Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

Computational Form-finding of Structures through Constraint Projections

Jonas WARMUTH*, Pierluigi D'ACUNTO[a], Corentin FIVET[b]

*[b] Structural Xploration Lab, Ecole Polytechnique Fédérale de Lausanne (EPFL), Fribourg, Switzerland
Passage du Cardinal 13B, CH-1700 Fribourg
jonas.warmuth@epfl.ch

[a] Professorship of Structural Design, TUM School of Engineering and Design, Technical University of Munich, Munich, Germany

Abstract

This paper presents a computational form-finding framework for discrete equilibrium models aimed at early design stages and the exploration of innovative structural forms in static equilibrium. Geometric, static, and combined design constraints are incorporated here without computing derivatives. The method is based on an optimization scheme that works with a proximity function and constraint projections, a technique derived from computer graphics. It provides instant feedback, supporting a flexible and interactive design process. The framework is implemented in the Rhino/Grasshopper environment. To demonstrate its applicability, form-finding examples for a roof structure, a curved bridge, and a grid shell are carried out, and potential implications for the design process are discussed.

Keywords: Form-finding, Conceptual Structural Design, Interactive Design, Structural Optimization, Constraint Projection

1. Introduction

In the realm of structural design, the conceptual phase stands as a critical moment, signifying the most significant impact of a designer on the final product. At this stage, the exploration of design options assumes key importance, laying the basis for a sound design process. Form-finding can serve as a powerful tool in this process, facilitating the generation and exploration of efficient geometries. For the design of cable nets and structural membranes, form-finding has long been indispensable due to the absence of inherent bending capacity in elements. Consequently, finding an appropriate form becomes essential, as forces must be borne through the form itself. However, the advantages of form-finding extend beyond these specific structural types, as it empowers the design of more efficient structures across diverse domains. Traditionally, form-finding relied on physical models, exemplified by the hanging models of Heinz Isler [1] or the soap film models of Frei Otto [2]. With the advent of the computational age, this paradigm has shifted towards digital methodologies [3]. An important milestone was set by Linkwitz and Schek [4] with the force density method (FDM). It is noteworthy to emphasize that with the FDM, no materialization occurs, rendering the search for equilibrium solely a matter of reconciling form and forces.

1.1 Related work and contributions

A great advantage of the FDM is that it converts a nonlinear form-finding problem to a linear one. This advantage, however, is brought with the cost of flexibility as force densities (i.e., force-to-length ratios for all members of the structure) have to be introduced a priori. Efforts to overcome this limitation were made by Schek [5], Linkwitz and Veenendaal [6], and Tamai [7], among others, who introduced nonlinear extensions of the FDM. Nevertheless, utilizing force densities as design drivers still presents a challenge due to their abstract nature. A solution to this is the statically-geometrically coupled method by Kemmler [8]. This method works with explicit forces when describing equilibrium conditions. In addition, internal forces are described as variables. While this brings the problem back to a nonlinear state, it also enables the incorporation of supplementary design constraints. However, it demands careful attention during problem setup, as the formulated constraints must avoid contradictions and ambiguity.

In recent years, graphic statics [9], [10] has gained popularity for the exploration and form-finding of spatial structures. It describes a principal force flow of a structure through form and force diagrams. Ohlbrock and Schwartz [11] and subsequently Ohlbrock and D'Acunto [12] formalized this in the Combinatorial Equilibrium Modelling (CEM), a computational framework for the form-finding and conceptual design of structures. Later on, the CEM was combined with Graph Neural Networks [13] to simplify the input handling. Jasienski et al. [14] have introduced a computational implementation for 3D vector-based graphic statics that allows the design of spatial structures in static equilibrium. Mirtsopoulos and Fivet [15] presented a generative approach based on grammar rules to explore a vast amount of structural forms and combined it with interactive genetic algorithms [16]. While their approach is powerful for exploration purposes, it remains challenging to maintain control over the resulting forms, introduce design constraints, and offer fast feedback on intended designs.

