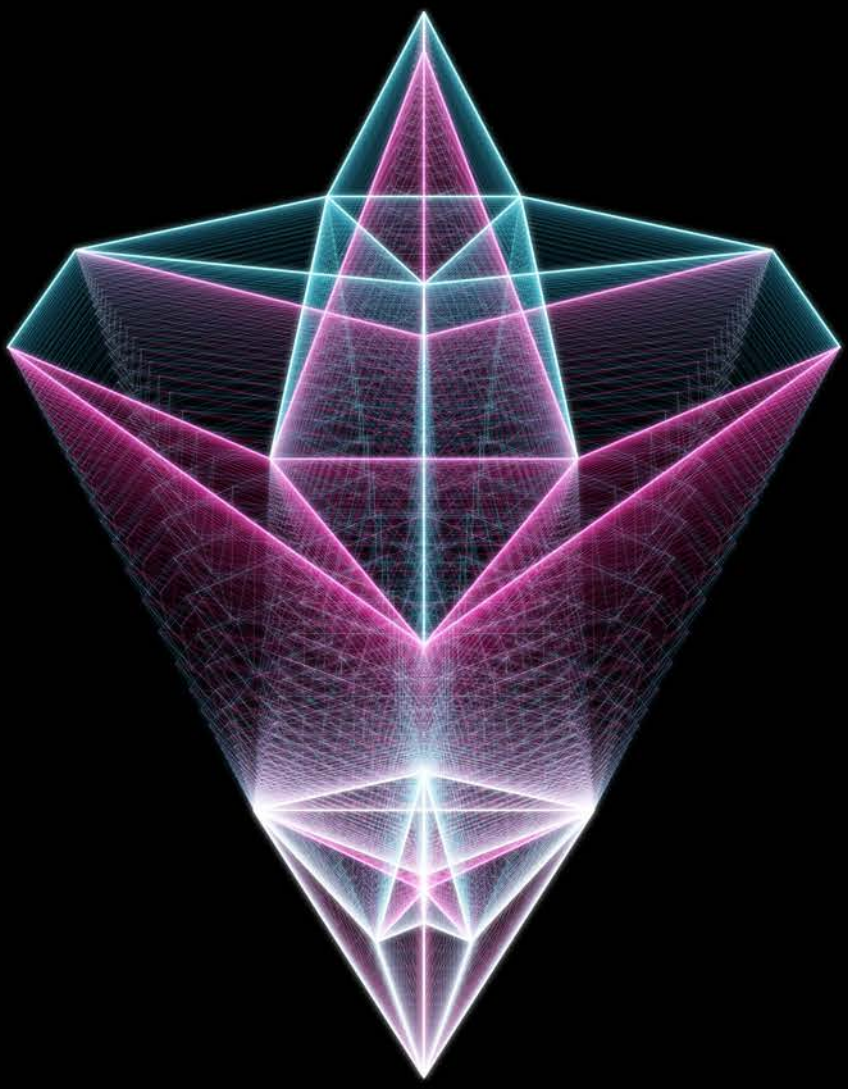
Computational Design Framework for 3D Graphic Statics

Juney Lee



ETH Zurich • PhD Dissertation No. 25526

Diss. ETH No. 25526
DOI: 10.3929/ethz-b-000331210

Computational Design Framework
for
3D Graphic Statics

A thesis submitted to attain the degree of

Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by

Juney Lee
2015 ITA Architecture & Technology Fellow

Supervisor

Prof. Dr. Philippe Block

Technical supervisor

Dr. Tom Van Mele

Co-advisors

Hon. D.Sc. William F. Baker
Prof. Allan McRobie

PhD defended on October 10th, 2018
Degree confirmed at the Department Conference on December 5th, 2018
Printed in June, 2019

For my parents who made me,

for Dahmi who raised me,

and for Seung-Jin who completed me.

Acknowledgements

I am forever indebted to the Block Research Group, which is truly greater than the sum of its diverse and talented individuals. The camaraderie, respect and support that every member of the group has for one another were paramount to the completion of this dissertation. I sincerely thank the current and former members of the group who accompanied me through this journey from close and afar. I will cherish the friendships I have made within the group for the rest of my life.

I am tremendously thankful to the two leaders of the Block Research Group, Prof. Dr. Philippe Block and Dr. Tom Van Mele. This dissertation would not have been possible without my advisor Prof. Block and his relentless enthusiasm, creative vision and inspiring mentorship. I thank Dr. Van Mele for giving all of us the gift of COMPAS, his constant technical support and thorough reviews of my writing and presentations. They both have taught me so much on how to become an independent researcher and pursue a career in academia.

I am thankful to the three then-postdoctoral researchers of the Block Research Group, Dr. Matthias Rippmann, Dr. Tomás Méndez Echenagucia and Dr. Andrew Liew. I am deeply grateful for their constant support, enthusiasm, wisdom, humour, advice and patience with my countless questions about anything and everything, to which one of them always seemed to have an answer for.

I would also like to thank the PhD researchers of the Block Research Group, especially those who have been with me from the very beginning: David López López, Mariana Popescu, Cristián Calvo Barentin, Shajay Bhooshan and Robin Oval.

My special gratitude goes to Dr. Noelle Paulson for helping me with everything related to administration, German translations and processing endless receipts.

Furthermore, I would like to thank my co-advisors, Mr. William (Bill) F. Baker and Prof. Allan McRobie. Their expert knowledge and experience were profoundly resourceful during the development of the research, and their insightful feedback were greatly helpful during the final stages of the dissertation.

Finally, I thank my family, especially my dear wife Seung-Jin, for their unconditional love and support.

Contents

Abstract

This dissertation presents a computational framework for structural design applications based on 3D graphic statics using polyhedral force diagrams. At the core of this framework is the development of a generalised theoretical foundation for 3D graphic statics, underpinned by the formulation of three datastructures for addressing a wide range of equilibrium problems involving spatial systems of forces: a polyhedral cell, the multi-cell polyhedron, and generalised cell networks. The three datastructures along with the relevant operations and geometric algorithms are implemented in an open-source library with interactive user interfaces to common computational design environments targeting both architects and structural engineers.

The framework is developed with the goal of maximising the inherent benefits of computational graphic statics, which diminish with increasing complexity of spatial structures: legible visualisation of force equilibrium, intuitive designer interactivity in real time, and provision of new structural design insights. In contrast to conventional numerical approaches to structural design that are dependent on predefined forms before any analysis can be executed, the presented framework enables new design methodologies that explore the geometry of forces as the catalyst for design, analysis, and refinement of spatial structures.

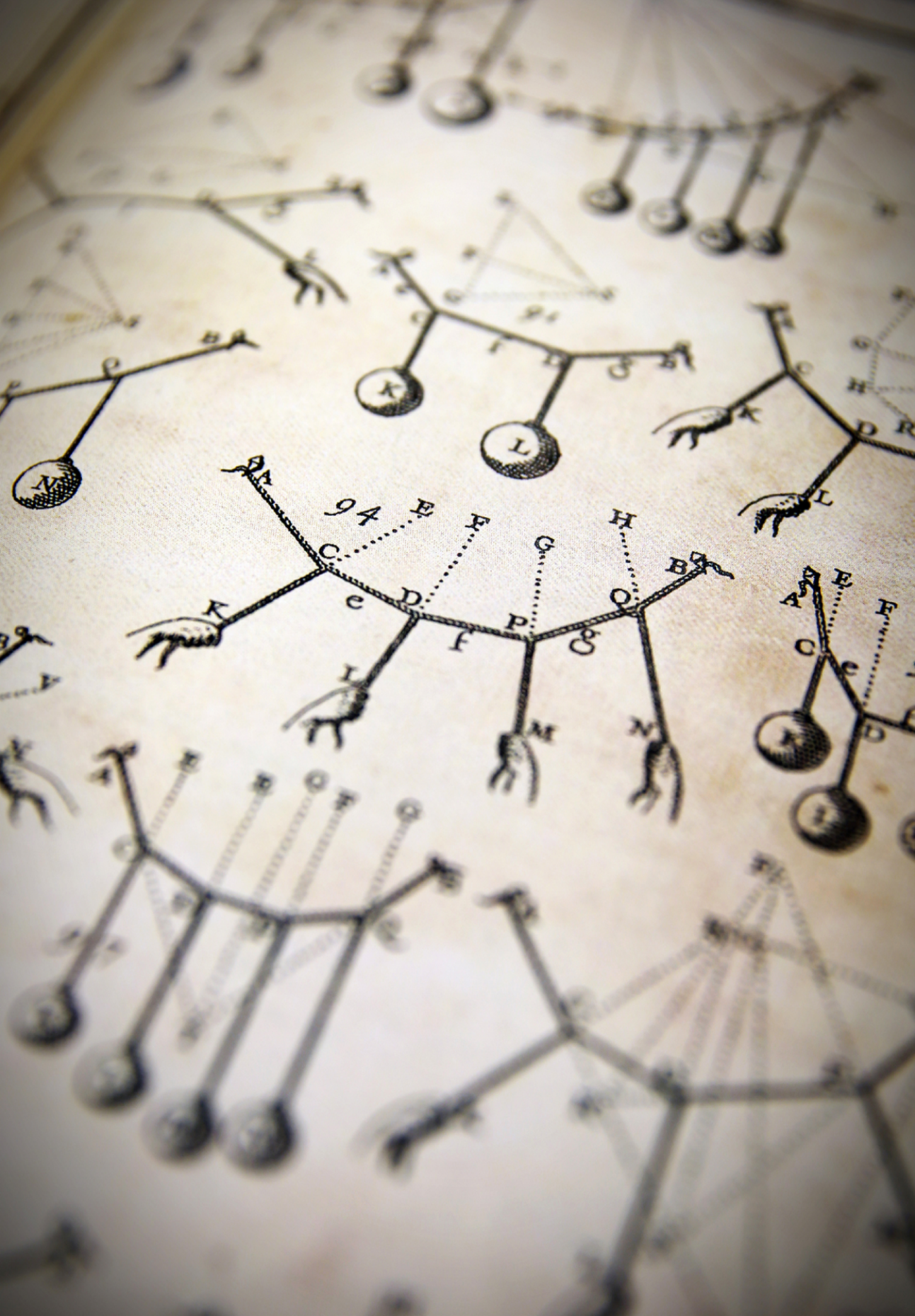
These new potentials are illustrated through numerous examples, showcasing how the framework could be used to address structural design problems in a variety of new ways that are simply not possible with existing, form-dependent tools. Beyond equilibrium design and analysis, the practical relevance of this research in architectural fabrication and construction is demonstrated through the MycoTree project, a 1-to-1 built prototype of a spatially branching structure made of load-bearing mycelium components.

Zusammenfassung

Die vorliegende Dissertation stellt rechnergestützte Methoden und Grundlagen für den Entwurf von Tragwerken auf Basis der dreidimensionalen graphischen Statik unter Verwendung polyedrischer Kraftdiagramme vor. Im Mittelpunkt der Arbeit steht die Entwicklung einer verallgemeinerten, theoretischen Grundlage für die dreidimensionale graphische Statik, die durch die Formulierung von drei verschiedenen Datenstrukturen zur Bewältigung einer Vielzahl von Gleichgewichtsproblemen räumlicher Tragwerke gestützt wird. Die drei Datenstrukturen basieren auf einzelnen Polyedern, mehrzelligen, zusammengesetzten Polyedern und verallgemeinerten Polyedernetzwerken. Zusammen mit entsprechenden Operationen und geometrischen Algorithmen sind sie Teil einer Open-Source-Bibliothek, die mittels interaktiven Benutzeroberflächen für gängige CAD Anwendungen auf die Bedürfnisse von Architekten und Ingenieuren zugeschnitten ist.

Die Programmbibliothek wurde mit dem Ziel entwickelt, die inhärenten Vorteile der rechnergestützten grafischen Statik zu maximieren, die mit zunehmender Komplexität der räumlichen Strukturen abnehmen. Hierzu gehört die Lesbarkeit der Visualisierung des Kräftegleichgewichts, eine intuitive Entwurfsumgebung mit Echtzeitinteraktivität und der Erkenntnisgewinn beim Entwerfen von Tragwerken. Im Gegensatz zu herkömmlichen, numerischen Ansätzen zur Entwicklung von Tragsystemen, die vor jeder Analyse eine vordefinierte Form benötigen, ermöglicht der vorgestellte Ansatz neue Entwurfsmethoden, für welche die geometrische Lesbarkeit der Kräfte als Katalysator für den Entwurf, die Analyse und die Optimierung räumlicher Strukturen dient.

Diese neuen Potenziale werden an zahlreichen Beispielen veranschaulicht und zeigen, wie mit der vorgestellten Herangehensweise tragstrukturelle Gestaltungsprobleme auf vielfältige und neue Weise angegangen werden können, die mit bestehenden Methoden nicht oder nur mit erheblichem Aufwand lösbar sind. Neben neuen Ansätzen zum Entwurf und zur Analyse von Tragwerken wird die praktische Anwendung der Forschungsarbeit am Beispiel der Herstellung und Konstruktion des MycoTree Projekts demonstriert. Der architektonische Prototyp im Massstab 1:1 besteht aus einer verzweigten Struktur bestehend aus tragenden Mycelelementen.


94
E
F
G
H
C
D
P
Q
K
L
M
N
B
e
f
g

Part I

Introduction

1 Research Statement

This chapter introduces the research topic, the motivation and the thesis statement. Also presented in this chapter is the summary of the original contributions that will be made by this dissertation. Finally, the importance and relevance of this research endeavour within the context of contemporary design culture in architecture and structural engineering are discussed.

1.1 Introduction

Graphic statics is a design and analysis method for two-dimensional (2D) discrete structures, that relies on geometrical rather than analytical or numerical representations of the relation between a structure's geometry and the equilibrium of its internal forces (Maxwell, 1864; Culmann, 1864; Cremona, 1890). Graphical representation of equilibrium using reciprocal form and force diagrams provides invaluable insight for designers and improves intuitive understanding of a structure's behaviour through a visual medium that is easier to digest and more transparent than conventional, arithmetic or numerical methods (Allen and Zalewski, 2009).

Graphic statics gradually disappeared from structural engineering practice over the 20^{th} century due to the advancement of computers and powerful numerical software. However, there has been a recent revival of graphic statics through computational implementations, which have enabled dynamic interaction between form and force diagrams with real-time, visual feedback (Van Mele et al., 2012). These interactive implementations of graphic statics allow users to directly control the geometry of forces, which enable force-driven design approaches with high formal freedom to explore structurally informed geometries during early stages of design.

More recently, graphic statics has been extended into the third dimension (3D) (Akbarzadeh et al., 2015b; McRobie, 2017a), where the equilibrium of spatial systems of forces can be represented by closed force polyhedrons (Rankine, 1864). It has also been shown that the reciprocal diagrams used in graphic statics, are planar projections or sections of polyhedral frames and its reciprocal force polyhedrons; 2D graphic statics is a special case of a more general 3D graphic statics (Crapo and Whiteley, 1993; Mitchell et al., 2016; McRobie and Williams, 2018). Therefore, 3D graphic statics not only provides the possibility to model and analyse equilibrium of spatial structures in an interactive manner, but also offers a profoundly new perspective and approach to 2D problems and applications.

One of the the most unique properties of computational graphic statics is that the form of the structure can be modified or controlled through the geometry of the force diagrams. Whereas most conventional structural design or analysis software are dependent on predefined forms, computational graphic statics provides drastically different ways of approaching structural design problems by using the geometry of the force diagrams as the catalyst for design, analysis and optimisation. Exploration of structural forms by constraining, optimising, manipulating and designing the geometry of the force diagrams has the potential to significantly broaden the design space. Computational implementations of graphic statics through interactive platforms enable uninhibited exploration of these new force-driven design spaces, and discovery of unforeseen structural possibilities freed from any institutionalised biases or prejudices.

1.2 Thesis statement

1.2.1 Open design problems in 3D graphic statics

This dissertation presents a computational framework for structural design applications based on 3D graphic statics using polyhedral force diagrams. It identifies key open problems in 3D graphic statics and addresses them from the perspective of a designer in practice, with particular focus on maximising applicability and usability of the mathematical theories behind the reciprocal relationship of form and force diagrams in three dimensions. The research objectives are defined based on these open problems, and they are are contextualised in a proposed design workflow using 3D graphic statics (Figure 1.1): 1) addressing realistic boundary conditions; 2) discovering new methods of generating spatial topologies; 3) exploring non-polyhedral forms; and 4) materialising spatial structures generated with 3D graphic statics. The ultimate goal of this research is to explore and understand the pragmatic design potential of 3D graphic statics in real-world applications.

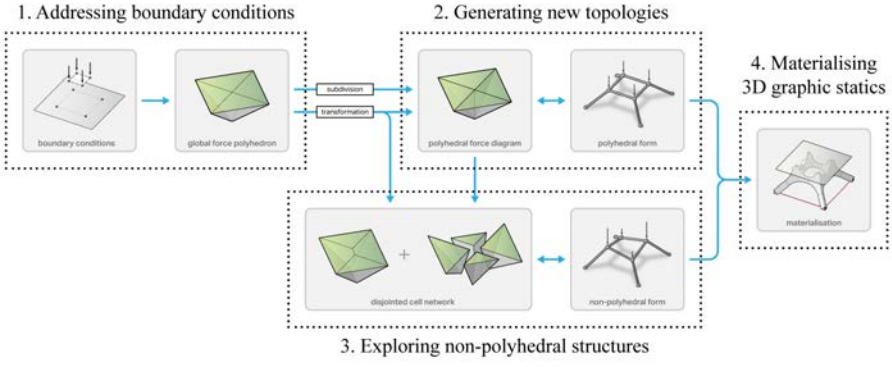


Figure 1.1: Proposed design workflow using 3D graphic statics and the four research objectives.

1.2.2 Lack of computational framework for 3D graphic statics

Despite its benefits for both academic research and professional architecture and structural engineering practices, there currently exists no general computational design framework for 3D graphic statics. New publications of graphic statics are often stand-alone implementations with their own set of conventions, computational languages and software dependencies, which make them incompatible or unusable by other researchers. New knowledge is often shared through papers in physical or digital formats that describe the computational implementations with text and static images without actually delivering any computational means to the readers. A new researcher who is interested in 3D graphic statics must start his or her implementation from scratch, unless he or she joins a research group with an established computational framework. The lack of a unifying platform or computational environment for graphic statics makes it difficult to start new strands of research or continue existing ones.

This dissertation seeks to address this issue by developing a generalised computational framework for 3D graphic statics based on polyhedral reciprocal diagrams. First, a theoretical foundation will be developed with different datastructures geared towards various types of problems involving the equilibrium of spatial systems of forces. Then, a wide range of operations

Figure 1.2: Exchange of knowledge and implementations of computational graphic statics remain largely based on text-based paper formats, with no unifying computational framework that is openly available to the research community.

and algorithms will be developed that enable these datastructures to be used in a computational design environment. Integration of the framework with CAD software through interactive user interfaces enables exploration of new structural design applications using 3D graphic statics. Most importantly, the presented framework will be made available to the public through an open-source library, which is intended to be an evolving repository and platform for future research in computational geometry and graphic statics.

1.3 Originality

This dissertation makes original and unique contributions to the field of graphic statics in the following ways:

- **New datastructures for 3D graphic statics**

  Three datastructures are developed for addressing a wide range of equilibrium problems involving spatial systems of forces: a polyhedral cell, the multi-cell polyhedron, and general cell networks.

- **Generalisation of polyhedral force diagrams**

  The cell network datastructure is introduced as a generalisation of polyhedral reciprocal diagrams. While a polyhedral force diagram represents the static equilibrium of a form diagram that is also polyhedral in its geometry, a generalised cell network can represent the static equilibrium of either polyhedral and non-polyhedral spatial trusses, which are nodally loaded and comprised of straight bars. Cell networks expand the typologies of structures that can be explored with 3D graphic statics, and enable more explicit force-driven design investigations.

- **Iterative geometric approach**

  The operations and algorithms designed for these datastructures are developed with iterative geometric solvers. The more traditional, procedural construction of form and force diagrams is useful for teaching and understanding the principles of graphic statics. However, they are not ideal for interactive design environments where the reciprocal diagrams may require constant updates and repeated reconstructions. Especially for spatial structures, an algorithmic non-procedural approach to the construction, transformation and visualisation of polyhedral force diagrams is necessary.

- **Improving the visualisation of reciprocal diagrams**

  Techniques for enhancing the visual representation and legibility of polyhedral reciprocal diagrams are investigated. While the focus of graphic statics research is typically on the geometric properties of the diagrams, the user's ability to visually decipher, understand

and manipulate the complex polyhedral reciprocal diagram is critical in improving the overall usability of 3D graphic statics in design applications.

- **Linking graphic statics to fabrication**

  This dissertation investigates how the inherent geometric properties of polyhedral reciprocal diagrams can be exploited for the materialisation and fabrication of spatial structures. It is shown that complex spatial structures can be fabricated and assembled using flat sheet materials that are cut with relatively inexpensive 2-axis CNC processes.

- **Open-source library**

  The presented computational framework is implemented in an open-source library that can be freely downloaded by the public, making computational graphic statics available to all. Every line of code is accessible, which is necessary for understanding the implemented concepts, collaboration with other researchers and customisation of the computational methodologies. The open nature of the framework is intended to encourage users to constantly test and make contributions that improve upon the preceding releases of the framework, and thereby collectively help it grow and evolve.

1.4 Relevance

This section discusses the importance and relevance of this research, and research in graphic statics in general, within the context of contemporary design culture and practice.

1.4.1 Advent of research in graphic statics

In recent years, there has been a rise of interest and research within the field of graphic statics. This is mainly due to the new design and research possibilities that arise when graphic statics is combined with advanced parametric and computer-aided design (CAD) software, which are readily available today. The three-dimensional modelling capabilities of most CAD software used in architectural design allows structural design explorations using 3D graphic statics based on polyhedral reciprocal diagrams, which was challenging with 2D drafting tools or software.

Over the past three decades, the annual number of publications written in English on graphic statics have increased at an almost exponential rate (Figure 1.3). Although the origins of graphic statics date back to the 18th century, computational graphic statics and 3D graphic statics in particular, are new emerging areas of research with countless design and research opportunities yet to be discovered.

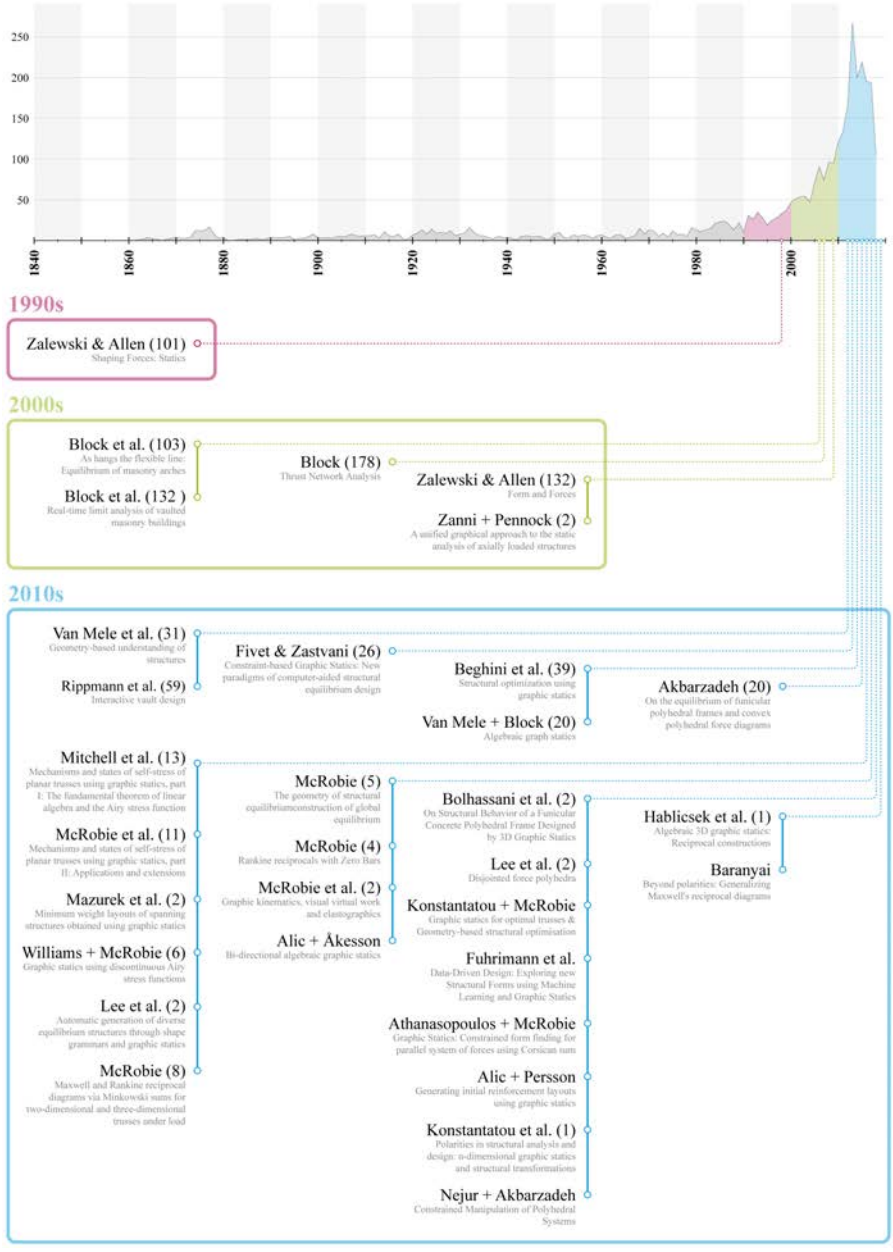


Figure 1.3: History of graphic statics related publications in English by year, where "graphic statics" or "graphical statics" is included in either the title or within the document. Some of the notable publications on Goolge Scholar (2018) that are relevant to the content of this dissertation from the past three decades are highlighted.

1.4.2 Towards a new design paradigm

Research in graphic statics continues to be relevant and important in today's "design-by-analysis" paradigm. While powerful finite element (FE) software can analyse almost any geometry and output significant amount of data, colour gradients and pages of numerical feedback do not provide nearly as much immediate insight compared to graphic statics through visible, geometric relationship between form and forces. More importantly, the visualisation of the structure's internal forces is also the mechanism for the user to interact with both the form and the forces of the structure, which is uniquely exclusive to graphic statics. The relevance of further research in graphic statics lies in describing increasingly more complex things in simpler ways, as opposed to an accumulation of simple procedures to describe complex things. New computational tools should distil and communicate large amounts of complex information in a meaningful and insightful manner, making designers smarter and better decision-makers.

1.4.3 Discovery of new structural forms

In addition to the intuitive, geometric representation of form and forces, there is relevance in investigating another powerful design potential of graphic statics: discovering new structures and design strategies through manipulations of the geometry of force diagrams. Force-driven structural design allows designers to discover new structural typologies and design possibilities that are not only spatially complex and always constrained to be in static equilibrium, but also freed from any previous biases towards known geometries or typologies. In order to overcome the predefined typologies, the computational logic of the tools that allow designers to generate and analyse such structures need to be reassessed, as pointed out by Laurent Ney (Adriaenssens et al., 2010, p. 41):

> "A choice is made from an extensive structural vocabulary built up over the last two centuries. ... A typology has a name, and the form and the relationship between the elements is described. The advantage of this is that it is easy to talk about structure, but the disadvantage is that how the structure looks is predetermined. ... the vocabulary freezes the object, and the objects thus frozen assume a sort of inviolable legitimacy. In order to arrive at new forms and concepts we have to free ourselves from such pre-defined typologies."

Unbiased exploration of unconventional structures are challenging to implement with conventional software and workflow, where an FE analysis software requires an input shape before it can run any analysis. Relevance lies in being able to discover and explore new structures while providing just as much control and freedom as with known typologies.

1.4.4 Practical relevance to digital fabrication

The institutionalised separation of form (architecture), forces (structure), and material (fabrication/construction) has resulted in a geometry-driven contemporary design culture (Oxman, 2010). The development and logic of CAD software used by the architects and FE software used by the structural engineers are also based on this separation of form and forces. This gap between form and forces can be bridged through use of the built-in geometric properties of the reciprocal diagrams to graphically represent the relationship between the geometry of the form and forces of a structure. However, as it is often the case with free-form geometries that are computationally generated, materiality is typically conceptualised as a passive property assigned to a finalised geometry during later stages of design (Menges, 2012). The inherent geometric properties of polyhedral force diagrams have numerous benefits that can be exploited to optimise and simplify the fabrication process of complex spatial structures. As graphic statics provides a geometry-based understanding of structures, geometry-based materialisation is an important topic of research for fabricating, assembling and constructing computational design models more efficiently, economically and sustainably.

1.5 Thesis structure and outline

This section summarises the structure and organisation of the dissertation. The dissertation is divided into four main parts. The first part, titled "Introduction," presents the research topic, the motivation and the thesis statement. It also outlines the originality and the relevance of the presented research. The second part, "Framework," presents the theoretical background and the technical implementation details of the computational framework. With illustrations and annotated diagrams, various components of the framework are described in detail. In the third part, "Applications," research results are presented, demonstrating the new force-driven design potentials of 3D graphic statics enabled by the framework. Finally, in the fourth part, "Conclusions," concluding remarks and possible directions for future research are discussed. The following synopses of each of the chapters provide a detailed overview of the contents of this dissertation.

Part I: Introduction

Chapter 1: Research statements

This chapter introduces the topic, the motivation, the originality and the relevance of the research conducted in this dissertation. It provides a brief contextualisation and preview of the entire dissertation. A much more in-depth literature review and scope of work are presented in the following chapters.

Chapter 2: Literature review

This chapter provides an in-depth review of the current state of the art on the relevant topics and previous research. First, a brief summary of the evolution of graphic statics is provided. Then, the recent developments of computational graphic statics and 3D graphic statics are surveyed and reviewed. Relevant computational implementations for polyhedral geometries are also reviewed and analysed. Finally, an overview of existing methods for materialisation and fabrication of spatial structures are presented.

Chapter 3: Scope of work

Based on the literature review, this chapter presents the scope of work for this dissertation. The problem statements are clearly identified and the specific challenges that will be addressed, are described. Research objectives are then defined, outlining the detailed goals that will be pursued.

Part II: Framework

Chapter 4: Theoretical background

This chapter presents the theoretical background of the computational framework. The described concepts are developed in response to the shortcomings and limitations of 3D graphic statics outlined in Chapter 3. At the core of the presented theoretical background is the establishment of three datastructures that can be used to address different types of equilibrium problems using 3D graphic statics: a polyhedral cell, the multi-cell polyhedron, and general cell networks.

In the section "Polyhedral cell," a datastructure is presented that can be used to represent the equilibrium of a single node of a structure, or to describe the global equilibrium of the external forces acting on a structure. This section also highlights and addresses numerous unresolved issues in the current state of the art in order to construct a generalised computational framework that can sufficiently handle both convex and non-convex polyhedrons. Much of the new theoretical foundation is formulated using the Extended Gaussian Image (Horn, 1984), which is a spherical representation of a polyhedron that has numerous benefits with regards to the topological visualisation, understanding and transformations of the polyhedron.

In the section "Multi-cell polyhedron," a datastructure is presented for representing the equilibrium of a multi-node structure that is polyhedral in its geometry. The topological structure, the interpretation of the corresponding forces in the form diagram and the cellular hierarchy of multi-cell polyhedrons are explained in detail.

In the section "Cell network," a datastructure is presented for representing the equilibrium of nodally-loaded spatial trusses comprised of straight bars.

A cell network is the generalisation of all polyhedral force diagrams. The theoretical background for cell networks is developed and its relevance to 3D graphic statics is discussed. A cell network is a hybrid of multiple datastructures, and the organisation of the different layers of datastructures is explained in detail.

Chapter 5: compas_3gs

This chapter presents **`compas_3gs`**, a computational implementation of the concepts presented in the previous chapter. This chapter provides an overview of the general approach and the organisational structure of the library. The main datastructures of the COMPAS framework (Van Mele et al., 2017) are introduced, and how they are used to represent the 3D form and force diagrams are explained. Some of the key algorithms of the **`compas_3gs`** are also presented. Finally, in the section "CAD integration," strategies for the integration of the presented framework with a CAD software is presented. Although the framework is not dependent on any specific CAD software, Rhinoceros (Robert Mcneel & Associates, 1993) was used as the "canvas" and the user interface for this dissertation. As such, various visualisation and built-in interactive functions of Rhinoceros that were used for the implementation are described.

Part III: Applications

Chapter 6: Addressing boundary conditions

This chapter demonstrates how the Extended Gaussian Image and arearisation algorithm presented in Chapters 4 and 5, respectively, can be used for constructing global force polyhedrons. It is shown how the magnitudes and orientations of the external forces can be constrained during the construction process to address various boundary condition scenarios.

Chapter 7: Generating new topologies

This chapter presents form-finding and shape explorations through geometric transformations and modifications of polyhedral cells and multi-cell polyhedrons. Utilising various geometric transformation functionalities of **`compas_3gs`**, cells of the polyhedral force diagrams can be repeatedly manipulated to generate, modify and refine the geometry of the form diagram. The workflow and the computational setup of this application are presented and described in detail.

The design examples in this chapter will demonstrate how the presented method can be used to discover new three-dimensional structural topologies without the biases towards conventional solutions. In addition, the methodology presented in this chapter provide an alternative design strategy for two-dimensional problems using prismatic polyhedral cells. This chapter

concludes by identifying directions for further research, and how the generative aspect of this method can potentially be combined with other data-driven design and machine learning techniques.

Chapter 8: Exploring non-polyhedral structures

This chapter presents how cell networks can be used to explore equilibrated spatial structures that are not polyhedral in its geometry. Additionally, cell networks enable a more force-driven design applications using 3D graphic statics through more precise control of the face areas of individual polyhedral cells. Whereas the shape explorations focused more on the control and manipulations of the geometry of the polyhedral force diagram, this chapter demonstrates how more quantitative, force-driven constraints can be incorporated during the design process.

The examples presented in this chapter showcase new structural typologies that are not realisable with previous implementations of polyhedral 3D graphic statics. The concept behind the workflow and the computational setup of this application is explained in detail with flowcharts and illustrations. Finally, this chapter concludes by discussing potential future research and applications particularly within the context of structural engineering.

