Computer-Aided Design 53 (2014) 104–116

Contents lists available at ScienceDirect

Computer-Aided Design

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

Algebraic graph statics☆

T. Van Mele*, P. Block

Institute of Technology in Architecture, ETH Zurich, 8093 Zurich, Switzerland

HIGHLIGHTS

- General, non-procedural approach to graphical analysis of two-dimensional structures.
- Equilibrium equations derived from reciprocal relation between form and force graphs.
- States of stress of structural systems from analysis of equivalent unloaded networks.
- Construction of planar straight-line drawings of planar form graphs.
- Computational back-end for interactive graphic statics software.

ARTICLE INFO

Article history:
Received 16 December 2013
Accepted 7 April 2014

Keywords:
Algebraic graphical analysis
Interactive graphic statics
Graph theory
Reciprocal figures
Matrix analysis of structures

ABSTRACT

This paper presents a general, non-procedural, algebraic approach to graphical analysis of structures. Using graph theoretical properties of reciprocal graphs, the geometrical relation between the form and force diagrams used in graphic statics is written algebraically. These formulations have been found to be equivalent to the equilibrium equations used in matrix analysis of planar, self-stressed structural systems. The significance and uses of this general approach are demonstrated through several examples and it is shown that it provides a robust back-end for a real-time, interactive and flexible computational implementation of traditional graphic statics.

1. Introduction

Graphic statics is a well-known method for analysis and design of two-dimensional structures based on Cremona's extensions of Maxwell's theory of reciprocal figures [1,2]. In graphic statics, the relation between form and forces of a structural system is contained in the reciprocal relation between two diagrams. A form diagram describes the geometrical configuration of the (axial) internal and external forces of a two-dimensional structural system, and a force diagram represents their equilibrium. The combination of these two diagrams allows for an intuitive evaluation of structural behaviour, performance and efficiency at a glance. The graphical nature of the method furthermore allows for a visual verification of both the evaluation process and results [3,4], making it more transparent than arithmetic or numerical methods.

Recent developments have demonstrated how the principles of graphic statics can be combined with modern computer technologies to create interactive drawings that provide real-time, visual feedback about the relation between form and forces in response to manipulations of the drawing by the user [5–7]. It has been demonstrated that such interactive implementations are not only extremely useful for educational purposes, but also for advanced research [8]. In addition, several graphic statics tools have also been developed as plug-ins for CAD environments (e.g. [9,10]).

1.1. Problem statement

Despite its strengths, computerised (interactive) graphic statics still has some drawbacks. The process of constructing drawings can easily become tedious and time-consuming and demands a profound familiarity with the specific geometric constructions involved (e.g. [4,11]). Furthermore, since the drawings produced by the CAD tools and interactive implementations are generated in a procedural manner according to the corresponding graphic statics "recipe", they tend to be designed for specific types of structures. Modifications to the initial setup of the drawing (e.g. the number and/or connectivity of structural elements, order of the loads, …) thus require a complete redraw of the entire construction. Although the process of making a graphic statics construction is important for teaching and learning, as it helps to get familiarised with the specific geometric and structural relationships between

☆ This paper has been recommended for acceptance by Charlie C.L. Wang.

* Corresponding author.
E-mail addresses: vanmelet@ethz.ch (T. Van Mele), pblock@ethz.ch (P. Block).

http://dx.doi.org/10.1016/j.cad.2014.04.004
0010-4485/

different elements of such construction, it is clear that it is inconvenient for research or practical purposes.

1.2. Objectives

To fully explore the benefits of computerised graphic statics, a general, non-procedural approach is needed, which allows drawings to be created without specific knowledge of the geometric construction procedures involved.

This paper presents a robust back-end for a graphics statics application. It allows the user to start from a connected two-dimensional line drawing. The graph of this line drawing is automatically constructed and analysed to assess the feasibility of the input as a structural system or set of forces in equilibrium. If possible, a reciprocal force diagram is constructed, based on user-defined loading or self-stress conditions. The two diagrams can then be manipulated interactively without breaking their topological and geometrical relationship. As such, the user can explore different states of equilibrium by explicit, geometric modifications of the connected diagrams, or redistribution of forces within given constraints.

1.3. Contributions and outline

The remainder of this paper is organised as follows.

In Section 2, we bring together concepts and techniques from graph theory and matrix analysis of structures and present them in a unified framework for algebraic graphical analysis built around the reciprocal relation between the form and force diagrams of graphic statics.

In Section 3, we discuss a general scheme for a computational implementation of the presented approach that can be used as back-end of a real-time, interactive graphic statics application. Different steps of the implementation are illustrated using a Fink truss, which is a statically determinate structure that cannot be calculated directly with traditional graphic statics, because it contains crossing edges. Relevant algorithms are provided as code snippets.

In Section 4, the use of this framework for non-procedural graphic statics is demonstrated through four examples: a three-hinged trussed frame, an externally statically indeterminate three-bar truss, a geometrically constrained thrust line, defining its funicular loading, and a pre-stressed net. Finally, we briefly discuss the relevance of the presented approach for three-dimensional equilibrium methods, such as Thrust Network Analysis [12].

2. Theoretical framework

In this section, we describe the theoretical framework for the graph-based, algebraic approach to graphic statics presented in this paper. First, we briefly revisit traditional graphic statics, and describe the graph interpretation of form and force diagrams. Next, we formulate the reciprocal constraints between these diagrams algebraically, and derive from them the typical equilibrium equations of a (self-stressed) structural system. We furthermore show how the geometry of the force graph can be readily derived from the solution of the equilibrium equations and the topological information of the form graph. Finally, we discuss the solution strategies for different types of structural systems based on *Singular Value Decomposition* (SVD) of the equilibrium matrix of their form graph, and describe the interpretation of the obtained results in the context of graphic statics.

2.1. Graphic statics

Fig. 1 depicts a typical graphic statics drawing consisting of two diagrams that together describe the static equilibrium of a bar-node structure and a set of applied loads and reaction forces. The

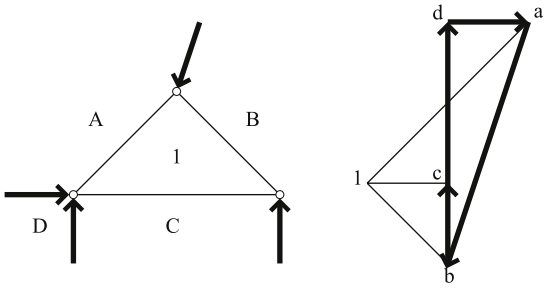

Fig. 1. Corresponding lines in reciprocal diagrams are parallel and corresponding lines which converge to a point in one diagram form a closed polygon in the other. In a graphic statics context, the diagram to the left is often called the form diagram and the one to the right the force diagram.

two diagrams are *reciprocal*: they consist of an equal number of lines, so that the corresponding lines in the two diagrams are parallel (or perpendicular, or at any constant angle), and corresponding lines which converge to a point in one diagram form a closed polygon in the other [2].

The diagram to the left is the *form diagram* and the one to the right the *force diagram*. Bow's notation is used to label spaces in the form diagram and their corresponding nodes in the force diagram [13].

A closed polygon in the force diagram represents the static equilibrium of the corresponding point in the form diagram, with the magnitude of force in the converging lines at that point equal (or proportional) to the length of the sides of the closed polygon. The form diagram thus contains the actual configuration of the bars, nodes, support forces and applied loads in space. The force diagram describes global equilibrium and the equilibrium of forces in the bars at each of the nodes. Note that it is common practice in graphic statics drawings to represent external forces in the form diagram by unit vectors, indicating only the direction and point of application of these forces. As with the internal forces in the structure, their magnitude is proportional to the length of the corresponding line segments in the force diagram.

2.2. Reciprocal graphs

The form and force diagrams can be interpreted as *directed graphs* for which (directed) *incidence* or *connectivity* matrices can be constructed describing the topological relation between branches and nodes (Fig. 2). We define the graph of the form diagram as the *form graph G*, and the one of the force diagram as its *reciprocal force graph* G^*. The force graph is the *topological dual* of the form graph with the added requirement that corresponding edges are parallel. The elements of G and G^* are *vertices*, *edges* and *faces*. Elements of the force graph are superscripted with an asterix (*).

Due to the presence of external forces in the form diagram, the form graph contains *leaf vertices*. These are vertices of degree one since they have only one connected edge, which corresponds to an external force. In Section 2.5, we will see that it is a requirement of the presented approach that the leaf vertices and their edges can be drawn on the outside of the graph, in the *outer* or *external space*. The leaf vertices will therefore be referred to as *outer* or *external vertices*; the others as *inner* or *internal*. Note that this requirement simply means that all external forces should be applied to nodes on the boundary of a structure. In Fig. 2, for example, vertices 1, 2, 4 and 6 are external vertices. They are, respectively, connected to edges 0, 1, 2 and 3 representing the external forces in the form diagram depicted in Fig. 1.

For a form graph G with e number of edges and v number of vertices, the entries of the jth column of the $[v \times e]$ connectivity

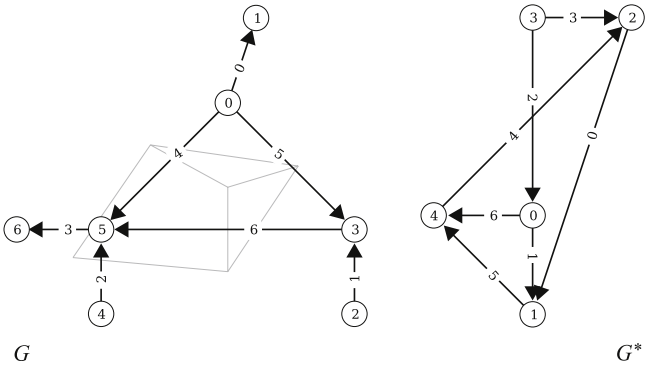

Fig. 2. Form and force diagrams can be interpreted as directed graphs. In this context, we define them as the form graph G and the (reciprocal) force graph G^*.

matrix $\mathbf{C}$ are:

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

Note that these edge directions may be chosen arbitrarily, but some convention can be imposed for clarity. Here, for example, all edges point from vertices with lower index to vertices with higher index.

