Computer-Aided Design 93 (2017) 26–37

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Bi-directional algebraic graphic statics

Vedad Alic *, Daniel Åkesson

John Ericssons väg 1, 223 63 Lund, Sweden

ARTICLE INFO

Article history:
Received 22 December 2016
Accepted 15 August 2017

Keywords:
Interactive graphic statics
Matrix analysis of structures
Graph theory
Reciprocal figures
Force diagram manipulations
Conceptual design

ABSTRACT

A pre-existing algebraic graphic statics method is extended to allow for interactive manipulations of the force diagram, from which an updated form diagram is determined. Newton's method is used to solve a set of non-linear equations, and the required Jacobian matrix is derived. Additional geometric constraints on the form diagram are introduced, and methods for improving the robustness of the method are presented. We discuss the implementation of the method as a back-end to an interactive application, and demonstrate the usability of the method in several examples where the qualities of directly manipulating the force diagram are emphasized.

1. Introduction

Graphic statics is a graphical method to compute the forces within a 2D axially loaded static structure. As forces are represented graphically by using vectors, no numerical analysis is required to calculate the magnitude of the forces. The use of graphic statics grew from the 19th and 20th centuries but lost popularity due to the emergence of computational analysis methods. However, being an intuitive method of visualizing form and forces, graphic statics has recently seen an increase in popularity.

Maxwell established that for axially loaded structures, the nodes and polygons in the form diagram have reciprocal polygons and nodes in the force diagram. The form diagram represents the structure, and the force diagram represents the static equilibrium [1] for that structure. Maxwell's reciprocal diagrams were extended by Cremona [2], which provided a base for graphic statics. However, the first comprehensive presentation of graphic statics was by Culmann [3]. The method has historically been used to design structures by, for instance, Maurice Koechlin (co-designer of the Eiffel Tower) and Robert Maillart [4]. The reciprocity between the form and force diagrams can provide the designer with insight into the force distribution within a structure, and aid the intuitive understanding of the relationship between form and forces.

In recent literature it has been applied to the design of structural masonry [5–7] and structural optimization [8]. Methods for using graphic statics for the design of post-tensioned funiculars has been presented in [9]. The relationship between graphic statics and the discrete Airy stress function have been the topic of [6,7,10]. The relationships are generalized to discontinuous Airy stress functions in [11], enabling the analysis of two-dimensional frames with bending moments. Extensions of graphic statics to three-dimensional structures are presented in [12–14]. Graphic statics has been combined with shape grammars in [15] for rapid generation of diverse structures in equilibrium. In [16,17] a constraints-based approach to graphic statics is presented, where geometric constraints are employed to enforce the conditions between the form and force diagrams.

Digital implementations of graphic statics have leveraged the graphical nature of the method into interactive applications that enable real-time feedback of the relationship between form and forces [18–20].

1.1. Problem statement

Several computer tools that make use of graphic statics have been developed [18–20]. However, until recently most of the tools and literature have only presented digital versions of graphic statics for some specific pre-defined structures. A general algebraic version of graphic statics using a graph theoretic approach is presented by Van Mele and Block in [21] that given a form diagram is able to generate a force diagram directly. A graph theoretic approach for self-stressed networks is also presented in [22]. The equilibrium of the structural systems in [21] are investigated using the states of self-stress of an equivalent unloaded network, making use of [23]. The algebraic approach of the graphic statics framework presented in [21] is well suited for computer implementation as a back end for a Computer Aided Design (CAD) software.

The ability to make changes in a general force diagram and subsequently compute a reciprocal form diagram has not been presented. The extension could be done in several different ways,

* Corresponding author.
E-mail address: vedad.alic@construction.lth.se (V. Alic).
URL: http://www.byggmek.lth.se/ (V. Alic).

http://dx.doi.org/10.1016/j.cad.2017.08.003
0010-4485/© 2017 Elsevier Ltd. All rights reserved.

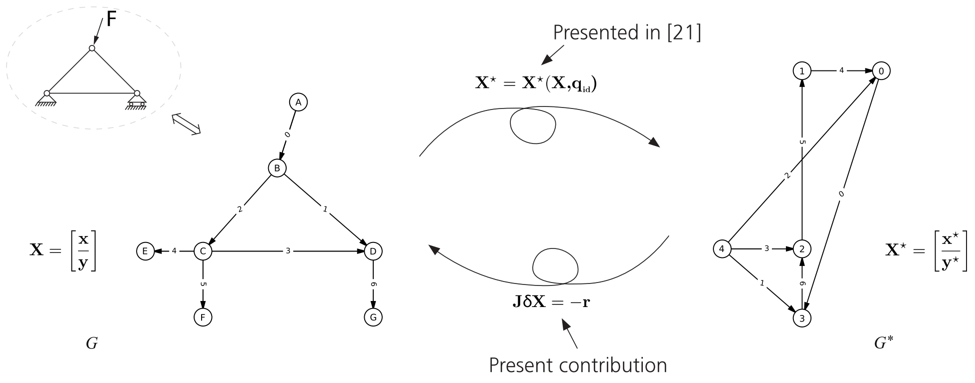
F
Presented in [21]
$\mathbf{X}^\star = \mathbf{X}^\star(\mathbf{X},\mathbf{q}_{id})$
$\mathbf{X} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$
$\mathbf{X}^\star = \begin{bmatrix}\mathbf{x}^\star\\ \mathbf{y}^\star\end{bmatrix}$
$\mathbf{J}\delta\mathbf{X} = -\mathbf{r}$
Present contribution
G
G*

Fig. 1. Schematic illustration of the extension of the framework presented in [21] by also allowing for interactive manipulations of the force diagram.

and two of those are to formulate the extension as a constrained optimization problem (suggested in [21]), or to find the roots of a set of non-linear equations. The challenges with the former are constructing an objective function and avoiding local minima, and the risks of the latter are the method might fail to find solutions. In this paper we present an approach based on the root finding for a set of non-linear equations, which we solve using Newton's method.

1.2. Objective

The objective of this paper is to extend the framework presented in [21] to a bi-directional approach. Given a change in the force diagram, our objective is to compute and present the updated form diagram, see Fig. 1, and to present how this can be implemented as a computational tool to enable manipulations in both the form diagram and the force diagram, with continuous interactive updates of the reciprocal diagrams. Further, we present the benefits and applications of our approach using examples.

1.3. Outline

Section 2 reviews algebraic graphic statics and presents a method to compute the form diagram from a given force diagram. It is formulated as a root finding procedure for a set of non-linear equations, and the required derivatives for Newton's method are presented.

Section 3 presents implementation aspects of the method and presents how the implementation can be used as a back-end for a design application. A general overview of the implementation and the requirement of further geometric constraints are presented. An approach on how to identify appropriate constraints by making use of the null space is presented. Further, vertex perturbations and line searches are introduced to improve the robustness of the solver.

Section 4 demonstrates the method in a set of examples. The examples increase in complexity, and are presented in the following order: a simple truss, a practical arch example with constraints, a form finding arch example, a study of the null space of an arch, a truss with constant force in the chords, and finally an externally post-tensioned funicular geometry.

2. Theoretical framework

This section starts with a brief review of graphic statics and the algebraic method which is presented in [21]. The section then continues with the extension of the method to enable manipulation of the force diagram, and to compute the equivalent form diagram by solving a set of non-linear equations using Newton's method.

2.1. Graphic statics

Fig. 1 presents a simple graphic statics example, with the structure, applied loads, and support forces to the left and the reciprocal force diagram to the right. The equivalent structural mechanics diagram of the form diagram is shown in the top left of the figure. Equilibrium of forces in each internal point in the form diagram (points B, C, D in Fig. 1) is represented by closed polygons in the force diagram. The lengths of the edges in the force diagram are proportional to the magnitude of the forces in the corresponding edges in the form diagram. The interpretation of tension and compression can be done by making use of Bow's notation [24].

2.2. Reciprocal graphs

The diagrams in Fig. 1 are represented as directed graphs. The diagrams are projections of dual polyhedra, where the rules of 3D projective geometry apply [10]. Vertices in the form diagram map to faces in the force diagram and faces in the form diagram map to vertices in the force diagram. Edges map to edges, and in the convention adopted here, remain parallel (another option is for edges to map to perpendicular edges).

In Fig. 1 the form diagram G is shown to the left, and the force diagram $G^\star$ is to the right. In graph theory these are referred to as the primal, and dual graphs, respectively. The connectivity matrix $\mathbf{C}$ of graph G and incidence matrix $\mathbf{C}^\star$ of the reciprocal graph $G^\star$ are constructed as described in [21], and are also briefly presented here. The connectivity matrix is

$$C_{ij} = \begin{cases} +1, & \text{if vertex } i \text{ is the head of edge } j \\ -1, & \text{if vertex } i \text{ is the tail of edge } j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

and the rows of the incidence matrix are constructed from the form graph by cycling through its faces in a counter-clockwise direction. The ith row of $\mathbf{C}^\star$, corresponding to the ith face of G is

$$C^\star_{ij} = \begin{cases} +1, & \text{if edge } i \text{ is traversed in the same direction as its orientation } j \\ -1, & \text{if in the opposite direction} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The size of $\mathbf{C}$ is $[v \times e]$, and the size of $\mathbf{C}^\star$ is $[v^\star \times e]$. Here, e is the total number of edges, v is the number of vertices in the form diagram and $v^\star$ is the number of vertices in the force diagram. The number of faces f in the form diagram G is equal to the number of vertices $v^\star$ in the force diagram $G^\star$, and the opposite holds true since $v = f^\star$.