Chapter 9: Materialising 3D graphic statics

This chapter presents fabrication-related applications of 3D graphic statics. These applications are demonstrated through the form finding and fabrication design of the MycoTree project, exhibited at the 2017 Seoul Biennale of Architecture and Urbanism. In addition to the general form finding and equilibrium analysis of the structural geometry, this chapter demonstrates how the polyhedral functionalities and transformations can facilitate in formulating logical and feasible fabrication and construction pipelines. The design methodology, form-finding process and the development of the fabrication geometry are all presented in detail with diagrams and photographs. This chapter concludes by discussing key issues that have room for improvements in future research.

Part IV: Conclusions

Chapter 10: Conclusion

The final chapter summarises the key contributions of this dissertation and discusses how the research objectives were met. The relevance and potential impact of the contributions within the fields of architecture, structural engineering, digital fabrication and computational geometry are briefly discussed. Finally, the limitations of the presented framework and methods are explained, and directions for future work are discussed. This chapter concludes by stating the final reflections.

2 Literature Review

This chapter provides an in-depth review of the relevant literature and previous work. First, a brief overview of the evolution of graphic statics is presented. The causes of its rise and decline over the course of the 20th century are discussed. The recent revival and research developments of graphic statics using computational tools is reviewed. Ranging from algebraic formulations of graphic statics, structural optimisation using reciprocal diagrams to 3D graphic statics, the latest state of the art in computational graphic statics is reviewed and discussed. Additionally, existing force-driven design methods, computational tools for polyhedral geometries and methods for materialising spatial structures are surveyed. This chapter concludes by providing a summary of the literature review and identifying key research gaps which will be used to define the problem statements and research objectives in the next chapter.

2.1 Graphic statics

This section provides a brief history of graphic statics. A thorough historical summary of graphic statics can be found in *The History of the Theory of Structures* by Karl-Eugen Kurrer (2008) and *Symmetrie Gruppe Dualität* by Erhard Scholz (1989).

Origins of graphical analysis of forces dates back to as early as the 16th century. Simon Stevin (1586) graphically showed the balance of forces on inclined planes using diagrams of weighted ropes. Pierre Varignon's *Nouvelle Mécanique ou Statique* (1725) is one of the first evidence of the use of funicular polygons or polygons of forces to describe the static equilibrium of internal forces of simple structures and systems of tensioned ropes (Figure 2.1).

By early 19th century, the concepts of funicular polygons are used by mathematicians, scientists and engineers to visualise, analyse and understand the equilibrium of a system of forces (Poncelet, 1822; Lamé and Clapeyron, 1828). William John Macquorn Rankine (1858) demonstrated how funicular polygons can be used to compute the internal forces of statically determinate trusses. James Clerk Maxwell's publications (1864; 1870) established the theory of the reciprocal relationship between the form and force diagrams of structures. Graphical methods of analysing static equilibrium was formally introduced to the field of structural engineering as "graphic statics," through Karl Culmann's seminal book, *Die Graphische Statik* (1864). Based on Culmman's graphical methods and Maxwell's theory of reciprocal diagrams, Luigi Cremona introduced a procedural method

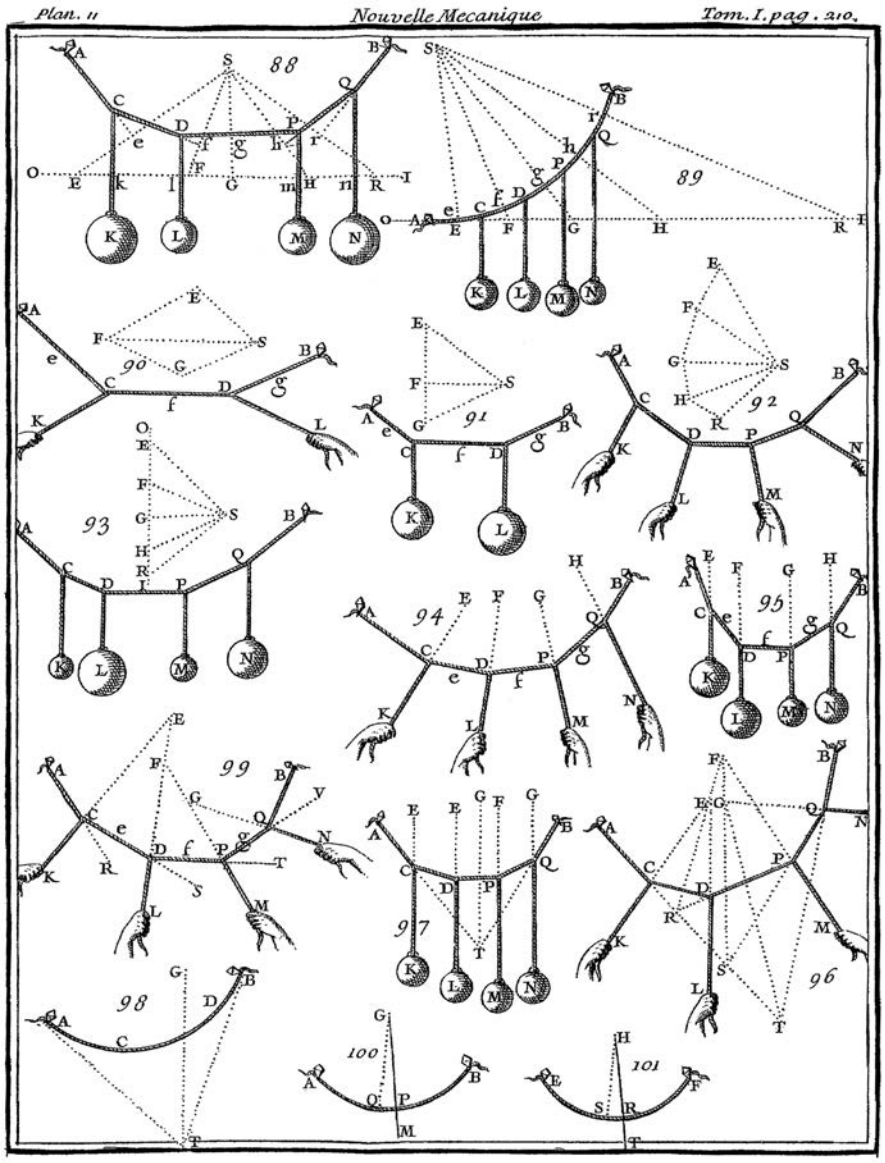


Figure 2.1: Drawings of funicular polygons in *Nouvelle Mécanique ou Statique* (Varignon, 1725) that graphically represent the static equilibrium of a system of ropes, which are assumed to be not extensible, under various loading conditions.

for constructing reciprocal diagrams (1872). It is not until Thomas Hudson Beare's English translation of Cremona's book (1890) that graphic statics spreads to English-speaking markets.

Figure 2.2: Notable cast-iron structure built in the late 19th century: a) Royal Albert Bridge by Isambard Kingdom Brunel in 1859; b) Garabit Viaduct by Gustav Eiffel and Maurice Koechlin in 1884; c) The Eiffel Tower by Eiffel and his apprentices, Maurice Koechlin and Emile Nouguier in 1887; and d) Forth Bridge by Sir John Fowler and Sir Benjamin Baker in 1890. (Image credit: a) Norman Lockett; b) rochagneux – Fotolia; c) Wiki Commons; and d) Ray Devlin)

During the late 19th century, the rise of the popularity of graphic statics was due to the fact that it was the perfect, complementary analysis technique for emerging discrete and pin-jointed cast-iron structures (Kurrer, 2008). Compared to more traditional stone and timber structures, pin-jointd cast-iron structures are made of a series of linear elements that carry axial forces only. The rapid change in the use of building materials, especially in bridge construction, introduced drastically new structural typologies such as trusses, suspended structures and lattice systems (Ewert, 2002). Because the equilibrium of force can be graphically resolved, these new complex structural forms could be designed and analysed without the need of complicated numerical methods or tedious manual calculations. Notable structural engineers of the 19th century are known to have used graphic statics to varying degrees, to design and analyse some of the world's most recognisable structures that are still standing today (Figure 2.2).

At the turn of the 20th century, the building industry witnessed a growth in reinforced concrete construction. Graphic statics, which is appropriate for structures with linear members carrying axial forces only, was no longer adequate for analysing structures made of solid and indeterminate materials like concrete. As a result, the methods of designing and analysing structures gradually became more dependent on numerical methods based on linear algebra. Additionally, the increasing computational power of computers gave birth to FE modelling and analysis, which was much faster and efficient than manually drafting form and force diagrams by hand on paper.

The emergence and decline of graphic statics in the field of structural engineering is clearly reflected by the rise and fall of the use of the phrase "graphic statics" in literature over the course of the 20th century. Figure 2.3 shows Ngram Viewer (Schmidt and Heckendorf, 2017) results for: "graphische Statik" (Figure 2.3-a); and "graphic statics" and "graphical

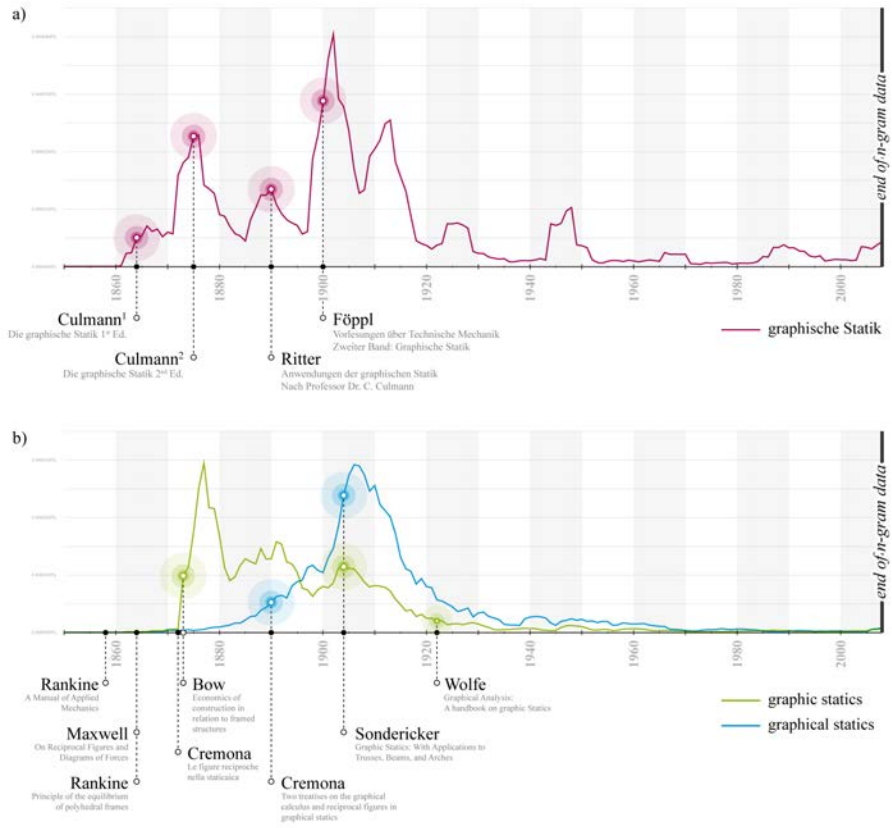


Figure 2.3: Google Ngram Viewer (Schmidt and Heckendorf, 2017) results recreated by the author for: a) "graphische Statik"; and b) "graphic statics" and "graphical statics."

statics" (Figure 2.3-b). These graphs are showing yearly count of these phrases in printed sources between 1850 and 2008. As evident in these graphs, the use of these phrases peaked around the years of the notable publications mentioned previously in this section.

In German, the term "graphische Statik" was coined by Culmann in his 1st edition of *Die Graphische Statik* (1864). Expectedly, there is a steady increase of n-grams around this time, with a sharp increase just before the publication of the 2nd edition of *Die Graphische Statik* in 1875. In English, the initial spike can perhaps be attributed to Robert H. Bow's *Economics of Construction in Relation to Framed Structures* (1873), which was one of the first notable English publications that applied graphic statics with regards to efficiency of structures. As graphic statics travelled across the Atlantic Ocean into the American market, numerous publications including Jerome Sondericker's *Graphic Statics with Applications to Trusses, Beams and Arches* (1903) popularised graphic statics in the United States. In both German and

English, there is a drastic decline at the turn of the 20th century, and is almost non-existent over the latter half of the century and into the 2000s. Karl-Eugen Kurrer writes in *The History of the Theory of Structures*, that with the decline of graphic statics (2008, p. 330):

> "... [the structural engineering community] lost an extremely compact form of rendering visually the play of forces in the analysis and synthesis of load-bearing systems in the conception and design activities of the civil and structural engineer."

2.2 Computational graphic statics

This section gives an overview of research and previous work related to computational applications of graphic statics, and force-driven structural design methodologies.

2.2.1 Interactive Graphic Statics Tools

Recent revival of graphic statics is mainly due to its amplified potential when combined with modern computational design tools. When drawn by hand, the construction of reciprocal diagrams is a tedious process, and freely manipulating the diagrams after they have been drawn is not possible. With computers, however, the drawing process can be automated and users can manipulate the structure in real time by modifying the force diagrams, or vice versa. Inherent parametric qualities of graphic statics can be exploited with CAD tools to exploit its new design potentials. The merits of computational graphic statics for the conceptual design of structures are gradually becoming recognised in the industry.

The power of computational graphic statics has been demonstrated through the development of applications such as ActiveStatics (Greenwold and Allen, 2003), InteractiveTHRUST (Block, 2005) and eQuilibrium (Block Research Group, 2010). The interactive graphic statics drawings of eQuilibrium are constructed using GeoGebra (Hohenwarter et al., 2002), an interactive geometry and mathematics application. Alternatively, parametric CAD software can also be used to construct interactive drawings. While interactive graphic statics drawings provide real-time interaction and visualisation, they require tedious pre-programming and need to be drawn by a user with previous knowledge of graphic statics. Any major design modifications, such as change in topology, to a pre-made drawing are difficult to implement, and exploring new shapes require a complete reconstruction of the entire interactive drawing.

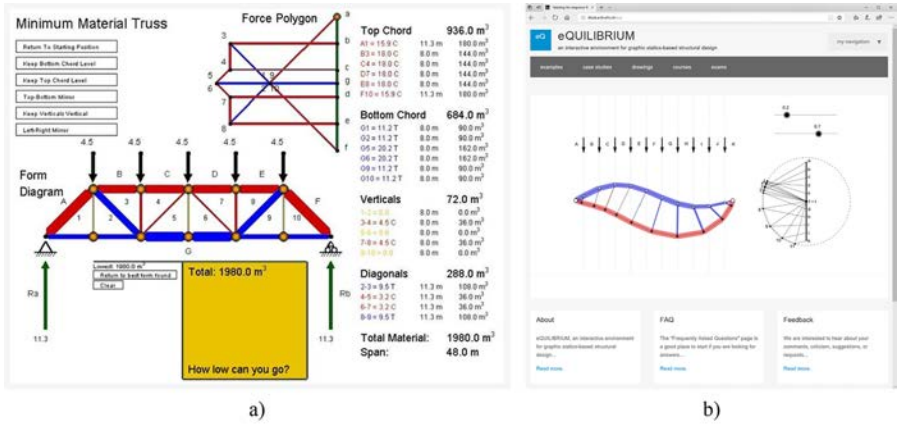


Figure 2.4: a) ActiveStatics developed by Greenwold and Allen (2003), a web implementation of graphic statics for preset structural typologies, where the coordinates of the form diagram can be interactively changed in real time based on techniques described in Zalewski and Allen (1998); and b) eQuilibrium, a website developed by the Block Research Group (2010), on which the relationship between structural forms and forces are taught using interactive graphic statics drawings.

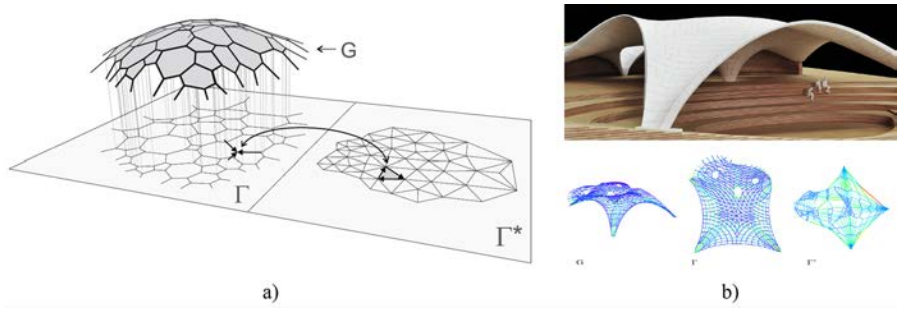


Figure 2.5: a) Thrust Network Analysis (Block and Ochsendorf, 2007); and b) RhinoVAULT, a funicular shell design plug-in for Rhinoceros (Rippmann et al., 2012).

Tools like RhinoVAULT (Rippmann et al., 2012), a plug-in for Rhinoceros based on Thrust Network Analysis (TNA) (Block and Ochsendorf, 2007), is an interactive tool for form finding of free-form shell structures using reciprocal force diagrams. RhinoVAULT extended graphic statics to "2.5" dimensions, by using horizontal projections of discretised shell structures to construct interactive 2D form and force diagrams on a plane. While it provides a significant amount of control of both form and forces during early stages of design, the loading conditions can only be vertical and the plug-in is ultimately limited to a very specific structural typology: funicular surface structures.

2.2.2 Algebraic Graph Statics

Algebraic Graph Statics (AGS) introduced a non-procedural approach to 2D graphic statics, by algebraically formulating the geometrical relationship between the form and force diagrams (Van Mele and Block, 2014). AGS not only automated the drawing process for complicated 2D structures, but it also illustrated a clear algebraic relationship between graphic statics and more conventional equilibrium equations used in matrix analysis. AGS also demonstrated that the construction of reciprocal diagrams used in graphic statics is possible only for structures that satisfy a specific topological condition; the form diagram must be a graph that is re-arrangeable without crossing edges, from which a a topological dual graph can be constructed, and subsequently the reciprocal force diagram.

Bi-directional AGS extended AGS such that interactive manipulations of both the form or the force diagrams are possible (Alic and Åkesson, 2017). It demonstrated how various geometric constraints can be imposed on the force diagram to enable various force-driven design explorations. Due to the algebraic construction of the reciprocal diagrams, any manipulation of either the form or the force diagram results in a real-time update of the other.

In addition to 2D graphic statics, the algebraic formulation for constructing polyhedral reciprocal diagrams used in 3D graphic statics has been introduced by Hablicsek et al. (2019). These implementations of graphic statics have demonstrated how the computational formulations of AGS are equivalent to the equilibrium equations used in matrix analysis of planar, pin-jointed trusses (Van Mele and Block, 2014). AGS also provides a robust back-end for a real-time, interactive and flexible computational implementation of graphic statics.

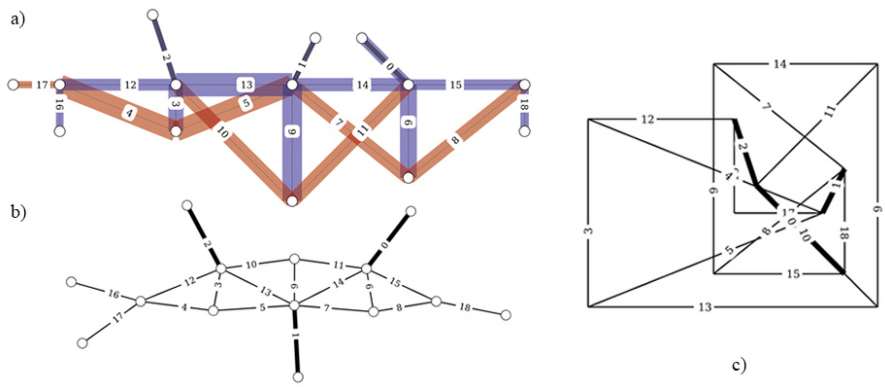


Figure 2.6: Algebraic Graph Statics, a method for non-procedural construction of force diagrams (Van Mele and Block, 2014): a) the graph interpretation of the form diagram of a fink truss with crossing edges; b) planar re-drawing of the form graph; and c) the reciprocal force graph.

2.3 3D graphic statics

The three-dimensional modelling capabilities of most CAD software available today have enabled structural design explorations using computational graphic statics in three dimensions. This section provides a brief overview of the theoretical background behind 3D graphic statics, and a review of previous work and the latest developments in the field.

2.3.1 From 2D to 3D

Although graphic statics is widely known as a graphical method of analysis for 2D structures, the reciprocal form and force diagrams on a plane is based on three-dimensional relationships. Maxwell's 1864 and 1870 papers provided the foundation of graphic statics and principles of reciprocal diagrams. In the 1864 paper, *On reciprocal figures and diagrams of forces*, Maxwell observed that a 2D form diagram is the projection of a 3D plane-faced polyhedron, and the projection of its reciprocal polyhedron is the 2D force diagram. Within the context of structural engineering, a "plane-faced polyhedron" can be interpreted as a linear or continuous Airy stress function, as clarified by Mitchell et al. (2016) and McRobie et al. (2016). As a corollary, any n-dimensional reciprocal diagrams are projections of $(n+1)$-dimensional Airy stress functions (Crapo and Whiteley, 1994; Konstantatou and McRobie, 2016). Using projective geometry and polarity, a generalised method for constructing 2D and 3D reciprocal diagrams through higher dimension stress functions was presented by Konstantatou et al. (2018). These contributions have provided important insights in establishing a generalised theoretical framework for reciprocal diagrams in any dimension. However, the examples of 3D structures in this context largely remain as abstract and small self-stressed objects with specific topologies predefined by the user.

2.3.2 Vectorial 3D graphic statics

The equilibrium of spatial forces can be resolved using weighted vectors (Föppl, 1892; Schrems and Kotnik, 2013). Vector-based resolution of spatial equilibrium is also referred to as "Cremona reciprocals" (Crapo, 1979) or "3D Cremona diagrams" (McRobie, 2016b; Konstantatou and McRobie, 2016). While the resolution of equilibrated force vectors at a single node is straightforward, assembling a "complete" force diagram for an entire, fully 3D structure is not possible without having duplicate or "non-overlapping" edges (Jasienski et al., 2016). The resulting force diagram is neither dual or reciprocal to the form diagram. Therefore, additional numerical solvers are required to enforce the parallel relationship between the form and the force diagram (D'Acunto et al., 2017), failing to preserve the intuitive and transparent aspects of graphic statics. The main argument for vector-based 3D graphic statics is that the method stays true to the original 2D graphic

statics techniques by using lengths of vectors to represent force magnitudes. However, if the weighted vectors are not oriented on the same plane, the foreshortening makes the dense clutter of lines in space challenging to discern in a quantitative manner.

2.3.3 Polyhedral 3D graphic statics

Another branch of 3D graphic statics is based on polyhedral representations of force equilibrium. Rankine generalised Maxwell's principles for 2D reciprocal diagrams to 3D reciprocal diagrams in his 1864 paper, *Principle of the Equilibrium of Polyhedral Frames*. As McRobie (2017a) points out, although both Maxwell and Rankine used the term "frame" in their seminal texts, they were referring to pin-jointed trusses carrying axial forces only. It is important to note that this differs from the modern use of the term "frame" in structural engineering, which typically refers to structures with rigid joints that may transmit moments. Since this dissertation is not based on vector-based 3D graphic statics, "3D graphic statics" will henceforth refer to 3D graphic statics based on polyhedral representation of equilibrium.

Rankine's seminal text on 3D graphic statics is one half-page long with only three short paragraphs and no figures. By combining the features and advantages of modern CAD software that were unavailable to Rankine, Akbarzadeh (Akbarzadeh et al., 2015b; Akbarzadeh, 2016) clarified Rankine's propositions graphically through three-dimensional diagrams and visualisations. In 2D graphic statics, the magnitude of the internal axial force of a member is represented by the length of the corresponding edge in the force diagram. In 3D graphic statics, the areas and orientations of the faces in the polyhedral force diagram represent the directions and the magnitudes of the forces in the corresponding members in the polyhedral form diagram. Polyhedral reciprocal diagrams are also referred to as "Rankine reciprocals" or "Rankine 3D diagrams" in the literature (McRobie, 2016a; Konstantatou and McRobie, 2016).

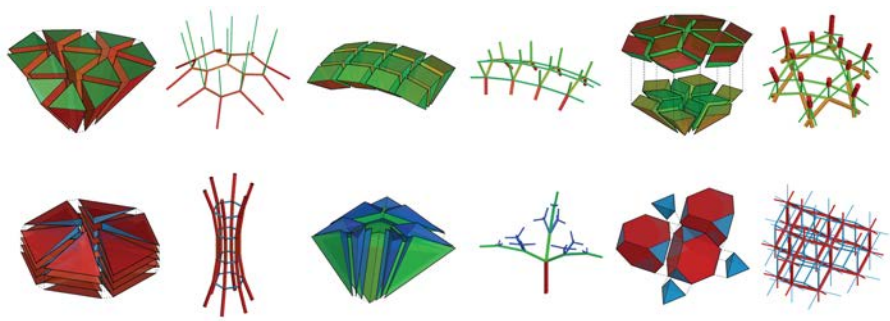


Figure 2.7: Convex polyhedral reciprocal diagrams for compression-only spatial structures (Akbarzadeh et al., 2015b).

The full reciprocal relationship between polyhedral form and force diagrams provides significant amount of control in modelling, designing and form finding of spatial structures. Over the past few years, there have been numerous design applications using 3D graphic statics, such as form-finding explorations through iterative subdivisions of force polyhedra (Akbarzadeh et al., 2015c; Ghomi et al., 2018), scale-model testing of compression-only spatial structures (Bolhassani et al., 2018) and a full-scale materialisation of a structure designed with 3D graphic statics (Bolhassani et al., 2018).

Beyond design and form-finding explorations, recent contributions from McRobie and Konstantatou have established a generalised theoretical foundation for 3D graphic statics by relating it to some of the most fundamental principles of structural engineering. Some of the notable contributions are: n-dimensional reciprocal construction methods using projective geometry (Konstantatou and McRobie, 2016; Konstantatou et al., 2018); Minkowski sum diagrams in relation to Maxwell's load path theorem (McRobie, 2016a); addressing incompleteness of Rankine reciprocals and the generalisation of 3D graphic statics for any three-dimensional frames (McRobie, 2016b, 2017a); incorporating shear and moment forces through discontinuous Airy stress functions (McRobie and Williams, 2018); and design explorations in relation to kinematics, virtual work and displacements (McRobie et al., 2017). As these contributions are primarily theoretical in nature, the examples presented are typically small, self-stressed structures and have not demonstrated their potential in an interactive design workflow. Work from Athanasopoulos have shown glimpses of how these principles can be applied to more realistic, larger-scale structures (Athanasopoulos and McRobie, 2017; Athanasopoulos et al., 2018). Nevertheless, these contributions provide the theoretical background and the insights necessary for developing new application concepts and ideas for 3D graphic statics.

Research in applied 3D graphic statics is still in its infancy, and has several important topics for further work. Above all, there is no openly available computational framework for 3D graphic statics. Without sufficient computational tools, it is challenging to explore or implement new 3D graphic statics applications in an interactive design environment. In addition, commercially available CAD software in architecture and structural engineering practices is not designed for handling polyhedral geometries with inherent planarity constraints and additional layers of data necessary for 3D graphic statics. In order to extend polyhedral force diagrams beyond just being a tool for visualisation or verification of spatial equilibrium, a computational environment that allows adequate modelling and manipulation of polyhedral geometries is necessary. Furthermore, the current design implementations of 3D graphic statics are focused on computing the corresponding reciprocal diagram from a given form or force diagram strictly from geometry point of view, without addressing any realistic boundary condition constraints or loading cases. Materialisation and fabrication of spatial form diagrams

also needs further investigation, to ensure that designs generated with 3D graphic statics result in practically meaningful and constructible structures.

Most importantly, with the growing complexity of spatial structures, the reciprocal diagrams become just as complex and visually cluttered. Maxwell noted in 1870 that "the mechanical interest of reciprocal figures in space rapidly diminishes with their complexity." In 2D graphic statics, the magnitudes of forces are represented by the lengths of lines in the force diagram, whereas the areas of polyhedral faces in 3D graphic statics represent the magnitudes of forces. Compared to lengths of lines, areas are more difficult for a human eye to perceive quantitative information (Cleveland and McGill, 1984; Mackinlay, 1986), and the inherent, intuitive benefits of graphic statics are lost (Figure 2.8-a). Even in 2D graphic statics, the force diagrams of complex structures are visually cluttered and difficult to read (Figure 2.8-b). For polyhedral reciprocal diagrams, the legibility is even more challenging with three-dimensional elements that are stacked and clustered on top of one another (Figure 2.8-c). In order to increase the legibility and usefulness of polyhedral reciprocal diagrams, the means of representation and user interface need significant improvements.

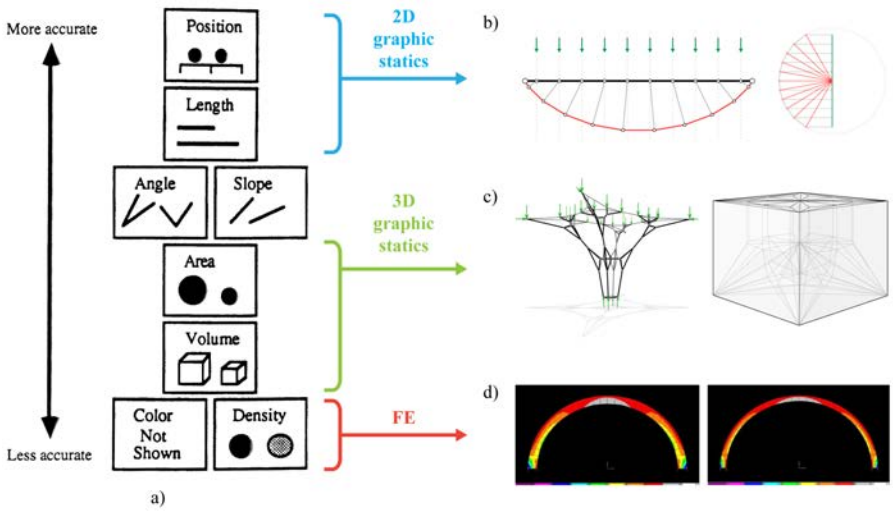


Figure 2.8: With increasing complexity of structures, the form and force diagrams can quickly become difficult to decipher: a) *Accuracy ranking of quantitative perceptual tasks* (Mackinlay, 1986), which shows that compared to positions and lengths, quantitative information that is represented through volumes and areas are more difficult for human eyes to perceive and understand; b) polygonal form and force diagram of 2D graphic statics (Van Mele et al., 2012), which use positions of points and lengths of lines to visualise force-related information; c) polyhedral form and force diagrams in 3D graphic statics (Akbarzadeh et al., 2015b), which use areas and volumes of polyhedrons to visualise force-related information; and d) FE analysis of arches (Block et al., 2006), which use colours to visualise force-related information.

2.4 Graphic statics and engineering applications

Graphic statics has also gained momentum within the structural engineering community due to some of its unique advantages over conventional numerical tools. Some of the latest developments and exemplary applications are reviewed in this section.

2.4.1 Graphic statics and structural optimisation

In addition to the interactive form-finding and design explorations, the parameterisation of the reciprocal relationship between from and force diagrams provides an alternative approach to conventional structural optimisation methods (Mazurek et al., 2011; Beghini et al., 2013). Unlike existing optimisation techniques, which typically manipulate the geometry of the structure (the form diagram) in search of optimum solutions, using graphic statics instead allows optimisation to be conducted using the geometry of forces (the force diagram), which offers several key benefits.

First, the total volume of the structure, a common performance metric used in structural optimisation (Hansen and Vanderplaats, 1990; Rahami et al., 2008), can be computed easily using the geometry of the reciprocal diagrams. To approximate the total volume of structures during conceptual design, several methods can be used: the total load path of the structure (Maxwell, 1870; Cox et al., 1965; Baker et al., 2013), energy density method (Baker, 1992), and by using morphological indicators (Samyn et al., 1999; Van Steirteghem et al., 2002). Total load path in particular can be seamlessly coupled with graphic statics; the total structural volume can be approximated by summing the products of the lengths of the members and the internal axial forces, which are both provided by the reciprocal diagrams. Second, because the existence of a force diagram entails an equilibrated form diagram, any manipulation of that force diagram will always result in an equilibrated structure. Third, fewer variables are needed to describe the connectivity of

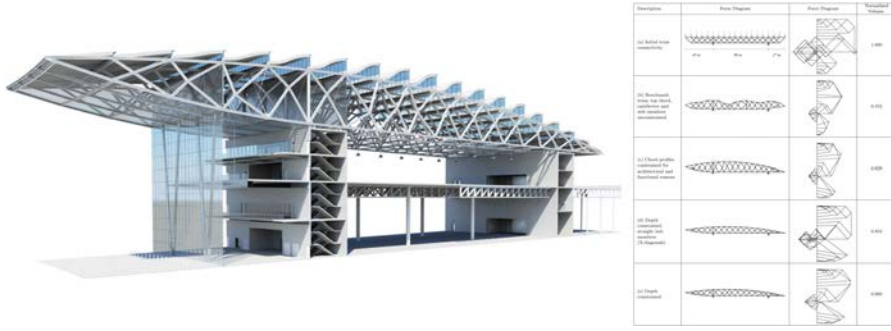


Figure 2.9: How graphic statics can be incorporated into a structural optimisation workflow to minimise the total load path or weight of the structure (Beghini et al., 2013).

an equilibrated structure with force diagrams, compared to the number of variables that would be required to describe the same structure using the geometry of the form. Therefore, the efficiency of optimisation processes can be significantly improved, and the range of optimisation problems that can be addressed is expanded (Beghini et al., 2013). In 3D, a geometry-based optimisation method for simple spatial trusses using polyhedral force diagrams was presented by Konstantatou and Mcrobie (2018).

However, work in this area remains mostly analytical and theoretical, with idealised design examples of highly specific typologies such as Michell trusses. Its application has been limited to specific, prescribed design scenarios and the methods as presented (Hartz et al., 2017), have not been generalised to address a wide range of problems and structural typologies.

2.4.2 Force-driven design

This subsection reviews some of the existing force-driven design methodologies, where the form finding process of the structural geometry is guided by force-driven constraints or objectives.

