IASS 2020/21 SURREY 7
International Conference on Spatial Structures 2020/21

Proceedings of the IASS Annual Symposium 2020/21 and the 7th International Conference on Spatial Structures
Inspiring the Next Generation
23 – 27 August 2021, Guilford, UK
S.A. Behnejad, G.A.R. Parke and O.A. Samavati (eds.)

Geometry-based Teaching of Structures Through Computational Graphic Statics

Juney LEE*, Lluís ENRIQUE, Tom VAN MELE, Philippe BLOCK

* ETH Zurich, Institute of Technology in Architecture, Block Research Group
Stefano-Franscini-Platz 1, HIB E 45, 8093 Zurich, Switzerland
juney.lee@arch.ethz.ch

Abstract

This paper presents how computational graphic statics can be used in university-level architecture and structural engineering curriculums for geometry-based teaching of structures. Compared to a more traditional paradigm of teaching structural design in academia where dimensioning and analysis of structures are prioritised through numerical calculations, graphic-statics-based approach enhances students' intuition and understanding of structures through graphical representations of the relation between a structure and its internal forces through reciprocal form and force diagrams. Visual representation of equilibrium enables students to challenge how typical structural design problems are addressed and provides tangible insights into how designs can be improved. By combining the inherent benefits of graphic statics with today's computational tools, interactive drawings of form and force diagrams can be constructed, which students can manipulate with real-time, visual feedback. Computational implementations of graphic statics are not only useful for educational purposes in more technical courses, but also for projects in architectural design studios as a new canvas and a kit of parts to design buildings in a more insightful and structurally-informed manner. The development of the presented computational graphic statics platforms and tools for teaching structures is based on the COMPAS framework. The potential of geometry-based teaching of structures using computational graphic statics is demonstrated through examples of student work from a new computational graphic statics course that was developed and taught at ETH Zurich in the fall semester of 2020. The detailed content, structure and teaching methodology of the course are also presented.

Keywords: computational graphic statics, graphic statics, computational design, structural design, teaching

1. Introduction

Graphic statics is a graphical method of design and analysis for two-dimensional (2D) discrete structure that relies on geometrical rather than numerical representations of the relation between a structure's geometry and the equilibrium of its internal forces (Maxwell [11], Culmann [5], Cremona [6]). Graphical representation of equilibrium using reciprocal form and force diagram provides an intuitive and insightful method of understanding a structure's behaviour through visualisation of forces that is easier to digest and more transparent than conventional, analysis-based methods of structural design (Allen and Zalewski [2]). Graphic statics gradually disappeared from structural engineering practice and education over the 20th century due to the advancement of powerful computational analysis tools, which made the tedious process of manually drawing form and force diagrams impractical and obsolete.

In recent years, there has been a renewed interest and rise of research within the field of graphic statics, mainly due to the new design and research possibilities that arise when graphic statics is combined with advanced parametric tools and computer-aided design (CAD) software (Van Mele *et al.* [14]). Computational tools can enable interactive implementations of graphic statics, where the users can