2.3. Computing the force diagram

This subsection describes the method for computing a force diagram from a form diagram, for further details, see [21]. The column matrices containing the respective coordinates of the dual and the primal graphs are

$$\mathbf{X}^\star = \left[\frac{\mathbf{x}^\star}{\mathbf{y}^\star}\right]; \qquad \mathbf{X} = \left[\frac{\mathbf{x}}{\mathbf{y}}\right]. \tag{3}$$

These are of size $[2v^\star \times 1]$ and $[2v \times 1]$, respectively.

Given a form graph, the related force graph can be computed from $\mathbf{X}^\star = \mathbf{X}^\star(\mathbf{X}, \mathbf{q}_{\mathrm{id}})$, which is written explicitly

$$\mathbf{X}^\star = \left[\frac{\mathbf{L}^{\star-1}\mathbf{C}^\star\mathbf{Q}\mathbf{u}}{\mathbf{L}^{\star-1}\mathbf{C}^\star\mathbf{Q}\mathbf{v}}\right] \tag{4}$$

where $\mathbf{Q} = \mathbf{Q}(\mathbf{x}, \mathbf{y})$ is a diagonal matrix containing force densities [25], Q_{ii} is the force density of edge i. $\mathbf{L}^\star$ is the Laplacian matrix of the force graph, $\mathbf{L}^\star = \mathbf{C}^\star\mathbf{C}^{\star\mathrm{T}}$. The coordinate difference vectors of the form diagram are $\mathbf{u} = \mathbf{C}^{\mathrm{T}}\mathbf{x}$ and $\mathbf{v} = \mathbf{C}^{\mathrm{T}}\mathbf{y}$. The coordinates for one vertex in the force diagram in Eq. (4) must be chosen in order to obtain a unique solution, [21]. To obtain a unique solution it is further required to prescribe a set of force densities, $\mathbf{q}_{\mathrm{id}}$, for details see [21]. It is evident that $\mathbf{X}^\star$ is a non-linear function of $\mathbf{X}$, since both $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{Q}$ depend on $\mathbf{X}$.

2.4. Computing the reciprocal form diagram

Here we extend the method presented in [21], by solving for the form diagram, given changes in the force diagram. We formulate the problem as a root finding procedure by setting up a residual, which is zero at a root. To find the roots we employ Newton's method, for which the Jacobian matrix is required.

When the force diagram $\mathbf{X}^\star$ is manipulated due to a user input $\Delta\mathbf{X}^\star_{\mathrm{user}}$ we need to find the coordinates of the form diagram $\mathbf{X}$ that give the updated $\mathbf{X}^\star$

$$\mathbf{X}^\star(\mathbf{X}, \mathbf{q}_{\mathrm{id}}) = \mathbf{X}^\star_{\mathrm{start}} + \Delta\mathbf{X}^\star_{\mathrm{user}} \tag{5}$$

where $\mathbf{X}^\star_{\mathrm{start}}$ are the coordinates of the force diagram prior to the user manipulation. To find a solution we seek to find a root $\mathbf{X}$ such that

$$\mathbf{r} = \mathbf{X}^\star - (\mathbf{X}^\star_{\mathrm{start}} + \Delta\mathbf{X}^\star_{\mathrm{user}}) = \mathbf{0} \tag{6}$$

where $\mathbf{r}$ is a residual. We proceed by linearizing the equation and using an iterative method

$$\mathbf{r}(\mathbf{X}^i) + \frac{\partial \mathbf{r}(\mathbf{X}^i)}{\partial \mathbf{X}}\boldsymbol{\delta}X = \mathbf{0} \tag{7}$$

where i is the current iteration, not to be confused with exponentiation. $\mathbf{X}^\star_{\mathrm{start}}$ and $\Delta\mathbf{X}^\star_{\mathrm{user}}$ are constants, and $\mathbf{q}_{\mathrm{id}}$ has been kept constant, giving

$$\mathbf{r}(\mathbf{X}^i) + \frac{\partial \mathbf{X}^\star(\mathbf{X}^i)}{\partial \mathbf{X}}\boldsymbol{\delta}X = \mathbf{0} \tag{8}$$

where

$$\boldsymbol{\delta}X = \mathbf{X}^{i+1} - \mathbf{X}^i. \tag{9}$$

The Jacobian matrix of $\mathbf{X}^\star$ with respect to $\mathbf{X}$ is

$$\mathbf{J} = \frac{\partial \mathbf{X}^\star}{\partial \mathbf{X}} \tag{10}$$

which is of size $[2v^\star \times 2v]$. The Jacobian matrix is used to solve for $\boldsymbol{\delta}X$ in Eq. (8).

2.5. Derivation of the Jacobian matrix

The kth column of the Jacobian is computed as

$$\frac{\partial \mathbf{X}^\star}{\partial X_k} = \left[\frac{\mathbf{L}^{\star-1}\mathbf{C}^\star\left(\frac{\partial \mathbf{Q}}{\partial X_k}\mathbf{u} + \mathbf{Q}\mathbf{C}^{\mathrm{T}}\frac{\partial \mathbf{x}}{\partial X_k}\right)}{\mathbf{L}^{\star-1}\mathbf{C}^\star\left(\frac{\partial \mathbf{Q}}{\partial X_k}\mathbf{v} + \mathbf{Q}\mathbf{C}^{\mathrm{T}}\frac{\partial \mathbf{y}}{\partial X_k}\right)}\right] \tag{11}$$

where the derivatives of $\mathbf{x}$ and $\mathbf{y}$ are trivial. In order to compute the derivatives of the force densities, $\mathbf{Q}$ [25] we introduce, $\mathbf{A}$, the equilibrium matrix [26]

$$\mathbf{A} = \left[\frac{\mathbf{C}_{\mathrm{i}}\mathbf{U}}{\mathbf{C}_{\mathrm{i}}\mathbf{V}}\right] \tag{12}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the diagonal matrices of $\mathbf{u}$ and $\mathbf{v}$, respectively, and $\mathbf{C}_{\mathrm{i}}$ contains the rows of $\mathbf{C}$ corresponding to the inner vertices of G.

For the force densities, we first write them in their vector form $\mathbf{q}$ and partition them into an independent part, $\mathbf{q}_{\mathrm{id}}$, and a rest part, $\mathbf{q}_{\mathrm{d}}$. The equilibrium matrix is similarly partitioned in an independent, $\mathbf{A}_{\mathrm{id}}$, and a dependent part, $\mathbf{A}_{\mathrm{d}}$, together satisfying the following equation [21]:

$$\left[\mathbf{A}_{\mathrm{id}} \mid \mathbf{A}_{\mathrm{d}}\right]\left[\frac{\mathbf{q}_{\mathrm{id}}}{\mathbf{q}_{\mathrm{d}}}\right] = \mathbf{0}. \tag{13}$$

The independent edges can be identified by transforming $\mathbf{A}$ into a row reduced echelon form [21]. The number of independent edges amounts to the number of associated force densities that must be prescribed.

From Eq. (13) we solve for the force densities:

$$\mathbf{q} = \left[\frac{\mathbf{q}_{\mathrm{id}}}{\mathbf{A}_{\mathrm{d}}^{-1}(-\mathbf{A}_{\mathrm{id}}\mathbf{q}_{\mathrm{id}})}\right]. \tag{14}$$

The derivative of $\mathbf{q}$, after partitioning, is

$$\frac{\partial \mathbf{q}}{\partial X_k} = \left[\frac{\mathbf{0}}{\frac{\partial \mathbf{A}_{\mathrm{d}}^{-1}}{\partial X_k}(-\mathbf{A}_{\mathrm{id}}\mathbf{q}_{\mathrm{id}}) + \mathbf{A}_{\mathrm{d}}^{-1}(-\frac{\partial \mathbf{A}_{\mathrm{id}}}{\partial X_k}\mathbf{q}_{\mathrm{id}})}\right]. \tag{15}$$

From Eq. (15) it is evident that the derivative of the independent part, $\mathbf{q}_{\mathrm{id}}$ is zero, since the independent force densities remain unchanged during manipulations of the force diagram. To compute the derivatives $\mathbf{A}$, Eq. (12) is differentiated with respect to $\mathbf{X}$.

$$\frac{\partial \mathbf{A}}{\partial X_k} = \left[\frac{\mathbf{C}_{\mathrm{i}}\frac{\partial \mathbf{U}}{\partial X_k}}{\mathbf{C}_{\mathrm{i}}\frac{\partial \mathbf{V}}{\partial X_k}}\right]. \tag{16}$$

From $\mathbf{u} = \mathbf{C}^{\mathrm{T}}\mathbf{x}$ it follows that

$$\frac{\partial \mathbf{U}}{\partial X_k} = \begin{cases}\mathrm{diag}(\mathbf{c}_k), & \text{if } X_k \in \mathbf{x}\\ \mathbf{0}, & \text{if } X_k \in \mathbf{y}\end{cases} \tag{17}$$

and similarly

$$\frac{\partial \mathbf{V}}{\partial X_k} = \begin{cases}\mathbf{0}, & \text{if } X_k \in \mathbf{x}\\ \mathrm{diag}(\mathbf{c}_k), & \text{if } X_k \in \mathbf{y}\end{cases} \tag{18}$$

where $\mathbf{c}_k$ is the kth row in $\mathbf{C}$

$$\mathbf{C} = \left[\mathbf{c}_1^{\mathrm{T}} \cdots \mathbf{c}_k^{\mathrm{T}} \cdots \mathbf{c}_v^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{19}$$

To complete this section $\partial\mathbf{A}/\partial X_k$ is partitioned into a dependent, $\partial\mathbf{A}_{\mathrm{d}}/\partial X_k$ and an independent part $\partial\mathbf{A}_{\mathrm{id}}/\partial X_k$, and finally the derivative of the inverse of the dependent part required in Eq. (15) is

$$\frac{\partial \mathbf{A}_{\mathrm{d}}^{-1}}{\partial X_k} = -\mathbf{A}_{\mathrm{d}}^{-1}\frac{\partial \mathbf{A}_{\mathrm{d}}}{\partial X_k}\mathbf{A}_{\mathrm{d}}^{-1} \tag{20}$$

which is the last part required to compute the Jacobian matrix.

