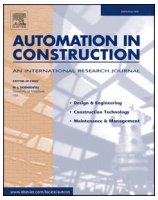
Automation in Construction 166 (2024) 105519



Contents lists available at ScienceDirect

Automation in Construction

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





Computational form-finding of single-layer spatial structures: Constructing force diagrams with Vector-based Graphic Statics and guiding trails

Yuchi Shen [a,*], Jean-Philippe Jasienski [b], Patrick Ole Ohlbrock [c], Pierluigi D'Acunto [c,d]

[a] Southeast University, School of Architecture, Sipailou 2#, Nanjing, China
[b] UCLOUVAIN, LOCI - LAB - Structures & Technologies, Place du Levant, 1 - 1348, Louvain-la-Neuve, Belgium
[c] Technical University of Munich, TUM School of Engineering and Design, Professorship of Structural Design, Arcisstraße 21, 80333 Munich, Germany
[d] Technical University of Munich, Germany, TUM Institute for Advanced Study, Lichtenbergstraße 2a, 85748 Garching, Germany

ARTICLE INFO

Keywords:
Structural design
Vector-based graphic statics
3D graphic statics
Single-layer spatial structures
Form diagram
Force diagram
Graph theory
Load-path
Planar graph
Planarization

ABSTRACT

This article extends the Vector-based Graphic Statics (VGS) computational framework to design single-layer spatial (SLS) structures. It introduces a novel computational workflow that produces ordered and controllable vector-based force diagrams for real-time interactive design. Key advancements include an algorithm for identifying guiding trails within the form diagram's underlying graph to control its planarization, an augmented half-edge (AHE) data structure for seamless force diagram assembly, and an interface for generating force diagrams using the Combinatorial Equilibrium Modelling (CEM) method. These advancements are demonstrated through practical design cases using the enhanced VGS framework, aiming to improve collaboration between structural engineers and architects by providing a direct and visual link between form and forces.

1. Introduction

1.1. Single-layer spatial structures

Single-layer spatial (SLS) structures represent a broad category of structures characterized by relatively thin cross-sections and their capacity to span significant distances. This study defines "SLS structures" to include various structural typologies such as cable nets, domes, membrane shells, grid shells, and folded plate structures. These structures are notably lightweight, attributable to their high surface-to-volume ratio, and they withstand applied loads through in-plane forces along the normal axis of the surface sections [1]. As form-active structures, SLS structures resist applied loads through their geometry rather than material accumulation, embodying the principle of resistance-through-form [2,3]. This research focuses on the form-finding of SLS structures using the Vector-based Graphic Statics (VGS) framework [4,5]. In this context, SLS structures are modeled as spatial networks of linear elements subject to axial internal forces, with external forces applied at the structural nodes.

1.2. Computational form-finding

Form-finding is a design approach that employs physical or digital models to generate the shape of form-active structures in static equilibrium, given specific boundary conditions and applied loads [1,6]. This process enables the creation of structural forms that minimize material use by reducing bending and shear forces, aligning with designated design criteria. Existing methods for computational form-finding can be generally grouped into three main categories [2,7,8]:

1. *Stiffness Matrix Methods*: These methods utilize standard elastic stiffness matrices, which are widely used for structural analysis and optimization. Examples include the *Finite Element Method (FEM)* [9,10] and the *Bi-directional Evolutionary Structural Optimization (BESO)* [11,12].
2. *Dynamic Equilibrium Methods*: This category addresses the dynamic equilibrium problem to achieve a steady-state solution equivalent to the static equilibrium solution. Examples include *Dynamic Relaxation* [13–15] and the *Particle-Spring Method* [16,17].
3. *Geometric Methods*: These methods only consider geometry and are, thus, both material and cross-section independent. Of relevance is

* Corresponding author.
E-mail addresses: yuchi_shen@seu.edu.cn (Y. Shen), jean-philippe.jasienski@uclouvain.be (J.-P. Jasienski), patrickole.ohlbrock@tum.de (P.O. Ohlbrock), pierluigi.dacunto@tum.de (P. D'Acunto).

https://doi.org/10.1016/j.autcon.2024.105519
Received 7 October 2023; Received in revised form 20 March 2024; Accepted 1 June 2024
Available online 14 July 2024
0926-5805/© 2024 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

the *Force Density Method (FDM)*, which introduced the concept of force density (i.e., the ratio of axial force to member length) to linearize the equilibrium problem [18,19]. *Direct Geometric Strategies (DGS)* use a non-linear process to minimize residual vectors, incorporating a preliminary geometric structure and initial internal force distribution to integrate diverse constraints into a single equilibrium system [20–22]. Additionally, computer-aided *Graphic Statics (GS)* methods are based on geometrical manipulations of form and force diagrams [4,8,23–25].

This paper focuses on the GS methods for their graphical and intuitive representation of the equilibrium, which is beneficial to both architects and structural engineers. The formalization of GS is attributed to the pioneering works of Culmann [26], Maxwell [27], Cremona [28], and Rankine [29]. GS relies on two interdependent diagrams representing the form of a structure in static equilibrium (form diagram) and the equilibrium of the forces at each structure node (force diagram), respectively. GS, widely used in the first half of the 20th century, was later replaced by analytical and numerical methods, primarily due to the advent of computers [30,31]. In the last two decades, GS has regained interest among researchers and practitioners. The extension of GS to 3D has been developed using two main approaches [32]: polyhedral-based [23,33,34] and vector-based [4,25]. The computational implementations of these methods offer powerful tools for direct control over form and forces throughout the design process, enabled by real-time geometric transformations between form and force diagrams [5,25,34–38].

1.3. Existing GS methods

This section briefly reviews the current GS-based methods for the form-finding of SLS structures (externally loaded or self-stressed) modeled as spatial networks of linear elements subject to only axial inner forces (Table 1). The methods presented are categorized according to the dimensionality of the form and force diagrams.

1.3.1. 2.5D methods

One of the most effective approaches to facilitate the assembly of the force diagram involves using a parallel projection of the structure onto a plane (generally horizontal). Force components parallel to the direction of the projection are nullified. The resolution of the 3D equilibrium is thus converted into a 2D problem. These techniques are labeled as *2.5D methods* throughout this article.

In the case of SLS structures, *2.5D methods* are constrained by the requirement that the direction of the external forces be perpendicular to the projection plane so that they are nullified when projected. Moreover, the structure should not be self-superposed, which would otherwise produce a projected 2D form diagram with self-overlapping elements. The design of SLS structures with 2.5D methods usually comprises two steps. The first step is to solve the 2D horizontal equilibrium of the structure projected onto the horizontal plane without external forces. In the second step, a form-finding method is implemented to define the heights of the nodes of the structure so that equilibrium with the external forces is attained.

The *Thrust Network Analysis (TNA)* [39] is one of the most prominent 2.5D methods. It uses a 2D force diagram to illustrate the structure's horizontal equilibrium. The force diagram is reciprocally linked to the corresponding 2D form diagram, which is the 2D projection of the structure on the horizontal plane [39]. The two diagrams are topologically planar (i.e., regarded as graphs, they can be drawn on a plane or a sphere without edge crossing). Reciprocity between the projected diagrams guarantees the equilibrium of the structure in the horizontal plane and allows for interactive manipulation of the diagrams. Afterward, the height of every node in the network is calculated using a linearized system of equilibrium equations. TNA facilitates the comprehension of the range of lower bound equilibrium by illustrating load paths throughout the structure in the horizontal plane, making it a particularly effective technique for vertically loaded compression or tension-only structures. [40,41]

Another example of a 2.5D method is the use of the *Airy Stress Function (ASF)* [42]. The self-stressed horizontal projection of an SLS structure can be seen as the orthographic projection of its corresponding 3D polyhedral Airy Stress Function. The internal axial forces in the 2D projection are proportional to the variation of the dihedral angle between the adjacent faces in the 3D polyhedral stress function [36,43,44]. The ASF method can be used to solve the self-stress state of a thrust network containing both tension and compression forces. The in-plane self-stress network can then be lifted to 3D with FDM to find the vertical components of the vertices of the structure [35,45].

1.3.2. 3D methods

According to Rankine's construction of 3D reciprocal diagrams [29], the equilibrium of a node in a 3D polyhedral structure is represented by a closed polyhedral cell in the 3D polyhedral force diagram. This way, an SLS structure with several nodes can be designed by repeatedly subdividing the corresponding 3D polyhedral force diagram [23,38,46]. However, for an SLS structure with medium to high network resolution, the elements constituting the reciprocal 3D polyhedral force diagram may be difficult to discern due to the hidden overlapping faces of the

Table 1
A review of existing GS-based methods for designing SLS structures.

GS-based Method	Dimensionality	Approach	Force diagram	Interactive Design	References
1. Thrust Network Analysis (TNA)	2.5D	The projected network's horizontal equilibrium in the 2D plane is generated using 2D form and force diagrams. Node heights are calculated numerically.	2D, topologically planar, force diagram	Transformation of 2D form and force diagrams	[39–41]
2. Airy Stress Function (ASF)	2.5D	The projected network's horizontal equilibrium in the 2D plane is generated using 2D form and force diagrams, which are projections of 3D stress functions. Node heights are calculated numerically.	2D force diagram as a projection of 3D stress function	Transformation of 2D form and force diagrams or 3D stress functions	[35,36,42,43,45]
3. Polyhedron-based Method	3D	The equilibrium of a 3D polyhedral network is determined by a 3D polyhedral force diagram, with each node's equilibrium expressed by a closed polyhedral cell.	3D polyhedral force diagram	Transformation of the 3D polyhedral force diagram and subdivision of its polyhedral cells	[23,29,34,38,46–49]
4. Vector-based Graphic Statics (VGS)	3D	The equilibrium of a 3D network is determined by a 3D vector-based force diagram, with each node's equilibrium expressed by a closed cycle of force vectors.	3D vector-based force diagram	Transformation of 3D vector-based form and force diagrams	[4,5,25,27,32,50]
4a. Combinatorial Equilibrium Modelling (CEM)	3D	The equilibrium of a 3D network is determined based on an input topology diagram and edges' force magnitudes and lengths.	3D force diagram generated via VGS	Adjustment of the equilibrium state by transforming the topology diagram and varying the edges' force magnitudes and lengths	[8,51–53]

constituent polyhedral cells. One way of addressing this issue is to synthesize 3D form and force diagrams in one geometric construction, the *Minkowski Sum* [47,48], which also intuitively illustrates the load path of the structure using volume – as the product of edge length and internal force magnitude [49].

In the *Vector-based Graphic Statics (VGS)* [4,5,25,27,32,50] approach, which is the focus of this paper, both the form and force diagrams are made of vectors. The corresponding vectors in the two diagrams are parallel. In this case, the equilibrium of a node in a vector-based 3D structure is represented by a closed cycle of force vectors in the force diagram. As a vector-based form-finding approach, the Combinatorial Equilibrium Modelling (CEM) method [8,51,52] offers a strategy for creating SLS structures in static equilibrium by controlling their underlying graph [53]. When combined with VGS, the CEM method can directly generate form and force diagrams.

2. Problem statement and objectives

2.1. Problem statement

To take advantage of VGS for the design of SLS structures, the 3D vector-based force diagrams must be organized to make them easy to read and understand. Preserving certain topological and geometric features of the structures, such as symmetries and hierarchical organization, is helpful in this context. The prerequisite for constructing the 3D vector-based force diagram of a given 3D form diagram is to planarize the underlying graph of the form diagram, as discussed in Section 3.1, which is generally not planar [4]. A planar graph secures the interdependency of form and force diagrams in static equilibrium.

A computational method was recently published to address the problem of automated graph planarization [25]. It includes a set of heuristic algorithms that manipulate and planarize the underlying graph of the form diagram, producing valid 3D force diagrams. However, at this stage, the method does not account for the specificities of the form diagram's topology or geometry during its graph's planarization. Hence, the method produces valid but sometimes difficult-to-read force diagrams, as their topologies (and thus force diagram configurations) do not correspond to those of their form diagrams. Direct perception of the graphical information in the VGS diagrams is critical for design purposes.

The potential advantages of a specific implementation of the VGS approach for SLS structures are as follows:

a) Unlike 2.5D methods, applied external forces would not be limited to one direction, enabling analysis of a broader range of SLS structures, including pre-stressed cable nets, compression-only grid shells, pneumatic structures, and tensegrity structures, using vector-based diagrams under various load cases.
b) The topological and geometric features of SLS structures could inform the topology and geometry of the vector-based force diagrams.
c) The *VGS tool* [25] enables interdependent transformations of the form and force diagrams for SLS structures in static equilibrium.

2.2. Objectives and contributions

To overcome the challenges highlighted in Section 2.1, this research aims to enhance the VGS approach's functionalities by introducing a dedicated procedure for SLS structures. This extension produces specifically ordered 3D vector-based force diagrams, improving their visual connection to the form diagram and supporting real-time interdependent transformations of the diagrams. Compared to 2.5D methods, the design space is expanded, and computational efficiency for these structure types is increased. In this regard, several sub-goals are achieved:

a) Introduce a new data type (i.e., *guiding trails*) that encapsulates the SLS structures' topological and geometric characteristics.
b) Develop an algorithm that planarizes the form diagram's underlying graph, accounting for its topological and geometric characteristics using guiding trails.
c) Propose an efficient graph-based data structure to assemble the force diagram from the planarized graph, ensuring compatibility with the CEM method.
d) Highlight the benefits of these developments, which increase the design freedom and expand the range of compatible SLS structures.

The main contribution of the proposed method is illustrated in Fig. 1, where a comparison is drawn between the outcomes of the new approach and those of the conventional VGS method (Jasienski et al., 2023 [25]). The comparison is based on two key aspects: calculation efficiency and the clarity of the force diagram. The new data structure reduces the calculation complexity of conventional VGS from $O(n^3)$ to $O(n^2)$. Consequently, the efficiency of the planarization procedure is evidenced by a 96% reduction in computation time for an 11×11 orthogonal network (226 ms vs 21,700 ms), as shown in Fig. 1 (all experiments presented in this article were performed on a machine with a 12th Gen Intel(R) Core(TM) i7-12700H CPU). Simultaneously, the force diagram's organization is increasingly improved, enhancing readability and making its relationship to the form diagram more direct.