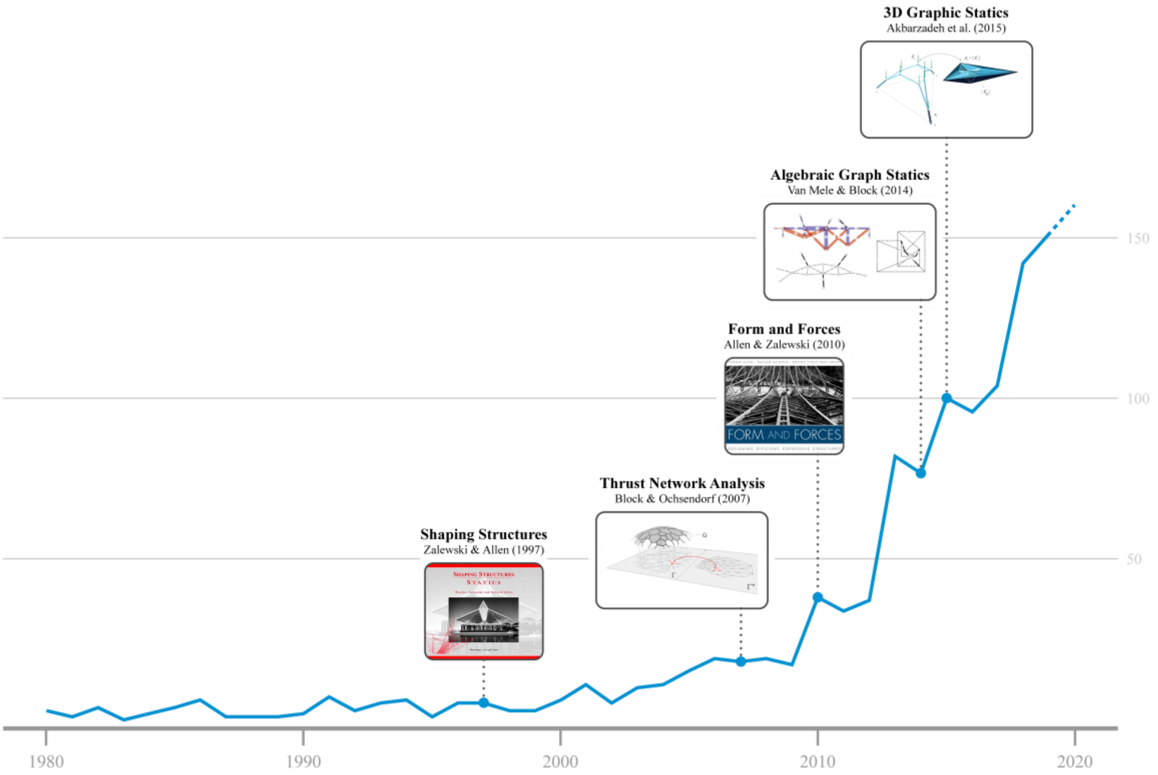

Figure 1: Recent rise in the number of graphic statics related publications and some of the notable contributions to the development of computational graphic statics based on data from Google Scholar [9].

interact with form and force diagrams with real-time, visual feedback. Over the past few decades, the annual number of scholarly publications related to graphic statics have increased at an exponential rate, which continues to expand the research and design potential of computational graphic statics (Figure 1).

2. Computational graphic statics course

Beyond design and research, the benefits of computational graphic statics can also be used for developing innovative methods of teaching structures in academia. Led by Professors Philippe Block of the Block Research Group (BRG) and Professor Joseph Schwartz of the Chair of Structural Design at ETH, graphic statics is used to teach structural design to architecture students across all levels. Continuing the legacy of former ETH educators including Karl Culmann, Karl Wilhelm Ritter, and Pierre Lardy, the goal of computational-graphic-statics-based teaching tools and platforms is to "whiten the black box," making the methods and their computational implementations explicit and intuitive to the students during early stages of learning and conceptual design process (Enrique *et al.* [7]).

In the fall semester of 2020 at ETH, BRG developed and taught a new master and doctoral level course on computational graphic statics titled Computational Structural Design 1 (CSD1). First of its kind, this new course was designed to introduce students to new structural design opportunities that emerge when the intuitive techniques of graphic statics are combined with computational tools. After briefly reviewing the fundamentals and basic principles of graphic statics, the contents of the course are focused on exploiting the student's ability to interact with form and force diagrams within a computational design environment, in order to explore new structural design methods that are simply not possible with conventional tools. The ultimate goal of the course is to present to the students the practical potential and relevance of the latest research and developments in computational graphic statics, which are demonstrated through various design-oriented tutorials and exercises.

3. Course content

This section summarises the learning goals, content structure and the computational teaching tools of the course.

3.1. Learning goals

After successfully completing the course, the students of CSD1 will have learned:

- fundamental principles of graphic statics;
- basic parametric modelling with Rhino and Grasshopper, and construction of interactive form and force diagrams;
- introductory skills in Python scripting and basic topological thinking by using the datastructures of the COMPAS framework;
- how to design and analyse 2D trusses using Algebraic Graphic Statics;
- how to design, form find and analyse funicular shell structures using Thrust Network Analysis and RhinoVAULT 2;
- how to design, form find and analyse spatial structures using polyhedral 3D Graphic Statics; and
- application of computational graphic statics in various design contexts.

3.2. Course structure

In order to give a broad overview of the latest developments in computational graphic statics and their relevant applications, the course is divided into six modules each with a specific focus, gradually building up complexity from single-node 2D structures to multi-node 3D structures (Figure 2). Each module will be described in detail in Section 4.

