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 (for example, the parametric hierarchy can be freely reorganized), visual feedback and dynamic control over mechanical considerations.
INTRODUCTION

SHAPING THE STRUCTURAL FORM

Historical testimonies of structurally pertinent and efficient architectural works (Figure 1), (Figure 2) and (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 (for example 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: (a) guess a geometry and other initial parameters (b) perform the structural analysis (respectively, the automated form-finding or optimization) (c) interpret the result, i.e. the resulting stresses (resp. the resulting geometry); (d) guess a change/refinement of geometry (resp. of contextual constraints) in order to enhance the next result; and (e) 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, (Von Buelow 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.

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.

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, for example 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; and equilibrium of stabilizing slopes, retaining walls and foundations. Because they use particular theories (for instance 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: (a) 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 (b) nodes, considered as the only variables, are constrained within graphical regions that represent the entire domain of solutions.

The three following sections (a) recall the meaning of the two diagrams of graphic statics, (b) explain how the particular transformations of static equilibria are ruled and (c) 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 Drawing (b) is parallel to and of equal magnitude of a force or a rod in Drawing (a). Since each node is in equilibrium, each vectorial addition must form a close cycle. Moreover, if this is true for every node in Drawing (a), both the translational and the rotational equilibrium of the entire structure in Drawing (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 Drawing (a) is always either clockwise or counter-clockwise.

The transition between Drawing (c) and Drawing (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 Drawing (d) provides the magnitude inside each bar and informs that the structure in Drawing (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 1858s works of J.W.M. Rankine (Rankine 1858, 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 tools 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 has been a comeback of graphic statics both in education (Zalewski 1997) and in practice (Beghini 2013; Conzett 2006, 100-109).

The recent enhancement of graphical user interfaces has 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.

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) and

---

4 Graphic Statics—intermediate relationships between the form diagram (a) and the force diagram (d)

5 Top: Operations—introduction of two forces in equilibrium

6 Bottom: Operations—the force \( F_1 \) has been resolved into the force components \( F_2 \) and \( F_3 \)

7 Top: Operations—forces \( F_A \) and \( F_B \) are opposite, parallel and of equal magnitude. They are pushing on their point of application.

8 Bottom: Operations—forces \( F_A \) and \( F_B \) have been combined into a new compression rod
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).

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 (for instance 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.

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.

**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 2013) 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.
Case Study–alignment of forces with the slope of the roof

Case Study–new set of forces in equilibrium

Constraints–the highlighted point in the force diagram remains inside the gray area, forces F0, F1, F2 and F3 will be read clockwise according to that order. Otherwise, pairs of forces must be swapped using a specific fundamental operation.

Case Study–introduction of the initial loads and of the boundary conditions (defining the gray area inside which every point has to stay)

Constraints–hidden primitive constraints that are applied in order to make p* stay on an ellipse centered in p0 and having p1 as focal point

Case Study–decomposition of a force

Reading Cycle–if the highlighted point in the force diagram remains inside the gray area, forces F0, F1, F2 and F3 will be read clockwise according to that order. Otherwise, pairs of forces must be swapped using a specific fundamental operation.

Case Study–alignment of forces with the slope of the roof

Constraints–the highlighted point in the force diagram must remain inside the gray region for the rod to be in traction

Case Study–new set of forces in equilibrium
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.

**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).

**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.
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 F1 and F2, and rod R* in Figure 27). The user then has to perform new operations (for example, decompose a rod or reshape an entire sub-network) to produce a new possible structure (Figure 28).

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) and (Figure 31). There is no distinction between inputs and outputs anymore.

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 (for instance a point and its grandfather point) and, as a result, the grandfather point will instantly become the child point of its grandchild. From
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).

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


IMAGE CREDITS


Figures 3-33. Image credits to Corentin Fivet and Denis Zastavni (2014).

CORENTIN FIVET holds a master in Architectural Engineering and a PhD in Engineering Sciences from the UCLouvain, Louvain-la-Neuve, Belgium. His main interests are at the intersection of architecture and structural engineering, with a greater focus on the development of innovative graphical tools for structural design. During his PhD, he developed Constraint-Based Graphic Statics for which he was awarded the IASS Hangai Prize. He also had the opportunity to be a teaching assistant in a number of architectural design studios as well as a member of various studio juries. He is currently pursuing post-doctoral fellowship in the Structural Design Lab at MIT.

DENIS ZASTAVNI is Dr. Arch. Eng. and Associate Professor at the Faculty of Architecture, Architectural Engineering and Urbanism [LOCI] of the Université Catholique de Louvain at Louvain-la-Neuve in Belgium. He has worked as a structural engineer and architect for over ten years. His PhD was on Robert Maillart’s design methods and won the Premio Edoardo Benvenuto in 1998. His main publications are on structural design, pedagogical approaches in teaching structure and technology, and Robert Maillart’s designs. His current research is focused on structural design approaches and tools and on deepening his knowledge of Robert Maillart’s structural methods.