2.3. Outline

The remainder of this article is organized as follows. Section 3 briefly reviews the current graph-based techniques within the VGS framework, outlining various planarization methods. Section 4 offers a detailed examination of the computational implementation of VGS for SLS structures. It covers the algorithms for creating specific vector-based force diagrams, including automatically generated guiding trails, illustrated with examples and condensed code snippets. An extension of the half-edge data structure is also introduced to tackle specific force diagrams construction challenges. Section 5 demonstrates the design potential of the presented framework through case studies of various SLS structures. Finally, the Conclusion section discusses enhancements in design freedom, visualization, and computational efficiency introduced by the presented method alongside directions for future research.

2.4. Notation

This paper follows a consistent and conventional notation used in previous publications on VGS (See Table 2). The red, blue, and green lines correspond to tensile, compressive, and external forces, respectively. Additionally, orange lines represent the newly introduced guiding trails, and gray lines denote the user-selected network boundary.

3. Theoretical backgrounds

3.1. Graph planarization approaches in VGS

The underlying graphs of most 3D structures represented as networks of linear edges are non-planar (i.e., they cannot be drawn on a plane or a sphere without edge crossing) [4]. Non-planar graphs lack dual graphs [54], a prerequisite for constructing Cremona reciprocals [55]. To address this issue, novel methods within the VGS framework have been introduced to planarize graphs by cutting the crossing edges and reconnecting them to one or more newly inserted auxiliary vertices [4]. The *VGS tool* [25] is the computational implementation of the VGS framework, automating graph planarization with algorithmic strategies named MED and QUAD (Fig. 2).

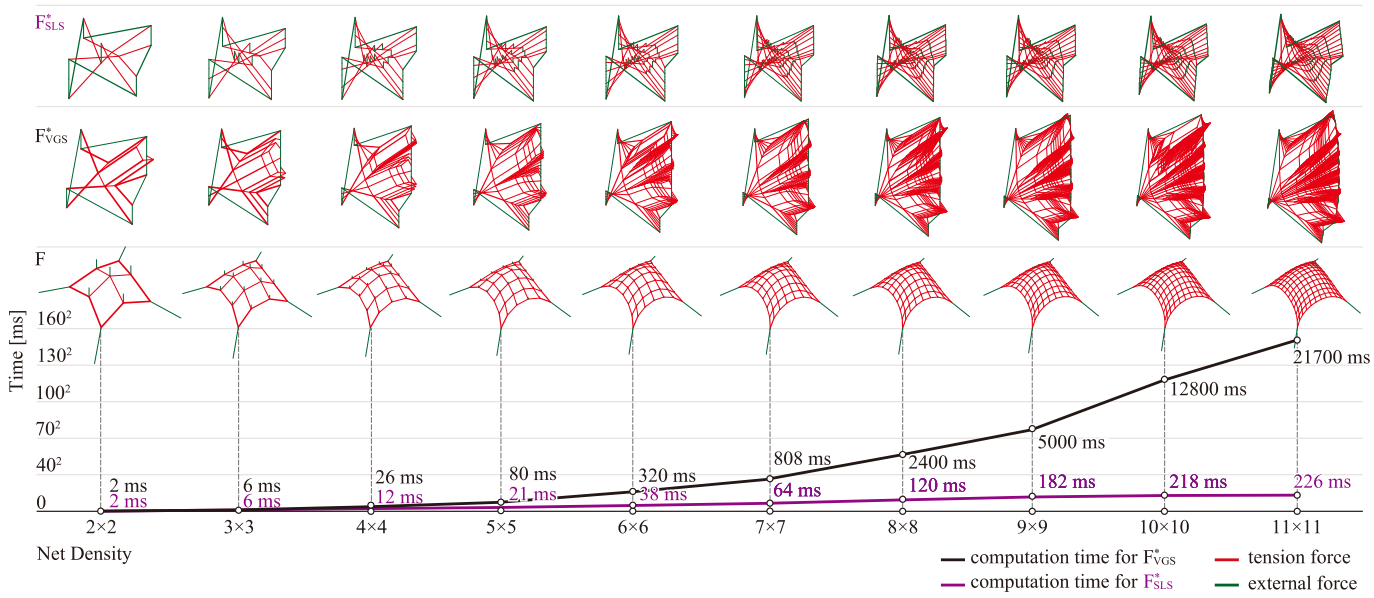


Fig. 1. Comparison between the conventional VGS algorithm [25] (force diagrams F^*_{VGS} in the second row, black line results in the third row) and the newly introduced one (force diagrams F^*_{SLS} in the first row, purple line results in the third row) in terms of computation time and orderliness of the force diagram of cable networks with different resolutions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
notation

	Topological diagram	Form diagram	Force diagram	Dual topological diagram	
	T, T_p, T_s, T_q, T_{sg}	F	F^*	T^*, T_p^*, T_s^*	
form edge	e^k_{i-j}	E_{i-j}	${E_{i-j}}^*$	${e^k_{i-j}}^*$	internal force vector
load edge	l^n_{i-j}	L_{i-j}	${L_{i-j}}^*$	${l^n_{i-j}}^*$	external force vector
vertex	v_i	V_i	$< V_i >^*$	$< v_i >^*$	vector polygon
vector polygon	$< v_i >$	$< V_i >$	V_i^*	v_i^*	vertex

F: The form diagram of the structure.
F^*: The force diagram of the structure.
T: The topological diagram (i.e., underlying graph) of F.
T^*: The dual topological diagram (i.e., dual graph) of T, when T is planar.
T_p: A planar embedding of T.
T_p^*: The dual topological diagram of T_p.
T_q: A subgraph of T corresponding to the complete set of load edges l_{i-j}.
T_s: A planar subgraph of T corresponding to the complete set of edges e_{i-j}. $T = T_s \cup T_q$.
T_s^*: The dual topological diagram of T_s.
T_{sg}: A guiding trails topological diagram as a subgraph of T_s highlighting the planarization sequences.

3.2. Customized planarization method in the VGS tool

The topologies of the graphs planarized by the MED and QUAD methods depend on how the algorithm executes the planarization. The user does not have specific control over this procedure. Therefore, the configurations of the resulting force diagrams are directly linked to the behavior of the chosen algorithm [5]. Taking into account the characteristics of the structure allows for the assembly of a specific configuration of the force diagram F^* that favors its readability and makes its relationship with the form diagram F explicit. For example, in Fig. 4, an externally loaded hyperbolic paraboloid based on a 6 × 6 network of edges has an underlying non-planar graph T due to the load edges attached to the inner vertices. A proposed manual planarization uses a procedure analogous to the one described in Fig. 2b. Load edges are sequentially allowed to enter and exit all intermediate vertices along topological paths (i.e. *guiding trails*) until they reach the boundary of the graph (Fig. 4c). As a result, T becomes planar without affecting the static equilibrium of the structure. Based on the configuration of the T_p (the planarized version of T), the corresponding force diagram F^* can be assembled. Even though the manual approach provides a maximum freedom to the user, it requires specific user's experience, and is time-consuming. For example, an n×n network with $(n-1)^2$ loads at the inner vertices would require at least $\sum_{s=1}^{\lfloor 0.5*n-1 \rfloor} 4s*(n-2s-1)$ steps to perform the manual planarization procedure. An automated procedure, therefore, would speed up the planarization and prove useful for solving complex cases without having to rely on the user's experience.

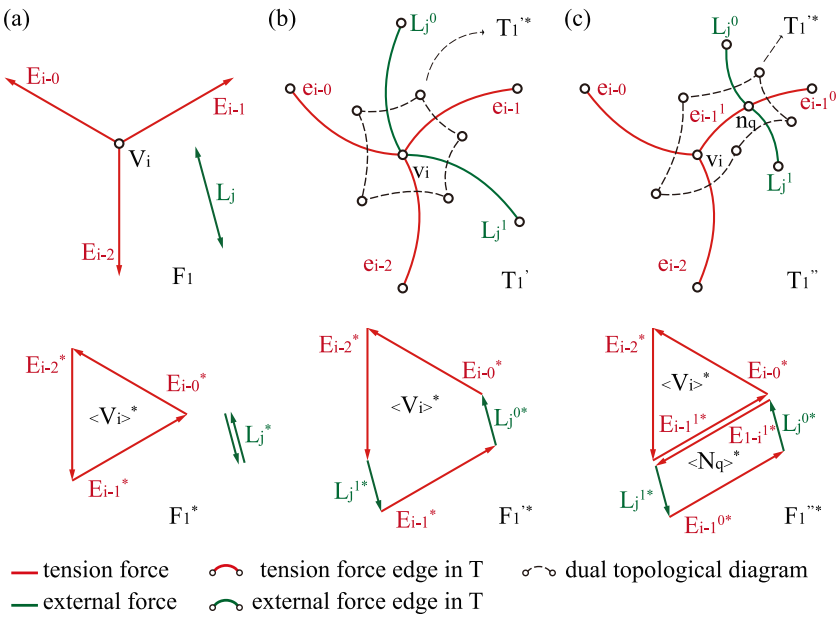


Fig. 2. Two strategies of graph planarization in the VGS tool.
(a) a structure in equilibrium F_1 with one vertex V_i and an external load L_j not applied on V_i; in the F_1^* they correspond to $< V_i >^*$ and a vector pair L_j^*. (b) In the MED planarization strategy, the two vectors of the pair L_j enter and exit vertex V_i and the vector pair L_j^* is merged into $< V_i >^*$ to become F_1^*. (c) In the QUAD planarization strategy, L_j intersects edge e_{i-1} at the newly introduced auxiliary vertex n_q and the vector pair L_j^* is merged into $< V_i >^*$, in which the E_{i-1}^* is duplicated. The three force diagrams are equivalent in terms of static equilibrium.

4. Method

The computational procedure to generate vector-based form and force diagrams for SLS structures is outlined in Fig. 3. In case T is non-planar, an algorithm based on the recursive Breadth-First Searching (BFS) is employed to analyze the network of an SLS structure and highlight the guiding trails. These form a branching data structure that defines the path of the edges corresponding to the external loads to planarize the graph. Subsequently, an algorithm that traverses the branches to generate the augmented half-edge data structure is introduced. Based on the resulting data structure, the 3D force diagram is constructed, activating the interactive form-finding process by interdependently modifying F and F^* using the VGS tool.

The improvements brought by the framework are illustrated in Fig. 4. The topology of the unloaded SLS structure is inherently a planar graph $T_s(T_s \subset T)$. Once loads and reactions at the vertices of T_s are added to T, the latter becomes a non-planar graph (Fig. 4a). In the existing VGS framework, the planarization algorithm calculates the shortest path in the graph to iteratively reintegrate external load edges to T_s until the load edges reach the graph boundary and T_s becomes planar (T_{QUAD}) [25]. This produces a force diagram F^*_{QUAD} (Fig. 4b) that is sometimes difficult to read mainly because the planarization sequence does not comply with the network configuration. The benefit of predefining the planarization sequence is presented in Fig. 4c. The arrangement of the load edges in T follows a topologically oriented logic. This results in the force diagram F^*_{SLS}, which is visually more related with the form diagram F compared to F^*_{QUAD}.

4.1. Planarization using guiding trails

The role of the guiding trails in constructing force diagrams can be demonstrated with a straightforward example (Fig. 5). The structure in Fig. 5a, subject to uniform loads, has a non-planar graph T. In Fig. 5b, the gray arrow indicates a guiding trail from v_4 to v_7, which directs the external force edge l_4 in T through v_7 to the boundary region (i.e., the topological region that contains all the external force edges). The edge l_4 is replaced by l'_4 and l''_4, following the MED procedure (Fig. 2b). This process enables the planarization of T to T_p (Fig. 5c). Since T_p is a planar graph, it encapsulates all information necessary for constructing the force polygons and force diagram (Fig. 5d, 5e). Therefore, the guiding trails link the SLS structure's inner vertices to boundary vertices, embodying the graph's topological traits within the force diagram's configuration. These trails span all vertices subject to external forces, facilitating the reintegration of edges responsible for non-planarity while maintaining the graph's planarity.

