

**Interactive Equilibrium
Modelling**

by

Lorenz Lachauer and Philippe Block

Reprinted from

**INTERNATIONAL JOURNAL OF
SPACE STRUCTURES**
Volume 29 · Number 1 · 2014

MULTI-SCIENCE PUBLISHING CO. LTD.
5 Wates Way, Brentwood, Essex CM15 9TB, United Kingdom

Interactive Equilibrium Modelling

Lorenz Lachauer^{1,*} and Philippe Block²

¹Research Assistant, Institute of Technology in Architecture, ETH Zurich, Wolfgang-Pauli-Strasse 15, 8093 Zurich, Switzerland. lachauer@arch.ethz.ch

²Assistant Professor, Institute of Technology in Architecture, ETH Zurich, Wolfgang-Pauli-Strasse 15, 8093 Zurich, Switzerland. block@arch.ethz.ch

(Submitted on 14/06/2013, Reception of revised paper 29/11/2013, Accepted on 13/01/2014)

ABSTRACT: This paper presents a novel method for computer-aided equilibrium modelling of structures in early design stages. Based on the force density method, an iterative procedure is developed that enables the generation of spatial kinematic pin-jointed structures that are in equilibrium close to a given input geometry, while satisfying additional constraints on both geometry and forces. This method forms the core of an interactive form-finding process that consists of alternating steps of modelling and computational optimization. In each modelling step, the user is able to modify geometry, topology, external forces and constraints of the structure. In each optimization step, equilibrium is re-established while respecting the user-defined constraints. A prototype has been implemented within an existing CAD software package, and three examples illustrate the use of the presented method, ranging from a playful exploration of surprising shapes to the rationalization of structural geometry. The method allows to intuitively explore the formal freedom of spatial equilibrium shapes with mixed compression and tension forces, within hard, user-defined constraints. In conclusion, it is claimed that by providing interactive equilibrium modelling methods, the design of new, surprising spatial forms with efficient structural behaviour is facilitated.

Key Words: computer-aided design exploration, constrained form finding, force density method, strut-and-tie models, real-time structural design tools

1. INTRODUCTION

Architectural freeform design and construction, based on powerful CAD software, has a tradition of more than 15 years: the icon of freeform design, the Guggenheim museum in Bilbao, Spain by Frank Gehry, opened in 1997. Since then, one of the main research objectives in the field of computational design has been focused on bridging the gap between formal design freedom and hard technical constraints. Such constraints mainly arise from limitations in construction, fabrication and structural behaviour. Strong advances have been made in the field of geometry rationalization and fabrication optimization [1], while the concepts for the integration of structural constraints remain fragmented. Recently, various

approaches in the conceptual framework of *morphogenesis* have been developed, combining parametric models with FEM analysis modules and genetic algorithms [2], or combining FEM with gradient-based optimization [3]. In this paper, a novel computational design method, rooted in the tradition of form finding using pin-jointed truss models, is presented.

Historically, pin-jointed truss models have played an important role in design processes of technically and aesthetically innovative buildings. Great structural designers used graphic statics as well as physical models for the generation equilibrium shapes. Graphic statics has been used by, for example, Gustave Eiffel, Antoni Gaudí and Robert Maillart [4]. Form finding

*Corresponding author e-mail: lachauer@arch.ethz.ch

based on physical models has been applied by, for example, Christopher Wren, Antoni Gaudí, Frei Otto and Heinz Isler [5]. Today, those methods are still used occasionally in early design stages; for example, Jörg Conzett used graphic statics for the design of the second *Traversiner Steg* in Grisons, Switzerland [6], and Laurent Ney used hanging models in the design of a curved suspension bridge in Kortrijk, Belgium [7].

Since the early 1970s, various computational methods have been developed for form finding of pure compression or pure tension structures; the *force density method* and *dynamic relaxation*, together with their non-linear extensions, constitute the fundamental approaches [8]. *Thrust Network Analysis* combines concepts of graphic statics with approaches from the force density method for the analysis and design of unreinforced masonry structures [9]. Recently, the force density method has been extended for the form finding of mixed compression and tension structures [10]. Kemmler mentions the *statically-geometrically coupled method* for general form-finding purposes, which is internally used in the office Schlaich, Bergermann & Partner, without describing its technical details. A spatial arch bridge and cable-stayed stadium roof are presented as applications of the method [11].

Through the availability of powerful interactive modelling environments, various computational form-finding techniques have been implemented as modelling methods, addressing problems of real-time user interaction: Kilian proposed a structural modelling system based on particle spring systems [12], Rippmann et al. presented a design method for compression vaults based on Thrust Network Analysis [13], and Tachi proposed a design process for tensegrity structures derived from the force density method [14].

This paper presents a novel computer-aided modelling approach for spatial pin-jointed structures with mixed compression and tension forces. Furthermore, the method allows to directly modify the equilibrium form, and to impose constraints on both inner forces and geometry during the design process. The paper is structured as follows: Section 2 summarizes the concepts of graphic statics, the relevance of pin-jointed models for design and construction, and the basics of the force density method; Section 3 describes the core form-finding method of the proposed design process in full detail; Section 4 illustrates a prototypical implementation of the method within the CAD software package Rhinoceros [15]; and, finally, Section 5 presents three applications of the method.

2. BACKGROUND

Graphic statics is an important historic reference of an intuitive approach to structural design for pin-jointed or truss models, with direct control of forces and geometry. Strut-and-tie models represent in an exemplary fashion the scientific link between truss model and construction detail. The force density method and its extensions are the foundation of the non-linear computational form-finding method presented here.

2.1. Graphic statics

Graphic statics has been developed in the mid-19th century as method for structural analysis and design by Culmann [16]. Based on the concepts of vector equilibrium and reciprocal diagrams, the method enables to graphically calculate the inner forces of a planar truss structures by the construction of *force diagrams*. Furthermore, the method allows to find efficient structural forms using a *funicular polygon*, considering constraints on forces and geometry. The free nodes of the funicular are constrained to the *lines of action* defined by the direction and position of the externally applied load vectors; the force diagram allows to directly control the inner forces. Recently, different approaches have been developed to extend the design capabilities of graphic statics to spatial systems. For example, Laffranchi [17] proposed a method for the determination of spatial funicular polygons for the design of curved bridges by introducing planes as constraints for the free nodes, Lachauer [18] combined graphic techniques with dynamic relaxation for spatial bridge design. Block [9] extended the concept of the funicular polygon to funicular surface networks with fixed projection. As in graphic statics, the free nodes are constrained to the lines of action of the load vectors, the forces are controlled directly through force diagrams.

2.2. Strut-and-tie modelling

Funicular polygons have wide applications in the early design of efficient, bending-free structures, beyond the typology of tension structures. Truss models often constitute a close link between form and construction detail. Based on the conceptual framework of plasticity theory, truss models are applied as *strut-and-tie models* [19] or *stress-fields* [20] in the design and constructive detailing of reinforced concrete. Schlaich presented the detailed design of a cast steel component for the anchorage of a bridge deck in concrete based on a truss model [19]. For masonry structures, *thrust lines* and *thrust networks* allow for the design of the stone geometry of masonry vaults [9].

