfaces strengthened by ribs. Middle: stress fields. Bottom: Airy stress functions. The upper bounds of kink angles between the Airy stress function and a cutting plane is set to free (infinite) in the left. Although optimization fails when the upper bound is set to exactly zero, it is possible to approximately achieve a “free” open edge (right) by setting the upper bounds to a small value (e.g., 0.001). Note that the computational result shown on the right nicely reproduces the design intents addressed in [Panizzo et al. 2013, Fig. 4].

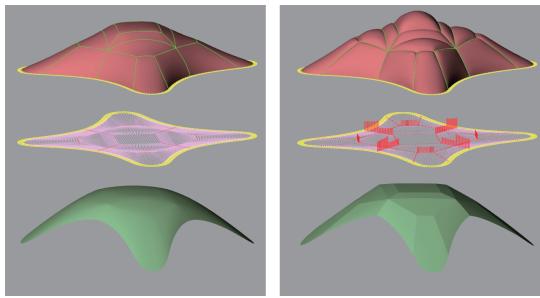


Figure 9: Computational results based on an input shape from [Vouga et al. 2012, Fig. 5]. It is possible to approximately achieve no-ribbed dome by setting upper bounds of kink angles to very small values (e.g., 0.001) (left). Strengthening effects by ribs are enhanced when setting the upper bounds of mean curvatures in Airy stress functions to very small values (e.g., 0.001) (right).

Figures 8, 9, and 10 show the computational results for input geometries that mimic those appearing in previous publications. As can be seen, further variations of shapes can be obtained by controlling the lower and upper bounds of kink angles and mean curvatures, which might be an advantage compared with the previous discrete methods. Although the placement of strengthening ribs are compulsory, one can approximately achieve a no-ribbed dome (Fig. 9 left and Fig. 10 left) or free-open edges (Fig. 8 right and Fig. 10 right) by setting the upper bounds of the kink angles to nearly zero. Note that we specified that target functions for the Airy stress functions in these examples.

Fig. 11 shows more free exploration of self-supporting surface designs. These different shapes were obtained by interactively controlling the lower and upper bounds of kink angles and mean curvatures. As such, although there is no direct control of shapes, dynamic shape manipulation is realized through these controllable parameters.

Fig. 1(a) shows another modular architectural plan for a self-supporting surface. In this example, we did not specify a target

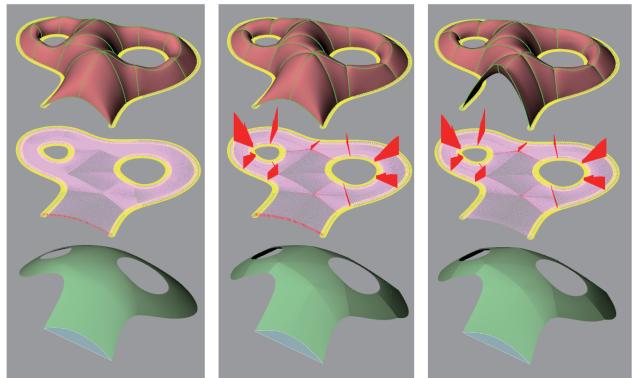


Figure 10: Computational results based on an input shape from [Vouga et al. 2012, Fig. 1]. It is possible to “indirectly” control the shape of self-supporting surfaces by changing the upper and lower bounds of kink angles and mean curvatures in the Airy stress function. A no-ribbed dome is approximately achieved by setting the upper bounds of kink angles for shared edges to nearly zero (left). Only the minimum necessary lower and upper bounds are set (middle). The upper bounds of kink angles for open edges are set to small values (e.g., 0.001) to approximately achieve free-open edges (right).

function for the Airy stress function but manually placed cutting planes at desired configurations. In addition, the lower and upper bounds of the kink angles and mean curvatures were set manually. We attempted to achieve exact free-open edges by constraining the kink angles for those edges to precisely zero. Optimization did not fail when we attempted to make two open edges free; however, it failed when we attempted to make over two open edges free. Therefore, we decided to allow only two open edges as free-open edges. As shown in Fig. 1(c), each fully supported edge was lifted to different heights. Note that the planar projection (top view) of Fig. 1(c) remains close to the input geometry (Fig. 1(a)) because x and y variables are independent of z variables. This example demonstrates the applicability of the proposed computational scheme to architectural design.

8 Conclusions

We have presented an optimization procedure that takes flat and watertight NURBS patches as inputs and gives an Airy stress function that satisfies all the conditions necessary for compression-only structures. The resultant Airy stress functions were also represented by NURBS. Ribs that naturally correspond to those seen in traditional vaulted masonry were added to the shared edges and arches were added to the open (i.e., unsupported boundary) edges. From the obtained Airy stress functions, a stress field can be computed, including axial forces in the ribs/arches, on the basis of the theories of continuum and discrete Airy stress functions. The obtained stress fields were relayed to the final stage, the equilibrium problems were formed and solved, and the shapes of the self-supporting surfaces were obtained. The resultant self-supporting surfaces were also represented by NURBS. We have shown that one can explore possible variations of self-supporting surfaces with the same input by controlling the lower and upper bounds of kink angles and mean curvatures in the Airy stress function. It was possible to approximately achieve a no-ribbed dome or free-open edges by setting the lower and upper bounds appropriately.