2.6. Jacobian matrix with respect to force densities

In this section we present the derivatives $\partial \mathbf{X}^{\star}/\partial \mathbf{q}_{\mathrm{id}}$ for completeness. The following derivatives could be used to relax the conditions on edges with prescribed force densities. In the implementation and the examples we keep these fixed in order to simplify the presentation.

The derivatives of $\mathbf{X}^{\star}$ with respect to the jth prescribed force density are [25]

$$\frac{\partial \mathbf{X}^{\star}}{\partial q_{\mathrm{id},j}} = \begin{bmatrix} \mathbf{L}^{\star-1}\mathbf{C}^{\star}\frac{\partial \mathbf{Q}}{\partial q_{\mathrm{id},j}}\mathbf{u} \\ \hline \mathbf{L}^{\star-1}\mathbf{C}^{\star}\frac{\partial \mathbf{Q}}{\partial q_{\mathrm{id},j}}\mathbf{v} \end{bmatrix} \tag{21}$$

where the column matrix version of $\frac{\partial \mathbf{Q}}{\partial q_{\mathrm{id},j}}$ is

$$\frac{\partial \mathbf{q}}{\partial q_{\mathrm{id},j}} = \begin{bmatrix} \frac{\partial \mathbf{q}_{\mathrm{id}}}{\partial q_{\mathrm{id},j}} \\ \hline \mathbf{A}_{\mathrm{d}}^{-1}\left(-\mathbf{A}_{\mathrm{id}}\frac{\partial \mathbf{q}_{\mathrm{id}}}{\partial q_{\mathrm{id},j}}\right) \end{bmatrix}. \tag{22}$$

Finally, $\frac{\partial \mathbf{q}_{\mathrm{id}}}{\partial q_{\mathrm{id},j}}$ has a one at the jth element, and zeros elsewhere

$$\frac{\partial \mathbf{q}_{\mathrm{id}}}{\partial q_{\mathrm{id},j}} = \begin{bmatrix} \frac{\partial q_{\mathrm{id},1}}{\partial q_{\mathrm{id},j}} \\ \vdots \\ \frac{\partial q_{\mathrm{id},j-1}}{\partial q_{\mathrm{id},j}} \\ \frac{\partial q_{\mathrm{id},j}}{\partial q_{\mathrm{id},j}} \\ \frac{\partial q_{\mathrm{id},j+1}}{\partial q_{\mathrm{id},j}} \\ \vdots \\ \frac{\partial q_{\mathrm{id},n}}{\partial q_{\mathrm{id},j}} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{23}$$

2.7. Inextensible mechanisms

The number of static and kinematic indeterminacies in a pin-jointed structure can be determined from Maxwell's rule. The extended Maxwell rule for a form graph is

$$k - m = e - 2v_i \tag{24}$$

where k is the number of independent states of self-stress, m is the number of inextensible mechanisms, and v_i is the number of internal vertices of G. By using a singular value decomposition (SVD) of the equilibrium matrix $\mathbf{A}$, the values of k and m can be determined [26,21].

Eq. (20) can only be determined if the number of inextensible mechanisms, m, is zero. When $m = 0$, Eq. (13) can be rewritten as [21]

$$\mathbf{q}_{\mathrm{d}} = -\mathbf{A}_{\mathrm{d}}^{-1}\mathbf{A}_{\mathrm{id}}\mathbf{q}_{\mathrm{id}}. \tag{25}$$

However, in the case when $m > 0$, then $\mathbf{A}_{\mathrm{d}}$ is not square and the remaining force densities are solved from the equivalent equation

$$\mathbf{A}_{\mathrm{d}}^{\mathrm{t}}\mathbf{A}_{\mathrm{d}}\mathbf{q}_{\mathrm{d}} = -\mathbf{A}_{\mathrm{d}}^{\mathrm{t}}\mathbf{A}_{\mathrm{id}}\mathbf{q}_{\mathrm{id}} \tag{26}$$

as described in [21].

In our method, when $m > 0$ for the equivalent of Eq. (15) we must compute the derivatives of the Moore–Penrose pseudoinverse $(\mathbf{A}_{\mathrm{d}}^{\mathrm{T}}\mathbf{A}_{\mathrm{d}})^{-1}\mathbf{A}_{\mathrm{d}}^{\mathrm{T}}$, since $\mathbf{A}_{\mathrm{d}}$ is not square. It is possible to calculate the derivatives if the rank is unchanged, however, very small perturbations of the shape will in general lead to a reduction of the number of inextensible mechanisms (m), and thus to a change of rank. Manipulations which keep the rank constant can be explored by selecting different states of self-stress, by selecting a new $\mathbf{q}_{\mathrm{id}}$, and thus generating a new force diagram.

Algorithm 1 Iterative solution loop

```
Get initial solution and store it:
$\mathbf{X}^0 = \mathbf{X}_{\mathrm{start}}$
$\mathbf{X}^{\star}_{\mathrm{start}} = \mathbf{X}^{\star}(\mathbf{X}^0)$

Get user change in force diagram, $\Delta\mathbf{X}^{\star}_{\mathrm{user}}$

while $||\mathbf{r}||_2 > \epsilon$ do
    Update:
    $\mathbf{J}(\mathbf{X}^i)$
    $\mathbf{r}(\mathbf{X}^i) = \mathbf{X}^{\star}(\mathbf{X}^i) - (\mathbf{X}^{\star}_{\mathrm{start}} + \Delta\mathbf{X}^{\star}_{\mathrm{user}})$

    Remove rows in $\mathbf{J}$ and $\mathbf{r}$ due to the
    constrained vertex in the force diagram.

    Solve for $\delta\mathbf{X}$ from:
    $\mathbf{J}(\mathbf{X}^i)\delta\mathbf{X} = -\mathbf{r}(\mathbf{X}^i)$

    Update $\mathbf{X}$:
    $\mathbf{X}^{i+1} = \mathbf{X}^i + \delta\mathbf{X}$

    update $\mathbf{X}^{\star}(\mathbf{X}^{i+1})$

    update $||\mathbf{r}||_2$
    increment $i$
end while
```

2.8. The solution loop

To find the coordinates of a form diagram that is reciprocal to the force diagram, the Jacobian matrix is used. The solution is implemented in an iterative scheme, employing Newton's method, as shown in Algorithm 1.

Generally, the Jacobian matrix is not a square and represents an underdetermined system. To ensure that at least one solution exists, the rank of the Jacobian matrix should be equal to the rank of the augmented matrix

$$rank(\mathbf{J}) \geq rank(\mathbf{J}\,|\,\mathbf{r}). \tag{27}$$

Further, if the rank of the Jacobian is less then the number of variables, then $2v - rank(\mathbf{J})$ linearly independent solutions exist. If this occurs, we choose a single solution by making use of the least squares principle [25].

The effects of prescribing a vertex's coordinates in the force diagram in Eq. (4) to zero means that two rows in the Jacobian will be zero. These rows, along with the corresponding rows of $\mathbf{r}$ can be reduced, resulting in a Jacobian of size $[(2v^{\star} - 2) \times 2v]$, and $\mathbf{r}$ of size $[(2v^{\star} - 2) \times 1]$. Applying manipulations to the position of the prescribed vertex in the force diagram can be achieved by applying the opposite manipulation to all other vertices.

2.9. Additional requirements

A requirement of the form graph is that it must be a planar graph [27,21]. A further requirement of the method is that edges that have a prescribed force density cannot have zero length in either of the reciprocal diagrams as this leads to division by zero.