The $[v^* \times e]$ incidence matrix $\mathbf{C}^*$ represents the topology of the reciprocal force graph G^*. The reciprocal graph has v^* vertices, equal to the number of faces of the form graph, and the same number of edges e as the form graph. The row vectors of $\mathbf{C}^*$ can be constructed from the form graph by cycling its faces in a counterclockwise direction [14]. The entries of the ith row of $\mathbf{C}^*$, corresponding to the ith face of G, are:

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

2.3. Algebraic reciprocal constraints

Having constructed $\mathbf{C}$ and $\mathbf{C}^*$, the *coordinate difference* vectors $\mathbf{u}$, $\mathbf{v}$, $\mathbf{u}^*$ and $\mathbf{v}^*$ of the form and force edges can be written as

$$\mathbf{u} = \mathbf{C}^{\mathrm{t}}\mathbf{x}, \qquad \mathbf{v} = \mathbf{C}^{\mathrm{t}}\mathbf{y} \tag{3a}$$

$$\mathbf{u}^* = \mathbf{C}^{*\mathrm{t}}\mathbf{x}^*, \qquad \mathbf{v}^* = \mathbf{C}^{*\mathrm{t}}\mathbf{y}^* \tag{3b}$$

with $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{x}^*$, $\mathbf{y}^*$ the x and y coordinate vectors of G and G^*, respectively. The reciprocal constraints, as described in Section 2.1, can be formulated in terms of the coordinate difference vectors (3a) and (3b).

The first set of constraints, requiring that lines intersecting at a node in the form diagram correspond to a closed polygon in the force diagram, can be imposed by expressing that the sum of coordinate difference vectors of edges of the force graph connected to the same vertex in the form graph should be zero [12]:

$$\begin{cases} \mathbf{C}_{\mathrm{i}}\mathbf{u}^* = 0 \\ \mathbf{C}_{\mathrm{i}}\mathbf{v}^* = 0 \end{cases} \tag{4}$$

with $\mathbf{C}_{\mathrm{i}}$ composed of the rows of $\mathbf{C}$ corresponding to the inner vertices of G.

The second set of constraints, requiring parallelity between corresponding edges in the form and force graphs, can be imposed by writing the coordinate difference vectors $\mathbf{u}^*$ and $\mathbf{v}^*$ of the force graph as a function of the corresponding vectors $\mathbf{u}$ and $\mathbf{v}$ of the form graph:

$$\begin{cases} \mathbf{u}^* = \mathbf{Q}\mathbf{u} \\ \mathbf{v}^* = \mathbf{Q}\mathbf{v}, \end{cases} \tag{5}$$

with $\mathbf{Q}$ the diagonal matrix of the vector $\mathbf{q}$ of length ratios between corresponding edges in G^* and G. The elements of $\mathbf{q}$ are, in fact, the well-known *force densities*, first introduced by Schek [15]. Note that after imposing (4), G and G^* are dual graphs. By furthermore imposing (5), they become reciprocal.

Combining the two sets of reciprocal constraint equations, (4) and (5), and using $\mathbf{Q}\mathbf{u} = \mathbf{U}\mathbf{q}$ and $\mathbf{Q}\mathbf{v} = \mathbf{V}\mathbf{q}$, with $\mathbf{U}$ and $\mathbf{V}$ the diagonal matrices of the coordinate difference vectors $\mathbf{u}$ and $\mathbf{v}$, one finds that

$$\begin{cases} \mathbf{C}_{\mathrm{i}}\mathbf{U}\mathbf{q} = 0 \\ \mathbf{C}_{\mathrm{i}}\mathbf{V}\mathbf{q} = 0 \end{cases} \tag{6}$$

or, equivalently

$$\mathbf{A}\mathbf{q} = \mathbf{0} \tag{7}$$

with $\mathbf{A}$ the $[2v_{\mathrm{i}} \times e]$ *equilibrium matrix* of G:

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}_{\mathrm{i}}\mathbf{U} \\ \hline \mathbf{C}_{\mathrm{i}}\mathbf{V} \end{bmatrix}. \tag{8}$$

Note that since all external forces are included in the form graph, the right-hand side of (7) is a null vector. In the presented approach, the equilibrium of a structural system is thus investigated using the *states of self-stress* of an equivalent unloaded network.

The number k of independent states of self-stress of the network, corresponds to the dimension of the *nullspace* of $\mathbf{A}$ [16]. The dimension of the nullspace can be calculated by SVD of $\mathbf{A}$ [17] and is thus equal to the number of edges of which the force densities can be chosen freely to explore different states of equilibrium of the represented structural system. We will call these the *free* or *independent edges*. The structural interpretation of the states of self-stress depends on the type of assembly or structural system represented by the form graph, as we will see in Section 2.6.

2.4. The force graph

It can be seen from Eqs. (3b) and (5) that the coordinate vectors $\mathbf{x}^*$ and $\mathbf{y}^*$ are related to the force densities in the following way:

$$\begin{aligned} \mathbf{C}^{*\mathrm{t}}\mathbf{x}^* &= \mathbf{Q}\mathbf{u} \\ \mathbf{C}^{*\mathrm{t}}\mathbf{y}^* &= \mathbf{Q}\mathbf{v}. \end{aligned} \tag{9}$$

This system of equations cannot be solved directly, since the incidence matrix of the force graph, $\mathbf{C}^*$, is not square. Instead, the geometry of the force graph can be determined by solving the equivalent system

$$\begin{aligned} \mathbf{L}^*\mathbf{x}^* &= \mathbf{C}^*\mathbf{Q}\mathbf{u} \\ \mathbf{L}^*\mathbf{y}^* &= \mathbf{C}^*\mathbf{Q}\mathbf{v}, \end{aligned} \qquad \mathbf{L}^* = \mathbf{C}^*\mathbf{C}^{*\mathrm{t}} \tag{10}$$

with $\mathbf{L}^*$ the *Laplacian matrix* of the force graph. This system has an exact solution which can be calculated efficiently [18], since $\mathbf{L}^*$ is square and *positive, semi-definite* [16].

Note that the vertex coordinates of the force graph are unique up to a translation. Therefore, in order to obtain one specific solution, we solve the system of equations with the location of one point chosen. In graphic statics, this is equivalent to choosing the first point of the load line somewhere on the drawing canvas to start the construction of the force diagram.

2.5. Requirements

The approach presented in this paper is based on the interpretation of the form and force diagrams of graphic statics as dual graphs, with the added constraint that corresponding edges should be parallel (or at any constant angle), as explained in Section 2.2. It is essential that these graphs can be constructed automatically and

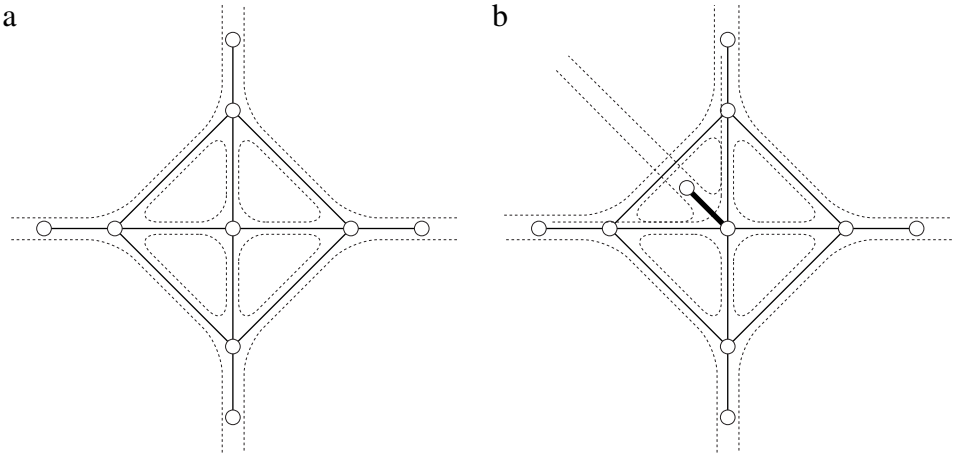

Fig. 3. (a) A planar form graph without overlapping spaces. (b) A planar form graph with overlapping spaces due to the spaces created by the leaf connected to the central vertex.

uniquely if the approach is to be used as back-end for an interactive graphic statics application.

A (form) graph has a dual if and only if it is *planar* [19]. Therefore, the presented approach only applies to structural systems that have a form diagram with a *planar graph*. A graph is planar if it can be drawn in the plane without crossing edges [20]. Furthermore, it should be possible to construct a straight-line drawing of the form graph that divides the plane into bounded and unbounded polygonal spaces with disjoint interiors [21], i.e. spaces without overlaps. A straight-line drawing is a drawing in which all vertices are connected with straight lines. According to [22], any planar graph has a planar straight-line drawing.

As an example, consider the two form graphs in Fig. 3. Fig. 3(a) is the planar straight-line drawing of a valid form graph. There are no crossing edges and none of the spaces overlap. All edges corresponding to external forces are on the outside of the drawing. Fig. 3(b) is the drawing of an invalid form graph. The drawing is crossing-free, but it inevitably has overlapping spaces due to the presence of the leaf edge at the central vertex. This edge corresponds to an external force applied to a node of the structure that is not on the boundary.