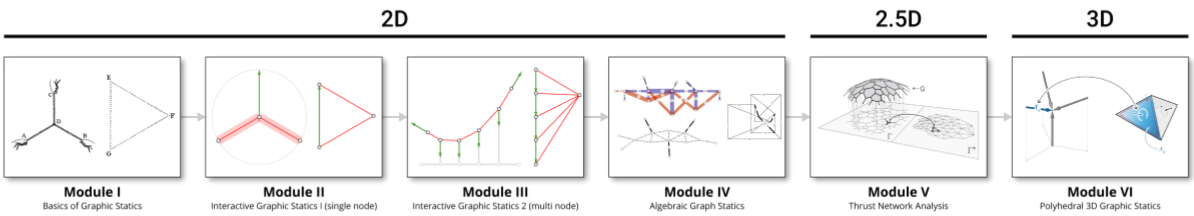

Figure 2: The six modules of the course progressing from 2D graphic statics to 3D graphic statics.

3.3. Computational graphic statics tools

The advanced topics (modules IV-VI) of the course are taught using computational graphic statics tools developed by the BRG with the COMPAS framework (Van Mele *et al.* [16]). These tools are open-source plugins for Rhino (McNeel [11]) based on previous and ongoing research on graphic statics within the BRG (Figure 3). These tools are freely downloadable through the GitHub repositories of the BRG (https://github.com/BlockResearchGroup). Each tool will be described in detail in the Section 4.

4. Selected student work

This section presents some of the exemplary student work from each course module.

4.1. Module I: Basics of graphic statics

The very first module of the course reviews the fundamental principles of graphic statics, such as construction procedure of a force diagram for a single-node structure, interpretation of forces as compression or tension, resultant of forces, etc. These basic graphic statics concepts are demonstrated on the chalkboard (Figure 4a) as well as through the online graphic-statics-based learning environment of eQUILIBRIUM [8], and students are asked to complete simple hand-drawn exercises to get familiar with the basic principles and the procedural geometric construction process of form and force diagrams.

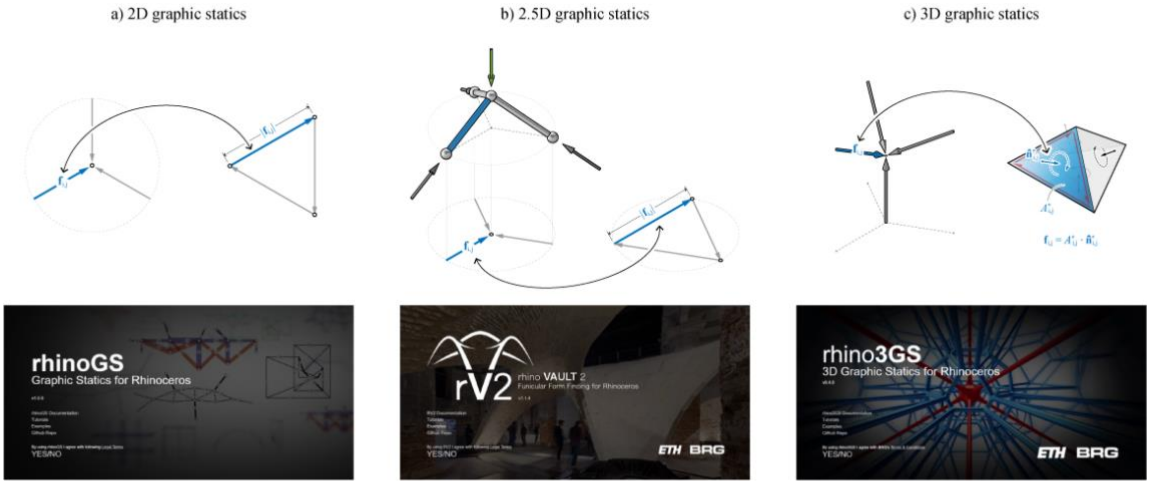

Figure 3: Computational graphic statics tools used for the advanced modules of the course:a) *rhinoGS* for non-procedural 2D graphic statics; b) "2.5D" graphic statics explorations of funicular shell structures using *RhinoVAULT 2*; and 3) form-finding explorations of spatial structures using *rhino3GS*.