3. Computer implementation

In this section a general overview of the implementation of the presented method is provided. It also indicates that the presented method can be used as a back-end in an interactive graphic statics application where the user can manipulate both form and

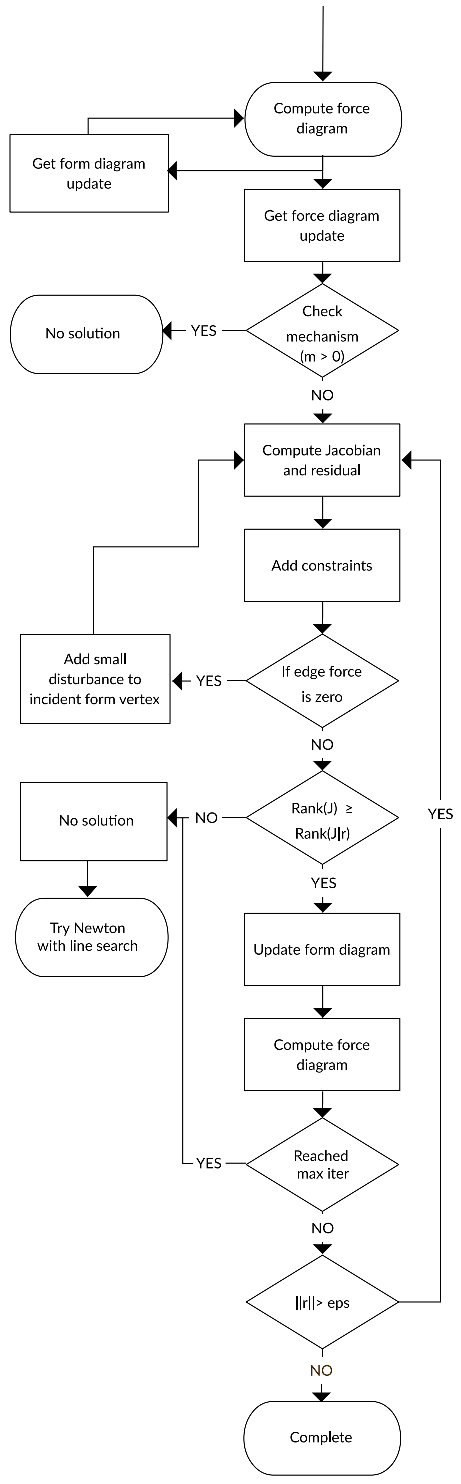

Fig. 2. Flow chart of implementation.

force diagrams with continuous bi-directional updates. A general overview of how the method is implemented is shown in Fig. 2, and various checks are performed to ensure that the user's manipulations to the force diagram are valid and that the problem has one or multiple solutions. The implementation of the method is described in more detail in Algorithm 1.

Our implementation has been made by using the programming language Python [28], together with Numpy [29] and Scipy [30] for performing matrix operations. The Python package NetworkX [31] has been used for graph operations and visualizations.

3.1. *Adding geometric constraints to form diagrams*

It is necessary to add additional geometric constraints for certain types of structures, e.g. to keep the span of a truss constant. This can be achieved through adding derivatives of constraint equations to the Jacobian matrix and changes to the constraint equation to the residual, i.e.

$$\left[\frac{\mathbf{J}}{\mathbf{J}_c}\right]\delta\mathbf{X} = \left[\frac{\mathbf{r}}{r_c}\right] \tag{28}$$

where

$$J_{c,k} = \frac{\partial f(\mathbf{X})}{\partial X_k}; \quad r_c = \delta f(\mathbf{X}) \tag{29}$$

where $f(\mathbf{X})$ is the added constraint equation to the cth row of the Jacobian matrix and X_k is the kth coordinate, corresponding to the kth column of the Jacobian matrix.

In the present work the following constraints have been implemented:

- Horizontal length between two vertices
- Vertical length between two vertices
- Length between two vertices
- Direction between two vertices
- Horizontal position of a vertex
- Vertical position of a vertex

In general, there are a few constraints that always should be applied. Rigid body motions of the form diagram will produce the same force diagram, and as such are of no interest. These can be removed by constraining the position of a vertex in the form diagram. An example is shown in Fig. 3a. It is possible to find solutions, i.e. form diagrams, that results in the same force diagram by scaling of the form diagram, as illustrated in Fig. 3b. In order to remove solutions that only scale the form diagram constraints can be introduced. It is often sufficient to constrain the horizontal length between the supports. Finally, there is a third type of constraint which in our implementation is automatically introduced. It is related to edges which connect to leaf vertices in the form diagram. A leaf vertex is a vertex that only has one edge connected to it. If one such edge does not have a prescribed force-density, it will result in the same force diagram regardless of its length in the form diagram. An example is presented in Fig. 3c. The change in lengths of these edges is of no interest and as such we constrain the lengths of all free edges.

3.2. *Null space of the Jacobian matrix*

If $2v > rank(\mathbf{J})$, then $2v - rank(\mathbf{J})$ linearly independent solutions, of form diagrams, exist. After a form diagram is found from a specific force diagram, the null space of the Jacobian is a vector space with dimension $2v - rank(\mathbf{J})$, which shows how the form diagram can be changed without affecting the force diagram. This allows the user to explore alternative form diagrams for a specific force diagram. The null space can also be studied to find suitable constraints in order to obtain more desirable solutions.

The null space of the Jacobian can be found by computing the SVD [23] or by using a rank-revealing QR factorization [32].

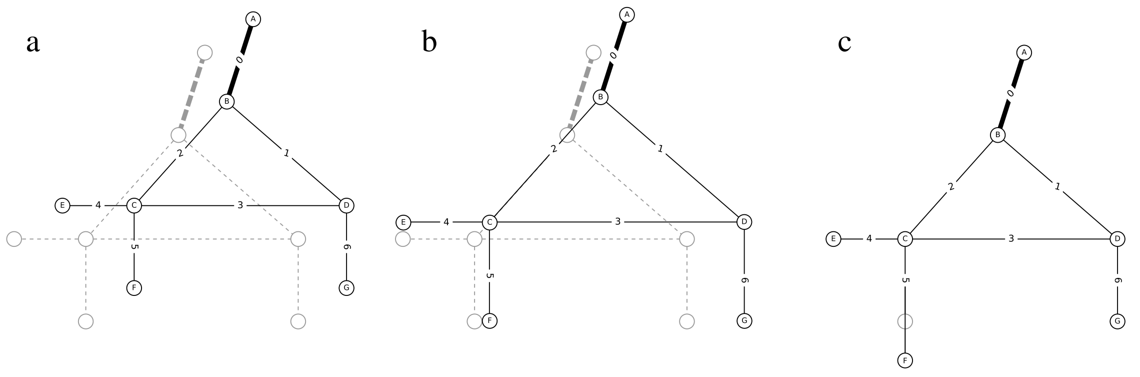

Fig. 3. Examples of changes in the form diagram that result in the same force diagram. (a) Rigid body motion. (b) Scaling motion, note that edge zero does not scale because of the prescribed force density. (c) Extension of an edge connected to a leaf vertex without prescribed force density.

3.3. *Vertex perturbation*

If the length of an edge in the force diagram is zero the reciprocal structural member obviously carries no force. If this is the case, that edge will not have any contribution to the Jacobian matrix, i.e. the corresponding derivatives are zeros. To solve this numerical problem, zero length edges in the force diagram are identified in each iteration. Zero length edges can easily be identified since their corresponding force densities in the vector **q** are zero. If such edges exist, a very small random value is added to the length of that zero edge by moving the incident vertices in the form diagram. For the movement we successfully use a pair of (for x- and y-coordinate) random numbers from a normal distribution, with zero mean and a standard deviation of 10^{-6}, $\mathcal{N}(0, 10^{-12})$. The implementation of the Newton method with the vertex perturbation is presented in Fig. 2.

3.4. *Large manipulations*

In order for the proposed method to work, an initial form diagram is required. The initial form is used as an initial guess for Newton's method. As Newton's method is a gradient based method it converges towards local extreme points. This is an advantageous property in case multiple solutions exists, as solutions close to the initial form are often preferred. When the initial guess is too far from the root Newton's method may not converge. For this case the method is modified to include a line search. For details on how to implement the line search, see [33]. The line search is used on iterations where the norm of the residual has increased compared to the previous iteration. The Jacobian matrix is modified with vertex perturbations even for the Newton method with line search, the modification is performed in the same way as presented in Fig. 2. In Section 4.2 we show why we try the method without line search first.

The non-linearity of the problem can sometimes lead to unexpected results, for instance, some small manipulations in the force diagram can result in large changes of the form diagram. This problem can be observed when a vertex that is connected to a short edge in the force diagram is manipulated. A small manipulation in distance can then result in a large change in direction of the adjacent edge. An example is provided in Fig. 4, where vertex 2 is moved in the vertical direction. Even though the method manages to find solutions, these solutions might sometimes become too distorted. In such cases it is, in a computer implementation, possible to revert to a previous state and continue exploring the design from that state.

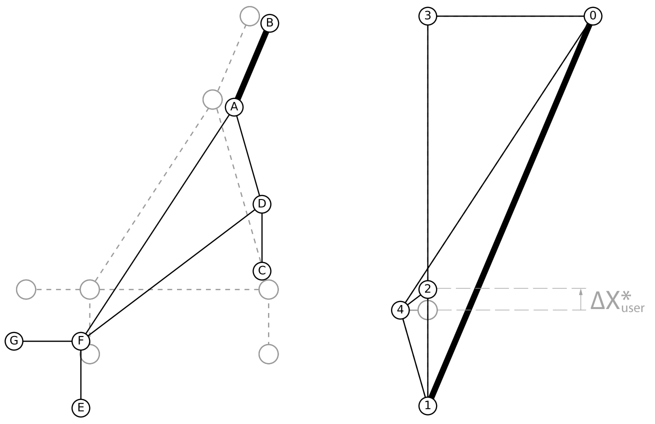

Fig. 4. A small manipulation of the force diagram leads to a large change in form.

