ACADIA 2014 · Design Agency
Los Angeles 23-25 October, 2014 · isbn 9781926724478 · pp. 347-356

Interactive Shaping of Forces

Corentin Fivet & Denis Zastavni

abstract · This paper discusses the interactive design exploration of static equilibria using constraint-based graphic statics. Constraint-based graphic statics is a computerized framework improving the construction, the control and the capabilities of classical graphic statics substantially. It allows the designer to shape any flow of forces interactively by means of successive purely graphical operations applied either on a diagram representing its form or on another diagram representing its vectorial equilibrium.

The first chapter contextualizes the essential role of this tool by expressing a critique about contemporary structural CAD tools and the narrow design chronology they impose. The paper then briefly browses the various key concepts and operations of constraint-based graphic statics. The last chapter highlights the various capabilities and benefits of this tool through a case study. As a conclusion, the paper illustrates how this fully graphical approach breaks new ground in terms of design chronology (*e.g.* the parametric hierarchy can be freely reorganized), visual feedback and dynamic control over mechanical considerations.

search keywords · Structural Design Exploration · Computer-Aided Graphical Design Methods · Graphic Statics · Parametric Design · Design Decision Making ·

Introduction

Shaping the Structural Form · Historical testimonies of structurally pertinent and efficient architectural works (Figure 1, Figure 2 & Figure 3) share similar design approaches: either mechanical considerations are taken into account from the beginning of the design process or the nature of the process allows the initial sketch to be altered in substance in order to fulfill mechanical requirements. That means that the form and the forces are handled together, they are not studied separately. Indeed, if a designer can define a shape and its inner stresses jointly, he is able, for instance, to minimize the amount of material needed, make the building more robust, make it easier to construct or make it meet the unforeseeable spatial and mechanical requirements determined by his own sensitivity.

Freeing the Design Chronology · Today, the designer lacks tools when it comes to steer the initial flows of forces. Contemporary tools dedicated to structures are mostly of the two following kinds: analysis tools and form-finding tools. While the former compute the inner stresses of a given geometry in order to size or assess its structural parts, the latter produce the optimal geometry according to a given set of contextual constraints (*e.g.* mechanical properties and boundary conditions). As a result, none of these tools is properly suited to assist the design from its inception, i.e. when neither the geometry nor the contextual constraints are known.

Contemporary structural design explorations consequently follow this sequence: (0) guess a geometry and other initial parameters; (1) perform the structural analysis (respectively, the automated form-finding or optimization); (2) interpret the

Figure 1 · Front view of the Töss Bridge, near Zürich (Switzerland), designed by Robert Maillart in 1934.

result, *i.e.* the resulting stresses (*resp.* the resulting geometry); (3) guess a change/refinement of geometry (*resp.* of contextual constraints) in order to enhance the next result; (4) reiterate from step 1 until the design becomes satisfactory. For the last couple of years, most research efforts are placed on the enhancement of step 1 only. Although they open new horizons of design they do not help the designer when it comes to steps 2 and 3, which does not encourage a direct return to initial formal choices and hence does not favor permanent control of the structural behavior being shaped.

Very few research efforts are being undertaken in order to make the entire sequence more flexible and interactive, or in other words, more design-oriented. For instance, VonBuelow·2012 and Mueller·2013 automate step 1 and assist step 3 by suggesting possible, relevant alterations. Also, Rippmann·2013 allows patterns of forces to be directly parameterized in steps 0 and 3, allowing better informed decisions during step 3.

The approach followed here, called constraint-based graphic statics, consists in altering step 1 so that the chronology of design is less restricted and both the geometry and the contextual constraints may emerge along the way. This conceptual change is achieved in three ways. Firstly, the geometry, the forces and the boundary conditions are homogenized, *i.e.* they are controlled by a single type of parameter: the position of a point on a plane. Secondly, the distinction between initial parameters and results vanishes: the range of positions that a point can hold is both a synthesis of applied boundary conditions and of available solutions. Moreover, the parametric hierarchy defining the order of precedence between points can be reorganized whenever desired. Thirdly, points are geometrically constrained beforehand so that the form and the forces always match, i.e. equilibrium always exists.

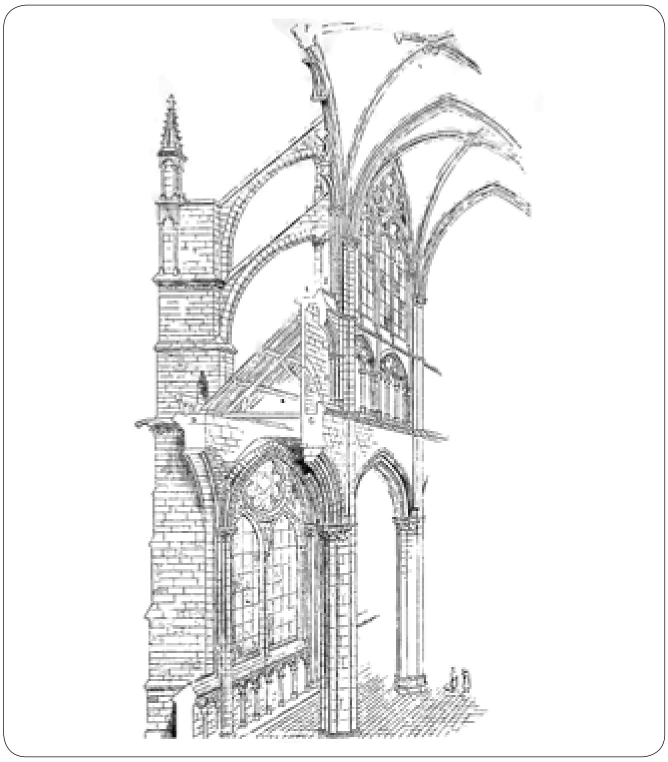

Figure 2 · Section of Amiens Cathedral (ViolletLeDuc·1883, page 203)

Abstracting the Model · As constraint-based graphic statics is designed to assist the architect or the engineer during early structural explorations, the structural problem involved is reduced to its bare essentials: the shaping of static equilibria.

The first advantage of this simplification is that it allows points in planes to be the only type of data controlling the forces, the struts and the ties in equilibrium. As seen in the previous sub-section, this makes the work more controllable, more intuitive and faster, fostering the designer's creative and sensitive expression.

Figure 3 · College Bridge, designed by Ney&Partners and SumProject (2002-2009).

This simplification also implies that the presented tool cannot be used alone and that a comprehensive structural analysis may still need to be performed afterwards with other tools, *e.g.* to check deflections and dynamic responses. However, it is expected that this subsequent check may often be redundant and does not require a substantial change of the initial sketch, as it is often the case when the structural behavior is mastered from the beginning (Fivet 2012).

Finally this simplification allows the model to be the abstraction of a variety of building systems: reticular structures (be they determinate or not, pre-stressed, self-stressed or mechanisms); thrust lines with masonry structures; load paths or discontinuous stress-fields inside reinforced concrete elements or other materials showing a plastic behavior; bending moments and deflections of regular and free-form beams; equilibrium of stabilizing slopes, retaining walls and foundations. Because they use particular theories (*e.g.* Euler-Bernoulli beam theory or the lower-bound theorem of plasticity), specific assumptions must apply for these various applications to be valid. These conditions may either be observed by the designer himself or imposed by specific geometric constraints.

Moreover, the designer can start the design exploration without having to define the effective building system beforehand and he can switch from one to another in the middle of his exploration.

Conceptual Framework

Constraint-based graphic statics is a computerized framework that improves the construction, the handling and the capabilities of the reciprocal diagrams of classical graphic statics by defining two new theoretical devices: (1) the user modifies these diagrams by means of successive operations whose geometric properties do not at any time jeopardize the static equilibrium of the strut-and-tie network; and (2) nodes, considered as the only variables, are constrained within graphical regions that represent the entire domain of solutions.

The three following sections (1) recall the meaning of the two diagrams of graphic statics, (2) explain how the particular transformations of static equilibria are ruled and (3) sketch how the graphical domains of solutions are created. More technical details about their implementation can be found in (Fivet 2013) and (Fivet 2014).

The framework presented in this paper is two-dimensional. Its extension to the third dimension is a matter of future research.

Graphic Statics · Graphic Statics consists of a set of methods acting on two distinct diagrams: the geometry of the strut-and-tie network is drawn in a so-called form diagram in which a distance between two points measures a length, while the corresponding inner forces are drawn in a force diagram in which a distance between two points measures a magnitude of force. Specific drawing rules make these two diagrams reciprocal.

Figure 4 synthesizes the rules binding them. Drawing (a) is a form diagram and drawing (d) is a force diagram. Drawing (b) is the vectorial representation of the forces acting on each node — *i.e.* each force in (b) is parallel to and of equal magnitude of a force or a rod in (a). Since each node is in equilibrium, each vectorial addition must form a close cycle. Moreover, if this is true for every node in (a), both the translational and the rotational equilibrium of the entire structure in (a) is guaranteed.

Because the mechanical action of a rod — either a strut, in compression, or a tie, in traction — is equivalent to the action of two equal, parallel, but opposite forces, pairs of forces can be superimposed as shown in drawing (c). This superimposition is always possible if the order in which concurrent forces are taken in (a) is always either clockwise or counter-clockwise.