4.2. Module II: Interactive graphic statics 1 (single-node structures)

The second module of the course introduces students to Rhino and teaches how the linear nature of the visual programming environment of Grasshopper can be used to replicate the procedural construction process of form and force diagrams for a single-node structure from Module I. Students learn that with a specific design intention or application in mind, interactive form and force diagrams can be constructed in such a way that maximises the benefits of parametric modelling for the design task at hand. Students also learn that various geometric objects can be used to impose simple force-driven constraints through the force diagram. For example, by using two circles, students are able to visually control and limit the magnitude of the maximum force (equivalently, the radius of the circles) in the two diagonal members of the single-node structure (Figure 4b).

4.3. Module III: Interactive graphic statics 2 (multi-node structures)

The third module extends procedural construction of interactive graphic statics drawings from single to multi-node structures. By directly juxtaposing the repetitive procedures involved in the construction of form and force diagrams for multi-node funicular structures with simple programming techniques, students develop a design-oriented understanding of basic programming concepts within the context of graphic statics.

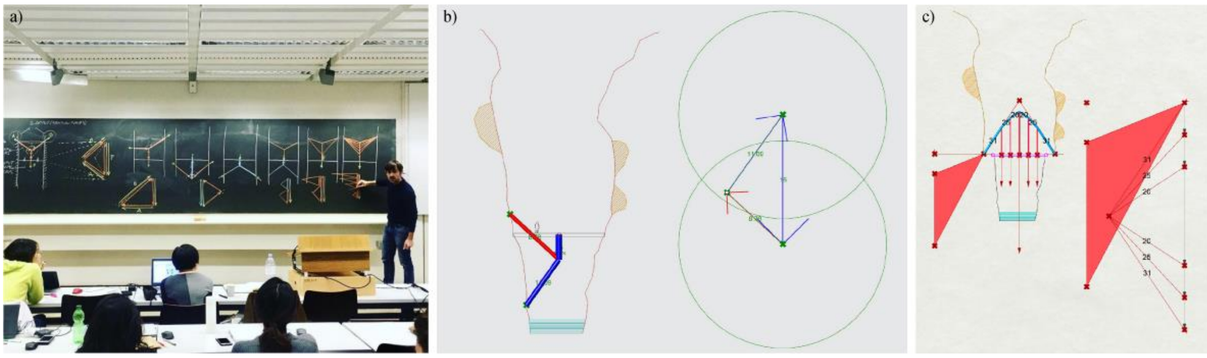

Figure 4: Modules I-III - a) Review of fundamental principles of graphic statics on chalkboard; b) procedural construction of the form and force diagram of a single-node bridge with basic geometric constraints to control force magnitudes; and c) procedural construction of the form and force diagrams of a multi-node bridge with more sophisticated force-based constraints.

Through exploration of more discretised multi-node funicular structures, students learn how to use interactive graphic statics drawings to understand and analyse more realistic structures. Furthermore, students are taught how to design structures with force-driven constraints by geometrically defining restricted or allowable areas within the form and force diagrams, that satisfy certain force-driven design constraints such as limiting horizontal reaction forces of an arch bridge (Figure 4c).

4.4. Module IV: Algebraic Graphic Statics

The fourth module moves on from the visual, procedural programming environment of Grasshopper to a fully scripting environment using Python. Although in-depth programming is beyond the scope of this course, students are introduced to basic programming concepts that enable automatic construction of form and force diagrams by using an interactive implementation of Algebraic Graph Statics (AGS) (Van Mele and Block [15]), rhinoGS (Avelino *et al.* [3]). Students learn how to interpret form and force diagrams as computational datastructures, and how to explore various boundary condition constraints and parameters to design and analyse trusses.

By using rhinoGS, students can now automatically generate form and force diagrams for complex truss structures. Beyond analysis, one of the most effective ways of teaching the computational principles behind AGS is to ask each student to design a truss of their choice and explain its behaviour through the corresponding force diagram (Figure 5). The computational implementation of each of the steps of rhinoGS workflow is designed to be intentionally explicit and strictly dependent on successful completions of the previous steps. For example, setting up a form diagram using rhinoGS requires that the diagram does not have any crossing edges and all of the externally applied and reaction forces are drawn as edges. Through the trial-and-error process of experimenting with various connectivity of lines, students quickly learn that the geometry and topology of the initial set of edges need to be carefully considered for graphic-statics-based structural design workflow.