4. Examples

In this section we present the method with a few practical examples. These emphasize the strengths of the possibility of manipulating the force diagram. The examples also show the importance of selecting proper constraints to obtain the desired solutions, for instance, a common design criteria can be used to span a certain length, this can be achieved by adding constraints that keep the positions of where the structure meets the supports fixed. The initial form and force diagrams in the following examples are visualized with gray dashed lines.

4.1. *Simple example*

A simple example is shown in Fig. 5. Vertex four in the force diagram is moved to the left, and the corresponding change in the form diagram is visualized. Only the edges that are connected to vertex four in the force diagram are affected.

In the following paragraphs we constrain the positions of vertices C and D of Fig. 5. These constraints keep the position, length, and direction of edge 3 in the form diagram fixed, thereby imposing limitations on the allowable manipulations of the force diagram. The manipulations of the force diagram must be such that the direction of edge 3 in the force diagram is unchanged. For instance, moving vertex 4, or vertex 2 in the vertical direction would change the direction of edge 3 and is thus not possible. However, moving both vertex 4 and 2 vertically by the same amount is allowed, since the direction of edge 3 is then constant during the manipulation.

In the implementation, information regarding inadmissible manipulations of parameters in the force diagram can be found by

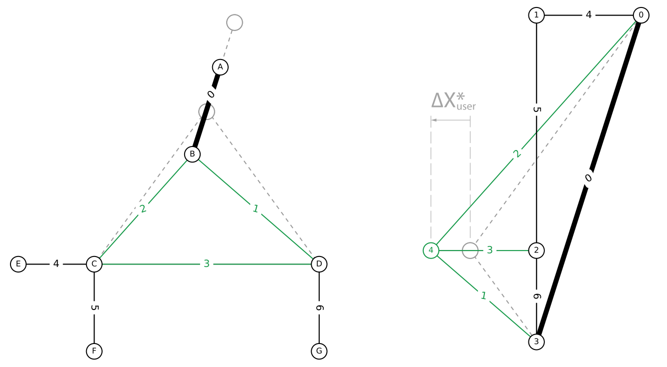

Fig. 5. Simple example where vertex 4 in the force diagram has been moved to the left, the corresponding change is visible in the form diagram.

computing the left null space of the Jacobian matrix. The left null space complements the column space of the Jacobian matrix, and any force manipulations, i.e. any **r**, that is not perpendicular to the left null space is invalid. To further clarify, we compute a basis for the left null space (consisting of one base vector) for the structure in Fig. 5 with vertices C, D constrained. The vector is of size $[2v^\star - 2 + n_{\text{constraints}} \times 1]$ and after normalization with respect to $y^\star_{r2}$ is

$$\mathbf{n}^{\mathrm{T}} = \begin{array}{c} \\ \end{array}\left[\begin{array}{cccccc} x^\star_{r0} & \cdots & y^\star_{r1} & y^\star_{r2} & y^\star_{r3} & c \\ 0 & \cdots & 0 & 1 & 0 & \cdots \end{array}\right] \tag{30}$$

where the parts related to the constraints have been omitted. Manipulations of the force diagram will lead to residual vectors with non-zero elements only on the parts related to x and y, and will thus always be perpendicular to the constraints part. The residual vector associated with moving vertex 2 a unit displacement vertically is

$$\mathbf{r}_1^{\mathrm{T}} = \left[\begin{array}{cccccccc} & x^\star_{r3} & y^\star_{r0} & y^\star_{r1} & y^\star_{r2} & y^\star_{r3} & c_1 & \\ \cdots & 0 & 0 & 0 & -1 & 0 & 0 & \cdots \end{array}\right] \tag{31}$$

and the residual vector associated with moving vertex 4 a unit displacement vertically is

$$\mathbf{r}_2^{\mathrm{T}} = \left[\begin{array}{cccccccc} & x^\star_{r3} & y^\star_{r0} & y^\star_{r1} & y^\star_{r2} & y^\star_{r3} & c_1 & \\ \cdots & 0 & 1 & 1 & 1 & 1 & 0 & \cdots \end{array}\right] \tag{32}$$

where vertices 0–3 are moved in the opposite direction, since vertex 4 is constrained in the implementation and not part of the vector. This leads to

$$\mathbf{n}\cdot\mathbf{r}_1 \neq 0,\ \ \mathbf{n}\cdot\mathbf{r}_2 \neq 0,\ \ \mathbf{n}\cdot(\mathbf{r}_1+\mathbf{r}_2) = 0 \tag{33}$$

where the dot is the scalar product. It is seen that only the application of both movements is perpendicular to the left null space, and thus the only admissible manipulation of the three.

In general, the left null space can consist of more than one vector, in such cases it is not always possible to, in a simple way, identify the parameters that can be changed in the force diagram. For solutions to exist, the requirement is that **r** must be perpendicular to the left null space.

4.2. Arch—Practical example with constraints

In Fig. 6 an example is presented of how the form diagram is changed when manipulating the force diagram.

In this example, constraints are added to obtain a desired solution. The Jacobian matrix has the dimension $[2v^\star - 2 \times 2v]$, in this example $[18 \times 36]$. Before constraints are added, the Jacobian matrix has *rank* $=$ 18. Three constraints are added to remove rigid body motions (horizontal, vertical and scaling, applied by fixing A_x, A_y, and H_x). Three additional constraints are added to constrain the size of the boundary forces. Note that the length of edge zero cannot be constrained since the force density for this edge has already been prescribed. The size of the external forces and their horizontal positions are also constrained, adding 12 more constraints. In total 18 constraints are added. After these constraints have been added, the Jacobian has the dimension $[18 + 18, \times 36]$ and as the added constraints are linearly independent *rank* $= 36$, a unique solution now exists.

After these constraints have been added, the force diagram is manipulated. In this example, vertices 0–2 are moved to the right, making the internal forces (edge 1–7) and the boundary forces (edge 0 and 8) smaller. As seen in the figure, the resulting form diagram shows an increased height of the arch, as expected. Moving vertices 0–2 further to the right past the externally applied loads (edges 11–16) flip the arch upside-down into a cable in tension. This solution is found when using the approach presented in Algorithm 1. Trying to solve for the manipulation by using the globally convergent Newton method with line search fails as the arch tends towards infinite height as vertices 0–2 get closer to the externally applied loads (edges 11–16) from the left, which is why we in Fig. 2 try the Newton method without line search first. Fig. 7 presents the convergence of the two methods, where the Newton method with line search fails to converge, and is aborted on iteration 32 due to the Jacobian matrix becoming rank deficient.

Adding another constraint to the example, by fixing the vertical position of vertex H, leads to a left null space of dimension 1. The basis vector for the left null space is not as simple as in the

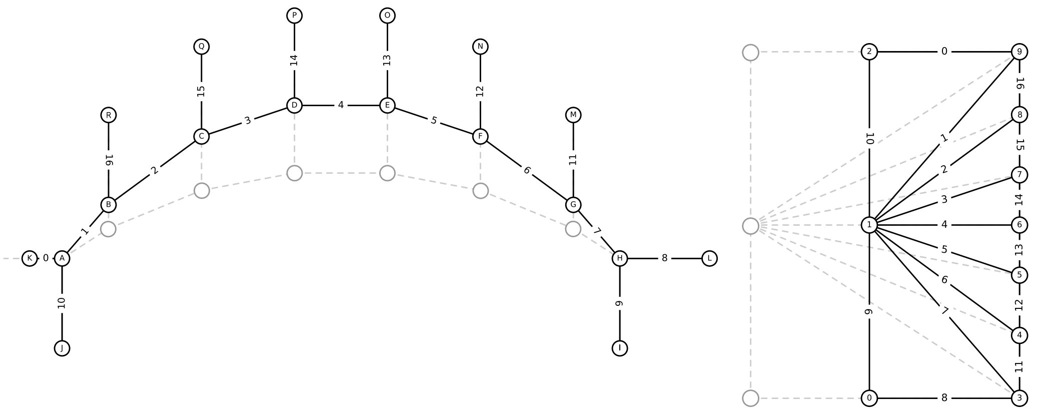

Fig. 6. Form finding for an arch, the dashed lines in the form diagram indicate the constrained horizontal positions.