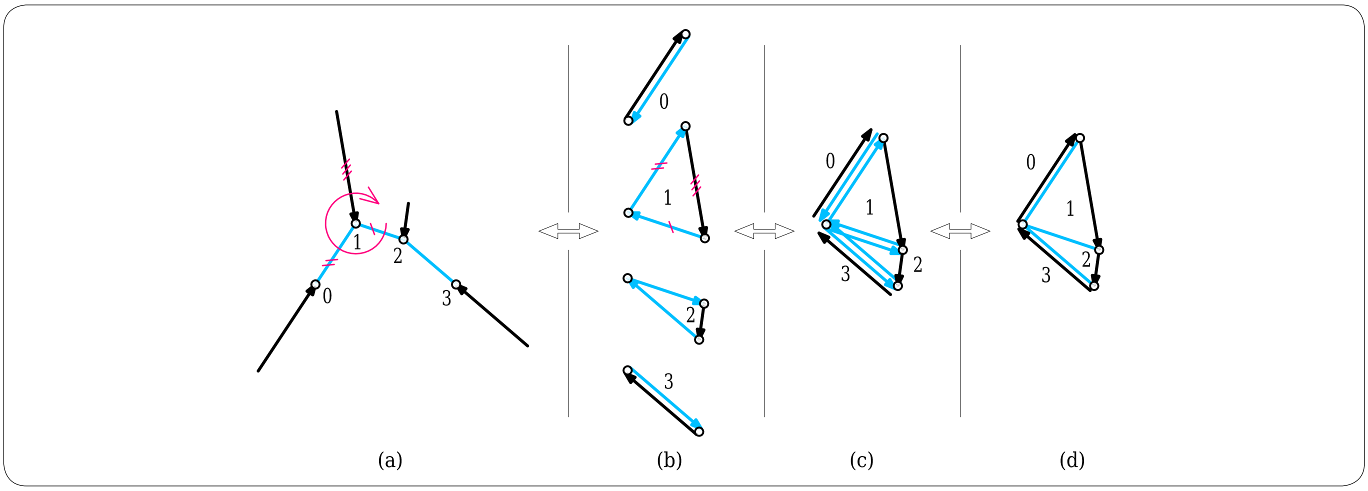

Figure 4 · Intermediate relationships between the form diagram (a) and the force diagram (d).

The transition between (c) and (d) is only a graphical simplification: pairs of forces representing the action of a same rod are replaced by a segment of line. As a result, the force diagram in (d) provides the magnitude inside each bar and informs that the structure in (a) is in static equilibrium.

Methods of graphic statics enable the structural designer to shape an initial structure while controlling at the same time both the geometry of the structure and its inner forces. Because the simultaneous understanding of both diagrams reveals the commonly hidden properties that govern the inner distribution of forces, graphic statics greatly favors the design of efficient and expressive structures (Fivet·2012).

The roots of graphic statics go back to the 1858's works of J.W.M. Rankine (Rankine·1858, pages 139-144) and J.C. Maxwell (Maxwell·1864) and, thereafter, to those of S. Earnshaw, F. Jenkin, C. Culmann, L. Cremona, R.H. Bow and M. Lévy. By the end of the nineteenth century, graphic statics became one of the most popular tool for structural design. In the late twentieth century, the development of computers capable of complex numerical calculus (rather than drawing calculus) ended the use of graphic statics in most structural design practices and in engineering schools amid numerical analysis tools. For a couple of years now there is a comeback of graphic statics both in education (Zalewski·1997) and in practice (Conzett·2003, pages 100-109) (Beghini·2013).

The recent enhancement of graphical user interfaces have led to the development of new computerized frameworks of graphic statics (Greenwold·2009) (VanMele·2011). However, these implementations currently lack the ability to explore new typologies interactively. The next sub-sections explain how constraint-based graphic statics overcome this specific limitation and the other limitations of classical graphic statics.

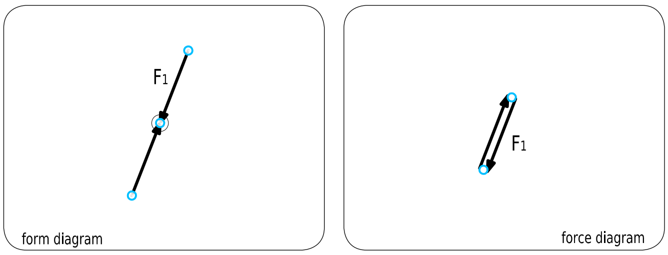

Figure 5 · Introduction of two forces in equilibrium

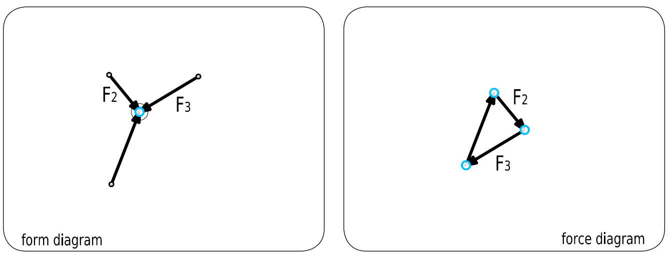

Figure 6 · The force F_1 are resolved into the force components F_2 and F_3.

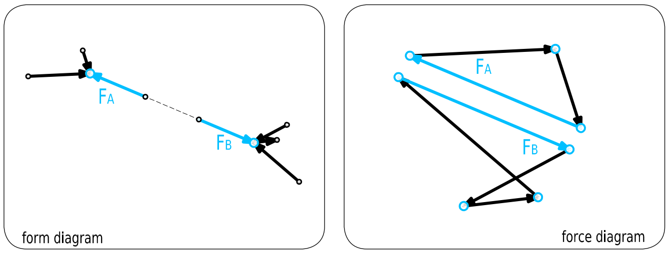

Figure 7 · Forces F_A and F_B are opposite, parallel and of equal magnitude. They are pushing on their point of application.

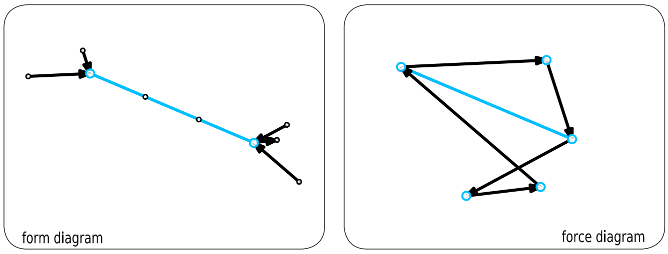

Figure 8 · Forces F_A and F_B are combined into a new compression rods.

Transformations of Static Equilibria · The main idea behind constraint-based graphic statics is to let the user transform existing strut-and-tie networks by means of successive operations that never jeopardize their static equilibrium. The very firstly drawn structure is already in equilibrium — meaning that the form diagram and the force diagram are already completed — and it remains in static equilibrium after each transformation. For instance, it is impossible to add a new force without adding an equal but opposite reaction (Figure 5). The insertion of two equilibrated forces is the first fundamental operation that can transform a network.

The next operation resolves a force into two components (Figure 6). These two components are adjacent in both the form and the force diagrams, so that the point on which these forces are applied remains in static equilibrium.

The addition of two adjacent forces can be done by nullifying one of them, *i.e.* by superimposing two points defining one of the two forces in the force diagram so that its length becomes zero.

The two remaining operations are those (1) converting two forces that are opposite, parallel and of equal magnitude (Figure 7) into the corresponding tension or compression rod (Figure 8) and (2) canceling this conversion.

Thanks to these five primitive operations, the construction of any strut-and-tie network is possible. Moreover, the user is ensured the static equilibrium of the network.

The user has to apply successive operations to achieve his goal. He can also apply predefined sequences of primitive operations available from a built-in library. These sequences allow more intuitive operations, such as the computation of *n* given forces into one resultant, the direct construction of *n* forces in equilibrium or the direct combination of two sub-networks in equilibrium. They also allow more complex operations such as the direct construction of a catenary or a truss network.

Since the only parameters of these networks are points in two diagrams, it is enough to constrain the position of these points inside a graphical region in order to constrain all the geometrical and mechanical properties of the strut-and-tie network. The following sub-section explains how this can be done.

Graphical Solution Domains · The intersection of all the regions applied on a point represents its domain of solution. Each graphical region defined inside the tool is parameterized by positions of points. If a point p* is constrained inside a given region and if the points defining that region are dragged such that p* happens to be located outside the region, p* is then automatically moved by the computer onto the position that is inside the region and closest to p* (Figure 9 & Figure 10). If, for instance, p* is constrained on the intersection of two lines, its domain is equal to a single position and its position is updated as soon as one orientation of these lines changes.

Each graphical region is actually either a primitive region or any logical combination of primitive regions (*i.e.* intersections, unions and inversions). There are three types of primitive regions. The first one is a half-plane (together with its boundary) defined by three points: one to set its position and two others to provide the direction on the left of which the half-plane extends (Figure 11). The second primitive region is the area (together with its boundary) inside the circle defined by three points: one is its center and the other two define its radius (Figure 12). The third primitive region is the area (together with its boundary) outside the circle defined in the same way (Figure 13).