2.3. Force density method

The force density method has been developed in the early 1970s by Linkwitz and Schek [21]. In its basic form, as described in Section 2 of [22], the force density method allows the form finding of equilibrium networks with defined topology for a given set of loads and support points. The key concept of the method is the replacement of inner forces and lengths of branches by force-length ratios, called *force densities*, in order to obtain a linear system of equations with a unique solution for the unknown coordinates of the free nodes.

For a given network with m branches and n_s nodes, the connectivity of the branches is described by a *branch-node* matrix C_s with dimensions $m \times n_s$. If branch K connects nodes i and j , then the element (k, i) of C_s equals +1, and element (k, j) -1; all other elements are 0. The columns of the branch-node matrix are ordered such that one can define two submatrices of $C_s = [C \ C_f]$, C corresponding to the free nodes, and C_f corresponding to the support nodes (for an illustration of C_s , see [22]). Each free node is interpreted as point i with coordinates (x_i, y_i, z_i) ; the coordinates of the fixed support points are (x_{fi}, y_{fi}, z_{fi}) . These coordinates constitute the free coordinate vectors x, y, z of length n and the fixed coordinate vectors x_f, y_f, z_f of length n_f , hence $n_s = n + n_f$.

The force densities are defined as the force-length ratios $q_i = \frac{f_i}{l_i}$, forming the vector \mathbf{q} of length m . For a set of force densities \mathbf{q} and nodal loads $\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z$, the coordinates x, y, z of the free nodes are uniquely defined and obtained by solving one system of linear equations.

Linkwitz and Schek furthermore describe several ways to extend the force density method using iterative procedures in order to incorporate additional constraints, such as defined branch lengths or force values [21–23]. In [23], two conceptually different, iterative procedures for the optimization of given network geometries are distinguished: the “Newtonverfahren” (“Newton approach”) and the “Ausgleichungsansatz” (“Variational approach”).

In the “Newton approach”, the parameter space is constituted by the coordinates of the free nodes; the force density method is used to compute the network geometry at iteration $t + 1$ based on its previous state at iteration t :

$$(\Delta x, \Delta y, \Delta z) = g_N(x^t, y^t, z^t) \quad (1)$$

$$(x^{t+1}, y^{t+1}, z^{t+1}) = (x^t + \Delta x, y^t + \Delta y, z^t + \Delta z) \quad (2)$$

In the “variational approach”, the parameter space is constituted by the force densities and the free node

coordinates; both force densities and network geometry at iteration $t + 1$ are calculated based on their previous states at iteration t :

$$(\Delta \mathbf{q}, \Delta x, \Delta y, \Delta z) = g_V(\mathbf{q}^t, x^t, y^t, z^t) \quad (3)$$

$$(\mathbf{q}^{t+1}, x^{t+1}, y^{t+1}, z^{t+1}) = (\mathbf{q}^t + \Delta \mathbf{q}, x^t + \Delta x, y^t + \Delta y, z^t + \Delta z) \quad (4)$$

Both methods work well for the optimization of network geometries that are close to an equilibrium state purely in compression or tension. In the next section, the latter approach is adopted for design purposes, allowing to find equilibrium networks close to a given input geometry that consist of a combination of compression and tension branches, while imposing additional constraints on form and forces.

3. METHOD

The presented method is the technical core of an interactive form-finding approach for pin-jointed structures within a CAD environment. The design process consists of alternating steps of user interaction with the model and computational optimization (Fig. 1). In each modelling step, the user is able to modify network topology and geometry, and external forces, or to apply constraints on force densities and the free nodes’ coordinates. After this, the structure is generally no longer in equilibrium, or, alternatively, the imposed constraints are not satisfied. The computational optimization method iteratively minimizes residual forces through redistribution of inner forces and through changes of the geometry, with respect to the user-defined constraints. In general, the optimization method converges to an equilibrium state (all residual forces are below the chosen tolerance) if the user-defined constraints are feasible. This alternating process of modelling and optimization is repeated until the equilibrium state of the model satisfies external constraints, such as for instance functionality or spatial and aesthetic qualities, at which stage the form-finding process terminates.

Inspired by the concepts of graphic statics, constraints can be imposed both on force densities (hence implicitly on the forces) as on the nodes’ coordinates. Force density constraints are defined as lower and upper bounds \mathbf{q}^{LB} and \mathbf{q}^{UB} . For a set of constrained force densities $\hat{\mathbf{q}}$, the condition $\mathbf{q}^{LB} \leq \hat{\mathbf{q}} \leq \mathbf{q}^{UB}$ is true. Each node that is not defined as support node, can either be geometrically free, constrained to a line or a plane, or fixed in 3D space.

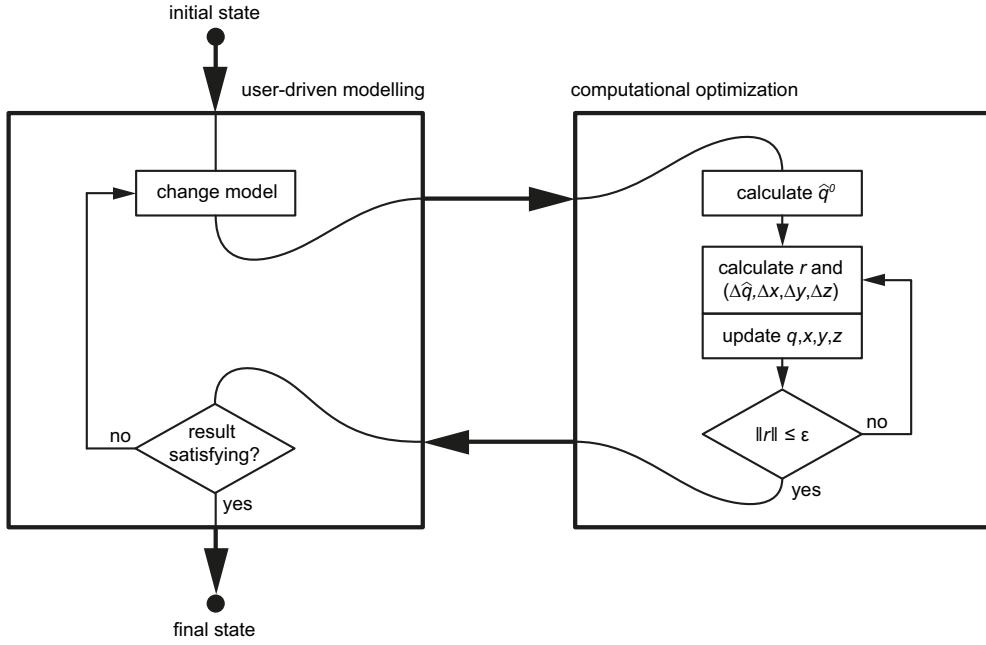


Figure 1. Flow chart of the form-finding process consisting of alternating user-driven modelling steps and computational optimization steps that re-establish equilibrium.