2.4.2.1 With numerical methods

With today's interactive digital modelling environments, various numerical form-finding techniques can be implemented to derive equilibrium geometries by controlling non-geometric properties: the force density method where the inner forces and lengths of the structural members are replaced by force-length ratios, called force densities (Linkwitz and Schek, 1971); the dynamic relaxation method where the axial stiffnesses of the members and the damping factors for each of the nodes are defined (Barnes, 1977); and the particle-spring method where each node is modelled as a spring with an assigned axial stiffness, initial length and a damping coefficient (Kilian and Ochsendorf, 2005).

The force density method in particular is a powerful method that allows the user to control and interact explicitly with both the geometry of the form and the magnitude of the internal forces. Lachauer and Block (2014) have shown how the force density method can be implemented in a CAD software to create an interactive design environment, where the user can modify the topology and geometry of the structure, or constrain the force densities of members and the coordinates of nodes. After each modification, a computational optimisation procedure is iteratively executed to update the geometry of the structure until an equilibrium solution is found that satisfies the newly defined geometric and force-driven constraints. Such interactive modelling methods are powerful in deriving equilibrium solutions for complex spatial topologies, while allowing the user to directly modify the geometry of the equilibrated form and impose both geometric and force-

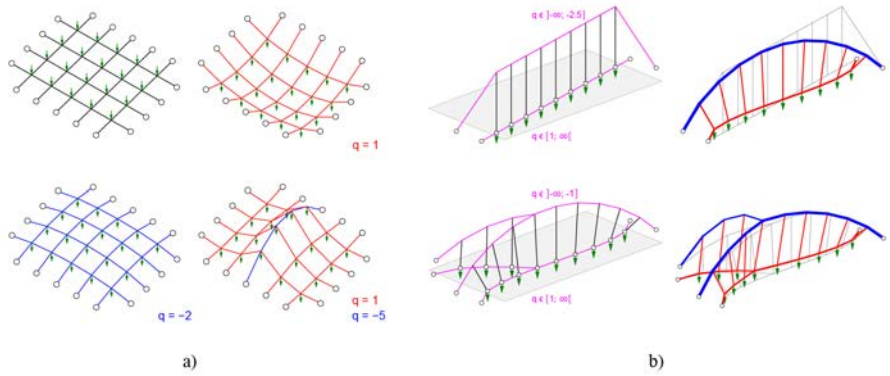


Figure 2.10: a) Various equilibrium solutions of a surface structure based on different distribution of target force density values (Lachauer, 2015); and b) equilibrium solutions for two examples of curved arch bridges, generated by setting two ranges of force densities for the compression members and tension members (Lachauer and Block, 2014).

driven constraints during the design process. However, the forces can be controlled only through manual assignment of specific numerical values or a range of values for the force densities. Therefore, manually assigning specific force density values or range of values for a large number of members can be tedious and time consuming.

2.4.2.2 With graphic statics

In graphic statics, the internal forces of equilibrium structures are represented graphically. Using a parametric modelling environment, the geometry of forces can be explicitly controlled and modified to explore various equilibrium solutions. Manipulating the geometry of the force diagram is a powerful way of exploring design variations. For example, the geometry of the force diagrams can be used for form finding of structures with specific force-driven objectives and constraints (Allen and Zalewski, 2009; Van Mele and Block, 2014; Lachauer and Kotnik, 2010) (Figure 2.11). The geometry of the force diagram can also be manipulated and optimised to search for optimal solutions (see Section 2.4.1). In addition, projected horizontal force diagrams can enable interactive design and exploration of free-form shell structures (Rippmann, 2016). However, these applications are based on modification of force diagrams, which means that a force diagram needs to exist before any transformations can take place. Therefore, a specific structural typology must be predefined by the user.

The geometry of the force diagrams can also be used to change and generate the topology of the form diagram, which is another unique feature of using the geometry of the force diagram for design applications. Given an initial force diagram, subdivision schemes can be applied to split polyhedral

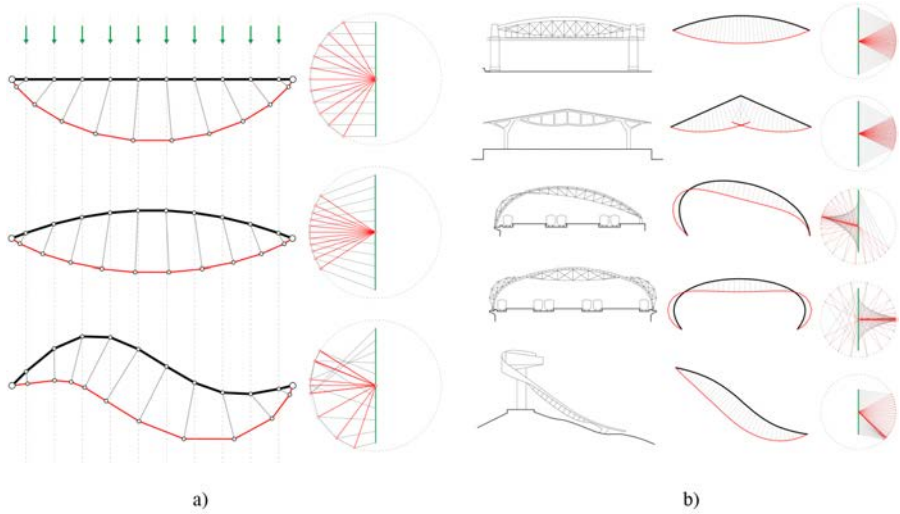


Figure 2.11: a) Implementation of the constant-force bottom chord principle using interactive drawings setup with eQuilibrium (Block Research Group, 2010); control over the shape of the top chord in combination with the force constraint on the bottom chord allows for an exploration of free-form constant force trusses (Van Mele et al., 2012); and b) interactive, parameter-driven explorations of the relation between form and forces demonstrate how the same structural principles can result in different architectural shapes (after Figure 5 of Lachauer and Kotnik (2010)).

cells into smaller cells, thereby creating new nodes and members in the corresponding form diagram (Akbarzadeh et al., 2014, 2015c; Nielsen et al., 2017; Ghomi et al., 2018). As subdivision can be applied only to an existing cell, it is essentially a topology refinement methodology where each subdivision adds new members that generally follow the initial geometry and topologyof the structure. Alternatively, additive transformation can be used in combination with shape grammars to automatically generate highly non-standard topologies (Lee et al., 2016). Additive transformation methodology has been implemented only for 2D structures.

2.5 Computational tools for polyhedral geometries

In order to create an interactive design environment using 3D graphic statics, sufficient computational back-end for handling and processing of polyhedral geometries is needed. This section discusses the pros and cons of some of the existing CAD software and computational implementations of polyhedral geometries that could potentially be suitable for 3D graphic statics. Tools and implementations for single polyhedral meshes as well as volumetric meshes are reviewed.

2.5.1 Polyhedral mesh

Most computational techniques in engineering sciences are based on finite discretisations of a physical specimen or a defined domain into manifold meshes made up of polygonal elements. Computational implementation of mesh datastructures is a well-researched topic across many disciplines. Depending on the purpose and the objectives of each discipline, variants of mesh datastructures have been proposed over the past few decades. Detailed explanations and comparative overview of various mesh datastructures can be found in Garimella (2002), Alumbaugh and Jiao (2005) and De Floriani and Hui (2005; 2007).

There also have been significant advances in modelling tools for free-form surface geometries in architecture. In contrast to engineering applications where predefined geometries are typically discretised into smaller polygonal elements, CAD software are designed for creating and modelling surfaces or solid geometries. Therefore, the representation of geometry in some of the most widely used CAD software in architecture such as Rhinoceros and Maya (Autodesk Inc., 2018) are based on a non-uniform rational B-spline (NURBS) model, which focuses on more mathematically precise representations of curves and free-form surfaces. Essentially, these CAD software allow users to create and generate complex free-form geometries using a few number of descriptors or "control points." Alternatively, the geometries can also be described and represented in a CAD environment by polygon meshes, which are collections of vertices, edges and faces. Polygon meshes commonly consist of faces that are triangles or quadrilaterals. A polygon mesh is also referred to as a "polygonal (or polyhedral) mesh" due to the fact that its constituent faces are polygonal.

Research in the applications of polygon meshes in CAD environments are extensive, which includes boolean logic, smoothing, ray tracing, collision detection, rigid-body dynamics and many others. Polygon meshes typically represent a single continuous surface, or a solid object as a surface that is self-enclosing. For 3D graphic statics application, a polygon mesh can be sufficient to represent a single polyhedral cell. However, as most structures usually contain multiple nodes, polygon meshes are not sufficient for representing the corresponding polyhedral force diagrams, which have multiple cells and enclosures. Furthermore, certain key geometric properties of polyhedral reciprocal diagrams such as the planarity of the faces, is not intrinsically enforced.

In the context of computational geometry, various mathematical models, geometric algorithms and design interfaces have been developed to enable designers to incorporate a wide range of geometric and practical constraints while maximising user interactivity and creative flexibility. Planarity constraint in particular, is an important topic of research in computational geometry of free-form architectural surfaces because of the advantage of using

flat materials with regards to fabrication and cost. With planarity as the main focus, numerous rationalisation algorithms have been developed to achieve various geometric goals: optimisation of planar quadrilateral meshes (Liu et al., 2006; Zadravec et al., 2010); constrained mesh modelling (Deng et al., 2015; Vaitkus and Várady, 2015); multi-layered mesh modelling (Pottmann et al., 2007, 2015; Jiang et al., 2015); surface penalisation (Pottmann et al., 2008; Eigensatz et al., 2010); statics-aware mesh modelling (Schiftner and Balzer, 2010; Vouga et al., 2012; Tang et al., 2014; Miki et al., 2015) (Figure 2.12-a); 3D mesh modelling (Poranne et al., 2013, 2015) (Figure 2.12-b); and numerous others in the literature. Interactive implementations of algorithmic modelling such as Shape-Up (Bouaziz et al., 2012) and ShapeOp (Deuss et al., 2015) have enabled common architectural design tools like Rhinoceros to become a much more sophisticated modelling tool with fully-integrated geometric constraints and algorithms (Figure 2.12-c).

Despite the rapid development of mesh modelling tools and algorithms, previous research in this area is primarily concerned with single surface geometries. With polyhedral force diagrams used in 3D graphic statics, the constituent cells are individually closed and often intersect with one another. Similar geometric constraints and algorithms are applicable to a system of force polyhedrons, and can certainly be used to drastically improve the way in which the user can interact with the geometry of polyhedral cells.

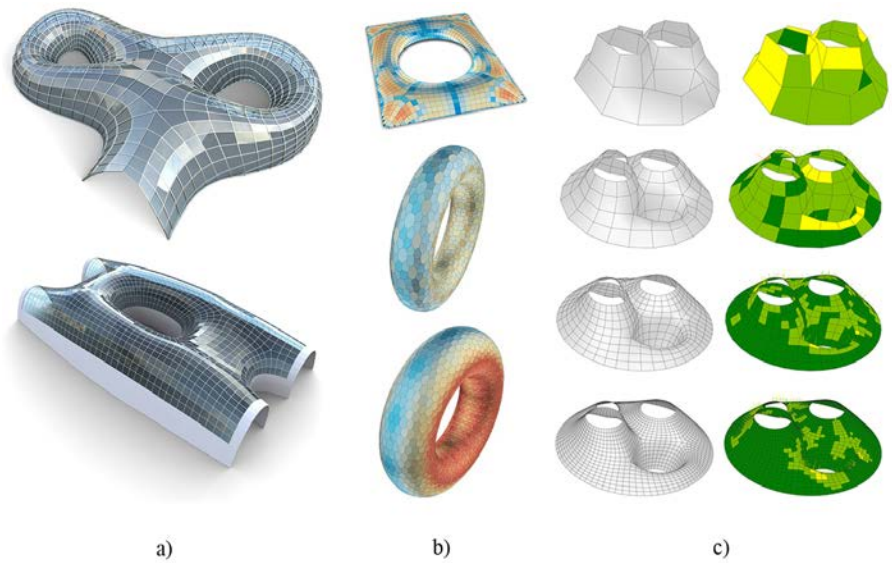


Figure 2.12: a) Planar meshing of self-supporting surface structures (Vouga et al., 2012); b) interactive planarisation and optimisation of 3D meshes (Poranne et al., 2013); and c) use of ShapeOp for constrained modelling of various meshing schemes with multiple geometric constraints imposed simultaneously (Deuss et al., 2015).

2.5.2 Volumetric mesh

In 3D graphic statics, a single force polyhedron can be computationally represented with a mesh. However, a polyhedral force diagram consisting of multiple cells cannot be represented with a single, manifold mesh. A multi-cell object is also referred to as a non-manifold, and a different datastructure is necessary to computationally handle such object. Several datastructures for representing and modelling non-manifold objects have been previously proposed: non-manifold indexed data structure with adjacencies (NMIA), which is a decomposition of non-manifold objects into simplical complexes (De Floriani and Hui, 2003); Algorithm Oriented Mesh Database (AOMD) (Remacle and Shephard, 2003); radial edge datastructure (RED) (Weiler, 1988) as an extension to non-manifold objects of winged-edge datastructure (WED) (Baumgart, 1975); Linear Cell Complex datastructure (Damiand, 2018) as a part of the Computational Geometry Algorithms Library (CGAL) (The CGAL Project, 2018) (Figure 2.13-a); the OpenVolumeMesh library (Computer Graphics Group Aachen, 2016) developed by Kremer et al. (2013) (Figure 2.13-b); and etc.

These datastructures implement different data processing strategies, organisation and querying techniques depending on the context of the discipline that they were intended for. While the computational formulation of the datastructures are usually the focus of these contributions, they overall lack the accompanying set of functions or operations to translate them to any practical applications. More importantly, the user manual or documentation of these libraries lack visual material, tutorial or examples to describe these complex datastructures, which may be difficult to understand and visualise for non-expert users.

Within the context of 3D graphic statics, notable computational implementations of polyhedral force diagrams where the specific type of datastructure used were explicitly mentioned, are based on WED (Akbarzadeh et al., 2015b) (Figure 2.13-c) and "volume mesh" (Reeves et al., 2016). In either of these applications, detailed explanations of how the data is computationally structured and implemented were never provided. The interpretation of the various components of datastructure as they relate to 3D graphic statics is also not thoroughly described or documented.

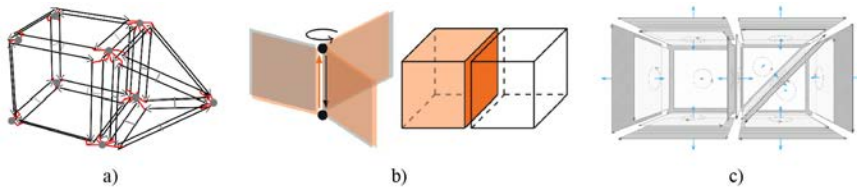


Figure 2.13: a) Linear Cell Complex datastructure (Damiand, 2018); b) winged-edge datastructure (Akbarzadeh et al., 2015b); and c) OpenVolumeMesh (Kremer et al., 2013).

2.6 Materialisation

The relationship between reciprocal form and force diagrams is inherently geometric and does not address any material information or properties. They are useful for early-stage design explorations and form finding of structural geometry. However, an additional procedure is required to materialise the skeletal line information of form diagrams into a realistic structure.

Skeletonisation is a procedure for extracting a topological skeleton or the medial axes (vertices and a network of edges) of a mesh or a solid object (Saha et al., 2016). The inverse procedure (the thickening or materialisation of a network of lines) has relevant applications in numerous disciplines such as computer graphics, character modelling and animations (Bærentzen et al., 2012). Solidifying of lines in computer graphics applications are primarily concerned with rapid geometric approximation with maximum visual perception and recognition. A well-known Rhinoceros plugin for solidifying wireframes is Exoskeleton and Cytoskeleton developed by Daniel Piker and David Stasiuk, which are based on concepts and techniques described in Srinivasan et al. (2005).

In general, these methods first materialise the vertices using spheres or a variation of convex hull, then interpolate those vertex masses along the edges using sweeps or lofts. The vertex masses and swept edges masses are then combined and refined to create a single, continuously smooth mesh surface (Figure 2.14). However, the smooth mesh around the nodes is complex in geometry and difficult to fabricate. The complexity is not problematic for small 3D printing applications, whereas it would quickly become unfeasible with larger structures. Subtractive milling from solid masses is another alternative, which is time consuming, inefficient use of material and expensive to fabricate.

First full-scale prototype of a 3D graphic statics design is the Hedracrete project (Akbarzadeh et al., 2017). A thorough structural analysis of the structure can be found in Bolhassani et al. (2018). Hedracrete is a self-

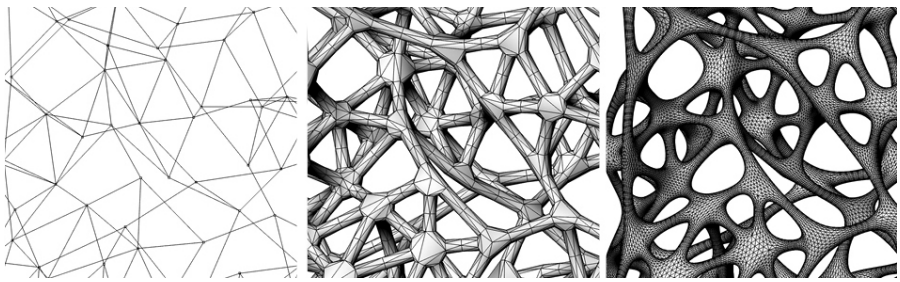


Figure 2.14: Conversion of wireframe data to thickened, wireframe meshes using Exoskeleton, a plugin for Rhinoceros developed by David Stasiuk.

Figure 2.15: Hedracrete, first full-scale prototype designed using 3D graphic statics (Bolhassani et al., 2018).

Figure 2.16: Built prototype of a spatial structure using CNC-cut sheet materials (Reeves et al., 2016).

supporting, funicular structure made of prefabricated nodes and linear members (Figure 2.15). The prefabricated elements are made of glass fibre reinforced concrete, with extra steel rebars embedded in the tensile members. Although this project showcased numerous benefits of 3D graphic statics, the process of generating the geometry of the individual elements is not explicitly described. It is not clear how the inherent polyhedral geometry was exploited to generate or optimise the geometry of the CNC-milled moulds, especially for the complex node elements.

David Reeves et al. (2016) proposed a method for materialising spatial structures using only developable strips cut from sheet materials and inexpensive 2-axis CNC technology (Figure 2.16). The fabrication geometry was generated by exploiting the the dual relationship between the geometry of the polyhedral form and force diagrams. This project demonstrated how the inherent planar properties of the polyhedral force diagrams can be used to develop the logic of the massing of the spatial structure, and demonstrate its advantages as they relate to fabrication and construction through modest sheet materials. However, this project was an exercise in processing and development of fabrication geometry, and the prototype is not based on any realistic structural design loads or boundary conditions.

2.7 Summary

This chapter presented the literature review of the relevant theoretical background, previous research and existing implementations of graphic statics with particular emphasis on computational graphic statics.

First, the various factors that led to the emergence of graphic statics as a widely accepted structural design and analysis method during the 19th century were discussed. At that time, graphic statics was an appropriate method for analysing truss-like structures made of linear, iron or steel members carrying axial loads only. However, as the building industry began to move towards using indeterminate materials such as reinforced concrete, the usefulness and relevance of graphic statics in structural engineering practice started to fade. The developments of computers and FE methods over the 20th century also contributed to the decline of graphic statics.

In recent years, graphic statics has been going through a revival. With today's rapid advancements in parametric CAD software, graphic statics drawings no longer have to be drawn manually on paper. In addition, drawings can be constructed parametrically such that real-time interactivity is possible even after the drawings have already been completed. However, interactive graphic statics drawings need to be procedurally constructed by someone who has previous knowledge and experience with graphic statics. Most importantly, it is difficult to make any design or topological changes to the interactive drawings once they are complete.

AGS introduced a method for automatically constructing the force diagrams from a given form diagram. In the case of bi-directional AGS, the automatic generation and real-time updating can happen from either from the form diagram to the force diagram, or vice versa. Algebraic formulation and construction of reciprocal diagrams means that the tedious and error-prone procedural construction of the reciprocal diagrams can be bypassed. Although it establishes a robust back-end for a real-time, interactive and flexible computational implementation of traditional graphic statics, algebraic graph statics remains applicable to only specific types of structures in 2D.

Extension of graphic statics to three dimensions introduced polyhedral reciprocal diagrams. Because the polyhedral reciprocal diagrams are constructed with faces and polyhedral cells opposed to lines and polygons in 2D, 3D graphic statics can be used for a variety of applications for the design and analysis of spatial structures. Although much of the basic theoretical foundation for polyhedral 3D graphic statics have been established, it is evident that there is a general lack of computational tools to sufficiently handle complex polyhedral geometries in an interactive design environment. The lack of visual clarity and inadequate legibility of polyhedral force diagrams are some of the major drawbacks of polyhedral 3D graphic statics.

Within the context of structural engineering, graphic statics provides many advantages over conventional FE software. For example, the formulation of a structural optimisation problem can be drastically simplified when the geometry of the force diagram is optimised, as opposed to the form diagram. Using only a few control points or parameters, the connectivity of the entire structure can be described and controlled. Another benefit of using reciprocal diagrams for structural design is the ability to explicitly set force-based constraints by geometrically constraining specified elements of the force diagram. In contrast to force-driven design using numerical methods like the force density method, force-driven design using graphic statics may be more intuitive for the user and has a higher potential for the user to draw new insightful conclusions. Graphic statics in engineering applications have been applied to mainly 2D structures only, and the topology of those structures are often predefined.

Mesh is a well-researched computational datastructure for representing and modelling surface and solid geometries. Theoretical background and computational implementation of meshes are well studied and established in the field of computational geometry and numerous other disciplines. In contrast, volumetric meshes, or 3-manifold meshes, have seen comparatively less development. This may be due to the fact that most computational models or representations of meshes are concerned with discretisation of real-life objects (i.e. animations, models of physical objects for FE analysis) which can typically be modelled as single surfaces or solid objects. Whereas in a volumetric mesh representation of a polyhedral force diagram, multiple cells are inter-linked with the possibility of self-intersections and overlaps. The established computational libraries built around meshes may not necessarily translate to or be compatible with volumetric meshes.

Graphic statics is widely accepted as an intuitive method for form-finding and equilibrium analysis of structural geometry. However, materialisation of the form diagram is not typically incorporated into graphic-statics-based design workflow. Although polyhedral 3D graphic statics allows exploration of fully spatial structures, the materialisation and fabrication design of spatial structures remain challenging. The nodes of spatial structures are often treated as special, one-off components that are difficult to resolve geometrically and expensive to fabricate. Various methods exist in literature which can turn wireframe models of lines into a single mesh surface. While such methods are appropriate for small models and prototypes which can be easily produced using additive manufacturing methods like 3D printing, it does not particularly scale well for structural components at a 1-to-1 scale. Although 3D graphic statics broadens the repertoire of spatial structural typologies, the fabrication aspect also needs further research to ensure that complex spatial structures can be realised as meaningful structures in a feasible manner.

3 Scope of Work

Based on the literature review presented in the previous chapter, this chapter defines the specific problem statements of the dissertation. This chapter concludes Part 1 of the dissertation by providing an outline of the research objectives.

3.1 Problem statements

This section presents the specific problem statements that will be addressed in this dissertation. These problems can be contextualised in a realistic design workflow that can be adapted by designers in practice (Figure 3.1).

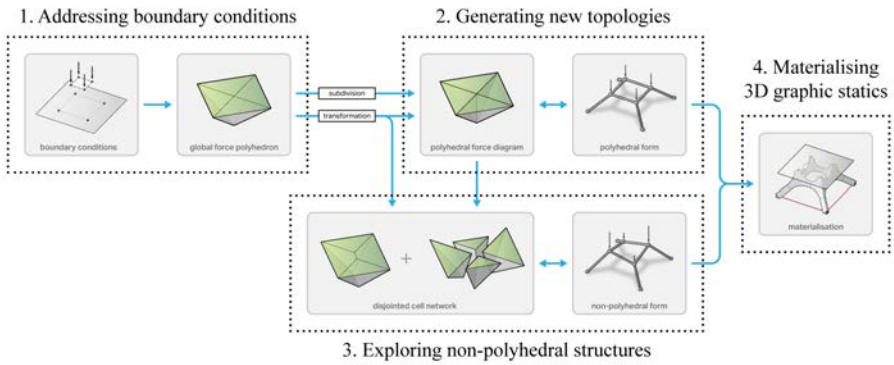


Figure 3.1: Design workflow using 3D graphic statics: 1) addressing realistic boundary conditions; 2) new topology generation methods beyond subdivision; 3) exploring non-polyhedral structures; and 4) materialisation of spatial structures.

1. **Addressing realistic boundary conditions**

   In the beginning stages of a project, the only known information are typically the type of loading the structure will carry (applied load locations and magnitudes) and the physical constraints of the site (i.e. support locations, types, allowable reaction force magnitudes, etc.) (Figure 3.1-1). Before any design explorations using 3D graphics can take place, the quantitative (force magnitudes) and spatial (force locations) site constraints which are unique for each project need to be incorporated during the construction of the initial global force polyhedron. Although the global force polyhedron for determinate and highly symmetric cases can be constructed procedurally (Akbarzadeh

et al., 2015a, 2016), there is no generalised method for addressing this "blank space" problem with sufficient control, especially with regards to controlling the face areas (force magnitudes) during the construction of the global force polyhedron.

2. **Topology generation method beyond subdivision**

   Once the global force polyhedron has been constructed, various subdivision schemes can be applied to generate and explore structural topologies (Akbarzadeh et al., 2015c) (Figure 3.1-2). Subdivision of internal polyhedral cells increases the complexity of the structural topology without changing the initial boundary conditions. As a result of subdividing a polyhedral cell, a single member of the corresponding form diagram is replaced with multiple members that together carry the same magnitude of axial force as the replaced member. Therefore, subdivision can be an effective method for addressing buckling, although it is a trade-off for challenging fabrication due to the diminishing member-to-member angles as the overall number of members is increased. Subdivision schemes are highly dependent on the initial funicular topology embedded within the global force polyhedron. The repeated subdivision of a global force polyhedron generally results in designs that are refinements of the initial funicular form and topology.

3. **Limitation to polyhedral forms**

   The inherent dual and reciprocal relationship between the polyhedral form and force diagrams means that the form diagrams are strictly limited to polyhedral geometries (i.e. subdivided tree structures, faceted domes, polyhedral mesh or surface structures, crystalline aggregations, etc.) (Figure 3.1-3). Therefore, polyhedral force diagrams can represent the equilibrium of only highly constrained and specific types of structures. With current implementations of polyhedral 3D graphic statics, it is impossible to construct polyhedral force diagrams for equilibrated structures that are non-polyhedral (i.e. structures with overlapping members, non-planar faces, spatial structures with 2D nodes, etc.). Polyhedral constraint is beneficial for applying global manipulations and transformations, but ultimately limits the range of structural typologies that can be explored.

4. **Materialisation of spatial structures**

   The final step of the 3D graphic statics design workflow is the materialisation of the network of lines that represents the form diagram (Figure 3.1-4). The form diagrams in graphic statics applications represent the topology and geometry of equilibrium structures, but

they do not carry any material information. While complex structural toplogies can be quickly generated using the 3D graphic statics design workflow, addressing the materialisation and practical feasibility of such complex network of lines in space, especially the nodes, is important in expanding 3D graphic statics applications beyond computational form-finding explorations of abstract shapes. Existing "wire thickening" methods can be acceptable for small scale applications where the geometric complexity of the components can be resolved through additive fabrication methods such as 3D printing. However, as the structure increases in scale, a more generalised method for rationalising the fabrication geometry of complex spatial nodes is needed for both economic and practical reasons.

5. **Insufficient user control and visualisation**

The 3D graphic statics design workflow provides the users with explicit control of both form and force diagrams. The user's interaction with these computational objects in an interactive design environment is dependent on the clarity of visualisation and control. Polyhedral force diagrams are spatial objects that are difficult to visualise and understand as static drawings or objects through 2D media, such as papers or computer monitors. With increasing complexity of structures, polyhedral force diagrams quickly become visually cluttered, losing all of the inherent benefits that makes 2D graphic statics a uniquely insightful and intuitive tool. In general, alternative techniques for visualising polyhedral form and force diagrams are needed. Computational frameworks can be constructed as efficiently and robustly as possible, but without sufficient user control and discernible visualisation, its impact and usefulness are limited within the design community. While the framework can be used to handle the back-end polyhedral computation and processing, conventional CAD software can be used as the canvas. However, most CAD software used by designers in architecture and structural engineering are not designed for handling cellular networks or volumetric meshes. Therefore, the built-in visualisation and interactive functionalities of those CAD software need to be customised and calibrated for 3D graphic statics applications.

6. **Lack of computational library for 3D graphic statics**

There is a lack of a generalised computational framework for 3D graphic statics, which hinders continuation of research, exchange of computational knowledge and collective development of the field. New publications of graphic statics are often stand-alone implementations with their own set of conventions, computational languages and software dependencies, which make them incompatible or unusable

by other researchers. New knowledge is often shared through papers in physical or digital formats that describe the computational implementations with text and static images without actually delivering any usable computational material to the readers. Anybody who is interested in doing research on 3D graphic statics must start his or her implementation from scratch, unless he or she joins a research group with an established computational framework. The lack of a unifying platform or computational environment for graphic statics makes it difficult to start new strands of research or continue existing ones.

3.2 Research objectives

The components of the presented computational framework are developed in response to the problem statements outlined in the previous section. The specific research objectives for each of the problem statements are:

1. **Addressing realistic boundary conditions**
    - Formulation of a generalised method for constructing global force polyhedrons from a spatial system of forces in a controlled manner, while incorporating various boundary condition constraints
    - Computational implementation of new and existing theories, concepts and methods for single polyhedral cells, which establishes a robust base for the modelling, transforming and processing of convex as well as non-convex polyhedrons

2. **New topology generation method beyond subdivision**
    - Development of a datastructure for multi-cell polyhedrons with related functionalities and algorithms, which enable controlled modelling and transformations of polyhedral form and force diagrams
    - Expansion of polyhedral transformation operations beyond subdivision
    - A new design workflow for allowing users to manually generate and sculpt equilibrium structures in space in a controlled manner

3. **Limitation to polyhedral structures**
    - Development of a datastructure for generalised cell networks that can represent the spatial equilibrium of non-polyhedral structures
    - Enabling force-driven design through more precise control of the areas of the polyhedral faces, incorporating more quantitative design criteria and realistic load cases (tributary area loads, internal point loads)

- Broadening the range of new structural typologies that can be addressed with polyhedral 3D graphic statics using the cell network datastructure

4. **Materialisation of spatial structures**

   - Utilisation of the geometry of 3D force diagrams to develop and rationalise the fabrication geometry

5. **Insufficient user control and visualisation**

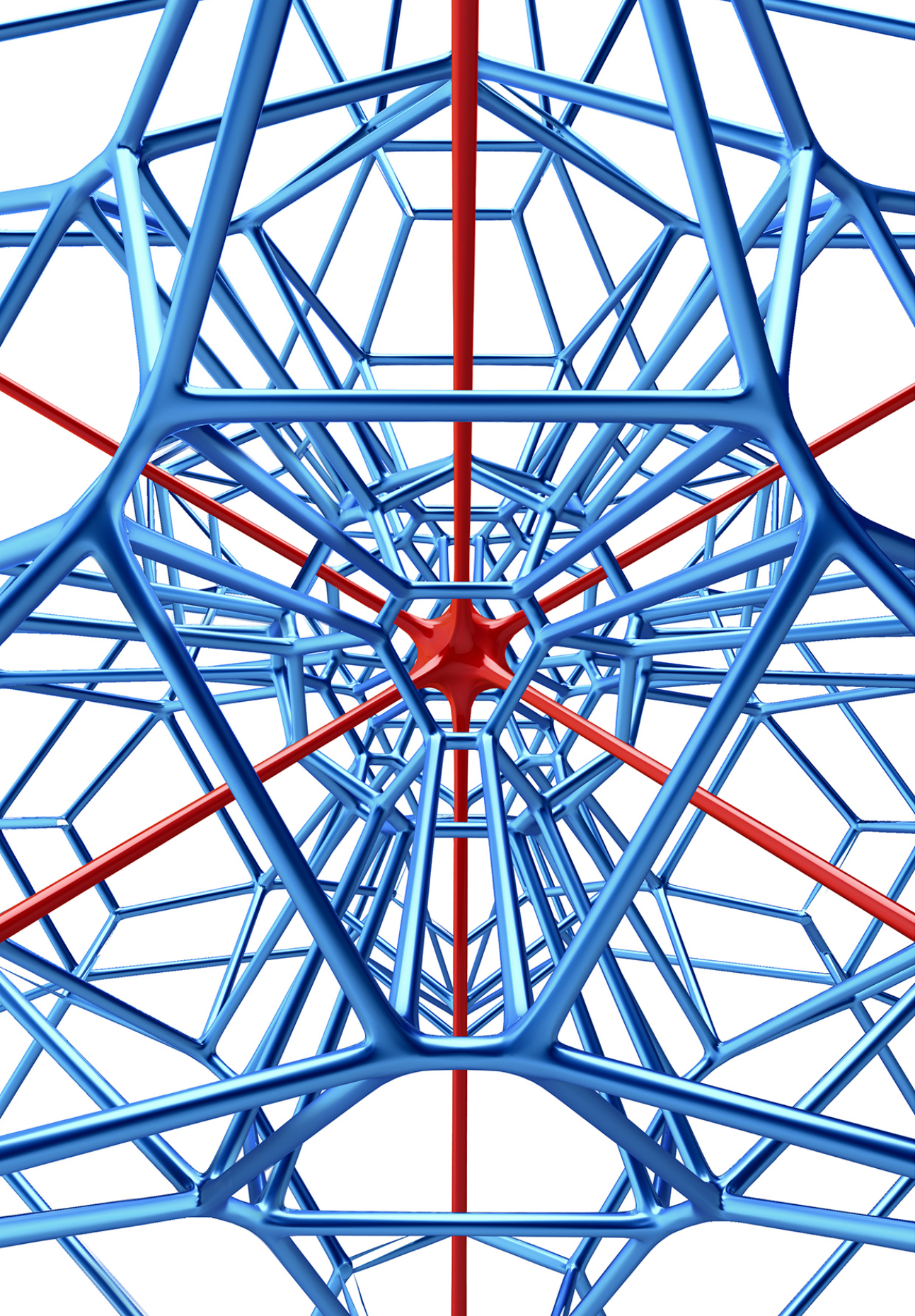
   - Improvement of user control of various polyhedral objects and datastructures in an interactive design environment
   - Dynamic drawing and visualisation strategies for enhancing the legibility of 3D form and force diagrams

6. **Lack of computational library for 3D graphic statics**

   - Implementation of the presented computational framework in an open-source library, making the contributions of this research available to the public
   - An online documentation of the library for future users, researchers and contributors

3.3 Summary

This chapter concludes Part 1 of the dissertation, which presented the motivation statements, an overview of the state of the art, the problem statements and an outline of specific research objectives. Based on the problem statements and research objectives identified in this chapter, the remainder of the dissertation is organised as follows. In Part 2, "Computational Framework," the theoretical background will be presented in Chapter 4, where the main datastructures for 3D graphic statics design workflow are introduced and formulated. Computational implementation of the theoretical background will be presented in Chapter 5, "compas_3gs." Then later in Part 3, "Applications," new structural design applications using the computational framework will be presented.