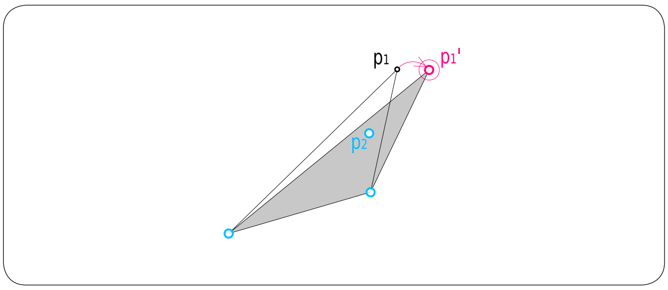

Figure 9 · Point p_2 is constrained inside a shape defined by p_1 and two other points. If p_1 is dragged onto p_1', the position of p_2 does not have to be updated because p_2 remains inside its domain.

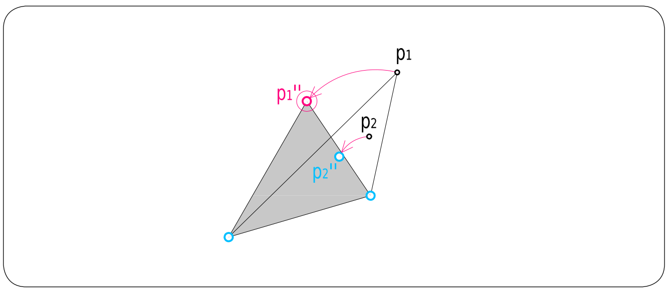

Figure 10 · If p_1 is dragged onto p_1'', the position of p_2 has to be updated onto the closest position p_2'' in order to remain inside its domain.

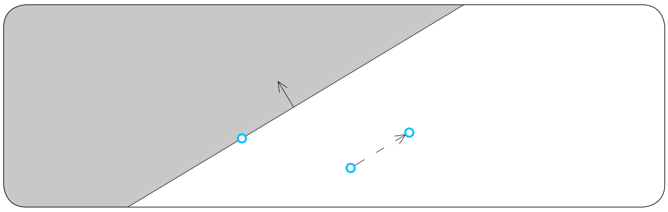

Figure 11 · First primitive constraint: the half-plane.

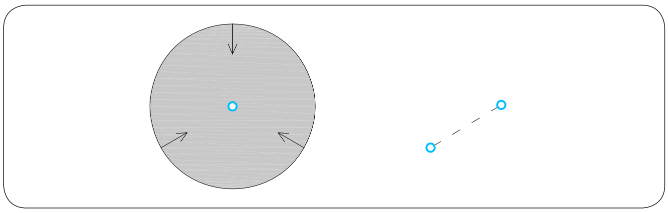

Figure 12 · Second primitive constraint: the inside of a disc.

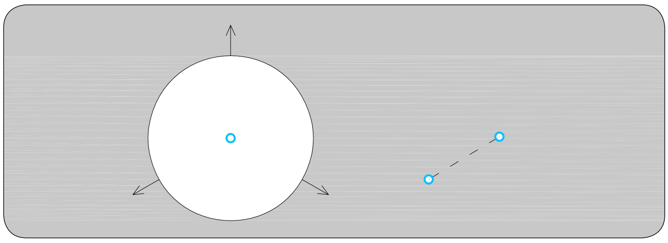

Figure 13 · Third primitive constraint: the outside of a disc.

As further explained in Fivet·2013 and Fivet·2014, these three primitive regions present interesting symmetry properties that can be used as a new fundamental operation to switch the hierarchy of dependencies between points. That means that an automatic procedure exists to rebuild symmetric constraints, *i.e.* to redefine how they are applied so that a point p_1 that was a parameter of point p_2 becomes constrained by p_2 itself, without affecting the geometric property that links p_1 and p_2.

Besides primitive regions, the user can also compound regions from a built-in library. These constraints have a very large range of applications. For instance, they can be used to produce other regions (*e.g.* a line is obtained by intersecting two parallel but opposite half-planes and a circle is obtained by intersecting the inside of a disc with its outside), to constrain a point onto a particular curve (Figure 14), to guarantee that a rod remains in compression or in traction (Figure 15), or to obtain the center of gravity of a given shape.

Constraints may also be added automatically by the computer in order, for example, to ensure that the cycle by which the forces are read around a point remains identical in both diagrams (Figure 16), to ensure that two forces representing a rod remain parallel, opposite and of equal magnitude.

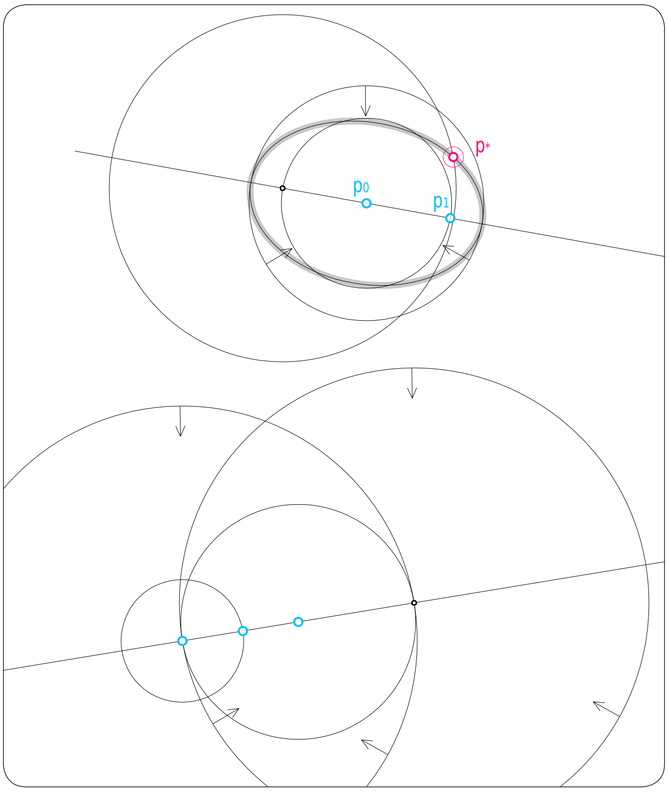

Figure 14 · Hidden primitive constraints that are applied in order to make p* stay on an ellipse centered in p_0 and having p_1 as focal point.

In some cases, the computer is also able to create and apply constraints that ensure that the domain of solutions of a point p* is never empty. These particular constraints are applied on each parameter of the constraints that are applied on p* so that these parameters cannot be moved on a position that would make the domain of p* empty (Fivet·2014). Therefore, the user knows that the domain of each point represents the unique and entire set of available solutions to the given geometric problem.

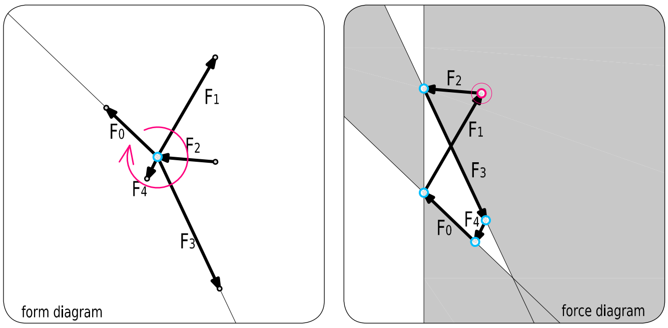

Figure 16 · If the highlighted point in the force diagram remains inside the gray area, the forces F_0, F_1, F_2 and F_3 will be read clockwise according to that order. Otherwise, pairs of forces must be swapped using a specific fundamental operation.

Case Study

This chapter is meant to illustrate a few examples of how constraint-based graphic statics can be used and what advantages it brings to computer-aided structural design explorations. The case study deals with the support structure of a peaked roof. This example can be related to Lachauer·2012 although it has three distinctive characteristics: the typology is here reduced to the planar case; the shape of the support structure is here constrained by more strict boundary conditions; and the form-finding process is here interactive rather than systematized — *i.e.* each step is controlled and desired by the user.

It is assumed that the governing load case is the dead load of the roof and that the design goal is to shape the support structure for minimum bending moments under dead load, hence for minimum sections.

Initial Requirements · Nothing is known at the beginning of the process except the approximate weight of the roof and some assumptions concerning its height and two rectangular spaces (frames A and B) reserved for the expected activity that will take place under the roof. These forces and constraints are built on Figure 17.

The structure is already in equilibrium but the reaction forces are not applied directly on the ground. The designer's goal is to apply operations onto this equilibrium in order to direct the flow of forces towards the ground. The next sub-sections detail one possible process out of the many possible ones.

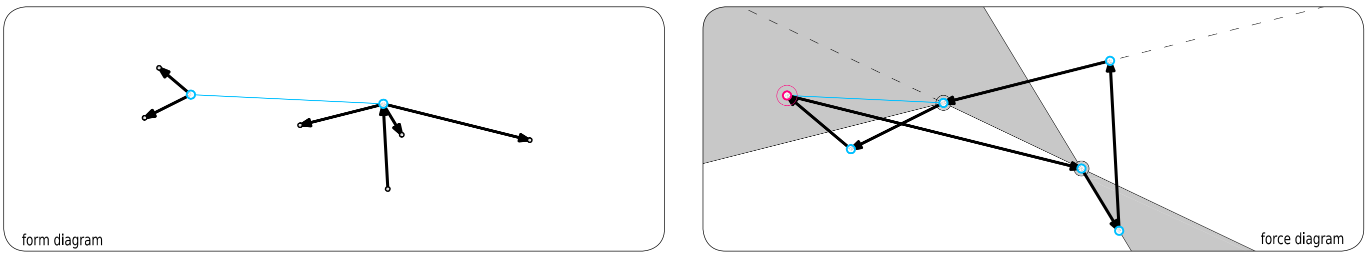

Figure 15 · The highlighted point in the force diagram must remain inside the gray region for the rod to be in traction.