For a given threshold value ϵ , the computational optimization process works as follows:

1. Calculate a set of constrained force densities $\hat{\mathbf{q}}^0$ from the initial network geometry $\mathbf{x}^0, \mathbf{y}^0, \mathbf{z}^0$ (Section 3.1);
2. Calculate the nodal residuals \mathbf{r} and the gradient $(\Delta\hat{\mathbf{q}}, \Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{z})$ with respect to the constraints (Sections 3.2 and 3.3);
3. Update the force densities \mathbf{q} and network geometry $\mathbf{x}, \mathbf{y}, \mathbf{z}$; and
4. If, $\|\mathbf{r}\| > \epsilon$, return to step II), and if $\|\mathbf{r}\| \leq \epsilon$, the algorithm terminates

A necessary condition is that the model forms a mechanism, and the support nodes can take reaction in x-, y- and z- direction, which, after Timoshenko [24], is the case if the following inequality is true:

$$\mathbf{m} < 3\mathbf{n}. \quad (5)$$

3.1. Initial step

First, the coordinate differences $\mathbf{u}, \mathbf{v}, \mathbf{w}$ per branch are calculated:

$$\begin{aligned} \mathbf{u} &= \mathbf{Cx} + \mathbf{C}_f \mathbf{x}_f \\ \mathbf{v} &= \mathbf{Cy} + \mathbf{C}_f \mathbf{y}_f \\ \mathbf{w} &= \mathbf{Cz} + \mathbf{C}_f \mathbf{z}_f \end{aligned} \quad (6)$$

Based on the vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$, their diagonal matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ are constructed. This allows to define the equilibrium matrix \mathbf{A} with dimensions $3n \times 3m$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}^T & \mathbf{U} \\ \mathbf{C}^T & \mathbf{V} \\ \mathbf{C}^T & \mathbf{W} \end{bmatrix}. \quad (7)$$

Together with the vertically stacked load vector

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \end{bmatrix}, \quad (8)$$

the vector of nodal residuals \mathbf{r} for a given set of m forceddensities \mathbf{q} is then formulated as

$$\mathbf{r} = \mathbf{Aq} - \mathbf{p}. \quad (9)$$

The system of equations (9) has m unknowns and $3n$ equations, and because of (5), it is thus overdetermined. The goal is to find a set of initial force densities \mathbf{q}^0 that minimizes the initial nodal residuals \mathbf{r}^0 for the given initial geometry, which can be formulated as a linear least square problem:

$$\mathbf{q}^0 = \min \|\mathbf{r}^0\|^2 = \min \|\mathbf{Aq}^0 - \mathbf{p}\|^2, \quad (10)$$

for which the solution can be written analytically using normal equations:

$$\mathbf{q}^0 = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{p}. \quad (11)$$

3.2. Constraints

Inspired by an approach for geometric modelling developed by Bouaziz et al. [25], constraints are implemented using *projections*. In the first step of the iterative method, the constraints on the force densities, given as lower and upper boundaries \mathbf{q}^{LB} and \mathbf{q}^{UB} , are imposed on \mathbf{q}^0 by projecting them on the boundaries of the constrained set, if at least one constraint is violated (Fig. 2). Analytically, the constrained set of initial force densities $\hat{\mathbf{q}}^0$ is defined as:

$$\hat{\mathbf{q}}_i^0 = \begin{cases} q_i^{LB}, & \text{if } q_i < q_i^{LB} \\ q_i, & \text{if } q_i^{LB} \leq q_i \leq q_i^{UB} \\ q_i^{UB}, & \text{if } q_i^{UB} < q_i \end{cases} \quad (12)$$

In each iteration step, the residual vector \mathbf{r} is computed from $\mathbf{q}, \mathbf{x}, \mathbf{y}, \mathbf{z}$, i.e. Equations (6–9), and \mathbf{r} is decomposed element-wise in “unconstrained” components \mathbf{r}^U and “constrained” components \mathbf{r}^C , such that $\mathbf{r} = \mathbf{r}^U + \mathbf{r}^C$.

The idea is to resolve the “free” components \mathbf{r}^U by changing the geometry according to $\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{z}$, and to resolve the “constrained” components \mathbf{r}^C by altering the force densities according to $\Delta\mathbf{q}$. In this section, the decomposition $\mathbf{r} = \mathbf{r}^U + \mathbf{r}^C$ is explained. In Section 3.3, the computation of the gradient ($\Delta\hat{\mathbf{q}}, \Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{z}$) from \mathbf{r}^U and \mathbf{r}^C is shown.

Each node that is not a support node, can either be free, constrained to a line, constrained to a plane, or fixed in space. Initially, all geometric constraints have

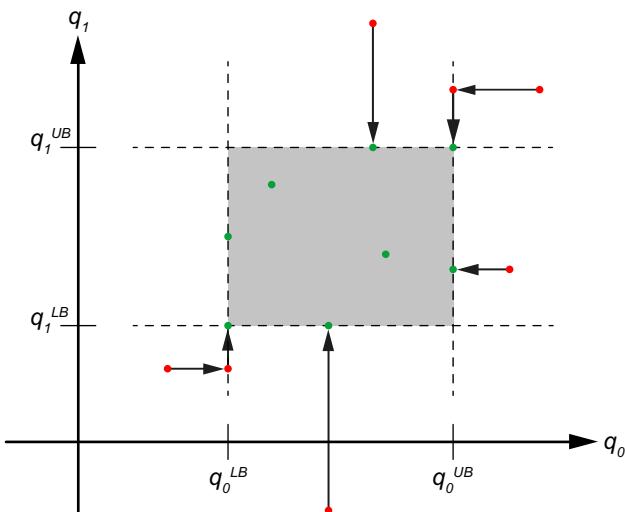


Figure 2. The diagram shows the imposing of the lower and upper bounds \mathbf{q}^{LB} and \mathbf{q}^{UB} on the force densities \mathbf{q} for a two-dimensional case. The red dots represent states of \mathbf{q} that are not within the bounds, and their projection bounded states $\hat{\mathbf{q}}$ represented as green dots.

to be satisfied by the model, thus a node that is constrained to a plane has to lie on this plane before the optimization process starts. For a given node i , depending on its degree of constraint, the corresponding residual \mathbf{r}_i is decomposed as follows:

- If node i is a free node, a global weighting factor $0 > \alpha > 1$ is introduced to balance change of geometry and change of inner forces, so $\mathbf{r}_i^U = \alpha \mathbf{r}_i$, hence $\mathbf{r}_i^C = (1 - \alpha) \mathbf{r}_i$ (Fig. 3a).
- If node i is constrained to a line Γ , \mathbf{r}_i^U is the orthogonal projection of the residual \mathbf{r}_i onto the line. Therefore, the projection matrix \mathbf{P}^Γ is formed based on the unit vector \mathbf{d} of the line’s direction: $\mathbf{P}^\Gamma = \mathbf{d}\mathbf{d}^T$. Then, $\mathbf{r}_i^U = \mathbf{P}^\Gamma \mathbf{r}_i$, and $\mathbf{r}_i^C = \mathbf{r}_i - \mathbf{P}^\Gamma \mathbf{r}_i$ (Fig. 3b).
- If node i is constrained to a plane Ψ , \mathbf{r}_i^U is obtained by an orthogonal projection of the residual \mathbf{r}_i onto the plane. Therefore, the projection matrix \mathbf{P}^Ψ is formed based on an orthonormal basis \mathbf{d}_1 and \mathbf{d}_2 (two orthogonal unit vectors lying within the plane Ψ): $\mathbf{P}^\Psi = [\mathbf{d}_1 \mathbf{d}_2][\mathbf{d}_1 \mathbf{d}_2]^T$. Then, $\mathbf{r}_i^U = \mathbf{P}^\Psi \mathbf{r}_i$, and $\mathbf{r}_i^C = \mathbf{r}_i - \mathbf{P}^\Psi \mathbf{r}_i$ (Fig. 3c).
- If node i is defined as a fixed node, the node should not move at all during the form-finding process, so the “free” component vanishes, $\mathbf{r}_i^U = 0$, hence $\mathbf{r}_i^C = \mathbf{r}_i$ (Fig. 3d).