Hence, this paper presents an original form-finding framework that allows any geometric and static constraints while maintaining fast feedback and control of an intended design. The framework is built upon constraint projections, a method derived from the field of computer graphics, more precisely from discrete geometry shaping, introduced by Bouaziz et al. [17]. This method allows the integration of a vast amount of design constraints while being relatively fast. In addition, design constraints can be contradictory, which will lead to a least-squares solution. The Grasshopper plugin Kangaroo2 [18] follows a related approach. Takahashi and Ney [19] presented a similar approach based on the concept of constraint projections to form-find structures. Takahashi [20] even extended it with the implementation of global objectives. Their approach is based on force densities for equilibrium computation, which can pose challenges for applying explicit force constraints. Furthermore, as described earlier, working with force densities involves an abstract procedure. In contrast, the approach presented here uses explicit force formulations, offering the significant advantage of (1) formulating direct force constraints and (2) providing an intuitive understanding of choices regarding forces.

1.2 Layout exploration as a pre-step to form-finding

It is worth mentioning that form-finding methods typically operate with pre-set topologies, limiting the scope of design exploration. It can, therefore, be useful to first explore options for a global layout – a rough global image of the final result – before diving into a form-finding process in which geometric and static boundary conditions are described explicitly and quantifiable. In previous research, Warmuth et al. [21] introduced a framework based on layout optimization aimed at obtaining an initial understanding of a structure's global appearance. Traditionally, layout optimization aims to minimize the structure's volume, following methods like the ground structure approach [22]. In this method, a network of interconnected nodes and lines is created, with the objective of selecting the combination of lines that minimizes the overall volume while adhering to static equilibrium. However, following a pure optimization – especially in early design stages – often yields structurally similar outcomes, which is contradictory to a proper exploration of diverse structural forms.

A key concept of the layout optimization presented by Warmuth et al. [21] is the customization of the ground structure from which a layout is derived. This customization involves designers interactively placing nodes within the ground structure, along with defining design objectives formulated as optimization constraints to guide the optimization process toward desired outcomes. This approach empowers designers to explore structural forms beyond predefined types or topologies and is, therefore, a suitable pre-step to a following form-finding process. It is crucial to emphasize that both customized layout optimization and form-finding can function independently, generating valid structures in static equilibrium. However, it is the synergistic combination of these steps that exploits the full potential for exploring diverse design options.

2. Method

The method presented in this paper is based on a proximity function to be minimized and the concept of constraint projections, as introduced earlier by Bouaziz et al. [17] and later used by Takahashi and Ney [19] for the form-finding of discrete structures based on force densities.

2.1 Local-global optimization scheme

The general approach is to resolve a form-finding problem by solving an optimization problem that is composed solely of constraints. These constraints may take various forms, i.e., geometric, static, or their combination. To satisfy these constraints a local-global optimization scheme is introduced. The key idea of this scheme is to replace the solving of constraint equations by the minimization of a proximity function that computes the weighted sum of squared distances of a point to its projection onto a collection of constraints $\{C_1, C_2, \ldots, C_i\}$. Let $d_i(\mathbf{x})$ measure the *'least amount of change'* in $\mathbf{x}$ to satisfy a constraint C_i. Then, the proximity function can be stated as

$$\Theta(\mathbf{x}) = \sum_{i}^{n} w_i d_i(\mathbf{x})^2 \tag{1}$$

where n denotes the number of constraints and w_i is a non-negative weight that controls the relative importance of each constraint C_i. $d_i(\mathbf{x})^2$ describes the distance of a set of variables $\mathbf{x}$ to its projection $P_i(\mathbf{x})$ onto a constraint C_i. Thus, the proximity function can be rewritten as

$$\Theta(\mathbf{x}) = \sum_{i}^{n} w_i \|\mathbf{x} - P_i(\mathbf{x})\|_2^2 \tag{2}$$

This function determines how well all constraints are satisfied. Finding a solution that minimizes the proximity function to $\Theta(\mathbf{x}) = 0$, therefore, satisfies all constraints. Otherwise, a least squares solution is found. In order to solve Equation (2), the optimization is divided into two disjoint, local and global, steps, executed in an alternating update scheme.

1. local step: Compute the projection $P_i(\mathbf{x})$ for each constraint C_i using the current $\mathbf{x}$.
2. global step: Compute a new $\mathbf{x}$ based on the projections $P_i(\mathbf{x})$ of all constraints $\{C_1, C_2, \ldots, C_i\}$.