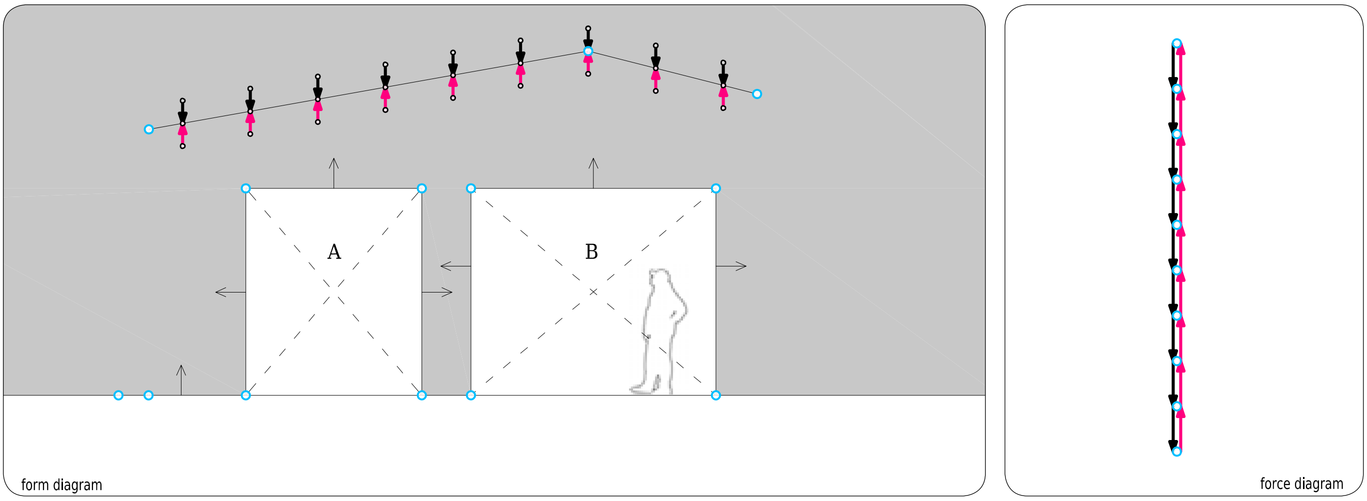

Figure 17 · Introduction of the initial loads and of the boundary conditions (defining the gray area inside which every point has to stay).

Altering the Typology · In order to give shape to that structure the designer can resolve forces (Figure 18) and rotate them by dragging points (Figure 19). He can add points in equilibrium. He can then drag the nodes that define the newly created forces in order to change their position, angle and magnitude (Figure 20) until pairs of forces present the required geometric properties so that they can be transformed into compression or tension rods (Figure 21).

Likewise, he can temporarily work on a partial sub-network, drag its nodes to produce opposite forces (Figure 22) and then merge it with another strut-and-tie network (Figure 23).

If desired, he can also break an existing rod in order to divide the flow of forces in two. Since the model is in static equilibrium at each step of the process, it is easier to interpret it and to decide how to reshape it. A completed model is shown in Figure 24.

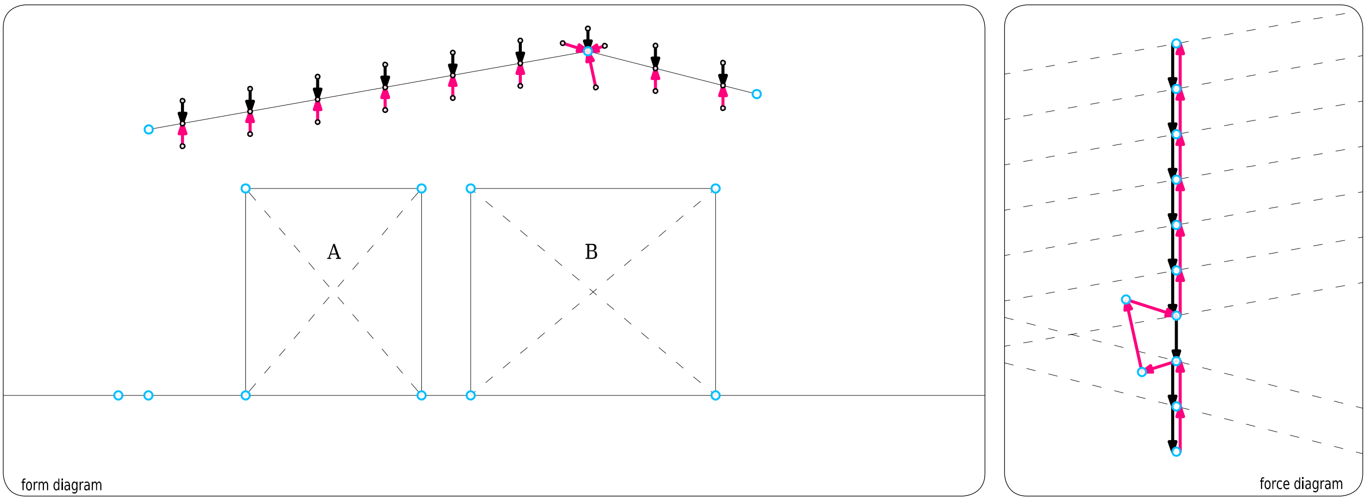

Figure 18 · Decomposition of a force.

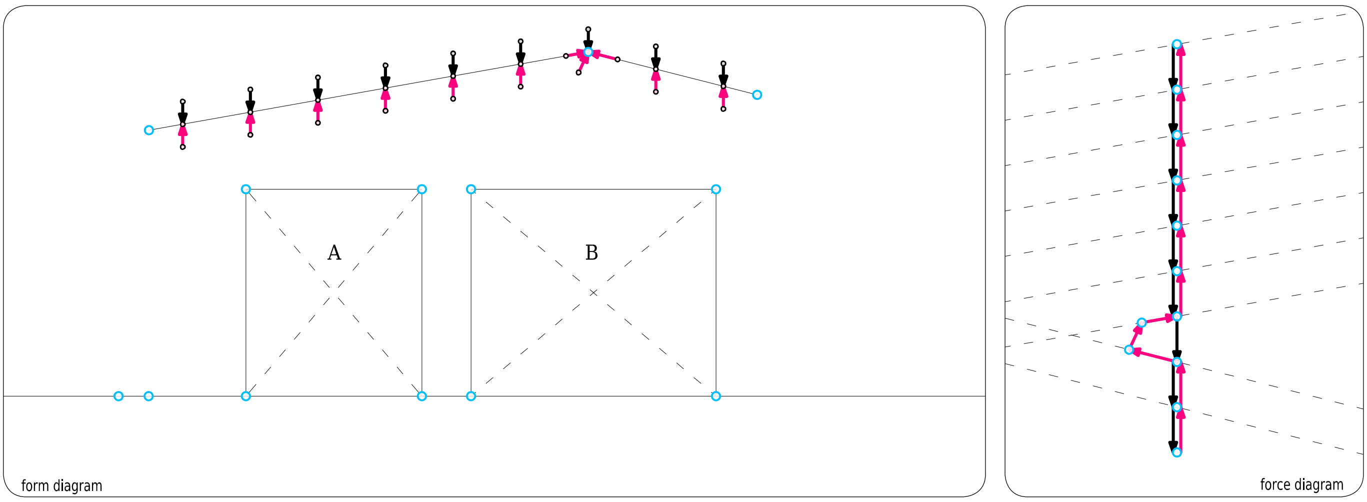

Figure 19 · Alignement of forces with the slope of the roof.

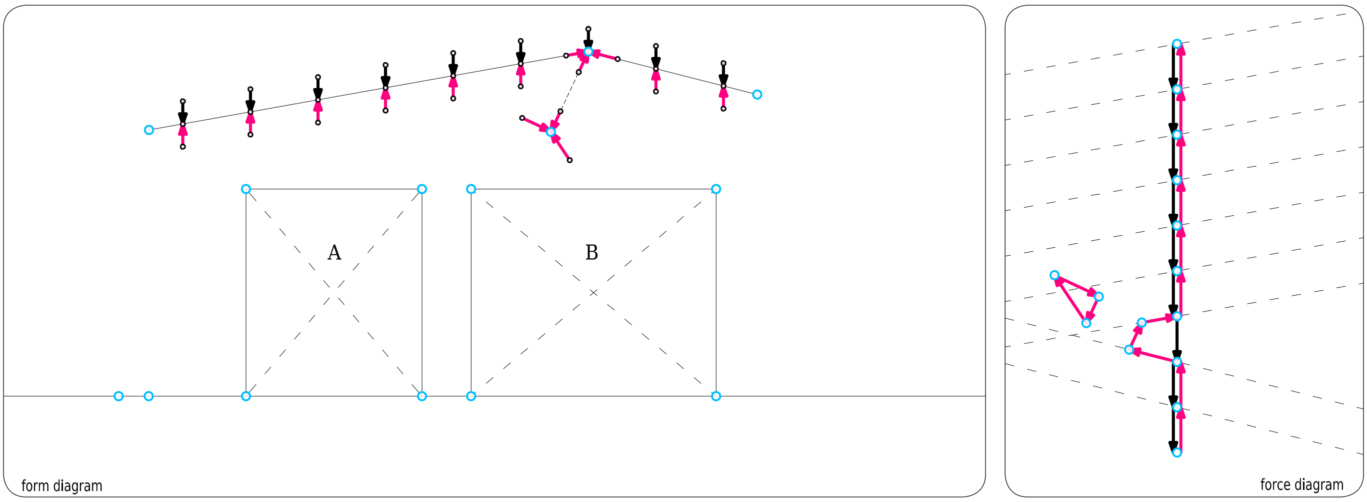

Figure 20 · New set of forces in equilibrium.