Acknowledgments

We thank all anonymous reviewers for their support in improving the quality of our work. This research was first initiated by the Igarashi Design Interface Project (Japan Science and Technology agency/ERATO). After the project was dissolved, the research was supported in part by an annual endowment from Asahi Glass Co., Ltd., (Japan).

References

- ARIS, R. 1962. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Dover, New York.
- ARNOLD, D. N., FALK, R. S., AND WINTHER, R. 2006. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica* 15 (May), 1.
- ARNOLD, D. N., FALK, R. S., AND WINTHER, R. 2010. Finite element exterior calculus: from Hodge theory to numerical stability. *Bulletin of the American Mathematical Society* 47, 2 (Jan.), 281–354.
- BAZILEVS, Y., CALO, V., COTTRELL, J., EVANS, J., HUGHES, T., LIPTON, S., SCOTT, M., AND SEDERBERG, T. 2010. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 199, 5–8 (Jan.), 229–263.
- BLOCK, P., AND OCHSENDORF, J. 2007. Thrust Network Analysis: A new methodology for three-dimensional equilibrium. *Journal of the International Association for Shell and Spatial Structures* 48, 3, 167–173.
- BUFFA, A., SANGALLI, G., AND VÁZQUEZ, R. 2014. Isogeometric methods for computational electromagnetics: B-spline and T-spline discretizations. *Journal of Computational Physics* 257 (Jan.), 1291–1320.
- CIARLET, P. G. 2006. *An Introduction to Differential Geometry with Applications to Elasticity*. Springer, New York.
- COTTRELL, J. A., HUGHES, T. J. R., AND BAZILEVS, Y. 2009. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons.
- DARLING, R. W. R. 1994. *Differential forms and connections*. Cambridge University Press.
- DE GOES, F., ALLIEZ, P., OWHADI, H., AND DESBRUN, M. 2013. On the Equilibrium of Simplicial Masonry Structures. *ACM Transactions on Graphics (TOG)* 32, 4.
- DE GOES, F., LIU, B., BUDNINSKIY, M., TONG, Y., AND DESBRUN, M. 2014. Discrete 2-Tensor Fields on Triangulations. *Computer Graphics Forum* 33, 5 (Aug.), 13–24.
- DÖRFEL, M. R., JÜTTLER, B., AND SIMEON, B. 2010. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering* 199, 5–8 (Jan.), 264–275.
- EISENHART, L. 1947. *An introduction to differential geometry: with use of the tensor calculus*. Princeton University Press.
- EISENHART, L. 1997. *Riemannian geometry*. Princeton University Press, London.
- FRATERNALI, F., ANGELILLO, M., AND FORTUNATO, A. 2002. A lumped stress method for plane elastic problems and the discrete-continuum approximation. *International Journal of Solids and Structures* 39, 25 (Dec.), 6211–6240.
- FRATERNALI, F., FARINA, I., AND CARPENTIERI, G. 2014. A discrete-to-continuum approach to the curvatures of membrane networks and parametric surfaces. *Mechanics Research Communications* 56, 18–25.
- FRATERNALI, F. 2010. A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions. *Mechanics Research Communications* 37, 2, 198–204.
- GURTIN, M. E. 1963. A generalization of the Beltrami stress functions in continuum mechanics. *Archive for Rational Mechanics and Analysis* 13, 1 (Dec.), 321–329.
- HARDY, R. L. 1971. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research* 76, 8 (Mar.), 1905–1915.
- HUGHES, T., COTTRELL, J., AND BAZILEVS, Y. 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194, 39–41 (Oct.), 4135–4195.
- JOHANNESSEN, K. A., KVAMSDAL, T., AND DOKKEN, T. 2014. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering* 269 (Feb.), 471–514.
- LI, X., DENG, J., AND CHEN, F. 2007. Surface modeling with polynomial splines over hierarchical T-meshes. *The Visual Computer* 23, 12 (Aug.), 1027–1033.
- LIU, Y., PAN, H., SNYDER, J., WANG, W., AND GUO, B. 2013. Computing self-supporting surfaces by regular triangulation. *ACM Transactions on Graphics (TOG)* 32, 4.
- PANOZZO, D., BLOCK, P., AND SORKINE-HORNUNG, O. 2013. Designing unreinforced masonry models. *ACM Transactions on Graphics (TOG)* 32, 4 (July).
- SCOTT, M., SIMPSON, R., EVANS, J., LIPTON, S., BORDAS, S., HUGHES, T., AND SEDERBERG, T. 2013. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 254 (Feb.), 197–221.
- SEDERBERG, T. W., ZHENG, J., BAKENOV, A., AND NASRI, A. 2003. T-splines and T-NURCCs. *ACM Transactions on Graphics* 22, 3 (July), 477.
- SEDERBERG, T. W., FINNIGAN, G. T., LI, X., LIN, H., AND IPSON, H. 2008. Watertight trimmed NURBS. In *ACM SIGGRAPH 2008 papers on - SIGGRAPH '08*, ACM Press, New York, New York, USA, vol. 27, 1.
- SIMPSON, R., BORDAS, S., TREVELYAN, J., AND RABCZUK, T. 2012. A two-dimensional Isogeometric Boundary Element Method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering* 209–212 (Feb.), 87–100.
- TANG, C., SUN, X., GOMES, A., WALLNER, J., AND POTTMANN, H. 2014. Form-finding with polyhedral meshes made simple. *ACM Transactions on Graphics (TOG)* 33, 4 (July).
- VOUGA, E., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2012. Design of self-supporting surfaces. *ACM Transactions on Graphics (TOG)* 31, 4.
- WILLIAMS, C. 1990. The Generation of a Class of Structural Forms for Vaults and Sails. *The Structural Engineer* 68, 12, 231–235.