According to Bouaziz et al. [17], this scheme is guaranteed to converge to a local minimum. The great advantage here is that each constraint can be described independently and computed in parallel in the local step. In addition, no derivatives are required if projections of constraints are directly available. For this research, one constraint is crucial, i.e., the one that ensures that the structure is in static equilibrium. This constraint is explained in the following.

2.2 Equilibrium constraint

To obtain static equilibrium in a structure, all nodes need to be in equilibrium. Therefore, the global equilibrium is decomposed into a local equilibrium for each node, which is solved independently of the other nodal equilibrium constraints. For the example in Figure 1a, this means that eight equilibrium constraints need to be applied to ensure that the entire structure is in equilibrium. Figure 1b shows how equilibrium is computed for one node, here $\mathbf{n}_0$. The variables at the local node level are the coordinates $[\mathrm{x}, \mathrm{y}, \mathrm{z}]^\mathrm{T}$ of $\mathbf{n}_0$, which can be observed in the form diagram in Figure 1b, and the internal bar forces s_i of all bars connected to $\mathbf{n}_0$, shown in the force diagram in Figure 1c. This means that, in the scope of the local node equilibrium, the connected nodes $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ do not move and serve as virtual supports.

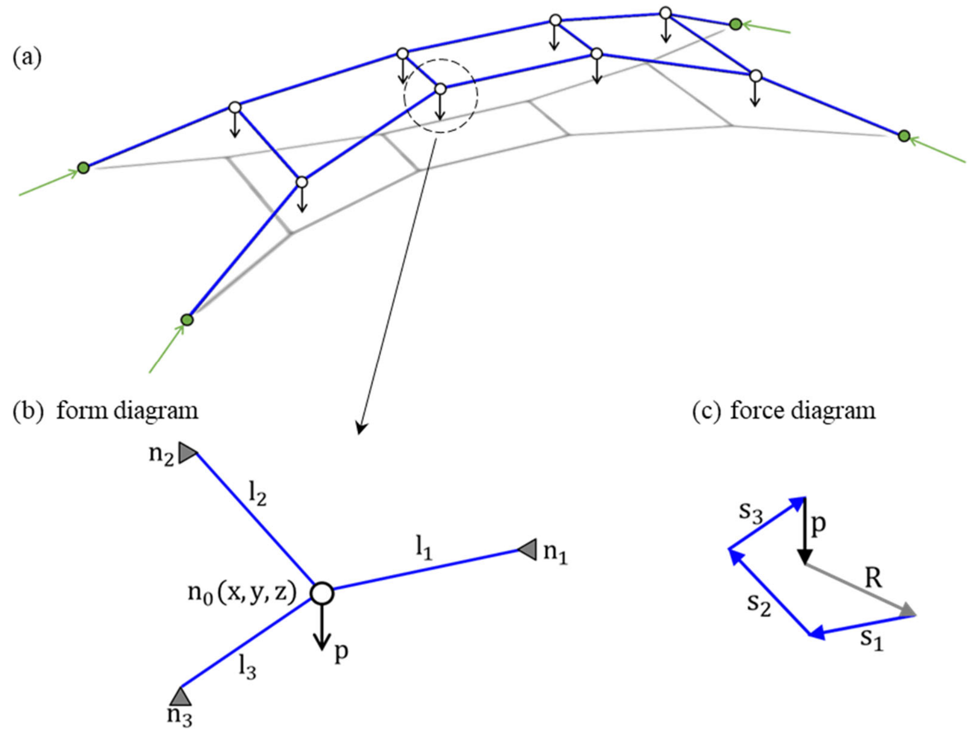

Figure 1. Computation of the equilibrium at the node level.