Part II

Computational Framework

4 Theoretical background

This chapter presents the theoretical background of the computational framework. The theoretical background and concepts are developed in response to the shortcomings and limitations of 3D graphic statics outlined in the previous chapter. At the core of the presented theoretical background is the formulation of three datastructures that can be used to address different types of equilibrium problems using 3D graphic statics. In the next chapter, "compas_3gs," the implementation of the computational framework will be presented.

Parts of this chapter are based on these publications by the author:

- *Form-finding explorations through geometric manipulations of force polyhedrons* (Lee et al., 2016)
- *Area-controlled construction of global force polyhedra* (Lee et al., 2017)
- *Disjointed force polyhedra* (Lee et al., 2018)

4.1 Introduction

This section provides a brief overview of this chapter: a summary of the key concepts and terminologies of the theoretical background; the notation that will be used throughout the dissertation; and an outline of the chapter.

4.1.1 Overview and key terminology

This chapter presents three datastructures for representing polyhedral force diagrams that address different types of equilibrium problems involving spatial systems of forces. Each datastructure is unique in its construct and has specific applicability and purpose in a 3D graphic statics design workflow. The three datastructures are summarised as follows:

- **Polyhedral cell**

  A *polyhedral cell* is a single force polyhedron that can represent the equilibrium of: 1) the external forces acting on a structure; or 2) a single node of a multi-node structure.

- **Multi-cell polyhedron**

  A *multi-cell polyhedron* is an aggregation of conjoined polyhedral cells that can represent the equilibrium of a structure with multiple nodes, which is polyhedral in its geometry.

- **Cell network**

  A *cell network* is a collection of disjointed polyhedral cells that are implicitly linked to one another. Cell networks can represent the static equilibrium of any nodally-loaded spatial truss comprised of straight bars, which may be either polyhedral or non-polyhedral in its geometry.

Investigation of single polyhedral cells is important for formulating generalised utility and geometric functions that are applicable to all three datastructures. A better understanding of the topological structure of different types of polyhedral cells is essential for developing a computational framework that can process convex as well as non-convex polyhedral cells. Methods for constructing polyhedral cells from a set of external forces is necessary for addressing boundary conditions and initialising the 3D graphic statics design pipeline.

Convex and regular polyhedrons as geometric or topological entities are well-studied topics in numerous disciplines. However, there remains a research gap between the established knowledge and their translation to a computational design environment. The goal of this chapter is to bring together dispersed knowledge from the related fields, such as the Extended Gaussian Image, to establish a robust theoretical base necessary for sufficient computational representation, construction and processing of polyhedral cells.

The current state of the art of 3D graphic statics is largely based on multi-cell polyhedrons that consist of conjoined polyhedral cells, where every pair of two adjacent cells is separated by a single face. Although the basic principles and design potentials (Akbarzadeh et al., 2015b) and mathematical descriptions (McRobie, 2017a) of multi-cell polyhedrons have been presented, a detailed formulation of the computational datastructure and its related functions have not been addressed in the literature.

A volumetric mesh datastructure that is based on a halfface datastructure is introduced in this chapter to represent multi-cell polyhedrons. A thorough description of the datastructure is not only important for research in 3D graphic statics, but also for establishing a more rigorous dialogue with researchers from other fields as computational geometry. The multi-cell polyhedron datastructure can be used to revisit previously presented design implementations and explore new design applications.

Because all of its constituent cells are conjoined, a multi-cell polyhedron can be especially useful for global transformations such as subdivision operations and applying general algorithms (Akbarzadeh et al., 2015c). However, multi-cell polyhedrons strictly limit the corresponding structures to be polyhedral in their geometry (i.e. subdivided tree structures, faceted domes, polyhedral mesh or surface structures, crystalline aggregations, etc.). The polyhedral geometry constraint also makes it difficult to perform transformation operations where the topology of the multi-cell polyhedron may need to change in order to accommodate the required geometric changes.

Cell networks are introduced to address these limitations of multi-cell polyhedrons. The neighbouring cells of a cell network are no longer conjoined through a single face. Instead, they are interfaced by two faces that are equal in area but may be different in their geometries. The individual cells represent the local equilibrium of a node of a structure, but collectively represent the equilibrium of the entire structure. Without the global topological constraint of a multi-cell polyhedron, cell networks enable transformation operations where the topology of individual cells can change independently from one another. Cell networks also allow a more explicit force-driven design by prioritising area-based constraints over geometric ones. Most importantly, cell networks can represent the equilibrium of structures that are not necessarily polyhedral in their geometry. As a result, the range of typologies of structures that can be explored with 3D graphic statics is expanded.

The three datastructures summarised in this section address the shortcomings of the current implementations of 3D graphic statics. Polyhedral cells, multi-cell polyhedrons and cell networks are the fundamental building blocks for expanding the range of structural design applications using 3D graphic statics. Together, they provide the theoretical foundation for the presented computational framework.

4.1.2 Notation

Figure 4.1 is a summary of key notations and symbols that will be used throughout this chapter as well as the remainder of the dissertation. In general, the force diagrams have the superscript $\perp$, which signifies that the elements of the force diagram are in perpendicular orientation to the corresponding element in the form diagram (edge to edge in 2D, and face to edge in 3D). The prime symbol ($\prime$) is used to signify dual elements or relationships. Any elements or variables that are exclusive to the force diagrams are suffixed with an asterisk ($*$).

Form diagram			Force diagram	
G	2D form diagram	face dual & perpendicularity	$G^{\perp}$	2D force diagram (Maxwell)
		face dual & parallelity	$G^{\parallel}$	2D force diagram (Cremona)
Λ	external forces	cell dual & perpendicularity	$\Lambda^{\perp}$	global force polyhedron
Γ	polyhedral form diagram	cell dual & perpendicularity	$\Gamma^{\perp}$	multi-cell polyhedron (polyhedral force diagram)
Ψ	non-polyhedral form diagram	perpendicularity only	$\Psi^{\perp}$	cell network (disjointed/non-polyhedral force diagram)
$\mathbf{v}_i$	vertex, form diagram	cell dual & perpendicularity	c^*_i	i-th polyhedral cell
			EGI_i	EGI of c^*_i
			face dual	
$e_{i,j}$	edge, form diagram (from $\mathbf{v}_i$ to $\mathbf{v}_j$)	cell dual & perpendicularity	$f^*_{i,j}$, $\hat{\mathbf{n}}^*_{i,j}$	face, normal (j-th face of i-th cell)
			hf^*_j, $\hat{\mathbf{n}}^*_j$	halfface, normal (j-th halfface of $\Gamma^{\perp}$)
$\hat{\mathbf{e}}_{i,j}$	unitised edge vector (from $\mathbf{v}_i$ to $\mathbf{v}_j$)		$h^*_{a,b}$	halfedge (from $\mathbf{v}^*_a$ to $\mathbf{v}^*_b$)
$\mathbf{f}_{i,j}$	force in $e_{i,j}$	amplitude	$A^*_{i,j}$, A^*_j	area (of j-th face of i-th cell, j-th halfface of $\Gamma^{\perp}$)

Coefficients and factors

α	unified diagram scale factor
γ	reciprocation weight factor
λ	edge to face normal direction coefficient
s	half-face scale factor
ζ^*	prism thickness scale factor

Superscripts and subscripts

$\_'$	dual element
$\_^*$	element in the force diagram
$\_^{(t=0)}$	@ time step t = 0
$\_^{target}$	target
$\__{lb}$	lower bound
$\__{ub}$	upper bound

Other notations

$\mathbf{b}$	barycenter
$\hat{\mathbf{t}}$	target unit vector

Figure 4.1: Nomenclature of the key terms and symbols.

4.1.3 Chapter outline

The remainder of this chapter is divided into three main sections.

In Section 4.2, the polyhedral cell datastructure is presented. The organisation and structure of the data of polyhedral cells will be described in detail, followed by a discussion on the dual relationships between the various elements of a polyhedral cell and the corresponding elements in the form diagram. The Extended Gaussian Image is introduced as an unitised dual representation of a polyhedral cell, which is useful for visually describing and understanding the topological structures of various types of polyhedral cells. A generalised method for computing the oriented normal and area of any closed polygon is presented, which is essential for interpreting the external or internal member forces in the form diagram. A method of interpreting 2D equilibrium using polyhedral cells is also described. Next, geometric operations that enable constrained transformations of a polyhedral cell through pulling and tilting of faces are presented. This section concludes by demonstrating how the geometry of a polyhedral cell can be computed from a set of equilibrated forces in space using the Extended Gaussian Image.

In Section 4.3, the multi-cell polyhedron datastructure is presented. A detailed explanation of the various layers of data that define a multi-cell polyhedron is provided. Interpretation of a multi-cell polyhedron's cell adjacency information and the relationship to the forces in the corresponding structure with multiple nodes is discussed. Unified diagrams are introduced and presented in this section, which have numerous benefits with regards to the visualisation of multi-cell polyhedrons. This section concludes by presenting an operation that enables a constrained geometric transformation of multi-cell polyhedrons.

Finally, in Section 4.4, the cell network datastructure is presented. A cell network is a set of disjointed polyhedral cells, which requires additional layers of data to manage and control the connectivity relationships between every pair of neighbouring cells. These additional layers, which are unique to cell networks, will be described in detail. An approximated visualisation strategy for the unified diagram of cell networks is also presented in this section.

4.2 Polyhedral cell

This section presents the polyhedral cell datastructure, which can represent the static equilibrium of the external forces acting on a structure, or a single node of a multi-node structure. This section also introduces the Extended Gaussian Image, a spherical representation of polyhedrons that has numerous benefits with regards to the topological understanding and transformations of polyhedral cells. The Extended Gaussian Image is used to develop the key utility functions, transformation operations and reconstruction methods for polyhedral cells.

4.2.1 Definition

In 3D graphic statics, the equilibrium of the external forces acting on a structure or a node of a structure can be represented by a polyhedral cell (Rankine, 1864; Akbarzadeh et al., 2015b). The equilibrium of the external forces of a structure, Λ (Figure ??-a), is represented by a polyhedral cell, which is the *global force polyhedron*, $\Lambda^{\perp}$ (Figure ??-c). The equilibrium of the i-th *node* $\mathbf{v}_i$ of a structure (Figure ??-b), is represented by a polyhedral cell c_i^* (Figure ??-c). For each polyhedral cell c_i^*, the normal $\hat{\mathbf{n}}_{i,j}^*$ and the area $A_{i,j}^*$ of the j-th face $f_{i,j}^*$ represent the direction and magnitude of the force $\mathbf{f}_{i,j}$, respectively. $\mathbf{f}_{i,j}$ corresponds either to an external force of Λ, or the internal force of the member $e_{i,j}$ of the structure.

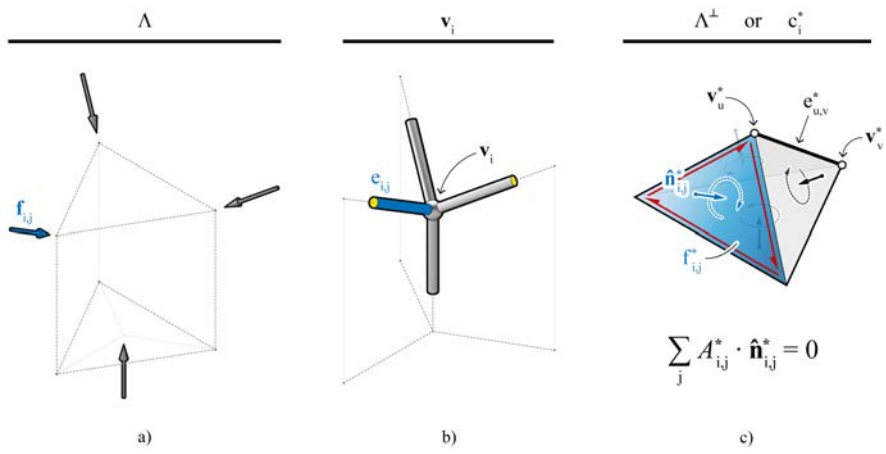


Figure 4.2: a) external forces acting on a structure, Λ; b) the i-th node $\mathbf{v}_i$ of a structure; and c) the global force polyhedron $\Lambda^{\perp}$ that represents the equilibrium of the external forces, or the polyhedral cell c_i^* that represents the forces acting on node $\mathbf{v}_i$.

4.2.2 Datastructure

The topology and geometry of a polyhedral cell, which is essentially a mesh, can be represented by a halfedge datastructure that stores incidence information of vertices, edges and faces. Each edge of a mesh is decomposed into two halfedges with opposite directions. Each halfedge stores the information of the face that it belongs to. Using the halfedge directions and incident face information, various data of the mesh can be accessed and traversed. The data of a mesh thus consists of vertices, edges, faces and halfedges.

Each k-th *vertex* $\mathbf{v}_k^*$ of the polyhedral cell c_i^* is stored as a vkey-value pair (Figure 4.3-a) in the mesh datastructure. The vertex key (**`vkey`**) is the unique identifier of the vertex, and the value itself stores the specific attributes for each vertex. **`x`**, **`y`** and **`z`** are the default attributes of each vertex, and they define the xyz coordinate of the vertex, and therefore the geometry of the entire mesh.

Each *edge* $e_{u,v}^*$ of the polyhedral cell c_i^* is stored as a vkey-vkey-value triple (Figure 4.3-b) in the mesh datastructure. An edge **`(u,v)`** is an ordered pair of two unique **`vkeys`** (from "tail" vertex **`u`** to "head" vertex **`v`**). The **`vkeys`** **`u`** and **`v`** are the two keys of the key-key-value triple. The edges are first organised by their tail **`vkeys`**, **`u`**. Each **`u`** then points to all **`vkeys`** **`v`** for all the edges **`(u,v)`** whose head **`vkey`** is **`v`**. Because each edge is directed and defined by two uniquely ordered **`vkeys`**, an edge is stored only once in the datastructure. For example, there is only one entry for edge **`(0,2)`**; there is no entry for **`(2,0)`**. Each edge **`(u,v)`** points to the value of the key-key-value triple, which contains the edge attributes.

Each j-th *face* $f_{i,j}^*$ of the polyhedral cell c_i^* is stored as a fkey-value pair (Figure 4.3-c). The face key (**`fkey`**) is the unique identifier of the face, and the value is an ordered list of **`vkeys`** that form the polygonal loop of the face. The first **`vkey`** and the last **`vkey`** are not the same. Every consecutive pair of this list of ordered **`vkeys`** (including the last to the first), is a halfedge of the mesh.

Lastly, each *halfedge* $h_{u,v}^*$ of a mesh is stored as a vkey-vkey-value triple (Figure 4.3-d). Each halfedge **`(u,v)`** is uniquely defined by a tail **`vkey`** **`u`** and head **`vkey`** **`v`**. The halfedges are first organised by the tail **`vkeys`** **`u`**, then by the head **`vkeys`** **`v`**. The **`vkeys`** **`u`** and **`v`** are the two **`vkeys`** of the vkey-vkey-value triple. Each halfedge **`(u,v)`** points to the value of the vkey-vkey-value triple, which is the **`fkey`** of the face that the halfedge belongs to. Halfedges do not have any attributes.

Figure 4.3 is a detailed graphical representation of a polyhedral cell c_i^* with six faces, which represents the equilibrium of six concurrent forces of Λ or $\mathbf{v}_i$. The 0-th face $f_{i,0}^*$ is defined by four **`vkeys`**, **`[7,1,2,4]`** (highlighted in blue, Figure 4.3-c). The halfedge $h_{4,7}^*$ is defined by tail **`vkey 4`** and head **`vkey 7`**.

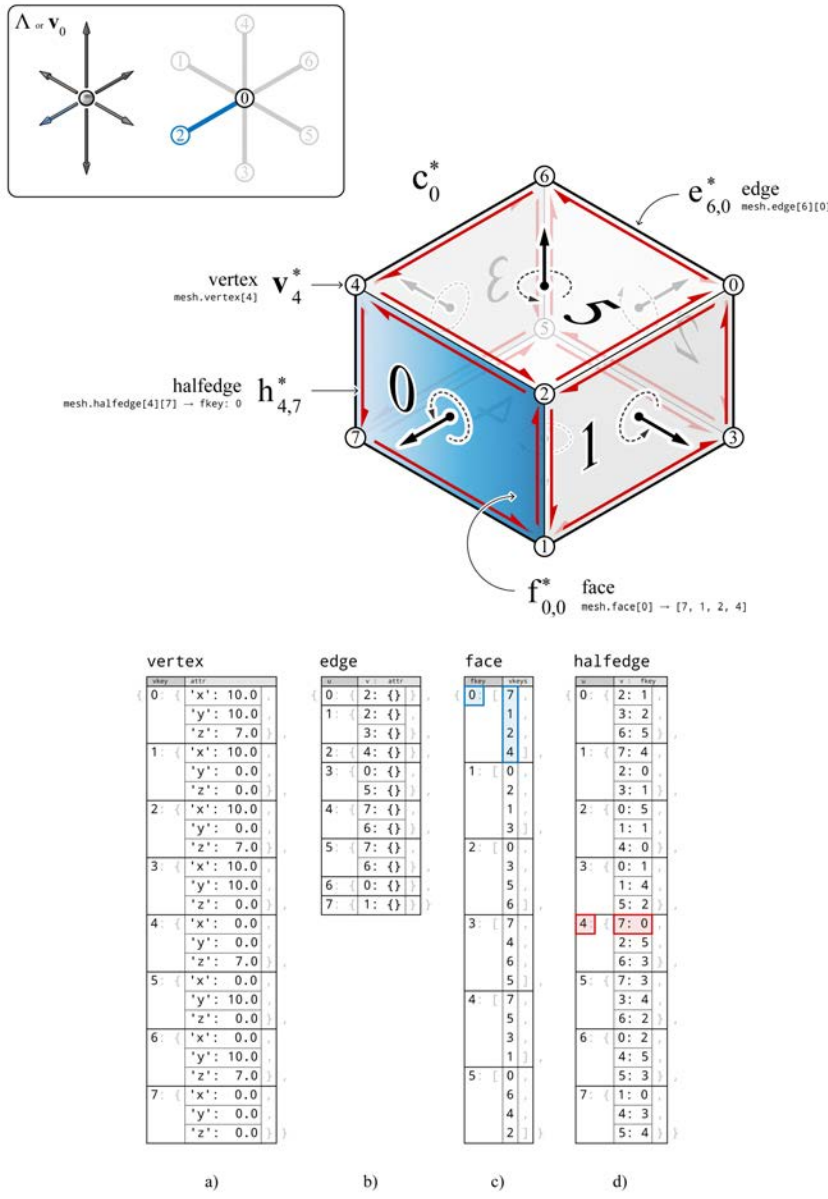


Figure 4.3: Polyhedral cell c_i^* represented as a mesh, a halfedge datastructure with four main layers of data: a) vertex; b) edge; c) face; and d) halfedge.

As **vkey 4** and **7** are two consecutive **vkeys** in the polygonal loop of $f^*_{i,0}$, the halfedge $h^*_{4,7}$ points to **fkey 0** (highlighted in red, Figure 4.3-d).

The direction of a face normal is defined by the right-hand rule, which is a common mnemonic for understanding the orientation conventions for vector cross products in three dimensions (i.e. the direction of the angular velocity vector in physics, the magnetic field in a coil of wire and the electric current in the wire in electromagnetics, the vorticity at any point in the field of flow of a fluid in continuum mechanics, etc.).

For example, the face normal direction of $f^*_{i,0}$ is pointing away from the centre of the cell. In general, the face normals of convex polyhedral cells are uniform (to be referred to as the polyhedral *cell direction*); the normals of all faces point either toward the centre of the cell (negative cell direction), or away from the centre of the polyhedron (positive cell direction) (Lee et al., 2016). For complex polyhedral cells which have complex faces, the face normal directions may not necessarily be uniform.

4.2.3 Duality

Maxwell used the principle of duality in projective geometry to express the reciprocal relationship between the form and force diagrams in the context of graphic statics (Harman, 2001). The 2D form and force diagrams have a dual relationship with each other; the points and lines of one diagram is mapped to the lines and points of the other. The mapping of one geometric object in one diagram to a different geometric object in the other diagram can be explained by Maxwell's observation that 2D form and force diagrams are projections of plane-faced three-dimensional polyhedrons (Maxwell, 1864, 1870).

The same relationship is true for any n-dimensional reciprocal form and force diagrams: n-dimensional reciprocal diagrams are projections of $(n+1)$-dimensional stress functions (Crapo and Whiteley, 1994; Konstantatou and McRobie, 2016). Based on this principle, three-dimensional form and force diagrams can be defined as projections of four-dimensional stress functions. The duality relationships between the elements of n-dimensional structures and the $(n+1)$-dimension of the corresponding stress functions are summarised in Figure 4.4.

The reciprocal diagrams of 2D structures have a "face dual" relationship, where the vertices one diagram corresponds to the faces of the other diagram. The reciprocal diagrams of 3D structures have a "cell dual" relationship, where the vertices of one diagram corresponds to the cells of the other diagram. A polyhedral cell and the corresponding form diagram have a cell dual relationship; the polyhedral cell and its faces correspond to a vertex and edges of the form diagram.

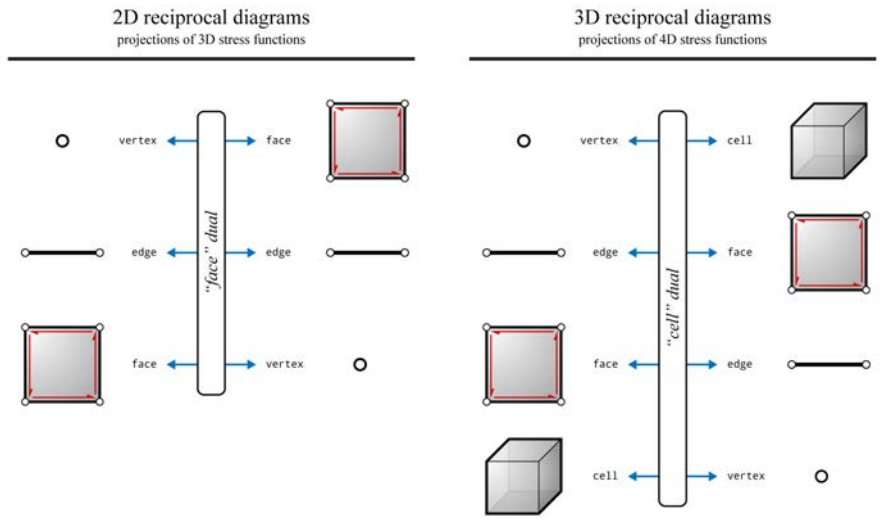


Figure 4.4: Summary of the duality relationships between the elements of n-dimensional structures and the $(n+1)$-dimension of the corresponding stress functions (force diagrams) (after Figure 1 of Konstantatou et al. (2018)).

4.2.4 Extended Gaussian Image

The geometry of a polyhedral cell, especially irregular and non-convex types, alone may not be sufficient for complex topological transformations. While there is a significant amount of research from numerous disciplines regarding convex and regular polyhedrons, non-convex polyhedrons have seen comparatively less development.

For both convex and non-convex polyhedrons, the dual polyhedral representations can be beneficial in providing a better understanding of their topological structure (Grünbaum, 2007). The *Extended Gaussian Image* (EGI) is an abstract dual representation of a polyhedron (Horn, 1984), and it can be particularly useful in representing, manipulating and transforming both the topology and the geometry of polyhedral cells.

An EGI is an unitised topological representation of a closed polyhedral surface or cell on a unit Gaussian sphere using its face orientations and areas. In EGI_i of polyhedral cell c_i^*, each j-th face $f_{i,j}^*$ is represented by a "point mass" $f_{i,j}'$ on the Gaussian sphere (Figure 4.5-c, e). Each point mass $f_{i,j}'$ can be located on the Gaussian sphere by placing the tail of the unitised normal $\hat{\mathbf{n}}_{i,j}^*$ of face $f_{i,j}^*$ at the centre of the sphere (Figure 4.5-b). The head of $\hat{\mathbf{n}}_{i,j}^*$ then lies on the surface of the sphere and represents a point mass of EGI_i. Each point mass is assigned a value that is equal to the area $A_{i,j}^*$ of $f_{i,j}^*$, or equivalently the magnitude of the corresponding force $\mathbf{f}_{i,j}$ (Figure 4.5-a).

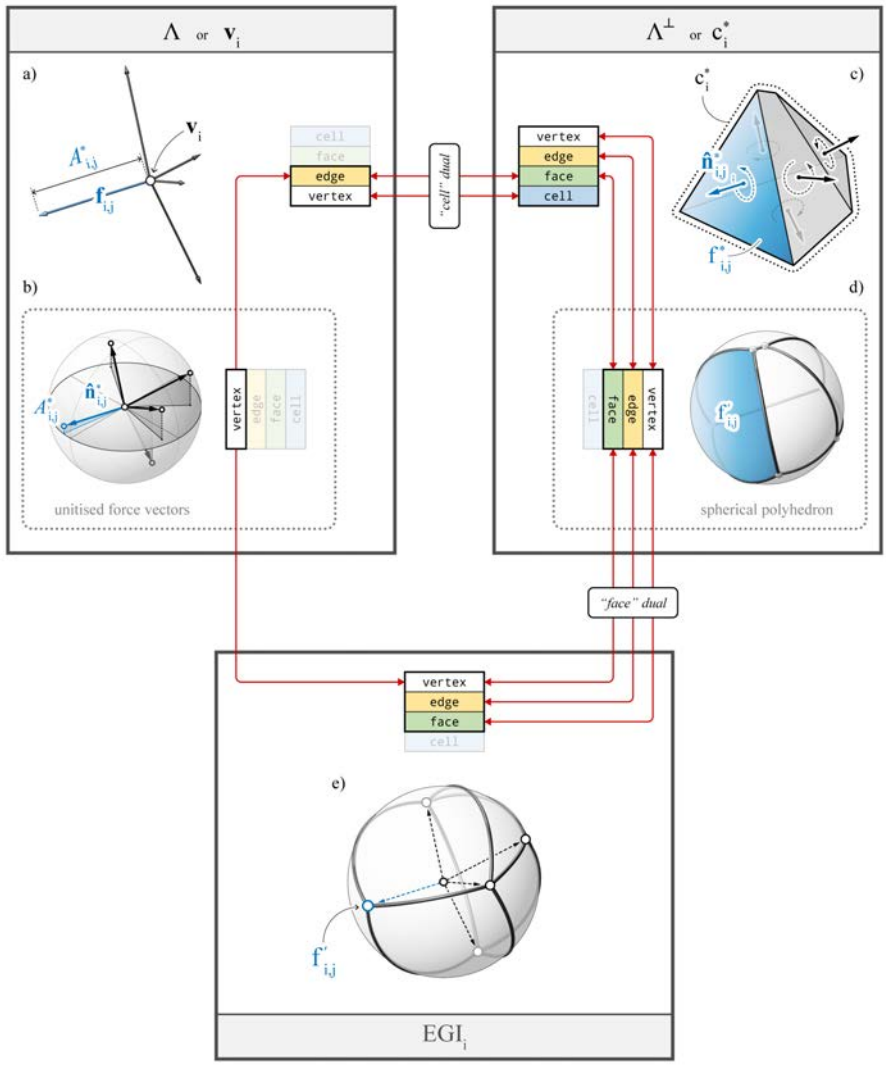


Figure 4.5: A graphical overview of the topological relationships between the form diagram Λ or $\mathbf{v}_i$, the polyhedral cell c_i^*, and the EGI: a) Λ or $\mathbf{v}_i$; b) unitised force vectors; c) polyhedral cell c_i^*; d) spherical polyhedron representation of c_i^*; and e) the EGI_i.

The point masses of an EGI describe the orientations and areas of the faces of a polyhedral cell. However, the point masses do not provide any of the face adjacency information of c_i^*. Two faces of c_i^* are adjacent if they share a common edge. Since the vertices and the faces of c_i^* are mapped to the face and the vertices of EGI_i, respectively, they are topological face duals of each other. By the definition introduced in Section 4.4, the edges of c_i^* are

then dual to the edges of EGI_i. An edge of EGI_i, which connects two point masses each representing a face of c_i^*, then represents the adjacency of the two faces of c_i^*. The edges of EGI are typically represented as arcs on the Gaussian sphere called "arcs of adjacency" or "adjacency arcs" (Moni, 1990). The EGI with point masses and adjacency arcs is the "dual image" (Roach and Wright, 1986) or the dual "spherical polyhedron" (Wenninger, 2012) of c_i^* (Figure 4.5-d). A spherical polyhedron is a set of arcs on a Gaussian sphere, each representing the projection of an edge of a polyhedron.

Although the conventional definition of an EGI only includes point masses, the adjacency arcs are essential in describing the complete topology of the corresponding polyhedron. Therefore, an EGI in this dissertation will consist of both point masses and adjacency arcs.

4.2.5 Types

For regular convex polyhedrons, various notations and methods of representation exist which attempt to compare and relate different types of polyhedrons, such as: the Schläfli symbol, which is a notation for defining regular convex polyhedrons whose faces are also regular n-sided polygons; the Coxeter–Dynkin diagram, which is a graph representation of the relationship between its constituent faces using symmetry and dihedral angles (Coxeter, 1973); and the Schlegel diagram, which is a planar projection of a polyhedron through a point beyond one of its faces (Schlegel, 1883).

However, these representations are mere notations and visualisations, which are established only for regular and highly specific convex polyhedrons. In 3D graphic statics, a force diagram may contain polyhedral cells that are not just convex, but also concave and complex. While convex polyhedrons are well-defined and well-studied, mathematical definitions of non-convex and irregular polyhedrons remains largely unresolved and unaddressed (Lakatos et al., 1976).

EGIs can be useful in visualising, classifying, understanding and processing the topological structures of different types of polyhedrons. Because an EGI itself has its own topological structure that is dual to the polyhedron, it can also be used to develop topological operations and geometric transformations of polyhedrons. This section provides an overview, descriptions and computational implementations of the three different types of polyhedrons that can frequently occur within the context of 3D graphic statics: convex, concave and complex.

4.2.5.1 Convex

This section introduces, defines and compares the different types of convex polyhedrons. For each type, its relevance and interpretation within the context of 3D graphic statics is discussed.

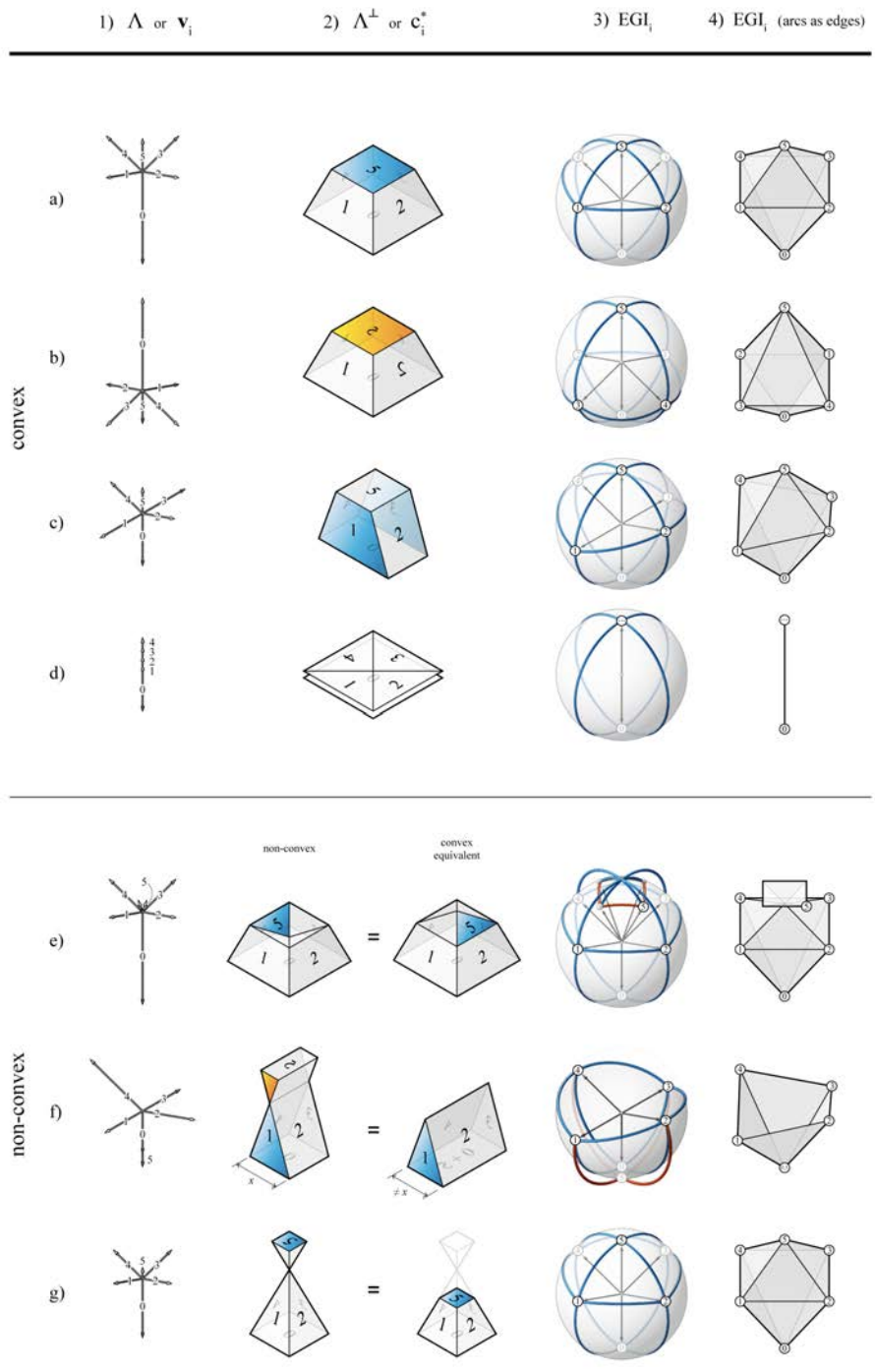


Figure 4.6: Various cell types and corresponding EGIs.