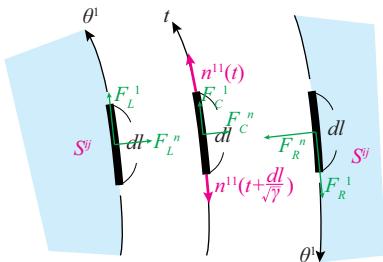


Figure 12: Horizontal force equilibrium of a rib (top view).

A Horizontal Equilibrium of a Rib

When a rib is added between two NURBS patches, stress in the two patches and stress (i.e., axial force) in the rib must satisfy horizontal equilibrium conditions. Here, we decompose those two patches and the rib to examine a tiny area whose length is $dl = \sqrt{\gamma}dt$ (Fig. 12). In this area, we observe six components of external forces, i.e., F_L^1 , F_L^n , F_C^1 , F_C^n , F_R^1 and F_R^n , that balance with those stresses and axial forces. Superscripts 1 and n represent forces parallel and perpendicular to the rib respectively. Subscripts L , C and R represent forces that balance with the stress of the left patch, the rib, and the right patch, respectively.

When there is no rib, the equilibrium conditions are simple:

$$F_L^1 - F_R^1 = 0 \text{ and } F_L^n - F_R^n = 0.$$

Note that F_L^1 or F_R^1 is a mixture of s^{21} and s^{22} . On the other hand, F_L^n or F_R^n is proportional to s^{22} . When these two conditions are not satisfied, the residual forces $R^1 = F_L^1 - F_R^1$ and $R^n = F_L^n - F_R^n$ must be canceled by another structural component. Thus, we add a rib.

When a rib is added, the axial force of the rib (denoted n^{11} in the main text) must satisfy $F_C^1 + R^1 = 0$ and $F_C^n + R^n = 0$; F_C^1 , where F_C^n is proportional to both n^{11} and the curvature of the horizontal projection of the rib. On the other hand, F_C^1 is given by a change (i.e., divergence) of n^{11} between both ends of this tiny element. This means that if the horizontal projection of the rib is straight, F_L^1 and F_R^1 should match. Even in this case, F_L^1 and F_R^1 may not match and the difference between them might be canceled by the divergence of n^{11} (i.e., F_C^1). If a nice stress distribution for n^{11} is found, R^1 and R^n are canceled by n^{11} , and the structure will be horizontally self-equilibrated.

B Coefficient Choice

Fig. 13 shows a comparison of the coefficient choices between kink angles D and axial forces n^{11} . These are shapes in horizontal equilibrium that were computed with stress field generated from the same Airy stress function shown in Fig. 7(a) but with different coefficient choices. Note that Fig. 13(c) matches the planar projections of Fig. 7(a), i.e., the input geometry shown in Fig. 2(a). This coefficient was identified through trial and error.

C Gallery

Fig. 14 shows other variations of self-supporting surfaces computed by the proposed computational scheme.

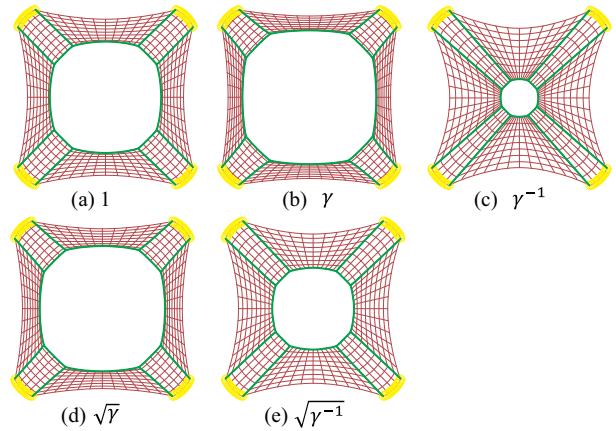


Figure 13: Comparison of different coefficients between kink angles and axial forces. Note that (c) is very close to Fig. 2(a).



Figure 14: Variations of self-supporting surfaces.