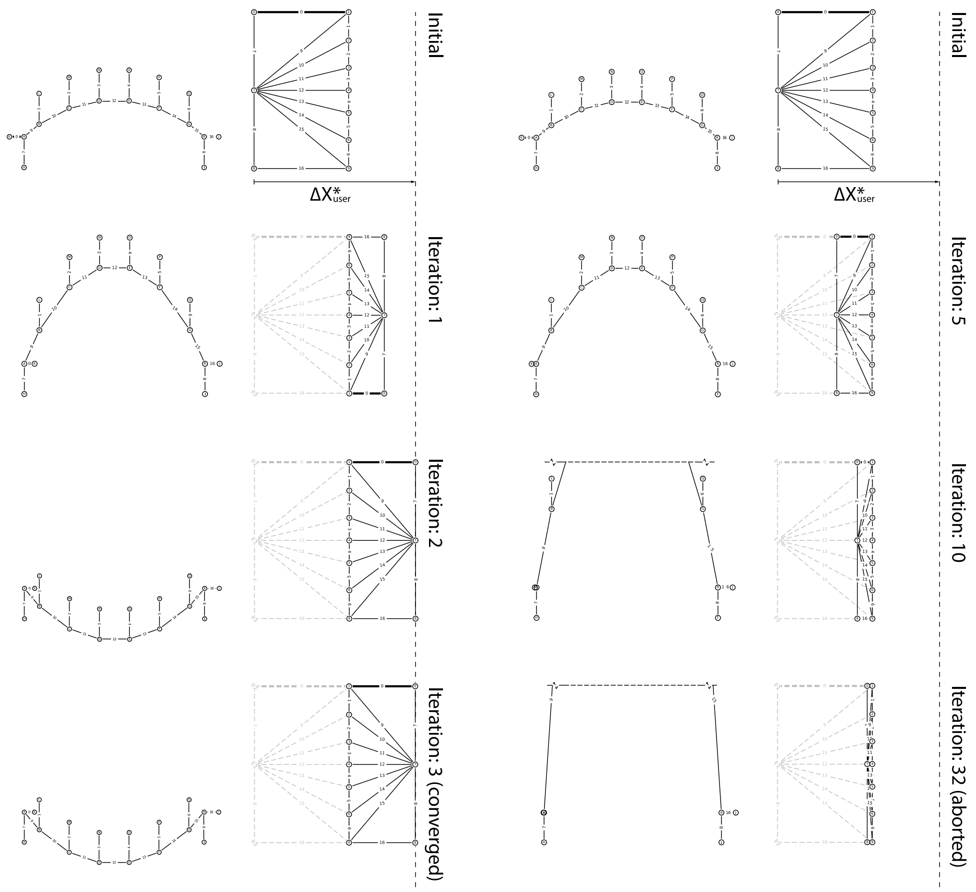

Fig. 7. Left side shows the convergence of Newton's method. Right side shows the convergence of Newton's method with line search, which fails to converge in this case. The form diagrams are all in the same scale, however, the force diagrams are not.

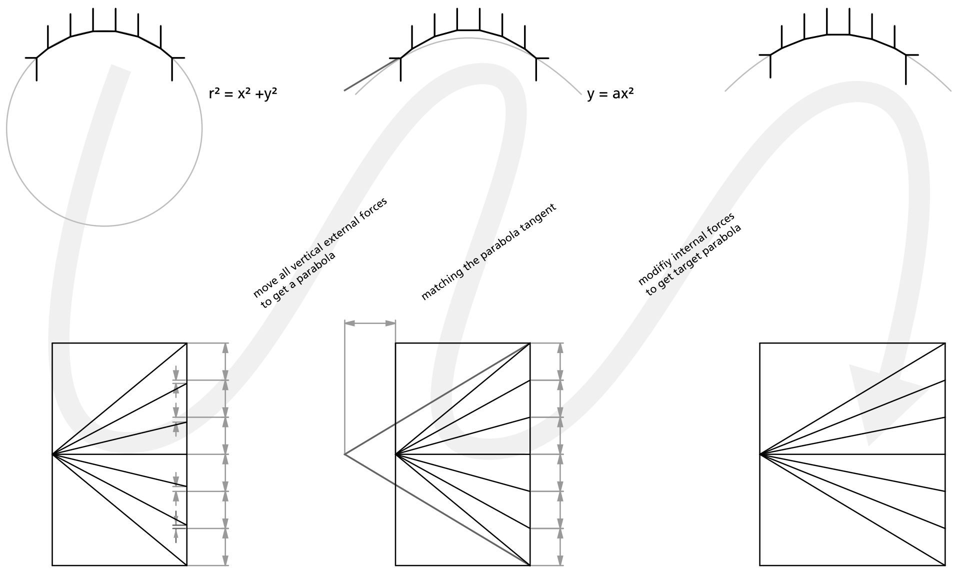
$r^2 = x^2 + y^2$
$y = ax^2$
move all vertical external forces to get a parabola
matching the parabola tangent
modify internal forces to get target parabola

Fig. 8. Form finding of an arch where the forces are manipulated to become uniformly distributed.

previous example, and restricts the possible manipulations of the force diagram parameters since they must be perpendicular to the basis vector. From studying the basis vector for the left null space, it can be seen that, for instance, vertices 4 and 8 have to be moved by the same amount horizontally, and by opposite amounts vertically. Introducing further constraints will increase the rank and number of basis vectors for the left null space. Finding an intuitive method for presenting possible manipulations of the force diagram would be good in such cases.

4.3. *Arch—Form finding*

An extension of the previous example is shown in Fig. 8, which indicates how this method can be used in the design process of

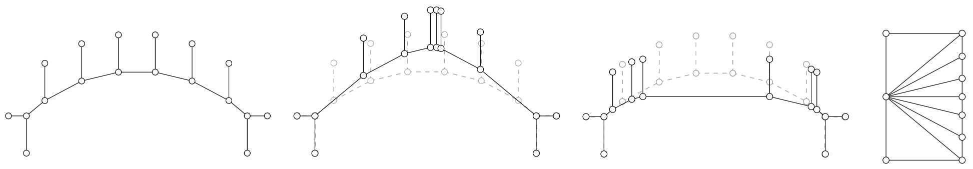
Fig. 9. All the forms in this figure result in the same force diagram that can be seen to the right.

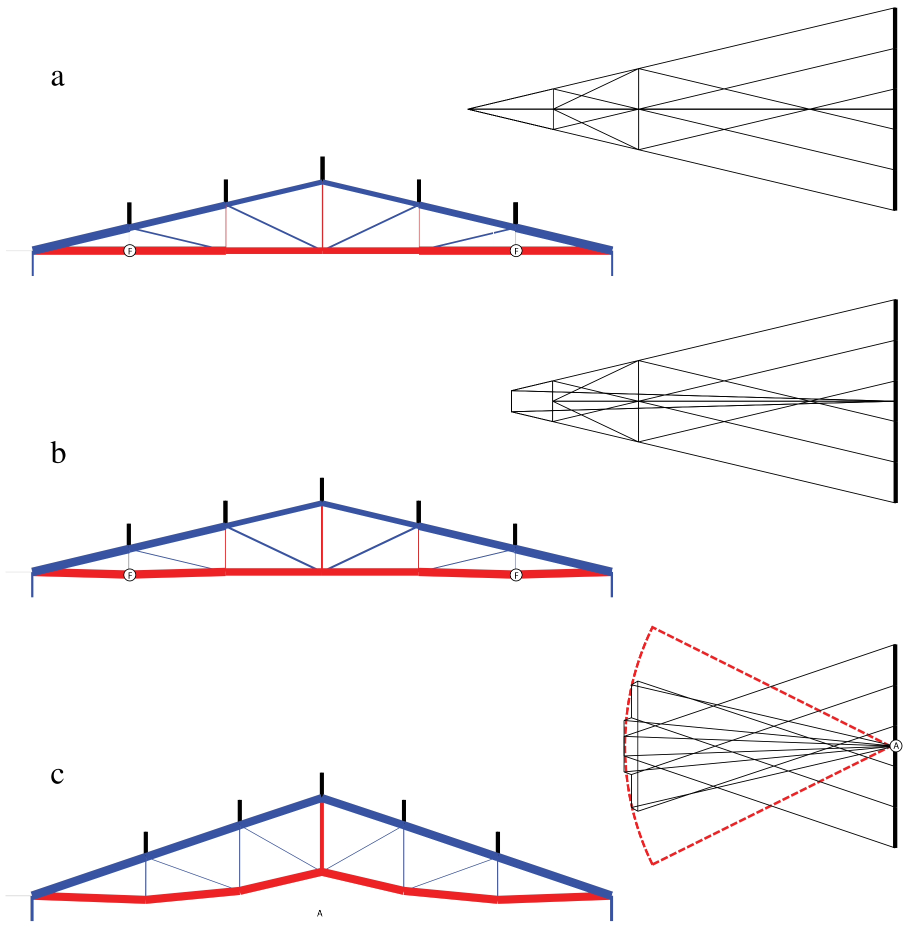

Fig. 10. (a) Gable truss with varying force magnitude in chords. (b) Vertices marked with F in the form diagram are translated slightly downwards, the forces in the connecting vertical edges are no longer zero. (c) Design of a constant-force gable truss.

a structure. The initial form in this example can be described as an arc of a circle. The external forces need to have different magnitudes in order for the structure to be in equilibrium. Using the same constraints for this example as in the previous example, the external forces are manipulated so that their magnitudes are equal. The resulting form after employing our method can, as expected, be described using a parabola, since the configuration of forces is a discrete representation of a uniformly distributed load, see Fig. 8b. A retained quality from the method of graphic statics is that edges in the force and form diagram have the same direction. Thus, to get a specific direction of an edge in the form diagram, the reciprocal edge in the force diagram can be manipulated. In the figure a sought tangent is drawn in the form diagram, then transferred and matched to the reciprocal edge in the force diagram, see Fig. 8c. The resulting form diagram now has the sought form.