Regular convex

The definition of a "polyhedron" may vary depending on the discipline. The general agreement across numerous disciplines is that a polyhedron is a solid or a closed surface that can be described by its vertices and flat polygonal faces. A *convex* polyhedron is a special case where a segment connecting two points on any two faces always lies in the interior of the polyhedron (Lyusternik, 1963). In other words, a straight line can cut a convex polyhedron through a maximum of two points. There are two main ways to define a convex polyhedron: 1) as a convex hull of a finite set of points, also known as the vertex representation, V-representation or V-description; and 2) as an intersection of a finite number of half-spaces, also known as the half-space representation, H-representation or H-description (Grünbaum, 2003). In a H-representation, the half-spaces can be defined algebraically as a system of linear inequalities.

The face normal directions of a positive convex polyhedron point away from the cell's centre of mass (Figure 4.6-a2), while the face normal directions of a negative convex polyhedron point towards the cell's centre of mass (Figure 4.6-b2). The EGI of a convex polyhedron is also convex (Figure 4.6-a4, b4). The face normal directions of the EGI also match those of the corresponding polyhedron; the face normal directions of the EGI of a positive polyhedron point away from the origin of the EGI, while the face normal directions of the EGI of a negative polyhedron point towards the origin of the EGI.

In 3D graphic statics, convex polyhedral cells are the most basic and common building blocks of force diagrams. For example, funicular spatial structures such as domes, compression-only shells, tensile nets or branching structures all have corresponding force diagrams with only convex polyhedral cells. The modelling and processing of convex polyhedral cells is comparatively straightforward, and hence most of the initial explorations of 3D graphic statics are indeed based on transformations of convex polyhedral cells (Akbarzadeh et al., 2015b,a).

Prism

A *prism* or a prismatic polyhedron is a special case of convex polyhedron where two of the non-adjacent faces are parallel and congruent (faces 1 and 3 in Figure 4.6-c2), and all remaining faces are parallelograms (Kern and Bland, 1948). A right prism, is a special case of a prism where the top and bottom faces are not only identical but are orthogonal translations of each other, and all of the side faces are rectangles. As prisms are convex, the EGI of prisms are also convex (Figure 4.6-c4).

In 3D graphic statics, a prismatic polyhedral cell can be used to represent the equilibrium of a 3D node, but also 2D nodes where the congruent faces can be interpreted as two external forces that cancel each other out.

The interpretation of the equilibrium of a system of coplanar forces using prismatic polyhedral cells will be discussed in further detail in Section 4.2.7.2.

Flat

A *flat* polyhedron consists of faces which are all coplanar. A flat polyhedron with two faces is a dihedron, which is a polyhedron composed of only two flat polygonal faces that share the same set of edges. A dihedron is also referred to as a bihedron (Kántor, 2003), or a doubly covered polygon (O'Rourke, 2010). The EGI of a dihedron is a hosohedron, which is a beach-ball-like tessellation of lunes, or an area on a sphere bounded by two half great circles where each lune shares the same two antipodal points (Coxeter, 1973).

In 3D graphic statics, a flat polyhedral cell may represent the equilibrium of a system of parallel forces. For example, a flat polyhedral cell could represent the equilibrium of external forces where the applied loads are vertical and all of the reaction forces are also vertical (i.e. tied arches or tied domes). A flat polyhedral cell that is a dihedron would represent an equilibrium of forces where there is only one applied load and only one reaction force such as a vertically loaded column. However, in a larger structure, there may be multiple, parallel external forces and reactions. A flat polyhedral cell with only two faces will have an EGI that is a hosoderon with only two antipodal points, whereas a flat polyhedral cell with more than two faces as shown in Figure 4.6-d4, the EGI will have multiple coincident vertices; there are actually four vertices at the top of the EGI in Figure 4.6-d3, d4.

4.2.5.2 Non-convex

This section introduces, defines and compares the different types of non-convex polyhedrons. For each type of non-convex polyhedron, its relevance and interpretation within the context of 3D graphic statics is discussed.

Concave

A *concave* polyhedron contains at least two points on any two faces such that the line segment connecting the two points contains points that are outside of the polyhedron. In other words, a straight line can cut a concave polyhedron through more than two points. In 2D, convexity of a polygon can be checked by evaluating the internal angles formed by its edges at the vertices. A convex polygon will have all internal angles of less than π, while a concave polygon may have internal angles greater than π. Similarly in 3D, the dihedral angles between the pairs of faces at an edge can be evaluated to check for convexity. However, this would require the measuring of the dihedral angle between all possible pairs of a cell, which would be an overly

unnecessary and inefficient procedure. Another method of checking would be to cut the cell with a plane, then evaluate the convexity of the resulting polygonal shape of the section cut. This method would be considerably more exhaustive and inefficient, since the number of sections cuts required to verify a cell's convexity cannot clearly be defined. However, by evaluating the EGI of a polyhedron, which is constructed only from the face normals and connectivity of the polyhedron, the convexity can be checked easily. If the EGI is complex (if any faces of EGI are self-intersecting), the corresponding polyhedron is concave. At any non-convex vertex of a concave polyhedron, the Gaussian curvature is negative, and the vertices of the EGI are traversed in different directions compared to the non-convex vertices; the polyhedron itself is one possible geometric realisation and is not unique to the EGI (Xu and Suk, 1995). The overlapping arcs in Figure 4.6-e3 are shown with an exaggerated offset for clarity.

In 3D graphic statics, concave polyhedral cells are essentially an alternate representation of an equivalent convex polyhedral cell. By rearranging the faces and changing the face adjacencies, an equivalent convex polyhedral cell representing the same force equilibrium of a concave polyhedral cell can be constructed (Figure 4.6-e2).

Complex

A *complex* polyhedron consists of faces that are self-intersecting. Additionally, the face normal directions are no longer necessarily uniform; some may point away while some point towards the cell's centre of mass (Figure 4.6-f2). The normal of face 5 is pointing towards the polyhedron's centre of mass, while all other face normals are pointing away from the centre of the cell. EGI of a complex polyhedron may have open hemispheres (Figure 4.6-f3) where not all of the Gaussian sphere is covered with spherical faces. The overlapping arcs in Figure 4.6-f3 are shown with an exaggerated offset for clarity. Symmetric complex polyhedrons are special cases of complex cells where self-intersections of some of the face occur, but all of the face normals are still uniform (Figure 4.6-f3).

In 3D graphic statics, complex polyhedral cells represent indeterminate states of equilibrium. As such, all complex polyhedral cells can be represented with an equivalent convex polyhedral cell. In Figure 4.6-f2, faces and 0 and 5 are pointing in the same direction. Therefore, if faces 0 and 5 were consolidated into one face, and convex polyhedral cell that represents the equivalent net force equilibrium can be constructed while maintaining the same areas for all the other face. This simplified convex polyhedral cell representation can be useful in certain applications when a simplified visualisation of the force diagram may be desired. Complex polyhedral cells can be used to illustrate possible equilibrium states of hyperstatic structures (Kilian and Ochsendorf, 2005; Block, 2005; Van Mele et al., 2012).

Other

The aforementioned typologies are generalisations, and there are many other irregular types of polyhedrons. Some of these additional types are: non-orientable polyhedral surfaces, such as the Klein bottle (Alling and Greenleaf, 1969) or the Boy's surface (Boy, 1903) (Figure 4.7-a); skew apeirohedron or infinite skew polyhedron (Garner, 1967); or cuploids or star polyhedrons which have repetitive non-convex elements or features (Conway et al., 2008, pp. 404-408) (Figure 4.7-c). These special types of polyhedrons can be sufficiently constructed and represented as a mesh. Although these types of polyhedrons may appear in some cases, these polyhedrons that have high geometric complexity as part of form diagrams are not generally addressed or investigated in detail.

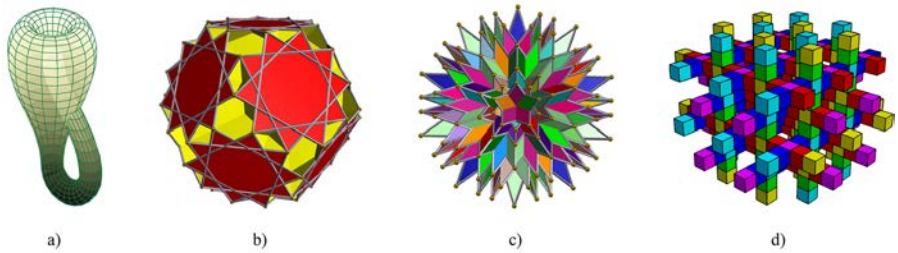


Figure 4.7: Special types of polyhedral cells: a) the Klein bottle; types of star polyhedra b) the "great dodecicosahedron" and c) the "great grand stellated polydodecahedron"; and d) pseudo-platonic cubic polyhedron, which is a regular skew apeirohedra; (Images: a) Wikimedia Commons; b) Robert Webb; c), d) Wikimedia Commons)

4.2.6 Faces

This section gives a detailed description of the faces of a polyhedral cell. It also describes how the oriented area and normal of a face are computed.

4.2.6.1 Polygonal area

As illustrated in the previous section, there are numerous types of polyhedral cells. Different types of cells are also made up of faces that vary in type; the constituent faces may be convex, concave, complex, both concave and complex, or "gauche" (non-planar) (McRobie, 2016a, 2017a). Because the relationship between polyhedral form and force diagrams is inherently dependent on the face areas and orientations, correctly computing the areas and normals of the faces are necessary. While the area and normal of any general convex polygon can be computed directly using triangulation and trigonometry, the procedure is not as straightforward for concave and complex polygons. In most computational geometry applications, the numerical value of the area is sufficient. However, in the context of 3D graphic statics, a face of a polyhedral cell represents a force vector. Not only

the numerical value of the *oriented area* of a face need to be computed, but also the direction or the *oriented normal* of the face. Computing the area of the polygonal loop that defines a face can be interpreted in different ways depending on the discipline and the problem at hand.

Every face of a polyhedral cell has a polygonal loop or a winding direction. A face can be decomposed into vertices and a series of directed edges connecting them. This series of edges are equivalent to the halfedges of the mesh datastruture (Figure 4.8-a). For a convex face, a simple procedure for computing the area would be to triangulate the polygon into sub-triangles using each of the edges and any point that is inside the polygon (Figure 4.8-c). Then, the areas of the sub-triangles can be computed and summed. However, as shown in Figure 4.8-d, this simple triangulation may result in overlapping sub-triangles for concave and complex cells.

A method of interpreting the area of a complex face is the "even-odd rule" method. This rule essentially counts or discounts the sub-faces of the face in an alternating matter; one of any pair of adjacent regions along an edge counts towards the area, while the other does not (Figure 4.8-e). The even-odd pattern can be determined by algorithms such as the Bentley-Ottman algorithm (Bentley and Ottmann, 1979), which sweeps a line across the entire polygon and determines the "interior" and "exterior" of the polygon by ordering and evaluating all intersection points. While this interpretation works for most cases in computing the nominal area of a self-intersecting face, it does not take into account the directions of faces, and therefore the oriented normal of the polygon cannot be computed correctly. In addition, such exhaustive procedure would be unnecessarily inefficient for force diagrams with multiple complex polyhedral cells, each consisting of several complex faces.

In order to determine the oriented area and normal of a face, the winding direction of each of the sub-regions of the face need to be taken into account (Figure 4.8-f). The winding direction of each sub-region can be determined by splitting up each of the edges at every self-intersection, while maintaining the parent edge directions. The split edges can be stitched back together to determine the new regions such that every split edge of the region starts where the previous one ends. This procedure can start with any split edge, and follow the next edge with the same direction. If there are more than one split edges that are available, convexity can be enforced such that the resulting region is not concave or complex. Convexity of the region can be enforced by maintaining a uniform direction of the cross product of every two consecutive split edges. This procedure continues until every split edge has been accounted for.

Each sub-region then has its own local winding direction, which can be used to determine whether the region contributes positively or negatively to the total area of the face. The positive areas (the sub-regions with

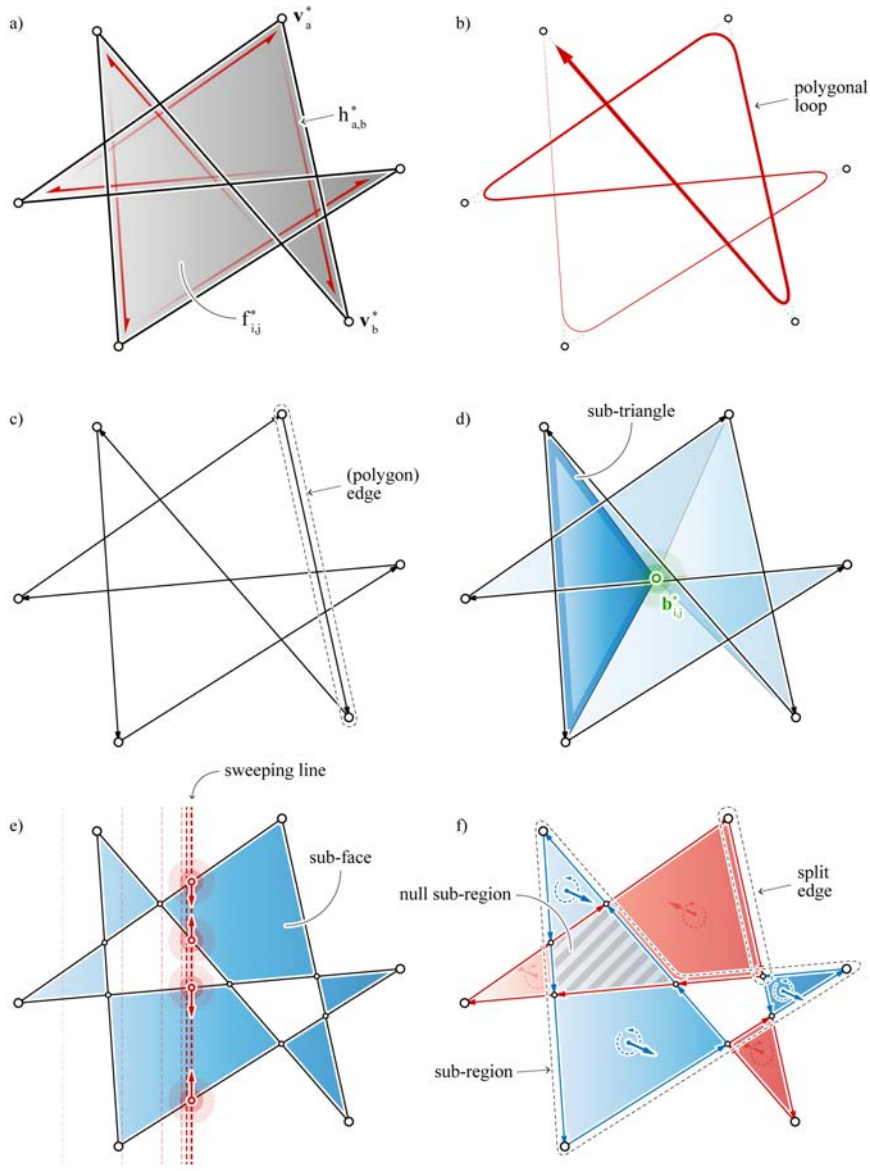


Figure 4.8: Different interpretations of the area of a complex face that is self-intersecting; a) a face of a polyhedral cell and its halfedges; b) the halfedges complete the polygonal loop that defines the winding direction of the face; c) polygonal loop broken up into individual edges; d) sub-triangles formed by each of the edges and the centroid, $\mathbf{b}^*_{i,j}$; e) the "even-odd" rule interpretation, where the sub-faces are counted in an alternating sequence; and f) the oriented area of the face, where the winding direction of each sub-region is taken into account.

counterclockwise winding directions and normals pointing away from the page) are shown in blue, and the negative areas (the sub-regions with clockwise winding directions and normals pointing into the page) are shown in red in 4.8-f). The total oriented area of the face is then the sum of the oriented areas of all the sub-regions. A positive oriented area will mean the oriented normal is pointing away from the page, and a negative oriented area will mean that the oriented normal of the face is pointing into the page.

4.2.6.2 Oriented area

The split-edge method works for most polygons. However, splitting edges, re-ordering split edges and finding sub-regions with consistent edge directions is rather an inefficient method for force diagrams with multiple polyhedral complex cells. For non-planar and self-overlapping faces, the edges may not necessarily intersect with one another. In addition, it is not always guaranteed that all found regions are convex.

If any polygonal loop of a face is interpreted as the planar projection of a face that makes up a fictitious polyhedral cell, the procedure for computing the oriented normal and area of that face becomes much simpler and more elegant. Rather than computing the area of a face by deconstructing it into smaller pieces and then reassembling them into sub-regions, the oriented area of a face can be computed by summing the oriented normals of all the other faces of the fictitious polyhedral cell. This is always true, regardless of the planarity, concavity or complexity of the face, because the sum of the oriented or weighted normals of all the faces (each representing a force vector) of a polyhedral cell is always equal to zero (equilibrium) by definition.

Consider the pentagon-shaped face $f^*_{i,j}$ in Figure 4.9-a1. This face can be interpreted as a projection of the bottom face of a pentagonal pyramid (Figure 4.9-a2). The simplest way to construct this pentagonal pyramid is to use the centroid of the face $\mathbf{b}^*_{i,j}$ (although any arbitrary point in space can be chosen), and moving it up in the $+z$ direction (towards the top of the page) to form the peak of the pyramid. By using each directed edge of $f^*_{i,j}$ and $\mathbf{b}^{*(z)}_{i,j}$, the missing triangular faces can be constructed to complete the fictitious polyhedral cell (Figure 4.9-a3).

The area of $f^*_{i,j}$ then is the inverse of the sum of the oriented normals of the triangular faces. To make the formulation even simpler, a flattened cell can be used; instead of using $\mathbf{b}^{*(z)}_{i,j}$, centroid $\mathbf{b}^*_{i,j}$ itself can be used. Regardless of where the peak of the fictitious cell is, the oriented normal of $f^*_{i,j}$ is always the same. In this particular convex case, the oriented normal of $f^*_{i,j}$ is in the $-z$ direction (towards the bottom of the page). The amplitude of the oriented normal of $\mathbf{f}_{i,j}$, which is the sum of the oriented normals of the five triangular faces pointing in the $+z$ direction, is the nominal area $A^*_{i,j}$ (Figure 4.9-a4).

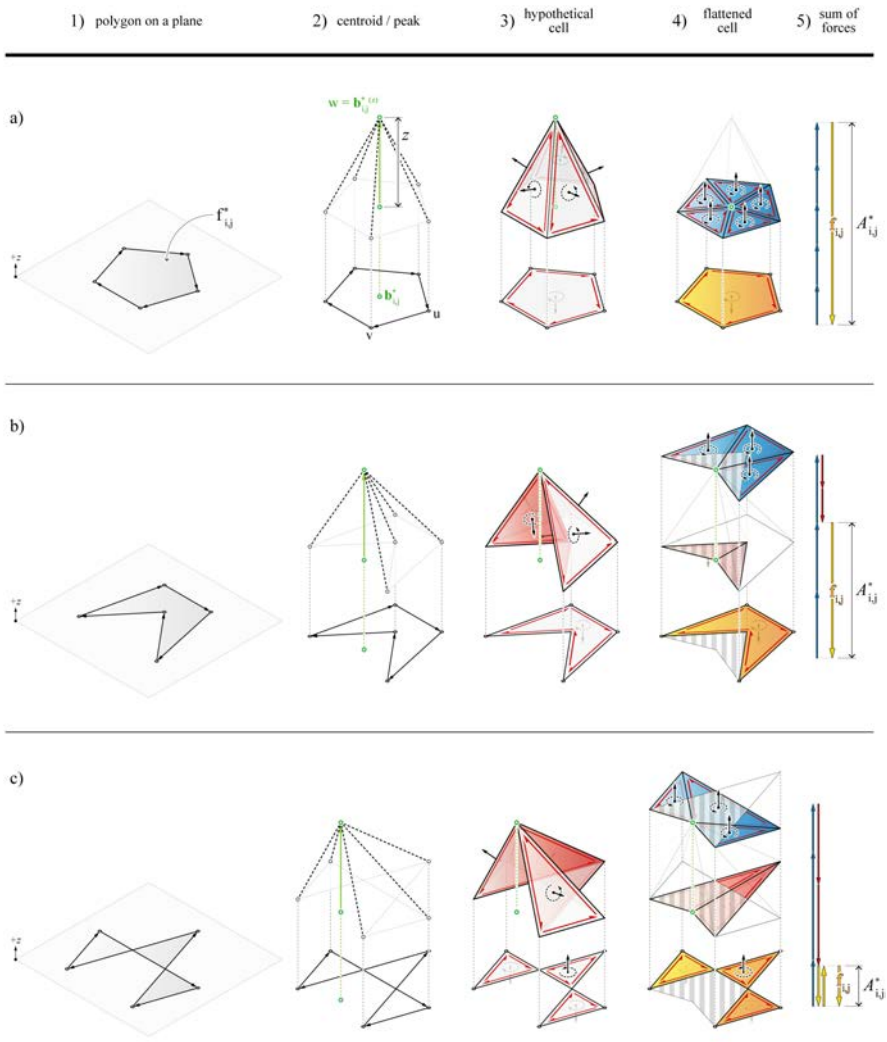


Figure 4.9: Computing the oriented area and normal of face of a polyhedral cell using the sum of the oriented normals of all of its sub-triangles of: a) a convex; b) a concave; and c) a complex face.

For a concave face shown in Figure 4.9-b1, the procedure is the same. Because of the concavity of the central vertex, the oriented normals of two of the triangular faces are pointing in the $-z$ direction, while the other three are pointing in the $+z$ direction (Figure 4.9-b3). The region where the positive and negative faces overlap naturally cancels each other out, which is equivalent to the sum of the weighted normals of all the triangular faces (Figure 4.9-b4).

For a complex face shown in Figure 4.9-c1, the fictitious cell has multiple enclosures, but the procedure remains the same because all of the missing faces are still triangular. With complex faces, there are multiple positive and negative sub-regions (Figure 4.9-c4). Once again, by summing the oriented normals of the sub-regions, or equivalently the force vectors corresponding to each of the missing triangular faces, the oriented normal of the face can be computed (Figure 4.9-c5). The magnitude of the oriented normal is the oriented area of the face.

This method of computing the oriented area of a face presented in this section is equivalent to the bivector description of polyhedral face areas based on Clifford algebra (McRobie, 2017a, pp. 3-4). The explicit directional information of the oriented face normals presented in this section will prove to be essential for the development of numerous operations and algorithms in later sections.

4.2.7 Interpretation

This section discusses how the oriented face normals of a polyhedral cell can be used for the interpretation of the corresponding forces in the form diagram. Also presented in this section is how prismatic polyhedral cells can be used to represent the equilibrium of a coplanar system of forces.

4.2.7.1 Compression and tension

Once the oriented normals of the faces have been computed, the interpretation of the corresponding external forces or the internal member forces in the form diagram can be made. The oriented normal of a face in the force diagram can be compared to the orientation of the corresponding edge in the form diagram to determine whether or not the internal force in that edge is in tension or compression.

A force $\mathbf{f}_{i,j}$ acting on node $\mathbf{v}_i$ can be either a compression force or a tension force; it is a compression force if it is pushing onto the node, or a tension force if it is pulling away from the node. In 3D graphic statics, the direction of $\mathbf{f}_{i,j}$ is the same as the direction of the normal $\hat{\mathbf{n}}^*_{i,j}$ of the corresponding face of the polyhedral cell. Therefore, a single force polyhedron can represent the equilibrium of various combinations of compression and tension forces (Akbarzadeh, 2016). For example in Figure 4.10-b1 and b2, the directions of the external forces are in the same orientations and directions as the corresponding face normals in c^*_i (Figure 4.10-a). However, the external forces are interpreted differently; $\mathbf{f}_{i,4}$ of Λ case 1 is in tension, while $\mathbf{f}_{i,4}$ of Λ case 2 is in compression.

The interpretation the j-th external force or the internal force of the j-th structural member at node $\mathbf{v}_i$ as either compression or tension, can be made

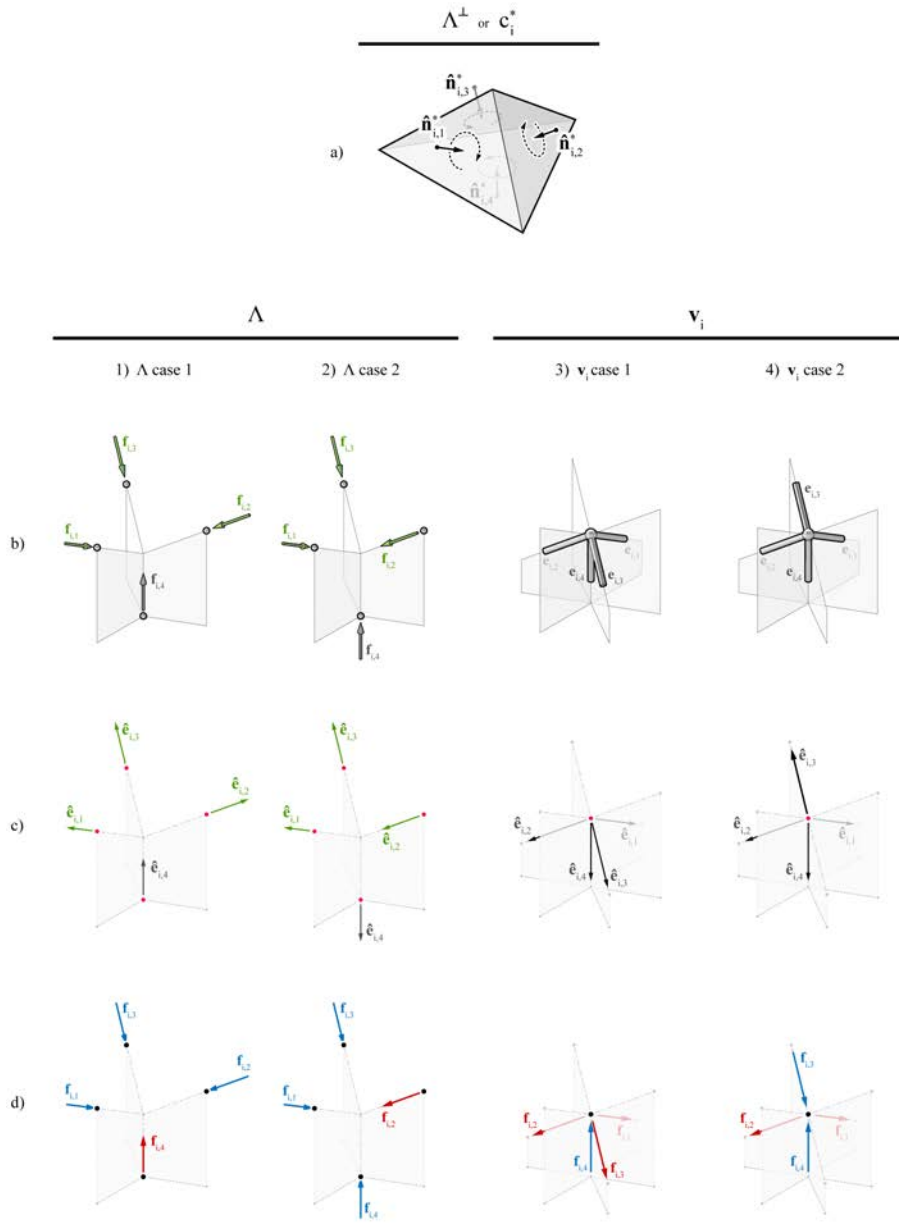


Figure 4.10: Multiple configurations of compression and tension elements are possible for a polyhedral cell: a) $\Lambda^{\perp}$ or polyhedral cell c_i^* of node $\mathbf{v}_i$; b) orientation of the members or external forces as unitised vectors $\hat{\mathbf{e}}_{i,j}$; and c) the direction and type of forces in the members.

by comparing the face normal $\hat{\mathbf{n}}^*_{i,j}$ and the orientation of the corresponding member (Lee et al., 2016). Locally defined for each node $\mathbf{v}_i$, the orientation of the j-th external force or member can be represented by a unitised edge vector $\hat{\mathbf{e}}_{i,j}$ with the tail of the vector at $\mathbf{v}_i$ (Figure 4.10-c). If $\hat{\mathbf{n}}^*_{i,j}$ and $\hat{\mathbf{e}}_{i,j}$ are pointing in the same direction, the force in the corresponding member is interpreted as positive, so in tension; if the vectors are in opposite directions, the force in the corresponding member is interpreted as negative, so in compression. The compression and tension forces are shown in blue and red, respectively, in Figure 4.10-d.

4.2.7.2 Two-dimensional equilibrium

Two-dimensional equilibrium can be interpreted as a special case of three-dimensional equilibrium; the equilibrium of a two-dimensional or coplanar system of forces can be represented by a prismatic polyhedral cell. Consider the node $\mathbf{v}_i$ of 2D form diagram G in Figure 4.11-a. This node is equilibrium and has three members in compression. In the Cremona representation of 2D reciprocal diagrams, edges of the force diagram $G^{\parallel}$ and the corresponding edges in the form diagram G are parallel to one another (Figure 4.11-a). In the Maxwell representation of 2D reciprocal diagrams, edges of the force diagram $G^{\perp}$ and the corresponding edges in the form diagram G are perpendicular to one another (Figure 4.11-b). Maxwell representation is simply the Cremona representation rotated 90 degrees on the plane.

A force polygon of $G^{\perp}$ can be interpreted as the planar projection or the polygonal loop of a face of a prismatic polyhedral cell. In this particular example, the force polygon in $G^{\perp}$ is a planar projection of face $f^*_{i,4}$ (Figure 4.11-c). $e^*_{i,1}$ of $G^{\perp}$ can then be interpreted as the projection of halfedge $h^*_{a,b}$ of face $f^*_{i,4}$ of c^*_i. Face $f^*_{i,4}$ then represents the force $\mathbf{f}_{i,4}$, acting along the viewing axis onto $\mathbf{v}_i$. Since this polyhedral cell is prismatic, $f^*_{i,4}$ and $f^*_{i,5}$ are parallel in their orientations and equal in areas. Therefore, the two corresponding forces $\mathbf{f}_{i,4}$ and $\mathbf{f}_{i,5}$ in $\mathbf{v}_i$ are equal in magnitudes but opposite in directions. The sum of the forces acting out of viewing axis is zero.

The length of the halfedges of $f^*_{i,4}$ are proportional to the magnitudes of the forces in the corresponding members, subject to a scale factor, ζ. ζ is the the vertical extrusion length of the prismatic polyhedral cell. Finally, $\mathbf{f}_{i,1}$ then corresponds to $f^*_{i,1}$, where $\hat{\mathbf{n}}^*_{i,1}$ is the direction and $\zeta \cdot h^*_{a,b}$ is the magnitude of the force (Figure 4.11-d).

This method of interpreting 2D equilibrium of forces with prismatic polyhedral cells is similar to the one described in McRobie and Williams (2018, p.43-44) using stress functions, where the authors point out that: "the 3D Maxwell–Rankine theory thus contains the standard Maxwell theory for 2D reciprocal trusses ... this feature may prove useful in 3D, as many 3D structures contain 2D substructures." Indeed, by using prismatic

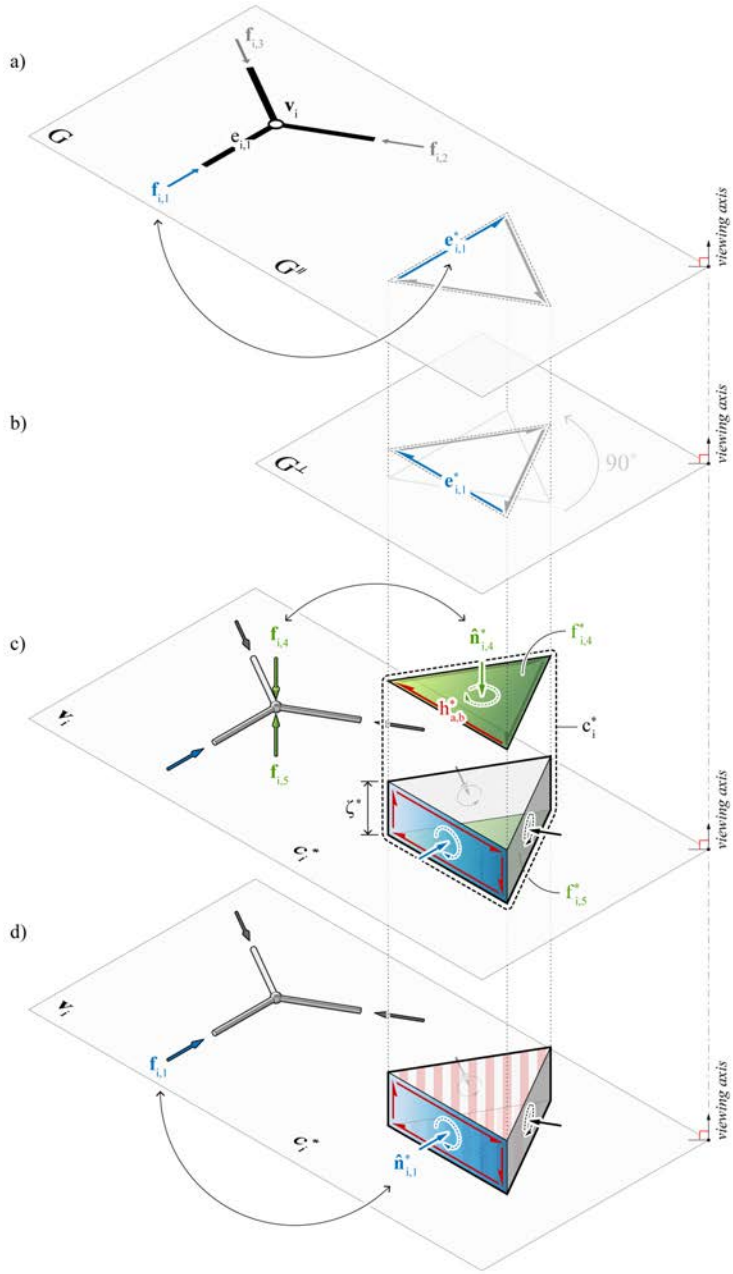


Figure 4.11: Interpretation of 2D reciprocal diagrams using prismatic polyhedral cells: a) 2D form diagram, G, and Cremona representation of the 2D force diagram, $G^{\parallel}$; b) Maxwell representation of the 2D force diagram, $G^{\perp}$; c) node $\mathbf{v}_i$ of the structure, and the corresponding polyhedral cell c_i^*; and d) the two horizontal faces of c_i^* shown as virtual faces, while the other vertical faces represent the equilibrium of the internal member forces at $\mathbf{v}_i$.

polyhedral cells as described in this section, the techniques developed in this dissertation for 3D structures can also be applied to 2D nodes or structures. Prismatic polyhedral cells can also be useful in explaining some of the important polyhedral concepts without the visual clutter. For various examples and auxiliary diagrams throughout the dissertation, prismatic cells will be used for simplicity and clarity.