From a *planar straight-line drawing without overlapping spaces*, the dual of a valid form graph can be constructed uniquely and efficiently using a "*wall following*" *maze solving algorithm*. A wall follower identifies the clockwise cycles of the drawing by traversing the graph and always taking the rightmost (or leftmost for counter-clockwise cycles) edge out of a vertex. All cycles are identified once all edges have been traversed in both directions. These cycles can then be inspected as discussed in Section 2.2 to determine the topology of the dual.

2.6. Structural systems

The presented approach can be used to investigate the possible states of equilibrium of different types of structural systems, with different degrees of static and kinematic (in)determinacy. The static and kinematic (in)determinacy of a structural system can be evaluated using the *extended Maxwell rule* [23]:

$$k - m = b - 2n + r, \tag{11}$$

with k the number of independent states of self-stress, m the number of inextensible mechanisms, b the number of bars, n the number of nodes, and r the number of kinematic restraints at the supports. Applied to a form graph, since all external forces (loads and support forces) are included in the graph, Eq. (11) simplifies to

$$k - m = e - 2v_i, \tag{12}$$

with e the number of edges and v_i the number of internal vertices. The values of k and m can be calculated by SVD of the equilibrium matrix $\mathbf{A}$ [17] and determine how the equilibrium problem should be solved and interpreted.

Rigid, statically determinate and (externally) indeterminate systems have a form graph with $m = 0$ and $k > 0$. In a determinate system, the number of independent edges k is equal to the number of applied loads. A form graph representative of a determinate system is depicted in Fig. 4(a). The number of independent edges in an indeterminate system is greater than the number of applied loads, as seen in Fig. 4(b).

Funicular systems have $m = 0$ and $k = 1$, regardless of the number of applied loads. These systems are stable in a specific geometry for only one specific equilibrium of internal and external forces (Fig. 4(c)). Therefore, there can be only one degree of freedom or independent edge. Any of the edges in the graph of a funicular system may be identified as the free edge.

Finally, if the form graph represents an unloaded, two-dimensional, self-stressed network, the $k > 0$ independent states of self-stress of the graph represent the actual states of self-stress of the structural system. Generally these systems have a form graph with $m = 0$. However, in axisymmetric configurations, such as the one depicted in Fig. 4(d), the form graphs of these networks can have $m > 0$. For example, the network represented by the form graph in Fig. 4(d) has $m = 1$. Indeed, the rotation of the vertices forming the square cycle around the central node is an inextensible deformation of the network [24].

Note that if $k = 0$, no states of self-stress exist for the form graph. Therefore, regardless of the value of m, the graph does not represent a stable structural system. This is the case if, for example, the form graph represents an equilibrium system in an infeasible configuration.

The free or independent edges cannot always be selected arbitrarily and it is not always obvious which combinations of edges are allowed. In those cases, a possible set can be identified by transforming $\mathbf{A}$ into *Reduced Row Echelon Form* (RREF) using *Gauss–Jordan Elimination* (GJE). The *non-pivot* columns of RREF($\mathbf{A}$) correspond to the free variables in the vector of force densities $\mathbf{q}$ and thus to a possible set of free or independent edges [24].

Having identified the independent edges, either by manual selection or using GJE, (7) can be rewritten as

$$\mathbf{A}_\mathrm{d}\mathbf{q}_\mathrm{d} = -\mathbf{A}_\mathrm{id}\mathbf{q}_\mathrm{id}, \tag{13}$$

with $\mathbf{A}_\mathrm{d}$ and $\mathbf{A}_\mathrm{id}$ containing the columns of $\mathbf{A}$ corresponding to the dependent and independent edges respectively, and $\mathbf{q}_\mathrm{d}$ and $\mathbf{q}_\mathrm{id}$ containing the force densities of those edges.

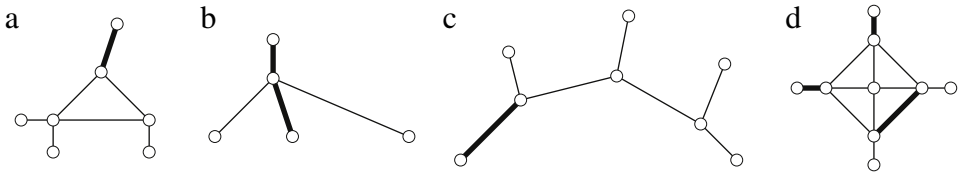

Fig. 4. (a) A statically determinate structural system, with the edge corresponding to the applied load as selected independent edge. (b) A statically indeterminate structural system, with a selection of independent edges corresponding to the applied load and one support force. (c) A funicular system of forces, with only one independent edge. (d) A two-dimensional, self-stressed network of forces, or "spiderweb", with three degrees of freedom.

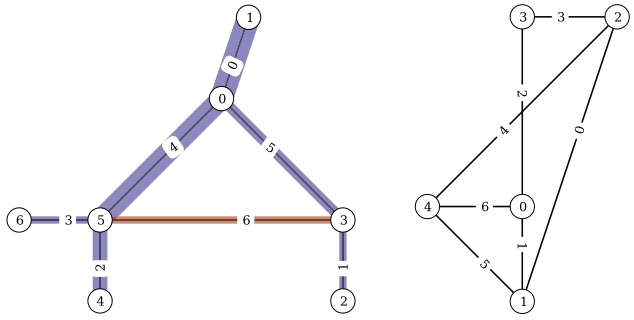

Fig. 5. Using the convention for constructing the dual graph described in Section 2.2, edges with positive force density values or with the same orientation in both graphs are in tension (red); those with negative values or opposite orientation in compression (blue). The thickness of an edge of the form graph (left) is proportional to the length of the corresponding edge in the force graph (right) and thus to the size of the internal force. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

If $m = 0$, the $[2v_i \times (e-k)]$ matrix $\mathbf{A}_\mathrm{d}$ is square and non-singular and thus invertible, such that (13) can be rewritten as

$$\mathbf{q}_\mathrm{d} = -\mathbf{A}_\mathrm{d}^{-1}\mathbf{A}_\mathrm{id}\mathbf{q}_\mathrm{id}. \tag{14}$$

The full set of force densities $\mathbf{q}$ can then be obtained directly as a function of the chosen $\mathbf{q}_\mathrm{id}$.

If $m > 0$, $\mathbf{A}_\mathrm{d}$ is not square. In this case, the remaining force densities $\mathbf{q}_\mathrm{d}$ can be calculated by solving the following equivalent linear system

$$\mathbf{A}_\mathrm{d}^\mathrm{t}\mathbf{A}_\mathrm{d}\mathbf{q}_\mathrm{d} = -\mathbf{A}_\mathrm{d}^\mathrm{t}\mathbf{A}_\mathrm{id}\mathbf{q}_\mathrm{id}. \tag{15}$$

2.7. Tension and compression

The interpretation of the force in a member as either tension or compression depends on the conventions that were used for deriving the incidence matrix $\mathbf{C}^*$ (Section 2.2).

For a typical graphic statics interpretation, meaning that the equilibrium of the nodes is inspected by cycling around those nodes in clockwise direction [4], the dual incidence matrix should be created by cycling through the primal faces in a *counter-clockwise* direction. By assigning $+1$ to edges that have the same orientation as the cycling direction (this is the convention used in Section 2.2), edges with positive force density values will represent primal edges in tension; and edges with negative force densities represent primal edges in compression. Whether an edge is in compression or tension can thus simply be determined from the sign of the calculated force density. Consequently, when visually inspecting G and G^*, a primal edge having a corresponding dual edge with the same orientation will be in tension; and a primal edge with a corresponding dual edge with opposite orientation will be in compression. This is a direct result of the parametric equations (5) and can be seen in Fig. 5.

2.8. Force-driven design

So far, we have described the required computational steps of a method for constructing the reciprocal force graph of a given form graph. This method can be used to investigate the possible (if any) states of stress/equilibrium of a structural system in a specific configuration and thus provides the basis for the back-end of a tool for graphic statics-based analysis of structures. As described in [8], graphic statics can also be used for sophisticated force-driven design explorations. Typically, the goal in these explorations is to find an equilibrium configuration for a structure based on a set of constraints; for example, the location of the supports, the spatial configuration of the loads, size of the forces in specific elements…The problem then becomes a constrained optimisation problem that is not simply the reverse problem of the one solved in this paper. Therefore, it will not be discussed here.

3. Computational setup

In this section, we discuss a general scheme for an implementation of the presented approach that can be used as the back end of an interactive application for algebraic graphical analysis of structures. Relevant code snippets are provided using the scripting language Python [25].

An overview of the implementation is depicted in Fig. 6. Different steps of the algorithm are illustrated using a Fink truss. This type of truss is a statically determinate structure that cannot be calculated directly with graphic statics, because its form diagram contains crossing elements.

However, we will demonstrate how the system can be solved with the presented method by generating a planar straight-line drawing of the form graph without overlapping spaces.

3.1. Form graph

The algorithm takes as input a set of connected lines representing a form diagram describing the spatial/geometrical configuration of all internal and external forces of a structural system. The only requirement for this drawing to be valid input is that all edges are properly connected.

The directed connectivity graph of this set of lines can be easily constructed by identifying all unique vertices among the start and end points of the lines, and assigning an ordered pair of vertices (an edge) to each line. The form graph of a Fink truss with six external forces (three loads and three reaction forces) is depicted in Fig. 7.