4.4. Arch—Null space

In Fig. 9 multiple form diagrams that have the same force diagram is shown. The original form is the same as in the previous two examples, however, the horizontal constraints on the external force positions have been removed. By computing the null space of the Jacobian a vector space is obtained. Linear combinations of null space basis vectors can be used to explore different forms, all of which have the same force diagram. An example is shown in Fig. 9. In a potential application the free variables could be connected to a parametric modeler which would allow the user to easily explore different solutions.

In all of the form variations in Fig. 9 the reaction forces at the supports remain unchanged. The position of the externally applied forces varies with the form variations, however, the position of the line of action for the resultant of the applied forces remains unchanged, which is a requirement for the resulting force diagrams to be the same.

4.5. Chord force truss

Fig. 10 shows an example where the developed method is applied to the design of a truss with constant force in the top and bottom chords. All of the external loads are applied vertically. In

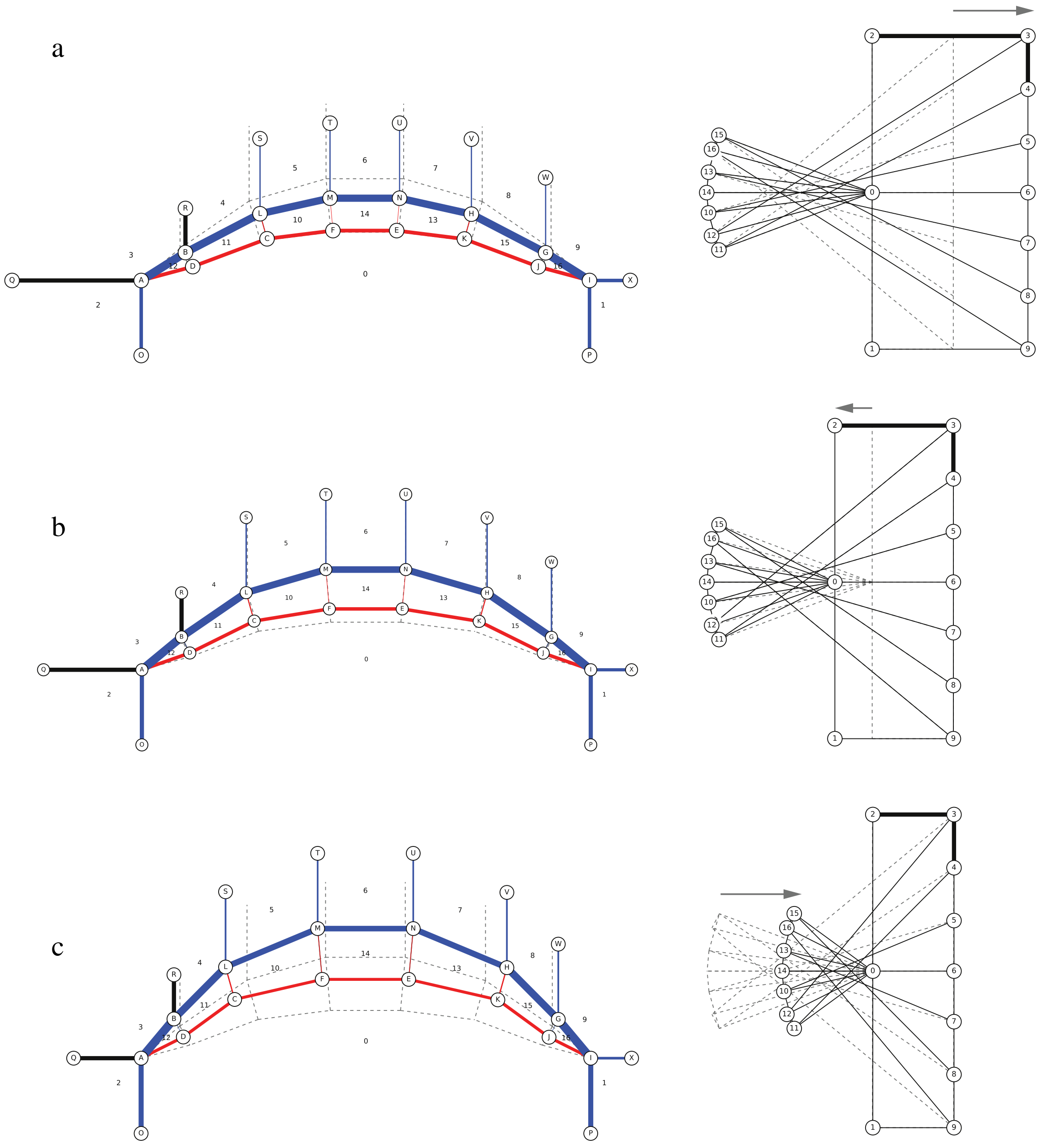

Fig. 11. Manipulations of the force diagram for a post-tensioned funicular structure. The gray dashed lines represent the starting geometry.

this example we have constrained rigid body motions, scaling, and lengths of free leaf edges.

Fig. 10a shows the starting position where the top and bottom chord edges do not carry the same amount of force in all edges. Edges in compression are shown in blue, edges in tension are shown in red in the form diagrams, and the thickness of the edge is proportional to the force.

Two edges in the initial form in Fig. 10a in the form diagram do not carry any load, and their bottom vertices are marked with (F). A small manipulation of adjacent vertices can lead to large changes in the force diagram, as shown in Fig. 10b. If the null space of the initial form is studied (Fig. 10a), it is seen that it contains null-modes where the zero force edges can rotate around their top vertices without affecting the force diagram. However, this is not present in the shape in Fig. 10b.

To obtain the form of a constant-force gable truss is fairly straight-forward. By manipulations in the force diagram of Fig. 10a we arrive at the example shown in Fig. 10c, where all the edges of the bottom chord carry the same amount of force. Since the bottom chord edges connect to the same vertex in the force diagram (marked with A in Fig. 10c), their opposite vertices are constrained to lie on a circle, shown in Fig. 10c. The diagonal elements inside of the truss carry a small amount of force compared to the other members of the structure. These could be made smaller in another iteration, but have been kept in the figure for clarity.

An example of such a truss is seen in the Magazzini Generali warehouse at Chiasso, Switzerland, designed by Robert Maillart [4]. The principle of restraining chord forces to the same length can be used to find other similar forms as well, see for instance [34].

4.6. Post tensioned funicular structure

Fig. 11 shows our method applied to the design of an externally post-tensioned funicular geometry. For further examples of such

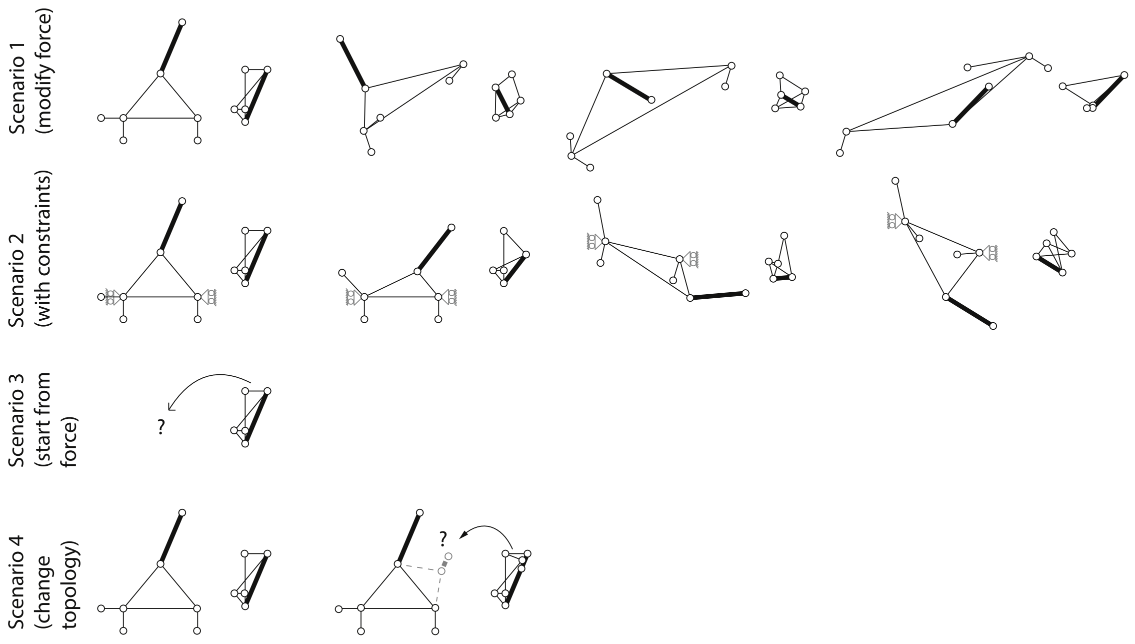
Scenario 1 (modify force)
Scenario 2 (with constraints)
Scenario 3 (start from force)
?
Scenario 4 (change topology)
?

Fig. 12. Force diagram manipulation exploration scenarios.

structures, see for instance [9]. Two force densities must be selected by the user to get a unique solution. In this case we have selected the force density for one of the horizontal supports (edge 0) and for one of the applied loads (edge 1). Vertex order 1, 0, 2–9, 1 in the force diagram represents the global equilibrium of the structure. The positions of vertices A and I in the form diagram are fully constrained.