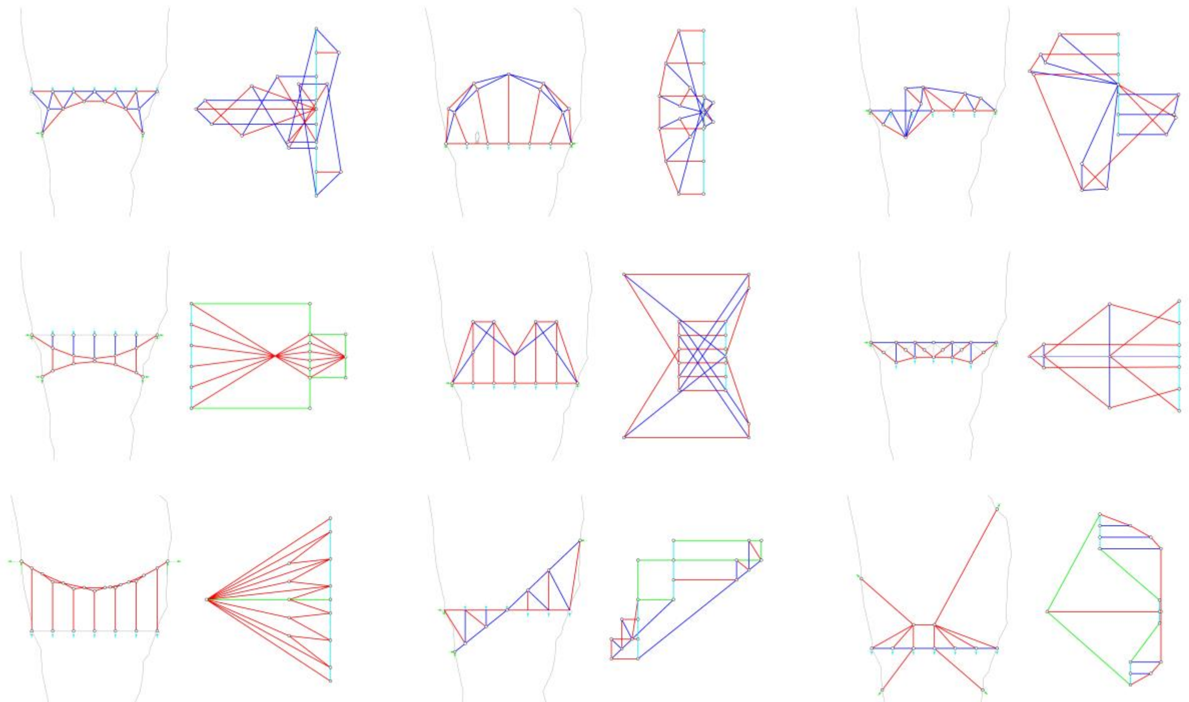

Figure 5: Module IV - Design exercise using rhinoGS, where students were encouraged to think beyond known structural typologies while always being aware of the internal force distribution, force magnitudes and the overall structural behaviour.

4.5. Module V: Thrust Network Analysis

The fifth module of the course introduces Thrust Network Analysis (TNA) (Block and Ochsendorf [4]), a graphical method for form-finding of funicular shell structures using horizontal projections of its geometry as the form diagram. The design principles and strategies using TNA are demonstrated through RhinoVAULT 2 (RV2), an interactive implementation of TNA for Rhino. RV2 is an open-source research and development platform for funicular form-finding built with COMPAS and replaces RhinoVAULT (Rippmann, Lachauer and Block [12]) for Rhino versions 6 and above. Students learn how to use various force-driven graphic statics design techniques to analyse and design complex shell structures in an interactive and intuitive manner. By controlling the geometry of funicular shell structures through manipulations of corresponding force diagrams shows the students the power of graphic-statics-based computational form finding and its advantages over other conventional alternatives that are based on numerical solvers.