3.3. Gradient

Based on the decomposition $\mathbf{r} = \mathbf{r}^U + \mathbf{r}^C$ described in the previous section, the computation of the constraint gradient ($\Delta\hat{\mathbf{q}}, \Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{z}$) is straightforward. The free components \mathbf{r}^U are used to compute “allowed” changes of the nodal positions within the geometric constraints using a stiffness \mathbf{k} :

$$\begin{aligned} \Delta\mathbf{x} &= \mathbf{k} \mathbf{r}_x^U \\ \Delta\mathbf{y} &= \mathbf{k} \mathbf{r}_y^U \\ \Delta\mathbf{z} &= \mathbf{k} \mathbf{r}_z^U \end{aligned} \quad (13)$$

Veenendaal and Block [8] have summarized a variety of different definitions of stiffness matrices used in form finding. Since this method is not counting on any material properties, only “geometric” stiffness is taken into account. In order to reduce the computational cost, the stiffness \mathbf{k} is implemented as a lumped geometric stiffness vector, similar as proposed by Barnes [26]. Here, the following definition for \mathbf{k} as unitized, lumped geometric stiffness is introduced:

$$\mathbf{k} = \frac{|\mathbf{C}^T \mathbf{L} \mathbf{q}|}{|\mathbf{C}^T \mathbf{L} \mathbf{q}|}, \quad (14)$$

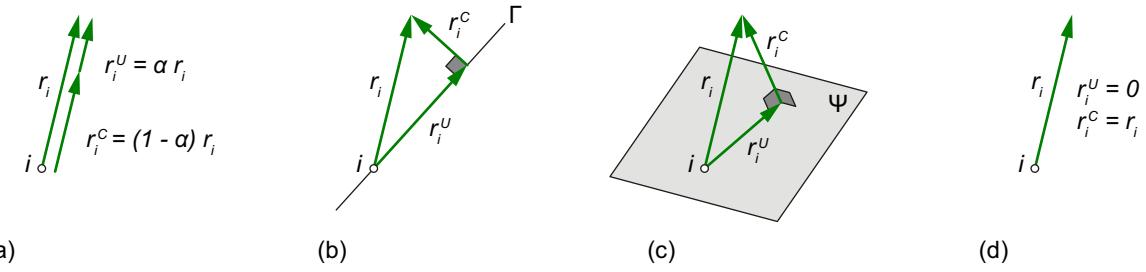


Figure 3. Constraint decomposition of residual \mathbf{r}_i if the node i is: (a) free node, (b) constrained to a line Γ , (c) constrained to a plane Ψ , and (d) fixed.

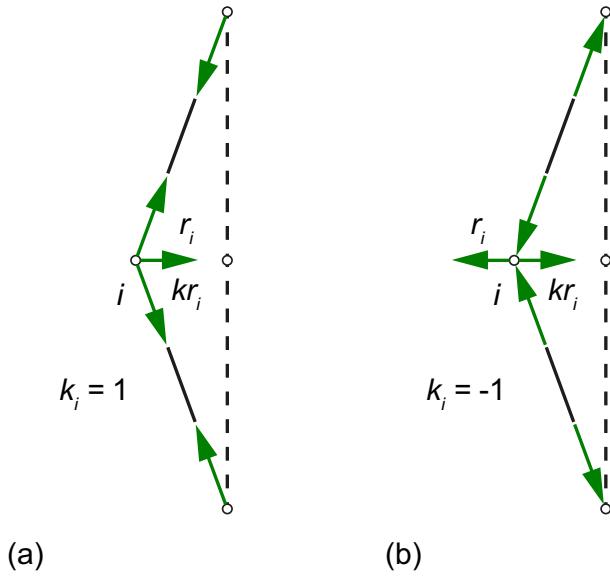


Figure 4. If node i is: (a) “tension-dominant” or (b) “compression-dominant”, residual \mathbf{r}_i is pointing towards, respectively away from, a close equilibrium state. In the latter case, the residual is flipped by the multiplication with -1 .

with element k_i having a value of $+1$ if the sum of forces in the adjacent branches of node i is greater than 0 (“tension-dominant” node), and -1 if the sum of forces in the adjacent branches of node i is smaller than 0 (“compression-dominant” node). Figure 4 illustrates the idea behind this definition for an unloaded node. For tension-dominant nodes, the residual points towards a close equilibrium position; for compression-dominant nodes, the residual points away from a close equilibrium position, so in this case the direction of the residual is flipped by the multiplication with -1 .

The matrix \mathbf{L} is constructed from the branch lengths along its diagonal, and is calculated as

$$\mathbf{L} = \sqrt{\mathbf{U}^T \mathbf{U} + \mathbf{V}^T \mathbf{V} + \mathbf{W}^T \mathbf{W}}. \quad (15)$$

The remaining “constrained” components \mathbf{r}^C of the residual vector \mathbf{r} are resolved with $\Delta\mathbf{q}$, for which the following system of equations is formulated:

$$\mathbf{r}^C = \mathbf{A}\Delta\mathbf{q}. \quad (16)$$

Since \mathbf{A} has dimensions $3n \times m$, and assumption (5) is true, the system of equations (16) is overdetermined, so $\Delta\mathbf{q}$ is computed as linear least-square approximation:

$$\Delta\mathbf{q} = \min \| \mathbf{A}\Delta\mathbf{q} - \mathbf{r}^C \|^2, \quad (17)$$

whose solution can be written analytically using normal equations:

$$\Delta\mathbf{q} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{r}^C. \quad (18)$$

In order to impose the force density bounds \mathbf{q}^{LB} and \mathbf{q}^{UB} during the iterative process, the bounded force density differences $\hat{\Delta}\mathbf{q}$ are calculated as follows:

$$\hat{\Delta}\mathbf{q}_i = \begin{cases} q_i^{LB} - q_i, & \text{if } q_i < q_i^{LB} \\ \Delta\mathbf{q}_i, & \text{if } q_i^{LB} \leq q_i \leq q_i^{UB} \\ q_i^{UB} - q_i, & \text{if } q_i^{UB} < q_i \end{cases} \quad (19)$$

3.4. Implementation

To set up an interactive computer-aided design process, a prototype has been implemented within the CAD software Rhinoceros [15], which is a tool, specifically developed for 3D freeform design. The build-in scripting language *IronPython* [27] with the additional open-source library *NumPy* [28] has been used for the linear algebra computations. Linear least square problems, as equations (10) and (17), have been