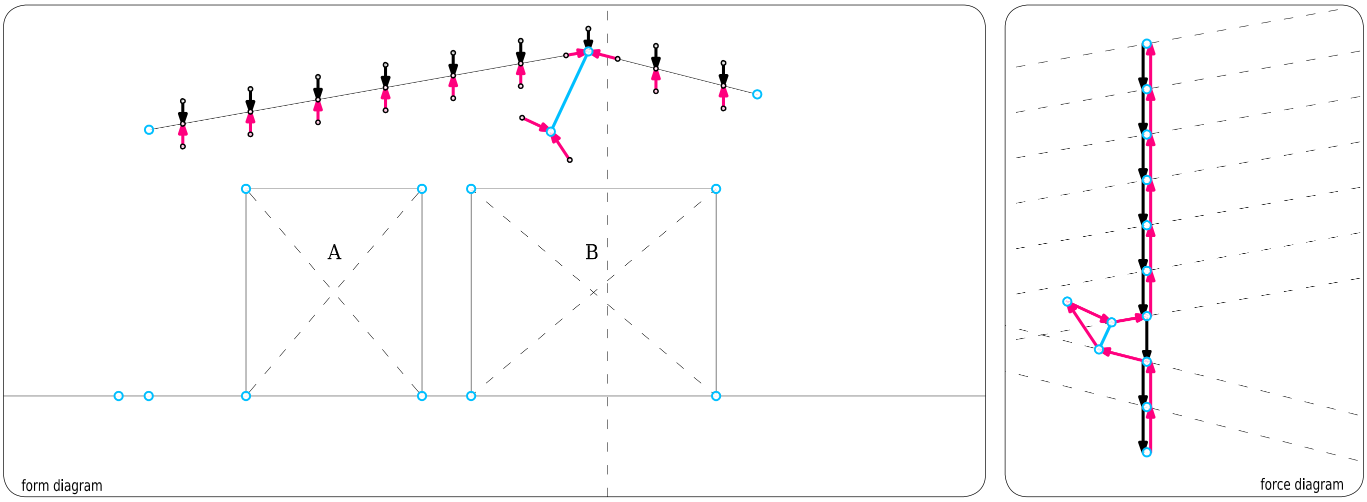

Figure 21 · Transformation of opposite forces into a compression rod.

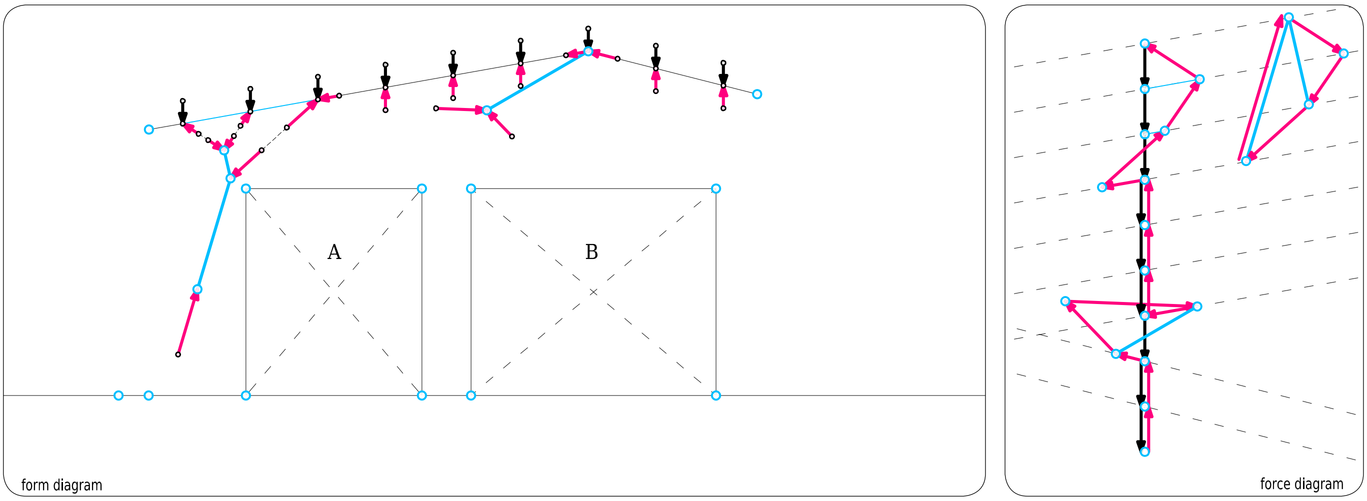

Figure 22 · New sub-network in equilibrium.

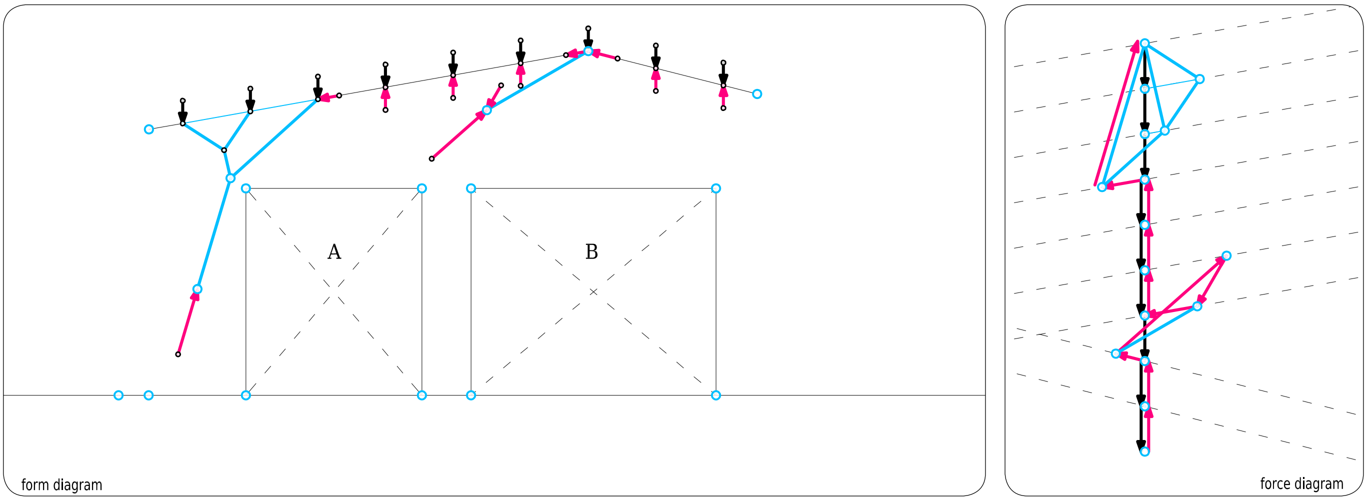

Figure 23 · Combination of two sub-networks..

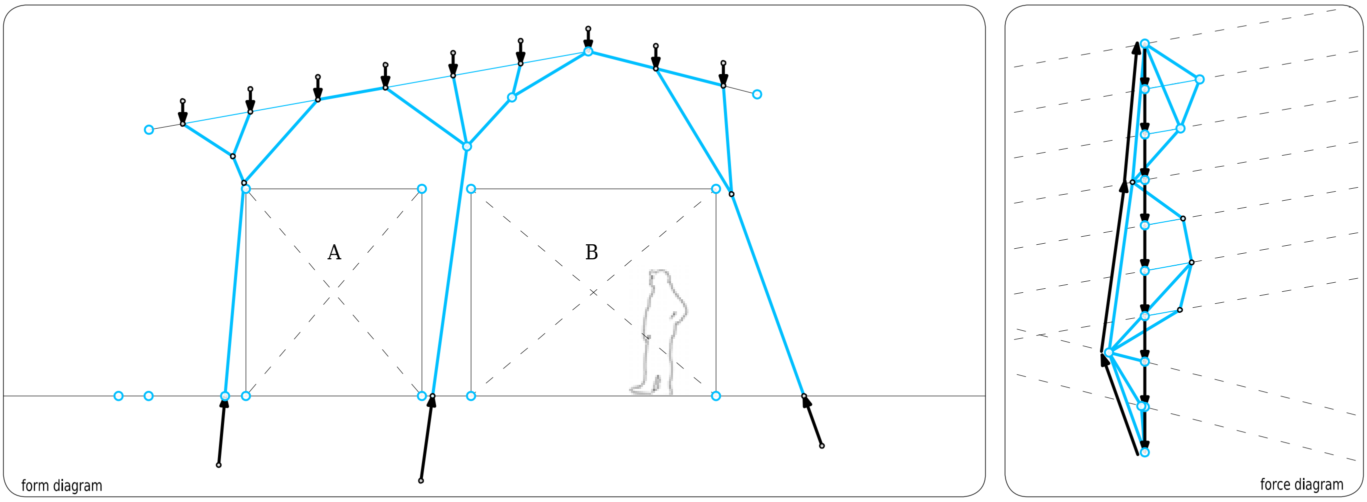

Figure 24 · Completed model.