3.2. Planar straight-line drawing with non-overlapping spaces

As discussed, the presented approach can only be applied to structural systems with planar form graphs. Furthermore, a planar straight-line drawing of the form graph with non-overlapping spaces is needed to construct the dual graph using a wall follower. This part of the algorithm therefore consists of the following steps:

1. Check if the current drawing of the form graph has crossing edges. If there are no crossing edges, skip the next step.
2. If the graph is planar, generate a planar straight-line drawing. Otherwise, stop.
3. Check if none of the leaf vertices is connected to an inner vertex of the graph. In other words, verify that all external forces are applied to boundary nodes of the structure.

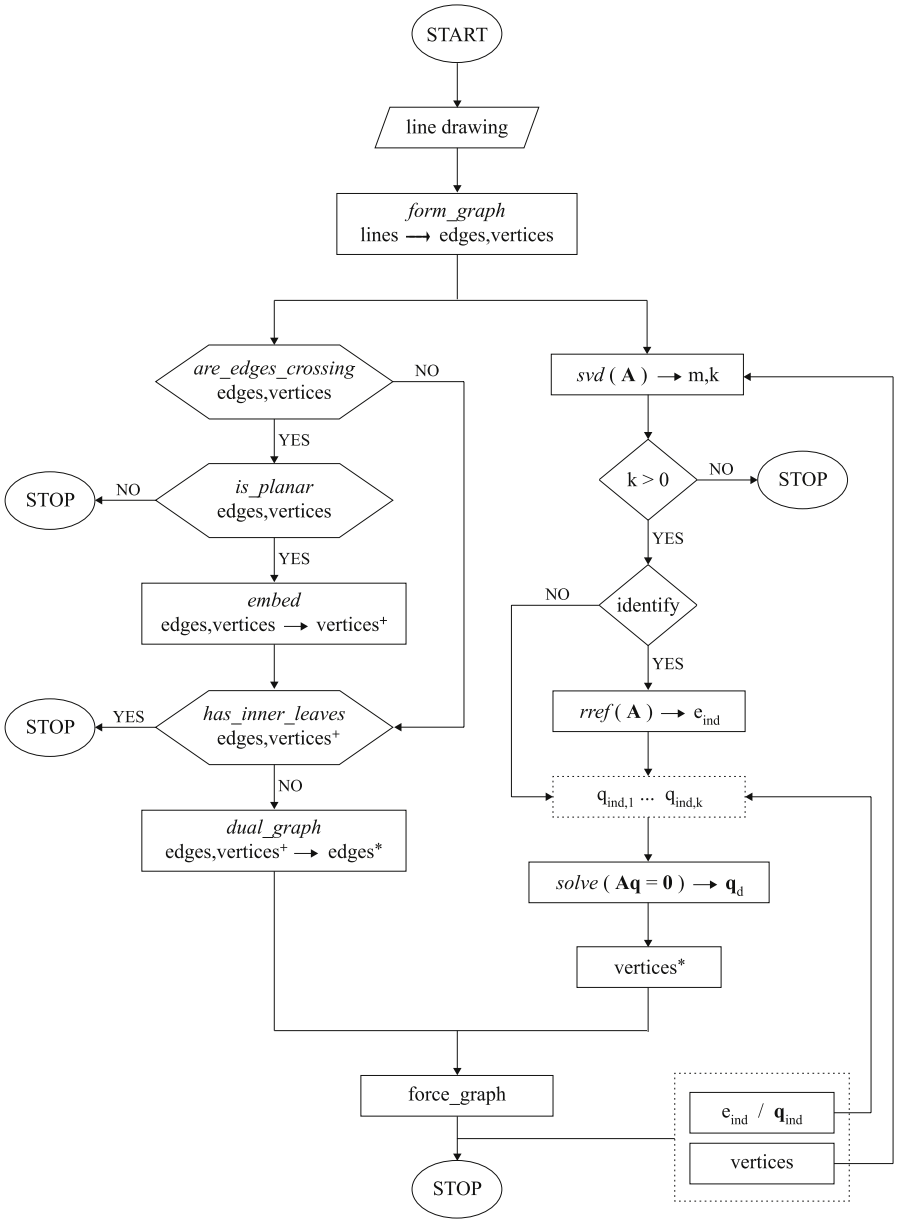

Fig. 6. Overview of different steps of an implementation of the presented approach.

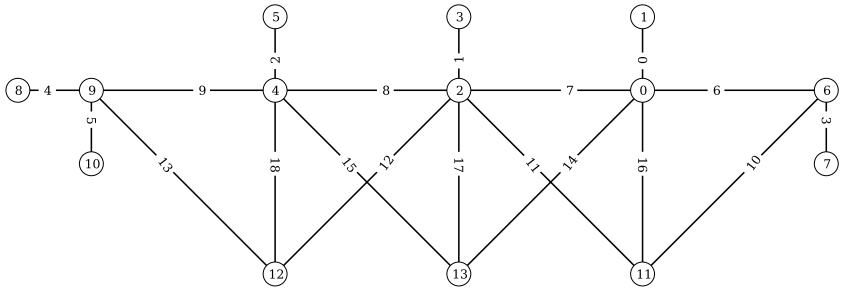

Fig. 7. The graph of the form diagram of a Fink truss has crossing edges.

3.2.1. Crossing edges test

A simple algorithm for verifying if the current representation of the form graph has crossing edges is included in Listing 1. The algorithm checks for each pair of edges if they intersect and returns *true* if a pair is found that does. Edges with a common vertex are omitted from the test. Two edges *AB* and *CD* intersect if the triangles *ABC* and *ABD*, as well as the triangles *CDA* and *CDB*, have opposite cycling directions. This indicates that *C* and *D* lie on

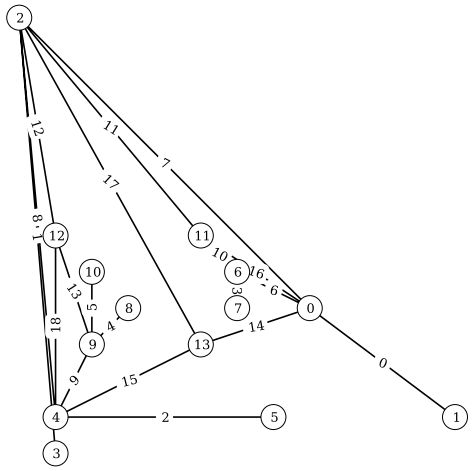

Fig. 8. A planar crossing-free straight-line drawing of the form graph generated using Schnyder's realizer method. The drawing is clearly not very useful for our purposes.

opposite sides of a line formed by A and B, and that A and B lie on opposite sides of a line formed by C and D, which can only be the case if the edges intersect. The cycle direction of three points A, B and C is determined by comparing the ratios of the x and y coordinate differences of B and C with respect to A.

The graph of the Fink truss depicted in Fig. 7 obviously has crossing edges, which is confirmed by the algorithm. Therefore, planarity should be checked.

3.2.2. Planarity testing

Several linear-time algorithms for planarity testing exist, e.g. [26–28]. The test used in this paper is an implementation of the classic algorithm by Hopcroft and Tarjan [26]. As expected, a test of the Fink graph confirms it is planar. This means a planar straight-line embedding exists.

3.2.3. Crossing-free drawing

As with planarity testing, several algorithms for generating planar, straight-line drawings exist, e.g. [22,29,30]. However, as (most of) these classic algorithms are merely concerned with establishing the existence of a planar (straight-line) drawing, for example to prove that a graph is planar, they often produce drawings that are not very useful in practice [31]. As an example, Fig. 8 depicts an embedding of the Fink graph generated with an implementation of Schnyder's *realizer method* [30], available through *Sage* [32].

Force-directed methods, on the other hand, produce layouts of graphs by positioning vertices as a result of a simulation of motion caused by repelling and attracting forces assigned to the vertices and the edges. Although these algorithms do not base their layouts on the topological information of a graph, they tend to produce crossing-free layouts for planar graphs [33]. Furthermore, because of the repelling forces between the nodes, leaf vertices tend to end up in the outside face. Since the graphs of form diagrams of relevant structural problems are relatively small and limited in complexity, force-directed algorithms can thus be used as a fast and reliable method for generating planar straight-line drawings of planar graphs by simply generating one layout after the other until one without crossing edges is obtained.

Fig. 9 is a drawing generated with an iterative implementation (Listing 1) of the Fruchterman–Reingold algorithm [34] available through *NetworkX*, a Python package for working with graphs [35], in combination with the crossing-edge check discussed above.

Listing 1: Python snippet for generating a planar, straight-line drawing without overlapping spaces of a planar graph. *vertices* is a list of vertex coordinates. *edges* is a list of ordered index pairs in the vertex list.

```
import networkx as nx

def draw_without_crossing_edges(edges,vertices):
    start = dict(enumerate(vertices))
    while True:
        G = nx.Graph(edges)
        coords = nx.spring_layout(G, pos=start, iterations=500)
        vertices = coords.values()
        if not are_edges_crossing(edges,vertices):
            break
    return vertices

def are_edges_crossing(edges,vertices):
    for edge in edges:
        A = vertices[edge[0]]
        B = vertices[edge[1]]
        for test in edges:
            if any(i in test for i in edge):
                continue
            else:
                C = vertices[test[0]]
                D = vertices[test[1]]
                if intersect(A,B,C,D):
                    return True
    return False

def intersect(A,B,C,D):
    return ( ccw(A,C,D) != ccw(B,C,D)
        and ccw(A,B,C) != ccw(A,B,D) )

def ccw(A,B,C):
    return (C[1]-A[1]) * (B[0]-A[0]) > (C[0]-A[0]) * (B[1]-A[1])
```

3.2.4. Inner leaves check

Finally, we verify if none of the edges representing an external force is connected to an inner vertex of the graph. To do so, we simply remove all leaves from the graph, identify the vertices of the boundary cycle, and verify that all leaves were connected to the graph at one of these vertices.