Fig. 11a shows the horizontal translation of vertices 3–9 in the force diagram, resulting in the lowering of the upper chord of the form. Fig. 11b shows the horizontal translation of vertices 0–2 in the force diagram, resulting in the raising of the lower chord in the form diagram. By these manipulations the relationship between the form and the force diagram can easily be understood, for instance by lowering the effective structural height (the distance between the lower and upper chords) an increase in the magnitude of the forces in the horizontal supports (edges 0 and 29) is obtained.

Fig. 11c shows the horizontal translation of vertices 10–16 in the force diagram, resulting in smaller force magnitudes in the upper and lower chords. This is achieved by increasing the overall height of the post tensioned structure, and a slight decrease in the distance between the upper and lower chords.

In the example, the lines of action of the externally applied loads have not been fixed, and the corresponding Jacobian matrix is under-determined. Accordingly the truss has a non-trivial null space, and alternative forms for the same force diagram, similar to the example in Fig. 9.

5. Conclusions

In this paper, we propose a new method for finding a reciprocal form diagram after manipulations have been carried out on the force diagram. This has previously only been possible through using the geometrical relationship between form diagram and force diagram, either by drawing or in predefined examples that follow graphic statics procedures for specific geometries. The method employs a gradient-based solver for solving the non-linear problem of finding the reciprocal form diagram to a force diagram. Further, the derivation of the Jacobian matrix, which is needed for Newton's method, is presented. We show how constraints can be added as additional equations to the Jacobian matrix in order to control solutions from the non-linear solver. In the examples of Sections 4.1 and 4.2 we show that the left null space of the Jacobian can be used to get information on which parameters cannot be changed in the force diagram. In Section 4.4 it is also shown how a vector space of possible solutions of form diagrams for the same force diagram can be computed by computing the null space of the Jacobian matrix.

When the problem becomes over constrained (has a non-trivial left null space) the parameters that can be changed in the force diagram become restricted, and the manipulations need to be perpendicular to the left null space of the Jacobian matrix. It might be worthwhile to investigate if there are graphical methods of presenting this information to the end user. The geometric constraints in the paper have been introduced to steer the solver towards solutions that meet certain design criteria, for instance, keeping the position of a support fixed, or keeping the line of action of an external force fixed. It could be of further interest to study if the constraints can be used to drive the design of 2D axially loaded structures.

The presented method can be used as a back-end for an interactive application, however, in order to enable full interactivity in computer aided graphic statics based on force diagram manipulations, some further work is required. Fig. 12 illustrates different scenarios based on manipulations of the force diagram. Scenario one shows explorations of a simple form using manipulations of the force diagram. Scenario two shows explorations with included constraints on the form diagram. In both scenarios a couple of manipulation steps are shown, in order to show that the resulting form and force diagrams can be far from the initial state. Both scenarios are supported by the presented framework, however, when there are multiple solutions the framework converges to one form out of many possible, and provides alternative forms through the use of the null space of the Jacobian matrix. These alternative forms are local to the current state, since they depend on the linearization. In an unbiased exploration the user should be able to get feedback from all possible manipulations and variations of solutions.

Scenarios three and four in Fig. 12 rely on starting from, or modifying the topology of, a force diagram. The topology matrices in Eqs. (1) and (2) are constructed from the form diagram. These can also be constructed from the force diagram, however, this has not been covered in the paper. To further adopt the methodology presented in this paper to scenarios three and four requires some additional methods to get suitable initial guesses (for instance, taking advantage of corresponding form and force edges being parallel, and that some vertices in one diagram correspond to closed polygons in the other), and a discussion on valid changes and valid starting topologies of the force diagram.

In order to improve robustness of the solver suitable methods are introduced. The global convergence is improved by including a line search, and the vertex perturbations help with difficult cases when some edges in the force diagram are of zero length. The least squares approach to solving the equation system could be used to get solutions in over-determined cases, however, this has not been done in this paper or discussed. A tangent to this for future work could be the introduction of softer constraints, which allow for some changes in the geometry. The strength and application of being able to manipulate the force diagrams have been shown through a number of examples.

Acknowledgment

This work was supported by the Swedish strategic research program eSSENCE.

References

[1] Maxwell, XLV JC. On reciprocal figures and diagrams of forces. London Edinburgh Dublin Philos Mag J Sci 1864;27(182):250–61.

[2] Cremona L, Beare TH. Graphical statics: Two treatises on the graphical calculus and reciprocal figures in graphical statics. Clarendon Press; 1890.

[3] Culmann K. Die graphische statik, vol. 2. Meyer & Zeller (A. Reimann); 1875.

[4] Allen E, Zalewski W. Form and forces: Designing efficient, expressive structures. John Wiley & Sons; 2009.

[5] Block P, Ochsendorf J. Thrust network analysis: A new methodology for three-dimensional equilibrium. Int Assoc Shell Spat Struct 2007;155:167.

[6] Fraternali F. A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions. Mech. Res. Commun. 2010;37(2):198–204.

[7] Fraternali F, Carpentieri G. On the correspondence between 2d force networks and polyhedral stress functions. Int J Space Struct 2014;29(3):145–59.

[8] Beghini LL, Carrion J, Beghini A, Mazurek A, Baker WF. Structural optimization using graphic statics. Struct Multidiscip Optim 2014;49(3):351–66.

[9] Todisco L, Fivet C, Peiretti HC, Mueller C. Design and exploration of externally post-tensioned structures using graphic statics. J Int Assoc Shell Spat Struct 2015;56(4):249–58.

[10] McRobie A, Baker W, Mitchell T, Konstantatou M. Mechanisms and states of self-stress of planar trusses using graphic statics, part II: Applications and extensions. Int J Space Struct 2016;31(2–4):102–11.

[11] Williams C, McRobie A. Graphic statics using discontinuous airy stress functions. Int J Space Struct 2016;31(2–4):121–34.

[12] Akbarzadeh M, Van Mele T, Block P. On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams. Comput. Aided Des. 2015;63:118–28. http://dx.doi.org/10.1016/j.cad.2015.01.006.

[13] Akbarzadeh M, Van Mele T, Block P. Three-dimensional graphic statics: initial explorations with polyhedral form and force diagrams. Int J Space Struct 2016;31(2–4):226.

[14] Dacunto P, Ohlbrock PO, Jasienski J-P, Fivet C, 2016. Vector-based 3d graphic statics (part I): Evaluation of global equilibrium. In: Proceedings of the IASS annual symposium 2016 "Spatial Structures in the 21st Century".

[15] Lee J, Mueller C, Fivet C. Automatic generation of diverse equilibrium structures through shape grammars and graphic statics. Int J Space Struct 2016; 31(2–4):147–64.

[16] Fivet C, Zastavni D. Constraint-based graphic statics: new paradigms of computer-aided structural equilibrium design. J Int Assoc Shell Spat Struct 2013;54(4):271–80.

[17] Fivet C, Zastavni D. A fully geometric approach for interactive constraint-based structural equilibrium design. Comput. Aided Des. 2015;61:42–57.

[18] Greenwold S, Allen E, Active statics. URL http://acg.media.mit.edu/people/simong/statics/data/index.html.

[19] Block P, Equilibrium. URL http://block.arch.ethz.ch/equilibrium/.

[20] Lachauer L, Jungjohann H, Kotnik T, Interactive parametric tools for structural design, in: Proceedings of the International Association for Shell and Spatial Structures (IABSE-IASS) Symposium, 2011.

[21] Van Mele T, Block P. Algebraic graph statics. Comput. Aided Des. 2014; 53:104–16.

[22] Micheletti A. On generalized reciprocal diagrams for self-stressed frameworks. Int J Space Struct 2008;23(3):153–66.

[23] Pellegrino S. Structural computations with the singular value decomposition of the equilibrium matrix. Int. J. Solids Struct. 1993;30(21):3025–35.

[24] Bow RH. Economics of construction in relation to framed structures. Cambridge University Press; 2014.

[25] Schek H-J. The force density method for form finding and computation of general networks. Comput. Methods Appl. Mech. Engrg. 1974;3(1):115–34.

[26] Pellegrino S, Calladine CR. Matrix analysis of statically and kinematically indeterminate frameworks. Int. J. Solids Struct. 1986;22(4):409–28.

[27] Crapo H, Whiteley W. Plane self stresses and projected polyhedra I: The basic pattem. Struct. Topol 1993;20.

[28] Python Software Foundation, Python 3.5. URL http://www.python.org.

[29] NumPy. URL http://www.numpy.org/.

[30] SciPy. URL https://www.scipy.org/.

[31] Networkx. URL http://networkx.github.io/.

[32] Gu M, Eisenstat SC. Efficient algorithms for computing a strong rank-revealing QR factorization. SIAM J. Sci. Comput. 1996;17(4):848–69.

[33] Press WH, Flannery BP, Teukolsky SA, Vetterling WT, Kramer PB. Numerical recipes: The art of scientific computing. AIP; 1987.

[34] Van Mele T, Lachauer L, Rippmann M, Block P. Geometry-based understanding of structures. J Int Assoc Shell Spat Struct 2012;53(4):285–95.