Deforming the Model · When points have been added to the model and forces have been combined, geometric constraints are propagated by the computer so that each region inside which a point must stay reflects all the conditions applied to the model (Fivet·2014), be they related to the frames A and B or to the constructed typology. That means that the designer can deform the model by moving each point inside its domain of solutions while being ensured that the structure remains in equilibrium and satisfies every other applied constraint. Figure 25 and Figure 26 show some deformations together with the domain of the point that has been dragged.

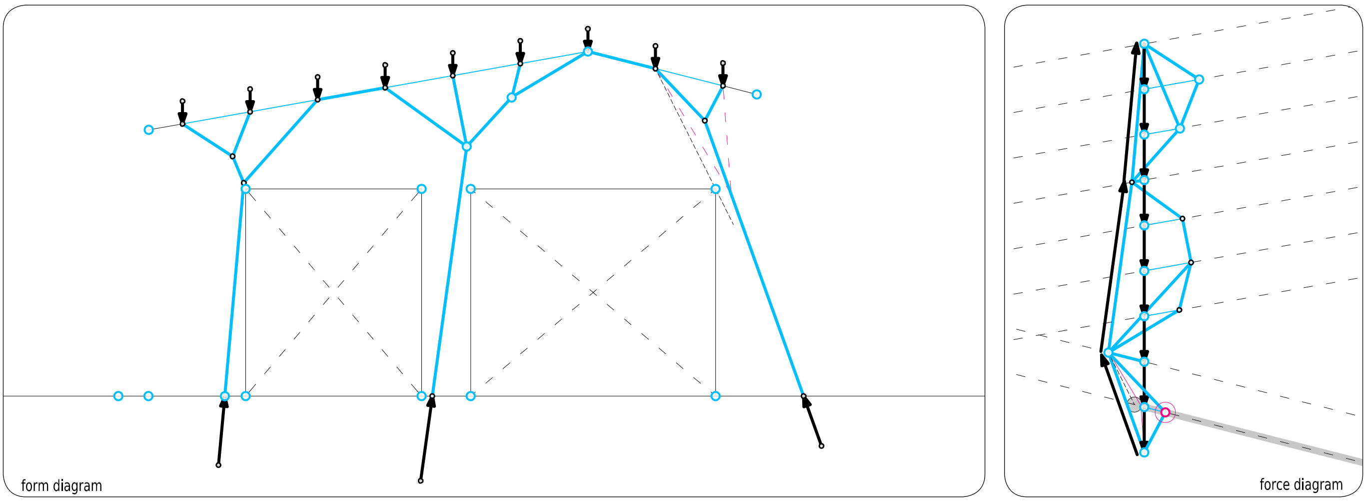

Figure 25 · The highlighted point has been dragged inside its domain (the gray half line).

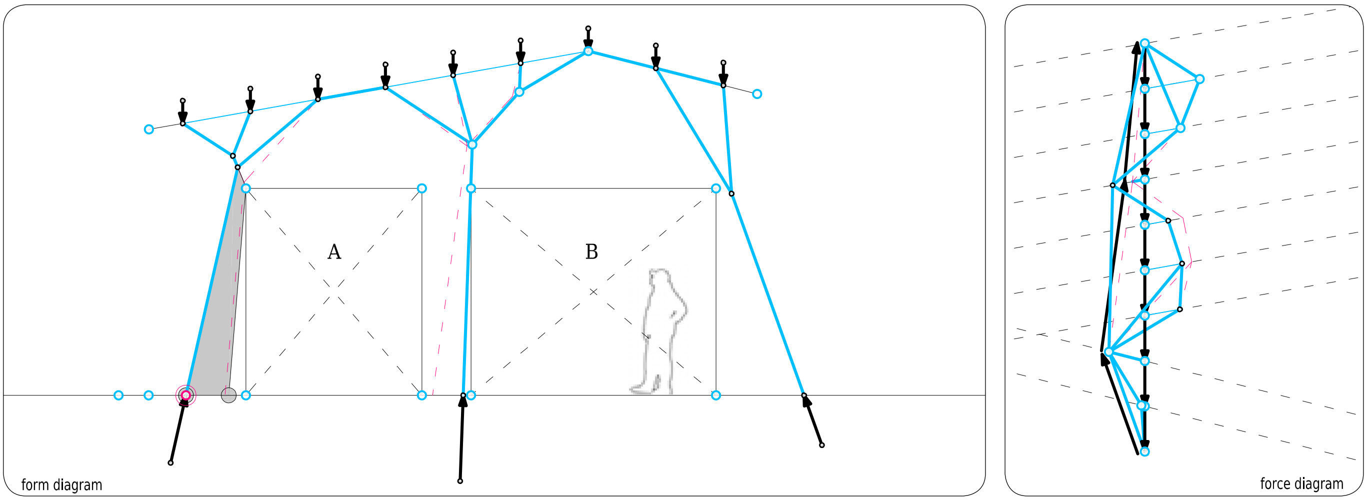

Figure 26 · The highlighted point has been dragged inside its domain (the gray region).

Constraining the Model Further · The designer can also add new explicit constraints to the model. For example, he may request every rod in the roof to be in traction and to be less than a given magnitude. This last request is here obtained by applying two half-planes in the force diagram (Figure 27). If disc constraints are applied on the form diagram, one can also request every rod to be less than a given length.

Sometimes it may happen that a constraint cannot be applied because it empties a domain of solutions, meaning that this typology cannot exist (see forces F_1 and F_2, and rod R* in Figure 27). The user then has to perform new operations (*e.g.* decompose a rod or reshape an entire sub-network) to produce a new possible structure (Figure 28).

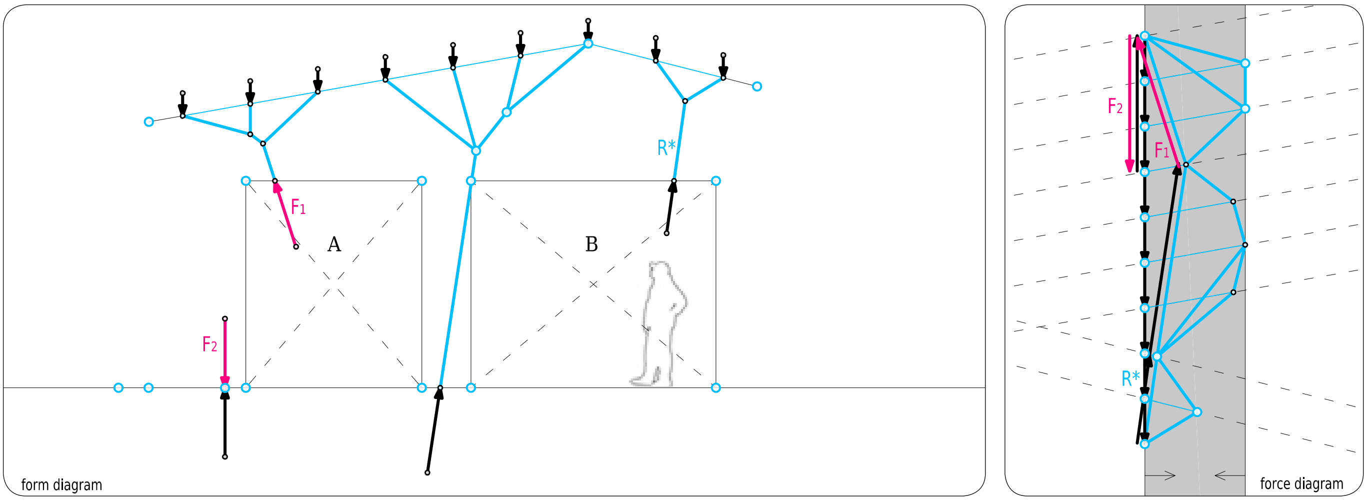

Figure 27 · Constraining the type of effort inside the roof.

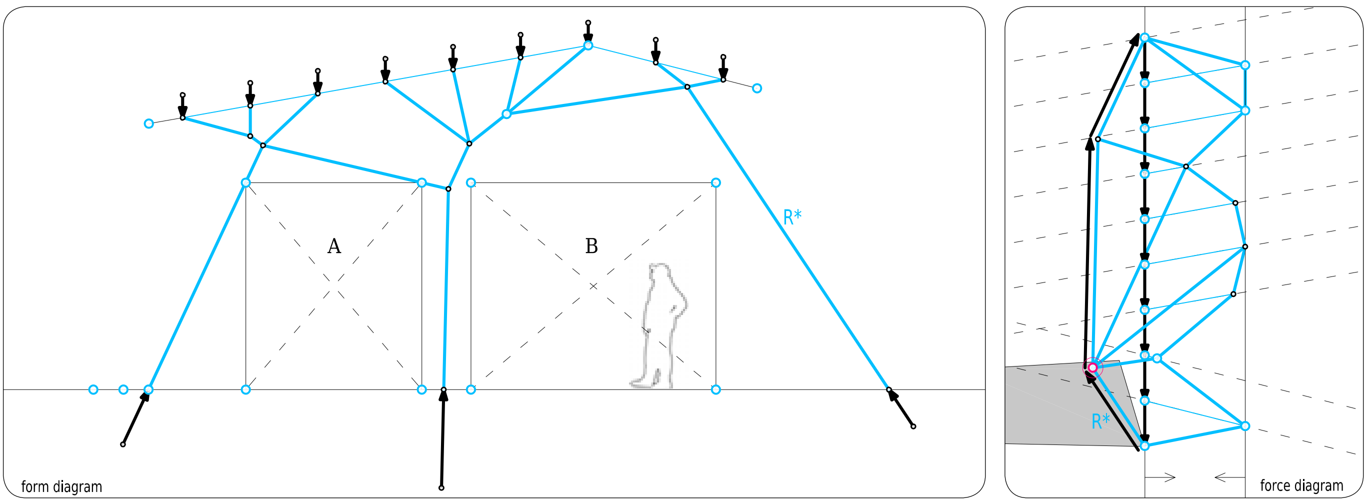

Figure 28 · Correcting the typology. The domain of the highlighted point is in gray.