The computational mechanisms for constructing a force diagram with the guiding trails can be formulated as follows:

a) The underlying graph of the structure T can be expressed with the adjacency matrix U (Fig. 6):

$$U = \begin{array}{c} \\ v_i \\ v_j \\ \vdots \\ v_e \end{array} \begin{array}{c} \begin{array}{cccc} v_i & v_j & \cdots & v_e \end{array} \\ \begin{bmatrix} - & E_{i-j} & \cdots & L_i \\ E_{j-i} & - & \cdots & L_j \\ \vdots & \vdots & - & \vdots \\ L_i & L_j & \cdots & - \end{bmatrix} \end{array} \tag{1}$$

The edges in the matrix are sets of half-edges:

$$E_{i-j}\left\{e^0_{i-j} \ldots e^k_{i-j}\right\} \tag{2}$$

$$L_i\left\{l^0_{i-e} \ldots l^n_{i-e}\right\} \tag{3}$$

Where:

E_{i-j} is a set of half-edges e^k_{i-j} sharing the same endpoints v_i and v_j

L_i is a set of load edges l^n_{i-e} sharing the same endpoints v_i and v_e. v_e is not a real vertex in T but an auxiliary vertex that is representative of the equilibrium of the external forces [25].

b) The guiding trails include the sequences of vertices to planarize T that can be calculated with the method $dp()(7)$(Section 4.2) and are stored in the binary matrix U_p (Fig. 7):

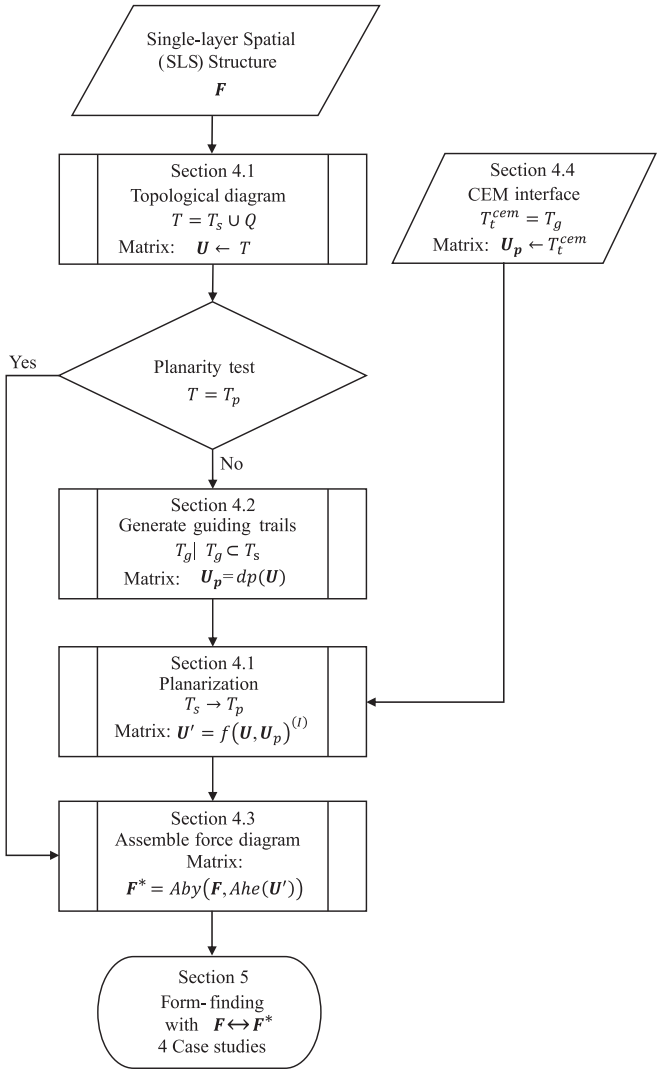


Fig. 3. The flowchart shows the computational procedure to build specific form and force diagrams for SLS structures within the VGS method.

$$U_p = \begin{array}{c} \\ v_i \\ v_j \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} v_i & v_j & \cdots & r \end{array} \\ \left[\begin{array}{cccc} - & u_{i-j} & \cdots & r_i \\ u_{j-i} & - & \cdots & r_j \\ \vdots & \vdots & - & \vdots \end{array}\right] \end{array} \tag{4}$$

$$u_{i-j} = \begin{cases} 0, & \textit{no trail from } v_i \textit{ to } v_j \\ 1, & \textit{exist one trail from } v_i \textit{ to } v_j \end{cases} \tag{5}$$

One extended column of U_p stores the vertex attribute (Fig. 7) in the branching data structure of the *guiding trails*:

$$r_i = \begin{cases} -1, & \textit{a root vertex in the tree} \\ 0, & \textit{an intermediate vertex in the tree} \\ 1, & \textit{a leaf} - \textit{end vertex in the tree} \end{cases} \tag{6}$$

To build the computational sequence (from leaf-end vertices to root vertices) based on U_p, a search routine $\widehat{S}()$(8) is illustrated in Fig. 8 that starts from one root vertex $v_i(r_i = -1)$ and follows the vertex adjacency $u_{i-j} = 1$ to pass through intermediate vertices $v_i(r_i = 0)$ until reaching the leaf-end vertices $v_i(r_i = 1)$. This routine produces a tree S^k as a three-dimensional array. Each root vertex in U_p corresponds to one S^k and all S^k consist of the complete set of computational sequences S. For each S^k,

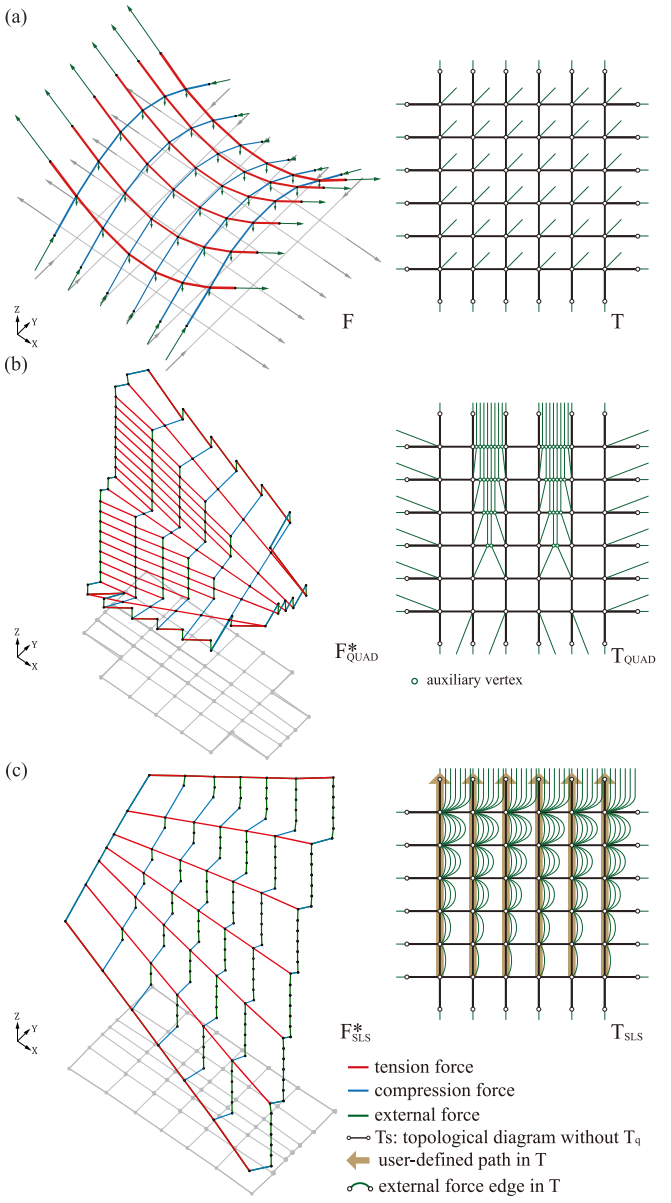


Fig. 4. An externally loaded hyperbolic paraboloid [56] mixing tension and compression internal forces and different configurations of related force diagrams (F^*_{QUAD},F^*_{SLS}). (a) The form diagram F and its graph $T = T_s \cup T_q$, where T_q is a subgraph containing external load. (b) The force diagram F^*_{QUAD} generated with QUAD method in VGS corresponds to the topology T_{QUAD}. (c) The force diagram F^*_{SLS} generated with the topology T_{SLS} based on the user-defined planarization sequece.

every vertex can be marked with $s^k_{[l,w]}$, in which l is the layer index in tree S^k and w is the vertex index in layer l. Moreover, every vertex can be also marked with $S^k_{[l,b,n]}$ using three indices: layer index l, branch index b, and vertex index n in branch b. Branch index b of a vertex is the w index of its parent vertex. This can be used to link the correspondence between any child vertex $v_c\left(S_{[l,b,n]} \rightarrow v_c\right)$ and its parent vertex $v_p\left(s_{[l-1,w]} \rightarrow v_p\right)$ with $w = b$.

$$U_p = dp(U) \tag{7}$$

$$S, s = \widehat{S}\left(U_p\right) \tag{8}$$

c) With the construction above, the algorithm can iteratively go through S by traversing $[l, b, n]$ to planarize all the load edges L_i, from the leaf-end vertices $(r_i = 1)$ to the root vertices $(r_i = -1)$ by passing

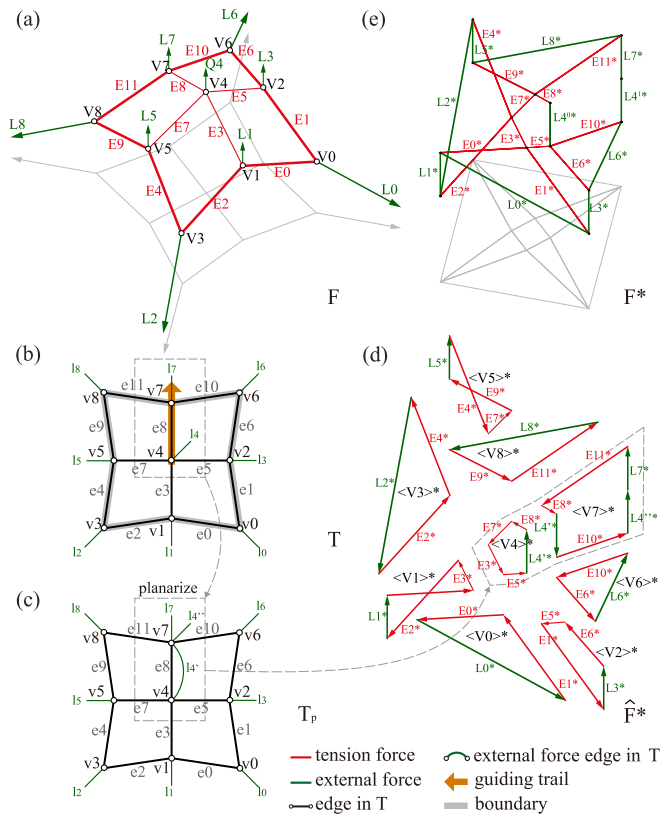


Fig. 5. A cable net subjected to upward point loads, serving as the primary example to demonstrate the construction of the force diagram using a guiding trail. (a) Step1: the form diagram; the non-planarity of the graph is due to the presence of the external force edge l4 connected to v4 (b) Step2: the structure's non-planar graph along with a designated guiding trail; (c) Step3: the graph planarized in accordance with the generated guiding trail; (d) Step4: the force polygons derived from the planarized graph; (e) Step5: The resultant force diagram

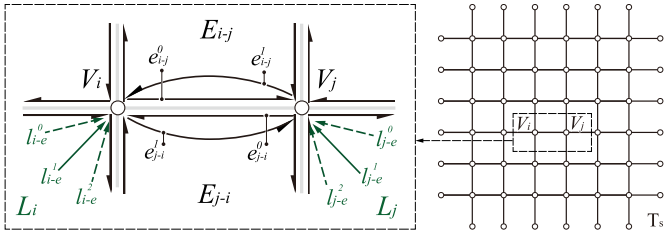


Fig. 6. Illustration of the notation used in the adjacency matrix U.

through the intermediate vertices ($r_i = 0$). This planarization process converts $T = T_s \cup T_q$ to a planar graph T_p by modifying T_q and adding it to T_s. In the iteration of $[l, b, n]$, the current vertex v_I and its parent vertex v_p and child vertices $V_c\{v_{c,0} \ldots v_{c,m}\}$ are:

$$v_I \leftarrow S_{[l,b,n]}, s_{[l,w]} \tag{9}$$

$$v_p \leftarrow s_{[l-1,b]} \tag{10}$$

$$V_c \leftarrow \{v_{c,0} | S_{[l+1,w,0]} \ldots v_{c,m} | S_{[l+1,w,m]}\} \tag{11}$$

The planarization procedure making $U \rightarrow U'$ in the iteration $[l, b, n]$ is:

$$E_{I-p} = \begin{cases} e^0_{I-p} + \{\hat{e}(L_{c,0}) \ldots \hat{e}(L_{c,m})\} + \hat{e}(L_I), & r_I = 0 \\ e^0_{I-p} + \hat{e}(L_I), & r_I = 1 \end{cases} \quad (E_{I-p} \in U') \tag{12}$$

$$L_I = \hat{e}(L_I) + \{\hat{e}(L_{c,0}) \ldots \hat{e}(L_{c,m})\}, \& r_I = -1 \ (L_I \in U')) \tag{13}$$

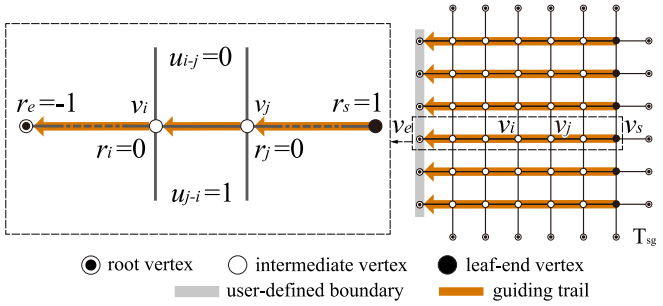


Fig. 7. Illustration of the guiding trail matrix U_p

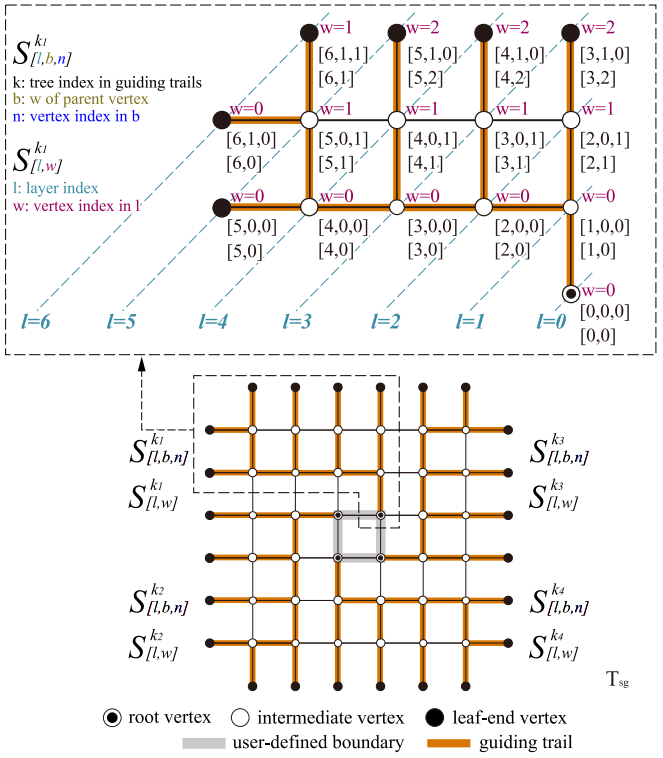


Fig. 8. Hierarchical calculation sequence S, s with tree-structured guiding trails

where the U' is a modified version of U throughout the planarization process ($U' = f(U, U_p)^I$). The vertex index I in iteration $[l, b, n]$ is achieved with the sequence definition in (9). $\hat{e}()$ is a converter that transfer load edges $L_i\{l^0_{i-e} \ldots l^k_{i-e}\}$ in the vertex v_i (i can be I, c) into half-edges $\{e^{d+1}_{i-p} \ldots e^{d+k+1}_{i-p}\}$ in E_{i-p} considering the existing edge duplication count d.