Equilibrium at a node is achieved when the residual force vector $\mathbf{R} = \left[R_x, R_y, R_z\right]^{\mathrm{T}}$, as shown in the force diagram in Figure 1c, is null. For $\mathbf{n}_0$ the residual force in x-direction R_x can be written as

$$R_x = \frac{s_1(n_{1,x} - n_{0,x})}{l_1} + \frac{s_2(n_{2,x} - n_{0,x})}{l_2} + \frac{s_3(n_{1,x} - n_{0,x})}{l_3} - p_x \tag{3}$$

or generalized as

$$R_x = \sum_i^k \frac{s_i \Delta x_i}{l_i} - p_x \tag{4}$$

where k is the number of connected bars, $\Delta x_i = n_{i,x} - n_{0,x}$, $l_j = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}$, and p_x is the x-direction component of a given external load. R_y and R_z are computed similarly in the y- and z-directions. Thus,

$$\min_{x,y,z,s_i} \mathbf{R} \rightarrow 0 \tag{5}$$

can be efficiently solved using the Newton-Raphson method since the gradient of $\mathbf{R}$,

$$\nabla \mathbf{R} = \begin{bmatrix} \sum_i \frac{-(\Delta y_i^2 + \Delta z_i^2)}{l_i^3} s_i & \sum_i \frac{\Delta x_i \Delta y_i}{l_i^3} s_i & \sum_i \frac{\Delta x_i \Delta z_i}{l_i^3} s_i & \frac{\Delta x_i}{l_i} \\ \sum_i \frac{\Delta y_i \Delta x_i}{l_i^3} s_i & \sum_i \frac{-(\Delta x_i^2 + \Delta z_i^2)}{l_i^3} s_i & \sum_i \frac{\Delta y_i \Delta z_i}{l_i^3} s_i & \frac{\Delta y_i}{l_i} \\ \sum_i \frac{\Delta z_i \Delta x_i}{l_i^3} s_i & \sum_i \frac{\Delta z_i \Delta y_i}{l_i^3} s_i & \sum_i \frac{-(\Delta x_i^2 + \Delta y_i^2)}{l_i^3} s_i & \frac{\Delta z_i}{l_i} \end{bmatrix} \tag{6}$$

is available analytically, where i indicates the number of bars connected to a node. $\nabla\mathbf{R}$ consists of three rows for the residual force vector in x-, y-, and z-directions. The number of columns, however, is not fixed as it depends on the number of bars connected to a node. Columns 1-3 refer to the coordinate variables x, y, and z; columns four and higher refer to the force variables s_i. For instance, if two nodes are connected to a node $\mathbf{n}_0$, there are two columns for force variables, i.e., five columns in total.

After solving, the new variables $\mathbf{x}^* = \left[n_{0,x}^*, n_{0,y}^*, n_{0,z}^*, s_1^*, s_2^*, s_3^*\right]^{\mathrm{T}}$ are the projection $P_i(\mathbf{x})$ for the equilibrium constraint C_i at node $\mathbf{n}_0$. By computing the projections for all other nodes, the local step of the alternating update scheme is fully computed, considering static equilibrium.

2.3 Other constraints

Besides static equilibrium, other constraints are available. The following constraints are implemented in the current framework. However, any constraint can be incorporated if a projection $P_i(\mathbf{x})$ can be computed.

Table 1. Available constraints in the current framework.

Constraint type	Geometric	Static
	- Target length	- Equilibrium
	- Equal length	- Target force
	- Points on plane	- Equal force
	- Points on surface	
	- Points on curve	
	- Colinear points	

A broader selection of constraints can be found in Takahashi and Ney [19], Bouaziz et al. [17], or Piker [18].

3. Examples

This section showcases three form-finding examples utilizing the framework introduced in Section 3, each representing a structure with diverse requirements in the form-finding process.

3.1 Curved bridge

First, the form-finding process is applied to a curved arch bridge. Here, the objective is to shape the bridge deck into an S-like curve. Figure 2a depicts the desired deck shape represented in light blue, alongside the initial setup featuring supports and loads. Additionally, an equal length constraint is imposed on the arch bars to ensure a visually cohesive appearance. In Figure 2b, the form-found structure in equilibrium is displayed. To satisfy equilibrium requirements, the arch has adopted an S-shaped configuration, counter to the movement of the deck. Figure 2c provides a schematic view of the curved bridge, encapsulating the achieved form and structural integrity. Overall, the entire computation process, encompassing the satisfaction of all constraints, converged within 5.8 seconds.