Optimizing the Model · The domains of solutions can also be used to perform some quick optimizations. In the force diagram, points can be dragged to the edge of their domains until all the rods have minimum magnitude.

Points can also be dragged to obtain the configuration where the reactions inside the ground are as vertical as possible (Figure 29).

Discussing the Initial Requirements · The propagation of constraints is also constrained onto the points that define the initial requirements. Their domain of solutions consequently provide the entire set of initial values that lead to the desired structural result (Figure 30 & Figure 31). There is no distinction between inputs and outputs anymore.

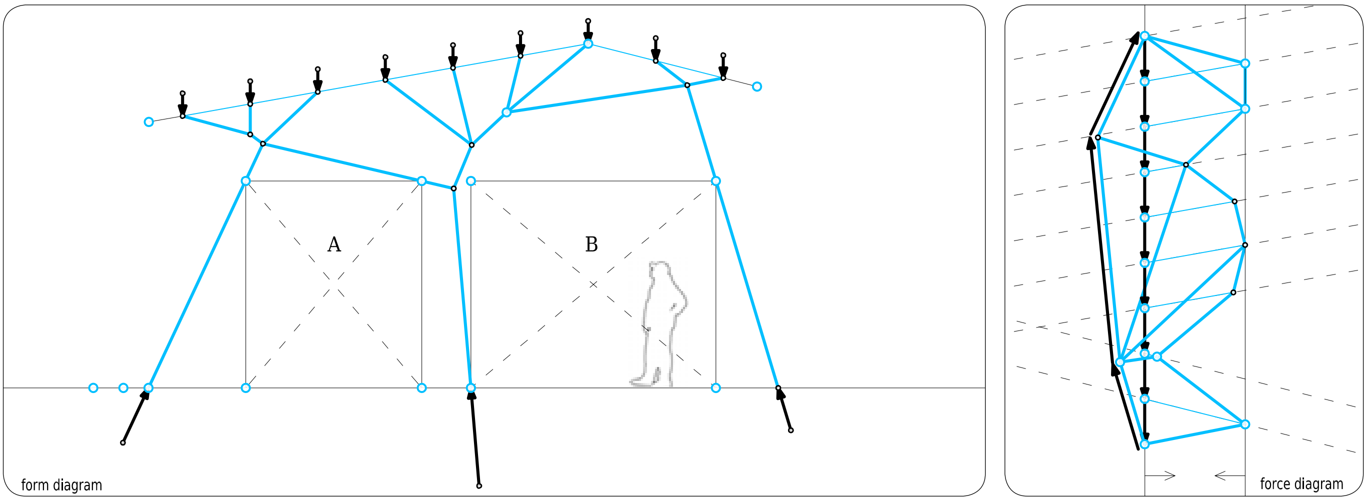

Figure 29 · Points have been dragged to make the end supports as vertical as possible.

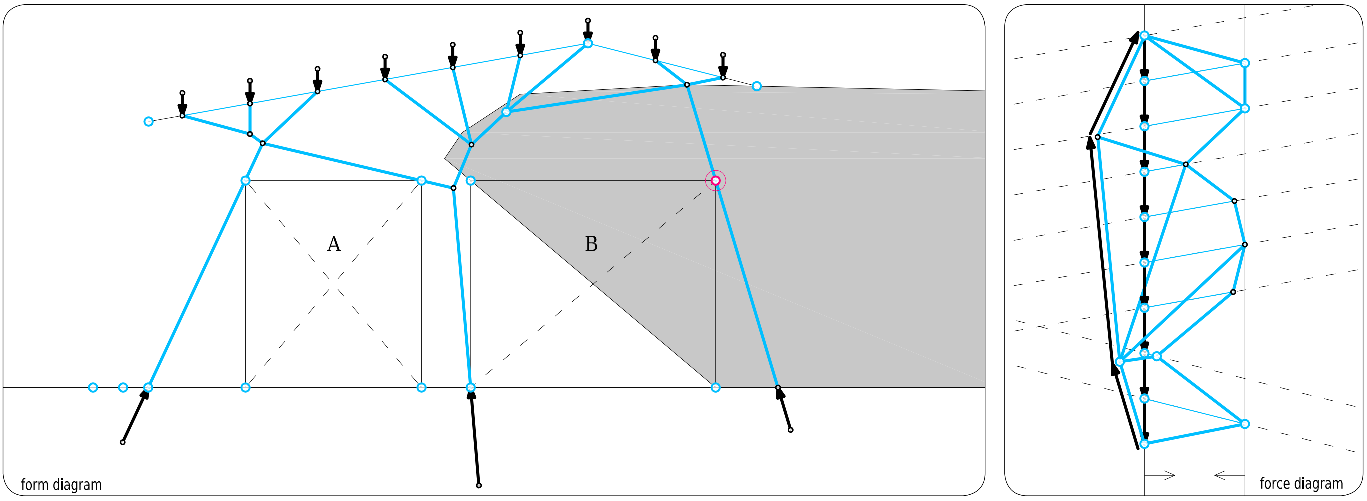

Figure 30 · The domain of solutions of one of the corners of the frame B.

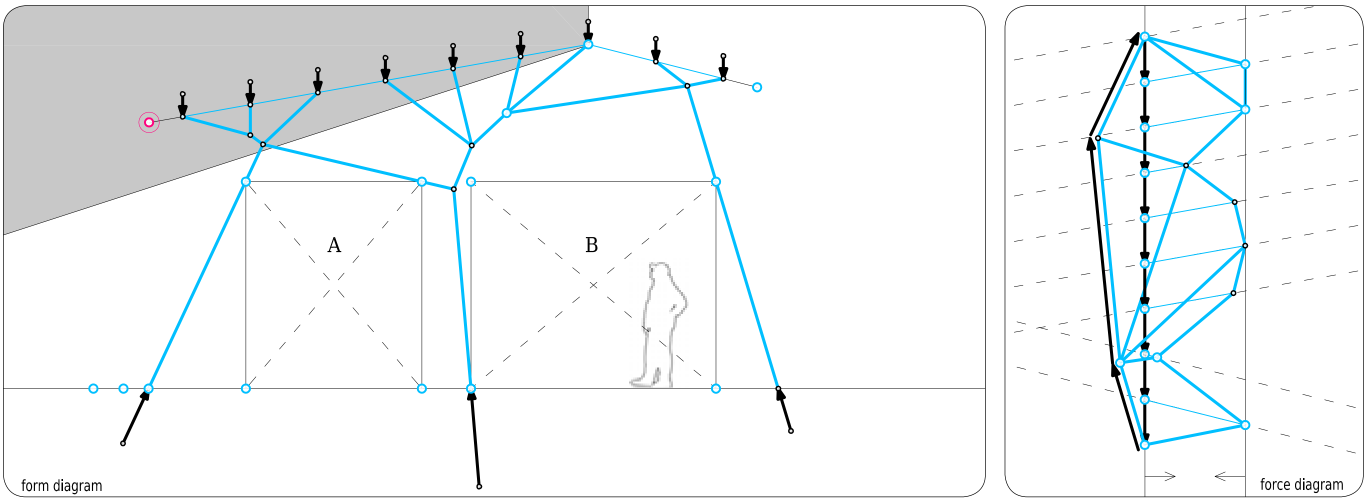

Figure 31 · The domain of solutions of one of the points that defines the slope of roof.

Swapping the Parametric Hierarchy · The computer uses priority when updating the position of points. If a point is dragged, all the 'children' points that have that point as a parameter are updated. All the 'grandchildren' points that have them as parameters are updated subsequently and so forth from one generation to the next.

Thanks to the symmetric properties of the geometric constraints, this parametric hierarchy can be here fully reorganized without having the rebuild the construction (Fivet·2014). This unprecedented capability completely vanishes the usual frustration when one must rebuild the entire parameterization because he wants a constrained point to become an initial parameter or vice versa.

Here, the designer only has to choose two points (*e.g.* a point and its grandfather point) and, as a result, the grandfather point will instantly become the child point of its grandchild. From the previous figure, the user can for instance directly swap the parametric hierarchy between point P_A and P_B. Point P_A was fully constrained before the operation, it had to remain on the intersection of two lines. Now (Figure 32) the designer can drag it and see how P_B and its other former parameters vary when it moves.