According to (12,13), when v_I is an intermediate vertex ($r_I = 0$), $E_{I-p}(E_{I-p} \in U')$ is the sum set of the half-edge e^0_{I-p} connected to its parent vertex, the load half-edges $\{\hat{e}(L_{c,0}) \ldots \hat{e}(L_{c,m})\}$ from all its child vertices and the loads on itself $\hat{e}(L_I)$. When v_I is a leaf-end vertex ($r_I = 1$), $U'.E_{I-p}$ is only considering the sum of e^0_{I-p} and $\hat{e}(L_I)$. When v_I is a root vertex ($r_I = -1$), $L_I(L_I \in U')$ is the sum set of $\hat{e}(L_I)$ and $\{\hat{e}(L_{c,0}) \ldots \hat{e}(L_{c,m})\}$. Code snippet 1 shows the corresponding computational procedure. After U' is achieved (see Section 4.3), the planarized graph T_p can be represented with an augmented half-edge structure (AHE) with the half-edge information in U'. With T_p, the assembly of the force diagram can be finally achieved.

Code snippet 1

Inputs:

F: Form diagram of the surface structure.

Paras:

T_s: The underlying graph without external edges.

U: Vertex adjacency matrix.

U_p: Guiding trail matrix.

v_i: The i index vertex in the surface network.

L_i: A set of loads applied to the vertex v_i.

S : One three-dimensional array that defines the computational sequence of vertices.

V_c: A set of child vertices of v_i.

v_c^m: The m index child vertex of v_i.

L_c: A set of load sets applying on the child vertices V_c.

L_c^m: The m-th set of loads applied to the child vertices V_c.

Outputs:

F^*: Force diagram

```
//01 Construct the vertex adjacency matrix
T(V, E) ← F
U ← T(V, E)
//02 Generate guiding trails
U_p ← dp(U) (see Section 4.2)
S_[l,b,n], S_[l,w] ← Ŝ(U_p)
//03 Planarization: modify T_q and add it to T_s
Foreach S_[l,b,n] in S:
    v_I, v_p, V_c ← S_[l,b,n]
    L_I ← {l^0_{I-e} ... l^k_{I-e}}
    L_c ← {}
    Foreach v_{c,m} in V_c:
        L_{c,m} ← {l^0_{c,m-e} ... l^k_{c,m-e}}
        L_c.add(L_{c,m})
    End
    U' ← U //copy the matrix for modification
    Select case v_I|r_I: //(E_{I-p} ∈ U')&(L_I ∈ U')
        case 0:
            E_{I-p} ← {e^0_{I-p}}
            E_{I-p}.add(ê(L_{c,0}) ... ê(L_{c,m}))
            E_{I-p}.add(ê(L_I))
            L_I ← {}
        case 1:
            E_{I-p} ← {e^0_{I-p}}
            E_{I-p}.add(ê(L_I))
            L_I ← {}
        case -1:
            L_I ← {ê(L_I)}
            L_I.add(ê(L_{c,0}) ... ê(L_{c,m}))
    End
End
//04 Assemble force diagram via augmented half-edge data structure
T_p* ← Ahe(U') (see Section 4.3)
F* ← Aby(F, T_p*) (see Section 4.3)
return F*
```

4.2. Generating guiding trails on SLS structure's graphs

In this subsection, we describe an algorithm $dp()$ that generates $U_p \in U$. For U_p to be valid, it must fulfill certain conditions so that all external force edges are reintegrated to the boundary of the network:

a) The root vertex of each trail should be positioned on the selected boundary of the graph. Guiding trails must be independent of each other, avoid vertex overlap, and form a directed branching structure without any self-cycle.
b) Every vertex within the graph should be visited only once by a guiding trail.

Beyond these two mandatory rules, the guiding trails should ideally follow the inherent topological characteristics of the structure. This contributes to a configuration of the force diagram that can be directly related to the topology of the form diagram. The overall structure of the presented algorithm that finds $U_p \in U$ of any discrete network is outlined in Code snippet 2.

The algorithm operates within the graph T_s. It makes use of the adjacency of the vertices U and simultaneously considers some geometric attributes of the form diagram F. Conventional Breath-First Search (BFS) techniques can generate a branching structure within a network and ensure no edges self-intersections. However, as the branching structure generated with BFS randomly decides the path from the viable links between ranked vertices, it lacks the capability to follow the topological characteristic inherent to the network pattern. The method suggested here leads to a directional path search approach. Each iteration of this method involves the subsequent steps:

a) Define the boundary vertices. In the first iteration, the boundary vertices $V^B \in T_s$ are defined via constructing the AHE data structure of T_s and prescribing one of the cycles in it. In the following iterations, all the visited vertices are marked as $v_i^T = true$.
b) Rank all the vertices (v_i^R) based on their topological distance from the boundary vertices with the BFS algorithm.
c) Highlight the guiding tails by connecting the vertices from the ones with lower rank v_i^R to the ones with higher rank $v_i^R = v_i^R + 1$. For each vertex, take the corresponding node in the form diagram and define the direction vectors $\vec{V}_I\left\{\vec{v}_{i-j}\ldots\vec{v}_{i-k}\right\}$ of the connected edges projected on the plane tangent to the node. Select the direction vector that forms the smallest angle with the previous trail vector and select the vertex connected to it. The search will stop when no adjacent vertex's rank is larger than the current.
d) Detect if every vertex in the network is already part of a trail $\forall v_i.T = true$. If not, define the visited vertices as boundary vertices and trigger another iteration of searching.

Fig. 9 illustrates two results due to an initial different boundary setting of the same network. In the first case, the boundary surrounds the network, and the root vertex count is 24. As the output, 20 guiding trails are generated as 4 root vertices have null trails (Fig. 9a). Only one iteration is required in this case, as the line-shaped trails can fill the whole graph. In the second case, the boundary is selected in the middle of the graph, which contains only 4 vertices (Fig. 9b). Therefore, the guiding trails start from the 4 vertices and are shaped with a branching structure to cover the entire graph. Logically, 4 iterations are employed to produce the 4 branching guiding trails. Fig. 10 illustrates the algorithm's performance implemented in two high-resolution and ruled pattern cases. The directional path search in the algorithm is used to find the inherent geometric rule in the pattern.

The algorithm detailed above demonstrates that the configuration of

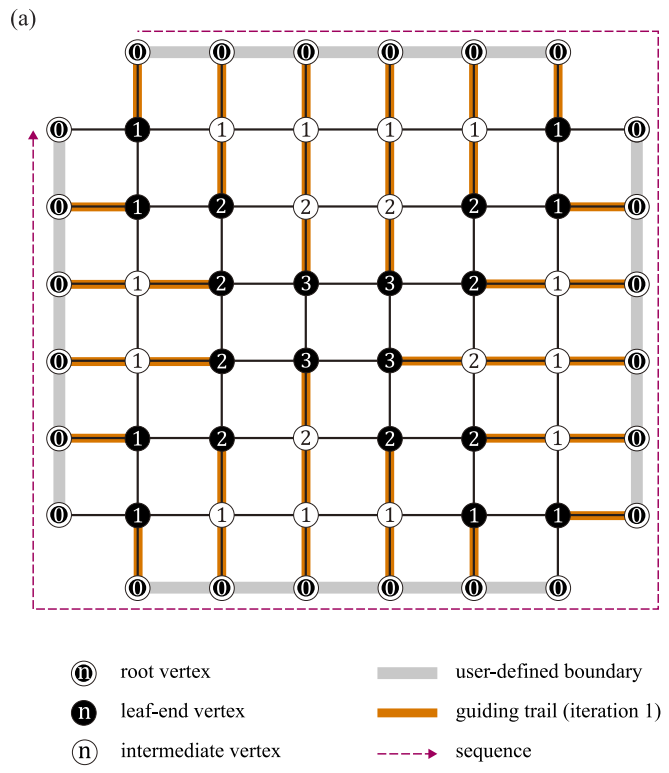


Fig. 9. (a) Boundary vertices defined outside the network: the guiding trails are generated with one iteration and are all lines. (b) Boundary vertices defined inside the network: The guiding trails generated require four iterations and are all branches. Vertices are marked with their ranks v_i^R.

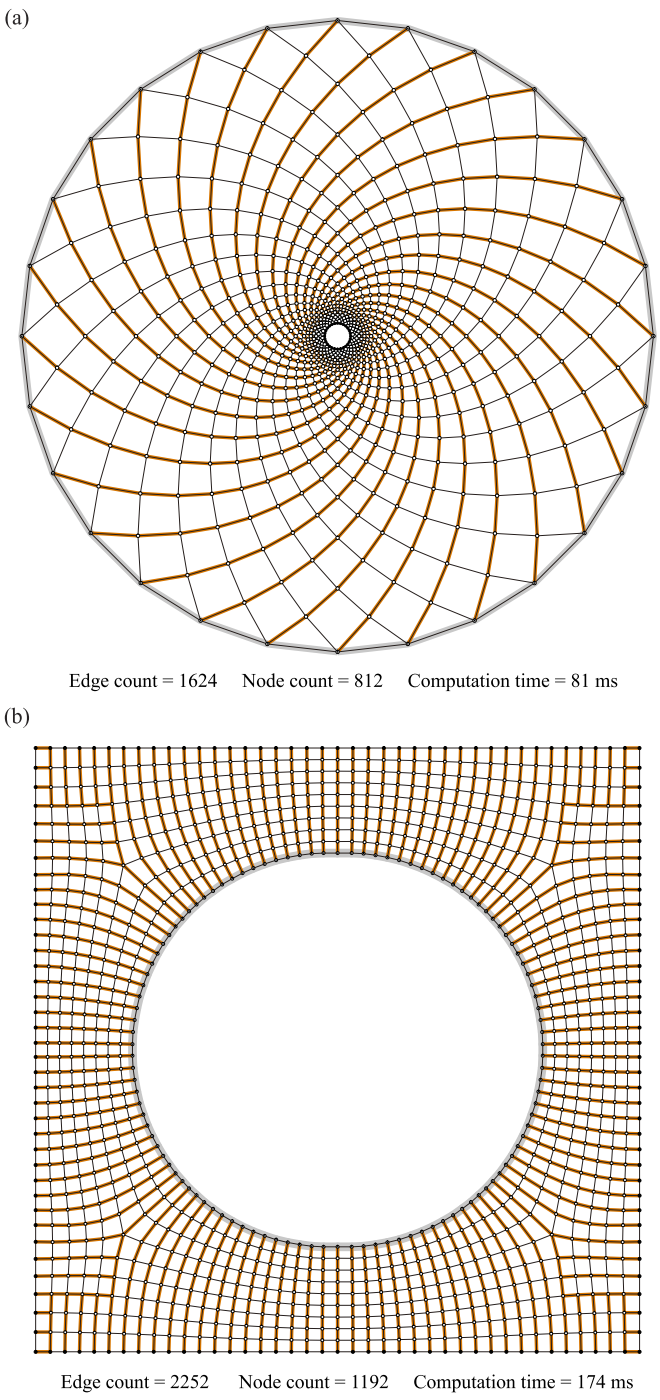


Fig. 10. The algorithm implemented on two high-resolution cases: (a) One spiral network with 1624 edges and 812 vertices and spends 81 ms to output guiding trails; (b) Another quadrilateral network with 2252 edges and 1192 vertices and spends 174 ms to output guiding trails in branch.

the guiding trails can be directed by predefining the network's boundary. This is achieved by designating vertices on the guiding trails as $v_i^T = true$ before the trail-connecting routine. Features such as symmetry can be achieved by predefining a symmetric boundary. Moreover, applying different guiding trail settings to the same form diagram results in distinct force diagrams, as illustrated in Figs. 4c and 12.

Code snippet 2

Inputs:

U: Vertex adjacency matrix

Paras:

${U_p}^0$: One initial guiding trail matrix that $\forall u_{i-j} = 0\ (u_{i-j} \in {U_p}^0)$ and $\forall r_i = 0\ (r_i \in {U_p}^0)$.

${V^{Ts}}$: The complete set of the vertices in T_s.

V^B: Vertices that are on the defined boundary of the network.

V^I: Vertices that are not on the defined boundary of the network.

v_i^T: Boolean value to mark whether the vertex v_i is already included in a trail.

v_i^R: Rank of v_i.

r: Integer value to record the current searching depth.

a: Boolean value to record whether a trail is found in the iteration.

V_l: A set of vertices as candidates for the next guiding trail link.

$\vec{V}_l\{\vec{v}_{l-j} \dots \vec{v}_{l-k}\}$: A set of unitized vectors that start from v_l and point toward the adjacent vertices in F.