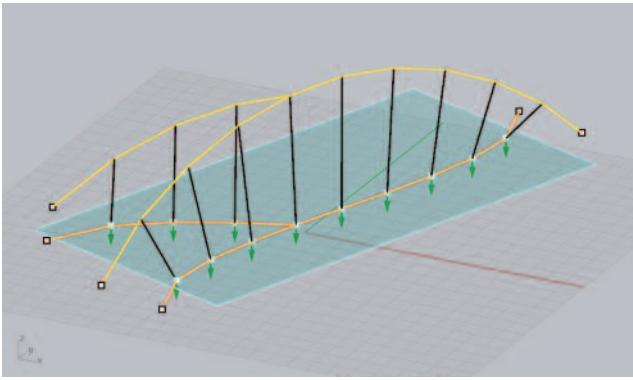


Figure 5. Screenshot of the CAD model used for the example presented in Section 4.1. Supports are modelled with white points, geometrically fixed nodes with points in cyan; planar constraints are encoded by points and planes with the same colour; force density constraints are imposed on the coloured branches.

solved with the function `numpy.linalg.lstsq`. The orthonormal basis of planes representing node constraints is obtained by the scripting command `rhinoscriptsyntax.SurfaceFrame`. For solving the iterative process, an explicit Runge–Kutta method with adaptive step size has been used, as described by Kiusalaas [29].

A pin-jointed model with applied external forces is represented by a set of lines and points in the 3D drawing space (Fig. 5), constraining objects such as lines and planar surfaces are modelled on a separate layer. Geometric constraints on the nodes are modelled with points with colours corresponding to their constraining objects (e.g. the points coloured in cyan are constrained to the cyan plan in Figure 5). Force density constraints are modelled with coloured branches; a table relates branch colours to the lower and upper boundaries q^{LB} and q^{UB} .

4. DESIGN EXAMPLES

For all examples presented below, the threshold value is defined as $\epsilon = r^0/100$, and the global weighting factor α is set to 0.5. In Table 1, the legend of the graphic symbols and conventions used in Figures 6–8 is given.

4.1. A bifurcating arch bridge

The first example illustrates a playful design exploration for a bifurcating arch bridge (Fig. 6). The process starts with a very rough approximation of a spatial arch bridge, inputted as a straight deck and a polygonal arch, connected to the deck with hangers. The deck's nodes are constrained to a horizontal plane; the bridge's dead load is represented by a set of

vertical unit forces attached to the nodes. The force densities of deck and arch branches (coloured in magenta) are constrained such that the deck is force to be in tension and the arch in compression (Fig. 6a).

The first optimization step generates a model in equilibrium with an almost planar arch and S-shaped deck (Fig. 6b). Through the imposing of an upper force density bound of -2.5, the structural height of the arch is limited. A higher bound, for example, -1, would result in smaller compression forces and thus in an arch with higher rise. A spatial configuration is enforced, by assigning a lower bound to the tension force in the deck; otherwise a planar arch with vertical hangers and no forces in the deck would emerge.

In the next step, shown in Figure 6c, the designer copies and rotates the first four segments of the deck and the arch, using standard CAD modelling tools to create the intended bifurcation of the bridge. In order to heighten the rise of the structure, the upper bound of the arch's force densities is raised to -1. The second optimization step generates the final equilibrium form of the bridge (Fig. 6d).

4.2. A curved suspension bridge

The example in Subsection 4.1 has been a playful, not very controlled design exploration. For example, the nodes of the bridge deck are no longer equally spaced in the final state (Fig. 6d). To obtain geometrically controlled equilibrium models, additional constraints have to be imposed. The example in this section demonstrates highly constrained form finding. It is inspired by Laffranchi [16], who states that any loaded, spatial polygon in space can be balanced by two funiculars.

The design process starts with a freeform curve, defining the intended pathway of the bridge (Fig. 7a). This curve is subsequently discretized as polygon, forces are added as dead load, and constructive axes are introduced (dashed lines), defining the intended layout of the structure in plan (Fig. 7b). Two different configuration as starting point of the form-finding process are explored (Fig. 7c and 7e).

In order to preserve the geometry of the intended pathway and the structural axes, the position of the deck nodes are fixed, and the nodes of the two funiculars are constrained to planes that are vertically extruded from the axes in both cases. Additionally, positive force density constraints are imposed on the part of the structure that is supposed to act as tension cable. In the first case (Fig. 7c), the form-finding process generates an equilibrium model with the deck in compression, horizontally balanced by a

Table 1. Legend of the graphic symbols and conventions used in Figures 6–8

Input model with constraints and external loads	Resulting constrained model in equilibrium
branch with force density bounds	compression force, line weight proportional to the square root of the force
branch without force density bounds	tension force, line weight proportional to the square root of the force
external force	external force
support node	support node
unsupported node constrained to position	
unsupported node constrained to plane	
constraining plane	geometry of the input model

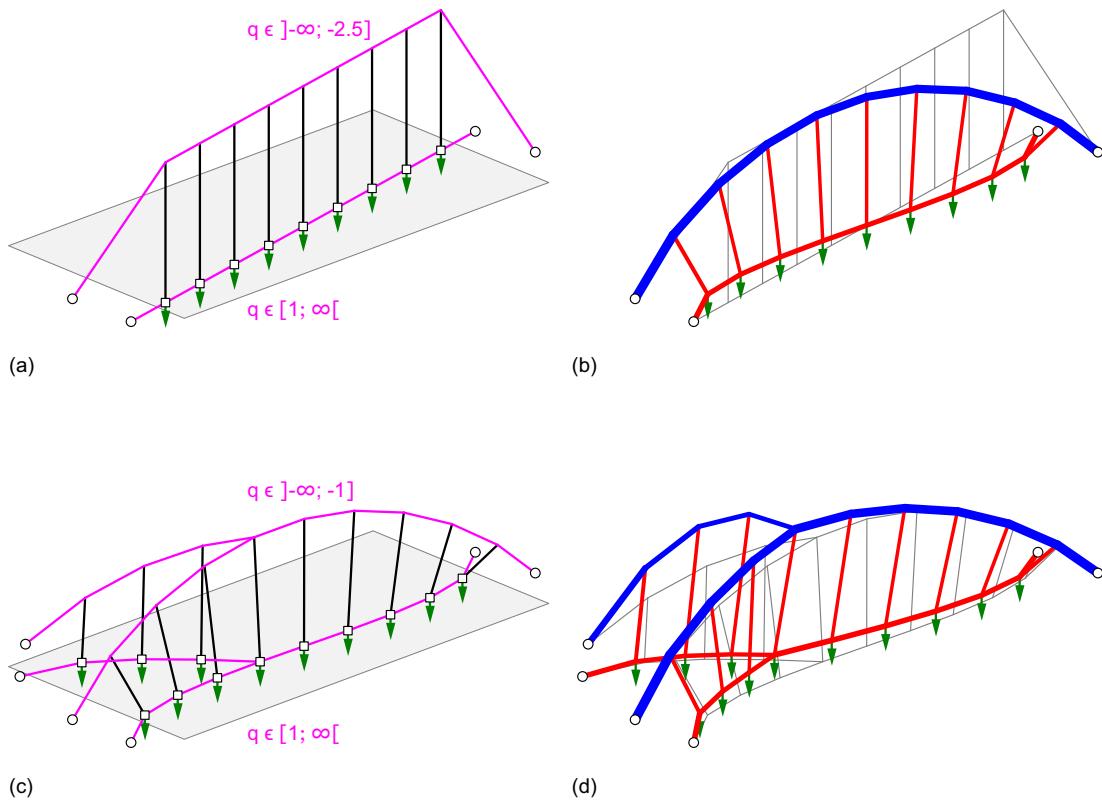


Figure 6. A spatial, bifurcating arch bridge: (a) Initial, user-defined input geometry (b) generated equilibrium geometry of the single arch: (c) the designer copies and rotates the first four segments of the deck and the arch; and (d) final form-found equilibrium geometry of the bifurcating bridge.