4.2.8 Operations

In most computational graphic statics applications in 2D or 2.5D, geometric manipulations of form and force diagrams are simple and straightforward using a mouse and a computer monitor. Because the diagrams are coplanar on a single viewing plane, one can simply click and move the vertices of the force diagram to observe the consequential effects on the form diagram in real-time. Moving the vertices of a 2D force diagram changes both the magnitudes and orientations of the corresponding external forces or members in the form diagram.

Geometric manipulations of polyhedral cells are not as straightforward or intuitive without the fixed viewing plane of 2D applications or the projection plane of 2.5D applications. Moving the vertices of a polyhedral cell changes its geometry, but it is not immediately clear to the user how much effect the geometric transformation has on the new distribution or orientations of forces. Vertex translation in 3D space also requires meaningful geometric guides or constraints that are based on the local geometry of the polyhedral cell to avoid arbitrary or counterproductive transformations. Furthermore, the translation of vertices could also cause some of the faces of the polyhedral cell to become non-planar. For manipulating the geometry of polyhedral cells while enforcing the planarity constraints of the faces, vertex translations are simply not sufficient enough. Instead, a constrained translation and rotation of the faces are needed.

In order to change the force distribution of a polyhedral cell while maintaining the initial face orientations, a face can be pulled along its normal vector. For the simple cell shown in Figure 4.12-a, all four vertices of face $f^*_{i,j}$ are 3-valent, which makes the face pull operation simple. A plane can be defined by normal vector and a point, which determine the orientation and location of the plane, respectively. The plane $P^*_{i,j}$ which $f^*_{i,j}$ lies on can be defined by the centroid $\mathbf{b}^*_{i,j}$ and the normal vector $\hat{\mathbf{n}}^*_{i,j}$. Suppose that $f^*_{i,j}$ is being pulled in the $+z$ direction and relocated from its initial position of $\mathbf{b}^*_{i,j}$ to $\mathbf{b}^{*(z)}_{i,j}$. Since all of the vertices of $f^*_{i,j}$ are 3-valent, there is only one trailing edge per vertex. For example, $e^*_{a,b}$ is a trailing edge of $\mathbf{v}^*_{i,b}$. The new position of the b-th vertex of $f^{*(z)}_{i,j}$ can be computed by intersecting the trailing edge $e^*_{a,b}$ with the newly located plane $P^{*(z)}_{i,j}$.

In order to change the orientation of a particular force while maintaining the initial orientations of all the other forces, the face can be tilted using one of its constituent edges as the tilt axis. In the example shown in Figure 4.12-b, edge $e^*_{b,c}$ is chosen as the tilt axis. This axis is then located to the centroid $\mathbf{b}^*_{i,j}$, around which $f^*_{i,j}$ can be tilted with the angle of θ.

If a face contains vertices that have valencies of four or more (more than one trailing edge), the pulling or tilting of the axis will result in faces that are no longer in their original orientations or possibly non-planar. Consider the polyhedral cell shown in Figure 4.13-b, where $f^*_{i,2}$ is being pulled along its normal. Vertex $\mathbf{v}^*_{11}$ of this cell has a valency of four and it has two trailing edges: $e^*_{0,11}$ and $e^*_{4,11}$. This also means that the dual spherical face of $\mathbf{v}^*_{11}$ in the corresponding EGI has four edges (Figure 4.13-a). Because there are two trailing edges at $\mathbf{v}^*_{11}$, intersecting those two edges with the plane at the new location of $f^{*(z)}_{i,2}$ results in an extra vertex (shown as green vertices in Figure 4.13-c). The eventual $f^*_{i,2}$ needs to incorporate these extra vertices resulting from the extra trailing edges, which means the topology of the face, and therefore the polyhedral cell, needs to be updated prior to the operation.

The topological transformation of a cell can be guided and significantly simplified by the EGI. Because the two edges at $\mathbf{v}^*_{11}$ create an extra vertex and therefore an additional edge, $\mathbf{v}^*_{11}$ needs to be split into two vertices. In addition, these two vertices also need to be ordered in a specific way such that the winding directions of all the faces are maintained. If this procedure were to be done using just the geometry and topology of the cell itself, it would require additional steps to sort and order the vertices in space, which would quickly become tedious with multiple vertices that may have several trailing edges.

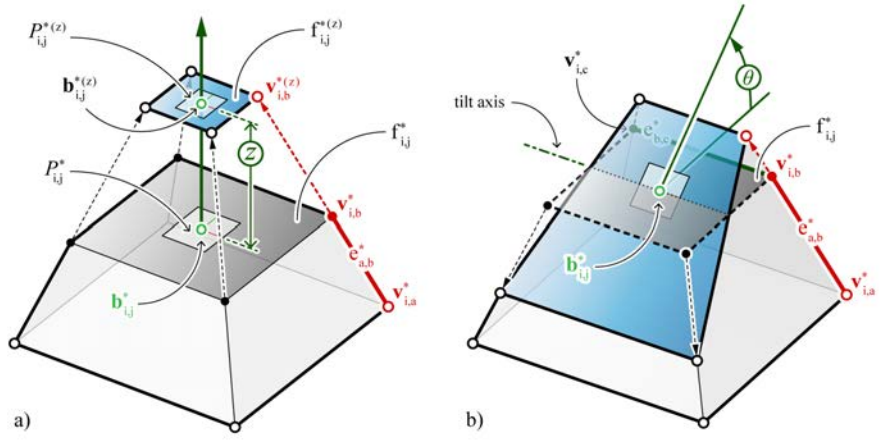


Figure 4.12: Constrained face operations on a simple polyhedral cell: a) pulling face $f^*_{i,j}$ in the $+z$ direction; and b) tilting face $f^*_{i,j}$, using the edge $e^*_{b,c}$ as the tilt axis.

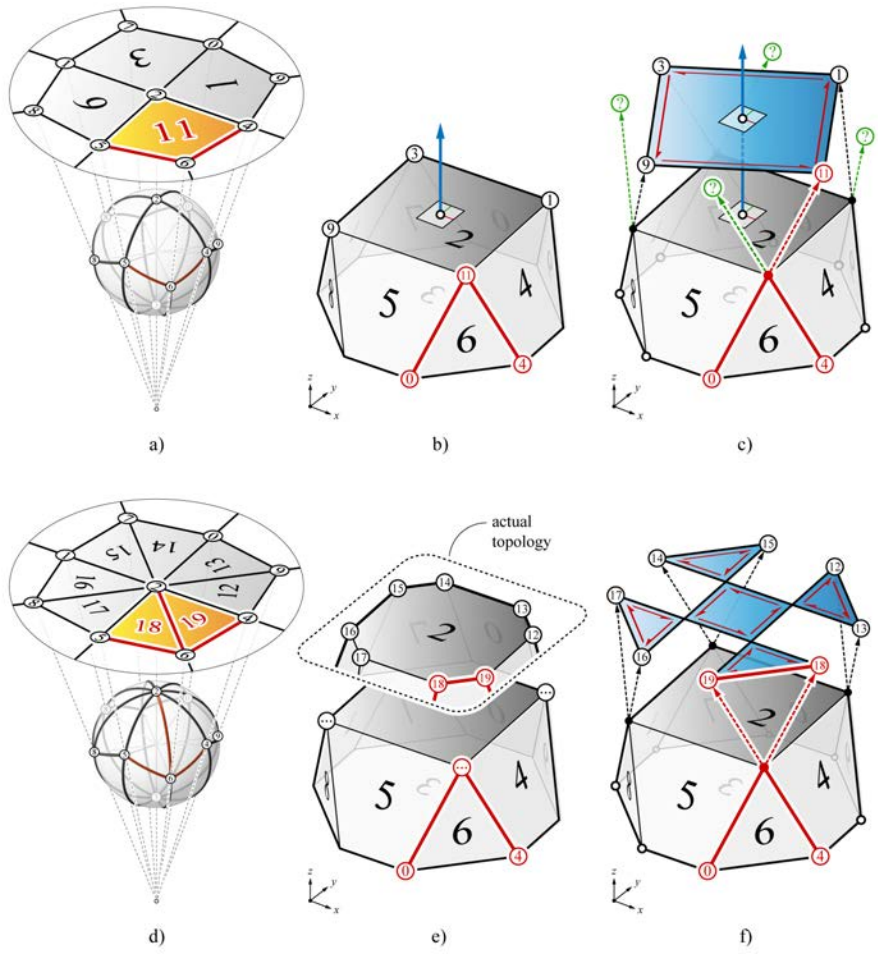


Figure 4.13: Faces with high-valent vertices require those vertices to be split before the face can be pulled or tilted: a) stereographic projection of vertex 2 of the EGI that corresponds to $f^*_{i,2}$ with 4-valent vertices; b) the corresponding polyhedral cell; c) the additional vertices (shown in green) created by the extra trailing edges; d) stereographic projection of vertex 2 of the EGI, with its neighbouring spherical faces now split in two; e) the corresponding polyhedral cell with the additional vertices and edges, and the new topology; and f) the resulting polyhedral cell.

The splitting of a cell vertex is topologically equivalent to splitting the corresponding spherical face of the EGI. By pulling $f^*_{i,2}$ in the $+z$ direction, an edge will be created between $f^*_{i,2}$ and $f^*_{i,6}$. This means that the vertices 2 and 6 in the EGI, which respectively represent $f^*_{i,2}$ and $f^*_{i,6}$ will need to be connected with a new edge. Using this edge, face 11 can be split into two new faces, 18 and 19 (Figure 4.13-d). The procedure can be repeated for any other vertices that have multiple neighbouring edges. Once all the required spherical faces have been split, the topology of the cell can be updated

by removing the old vertices and adding new vertices and edges. Using the topology of the spherical faces around vertex 2 in the EGI, the correct ordering of the new vertices of $f^*_{i,2}$ can be determined. Any new vertices will initially be placed at the same location as the parent vertex, although the topological structure is different from what the geometry may suggest (Figure 4.13-e). With the updated topology of the polyhedral cell, the face can be pulled or tilted with the same procedure described in Figure 4.12 (Figure 4.13-f).

4.2.9 Construction

So far, the polyhedral cells have been assumed to be an existing condition. However, in a typical design workflow, the only known information during early stages of design may be the boundary condition information such as the locations, magnitudes and orientations of the external forces. This subsection presents a polyhedral cell construction method using the EGI, which computes the geometry of the polyhedral cell from the orientation and magnitudes of spatial forces in equilibrium.

4.2.9.1 Background

Computation of polyhedral geometry, or more commonly known as "polyhedral reconstruction," is a well-researched topic for a variety of applications in many disciplines, such as computer vision, computational geometry and combinatorics. In most applications, the word "reconstruction" is typically used instead of "construction" because the objective is to retrieve or reconstruct the polyhedral geometry from partial information about the polyhedra (i.e. from projected images, vertex locations, edge lengths, face geometries, face normals, face areas, dihedral angles etc.) (Demaine and O'Rourke, 2007). In this dissertation, the word "construction" will be used to refer to these procedures that compute polyhedral cell geometries.

The polyhedral construction method that is most relevant to 3D graphic statics is the one based on face normals and area information. The theory of polyhedral construction from its face normals and areas, or modifying the polyhedral geometry with target face areas, originates from Minkowski's theorem (Minkowski, 1897). Alexandrov's (2005) interpretation of the theorem is recited below, with modified notations to stay consistent with the nomenclature used in this paper:

> If $\hat{\mathbf{n}}_0$, ... ,$\hat{\mathbf{n}}_\mathrm{m}$ are non-coplanar unit vectors and A_0, ... ,A_m are positive numbers such that $\sum_{\mathrm{j}=0}^{\mathrm{m}} A_\mathrm{j} \cdot \hat{\mathbf{n}}_\mathrm{j} = 0$, then there exists a closed convex polyhedron whose faces have outward normals $\hat{\mathbf{n}}_\mathrm{j}$ and areas A_j, [... with uniqueness up to translation].

Although proofs for this existence theorem can be found in numerous texts in the literature (Grünbaum, 2003; Alexandrov, 2005; O'Rourke, 2011), the construction procedure was never explicitly mentioned or developed in detail. It was not until the 1980s that this problem was revisited with adequate computational tools.

Ikeuchi (1981) first proposed a technique by using a constrained minimisation procedure, followed by Little's (1983) iterative minimisation solver using the EGI, which is a topological representation of a surface or a polyhedral object on a unit sphere (Horn, 1984). Moni (1990) added another layer to the EGI-based technique by using zero-area faces to address indeterminate face adjacencies. Xu and Suk (1995) introduced hierarchical EGI to reconstruct concave polyhedra. In these implementations, while robust in their theory and setup, the methods were demonstrated on only a few simple examples. The general computational complexity and hardness of this construction problem was addressed by Gritzmann and Hufnagel (1999).

Especially with EGI-based methods, the complexity and diversity of polyhedrons that can be reconstructed was not demonstrated through a large sampling of convincing examples. Furthermore, the ultimate goal of these methods is to simply demonstrate the improved efficiency over its predecessors, rather than manipulate or interact with the resulting geometry of the polyhedra. The designer's ability to customise the method and control the computed geometry of the polyhedrons is not addressed, which is crucial in an interactive design environment for architecture and structural design.

In a more numerical approach, Lachand-Robert and Oudet (2005) presented a variant of a convex hull method that could construct convex bodies with more than 1000 given face normals and areas. Such powerful numerical methods are necessary for large optimisation problems in various engineering applications. However, in the context of 3D graphic statics applications, individual polyhedral cells have relatively low number of faces; a structure would typically not have nodes where more than six members come together. Instead, structures are typically a large network of simple cells. Rather than a powerful solver that can reconstruct a single cell with a large number of faces, a flexible and interactive setup that can ultimately control a large network of simple cells is needed.

4.2.9.2 From force vectors to EGI

Previous work in polyhedral construction have been generally based on optimisation schemes that seek to maximise the processing efficiency or the output capacity of the method itself. Therefore, these methods are not ideal for a design environment where the geometry of the polyhedral cells may require constant interaction and change.

As described in Section 4.2.4, the construction of EGI from a set of equilibrated forces is a straightforward process. Once the point masses have been added to the EGI, the adjacency arcs that define the face adjacencies of the polyhedral cells can be added. An adjacency arc is defined as the minor arc of the great circle containing any pair of point masses on the Gaussian sphere (Moni, 1990). However, the face adjacency information of a polyhedral cell is not directly recoverable from the location of the point masses and arcs alone.

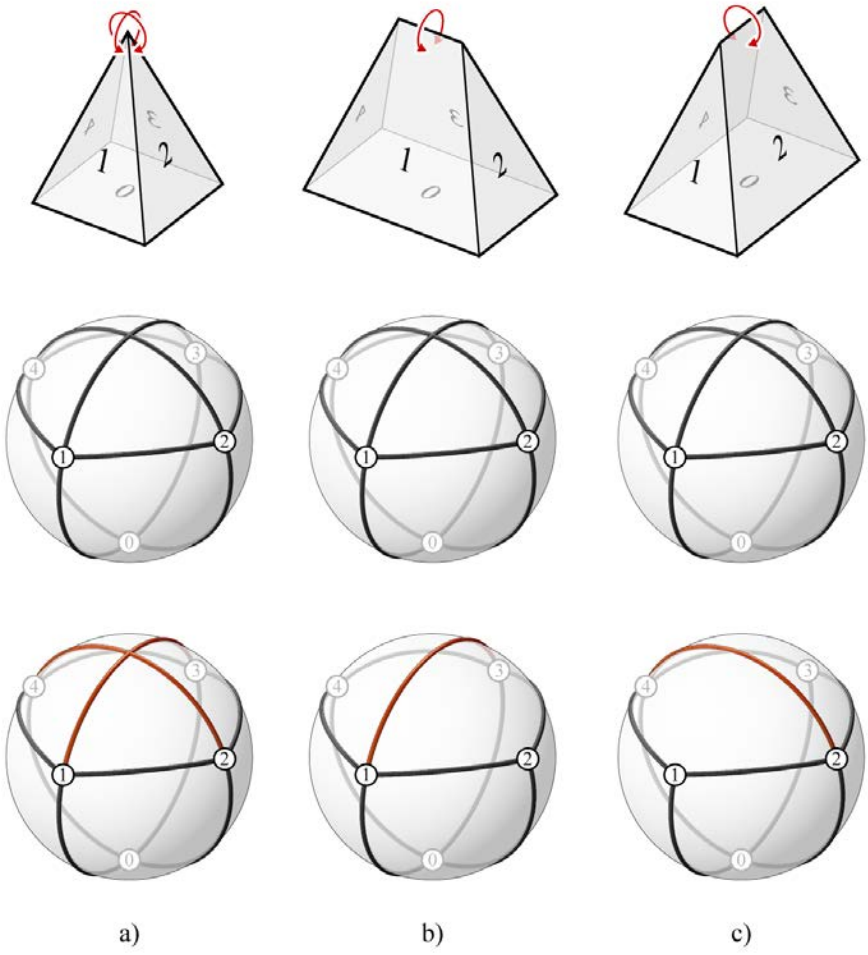


Figure 4.14: Three polyhedral cells with same face orientations, but with different areas: a) faces 1 through 4 are adjacent with one another at a single vertex; b) faces 1 and 3 are adjacent along an edge; and c) faces 2 and 4 are adjacent along an edge.

Consider the three cells shown in Figure 4.14. These three cells have the same face orientations but different face areas. Based on the definition of an arc stated above, the face normals and areas of the three polyhedral cells result in the same EGIs (middle row of Figure 4.14). However, the actual EGIs (bottom row of Figure 4.14) are different, as the different face area distributions result in different face adjacencies. Depending on the face area distribution, various face adjacencies occur: faces 1-4 are all adjacent with one another at a vertex; faces 1 and 3 are adjacent; or faces 2 and 4 are adjacent.

This indeterminacy of face adjacencies occurs where an arc crosses another arc on the EGI. Moni (1990) defines these arc intersections as "cross-adjacencies," where various face adjacencies could occur depending on different face area distributions. By adding a fictitious, "zero point mass" at these arc intersections, and subsequently a face with zero-area, or a "zero face," all possible face adjacencies can be embedded and represented in a single EGI (Figure 4.15-a). Once all zero point masses have been added, the datastructure of the EGI as the dual spherical polyhedron is complete, and the "unit cell" can now be constructed, which is simply the topological primal of the dual spherical polyhedron or the EGI (Figure 4.15-b). All zero faces have a target area of zero.

One of the most remarkable properties of EGI is that the centre of mass of the EGI's point masses has to lie at the origin of the Gaussian sphere (Horn, 1984). This means that there cannot exist a hemisphere on the EGI that does not have a point mass, which would represent an unbounded polyhedron. However, there are commonly used node elements in structural design for which the corresponding EGI may have one or multiple empty hemispheres, such as: a prismatic polyhedral cell representing the equilibrium of a coplanar system of forces (Figure 4.16-a); an open node (Figure 4.16-b); or a node with members that may temporarily be unequilibrated during the form-finding process (Figure 4.16-c).

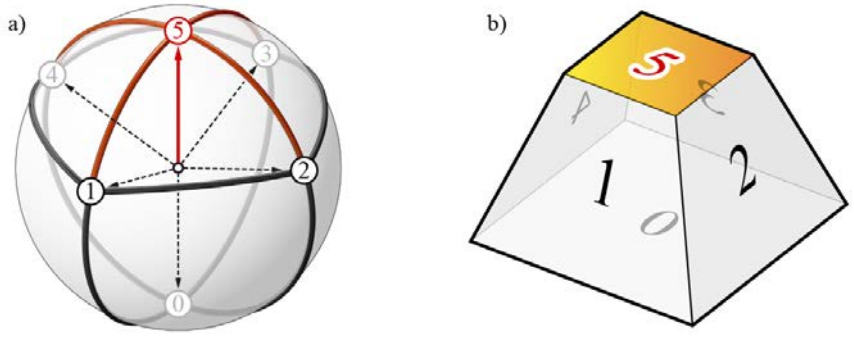


Figure 4.15: a) The EGI with zero point mass 5 added; and b) the corresponding unit cell with an added zero face.

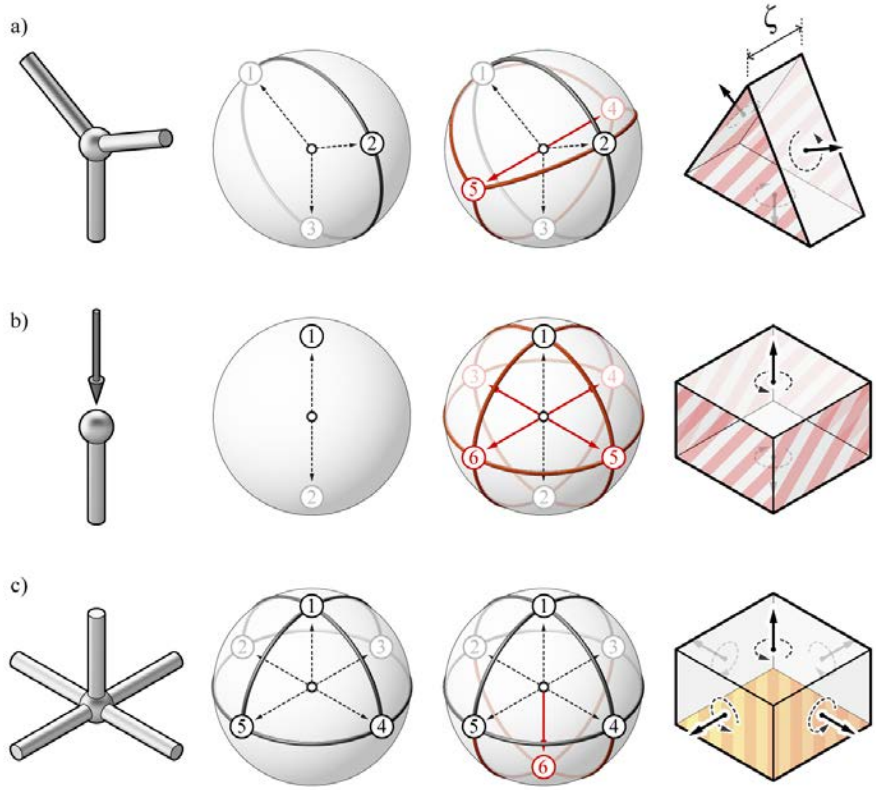


Figure 4.16: a) A 2D node, its EGI and the prismatic polyhedral cell with two added virtual faces; b) an open node, its EGI and the polyhedral cell with four added virtual faces; and c) a temporarily unequilibrated node, its EGI and the corresponding polyhedral cell with one added virtual face, which in this case is also a zero face.

For these special cases, "virtual faces" are introduced to complete the geometry of the unit cell. Virtual faces are not the same as zero faces, and have no target area values; the only purpose of the virtual faces is to facilitate the geometric construction of cells in these special situations, and have no corresponding force or member in the form diagram. In general, if the centre of mass is not at the origin of the Gaussian sphere, or if there are any empty hemispheres, virtual point masses are placed at the centre of the empty hemispheres. In the case of a prismatic polyhedral cell, only two virtual faces are needed, with an arbitrary scale factor of ζ (see Section 4.2.7.2). For an open node, at least three virtual faces are needed to complete the polyhedral cell (Figure 4.16-b). During the interactive design process, some polyhedral cells can become temporarily unbounded. A virtual face can be added for each empty hemisphere, and only in such unbounded cases, the virtual faces are treated as zero faces with target areas of zero.

Once the topological datastructure of the unit cell has been constructed, each face of the unit cell can be iteratively re-sized towards their target areas. Zero faces and virtual faces which have target areas of zero eventually collapses to an edge or vertex as a result of this iterative procedure. A more detailed explanation of the technical implementation of this procedure will be discussed in Chapter 5.

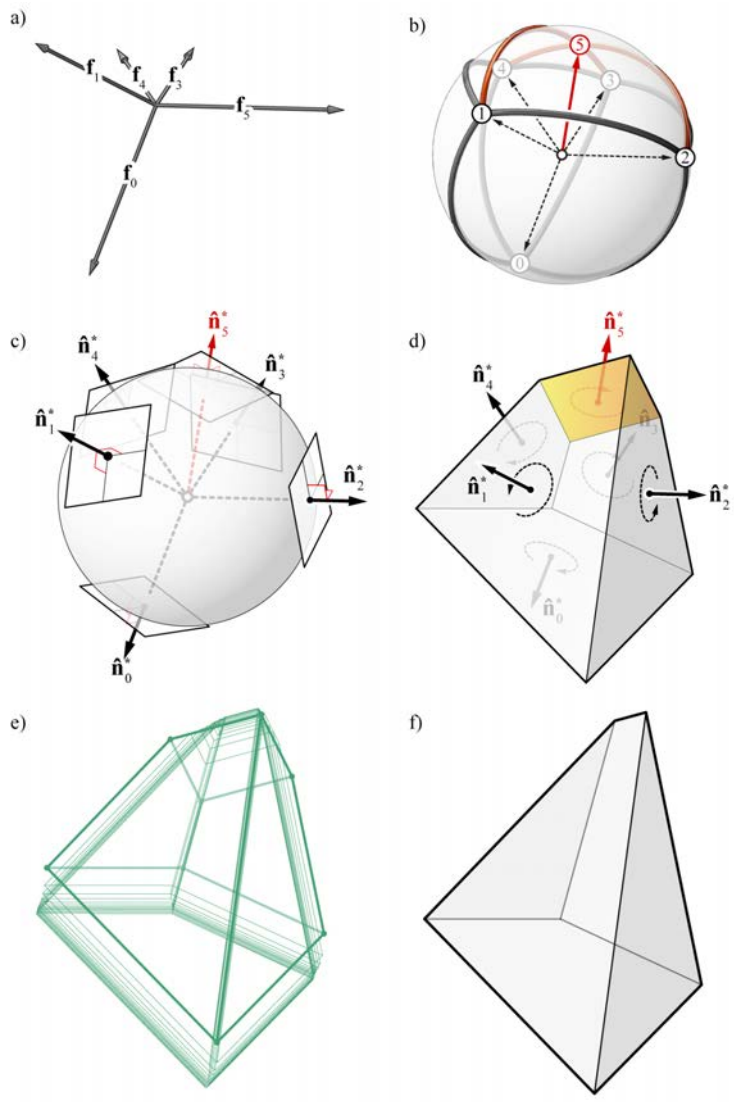


Figure 4.17: Construction sequence of a polyhedral cell using the EGI: a) equilibrated force vectors; b) the EGI with adjacency and cross-adjacency arcs; c) normal vectors at each of the point masses on the Gaussian sphere; d) unit polyhedral cell; e) iterative re-sizing of the faces towards their target areas; and f) geometry of the final polyhedral cell.

4.2.10 Summary

This section presented polyhedral cells, which can represent the equilibrium of: 1) the external forces of a structure; or 2) a single node of a multi-node structure. Investigation of single polyhedral cells is important in developing robust and generalised utility functions and understanding of various topological structures as they relate to the geometry of the polyhedral cells. The generalised utility functions then can be used to develop various transformation functions of a single polyhedral cell. The construction of polyhedral cells is also important in initialising the global force polyhedron for addressing boundary conditions and external forces. However, typical structures consist of multiple nodes. Therefore, in the following section, a datastructure for representing the equilibrium of multi-node structures will be presented.

4.3 Multi-cell polyhedron

This section presents the multi-cell polyhedron datastructure, which can represent the static equilibrium of spatial structures with multiple nodes.

4.3.1 Definition

A polytope is a geometric object with faces (closed polygonal loop of edges), which can exist in any general n number of dimensions as n-dimensional polytope or n-polytope (Coxeter, 1973). A n-polytope is made up of $(n-1)$-polytopes and also referred to as a $(n-1)$-manifold. For example, a polyhedral cell which is a 3-polytope is a 2-manifold. A vertex is a 0-polytope in zero dimension, while an edge is a 1-polytope in one dimension. A face in two dimensions is a 2-polytope made up of 1-polytopes, edges. A polyhedral cell in three dimensions is a 3-polytope made up of 2-polytopes, faces.

Following the same analogy, a 4-polytope in four dimensions is then made up of 3-polytopes, or polyhedral cells. Because a polyhedral cell represents the equilibrium of a single node, a force diagram of a structure with multiple nodes consists of multiple polyhedral cells. This set of conjoined polyhedral cells can be described as a 4-polytope (McRobie, 2017a). This 4-polytope is defined as multi-cell polyhedron, which represents the polyhedral force diagram $\Gamma^{\perp}$ of a polyhedral form diagram Γ with multiple nodes.

4.3.2 Datastructure

The adjacency information of the faces of a 3-polytope or a polyhedral cell, are defined by the halfedges. The adjacency information of the cells of a 4-polytope or a multi-cell polyhedron, can be defined by a combination of *halffaces* and *planes*. Whereas each edge of a polyhedral cell is split into two halfedges with opposite directions, each face of a multi-cell polyhedron is split into two halffaces with opposite orientations. The halfedges of a polyhedral cell store their incident face information, while the halffaces of a multi-cell polyhedron store their incident cell information. Using the halfface orientations and its incident cell information, various data of a multi-cell polyhedron can be accessed and traversed. The data of a multi-cell polyhedron consists of vertices, edges, halffaces, cells and planes. The vertices and edges of a multi-cell polyhedron have the same structure as the vertices and edges of a polyhedral cell (see Section 4.2.2).

Each j-th halfface hf_j^* of the multi-cell polyhedron $\Gamma^{\perp}$ is stored as a hfkey-value pair (Figure 4.18-c). The halfface key (**`hfkey`**) is the unique identifier of the halfface, and the value is an ordered list of **`vkeys`** that form the polygonal loop of the halfface. The first **`vkey`** and the last **`vkey`** are not be the same. Every consecutive pair of this list of ordered **`vkeys`** (including the last to the first), is a halfedge. Each cell c_i^* of $\Gamma^{\perp}$ is made up of halffaces which are unique to that cell.

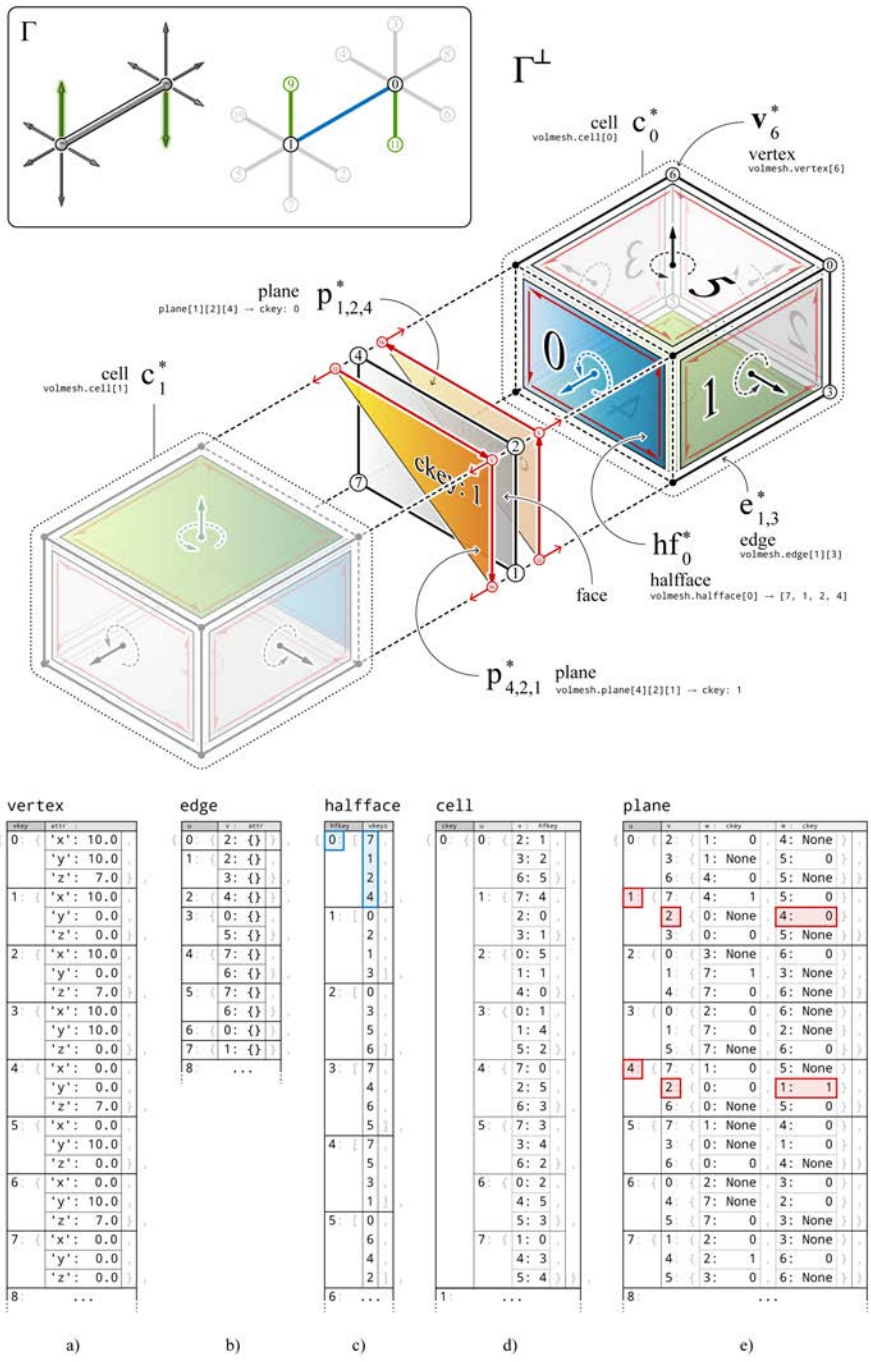


Figure 4.18: The datastructure of a multi-cell polyhedron with two cells: a) the vertex dictionary; b) the edge dictionary; c) the face dictionary; d) the half-edge dictionary; and e) the plane dictionary.