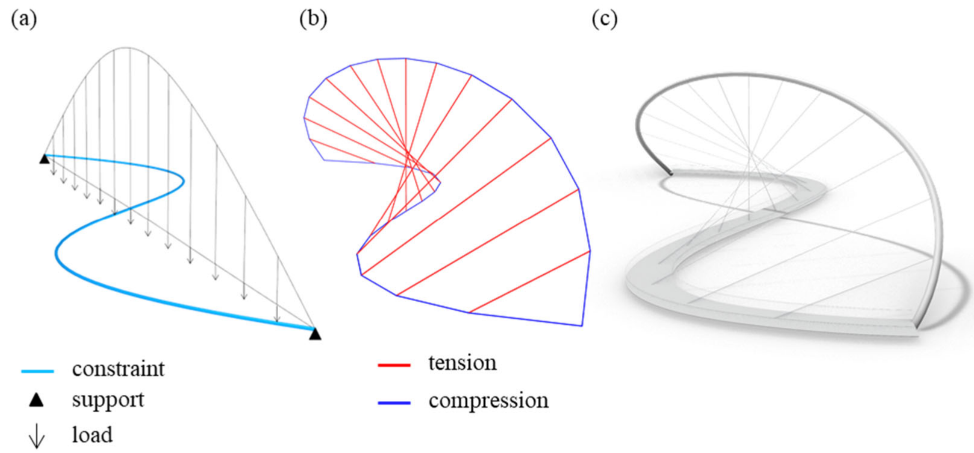

Figure 2. Form-finding of a curved bridge, (a) initial setup and design constraints, (b) discrete equilibrium model, (c) schematic view of the resulting form-found bridge.

3.2 Roof structure

Next, the form-finding of a roof structure is shown. The initial layout, illustrated in Figure 3a, comprises four arches intended to span the final roof. These arches are interconnected with bars, facilitating self-balancing. Furthermore, the arches are required to come close to a specific curvature, delineated by the light blue lines. In Figure 3b, the form-found roof structure is depicted, with the arches under compression and their connections under tension. Figure 3c provides a schematic view of the roof configuration, capturing its structural arrangement. Convergence was achieved after 6.7 seconds.

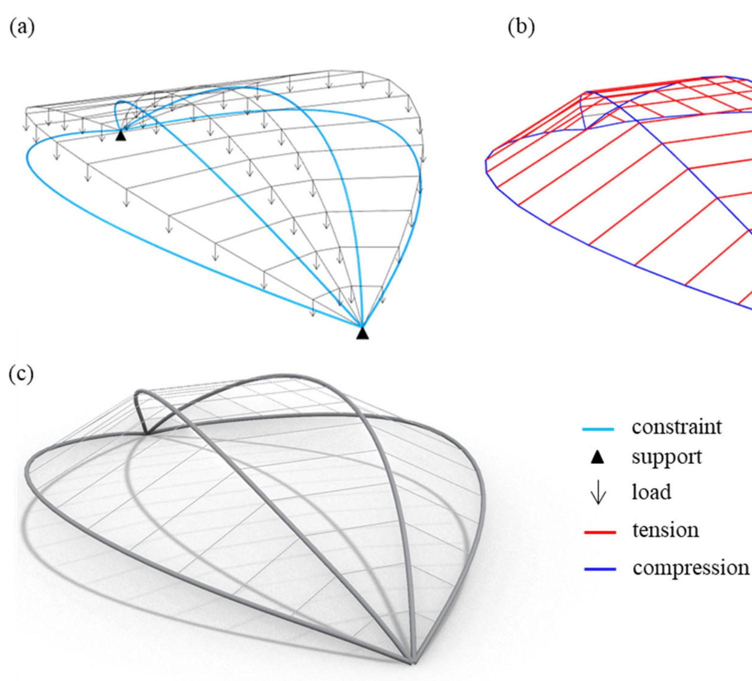

Figure 3. Form-finding of a roof structure, (a) initial setup and design constraints, (b) discrete equilibrium model, (c) schematic view of form-found roof structure.

3.3 Grid shell

As a third example, a free-form grid shell is conceptualized. In Figure 4a, the initial setup presents a grid composed of 11x11 bars. The three light blue curves delineate the desired shape of the grid shell. Additionally, uniformity in bar length is sought to achieve an aesthetically pleasing appearance. Figure 4b illustrates the resulting form-found grid shell in equilibrium, comprising ten arches in compression interconnected by bars in tension. Notably, the central *"arch"* is in tension, ensuring adherence to the imposed constraints. Figure 4c provides a schematic representation of the grid shell and its desired shape. The form-finding converged after 2.3 seconds.