Note that the leaf vertices have no influence on the outcome of the crossing edges test nor of the planarity test. In fact, planarity testing is performed on each biconnected component separately and would thus skip the leaf vertices anyway. Furthermore, generating a crossing-free drawing with the algorithm provided in Listing 1 is more robust and faster without the leaf vertices as well. Therefore, the leaf vertices could also be removed before any of the other steps, and re-added afterwards.

In any case, at this point, if everything checks out, the connectivity matrix $\mathbf{C}^*$ of the dual of the (generated) planar straight-line drawing can be determined using a wall follower, as discussed in Section 2.5. The planar straight-line drawing and its dual are depicted in Fig. 9. The dual is drawn with dotted lines and its vertices placed at the centroids of the corresponding faces of the original graph.

3.3. Structural system and force densities

The equilibrium matrix $\mathbf{A}$ of the form graph of the Fink truss example has $2v_i = 16$ rows and $m = 19$ columns. By SVD of the equilibrium matrix, using the implementation provided by the linear algebra module of *Scipy* [36], we obtain $m = 0$ and $k = 3$. Note that the vector of singular values $\mathbf{s}$ returned by this implementation contains all values greater than machine precision. Therefore, the actual number of relevant singular values still needs to be determined by comparison to an appropriate tolerance. The tolerance used in this paper is 10^{-3} of the highest singular value: $tol = \mathbf{s}[0] \times 10^{-3}$ [17].

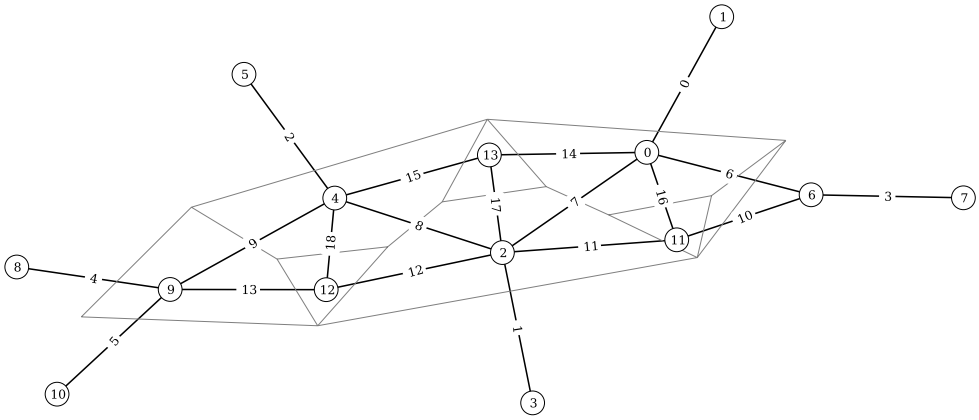

Fig. 9. A planar straight-line drawing of the form graph generated using a force-driven approach. The dual of the form graph is depicted with dotted lines. Its vertices are placed at the centroids of the corresponding faces of the form graph.

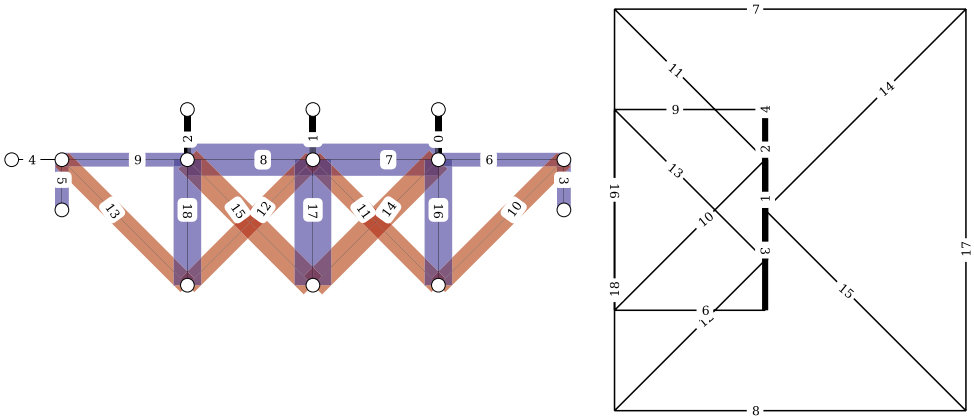

Fig. 10. The reciprocal form (left) and force (right) graphs of a Fink truss. The colours of the edges of the force graph indicate tension (red) or compression (blue). The thickness of an edge is proportional to the size of the force in it. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Since k is equal to the number of applied loads, the system is statically determinate. This means there is no need for identifying the independent edges; the force densities of any three edges of the graph can be chosen freely to obtain a state of equilibrium of the system. An obvious option, however, is to choose the force densities of the edges corresponding to the applied loads, which are edges 0, 1 and 2.

After selecting a set of independent edges, partitioning both $\mathbf{A}$ and $\mathbf{q}$ accordingly, and choosing values for the three $\mathbf{q}_{\text{id}}$, the remaining 16 $\mathbf{q}_{\text{d}}$ can be calculated directly since the matrix $\mathbf{A}_{\text{d}}$ is square and non-singular. With the force densities of edges 0, 1 and 2 chosen as -2, we obtain the following result:

$$\mathbf{q} = [-2, -2, -2, -3, 0, -3, -1.2, -2.8, -2.8, -1.2, 1.2, 1.2, 1.2, 1.2, 1.6, 1.6, -2.4, -3.2, -2.4]. \quad (16)$$

As discussed in Section 2.7, negative force densities correspond to forces directed towards the nodes and thus to edges in compression. In the case of our Fink truss example, this also means that the applied loads are acting in the downward direction. The force density in edge 4 is $q_4 = 0$. There is thus no horizontal reaction force at the left support, since, indeed, there is only vertical loading. Edges 3 and 5 correspond to the vertical reaction forces at the left and right supports.

3.4. Force graph

The final step is the construction of the force graph. Having determined the connectivity of the force graph in Section 3.2, the x and y coordinates of the vertices can be calculated from the obtained force densities using Eqs. (10).

The reciprocal form and force graphs of the Fink truss example are depicted in Fig. 10. The forces in the edges of the form graph are visualised using thickness, indicating the (proportional) size of forces, and colour, indicating tension or compression (red for tension and blue for compression). This convention has been used in all subsequent figures.

3.5. User interaction

Once the reciprocal relation between the form and force graph has been established and the graphs have been created in a scriptable drawing environment such as Rhino (or the browser), several types of user interaction are possible.

First of all, users can simply change the values of the force densities and/or the selection of independent edges to investigate different force distributions (Fig. 11).

The user can also change the geometry of the form graph to explore different configurations of the truss and different load and support conditions (Fig. 12). Note that for some structures, changing the geometry requires re-calculation and possibly re-identification of the independent edges (see Section 4.4). Here, however, this is not the case, since the structural system is rigid and statically determinate.

Finally, the data structures containing the topological information about the form and force graphs simplify visualisation of the reciprocal relation between the elements of the two graphs, for

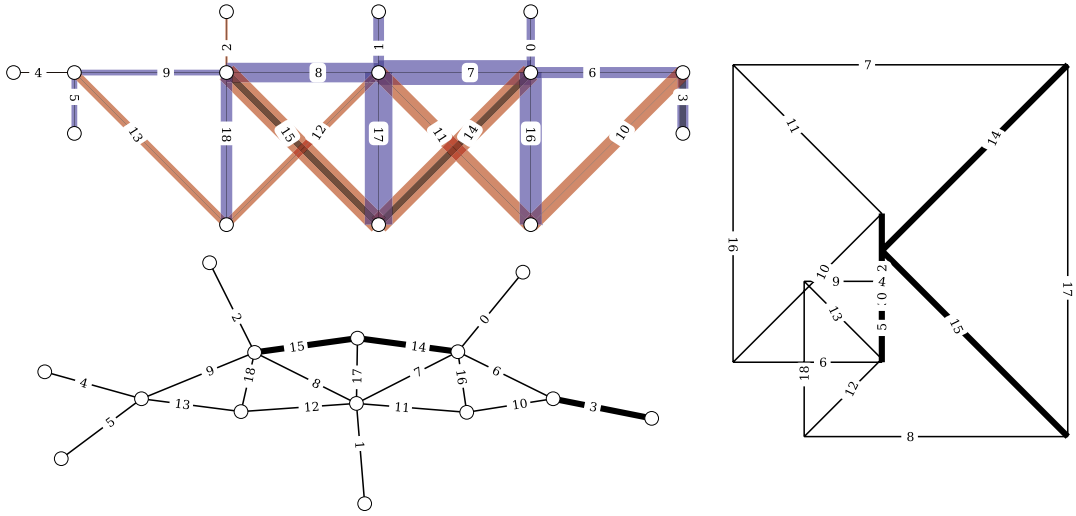

Fig. 11. Investigate different force distributions by changing values of force densities and/or the selection of independent edges. Left: The form graph (top) and its planar drawing (bottom). Right: The reciprocal force graph. In this case, the force densities of edges 3, 14 and 15 are chosen by the user, resulting in a different equilibrium compared to Fig. 10.