Outputs:

U_p: Guiding trail matrix

```
//01 Define the boundary of the network
U_p^0 ← U
V^Ts → V^B ∪ V^I
foreach v_i,B in V^B: v_i,B^R ← 0; r_i,B ← −1(v_i,B ∈ U_p^0); v_i,B^T ← true
foreach v_i,I in V^I: v_i,I^R ← int.max; r_i,I ← 0(r_i,I ∈ U_p^0); v_i,I^T ← false
//02 Find the trails with network vertex ranking
While ∃v_i^T = false:
    //02-1. rank the network vertices
    r ← 0
    While ∃v_i^R = int.max:
        Foreach v_i^R = r in U:
            Foreach E_i−j in the v_i row of U:
                If v_j^R > r and |E_i−j| > 0:
                    v_j^R ← r + 1
                End if
            End
        End
        r = r + 1
    End
    //02-2. build guiding trails
    Foreach v_i^R = 0 in U:
        v_l ← v_i; a ← true; r ← 0
        While a = true:
            V⃗_l ← {}; V_l ← {}; a ← false;
            Foreach E_i−j in the v_l row of U:
                If |E_i−j| > 0 & v_j^R = r + 1 & v_j^T = false:
                V⃗_l.Add(F.v⃗_i−j); V_l.Add(v_j)
                a ← true
                End if
            End
            If a = true:
                v_s ← V_l.selectMinGra(V⃗_l.projectOnTanPlane)
                v_l^T ← true
                u_l−s ← 1 (u_l−s ∈ U_p^0)
                v_l ← v_s
            End if
            r ← r + 1
        End loop
    End
    //02-3. update the rank
    Foreach v_i^T:
        If v_i^T = true: v_i^R ← 0
        Else: v_i^R ← int.max
    End
End loop
//03 Output guiding trails
Foreach v_i in U_p^0:
    If ∀ u_i−j = 0(u_i−j ∈ v_i): r_i = 1(r_i ∈ U_p^0)
End
U_p ← U_p^0
return U_p
```

4.3. Force diagram assembly with augmented half-edge data structure

The guiding trail concept introduced in Section 4.1 determines a sequence in the branching structure to planarize the underlying graph of SLS structures. The output planar graph always contains multiple edges in topological connections ($|E_{i-j}| > 1$) that usually represent multiple load edges. This scenario does not occur in the conventional Half-Edge (HE) data structure, where the geometric network edges cannot be duplicated between any pair of vertices [57,58]. However, this happens in the underlying graphs like T_{SLS} in Fig. 5c. For instance, the duplication in $E_{i-j}\left\{e_{i-j}^0 \dots e_{i-j}^k\right\}$ generates a "virtual face" in the topology. Fig. 11 shows a novel system of identification called Augmented-Half-Edge (AHE) that deals with this issue. In AHE, a face can be defined with at least 2 half-edges (instead of at least 3 in HE), and the 2 half-edges in one face share the opposite endpoints and enclose a "virtual face" in the underlying graph. An identification index k that counts the duplicate edges in E_{i-j} is introduced in e_{i-j}^k. Besides, e_{i-j}^k corresponds to a unique load l_{i-e}^n identified with n that represents the count of the loads l_{i-e} applied on the same node. In this regard, every edge in the topology has a unique label, even if it shares the same vertex endpoints with another edge (see Fig. 11).

In Fig. 12a, a sub system F^s of the SLS structure F is highlighted and a detailed overview of the planarization process for one trail is demonstrated in Fig. 12b. The loads $l_{1-e}, l_{2-e}, l_{3-e}$ are repeated 3, 2 and 1 times respectively to arrive at the cycle $< v_e >$, which represents the global equilibrium. Correspondingly, as illustrated in T_p^s in Fig. 12c, the force diagram depicts that the load edges applied at one of the inner vertices of the network is made pass through all vertices on the corresponding guiding trail until it reaches the boundary of the graph. For instance, the l_{1-e}^0 start from $< v_1 >$ and passes $< v_2 >, < v_3 >, < v_4 >$ following the in-and-out half-edge $\{e_{1-2}^0, e_{2-3}^0\}, \{e_{2-3}^0, e_{3-4}^0\}, \{e_{3-4}^0, e_{4-e}^0\}$ in the graph until it is connected to $< v_e >$.

The underlying graph T_p is the state of T (U as matrix format) after modifications to make it planar, in which all the loads l_{i-e} have been transferred into e_{i-j}^k or e_{i-E}^k (L_i becomes a null set). Therefore, T_p is a

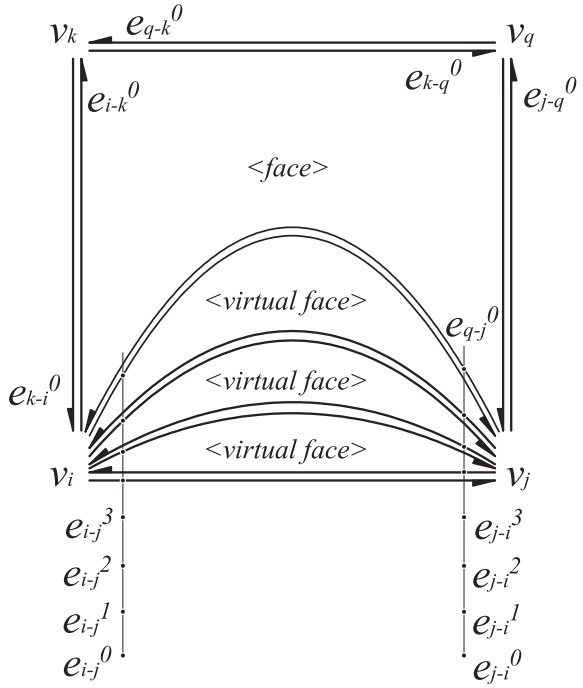


Fig. 11. Illustration of the augmented half-edge (AHE) data structure

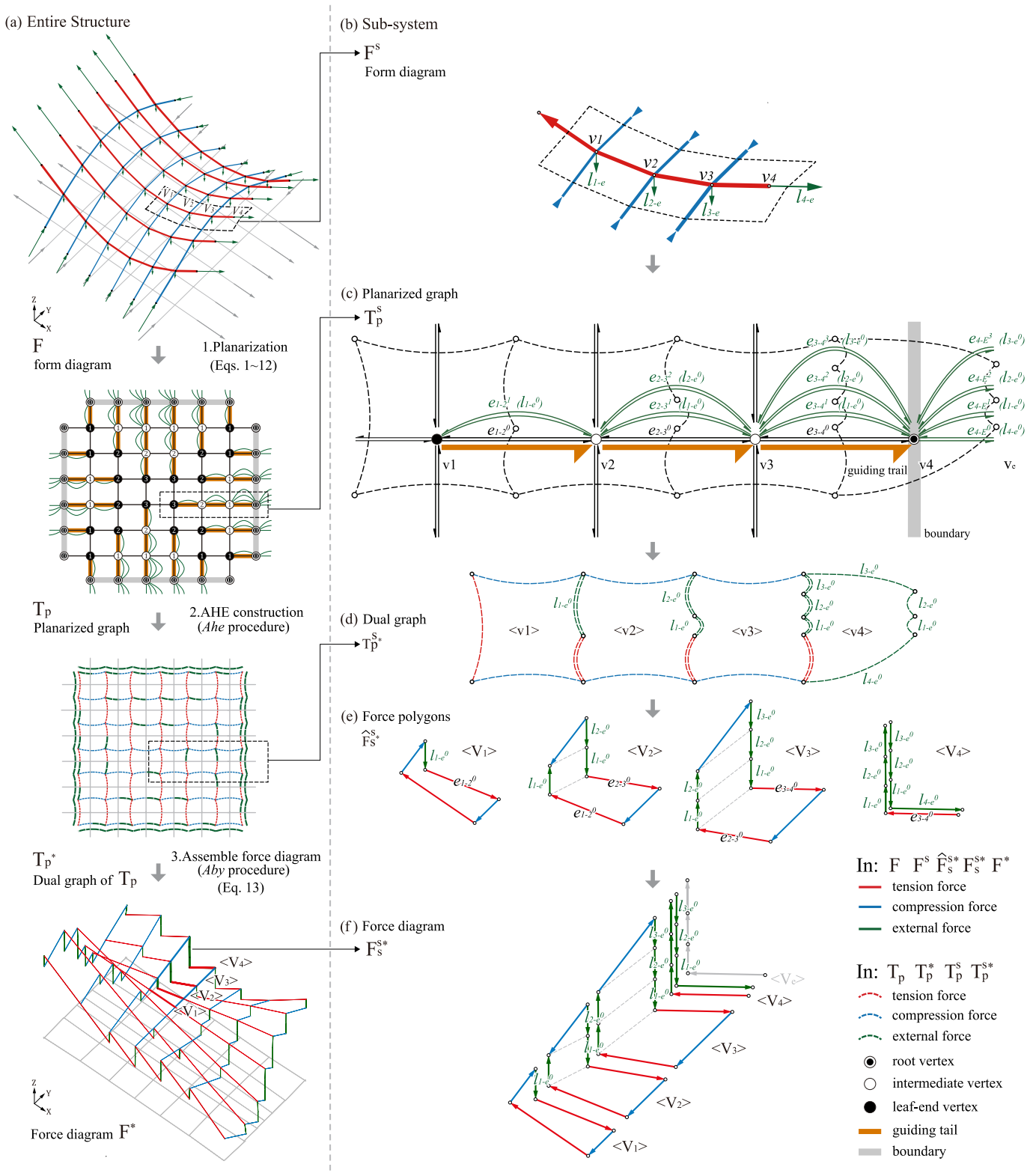


Fig. 12. Overview of the planarization process usign *guiding trails* and the force diagram assembly procudure with AHE data structure. (a) Entire structure system: the form diagram F, its underlying graph T_p with the *guiding trail* in colors, the corresponding dual graph T_p^*, and the generated force diagram F^*. (b) Sub-system: the structure form F^s. (c) Sub-system: the planarized graph T_p^s. (d) Sub-system: The dual graph T_p^{s*}. (e) Sub-system: four force polygons $\hat{F}_s^{s*}$. (f) Sub-system: the integrated force diagram F_s^{s*}.

planar graph with no leaf edge and corresponds to the matrix U'. The latter is obtained after running the iterations formulated in (12,13) with the *guiding trails* T_g (U_p in matrix format). According to the dual principle in the graph view [59], the dual T_p^* of the planar underlying graph T_p in half-edge expression is isomorphic with the force diagram F^*(see Fig. 12d, e). Therefore, the assembling procedure (presented in [25],

labeled with $Aby()$) of the force diagram follows the process that leads to T_p^* in AHE format, labeled with $T_p^* = Ahe(U')$. The adjacency between force polygons in F^* corresponds to the half-edge polygon $< v_i >$ adjacency in T_p^* (see Fig. 12e, f). The force diagram can then be constructed with the $Ahe()$ and $Aby()$ procedures:

$$F^* = Aby(F, Ahe(U'))) \tag{14}$$

4.4. Integrated workflow with CEM

With the CEM framework [8], the equilibrium is calculated sequentially following the topological diagram T^{cem}. The algorithm's structure follows a clear hierarchy and distinguishes trails (T_t^{cem}) and deviation (T_d^{cem}) edges. Apart from specific cases involving gradient-based optimization on "auxiliary trails" [52], the trails in the topological diagram of CEM determines the calculation sequence.

In this regard, a SLS structure can be modeled using CEM so that $T_t^{cem} = T_g$ when it meets the following conditions concurrently:

a) The topology of the structure, without applied external forces, is planar in nature, or a planarization process can be implemented beforehand to achieve planarity without disrupting the predefined trails.
b) All the vertices within the structure that are subject to loads must be contained in a trail.

Fig. 13 illustrates a case where the force diagram is generated straightaway from the CEM model. The guiding trails are extracted directly from the topological diagram, and the support vertices are inherently positioned as the boundary vertices of the guiding trails. The corresponding force diagram can be produced at once using the methodology presented in Sections 4.1 and 4.3. Users can manipulate the structural form within the CEM framework by altering the input parameters in the topological diagram and intuitively observe the consequent transformation in the force diagram.

5. Computational design implementation

In this section, we present several cases of SLS structures designed using the presented approach. The first examples illustrate simple structures (a cable net and ribbed dome), under pure tension or compression. Thereafter, we show structures that belong to categories that typically pose significant challenges using other computational methods based on graphic statics (such as pneumatic structures and self-overlapping spiral prestressed nets).

In all cases, a new method named *Vectorial Difference Method (VDM)* is implemented to achieve the equilibrium state of the structures. *VDM* can eliminate the extra residual forces in the structure's vertices by iteratively imposing the parallelism between the corresponding edges in the form and force diagrams. The parallelism between F and F^* can be evaluated with $\widehat{R}$:

$$\alpha_{i-j} = \frac{\vec{E}_{i-j} \bullet \vec{E}_{i-j}^{\,*}}{\left|\vec{E}_{i-j}\right| \bullet \left|\vec{E}_{i-j}^{\,*}\right|} \tag{15}$$

$$\widehat{R} = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} (\mu - 1)^2 \text{ where } \mu = \begin{cases} \alpha_{i-j} \text{ if } |E_{i-j}| > 0 \\ 1 \text{ if } |E_{i-j}| = 0 \end{cases} \tag{16}$$

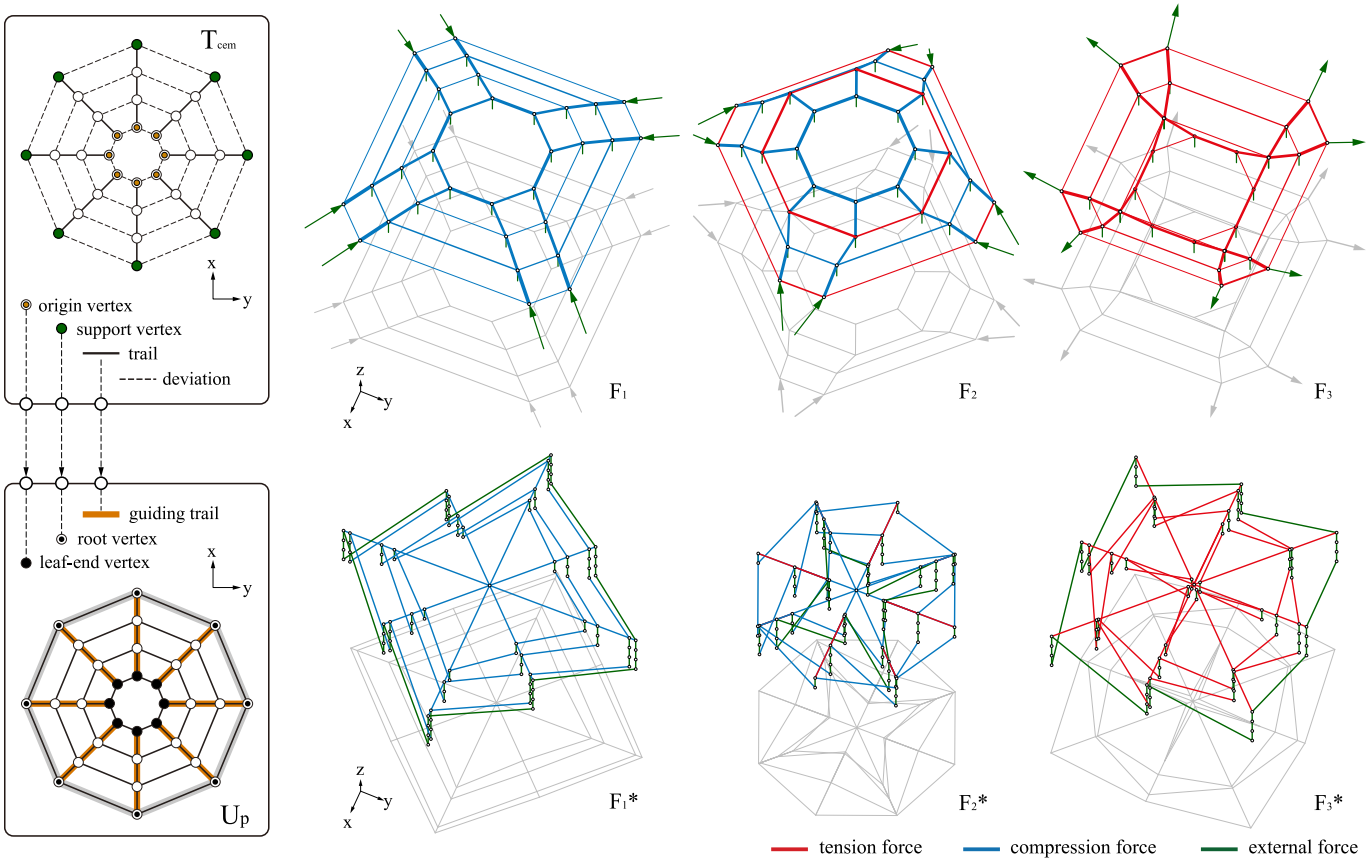


Fig. 13. The correspondence between the topology diagram of the CEM and the graph with the *guiding trails* highlighted with orange color (left). The force diagrams generated directly with the topological data in CEM (right).