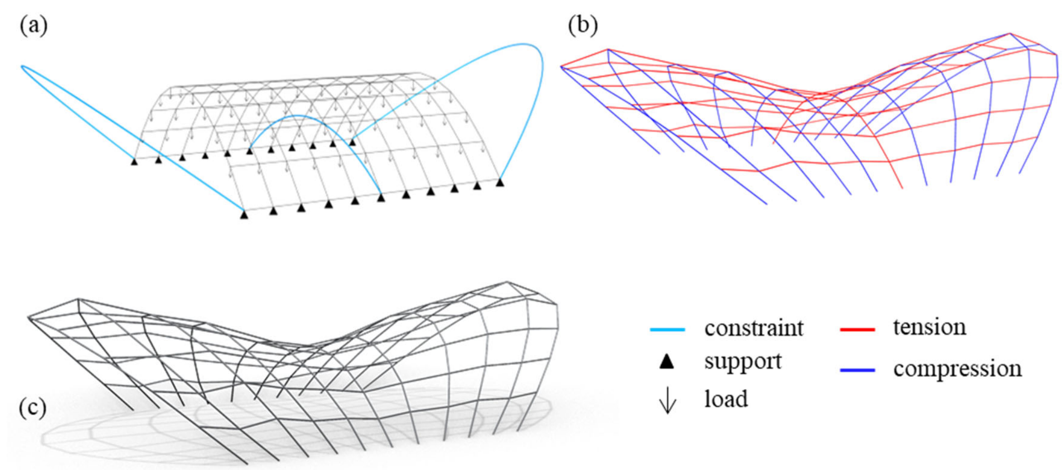

Figure 4. Form-finding of a free-form grid shell, (a) initial setup and design constraints, (b) discrete equilibrium model, (c) schematic view of the resulting form-found grid shell.

4. Conclusion

This paper presented a numerical form-finding framework for discrete equilibrium models. Central to this method is the minimization of a proximity function by employing constraint projections to fulfill static and geometric form-finding objectives, e.g., static equilibrium. The method's strength lies in its ability to compute constraints simultaneously, thereby significantly enhancing computational efficiency. Furthermore, the absence of derivative computations when constraints are expressible in closed form is a notable advantage.

Implemented within the Rhino/Grasshopper environment, the framework offers practical applicability, as demonstrated through various examples for different types of structural forms. However, it is essential to acknowledge certain limitations. The outcome of the form-finding process is contingent upon the initial form and the selection of initial forces. Additionally, due to the utilization of different variable sets (coordinates and internal bar forces), their relative relationship can introduce instability into the process. An a priori normalization of variables could, however, solve this and is the subject of future development. Moreover, proper problem setup, a profound understanding of structural principles, and a conceptualization of the intended structure are crucial, as improperly defined scenarios may yield unsatisfactory results. Nonetheless, the framework presented herein offers a fast, general, and intuitive approach that facilitates the constraint-based exploration of innovative structural forms.

References

[1] G. Boller, "The model as a working method. Heinz Isler's experimental approach to shell design," Doctoral Thesis, ETH Zurich, 2022. doi: 10.3929/ethz-b-000605815.

[2] F. Otto and B. Rasch, *Finding Form : Towards an Architecture of the Minimal*. Edition Axel Menges, 1995.

[3] G. Boller and P. D'Acunto, "Structural design via form finding: Comparing Frei Otto, Heinz Isler and Sergio Musmeci," in *History of Construction Cultures Volume 2*, CRC Press, 2021.

[4] K. Linkwitz and H.-J. Schek, "Einige Bemerkungen zur Berechnung von vorgespannten Seilnetzkonstruktionen," *Ing. arch*, vol. 40, no. 3, pp. 145–158, May 1971, doi: 10.1007/BF00532146.

[5] H.-J. Schek, "The force density method for form finding and computation of general networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 3, no. 1, pp. 115–134, Jan. 1974, doi: 10.1016/0045-7825(74)90045-0.

[6] K. Linkwitz and D. Veenendaal, "Nonlinear force density method: Constraints of force and geometry," in *Shell Structures for Architecture*, Routledge, 2014.