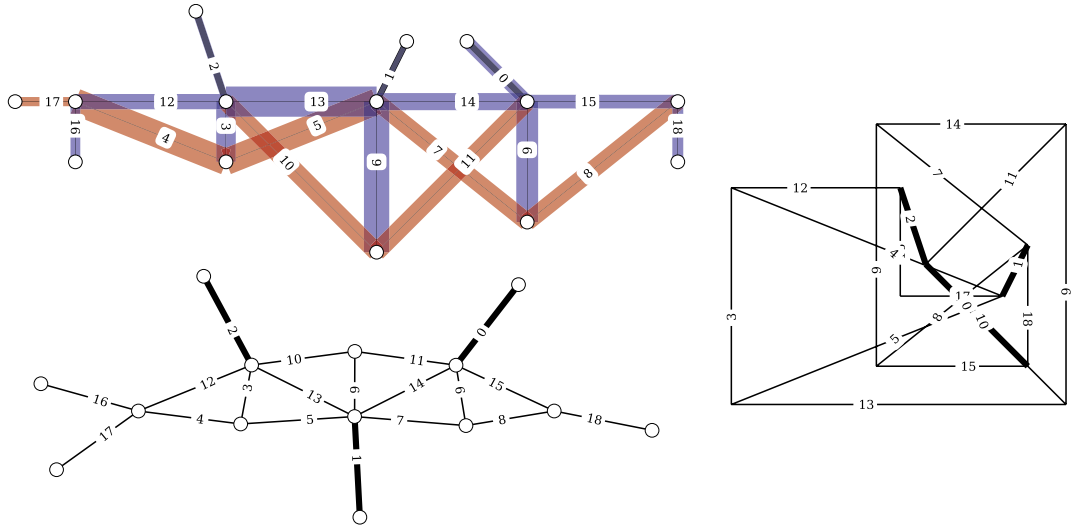

Fig. 12. The geometry of the form graph can be changed to explore different truss configurations or different load and support conditions. Left: The form graph (top) and its planar drawing (bottom). Right: The reciprocal force graph.

example by reducing the identification of corresponding vertices and faces and the connected edges to a simple topological operation (Fig. 13).

4. Results

Here, we illustrate the versatility of the presented approach using a series of examples of different structural problems. In all figures, edges of the form graph marked in red are in tension, blue edges in compression.

4.1. A three-hinged trussed frame

Finding the reaction forces, and subsequently the bar forces, of a three-hinged trussed frame is a well-known but more advanced example of a graphic statics procedure. A detailed recipe for this procedure can be found in e.g. [4].

The form graph of a three-hinged trussed frame with applied loads and reaction forces is depicted in Fig. 14(a). By SVD of the equilibrium matrix we find that $m = 0$ and $k = 5$, which is equal to the number of loads. Therefore, the structure is statically determinate. By choosing values for the force densities of the edges representing the loads, both the reaction forces and the bar forces can be calculated directly as shown before. The resulting reciprocal force graph is depicted in Fig. 14(b).

4.2. Statically indeterminate structures

Consider the bar-node structure and its corresponding form graph G, depicted in Fig. 15. SVD($\mathbf{A}$) results in $m = 0$ and $k = 2$ indicating that the network has two independent states of self-stress, which means that the structure is statically indeterminate since there is only one applied load.

Having chosen a value for the force density of the edge representing the applied load (edge 3), every choice for the second independent edge results in a state of self-stress of the network. Three such states are shown in Fig. 15. For a discussion on the meaning of the existence of different states of self-stress in an indeterminate structure, we refer the interested reader to Van Mele et al. [8].

4.3. Funicular geometry

The form graph in Fig. 16 represents a freeform discretised spline with 9 applied loads. The spline is not a rigid structure

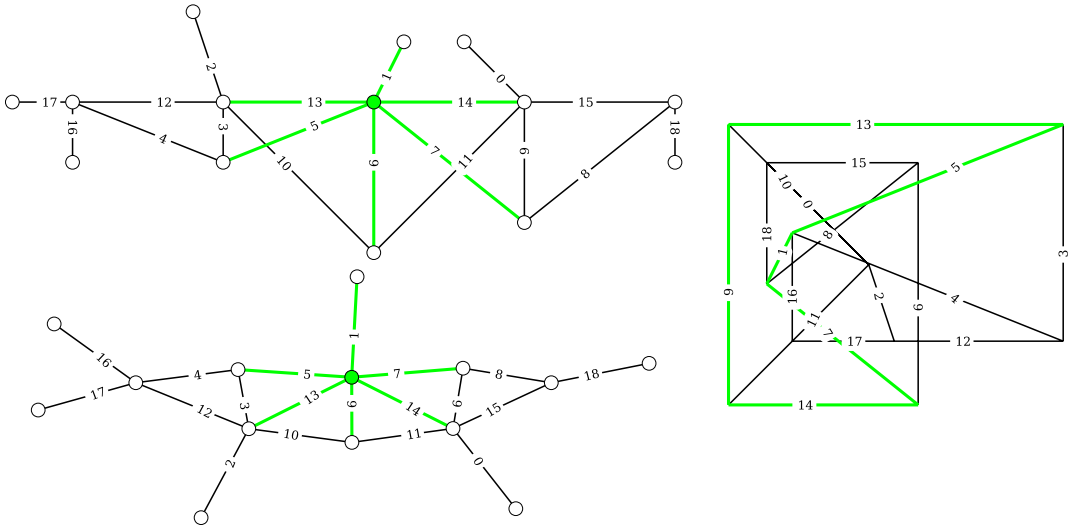

Fig. 13. Easy visualisation of reciprocal relation between form and force graphs based on topological data structures. Left: The form graph (top) and its planar drawing (bottom). Right: The reciprocal force graph. The edges converging at the selected node (marked in green) in the form graph form a closed polygon in the force graph, representing the equilibrium of forces at that node. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

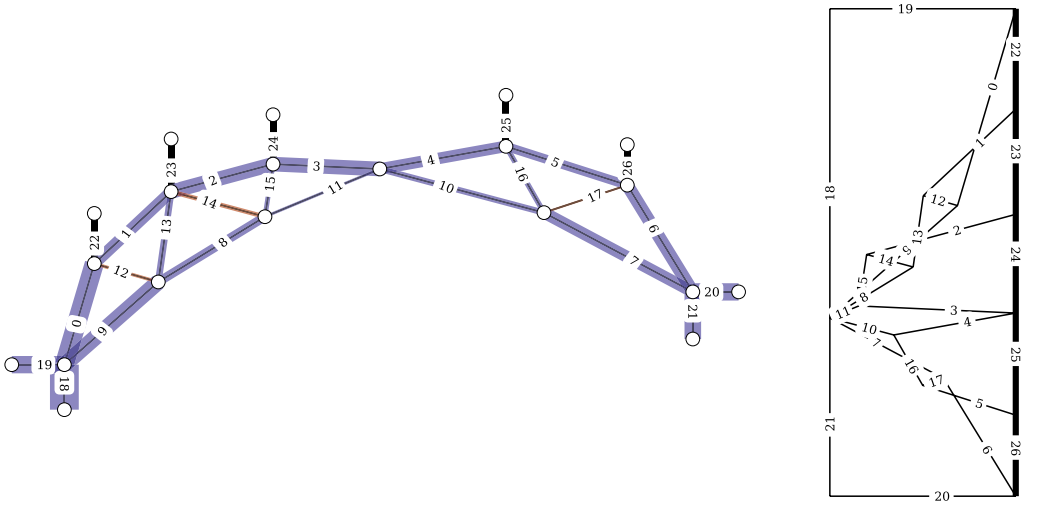

Fig. 14. Reciprocal form (left) and force (right) graphs of a three-hinged, trussed frame.

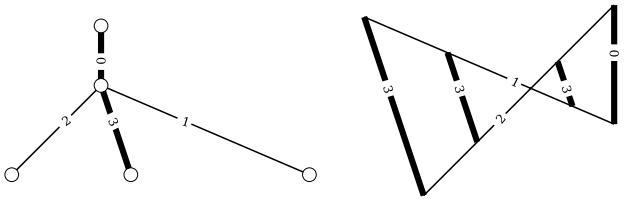

Fig. 15. Left: The form graph of an indeterminate, planar, three-bar structure. Right: Its reciprocal force graph depicting three possible states of self-stress.

and can only be in equilibrium in this geometry and for the given loading conditions (i.e. the given directions of external loads) for one combination of internal and external forces, which is unique up to a scale factor. There should thus be only one degree of freedom. Indeed, SVD of the equilibrium matrix of the form graph yields $m = 0$ and $k = 1$. As discussed in Section 2.6, any of the edges can be selected as the independent edge of which the force density may be chosen freely. The solution in Fig. 16 results from $q_{18} = -6$. It provides the required values of the applied loads for which the spline is in equilibrium in this geometry such that the reaction force at the left support is exactly 6 kN. Note that edges 3 and 4 are in tension (red), whereas the other edges corresponding to applied loads are in compression. This is reflected by a change in direction of the edges of the load line in the force diagram. The 'tension' loads are in fact upward loads which support the 'dimple' in the freeform arch.

Fig. 17 depicts the reciprocal form and force graphs of a spline with circular geometry and applied vertical loads. The horizontal and vertical components of the external forces at the supports have been included such that they can be controlled explicitly. Also in this case, SVD($\mathbf{A}$) of the form graph returns $m = 0$ and $k = 1$, indicating there is only one degree of freedom. Again, for any choice of force density of any one of the edges the force distribution is thus identical up to a scale factor.

The solution shown in Fig. 17 is the result of choosing $q_1 = H$, representing the horizontal force at the supports. Note that the result clearly visualises that in order for this circular arch to be funicular for a set of gravitational (i.e. downward, vertical) loads, the loads should be distributed as seen in the force graph: decreasing towards the middle of the arch.