compression arch, and vertically supported by a tension funicular (Fig. 7d). In the second case (Fig. 7e), the process generates an equilibrium model also with the deck in compression, but now horizontally and vertically equilibrated by tension funiculars (Fig. 7f). In both cases, the horizontal funicular in compression (Fig. 7d) and the horizontal funicular in tension (Fig. 7f) are inducing almost the

same force in the deck. Hence, the other parts of the structure, the deck and the vertical funicular, are almost identical in inner forces and geometry.

4.3. A pre-stressed, cable-stayed roof

The examples shown in Subsections 4.1 and 4.2 are linear structures. This subsection presents the application of the method to a two-dimensional

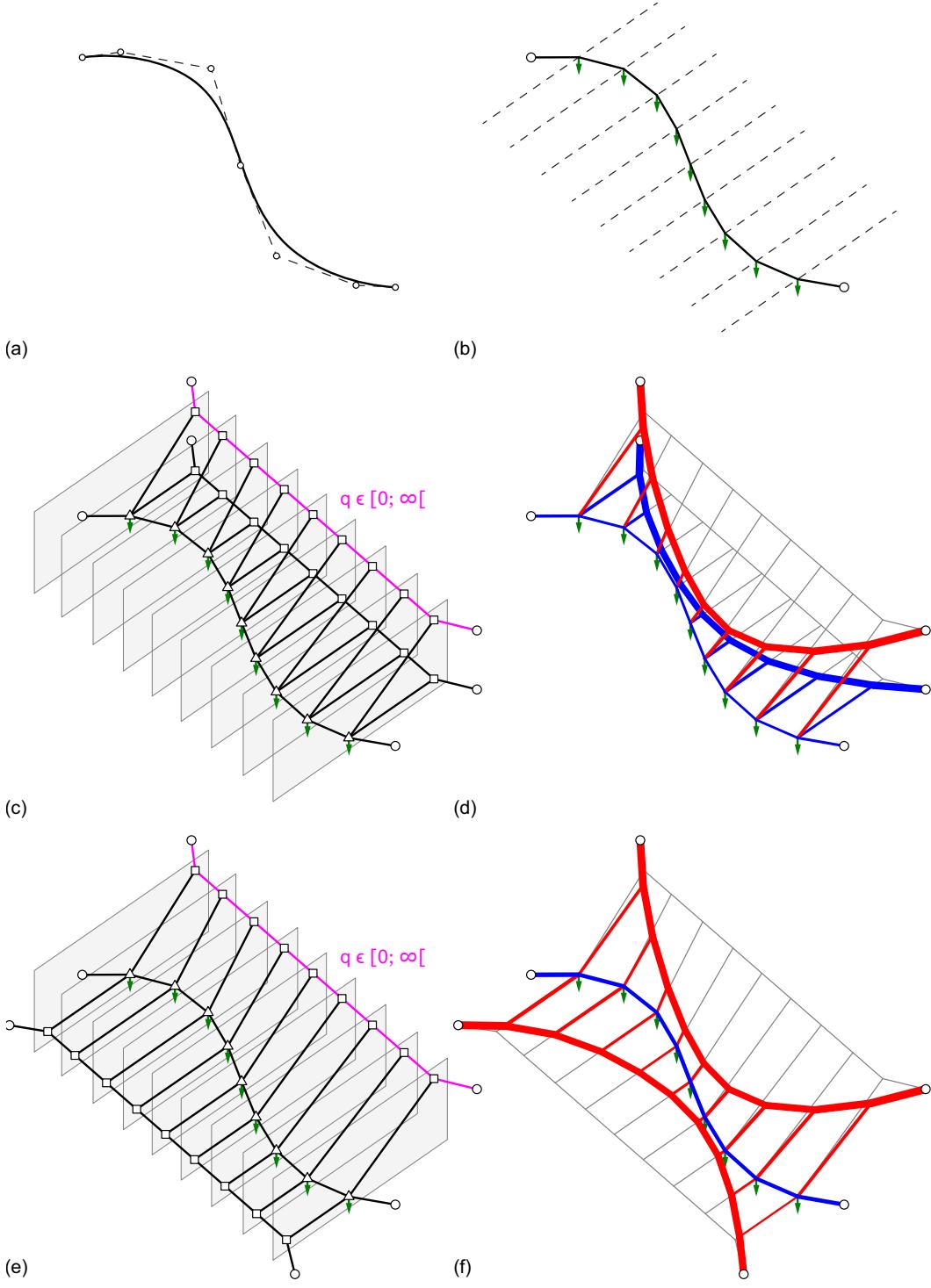


Figure 7. An example of highly constrained form finding of a bridge: (a) the intended path geometry, defined by a freeform curve, and (b) the discretized path with dead load and structural axes (c) constrained model, with positive force density constraints on the cable, and second funicular on the same side of the deck, and (d) the resulting equilibrium model with the deck in compression, horizontally supported by a compression arch, and vertically supported by a tension funicular (e) constrained model, with positive force density constraints on the cable, and second funicular on the opposite side of the deck and (f) resulting equilibrium model with the deck in compression and the two funiculars in tension.

structure. Inspired by an article by Göppert and Stein [30], describing the principle of pre-stressed “spoked wheel” roof structures for sports stadiums, the aim in this example is to design a pre-stressed wheel structure

based on a given freeform surface. This is not an obvious goal as the geometry of both rings depend on each other. For this type of structures, an outer compression ring is connected to an inner tension ring