Each i-th cell c_i^* of the multi-cell polyhedron $\Gamma^\perp$ is stored as a ckey-vkey-vkey-hfkey quadruple (Figure 4.18-d). The **ckey** or the cell key is the unique identifier of a cell. Each **ckey** points to two ordered **vkeys** that define a halfedge which belongs to one of the cell's halffaces. This halfedge then points to the **hfkey** of the incident halfface.

Each plane of a multi-cell polyhedron is stored as vkey-vkey-vkey-ckey quadruple (Figure 4.18-e). The three ordered **vkeys**, **u**, **v** and **w** as a sequence, is the unique identifier of the plane **(u,v,w)**. **u**, **v** and **w** are three consecutive vkeys of a halfface, or two consecutive halfedges **(u,v)** and **(v,w)**. Each plane then points to the incident cell. Planes essentially store the halfface and their incident cell information.

Consider the multi-cell polyhedron shown in Figure 4.18. This multi-cell polyhedron has two cells. Halfface hf_0^* of cell c_0^*, is made of four vertices: **7**, **1**, **2** and **4** (highlighted in blue, Figure 4.18-c). Plane **(1,2,4)** is defined by two halfedges **(1,2)** and **(2,4)** of hf_0^*. This plane points to c_0^*, which is the incident cell of hf_0^* (highlighted in red, Figure 4.18-e). Plane **(4,2,1)** belongs to the halfface which is in the opposite orientation compared to hf_0^*. Because plane **(4,2,1)** belongs to a halfface which does not belong to c_0^*, it points to the adjacent cell of cell 0, which in this case is 1 (highlighted in red, Figure 4.18-e).

4.3.3 Interpretation

As discussed in Section 4.2.7, the interpretation of the internal force of member at a node as either compression or tension can be made by comparing its orientation relative to the oriented normal of the corresponding halfface of $\Gamma^\perp$. For a $\Gamma^\perp$ that contains multiple cells, the interpretation of forces can also be made by comparing cell directions of neighbouring adjacent cells (Akbarzadeh et al., 2017).

Based on the convention established in Section 4.2.7, the two nodes that define a member in tension will correspond to two cells with positive cell directions in $\Gamma^\perp$ (Figure 4.19-a); two nodes that define a member in compression will correspond to two cells with negative cell directions (Figure 4.19-b). $\Gamma^\perp$ for compression-only funicular structures will only consist of cells with negative directions, whereas tension-only funicular structures will only consist of cells with positive directions. For a structure that contains members in both compression and tension, $\Gamma^\perp$ contains both positive and negative cells.

It is possible for a polyhedral cell to have no cell direction. For example, a complex polyhedral cell where all of its faces are complex and have oriented areas of zero (and therefore no oriented normals), would have no cell direction. Therefore, if either of the pair of adjacent cells have no cell direction, the corresponding member in Γ would be a zero-force member.

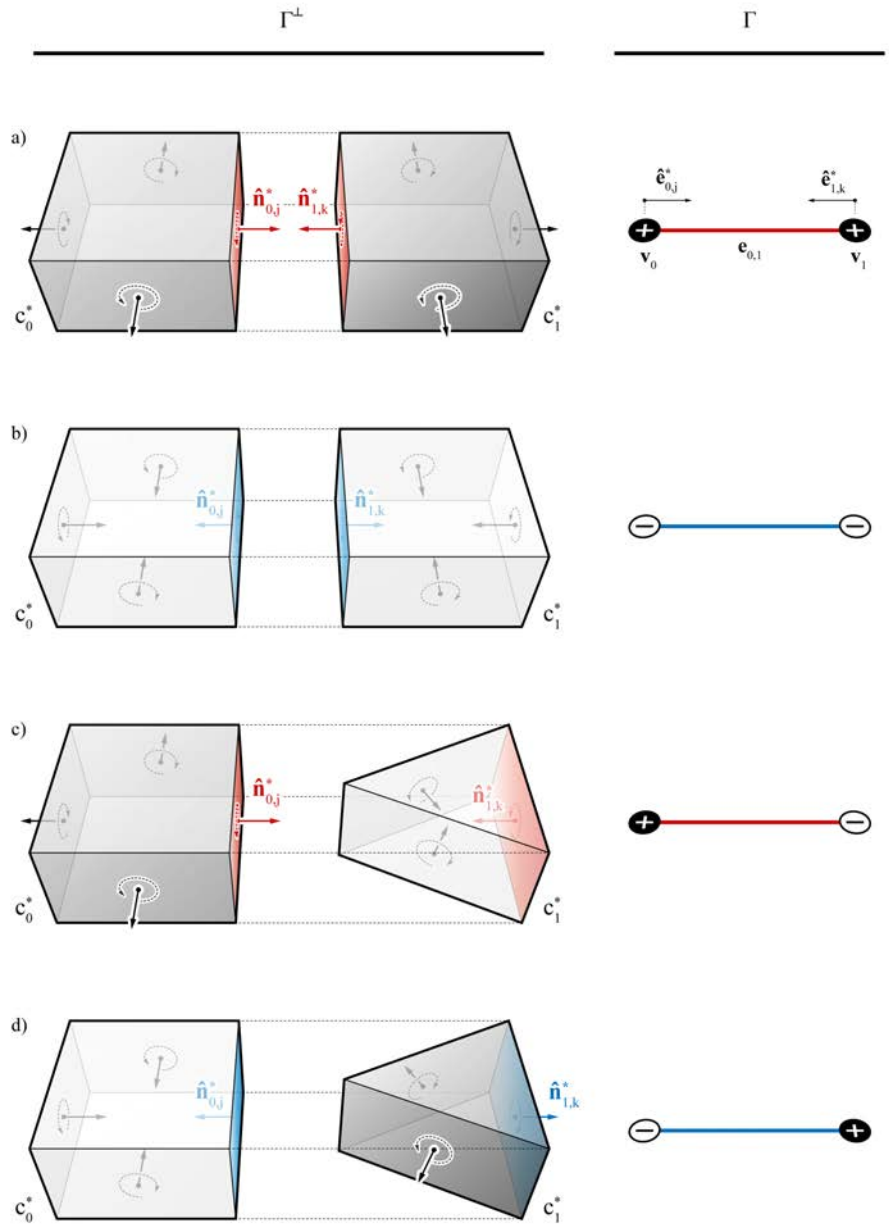


Figure 4.19: Interpretation of member forces in relation to the cell directions of the corresponding pair of adjacent cells: a) two adjacent cells with positive cell directions, which corresponds to a member in tension; b) two adjacent cells with negative cell directions, which corresponds to a member compression; c) and d) two adjacent cells that have different cell directions may correspond to a member in either compression or tension.

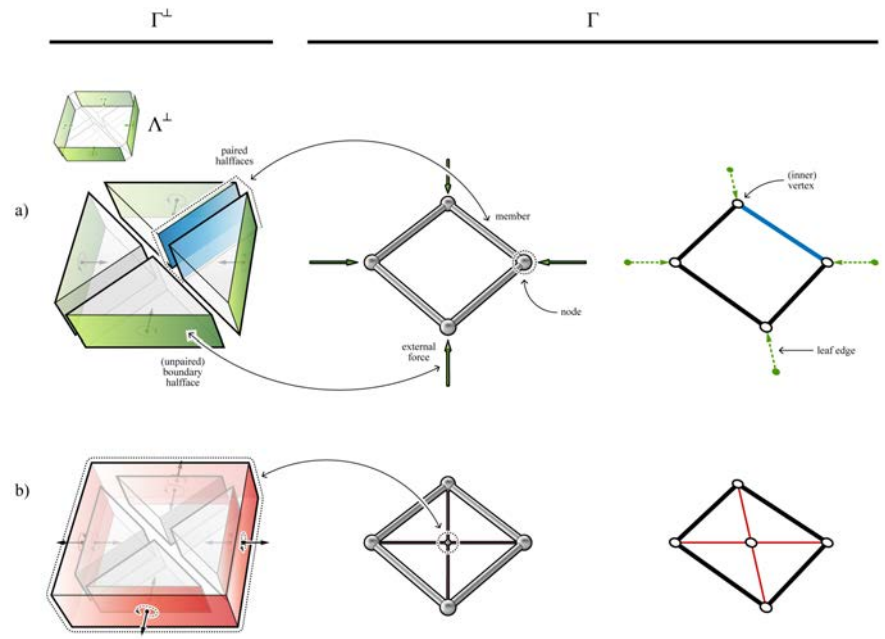


Figure 4.20: A force diagram $\Gamma^{\perp}$ with prismatic polyhedral cells and the corresponding 2D form diagram Γ (vertical forces not shown): a) a structure with four external forces; and b) a self-stressed structure with five cells and no external forces.

4.3.4 Hierarchy

For externally loaded structures, the cells of a $\Gamma^{\perp}$ can be categorised into two groups: the global force polyhedron Λ representing the equilibrium of the external forces; and nodal polyhedral cells c_i^* that each represent the local equilibrium at node $\mathbf{v}_i^*$. While each cell c_i^* corresponds to a node, a pair of conjoined cells represent a member in Γ (Figure 4.20-a). At the interface of two conjoined cells is a pair of halffaces. The boundary halffaces at the extremities of a $\Gamma^{\perp}$ are not paired, and therefore represent external forces. The combination of all of the boundary halffaces represents the global force polyhedron, $\Lambda^{\perp}$.

Self-stressed structures have no external forces. Therefore, the corresponding $\Gamma^{\perp}$ does not have any unpaired, boundary halffaces. The example shown in (Figure 4.20-b) has four triangular prismatic cells which are all contained within a single outer rectangular cell. In contrast to the example in Figure 4.20-a, the four external forces represented by the four boundary halffaces, is now internally resolved at a single central node whose equilibrium is represented by the outer rectangular cell.

4.3.5 Unified diagram

Although one of the most valuable benefits of computational graphic statics is the visualisation and explicit control of both the structure's geometry and its equilibrium of forces, the form and force diagrams increasingly become more visually cluttered and illegible as structures become more complicated. The illegibility is even more severe for polyhedral reciprocal diagrams, where it is quite difficult to perceive quantitative information through volumes and face areas of solid geometries (Cleveland and McGill, 1984), especially when they are represented as 2D images on flat media. Therefore, the visualisation of $\Gamma^{\perp}$ needs to be improved in order to fully take advantage of the inherent benefits of graphic statics, and make the polyhedral reciprocal diagrams more legible, usable and interactive.

The *unified diagram*, $\Gamma^{\perp}(\alpha)$, represents both the geometry and internal forces of a structure in a single diagram, thereby improving the legibility of reciprocal diagrams (Zanni and Pennock, 2009) (Figure 4.21). A method for computing and displaying the unified diagrams was presented by McRobie (2016a) as "Minkowski sum" diagrams: "Maxwell-Minkowski" diagrams for 2D cases, and as "Rankine-Minkowski" diagrams for 3D cases. This method was later refined and generalised as the "Corsican sum" (McRobie, 2017a). In order to avoid confusion of the terminology, the combined form and force diagrams are henceforth referred to as unified diagrams.

In unified diagrams, the constituent polygons or cells of the force diagram are scaled relative to the corresponding nodal coordinates in the form diagram, resulting in an "exploded view" of the force diagram. Each pair of neighbouring polygons or cells are then connected in two dimensions by interstitial rectangles and in three dimensions by interstitial prisms.

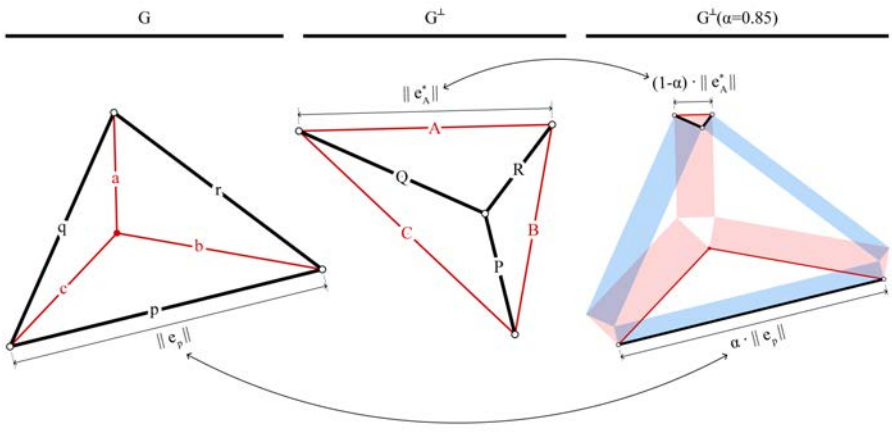


Figure 4.21: Form diagram G, the reciprocal force diagram $G^{\perp}$ and the unified diagram $G^{\perp}(\alpha)$ for a simple truss (After Maxwell (1864) and McRobie (2016a)).

In contrast to traditional, side-by-side representation of form and force diagrams, the unique visualisation method of unified diagrams provide new insights and perspectives. $\Gamma^{\perp}(\alpha)$ is not only more discernible, but also provides an interesting visual representation of the material required for a uniform stress design (McRobie, 2016a). Furthermore, the unified diagram reveals visual insights in relation to some of the most fundamental principles of structural engineering and analysis, such as: kinematics and mechanisms (Zanni and Pennock, 2009); virtual work and displacements (McRobie et al., 2017); and stress-fields and strut-and-tie models (Schlaich and Anagnostou, 1990; Muttoni et al., 1997). Most importantly, the usefulness of unified diagrams is even more apparent for 3D trusses. Because conjoined polyhedral diagrams have volumetric objects stacked on top of each other, it is difficult to read, understand and discern in an intuitive manner.

The generalisation of the unified diagram as the Corsican sum (McRobie, 2017a) allows any geometry that is topologically dual to the polyhedral force diagram to be in equilibrium; the members of the structure do not need to be straight, nor do they need to be perpendicular to the corresponding faces in the force diagram. However, only 3D trusses carrying axial loads are considered in this dissertation, and therefore, the members of the structure needs to be straight and be perpendicular to the corresponding faces in the force diagram. In order to construct the unified diagram for trusses that are under axial loads only, the nodal coordinates of Γ and the cell geometries of $\Gamma^{\perp}$ are needed.

Once Γ and $\Gamma^{\perp}$ are both computed, a simple geometric algorithm can be used to visualise $\Gamma^{\perp}(\alpha)$. By parametrically modifying the scaling factor α, all cells of $\Gamma^{\perp}(\alpha)$ are scaled relative to its corresponding nodes of the structure such that the distance between any pair of adjacent cells is $\alpha \cdot L$, where L is the length of the corresponding member in Γ. Each cell is now positioned in space such that every pair of halffaces that represent a member in Γ are not only parallel, but also orthogonal translations of each other.

For a conjoined polyhedral force diagram $\Gamma^{\perp}$, every pair of adjacent halffaces have the same topology and geometry. Therefore, one of the halffaces can be simply extruded by length $\alpha \cdot L$ to visualise the interstitial prisms. A scaling factor of 1 results in Γ, whereas lower values of α closer to 0 will result in a $\Gamma^{\perp}(\alpha)$ that more closely resembles the polyhedral force diagram $\Gamma^{\perp}$ (Figure 4.23-b). The volume of the interstitial prisms that are formed in between the adjacent cells is equivalent to the work $\mathbf{f} \cdot L$ being done by the corresponding member, where $\mathbf{f}$ is the internal force in the member between the two nodes, and L is the length of that member.

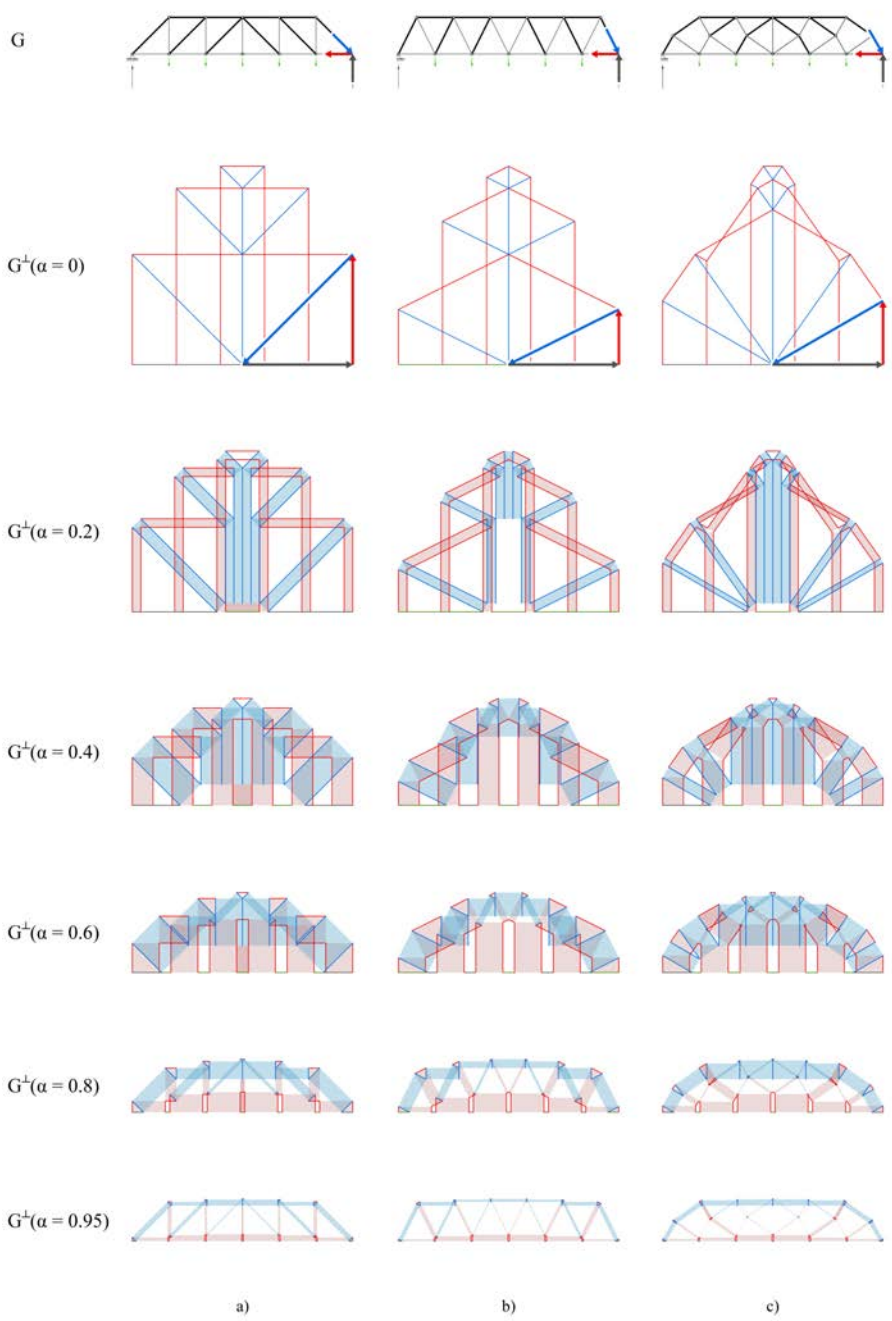


Figure 4.22: Form diagram G and the unified diagram $G^{\perp}(\alpha)$ with varying scale factor values for α, for three 2D trusses: a) Howe truss; b) Warren truss; and c) Michell truss;

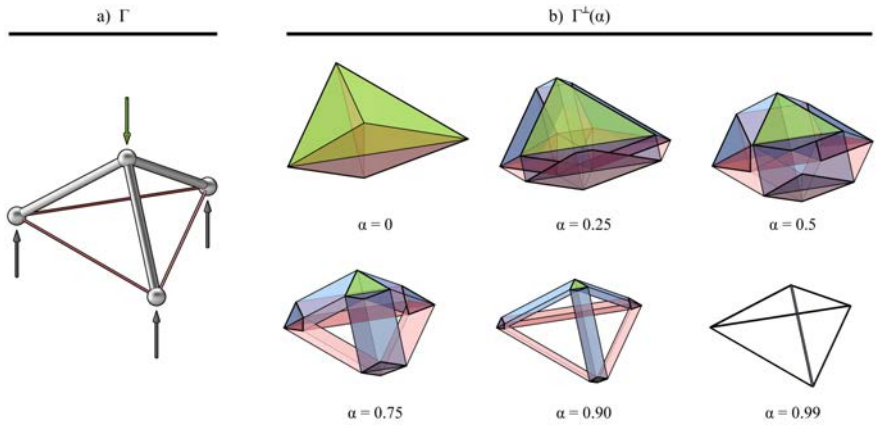


Figure 4.23: A simple 3D truss and its unified diagram: a) Γ of the truss, which is equivalent to $\Gamma^{\perp}(\alpha = 1)$; and b) the $\Gamma^{\perp}(\alpha)$ with varying scaling factor of α.

4.3.6 Operations

Compared to polyhedral cells, geometric transformations of multi-cell polyhedrons are not as straightforward. Pulling or tilting a face of a multi-cell polyhedron will have a cascading effect on a series of faces at once. A vertex-based method for constrained geometric manipulations of multi-cell polyhedrons was presented in Nejur and Akbarzadeh (2018). In this approach, the new geometry of multi-cell polyhedrons are computed through local edge intersections to determine the new vertex locations. The face orientation constraint is enforced edge-per-edge by maintaining the initial edge orientations. Because the computation of the resultant geometry is dependent on series of local geometric procedures, vertex-based manipulation requires several steps: formulation of transformation graphs, cluster graphs, transformation prioritisation and transformation propagation. Alternatively, geometric manipulations of multi-cell polyhedrons while maintaining the initial face orientations can be conceptualised as a face relocation problem, where the localised operations are applied at the cell level. Because face-based transformations inherently enforce the planarity and orientation constraints, a simpler method can be formulated.

The first step for applying a constrained face operation to an arbitrary face $f^*_{i,j}$ (shown in blue, Figure 4.24) of a multi-cell polyhedron is to identify the dependent faces (shown in gold, Figure 4.24). The dependent faces are other neighbouring faces of $f^*_{i,j}$ that will also need to be transformed as a by-product of the face pull operation on $f^*_{i,j}$. A dependent face is a face of the neighbouring cells which share an edge with the initial face $f^*_{i,j}$, or other dependent faces. There can be only one dependent face per cell. The dependent face finding procedure is repeated until there are no more faces left in the multi-cell polyhedron that meets this requirement.

Once all of the dependent faces have been identified, the transformation operation is a linear process. The initial face $f^*_{i,j}$ can be locally pulled or tilted, following the procedure described in Section 4.2.8. With the new location of $f^*_{i,j}$, the new positions of its dependent faces can be determined one after another.

For each dependent face, the new location is one of the vertices of the edge that is shared with $f^*_{i,j}$. The new dependent face locations can be used to update the geometry of the corresponding cells. The transformation continues until all of the cells with dependent faces have been updated. Instead of using the initial normals of $f^*_{i,j}$ and its dependent faces for the recursive face pull operations, a target normal can be used. This effectively projects $f^*_{i,j}$ and its dependent faces to a target plane, which could be a useful in constraining multiple faces at once (Figure 4.24-d). This transformation can be applied to the multi-cell polyhedrons as well as their corresponding polyhedral form diagrams.

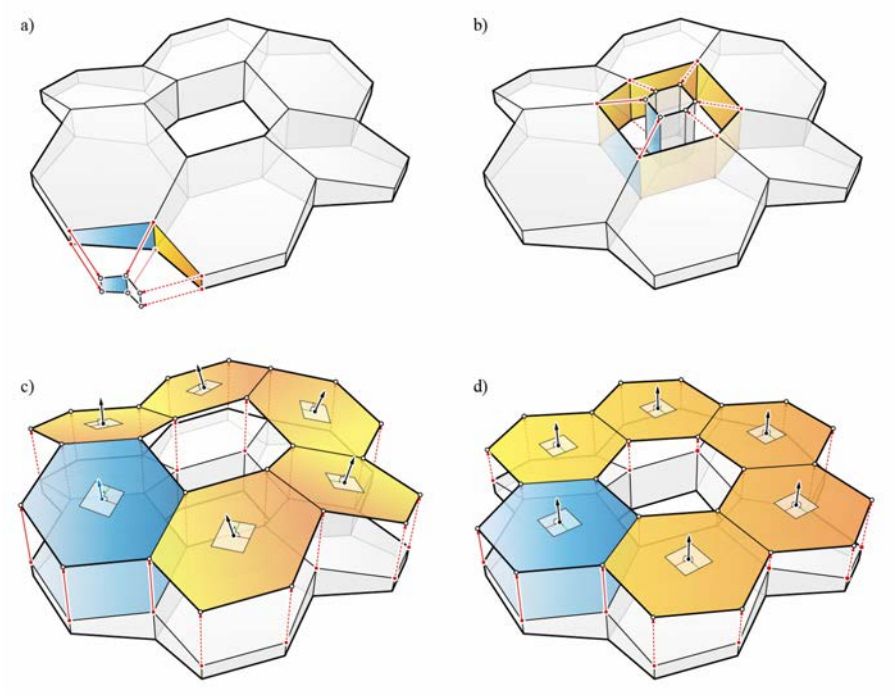


Figure 4.24: a), b) , c) Constrained transformations of multi-cell polyhedrons through recursive face pull operations (selected face for transformation shown in blue, and their dependent faces shown in gold); and d) face pull operations of the selected face and its dependent faces with a target plane.

The resulting geometry of multi-cell polyhedrons is not unique, and is subject to various global constraints enforced by the user. For example, tilting faces of a multi-cell polyhedron cannot be executed unless generalised rules or assumptions are provided, such as how much the dependent faces are allowed to change relatively to one another. Furthermore, pulling of interior faces can have multiple solutions, depending on which faces are allowed to move and which are not. Most importantly, a large number of triangular faces, especially in the interior cells of a multi-cell polyhedron can be extremely constraining for any geometric operation. Unifying transformations such as aligning all dependent faces to a target plane, especially for boundary faces, can be useful in enforcing geometric constraints to multiple cells at once. However, geometric manipulations of multi-cell polyhedrons in general, are highly constrained problems that are directly dependent on the the initial topology.

4.3.7 Summary

Multi-cell polyhedrons have intrinsic topological constraints that are globally enforced onto all of its constituent cells. As a result, it is an ideal datastructure for exploring top-down operations such as subdivisions and global geometric manipulations. However, such inherent constraints of multi-cell polyhedrons also results in some key limitations. It is difficult to impose specific distribution of external forces based on realistic tributary areas or loading conditions. Also, the corresponding form diagrams of multi-cell polyhedrons are strictly polyhedral in its geometry, which is only a small subset of equilibrated spatial structures that can be explored. Furthermore, the globally enforced topological constraint makes it challenging to perform any local geometric transformations of multi-cell polyhedrons, especially ones involving internal cells.

To address these limitations of multi-cell polyhedrons, the next section presents a datastructure for an unconstrained polyhedral force diagram, which can liberate the individual cells from global topological constraint while maintaining the equilibrium of the structure both locally and globally.

4.4 Cell network

This section presents cell network datastructure, which is a generalised polyhedral force diagram for representing the static equilibrium of spatial trusses, which are nodally loaded and comprised of straight bars.

4.4.1 Definition

Previously presented design explorations using 3D graphic statics have been based on aggregations, subdivisions or transformations of polyhedral force diagrams where all pairs of adjacent cells have matching halfface geometries (halffaces hf_1^* and hf_0^* in Figure 4.25-a), and therefore can be assembled into a conjoined multi-cell polyhedron, $\Gamma^\perp$. The concept of neighbouring cells with dissimilar or mismatching *contact faces* was first introduced by McRobie (2016b) (faces $f_{0,1}^*$ and $f_{1,0}^*$ in Figure 4.25-b). In more recent papers, McRobie (2017a; 2017b) showed that the equilibrium of two neighbouring cells with mismatching contact face geometries can be explained using "face cushions" or collapsed cells (cell c_2^* and in Figure 4.25-c). The "face cushions" represent fictitious nodes in Γ with a net force of zero at that node, which is important in constructing a $\Gamma^\perp$ for more complicated structures. The collapsed cell in-between two adjacent cells allows the two initial cells to no longer be conjoined; the two cells are "disjointed" in the sense that the contact faces in between the two cells only need to have equal areas, but not necessarily matching geometries.

A collection of disjointed cells in equilibrium is defined as a *cell network*, $\Psi^\perp$. Each individual cell of $\Psi^\perp$ represents the local equilibrium of a node in the corresponding form diagram, Ψ. At the local level of a node, the corresponding cell behaves like a polyhedral cell. Unlike the polyhedral form diagram Γ, the form diagram Ψ which corresponds to a cell network $\Psi^\perp$ is not necessarily polyhedral in its geometry. The topological structures of Ψ and $\Psi^\perp$ do not have a dual relationship, which means they can not be defined as reciprocal; $\Psi^\perp$ is a collection of cells that are individually reciprocal to its corresponding nodes in Ψ. On the other hand, the polyhedral form and force diagrams Γ and $\Gamma^\perp$ are mutually dual and reciprocal, and are special cases of a non-polyhedral form diagram Ψ and the corresponding cell network $\Psi^\perp$, respectively.

4.4.2 Datastructure

A cell network is a collection of polyhedral cells. Because the polyhedral cells of a cell network are disconnected from one another and not explicitly conjoined into a single topological object, another layer of datastructure is needed to manage and control the inter-cell connectivities and relationships.

The cell connectivity information is embedded into the cell network at the global level, as a network or graph datastructure. A network is a collection of

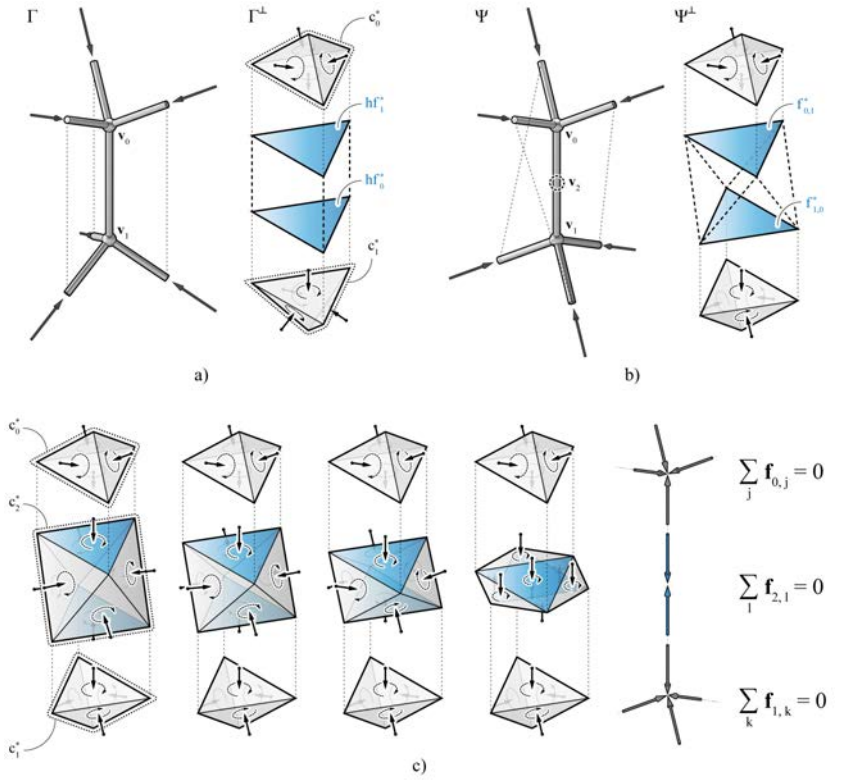


Figure 4.25: a) Two nodes in equilibrium where the two cells have matching halffaces; b) two nodes in equilibrium where the two cells are disjointed with mismatching contact faces; and c) fictitious node $\mathbf{v}_2$, and the corresponding cell c_2^* linking the two cells with mismatching contact faces, in convex and collapsed states.

vertices and a set of edges connecting the vertices. The vertices store various data and information, while the edges defined the relationship between pairs of vertices. With the coordinate information stored in each of the vertices, this global layer of the cell network datastructure also defines the geometry and topology of the non-polyhedral form diagram, Ψ (Figure 4.26-a, b). At the local level, each vertex of the cell network contains the data of the polyhedral cell that represents the equilibrium of that vertex (Figure 4.26-c).

The vertices and edges of cell networks are stored in the same manner as in polyhedral cells and multi-cell polyhedrons; the vertices are stored as vkey-value pairs, while the edges are stored as vkey-vkey-value triples. Each **`fkey`** of the faces of a cell corresponds to the **`vkey`** of the vertex to which the face is pointing. In the simple example shown in Figure 4.26, **`fkey`** of 1 is assigned to the face of cell of vertex 0 which is adjacent to the cell of vertex 1. Similarly, **`fkey`** of 0 is assigned to the face of cell of vertex 1 which is adjacent to the cell of vertex 0.