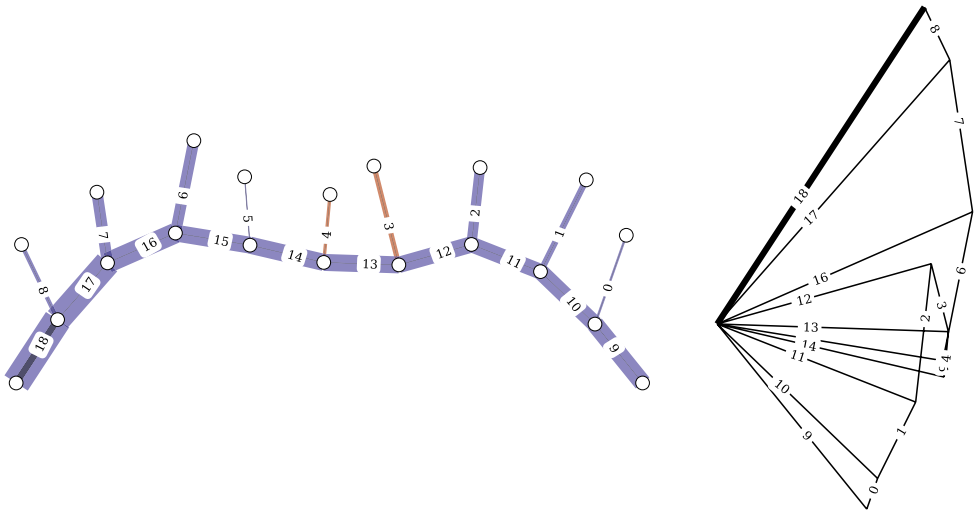

Fig. 16. Left: The form graph of a freeform discretised spline with a set of applied loads with given direction. Right: The reciprocal force graph as a result of choosing $q_{18} = -6$. Red edges of the form graph are in tension, blue edges in compression. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

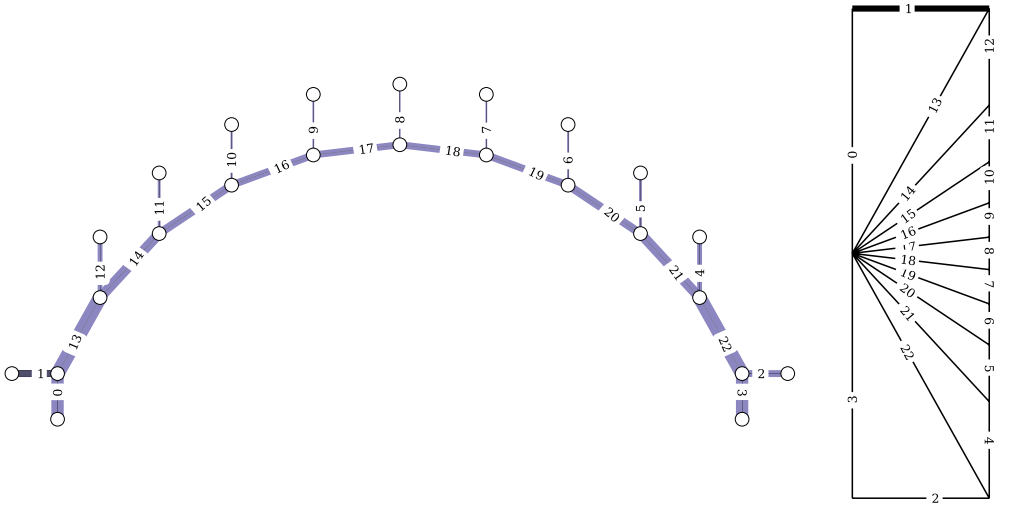

Fig. 17. Left: A form graph consisting of a discretised semicircular arc, applied vertical loading and reaction forces. Right: the reciprocal force graph of the form graph.

4.4. Self-stressed equilibrium nets

The last three examples deal with planar, self-stressed networks of forces, like spider webs [21].

Fig. 18 depicts the form graph of a network consisting of two rings of edges connected by two sets of crossing edges. Singular value decomposition of the equilibrium matrix of the form graph returns $m = 2$ and $k = 4$. Two inextensible mechanisms exist. Due to the axisymmetric configuration, the rings can rotate independently around the centre of the net, by an infinitesimal amount, before this rotation is prevented by the internal forces in the other elements. As a result, the number of states of self-stress is four, rather than the expected two. Indeed, four sets of edges can be tensioned independently, without changing the geometry of the net: 9-10-11-5-4-3, 0-1-2-8-7-6, 18-17-16-19 and 15-14-13-12. Assigning positive values to the force densities of a selected set of edges, for example, $q_8 = 1$, $q_{11} = 1$, $q_{15} = 1$ and $q_{19} = 0.5$, we obtain the result depicted in Fig. 18. Note that since $m > 0$, the matrix $\mathbf{A}_d$ is not square. Therefore, the full set of force densities was calculated with (15) rather than (13), as discussed in Section 2.6.

The example depicted in Fig. 19 represents the reciprocal form and force graphs of a self-stressed network with its edges arranged in an orthogonal grid. Here, SVD($\mathbf{A}$) yields $m = 0$ and $k = 5$. Note that this is yet another situation in which the independent edges cannot be chosen freely. The grid consists of five sets of edges forming straight continuous lines. Due to the (perfect) orthogonality of the grid, these sets can be tensioned independently, without affecting the forces in the others. Therefore, only selections of edges containing exactly one edge of each of these series are valid. RREF($\mathbf{A}$) identifies edges 2, 5, 8, 12, 16, which indeed contains one edge of each of the sets forming straight orthogonal lines. Choosing $q_2 = -0.5$, $q_5 = -1.0$, $q_8 = -0.5$, $q_{12} = -1.0$ and $q_{16} = -1.0$ we obtain the result of Fig. 19.

The last example demonstrates that it is not always straightforward to determine the number of independent edges and identify a possible set. The form graph in Fig. 20 represents an irregular network of bars and nodes. The system has 11 independent edges. A possible independent set has been determined by GJE of the equilibrium matrix and consists of the edges 23, 29, 32, 34, 35, 36, 37, 38, 39, 41, 42. The force densities of these edges have been determined by trial-and-error such that a compression only equilibrium is obtained.

As a final remark, the authors would like to point out the similarity between the form and force graphs of the last example and the *primal* and *dual grid* used in *Thrust Network Analysis* [12]. Indeed, if the form graph is interpreted as the horizontal projection

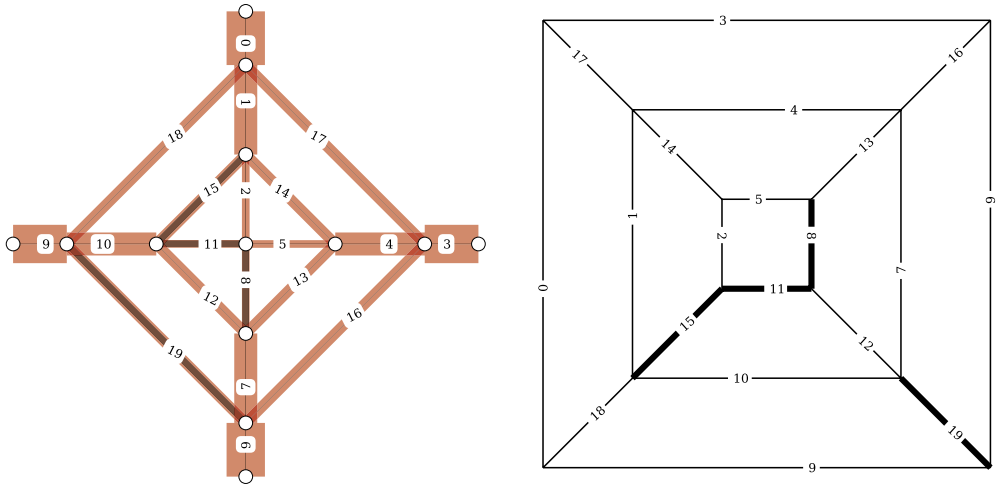

Fig. 18. Reciprocal form (left) and force (right) graphs of a self-stressed network with two concentric rings of edges in an axisymmetric configuration. The network is in tension-only equilibrium indicated by the red colour of the edges. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

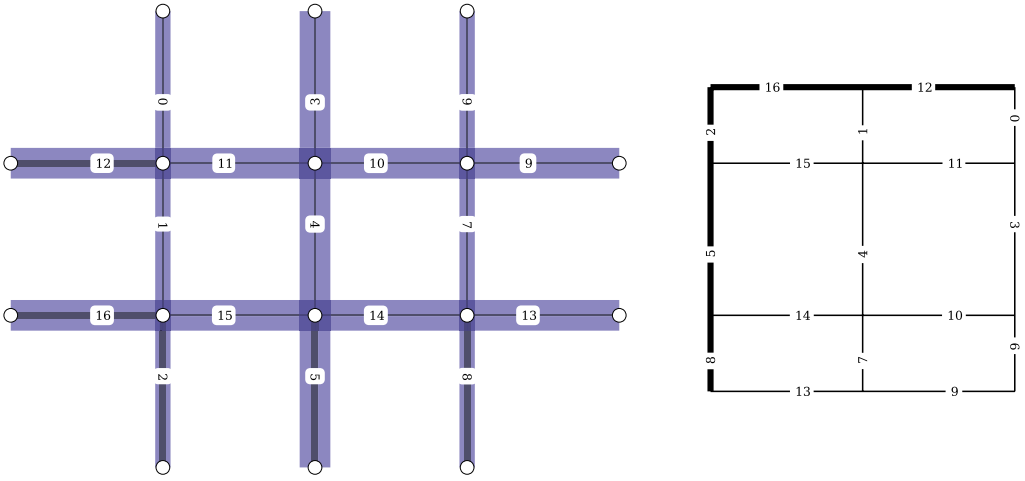

Fig. 19. Reciprocal form (left) and force (right) graphs of a self-stressed, orthogonal grid of forces in compression-only equilibrium. The grid is in compression-only equilibrium indicated by the blue colour of the edges. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