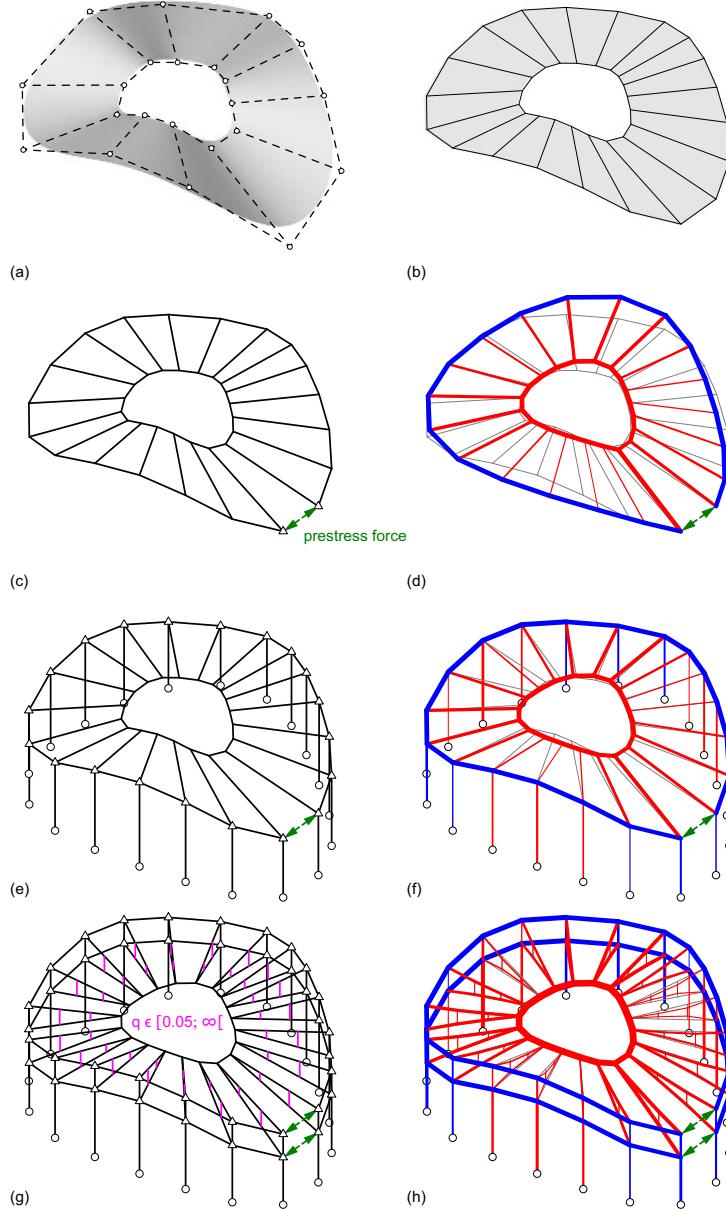


Figure 8. Modelling of a spoked-wheel roof structure: (a) the intended geometry of the roof defined by a freeform surface in space, and (b) discretized surface as polygonal model (c) structural model without any supports, externally pre-stressed, and (d) resulting equilibrium model collapsed to a planar structure (e) geometrically constrained input model with additional columns and supports, and (f) the resulting equilibrium model (g) refined input model with doubled compression ring and radial cable girders, and (h) final equilibrium model.

via radial cables. The dominant loading case is self-stress. To create structural depth in the vertical cross-section, usually either the compression or the tension ring is doubled. The radial cables are then also doubled, forming radial cable girders with additional, connecting vertical cables.

The design process starts with a 3D freeform loop surface, representing the intended geometry of the roof (Fig. 8a). In the next step, the surface is discretized in quadrilateral polygons (Fig. 8b). The edges of the

polygons are interpreted as branches. In the current implementation, it is not possible to run the method without application of external forces, so one branch is replaced by two forces of same magnitude and opposing direction, representing the pre-stressing force. The structure does not have any structural supports (Fig. 8c). The form-finding process generates an equilibrium model working as pre-stressed wheel, but the spatial shape of the initial structure collapses to a plane (Fig. 8d).

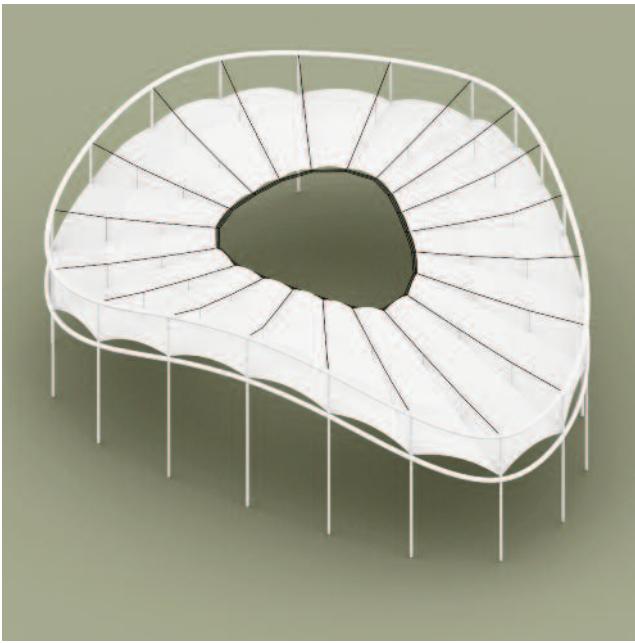


Figure 9. Rendering of a cable-stayed roof based on the equilibrium model presented in Figure 8h. The pre-stressed, spoked-wheel construction forms the primary supporting structure for a membrane roof.

It is the designer's intention to stay closer to the original freeform surface, so the idea is to use columns for balancing the out-of-plane forces of the ring. In order to preserve the exact input geometry of the outer surface edge, the nodes of the ring are fixed in space (Fig. 8e). The resulting equilibrium model is closer to the initial form, only the inner edge deforms into a tension hoop (Fig. 8f).

Based on this geometry, a refined model is built up by the designer. The compression ring is doubled, and the radial cables are replaced by two layers of cables, connected by short vertical elements (coloured in magenta). The force densities of these vertical elements are bounded to a small positive value in order to induce a minimum tension pre-stress (Fig. 8g). Based on the equilibrium geometry yielding from the refined model (Fig. 8h), a generic spoked wheel roof construction with a membrane cover is rendered (Fig. 9).

5. DISCUSSION AND FUTURE WORK

The presented examples demonstrate that the method offers both an intuitive approach to playfully explore spatial, mixed compression and tension equilibrium forms in early design stages, but also enables form finding of equilibrium shapes with bounds on the force densities and constraints on the geometry. The equilibrium model resulting from the optimization process usually forms the starting point for a following

part of the design process that deals with questions of constructive detailing and erection. Since the approach is limited to structures that form mechanisms, a stiffening and bracing scheme has to be developed after a satisfying equilibrium model is obtained. Subsequently, member dimensions are assigned, and analysis regarding structural safety for live loads, stability and serviceability has to be carried out.

In general, form-finding methods are mainly relevant for the design of structures with high permanent loading compared to live loads; such permanent loads are, for instance, high self-weight or high pre-stress. For structures with live loads that have a similar or higher magnitude than the permanent loads, form found geometry might serve as starting point for the design process; but these live loading cases become the dominant factors for structural design. By running the form-finding process with permanent loading combined with live load cases, the designer could gain qualitative insight in equilibrium states for these specific combinations. Furthermore, the designer could even use superposition strategies, combining several equilibrium states to generate form.

The method, implemented using an explicit Runge–Kutta solver with adaptive step size, performs well. Simpler third- and fourth-order Runge–Kutta methods have been tested and work in principle, but need much higher number of iterations, since small step sizes have to be used. Table 2 shows the number of iterations and calculation times for the examples discussed in the paper. The iteration number, for example, of the example shown in Figure 8d is remarkably high. This probably is a consequence of the “floating” setup of the model; without any supports, the system is determinate up to a rigid-body translation. In general, the solving times are not competitive in the sense of a real-time simulation environment; this is mainly a result of the lack of sparse matrix computation routines in the *NumPy* implementation available for *IronPython*. An implementation of the method as compiled plugin using a fast linear systems solver should speed up the solving time by two orders of magnitude.