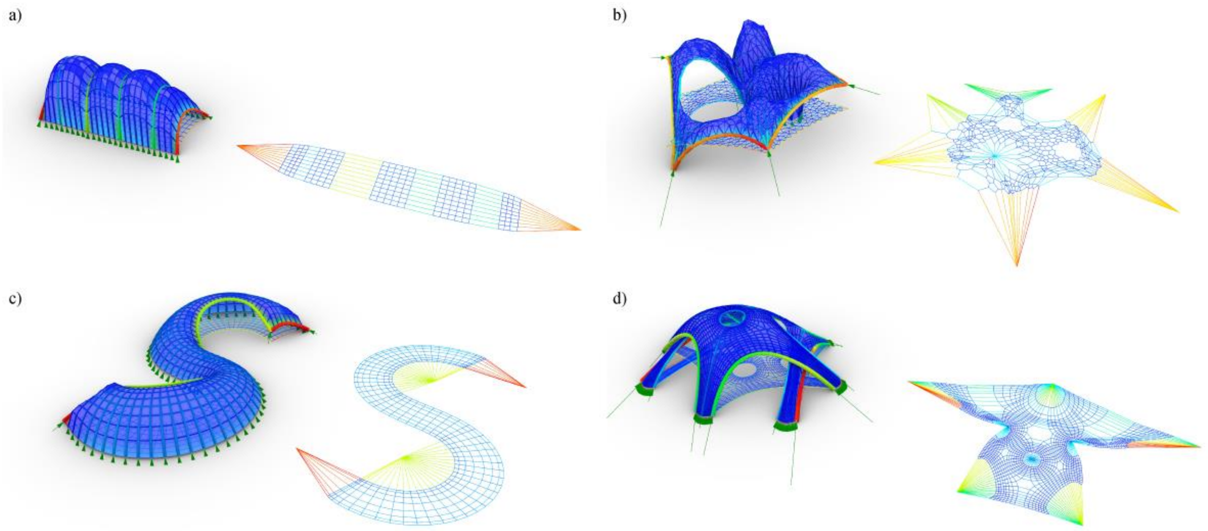

Figure 6: Module V - a) Generation of creases by elongating edges in the force diagram; b) exploration of various boundary condition parameters (openings, identification of supports and their z-coordinates); c) use of the “skeleton” feature to design a shell structure following the first letter of their names; and d) design of a free-form shell structure incorporating all of the design techniques.

4.6. Module VI: Polyhedral 3D graphic statics

The sixth and the last module of the course introduces 3D graphic statics based on polyhedral form and force diagrams (Akbarzadeh, Van Mele and Block [1]). Using rhino3GS, a computational design framework and plugin for 3D graphic statics (Lee [9]), students learn the similarities and differences between the aforementioned vector-based graphic statics and polyhedral extensions in 3D (Figure 7a, b). Students also learn how to exploit the inherent spatial geometries of polyhedral form and force diagrams to develop structurally-informed materialisation and digital fabrication techniques (Figure 7c).

5. Conclusion

This paper presented the contents of a new structural design course developed and taught at ETH in the fall semester of 2020 to master and doctoral level students from architecture and structural engineering departments. By combining fundamental design and analysis techniques of graphic statics with computational tools, the course demonstrated the potential of computational graphic statics as an innovative, geometry-based teaching of structures.