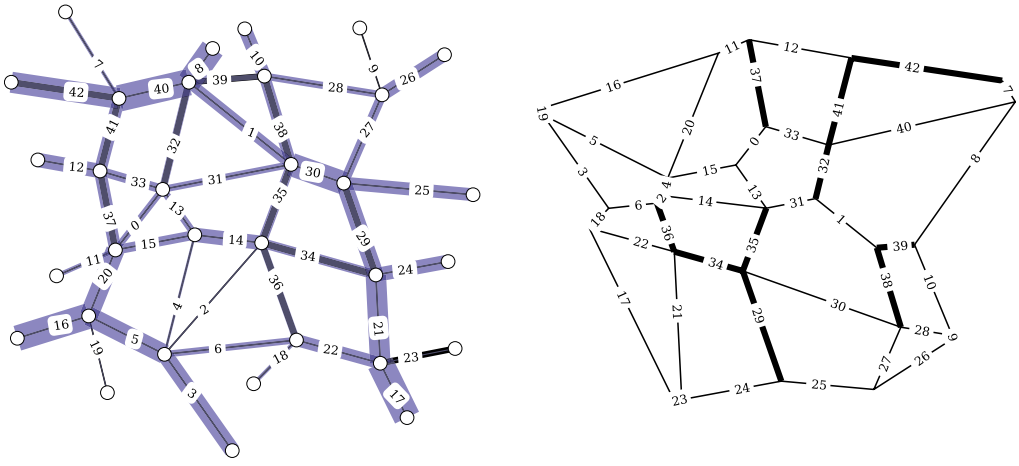

Fig. 20. Reciprocal form (left) and force (right) graphs of an irregular grid of forces in compression-only equilibrium. The grid is in compression-only equilibrium indicated by the blue colour of the edges. In this geometry, the system has 11 independent edges. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of a three-dimensional network of forces in equilibrium with vertical loads applied to its vertices (thus disappearing in the projection), then the reciprocal form and force graphs represent the horizontal equilibrium of this system of forces. Different states of self-stress of the form graph obtained by choosing different values for the force densities of a possible set of independent branches correspond to different three-dimensional equilibrium networks for the given loading, all with the same horizontal projection. The identification of the independent edges of a self-stressed network lies at the basis of techniques such as "best-fit TNA" [37] and the design of cable net formwork for concrete shells [38].

5. Conclusions

This paper presented a general, non-procedural approach to graphical analysis of two-dimensional structures.

In Section 2.2, we discussed the interpretation of the form and force diagrams of graphic statics as reciprocal graphs. We described how an algebraic formulation of the reciprocal relation between these two graphs results in the equilibrium equations of an unloaded network that is equivalent to the structural system represented by the form graph (Section 2.3). We furthermore showed how the possible states of equilibrium of this structural system can be controlled using the force densities of the system's free or independent edges, which can be identified by rank analysis using singular value decomposition and Gauss–Jordan elimination of the equilibrium matrix of the unloaded network (Section 2.6).

In Section 3.2, we discussed how the topology of the force graph can be derived from a planar straight-line drawing of the form graph using a wall following maze solving algorithm. We furthermore described an algorithm for generating planar straight-line drawings of (planar) form graphs based on the repetitive application of a force-driven layout method in combination with a crossing-edges check.

A computational setup allowing the presented approach to be used as back-end for a graphic statics application in an interactive CAD environment was described in Section 3.

Through a series of examples (Section 4), we showed how an implementation of the computational setup can be used to investigate the equilibrium of different types of structural systems, including a fink truss, a three-hinged arch, an indeterminate structure, funicular systems and self-stressed networks.

Finally, the relation to advanced form-finding and analysis techniques for three-dimensional equilibrium structures were briefly discussed.

References

[1] Cremona L. Graphical statics: two treatises on the graphical calculus and reciprocal figures in graphical statics [Thomas Hudson Beare Trans.]. Oxford: Clarendon Press; 1890.
[2] Maxwell JC. On reciprocal figures and diagrams of forces. Phil Mag J Ser 1864; 4(27):250–61.
[3] Maxwell JC. On reciprocal diagrams in space and their relation to Airy's funtion of stress. Proc Lond Math Soc 1869;2:58–60.
[4] Wolfe WS. Graphical analysis: a text book on graphic statics. New York, NY, USA: McGraw-Hill Book Co. Inc.; 1921.
[5] Active Statics. website; 2003, Last visited on 26.02.2014; http://acg.media.mit.edu/people/simong/statics.
[6] Block P, Ciblac T, Ochsendorf J. Real-time limit analysis of vaulted masonry buildings. Comput. Struct. 2006;84:1841–52.
[7] eQUILIBRIUM, website; 2011, Last visited on 26.02.2014; http://block.arch.ethz.ch/equilibrium.
[8] Van Mele T, Rippmann M, Lachauer L, Block P. Geometry-based understanding of structures. J Int Assoc Shell Spatial Struct 2012;53(2):285–95.
[9] Shearer MS. Analyzing and creating forms: rapid generation of graphic statics solutions through rhinoscript [Master Thesis], 2010.
[10] Lachauer L, Jungjohann H, Kotnik T. Interactive parametric tools for structural design, In: Proceedings of the IABSE-IASS Symposium 2011, London, UK, 2011, p. 199–, Full paper on CD-ROM.
[11] Allen E, Zalewski W. Form and forces: designing efficient and expressive structures. New York: Wiley; 2009.
[12] Block P. Thrust network analysis: exploring three-dimensional equilibrium [Ph.D. thesis]. Cambridge, MA, USA: Massachusetts Institute of Technology; 2009..
[13] Bow RH. Economics of construction in relation to framed structures. London: Spon; 1873.
[14] Micheletti A. On generalized reciprocal diagrams for self-stressed frameworks. Internat J Space Struct 2008;23(3):153–66.
[15] Schek HJ. The force density method for form finding and computation of general networks. Comput Methods Appl Mech Eng 1974;3(1):115–34.
[16] Strang G. Linear algebra and its appplications. 4th ed Belmont, CA, USA: Brooks/Cole, Cengage Learning; 2005.
[17] Pellegrino S. Structural computations with the singular value decomposition of the equilibrium matrix. Internat J Solids Struct 1993;30(21):3025–35.
[18] Nocedal J, Wright SJ. Numerical optimization. Springer; 2000.
[19] Harary F. Planarity. In: Graph theory. Reading, MA: Addison Wesley; 1994. p. 102–25. [chapter 11].
[20] Bondy J, Murty USR. Graph theory. In: Graduate texts in mathematics. Springer; 2008.
[21] Ash P, Bolker E, Crapo H, Whiteley W. Covex polyhedra, Dirichelet tessalations, spider webs. In: Senechal M, Fleck G, editors. Shaping Space: A Polyhedral Approach. Boston: Birkhaüser; 1988. p. 231–50. [chapter 17].
[22] Fáry I. On straight line representation of planar graphs. Acta Sci. Math. 1948; 11:229–33.
[23] Calladine CR. Buckminster Fuller's 'Tensegrity' structures and Clerk Maxwell's rules for the construction of stiff frames. Internat J Solids Struct 1978;14: 161–72.
[24] Pellegrino S, Calladine CR. Matrix analysis of statically and kinematically indeterminate frameworks. Internat J Solids Struct 1986;22(4):409–28.
[25] Python. website; 1990. Last visited on 26.02.2014; http://www.python.org.
[26] Hopcroft J, Tarjan R. Efficient planarity testing. J Assoc Comput Mach 1974; 21(4):549–68.
[27] Shih WK, Hsu WL. A new planarity test. Theoret Comput Sci 1993;223(1–2): 179–91.
[28] Boyer JM, Myrvold WJ. On the cutting edge: simplified O(n) planarity by edge addition. J Graph Algorithms Appl 2004;8(3):241–73.
[29] de Fraisseix H, Pach J, Pollack R. How to draw a planar graph on a grid. Combinatorica 1990;10(1):41–51.
[30] Schnyder W. Embedding planar graphs on the grid, In: Proc. 1st ACM-SIAM Sympos. Discrete Algorithms, 1990, p. 138–148.
[31] Tamassia R, editor. Handbook of graph drawing and visualization. Boca Raton, FL, USA: Chapman and Hall/CRC Press; 2014.
[32] Sage, website, 2005, Last visited on 26.02.2014; http://www.sagemath.org.
[33] Kobourov SG. Spring embedders and force directed graph drawing algorithms, arXiv.org; 2012. arXiv:1201.3011, http://arxiv.org/abs/1201.3011.
[34] Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. Software—practice and experience 1991;21(11):1129–64.
[35] NetworkX, website; 2013, Last visited on 26.02.2014; http://networkx.github.io.
[36] Scipy, website; 2013, Last visited on 26.02.2014; http://www.scipy.org.
[37] Block P, Lachauer L. Closest-fit, compression-only solutions for free-form shells, In: Proceedings of the IABSE-IASS Symposium 2011, London, UK, 2011, p. 108. Full paper on CD-ROM.
[38] Van Mele T, Block P. Novel form finding method for fabric formwork for concrete shells. J Int Assoc Shell and Spatial Struct 2011;52(4):217–24.