[7] H. Tamai, "Advanced application of the force density method in multidisciplinary design practice by incorporating with optimization using analytical derivatives," in *Proceedings of IASS Annual Symposia*, Sep. 2013, pp. 1–9.

[8] R. Kemmler, "Formfindung: Die Interaktion von Kraft und Geometrie," *Stahlbau*, vol. 81, Jun. 2012, doi: 10.1002/stab.201201564.

[9] P. D'Acunto, J.-P. Jasienski, P. O. Ohlbrock, C. Fivet, J. Schwartz, and D. Zastavni, "Vector-based 3D graphic statics: A framework for the design of spatial structures based on the relation between form and forces," *International Journal of Solids and Structures*, vol. 167, pp. 58–70, Aug. 2019, doi: 10.1016/j.ijsolstr.2019.02.008.

[10] M. Akbarzadeh, T. Van Mele, and P. Block, "On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams," *Computer-Aided Design*, vol. 63, pp. 118–128, Jun. 2015, doi: 10.1016/j.cad.2015.01.006.

[11] P. O. Ohlbrock and J. Schwartz, "Combinatorial equilibrium modeling," *International Journal of Space Structures*, vol. 31, no. 2–4, pp. 177–189, Jun. 2016, doi: 10.1177/0266351116660799.

[12] P. O. Ohlbrock and P. D'Acunto, "A Computer-Aided Approach to Equilibrium Design Based on Graphic Statics and Combinatorial Variations," *Computer-Aided Design*, vol. 121, p. 102802, Apr. 2020, doi: 10.1016/j.cad.2019.102802.

[13] L. Bleker, R. Pastrana, P. O. Ohlbrock, and P. D'Acunto, "Structural Form-Finding Enhanced by Graph Neural Networks," in *Towards Radical Regeneration*, C. Gengnagel, O. Baverel, G. Betti, M. Popescu, M. R. Thomsen, and J. Wurm, Eds., Cham: Springer International Publishing, 2023, pp. 24–35. doi: 10.1007/978-3-031-13249-0_3.

[14] J.-P. Jasienski, Y. Shen, P. O. Ohlbrock, D. Zastavni, and P. D'Acunto, "A computational implementation of Vector-based 3D Graphic Statics (VGS) for interactive and real-time structural design," *Computer-Aided Design*, vol. 171, p. 103695, Jun. 2024, doi: 10.1016/j.cad.2024.103695.

[15] I. Mirtsopoulos and C. Fivet, "Design space exploration through force-based grammar rule," *archiDOCT*, vol. 8, no. 1, pp. 50–64, Jul. 2020.

[16] I. Mirtsopoulos and C. Fivet, "Structural topology exploration through policy-based generation of equilibrium representations," *Computer-Aided Design*, vol. 160, p. 103518, Jul. 2023, doi: 10.1016/j.cad.2023.103518.

[17] S. Bouaziz, M. Deuss, Y. Schwartzburg, T. Weise, and M. Pauly, "Shape-Up: Shaping Discrete Geometry with Projections," *Computer Graphics Forum*, vol. 31, no. 5, pp. 1657–1667, 2012, doi: 10.1111/j.1467-8659.2012.03171.x.

[18] D. Piker, "Kangaroo: Form Finding with Computational Physics," *Architectural Design*, vol. 83, no. 2, pp. 136–137, 2013, doi: 10.1002/ad.1569.

[19] K. Takahashi and L. Ney, "Advanced form finding by constraint projections for structural equilibrium with design objectives," in *Proceedings of IASS Annual Symposia*, Jul. 2018, pp. 1–8.

[20] K. Takahashi, "Advanced form-finding with design constraints and objectives through constraint projection," Aug. 2021.

[21] J. Warmuth, P. D'Acunto, and C. Fivet, "Computational Conceptual Design – Typological Exploration Of Spatial Truss Systems Through Optimization," *Journal of the International Association for Shell and Spatial Structures*, vol. 64, no. 4, pp. 289–297, Dec. 2023, doi: 10.20898/j.iass.2023.026.

[22] W. Dorn, R. Gomory, and H. J. Greenberg, "Automatic design of optimal structures," *Journal de Mècanique*, vol. 3, pp. 25–52, 1964.