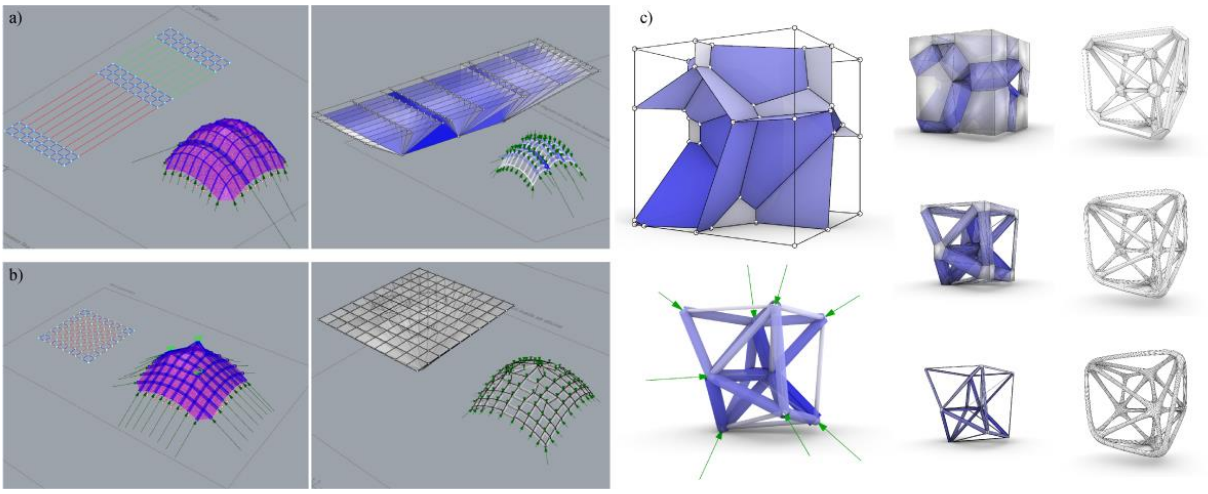

Figure 7: Module VI - a) comparing the process of generating creases using RV2 and Rhino3GS; b) comparing the consequence of imposing point loads using RV2 and Rhino3GS; and c) materialisation of spatial structures using unified diagrams and 3D skeleton features.

Acknowledgements

The images and selected student work from Section 4 are credited to:

- Figure 4: a) photo by Philippe Block; b) Lorenzo Lamarche; c) Blanka Major
- Figure 5, from left to right, row by row: Nicolas Carlier, Philip Meile, Jonas Schuepbach, Michele Capelli, Minu Lee, Blanka Dominika Major, Félix Dillmann, Joel Hösle, Rico Muth
- Figure 6: a) Sebastian Meier; b) Minu Lee; c) Stanislaw Modrzyk; d) Michele Capelli
- Figure 7: a), b) Xingyu He; c) Pedro Tosatto

The authors would also like to acknowledge the effort and work of PhD students (Lotte Aldinger, Ricardo Maia Avelino, Alessandro Dell'Endice, Gene Ting-Chun Kao, Francesco Ranaudo, Kam-Ming Mark Tam) and research assistants (Chaoyu Du, Wenqian Yang) of the BRG for the development and preparation of this course.

References

[1] Akbarzadeh M., Van Mele T. and Block P., On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams. *Computer-Aided Design*, 2015; **63**; 118-128.

[2] Allen E. and Zalewski W., *Form and Forces: Designing Efficient, Expressive Structures*. John Wiley Sons, New York, 2010.

[3] Avelino et al., *An interactive implementation of algebraic graphic statics for geometry-based teaching and design of structures*. Proceedings of the International fib Symposium on Conceptual Design of Structures 2021, Switzerland, 2021.

[4] Block P. and Ochsendorf J., Thrust Network Analysis: A new methodology for three dimensional equilibrium. *Journal of the IASS*, **48** (3), 2007, pp. 167-173.

[5] Culmann C., *Die graphische Statik*. Verlag von Meyer & Zeller, Zurich, 1864.

[6] Cremona L., Graphical Statics: Two Treatises on the Graphical Calculus and Reciprocal Figures in Graphical Statics. Clarendon Press, Oxford, 1890.

[7] Enrique et al., Design-oriented approach to teach structures in architecture based on graphic statics. *Proceedings of the IASS Symposium 2019*, Barcelona, Spain, 2019.

[8] 'eQUILIBRIUM', 2010. [Online]. Available: http://block.arch.ethz.ch/equilibrium. [Accessed: 15-Apr-2021].

[9] 'Google Scholar'. [Online]. Available: https://scholar.google.com. [Accessed: 15-Apr-2021].

[10] Lee J., *Computational Design Framework for 3D Graphic Statics*. PhD thesis, ETH, Zürich, Switzerland, 2018.

[11] Maxwell J.C., On reciprocal figures and diagrams of forces. *Philosophical Magazine and Journal Series*, 1864; 4(27); 250–261.

[12] McNeel, R., *Rhinoceros: NURBS modeling for Windows*, computer software. 2021. http://www.rhino3d.com/

[13] Rippmann M., Lachauer L. and Block P., Interactive vault design. *International Journal of Space Structures*, 2012; **27** (4); 219-230.

[14] Van Mele T., Lachauer L., Rippmann M. and Block P., Geometry-based understanding of structures. *Journal of the International Association of Shell and Spatial Structures*, **53** (4): 285-295, 2012.

[15] Van Mele T. and Block P., Algebraic graph statics. *Computer-Aided Design*, 2014; **53**: 104-116.

[16] Van Mele et al., COMPAS: A framework for computational research in architecture and structures, 2017. http://compas-dev.github.io/compas/