Discussing Other Load Cases · The generalization of solution domains and the ability to swap the parametric hierarchy open up the opportunity to discuss the impact of other load cases on the resulting model.

If snow happens to cover the roof, the load will increase uniformly. As a result the force diagram is scaled up uniformly and the geometry of the structure remains identical since parallelism is maintained.

However, if one point load varies in magnitude and orientation, the flow of forces behaves differently. The strut-and-tie network can help the study of these variations. As an illustration, a pulling force has been added on the right side of the roof. (Figure 33).

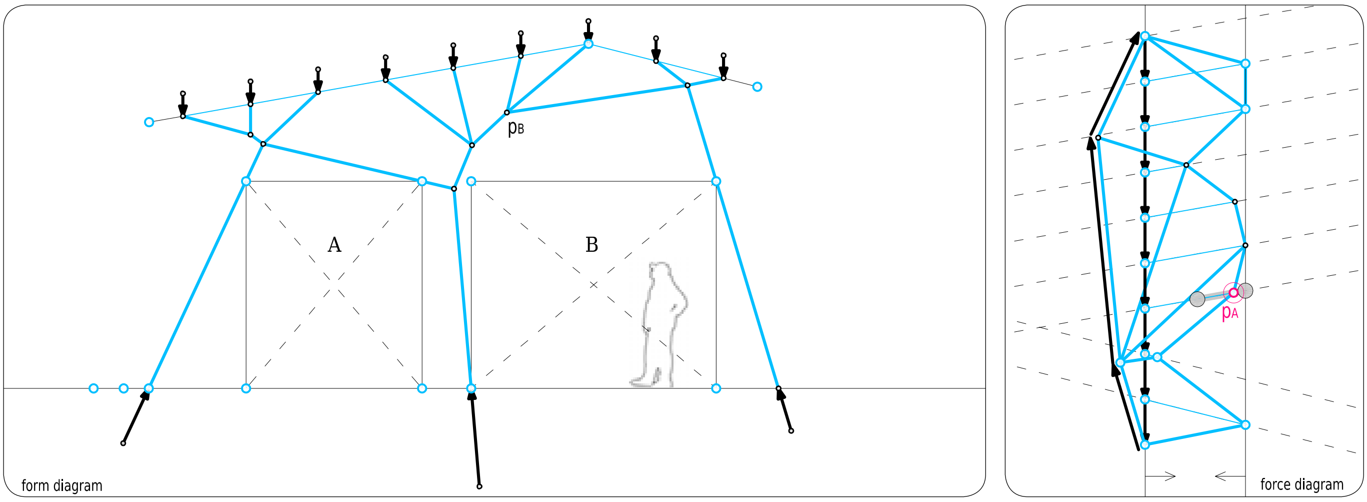

Figure 32 · The point p_A and its new domain of solutions.

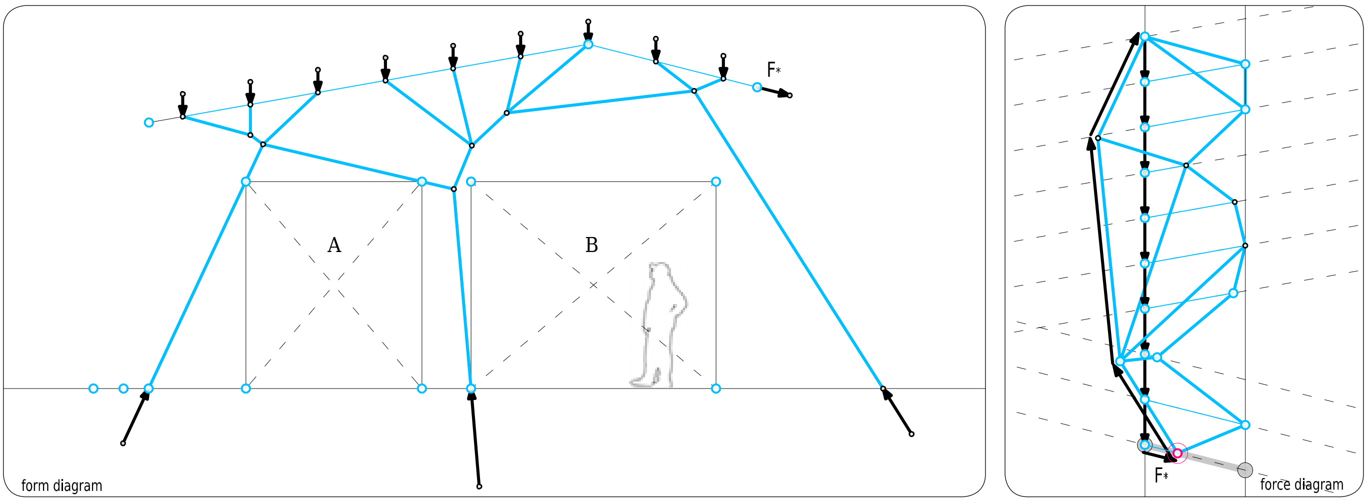

Figure 33 · The newly created ponit and its new domain of solutions.

Conclusion

After a short conceptualization expressing the role of the presented tool in the design process, the paper browsed the various key concepts and operations allowing the interactive shaping of forces. The third part of the paper developed a case study to highlight the practical benefits of constraint-based graphic statics.

In short, it has been shown that constraint-based graphic statics encourages the emergence of new structural design approaches that are highly interactive, that do not impose a specific sequence of resolution and that qualify all the solution domains before they are even explored. Compared to other form-finding tools, constraint-based graphic statics gives the designer the ability to provide personal inputs at each step of the exploration process and to redirect the process any time and to any desired direction.

Various developments are still needed before constraint-based graphic statics reaches its highest potential. Its extension to third dimension is certainly one of them. The automation of specific constructions of force flows and geometry may be another one, for instance to automate the subdivision of a uniform load into a given number of punctual loads, to handle discontinuous stress fields or to optimize given equilibria with graphical techniques.

More than a tool, this framework is also a fundamental theoretical background that paves the way for a new understanding of what a bearing structure is and how to describe its inner distribution of forces.

References

Beghini·2013 *Rankine's theorem for the design of cable structures.*
Alessandro Beghini / Lauren L. Beghini / Joshua A. Schultz / Juan Carrion / William F. Baker
Struct Multidisc Optim. · Springer-Verlag · Berlin, Heidelberg · DOI 10.1007/s00158-013-0945-2 · 2013

Conzett·2003 *Structure as Space: Engineering and Architecture in the Works of Jürg Conzett.*
Jürg Conzett / Moshen Mostafavi
AA Publications · 2003

Fivet·2012 *The Salginatobel Bridge design process by Robert Maillart (1929).*
Corentin Fivet / Denis Zastavni
Journal of the International Association for Shell and Spatial Structures · vol.53 · no.1 · pages 39-47 · 2012

Fivet·2013 *Constraint-Based Graphic Statics: new paradigms of computer-aided structural equilibrium design.*
Corentin Fivet / Denis Zastavni
Journal of the International Association for Shell and Spatial Structures · vol.54 · no.4 · pages 271-281 · 2013

Fivet·2014 *A fully geometric approach for interactive constraint-based structural equilibrium design.*
Corentin Fivet / Denis Zastavni
Computer-Aided Design Journal · http://dx.doi.org/10.1016/j.cad.2014.04.001 · 2014

Greenwold·2009 *Active Statics.*
Simon Greenwold / Edward Allen
http://ocw.mit.edu/ans7870/4/4.461/f04/module/Start.html · MIT · Boston · 2009

Lachauer·2012 *Compression support structures for slabs.*
Lorenz Lachauer / Philippe Block
Advances in Architectural Geometry 2012 · edited by Hesselgren, Sharma, Wallner, Badassini, Bompas, Raynaud
pages 135-146 · Springer · Wein, NewYork · 2012

Maxwell·1864 *On reciprocal figures and diagrams of forces.*
James Clerk Maxwell
Philosophical Magazine · vol.27 · no.4 · pages 250-261 · 1864

Mueller·2013 *An integrated computational approach for creative conceptual structural design.*
Caitlin Mueller / John Ochsendorf
Paper presented at the International Association for Shell and Spatial Structures (IASS) Symposium 2013
Wroclaw, Poland · September 23-27 · 2013

Rankine·1858 *A manual of applied mechanics.*
William John Macquorn Rankine
published by Richard Griffin · London, Glasgow · 1858

Rippmann·2013 *Funicular Shell Design Exploration.*
Matthias Rippmann / Philippe Block
Paper presented at ACADIA 2013 Adaptive Architecture · Cambridge, Ontario, Canada · October 24-26 · 2013

VanMele·2011 *eQuilibrium.*
Tom Van Mele / Philippe Block
http://block.arch.ethz.ch/equilibrium · ETHZ · Zürich · 2011

ViolletLeDuc·1883 *Dictionnaire raisonné de l'architecture française.*
Eugène Viollet-le-Duc
B.Bange Editeurs · Paris · vol.1 · 1883

VonBuelow·2012 *ParaGen: performative exploration of generative systems.*
Peter Von Buelow
Journal of the International Association for Shell and Spatial Structures · vol.53 · no.4 · pages 271-284 · 2012

Zalewski·1997 *Shaping structures.*
Waçlaw Zalewski / Edward Allen
Wiley & Sons · 1997