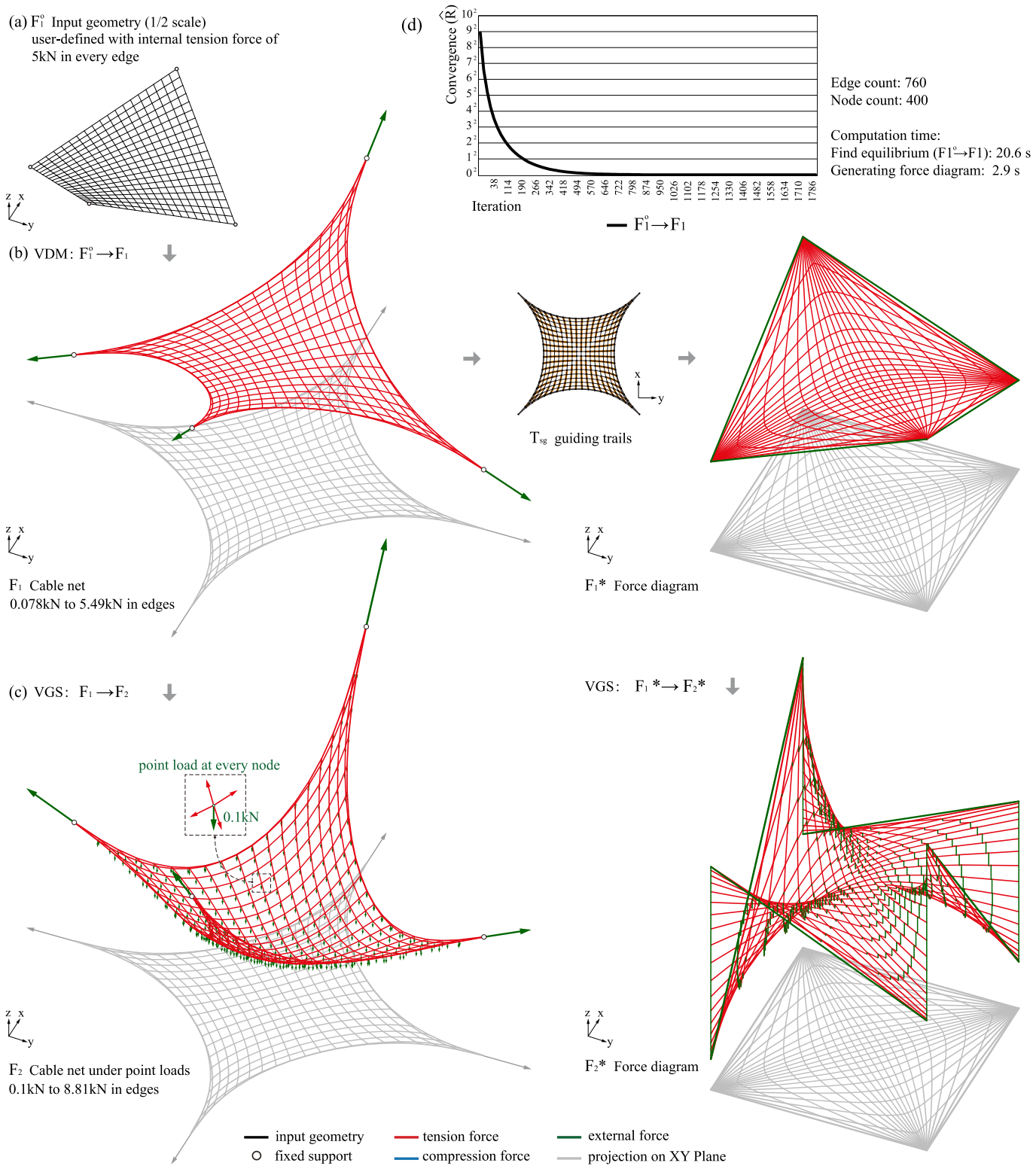


Fig. 14. A cable net structure: (a) The input geometry (F_1^0) user-defined with 5kN in every edge; (b) the form diagram F_1 of the cable net subject to self-stress generated using VDM ($F_1^0 \rightarrow F_1$). The force diagram F_1^* generated according to the guiding trails T_{sg}; (c) Form diagram F_2 and the force diagram F_2^*, after transformations with VGS tool to adapt to the new loadcase (0.1kN loads applied on every node) ($F_1 \rightarrow F_2$ *and* $F_1^* \rightarrow F_2^*$); (d) The graph shows the convergence to equilibrium throughout the form-finding process.

where i and j represents the indices of the vertices in the form diagram. α_{i-j} represents the angle between the edge vector $\vec{E}_{i-j}$ in F and the $\vec{E}_{i-j}^{\,*}$ in F^*. $\widehat{R}$ evaluates the cumulative angles between the corresponding edges in F and F^*. Once the condition $\widehat{R} \rightarrow 0$ is achieved, which means the structure is in equilibrium, the VGS tool can be implemented to perform geometrical and structural transformations while keeping the interdependency between the pair of diagrams.

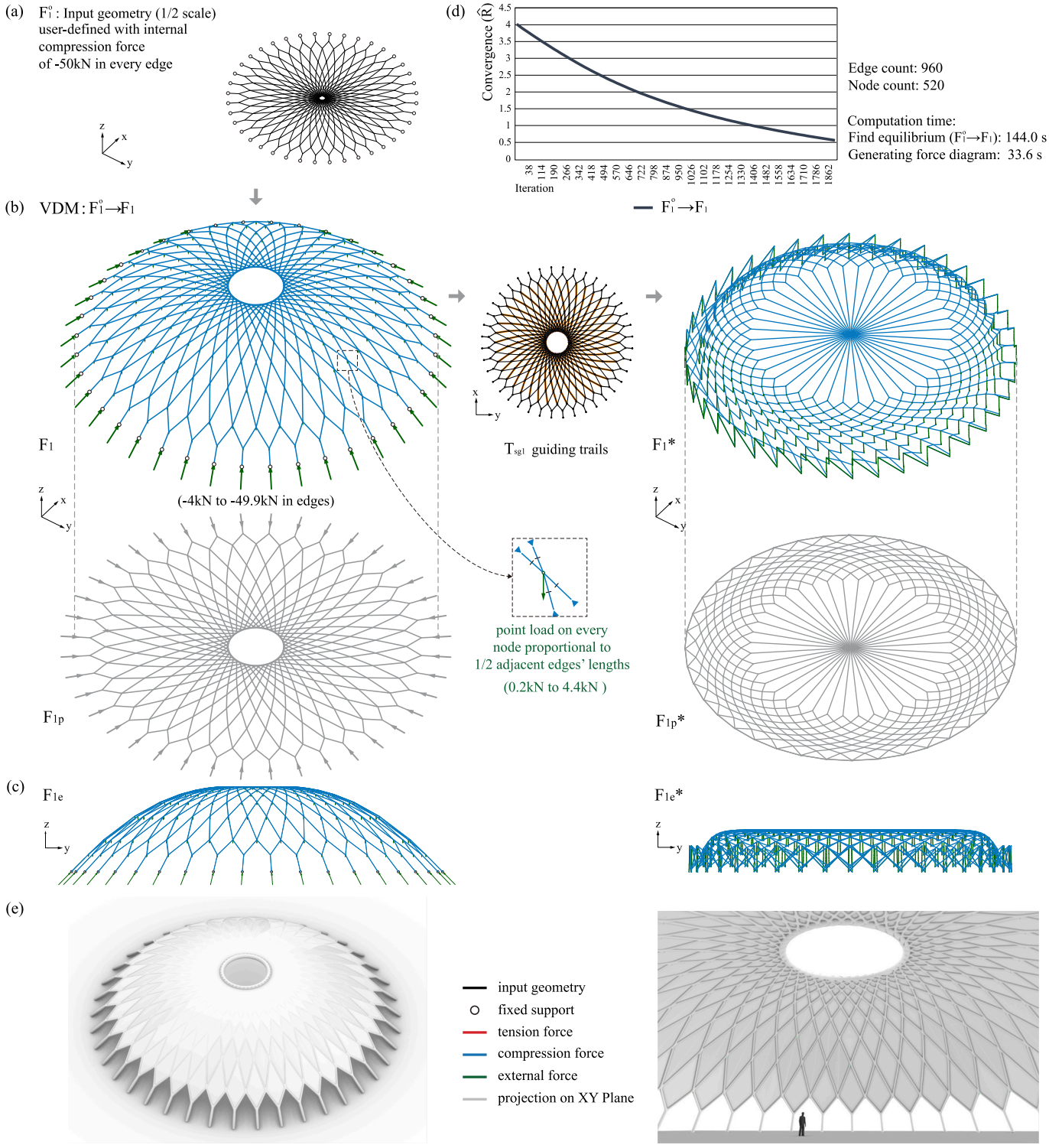


Fig. 15. A gird shell structure: (a)The input geometry (F_1^0), user-defined with -50kN in every edge, generated with the Golden Spiral rule; (b) The form diagram F_1 form found with VDM ($F_1^0 \rightarrow F_1$) while subjected to the loadcase (the load, ranging from 0.2kN to 4.4kN, applied to every node is propotional to the length of adjacent ribs), and the corresponding force diagram F_1^* generatd following the spirally patterned guiding trails T_{sg}. F_{1p} and F_{1p}^* are the the XY-plane view of F_1 and F_1^*; (c) F_{1e} and F_{1e}^* are the YZ-plane view of F_1 and F_1^*; (d) The graph shows the convergence to equilibrium throughout the form-finding process; (e) Renders of the resulting structure: bird view (left), interior view (right).

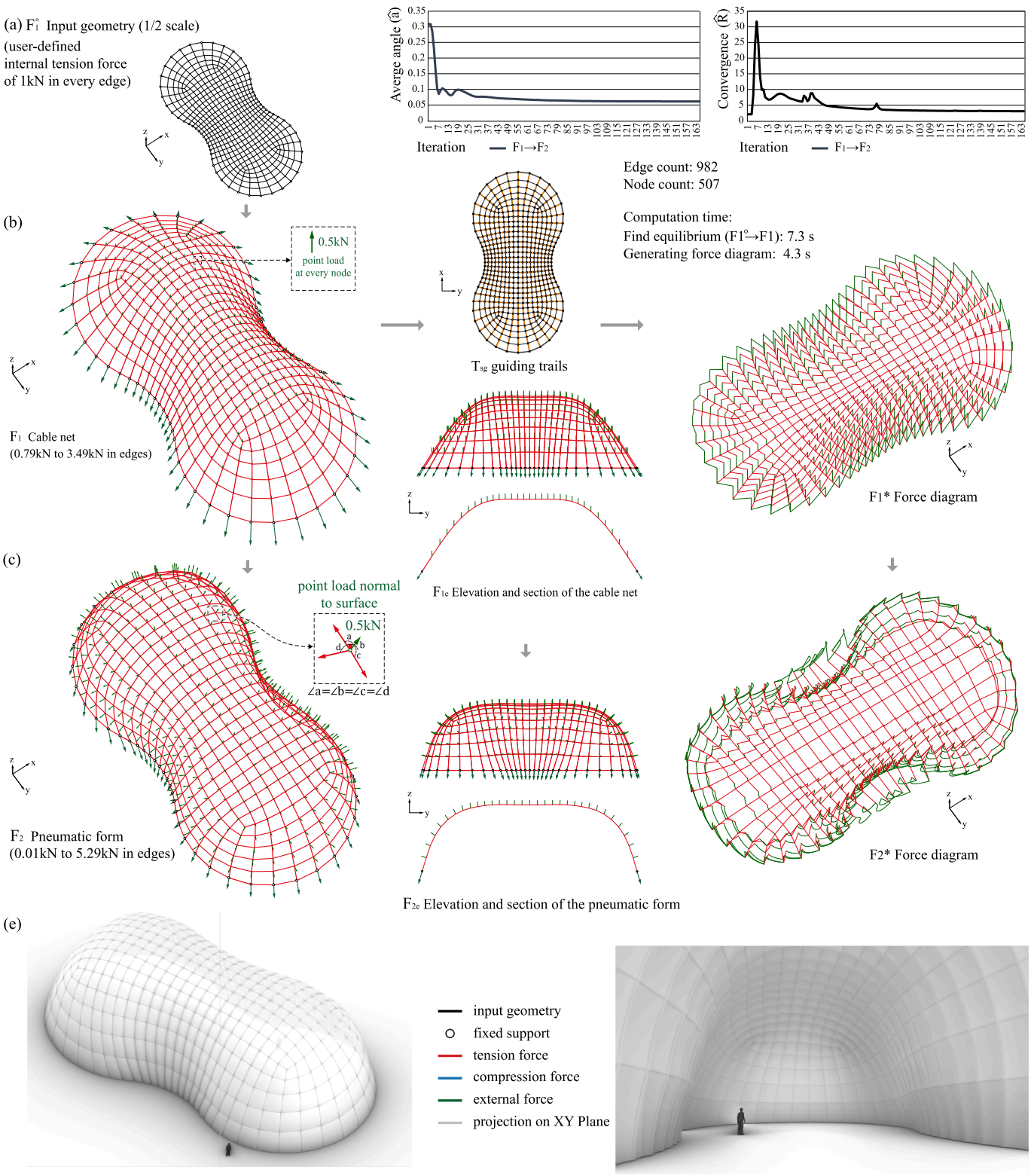


Fig. 16. Form-finding of a pneumatic structure: (a) The input network generated with a SubD function (F_1^0) and is user-defined with 1kN in every edge; (b) The form diagram F_1 form found with VDM ($F_1^0 \rightarrow F_1$) is subjected to unitary 0.5kN vertical upwards load per node and the corresponding force diagram F_1^* generatd following the *guiding trails* T_{sg}. F_{1e} is a YZ-plane view to show the relation between force and surface in direction; (c) F_2, F_{2e} and F_2^* are the form found form diagram, form diagram in YZ-plane and force diagram after applying the pneumatic forces (Imposing the direction of the load vector at an equal angle between all its adjacent edges without altering its magnitude) ($F_1 \rightarrow F_2$); (d) The graph shows the average angle formulated in (18) and the convergence to equilibrium throughout the form-finding process; (e) Renders of the resulting structure: bird view (left), interior view (right).