Although the method allows an intuitive modelling of equilibrium forms, background knowledge in structural design is obviously required from the designer. A general issue with the equilibrium modelling process is the possibility to create over-constrained models easily, resulting in slow convergence or no convergence at all. Developing a general approach to detect such situations is difficult and remains future work. Another source for possible

Table 2. Number of branches, free nodes, iterations, and solving times for the examples presented in the paper. The computation has been executed on a standard PC (Intel Core Duo, 2.8 GHz)

	Fig. 6b	Fig. 6d	Fig. 7d	Fig. 7f	Fig. 8d	Fig. 8f	Fig. 8h
branches m	29	40	48	48	59	79	258
free nodes n	18	24	27	27	40	40	140
Iterations t	50	48	121	136	636	48	57
time [s]	2.6	3.1	9.8	11.3	68.6	4.9	89.0

failing of the form-finding process is topologic degeneration, i.e. branches with vanishing lengths.

Future work will also include the implementation of further geometric constraints, such as nodes constrained to spatial curves, non-planar surfaces or polygonal meshes. In this case, the “free” component of the residual will be obtained by projecting the residual onto the tangential line or plane of the constraining object. The gradient decomposition described in Section 3.2 can easily be extended to other constraints such as e.g. fixed branch lengths or bounds, or even constraints, on forces directly, rather than on force densities. Furthermore, more specific support conditions, such as movable supports constrained to geometric objects, or supports that can take forces only in specific directions, will be implemented.

6. CONCLUSION

The presented form-finding method, implemented within a CAD environment, allows for the intuitive exploration of spatial, pin-jointed, kinematic structures with mixed compression and tension forces in equilibrium for given design loads. Through the definition of various constraints on form and forces, the method can also be used for the geometric rationalization of structures according to constructive requirements. This opens up new possibilities in the design of efficient structural forms that visually blend between freeform design and known typologies of lightweight structures.

ACKNOWLEDGEMENTS

We owe thanks to Diederik Veenendaal and Dr. Tom Van Mele for their suggestions and critical comments during the development of the method. The first author would furthermore like to thank Prof. Dr. Roman Kemmler (Schlaich, Bergermann & Partner) for inspiring discussions and fruitful advice.

REFERENCES

- [1] Pottman, H., Asperl, A., Hofer, M. and Kilian, A., *Architectural Geometry*, Bentley Institute Press, Exton, 2007.
- [2] Tessmann, O., *Collaborative Design Procedures*, PhD thesis, Universität Kassel, 2008.
- [3] Sasaki, M., *Morphogenesis of Flux Structure*, AA Publications, London, 2007, 100–109.
- [4] Allen, E., Guastavino, Dieste, and the two Revolutions in Masonry Vaulting, in: Anderson, S. ed., *Eladio Dieste - Innovations in Structural Art*, Princeton Architectural Press, New York, 2004, 66–75.
- [5] Gaß, S., *Experiments*, Institute for Lightweight Structures, Stuttgart, 1990, 1.1–1.6.
- [6] Mostafavi, M., *Structure as Space*. AA Publications, London, 2006, 100–103.
- [7] Brunetta, V. and Patteeuw, V., *Ney & Partners: Freedom of Form Finding*, A16 & VAI, Antwerpen, 2005, 36–38.
- [8] Veenendaal, D. and Block, P., An overview and comparison of structural form finding methods for general networks, *International Journal of Solids and Structures*, 2012, 49(26), 3741–3753.
- [9] Block, P. and Ochsendorf, J., Thrust Network Analysis: A new methodology for three-dimensional equilibrium, *Journal of the International Association for Shell and Spatial Structures*, 2007, 48(3), 167–173.
- [10] Miki, M. and Kawaguchi, K., Extended force density method for form finding of tension structures, *Journal of the International Association for Shell and Spatial Structures*, 2010, 51(4), 291–303.
- [11] Kemmler, R., Formfindung: Die Interaktion von Kraft und Geometrie, *Stahlbau*, 2012, 81(6), 476–480.
- [12] Kilian, A. and Ochsendorf, J., Particle-Spring Systems for Structural Form Finding, *Journal of the IASS*, 2005, 46(147), 77–84.
- [13] Rippmann, M., Lachauer, L. and Block, P., Interactive Vault Design, *International Journal of Space Structures*, 2012, 27(4), 219–230.
- [14] Tachi, T., Interactive Freeform Design of Tensegrity, in: Hesselgren, L., Sharma, S., Wallner, J., Baldassini, N., Bompas, P., Raynaud, J. eds., *Advances in Architectural Geometry*, Springer, Paris, 2012, 259–268.
- [15] McNeel, B., *Rhinoceros - modeling tools for designers*, www.rhino3d.com, 2013.
- [16] Culmann, K., *Die graphische Statik*, Meyer & Zeller, Zürich, 1866.

- [17] Laffranchi, M., *Zur Konzeption gekrümmter Brücken*, PhD thesis, ETH Zurich, 1999, 23–28.
- [18] Lachauer, L. and Kotnik, T., Curved Bridge Design, in: Gengnagel, C., Kilian, A., Palz, N. and Scheurer, F. eds., *Computational Design Modeling*, Springer, Berlin, 2011, 145–152.
- [19] Schlaich, J. and Schäfer, K., Design and detailing of structural concrete using strut-and-tie models, *The Structural Engineer*, 1991, 69(6), 113–125.
- [20] Muttoni, A., Schwartz, J. and Thürlmann, B., *Design of Concrete Structures With Stress Fields*, Birkhäuser, Basel, 1997.
- [21] Linkwitz, K. and H.-J. Schek, Einige Bemerkungen zur Berechnung vorgespannter Seilnetzkonstruktionen, *Ingenieur-Archiv*, 1971, 40(3), 145–158.
- [22] Schek, H.-J., The Force Density Method for Form Finding and Computation of General Networks, *Computer Methods in Applied Mechanics and Engineering*, 1974, 3(1), 115–134.
- [23] Linkwitz, K., H.-J. Schek, and L. Gründig, Die Gleichgewichtsberechnung von Seilnetzen unter Zusatzbedingungen, *Ingenieur-Archiv*, 1974, 43(4), 183–192.
- [24] Timoshenko, S. and Young, D., *Theory of Structures*. McGraw-Hill, New-York, 1945, 188–194.
- [25] Bouaziz, S., Deuss, M., Schwartzburg, Y., Weise, T. and Pauly, M., Shape-Up: Shaping Discrete Geometry with Projections, *Computer Graphics Forum*, 2012, 31(5), 1657–1667.
- [26] Barnes, M., R., Form Finding and Analysis of Tension Structures by Dynamic Relaxation, *International Journal of Space Structures*, 1999, 14(2), 89–104.
- [27] Baer, S., *Rhino.Python - Cross platform scripting for Rhino*, <http://python.rhino3d.com>, 2012.
- [28] NumPy, <http://www.numpy.org>, 2013.
- [29] Kiusalaas, J., *Numerical Methods in Engineering with Python*, Cambridge University Press, Cambridge, 2005, 275–281.
- [30] Göppert, K. and Stein, M., A Spoked Wheel Structure for the World’s largest Convertible Roof – The New Commerzbank Arena in Frankfurt, Germany, *Structural Engineering International*, 2007, 4(17), 282–287.