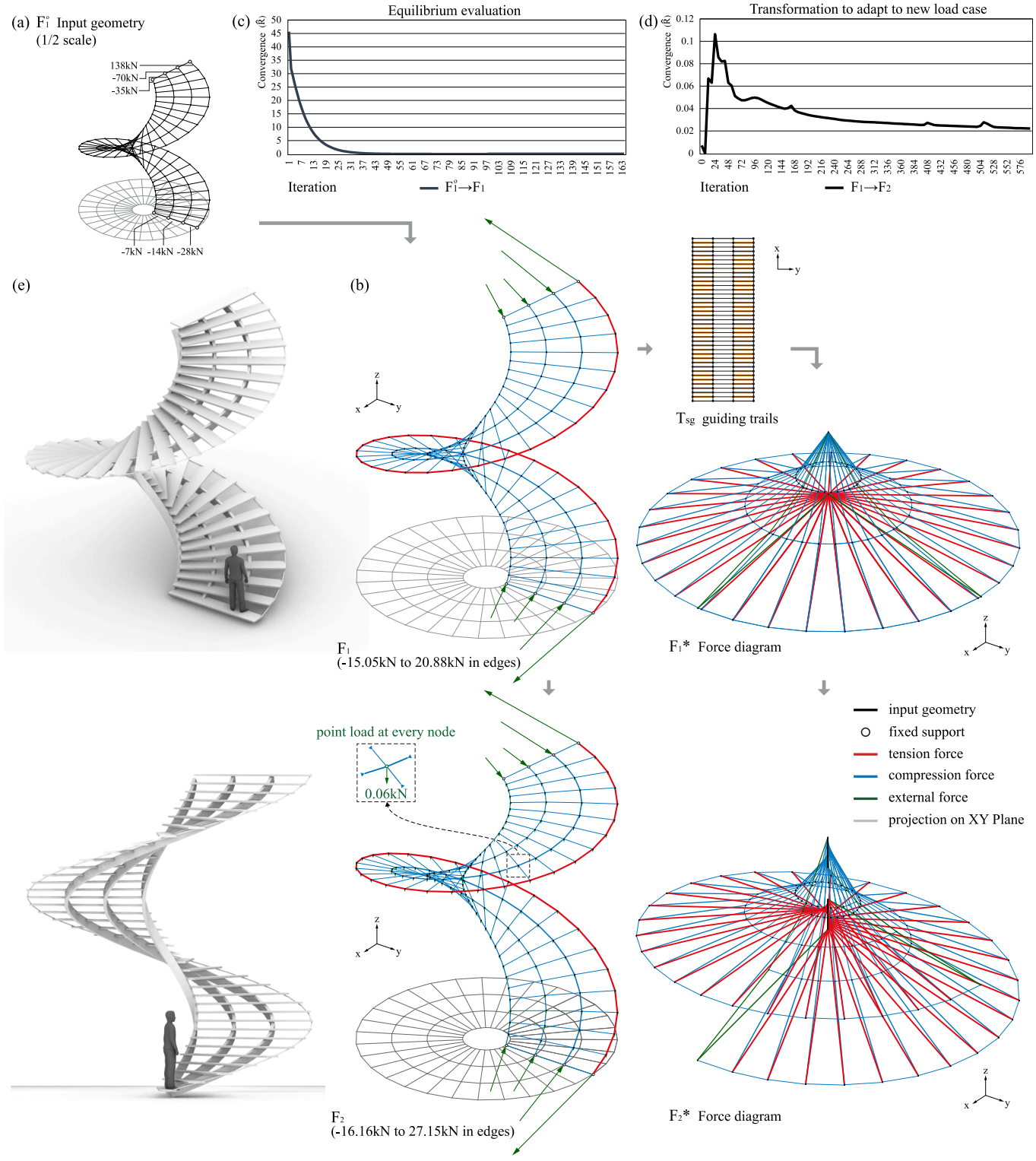


Fig. 17. A spiral staircase: (a) The input geometry (F_1^0) and the equilibrium structure F_1 supported on two ends and the inner force calculated with VDM ($F_1^0 \rightarrow F_1$). The force diagram generated following the *guiding trails* T_{sg} (b) The Form diagram F_2 form found the equilibrium with VGS-transformation ($F_1 \rightarrow F_2$ and $F_1^* \rightarrow F_2^*$) under the load case of 0.06kN unitary vertical downwards load per node; (c) The graph shows the convergence to equilibrium throughout the form-finding process ($F_1^0 \rightarrow F_1$); (d) The graph shows the convergence to equilibrium throughout the form-finding process ($F_1 \rightarrow F_2$); (e) Two renders of the resulting structure.

5.1. A prestressed cable net

In Fig. 14a, one grid net F_1^o composed of 19 × 19 cell is prestressed (with 5kN tension forces in every edge) on four fixed supports at different heights in space. The equilibrium state (with internal forces ranging from 0.078kN to 5.49kN) is found with VDM (Fig. 14b). The underlying graph of the structure is planar, and there are no applied loads, i.e. $\forall L_i \in U = \{\}$. The load case can be changed by adding vertical downwards point loads (0.1kN per node) at the nodes of the net to reach the new equilibrium state F_2(Fig. 14c). In the corresponding force diagram F_2^* all point loads are redirected to a closed force polygon on the boundary of F_2^* that represents the equilibrium of the external forces.

5.2. A ribbed dome

The concrete dome of the Palazzetto dello Sport [60], engineered by Pierluigi Nervi is studied as a reference for the second example. This structure employs a spiral pattern (like F_1^o in Fig. 15a) for the dome's ribs. The form-finding starts from the state F_1^o (with only prescribed internal compression forces of -50kN in every edge) and reaches the externally loaded F_1 (with internal compression forces ranging from -4kN to −49.9kN) after applying the VDM algorithm (Fig. 15d). Vertical downwards point loads are applied at each node of the structure and their magnitudes are proportional to the length of the adjacent members (0.2kN to 4.4kN) (Fig. 15b). Once F_1 is achieved, the network vertex ranking algorithm generates the guiding trails T_{sg} from F_1. The resulting planarized graph allows to construct the force diagram F_1^*. Thanks to T_{sg}, the latter forms a dome-like shape that visualizes the internal force distributions in the dome (Fig. 15c).

5.3. Pneumatic structure

This case study illustrates a novel procedure to form find pneumatic structures through the interactive transformation of form and force diagrams. In pneumatic structures, air pressure – which in the discretized representation of this example is replaced by point loads – is consistently applied perpendicularly to the surface of the structure. Therefore, the applied loads do not follow the same direction. For this reason, using a *2.5D method* (Sections 1.3.1) by projecting the structure onto a plane is not feasible. As shown in Fig. 16, the form-finding process is composed of two stages. First, the geometry of the input net (with 1kN tension force in every edge) is subject to unitary vertical upwards point loads (0.5kN per node), and F_1^o (Fig. 16a) is form found with VDM (with internal forces ranges from 0.79kN to 3.49kN). Then the guiding trails are generated (Fig. 16b). Afterwards, using the transformation component in the VGS tool, directional constraints are imposed on the load edges at each node of the network to ensure that forces are always perpendicular to the network face while remain unchanged. Given that the surface of the pneumatic structure is approximated using a network, perpendicularity is achieved by imposing an equal angle constraint on the load vector and adjacent edges [53] (Fig. 16c). Throughout the form-finding process ($F_1 \rightarrow F_2$), perpendicularity at each node of the network is monitored with a parameter $\hat{a}$:

$$\theta_{i-j} = \arccos\left(\frac{\vec{L}_i * \vec{E}_{i-j}}{\left|\vec{L}_i\right| * \left|\vec{E}_{i-j}\right|}\right) \tag{17}$$

$$\overline{\theta}_i = \frac{1}{m_i} * \sum_{j=0}^{m_i} \theta_{i-j} \tag{18}$$

$$\hat{a} = \sum_{i=0}^{n} \sum_{j=0}^{m_i} |\overline{\theta}_i - \theta_{i-j}| \tag{19}$$

Where i represents the index of the node of the structure and the count of V_i is n. j represents the index of the edge link to L_i, whose count is m_i. θ_{i-j} depicts the angle between the $\vec{L}_i$ and $\vec{E}_{i-j}$. $\overline{\theta}_i$ is the average value of the angles between $\vec{L}_i$ and its every adjacent edge $\vec{E}_{i-j}$. Thus $\hat{a}$ represents the cumulative distance from every θ_{i-j} to $\overline{\theta}_i$, and evaluates the angle equality between L_i and the adjacent network elements.

Fig. 16b, c shows the origin and the form-found structure (with internal forces ranging from 0.01kN to 5.29kN) by constraining the cable net with $\hat{a} \rightarrow 0$ during the transformation. Meanwhile, the table in Fig. 16d demonstrates the convergence of the entire process for both parameters $\hat{a}$ and $\hat{R}$, where $\hat{R}$ is the convergence parameter ($\hat{R} \rightarrow 0$) depicting the interdependency between form and force diagrams in VGS.

5.4. Self-overlapping funicular structure

The masonry spiral staircase (also known as Guastavino staircase [62]) is one of the well-known cases of composite masonry structure. It is a single-layer tile vault that can be regarded as a uniform thin helicoidal shell [63]. To achieve static equilibrium of the spiral plate, a vector-based form-finding process is implemented. The input geometry is generated via parameterized modelling (F_1^o in Fig. 17a). The tension and compression forces in the structure are then evaluated with VDM (convergence process shown in Fig. 17c), to produce F_1 (with internal forces ranging from −15.05kN to 20.88kN). Following the same procedure as in the previous cases, a force diagram F_1^* that is radially symmetric is constructed.

As depicted in Fig. 17b, when the spiral plate is subjected to vertical downwards point loads with 0.06kN per node, noticeable geometrical transformations occur in the middle of the structure (see F_2) and the internal forces' magnitudes changes to the range between −16.16kN and 27.15kN.

6. Conclusions

This article presented a series of techniques to build specifically ordered vector-based force diagrams for SLS structures. Compared to the current VGS method, the newly proposed algorithm represents a more user-controllable and responsive computer-aided design (CAD) tool for structural form-finding of SLS structures. We highlight here the contributions brought by this research:

a) Form-finding SLS structure with the extended VGS tool exhibits significant intuitiveness with real-time responsiveness. The graphical representation of the equilibrium state in the 3D form and force diagrams is easy to read, improving the operability for the user. This creates an intuitive tool for collaboration between engineers and architects that promotes communication and understanding of structural behavior.
b) Regarding design freedom, some of the limitations in 2.5D methods are overcome. In particular, any load configuration other than vertical loads can be applied to evaluate the distribution of forces in SLS structure and freely explore potential novel equilibrium systems.
c) The introduced AHE data structure provides a link between the form and force diagrams and their underlying graphs. This data structure can be expressed with a matrix format and provides a foundation for future developments of high-performance computing algorithms.

Despite its progress, the current work still presents certain limitations and necessitates additional developments:

a) Currently, the VGS tool transformation module primarily relies on the Kangaroo2 library [17]. It struggles to support real-time feedback for large-scale networks. For example, the case in Section 5.2 encompasses 4666 constraint goals being applied to the physical system, making each iteration require approximately 500 ms. Future

developments will consider integrating other simulation approaches to accelerate the form-finding process.

b) For statically indeterminate structures, multiple equilibrium solutions based on various load cases and prestresses are possible. In this scenario, the extra internal or external forces other than those necessary for static determinacy, can be regarded as parameters within the equilibrium problem. This study has not fully explored the potential advantages offered by this type of structures. In future work, the implementation of the presented approach to statically indeterminate structures will allow for efficient exploration of the equilibrium solution space through systematic adjustment of internal forces' magnitudes. Coupling this method with the limit analysis will provide a robust tool for evaluating the structure's collapse load factor using a parametric force diagram [64,65].

Fundings

(The author) Yuchi Shen received support by the Natural Science Foundation of China (NSFC#52208010) and the China Postdoctoral Science Foundation (#2022M720716).

CRediT authorship contribution statement

Yuchi Shen: Conceptualization, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Jean-Philippe Jasienski:** Conceptualization, Methodology, Writing – review & editing, Investigation, Software. **Patrick Ole Ohlbrock:** Conceptualization, Investigation, Methodology, Writing – review & editing. **Pierluigi D'Acunto:** Conceptualization, Investigation, Methodology, Supervision, Writing – review & editing, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] S. Adriaenssens, P. Block, D. Veenendaal, C. Williams (Eds.), Shell Structures for Architecture, Routledge, 2014, https://doi.org/10.4324/9781315849270.
[2] D. Veenendaal, P. Block, An overview and comparison of structural form finding methods for general networks, Int. J. Solids Struct. 49 (2012) 3741–3753, https://doi.org/10.1016/j.ijsolstr.2012.08.008.
[3] R. Pedreschi, The Engineer's Contribution to Contemporary Architecture: Eladio Dieste, Thomas Telford, London, 2000.
[4] P. D'Acunto, J.P. Jasienski, P.O. Ohlbrock, C. Fivet, J. Schwartz, D. Zastavni, Vector-based 3D graphic statics: a framework for the design of spatial structures based on the relation between form and forces, Int. J. Solids Struct. 167 (2019) 58–70, https://doi.org/10.1016/j.ijsolstr.2019.02.008.
[5] P. D'Acunto, J.-P. Jasienski, P.O. Ohlbrock, C. Fivet, Vector-based 3D graphic statics: transformations of force diagrams, in: Proceedings of IASS Annual Symposia, Hamburg, 2017, pp. 25–28.
[6] G. Boller, P. D'Acunto, Structural design via form finding: comparing Frei Otto, Heinz Isler and Sergio Musmeci, in: History of Construction Cultures, CRC Press, London, 2021, pp. 431–438, https://doi.org/10.1201/9781003173434-168.
[7] K.-U. Bletzinger, Form finding and morphogenesis, in: I. Munga, J.F. Abel (Eds.), Fifty Years of Progress for Shell and Spatial Structures, Multi-Science, 2011, pp. 459–482.
[8] P.O. Ohlbrock, Combinatorial Equilibrium Modelling - a computational framework for equilibrium-based structural design,, PhD Thesis,, ETH Zürich, 2020, https://doi.org/10.3929/ethz-b-000478732.
[9] K.-U. Bletzinger, E. Ramm, A general finite element approach to the form finding of tensile structures by the updated reference strategy, Int. J. Space Struct. 14 (1999) 131–145, https://doi.org/10.1260/0266351991494759.
[10] J.H. Argyris, T. Angelopoulos, B. Bichat, A general method for the shape finding of lightweight tension structures, Comput. Methods Appl. Mech. Eng. 3 (1974) 135–149, https://doi.org/10.1016/0045-7825(74)90046-2.
[11] X.Y. Yang, Y.M. Xie, G.P. Steven, O.M. Querin, Bidirectional evolutionary method for stiffness optimization, AIAA J. 37 (1999) 1483–1488, https://doi.org/10.2514/2.626.
[12] X. Meng, Y. Xiong, Y.M. Xie, Y. Sun, Z.-L. Zhao, Shape–thickness–topology coupled optimization of free-form shells, Autom. Constr. 142 (2022) 104476, https://doi.org/10.1016/j.autcon.2022.104476.
[13] M.R. Barnes, Form-finding and analysis of prestressed nets and membranes, Comput. Struct. 30 (1988) 685–695, https://doi.org/10.1016/0045-7949(88)90304-5.
[14] M.R. Barnes, Form finding and analysis of tension structures by dynamic relaxation, Int. J. Space Struct. 14 (1999) 89–104, https://doi.org/10.1260/0266351991494722.
[15] V. Alic, K. Persson, Form finding with dynamic relaxation and isogeometric membrane elements, Comput. Methods Appl. Mech. Eng. 300 (2016) 734–747, https://doi.org/10.1016/j.cma.2015.12.009.
[16] A. Kilian, J. Ochsendorf, Particle-spring systems for structural form finding, Int. Assoc. Shell Spat. Struct. 46 (2005) 77–84.
[17] D. Piker, Kangaroo: form finding with computational physics, Archit. Des. 83 (2013) 136–137, https://doi.org/10.1002/ad.1569.
[18] H.-J. Schek, The force density method for form finding and computation of general networks, Comput. Methods Appl. Mech. Eng. 3 (1974) 115–134, https://doi.org/10.1016/0045-7825(74)90045-0.
[19] K. Linkwitz, H.-J. Schek, Einige Bemerkungen zur Berechnung von vorgespannten Seilnetzkonstruktionen, Ingenieur-Archiv 40 (1971) 145–158, https://doi.org/10.1007/BF00532146.
[20] M. Rippmann, Funicular Shell Design: Geometric approaches to form finding and fabrication of discrete funicular structures, PhD Thesis, ETH Zürich, 2016, https://doi.org/10.3929/ethz-a-010656780.
[21] T. Tachi, Interactive freeform design of tensegrity, in: L. Hesselgren, S. Sharma, J. Wallner, N. Baldassini, P. Bompas, J. Raynaud (Eds.), Advances in Architectural Geometry 2012, Springer Vienna, Vienna, 2013, pp. 259–268, https://doi.org/10.1007/978-3-7091-1251-9_21.
[22] M. Bahr, Form Finding and Analysis of Shells and Slabs Based on Equilibrium Solutions,, PhD thesis,, ETH Zürich, 2017, https://doi.org/10.3929/ethz-b-000182853.
[23] Y. Lu, M. Hablicsek, M. Akbarzadeh, Algebraic 3D Graphic Statics with Edge and Vertex Constraints: A Comprehensive Approach to Extend the Solution Space for Polyhedral Form-Finding, CAD Comput, Aid. Design 166 (2024) 103620, https://doi.org/10.1016/j.cad.2023.103620.
[24] V. Alic, D. Åkesson, Bi-directional algebraic graphic statics, CAD Comput. Aid. Design 93 (2017) 26–37, https://doi.org/10.1016/j.cad.2017.08.003.
[25] J.P. Jasienski, Y. Shen, P.O. Ohlbrock, D. Zastavni, P. D'Acunto, A computational implementation of Vector-based 3D Graphic Statics (VGS) for interactive and real-time structural design, CAD Comput. Aid. Design, 171(2024)103695, doi: 10.1016/j.cad.2024.103695.
[26] C. Culmann, Die Graphische Statik, VonMeyer&Zeller, Zurich, 1866.
[27] J.C. Maxwell, On reciprocal figures and diagrams of forces, the London, Edinburgh, and Dublin, Philos. Mag. J. Sci. 27 (1864) 250–261, https://doi.org/10.1080/14786446408643663.
[28] L. Cremona, Le Figure Reciproche Nella Statica Grafica, TipografiaBernardoni, Milano, 1872.
[29] W.J.M. Rankine, XVII., Principle of the equilibrium of polyhedral frames, the London, Edinburgh, and Dublin, Philos. Mag. J. Sci. 27 (1864) 92, https://doi.org/10.1080/14786446408643629.
[30] W.S. Wolfe, Graphical Analysis: A Handbook on Graphic Statics, McGraw-Hill Book Company, New York, 1921.
[31] W. Zalewski, E. Allen, Shaping Structures, John Wiley & Sons, New York, 1998.
[32] M. Konstantatou, P. D'Acunto, A. McRobie, Polarities in structural analysis and design: n-dimensional graphic statics and structural transformations, Int. J. Solids Struct. 152–153 (2018) 272–293, https://doi.org/10.1016/j.ijsolstr.2018.07.003.
[33] M. Akbarzadeh, T. Van Mele, P. Block, On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams, CAD Comput. Aid. Design 63 (2015) 118–128, https://doi.org/10.1016/j.cad.2015.01.006.
[34] A. Nejur, M. Akbarzadeh, PolyFrame, efficient computation for 3D graphic statics, CAD Comput. Aid. Design 134 (2021), https://doi.org/10.1016/j.cad.2021.103003.
[35] Y.C. Chiang, Maxwell–Rankine stress functions of membrane shells and their relation to that of planar funicular gridshells, Int. J. Solids Struct. 252 (2022), https://doi.org/10.1016/j.ijsolstr.2022.111768.
[36] M. Konstantatou, C. Millar, S.A. Behnejad, G.A.R. Parke, O.A. Samavati, A. Mcrobie, G. Athanasopoulos, G. Torpiano, W. Baker, Simple Rankine Gridshells, Proceedings of the IASS Annual Symposium 2021, Guilford (2021).
[37] E.S. Ferguson, The mind's eye: nonverbal thought in technology, Science 197 (1977) (1979) 827–836, https://doi.org/10.1126/science.197.4306.827.
[38] J. Lee, T. Van Mele, P. Block, Form-Finding Explorations through Geometric Transformations and Modifications of Force Polyhedrons, International Association for Shell and Spatial Structures (IASS) Annual International Symposium, Tokyo. 2016, 26-30.
[39] P. Block, Thrust Network Analysis : exploring three-dimensional equilibrium, PhD Thesis, Massachusetts Institute of Technology, 2009, http://hdl.handle.net/1721.1/49539.
[40] M. Rippmann, L. Lachauer, P. Block, Interactive vault design, Int. J. Space Struct. 27 (2012), https://doi.org/10.1260/0266-3511.27.4.219.
[41] D. López López, P. Roca, A. Liew, T. Méndez Echenagucia, T. Van Mele, P. Block, A three-dimensional approach to the extended limit analysis of reinforced

masonry, Structures 35 (2022) 1062–1077, https://doi.org/10.1016/j.istruc.2021.06.013.
[42] G.B. Airy, IV. On the strains in the interior of beams, Philos. Trans. R. Soc. Lond. 153 (1863) 49–79, https://doi.org/10.1098/rstl.1863.0004.
[43] M. Konstantatou, W. Baker, T. Nugent, A. McRobie, Grid-shell design and analysis via reciprocal discrete Airy stress functions, Int. J. Space Struct. 37 (2022) 150–164, https://doi.org/10.1177/09560599221081004.
[44] M. Miki, T. Mitchell, Interactive exploration of tension-compression mixed shells, ACM Trans. Graph. 41 (2022) 1–14, https://doi.org/10.1145/3550454.3555438.
[45] C. Millar, A. Mazurek, A. Beghini Skidmore Owings, F.A. Mcrobie, S. Behnejad, G. Parke, O. Samavati, T. Mitchell, A. Chhabra, A. Beghini, A. Mcrobie, W.F. Baker, M. Consulting, On funicular gridshells and Airy stress functions, Proceedings of the IASS Annual Symposium 2021, Guilford, 2021.
[46] M. Akbari, M. Akbarzadeh, M. Bolhassani, A. Spitzer, From Polyhedral to Anticlastic Funicular Spatial, Structures, Proceedings of the IASS Annual Symposium 2019 (Barcelona, 2019).
[47] G. Zanni, G.R. Pennock, A unified graphical approach to the static analysis of axially loaded structures, Mech. Mach. Theory 44 (2009) 2187–2203, https://doi.org/10.1016/j.mechmachtheory.2009.07.002.
[48] A. McRobie, Maxwell and Rankine reciprocal diagrams via Minkowski sums for two-dimensional and three-dimensional trusses under load, Int. J. Space Struct. 31 (2016) 203–216, https://doi.org/10.1177/0266351116660800.
[49] J. Lee, T. Van Mele, P. Block, Disjointed force polyhedra, CAD Comput. Aid. Design 99 (2018) 11–28, https://doi.org/10.1016/j.cad.2018.02.004.
[50] A. Micheletti, On generalized reciprocal diagrams for self-stressed frameworks, Int. J. Space Struct. 23 (2008) 153–166, https://doi.org/10.1260/026635108786260974.
[51] R. Pastrana, P.O. Ohlbrock, T. Oberbichler, P. D'Acunto, S. Parascho, Constrained form-finding of tension–compression structures using automatic differentiation, Comput. Aided Des. 155 (2023) 103435, https://doi.org/10.1016/j.cad.2022.103435.
[52] P.O. Ohlbrock, P. D'Acunto, A computer-aided approach to equilibrium design based on graphic statics and combinatorial variations, Comput. Aided Des. 121 (2020) 102802, https://doi.org/10.1016/J.CAD.2019.102802.
[53] Z. Wan, P.O. Ohlbrock, P. D'Acunto, Z. Cao, F. Fan, J. Schwartz, A form-finding approach for the conceptual design of air-supported structures using 3D graphic statics, Comput. Struct. 243 (2021), https://doi.org/10.1016/j.compstruc.2020.106401.
[54] H. Crapo, W. Whiteley, Plane self stresses and projected polyhedral: the basic pattern, Struct Topol. 20 (1993) 55–78.
[55] H. Crapo, Structural rigidity, Struct Topol. 1 (1979) 26–45.
[56] T. Cao, P. D'Acunto, J.J. Castellón, A. Tellini, J. Schwartz, H. Zhang, Design of prestressed gridshells as smooth poly-hypar surface structures, Structures 30 (2021) 973–984, https://doi.org/10.1016/j.istruc.2021.01.047.
[57] L. Kettner, Using generic programming for designing a data structure for polyhedral surfaces, Comput. Geom. 13 (1999) 65–90, https://doi.org/10.1016/S0925-7721(99)00007-3.
[58] K.J. Weiler, Topological Structures for Geometric Modeling, Rensselaer Polytechnic Institute, 1986.
[59] T. Van Mele, P. Block, Algebraic graph statics, Comput. Aided Des. 53 (2014) 104–116, https://doi.org/10.1016/j.cad.2014.04.004.
[60] T. Iori, Pier Luigi Nervi, Annibale Vitellozzi: Palazzetto Dello Sport a Roma: un prototipo ripetibile e a buon mercato, Casabella 782 (2009) 50–65. http://hdl.handle.net/2108/22080.
[62] M. Angelillo, Static analysis of a Guastavino helical stair as a layered masonry shell, Compos. Struct. 119 (2015) 298–304, https://doi.org/10.1016/j.compstruct.2014.09.007.
[63] J. Heyman, The stone skeleton, Int. J. Solids Struct. 2 (1966) 249–279, https://doi.org/10.1016/0020-7683(66)90018-7.
[64] J.-F. Rondeaux, P. D'Acunto, J. Schwartz, D. Zastavni, Limit state analysis of 2D statically indeterminate networks using graphic statics, in: Proceedings of the IASS Annual Symposium 2017, Hamburg, 2017.
[65] J.-F. Rondeaux, P. D'Acunto, A. Deschuyteneer, D. Zastavni, Limit state analysis of 2D externally statically indeterminate networks using graphic statics, in: Proceedings of the IASS Annual Symposium 2018, Boston, 2018.