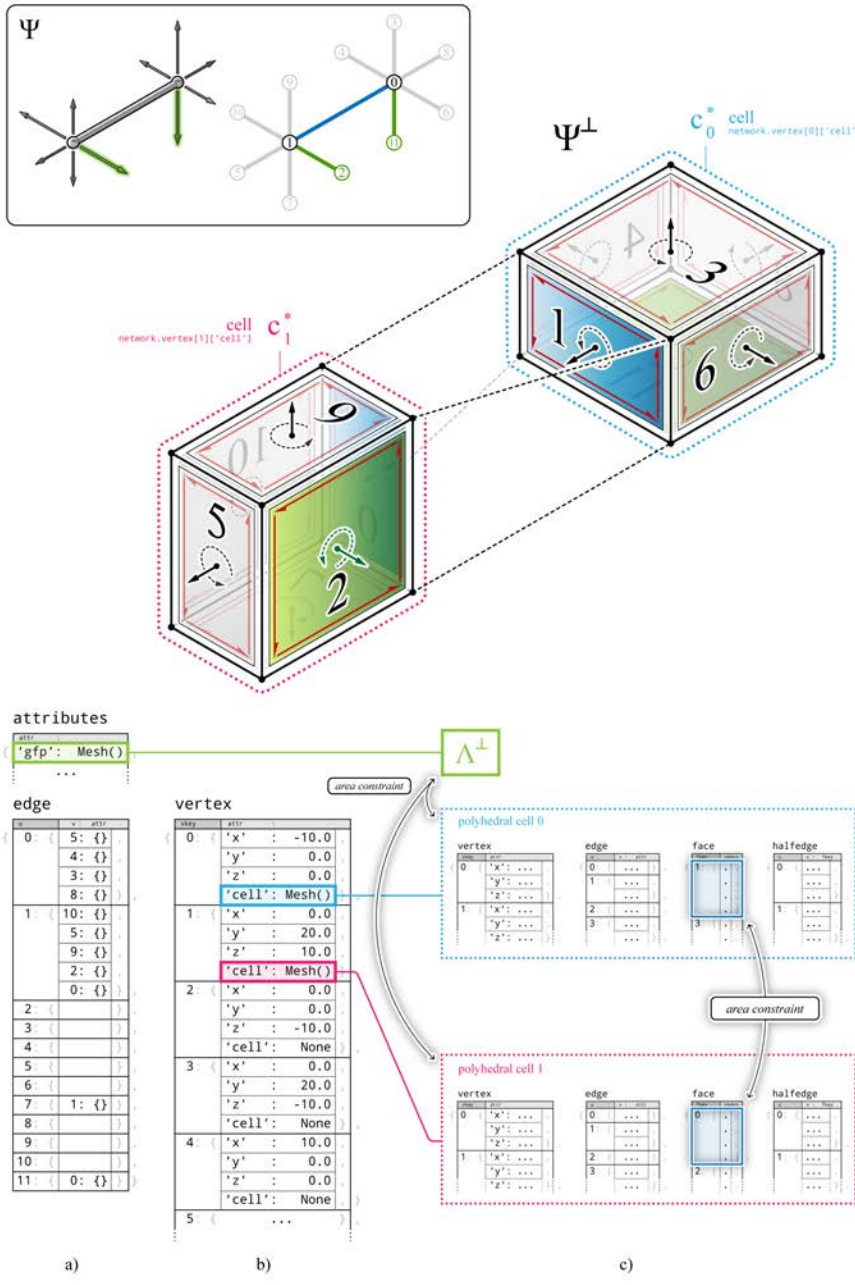


Figure 4.26: The datastructure of a cell network $\Psi^{\perp}$ with two cells.

4.4.3 Hierarchy

For a multi-cell polyhedron, the unpaired boundary halffaces can be joined to construct the global force polyhedron, $\Lambda^{\perp}$. For a cell network, the unpaired faces at the extremities of the structure cannot simply be joined because of potential topological incompatibilities of those faces. The global force polyhedron $\Lambda^{\perp}$ of a cell network needs be constructed independently from the cells. $\Lambda^{\perp}$ can be stored as an independent object within the cell network datastructure. It can carry the necessary boundary condition information, such as the magnitudes and orientations of the known external forces. As a result, $\Lambda^{\perp}$ imposes the boundary condition constraints onto the cell network starting from the peripheral cells and propagating towards the interior cells.

4.4.4 Unified diagram

For cell networks, the construction of unified diagrams, $\Psi^{\perp}(\alpha)$, is not as straightforward since the prisms cannot simply be extruded due to the mismatching contact faces in between pairs of cells. A true $\Psi^{\perp}(\alpha)$ would show the collapsed cells between two disjointed cells, with two prisms instead of one (Figure 4.27-b). However, with the priority being placed on maximising the visual clarity and legibility of unified diagrams, the representation of the two prisms and a collapsed cell can be simplified by using a convex-hull of the two contact faces (Figure 4.27-c, d). Note that the volume of this convex-hull is not $\mathbf{f} \cdot L$.

The unified diagrams for cell networks are applied purely as a visual approximation to improve the legibility of $\Psi^{\perp}$. A $\Gamma^{\perp}$ for a structure having both compression and tension elements typically consist of cells with positive and negative cell directions; the face normals of some cells point inward, and some cells outward (Lee et al., 2016; Akbarzadeh, 2016). In cell networks, all cells of a $\Psi^{\perp}$ are locally constructed and have the same directions (either all negative or all positive, depending on the conventions chosen by the user). Therefore, the distance between any pair of neighbouring cells in a $\Psi^{\perp}(\alpha)$ is not precisely $\alpha \cdot L$.

4.4.5 Summary

Cell networks are unconstrained polyhedral force diagrams that enable more area-based (force-driven) constraints to be imposed. Because the topology of the individual cells can change independently from one another, cell networks allow force-driven design explorations, which may be not be possible using multi-cell polyhedrons that have globally enforced topological and geometric constraints. Cell networks can be particularly useful in later stages of design, when more realistic and force-driven considerations need to be enforced once the overall design of the structural geometry has been finalised. However, the lack of intrinsic global constraints of cell networks

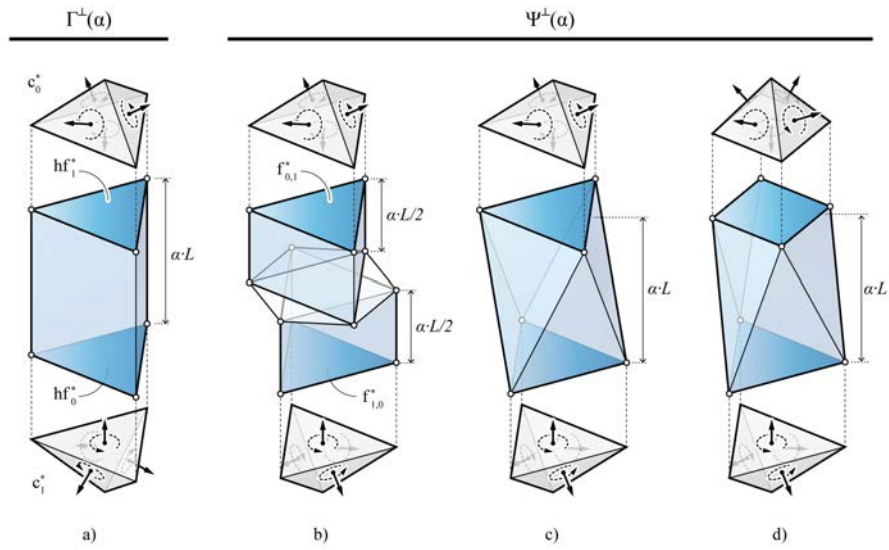


Figure 4.27: a) Interstitial prism between two cells of a $\Gamma^{\perp}(\alpha)$; b) the collapsed cell and the two interstitial prisms between two cells of a $\Psi^{\perp}(\alpha)$; c) convex-hull as a visual alternative to b) with the collapsed cell omitted, for two cells with the same valency; and d) two cells with different valencies.

require that users are more cognisant of the boundary condition constraints, and what parameters can remain free.

In general, Ψ for a given force distribution is not always unique, and is subject to certain geometric constraints such as maximum and minimum length of edges allowed, and node location constraints. Consequently, a design problem can become over-constrained and an equilibrium solution may not be found that satisfies all of the input constraints. In such cases, the converged solution then provides the designer with the closest solution given the input constraints, and indicates where certain constraints can be removed or modified. It will be shown in Part 3 of the dissertation, that these constraints can be used as an opportunity for design rather than a hindrance; the users can interactively set different combinations of constraints to explore various equilibrium solutions.

4.5 Summary

This chapter presented three datastructures for addressing different types of equilibrium problems involving spatial system of forces. Polyhedral cells, multi-cell polyhedrons and cell networks were developed in response to some of the key limitations of 3D graphics. These three datastructure are the fundamental building blocks for expanding the range of structural design applications using 3D graphic statics. Based on this theoretical foundation, the next chapter presents **`compas_3gs`**, which explains how the various concepts introduced in this chapter are implemented in an interactive design environment.

5 compas_3gs

The presented computational framework is implemented in an open-source library, **compas_3gs**. This chapter presents the general implementation methodology, organisational structure and key features of **compas_3gs**. Development of interfaces for integrating **compas_3gs** with a CAD software is also presented, which allows the computational framework to be used in an interactive design environment.

5.1 Introduction

This section provides some basic information about the library, explains the general approach, and gives an overview of the organisational structure of **compas_3gs**.

5.1.1 About

compas_3gs is an additional package for the COMPAS framework (Van Mele et al., 2017). COMPAS is an open-source, Python-based computational framework for collaboration and research in architecture, engineering and digital fabrication. **compas**, the main library of the COMPAS framework, contains all of the basic datastructures, algorithms, utilities and functionality. Building on this core library, **compas_3gs** provides additional features that are specifically geared towards 3D graphic statics applications.

Developed entirely independent of the functionality of CAD software, **compas_3gs** can be used on different platforms and in combination with external software and libraries. It can also take advantage of the extensive libraries for design and research that are widely available in the Python ecosystem. Because it is not dependent on any other software, **compas_3gs** in combination with **compas**, is intended to bring 3D graphic statics to a wide audience with diverse range of academic backgrounds, expertise and experience.

5.1.2 General approach

The functionalities and algorithms of **compas_3gs** are mainly based on transparent, geometry-based solvers and optimisation techniques as opposed to "black-box," numerical methods. **compas_3gs** is developed with the following goals in mind:

1. **Flexibility**

   The implementation needs to be general and flexible enough to cover a wide range of both known and unknown structural typologies. It also needs to have as few software dependencies as possible, so that users from a variety of backgrounds and expertise can adapt the library for various applications regardless of the CAD software being used.

2. **Simplicity**

   In computational geometry, improvement of the computational efficiency of solving procedures and algorithms are often prioritised over the user's ability to modify and interact with the resulting solutions. Instead of focusing on computing the absolute solution in the shortest amount of time possible, the solution as well as the procedure should communicate complex information in simple ways that are easy to understand and potentially provide meaningful insights.

3. **Customisability**

   During early stages of design, it may be desirable to explore multiple feasible solutions as rapidly as possible while meeting the requirements that are specific to the design problem at hand. This requires a set of flexible yet robust functions and operations, which can be easily mix-and-matched to create customised toolbars and workflows that are tailored for the needs of the user.

4. **Open source**

   `compas_3gs` is developed as an open-source library, encouraging researchers from a wide range of disciplines and expertise to make contributions that all users can benefit from. In order to incentivise the researchers to contribute their work, each contribution to `compas_3gs` is treated like a publication which can be cited and referenced. The concept and philosophy of this open source framework are modelled after the CGAL Project (2018).

5.1.3 Library structure

The contents of **compas_3gs** is organised as follows:

- **Datastructure**

  This folder inherits the main datastructures of the COMPAS framework, and incorporates additional functionalities and properties that are tailored for 3D graphic statics applications.

- **Utility**

  This folder contains utility and helper functions that are necessary for geometric and topological operations on polyhedral cells and multi-cell polyhedrons (i.e. computing the oriented normal, re-sizing of faces with target areas, etc.).

- **Operations**

  This folder contains transformation and manipulation functionalities (i.e. face pull and tilt, cell subdivisions, splitting faces, etc.).

- **Algorithms**

  This folder contains the algorithms and iterative solvers that perform mass operations (i.e. planarisation, reciprocation, arearisation, etc.).

- **CAD helpers**

  This folder contains wrappers, display and control functionalities that facilitate the implementation of **compas_3gs** in the interactive design environments of various CAD software.

5.1.4 Online documentation

Full documentation of the **compas_3gs** library is available online (https://compas-dev.github.io/compas_3gs/). The online documentation is designed to be a manual and reference for users, researchers and practitioners. It contains examples, tutorials, full API reference, and guide for developers and contributors.

5.1.5 Public release

The first announcement of the public release of **compas_3gs** was made in August, 2018 at the IASS (International Association for Shell and Spatial Structures) Symposium in Boston, USA. During the "Computational Graphic Statics using COMPAS" workshop, the main contents of this research were presented to the attendees. Main concepts of the theoretical background and the computational framework were presented, and some of the key features of **compas_3gs** were showcased. The official public release of **compas_3gs** is scheduled for June of 2019.

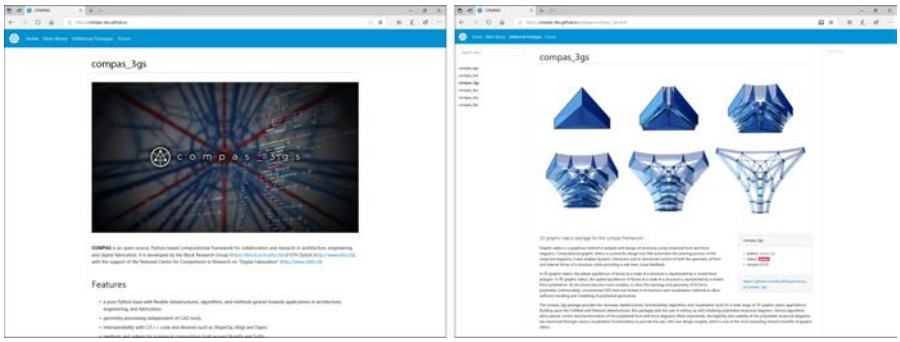


Figure 5.1: Online documentation of **compas_3gs**.

Figure 5.2: Introduction and announcement of **compas_3gs** by the author during the "Computational Graphic Statics using COMPAS" workshop at the IASS 2018 Symposium.

5.2 Datastructures

The main library of the COMPAS framework, **compas**, is based on three main datastructures which are designed for a wide range of research and design applications in architecture and structural engineering. These datastructures can be used in different combinations for various 3D graphic statics applications.

5.2.1 Types

Network

A **network** is a directed graph, made up of a finite set of vertices, which are connected by a finite set of edges (Figure 5.3-a). In mathematics, graphs are abstract topological datastructures used to model and depict relationships between pairs of objects. In **compas**, the geometry of a **network** is defined by adding xyz coordinate information to each of the vertices. A **network** can be 2D on a plane, or 3D in space. A **network** does not carry any face information.

Mesh

A **mesh** is a network of faces, which has a halfedge datastructure (Figure 5.3-b). A **mesh** can be open with boundary edges (polyhedral mesh or surface), or closed without any boundary edges. As implemented in **compas**, a **mesh** is strictly a 2-manifold; an edge is shared by no more than two faces. In essence, a **mesh** is a **network** with vertices and edges, with the addition of face information. The edges of a **mesh** are directed, although the directions of the edges are arbitrary and not used for querying or traversing the datastructure. A face of a **mesh** is defined by an ordered list of vertices, and need not necessarily be planar (flat) in its geometry.

Volmesh

A **volmesh** is a 3-manifold volumetric mesh; an edge of a **volmesh** can be shared by more than two faces. Embedded within this volumetric mesh is a network of cells, where the cell-to-cell relationships are defined by a combination of halffaces and planes (Figure 5.3-c). Similar to a **network** or **mesh**, a **volmesh** has a finite set of vertices and edges. The edges of a **volmesh** are directed, although the directions of the edges are arbitrary and not used for querying or traversing the datastructure. Whereas every edge of a **mesh** is split into two halfedges, every face of a **volmesh** is split into two halffaces. Because the edge directions are arbitrary, a face of a **volmesh** does not have a winding direction. Locally, each cell of a **volmesh** is structured like a **mesh**.

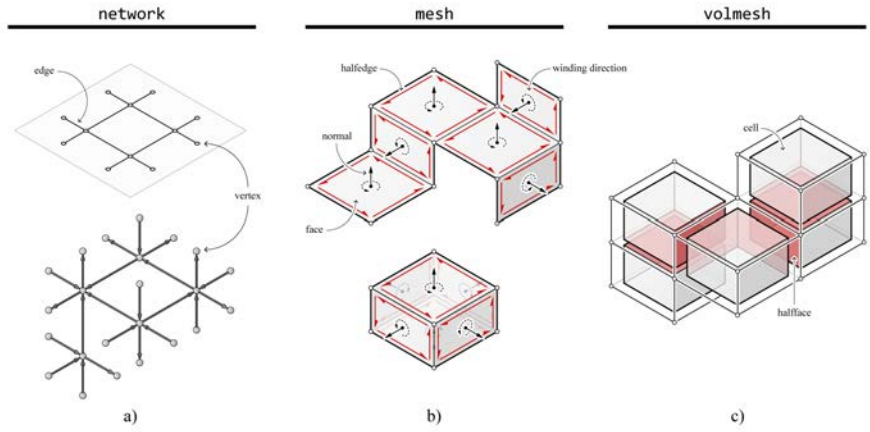


Figure 5.3: The three main datastructure of **compas**: a) **network**s: 2D on a plane (top) and 3D in space (shown with thickened arrows and spheres for clarity only); b) **mesh**es: an open, polyhedral surface with boundary edges (top), and a closed, polyhedral cell with no border edges (bottom); and c) a **volmesh**.

5.2.2 3D form and force diagrams

The three datastructures, **network**, **mesh** and **volmesh** can be used to represent the form and force diagrams used in 3D graphic statics.

A polyhedral cell which is typically used for the representation of a global force polyhedron $\Lambda^{\perp}$, is modelled as a **mesh** (Figure 5.4-a). The initial form diagram Λ, which contains the information of the magnitudes and locations of the external forces, are typically a disconnected set of vectors or "lines" in space. These vectors can be moved to one location such that the heads of the vectors are coincident. The consolidated vectors then can be modelled as a **network**, where each external force is represented with an edge.

A multi-cell polyhedron $\Gamma^{\perp}$ and its corresponding polyhedral form diagram Γ, are modelled as **volmesh** and **network**, respectively (Figure 5.4-b). Because $\Gamma^{\perp}$ and Γ are topologically dual to each other, Γ can also be modelled as a **volmesh**. However, because the topological information of Γ is embedded in $\Gamma^{\perp}$, some information do not need to be duplicated. For example, the face and cell information of Γ can be computed from the topology of $\Gamma^{\perp}$. The polyhedral form diagram Γ as a **network** is essentially a visualisation of the cell connectivity information of $\Gamma^{\perp}$.

A cell network $\Psi^{\perp}$, which contains the form diagram Ψ and a nodal polyhedral cell at each node, is modelled as a **network** (Figure 5.4-c). Because Ψ and $\Psi^{\perp}$ are intrinsically linked and are inter-dependent, a few algorithms need to enforce the necessary global constraints. The EGI of each node ensures that the topology and geometry of the individual cells are correct and up to date. The arearisation algorithm ensures that all pairs of adjacent cells have contact faces with equal areas. Finally, the reciprocation algorithm, enforces perpendicularity between Ψ and $\Psi^{\perp}$. These algorithms will be presented in the next section.

5.2.3 Computational interpretation of diagrams

In Section 4.2.7.1, it was shown how the interpretation of the force in a member as either tension or compression can be made locally for a single node, by comparing the orientations of the member at that node and the normal of the corresponding face in the force diagram. In a **network** representation of a form diagram, the directed edges can also be used for the interpretation of forces.

In 2D and 2.5D applications of computational graphic statics, a directed edge of the form diagram $e_{i\to j}$ and the corresponding directed edge $e^{*}_{i\to j}$ in the force diagram are parallel and have the same orientations for tensile forces or members (Van Mele and Block, 2014). If they are parallel but have opposite orientations, $e_{i\to j}$ is in compression. In 3D graphic statics, the same interpretation can be made by comparing the orientations of the directed

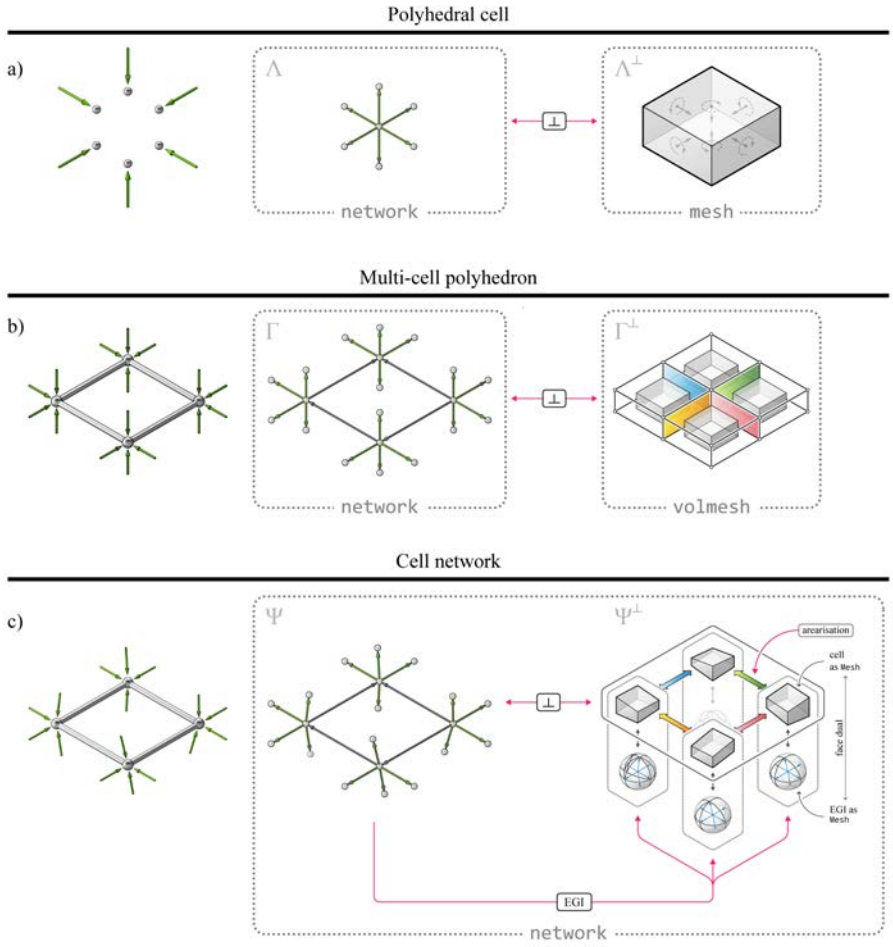


Figure 5.4: Representation of the three types of form and force diagrams using **`compas`** datastructures: a) a group of equilibrated external forces Λ and the corresponding global force polyhedron $\Lambda^{\perp}$ modelled as a **`network`** and **`mesh`**; b) a polyhedral form diagram Γ and the corresponding multi-cell polyhedron $\Gamma^{\perp}$ modelled as a **`network`** and **`volmesh`**; and c) a cell network $\Psi^{\perp}$ and its non-polyhedral form diagram Ψ modelled together as a **`network`**.

edges in the form diagram and the normals of the corresponding "directed faces."

For a **`mesh`** representing a polyhedral cell, the directed face is simply the face of the **`mesh`** that corresponds to its dual edge. However, a **`volmesh`** representing a multi-cell polyhedron does not have face information in its datastructure. For a **`volmesh`**, a directed face that corresponds to $e_{i \to j}$ is equivalent to the halfface of the i-th cell of $\Gamma^{\perp}$, whose paired halfface belongs to the j-th cell. In the example shown in Figure 5.5, the directed face of directed edge $e_{0 \to 2}$ is the halfface of c_0^*, whose paired halfface belongs to

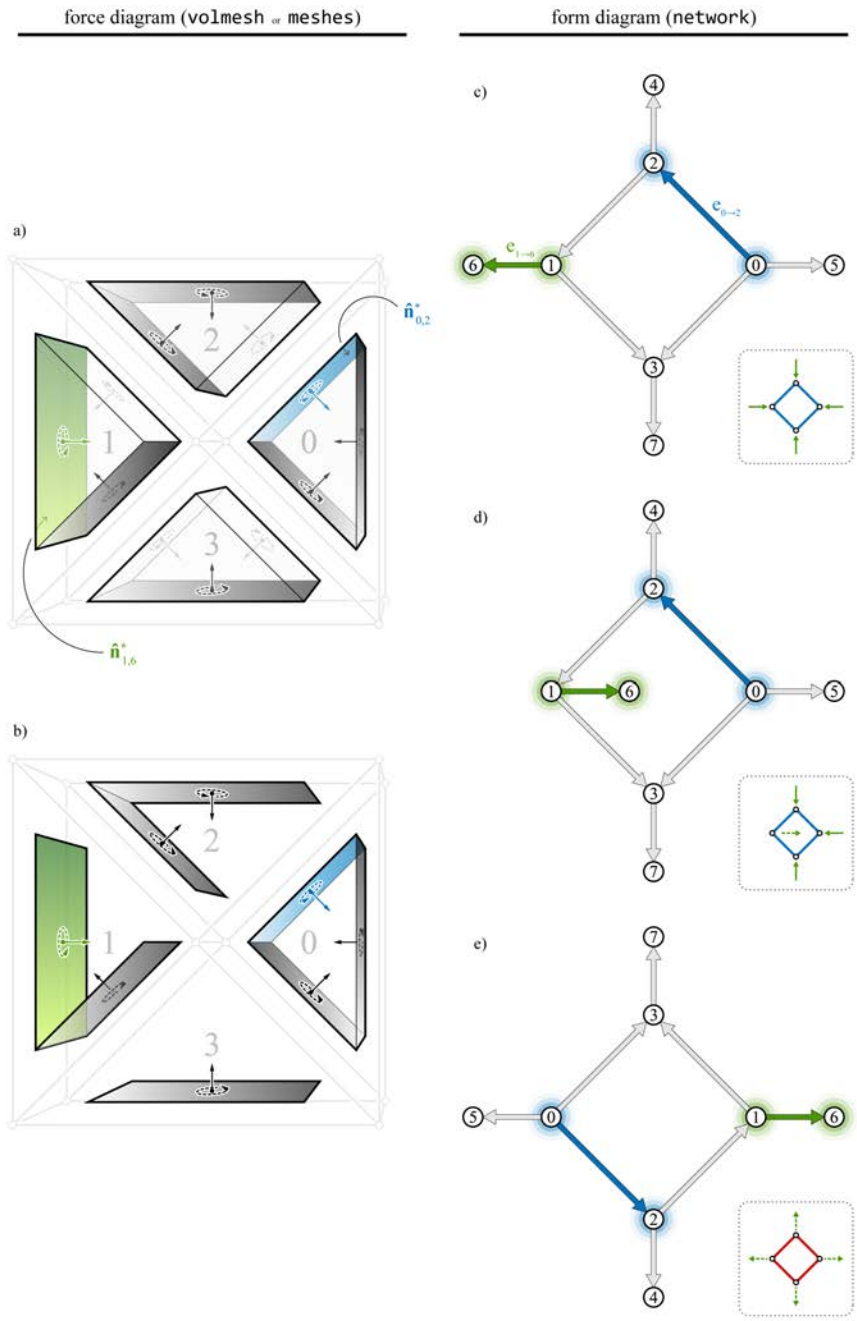


Figure 5.5: Interpretation of the forces using the directed edges of the **network** and the corresponding directed faces of the **mesh** or **volmesh**: a) a multi-cell polyhedron $\Gamma^{\perp}$ with four prismatic cells; b) the "directed faces" of $\Gamma^{\perp}$; c), d) and e) various configurations of the form diagram Γ, where perpendicularity is enforced but the force distributions are different.

c_2^* (highlighted in blue). For $e_{1\to6}$, the directed face is simply the halfface of c_1^* that corresponds to that edge (highlighted in green). Figure 5.5-c, d, and e are three different configurations of Γ where each directed edge maintains the same connectivity and perpendicularity with the corresponding directed faces, but have different force distributions.

5.3 Key algorithms

This section presents some of the key features and algorithms of **`compas_3gs`**, which enable interactive transformations of the form and force diagrams while enforcing the planarity and perpendicularity constraints.

5.3.1 Planarisation

At any point during the design process, the faces of polyhedral cells or multi-cell polyhedrons may become non-planar. Before any further design explorations can be made, the non-planar faces need to be planarised. Planarisation of non-planar faces can be formulated as an iterative projection method, which is a tried-and-tested methodology in computational geometry (Liu et al., 2006; Bouaziz et al., 2012; Deuss et al., 2015).

Figure 5.6 shows an example application of the planarisation algorithm, implemented using the iterative projection method. At each time step, each face is projected onto either: the plane defined by its initial normal and the current centroid; a plane defined by a target normal; or a best-fit plane computed from its current vertex coordinates. Because faces are projected independently from one another, there will be multiple coordinates for a single vertex at the end of each time step. The average or the barycenter of the coordinates of a vertex is its new location for that time step. The procedure continues until a desired tolerance has been reached.

If there are no constraints enforced, each face projects itself to the plane defined by its current centroid and normal (Figure 5.6-b). In some 3D graphic statics applications, specified faces may need to stay fixed in their orientations. For example, some of the boundary faces of a multi-cell polyhedron corresponds to externally applied loads, which typically do not change in their magnitudes or locations during the design process. In addition, it may sometimes be desired to fix the orientations of certain members in the form diagram.

The orientation constraint can be enforced by updating the **`'fix_normal'`** attribute of the specified faces. Individual vertex constraints can also be set by updating the **`'x_fix'`**, **`'y_fix'`**, **`'z_fix'`** attributes for the specified vertices. Figure 5.6-c shows the planarisation, but this time with faces 2, 1 and 5 constrained to be perpendicular to the x, y and z axes, respectively.

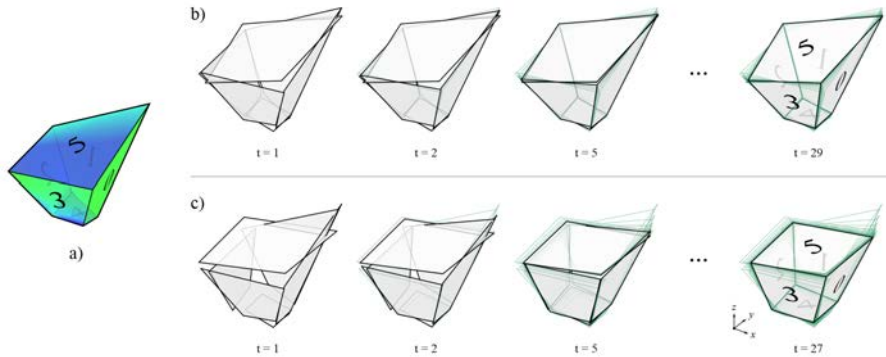


Figure 5.6: Planarisation of a polyhedral cell with non-planar faces: a) unconstrained planarisation, where the best-fit planes is used for each face at every iteration; and b) constrained planarisation, where some of the faces are given target normal vectors (faces 2, 1 and 5 are constrained to be perpendicular the x, y and z axes, respectively).

5.3.2 Arearisation

Controlling the magnitudes of forces in 2D force diagrams is straightforward and intuitive, since the vertices can be easily constrained to and moved around on the plane. However, in 3D graphic statics, modifying the magnitudes of forces is not as simple since polyhedral face areas have to be controlled. There is no clear and intuitive means for a user to control the area of a polyhedral face through translation of vertices or edges. This section presents the arearisation algorithm, which allows users to more precisely control the face areas of force diagrams, and consequently the magnitudes of the forces in the structure.

5.3.2.1 Arearisation of polyhedral cells

It was demonstrated in Section 4.2.8 how a face of a polyhedral cell can be pulled along its normal while maintaining the initial orientations of all the other faces. As a face is pulled, the distribution of forces changes. However, the precise amount of change in the area of the pulled face is not visually quantifiable, and it is not immediately apparent how the operation will affect the face's oriented normal. Rather than pulling the faces in arbitrary amounts, finding the new face location such that the resulting face area matches a target value will enable a more force-controlled geometric operation.

For the symmetric polyhedral cells shown in Figure 5.7, the area $A^*_{i,0}$ of face $f^*_{i,0}$ as a function of its position along its normal can be directly formulated. For the fully symmetric case shown in Figure 5.7-a, $f^*_{i,0}$ is a square, and remains as a square regardless of its z position. It can also be observed here that there are two values of z that result in the same face area as well

as the same normal direction. For a partially case shown in Figure 5.7-b, there are also two values of z that result in the same face area, but with opposite normal directions. In this particular case, the function of $A^*_{i,0}$ is linear, with the oriented normal changing directions around the inflection point (Figure 5.8-a). The example shown in Figure 5.7-c is a case where the face area remains the same regardless of its position z. Changing the area of this face is only possible by scaling the entire polyhedral cell, or allowing the orientations of other faces to change.

For more general, non-symmetric cells, a face's area as a function of its z position along its normal is not linear (Figure 5.8-b, c, d). As evident by the plots of $A^{*(z)}_{i,0}$ for each example, the behaviour of the function is not immediately apparent through a visual inspection of the geometry of the polyhedral cell. However, since the function is parabolic, and therefore unimodal, a simple minimisation procedure can be used to determine the target z value for a face, given a target area $A^{*,target}_{i,0}$.

For this minimisation procedure, the Golden-section search method (Kiefer, 1953) is used, which is a simple technique for finding the extremum (minimum or maximum) of unimodal functions by iteratively narrowing the range of values $[a, b]$ inside which the extremum exists:

$$\begin{aligned} \text{Minimise} \quad &: \quad f(z) \;=\; \mid A^{*,target}_{i,j} \;-\; A^{*(z)}_{i,j} \mid \\ \text{subject to} \quad &: \quad a \;\leq\; z \;\leq\; b \end{aligned} \tag{5.1}$$

First, in order to determine which direction (either in the direction of the face normal, or the opposite) cause the area of the face to increase, a face is pushed in the direction of its normal a small amount. If the area increases, then the

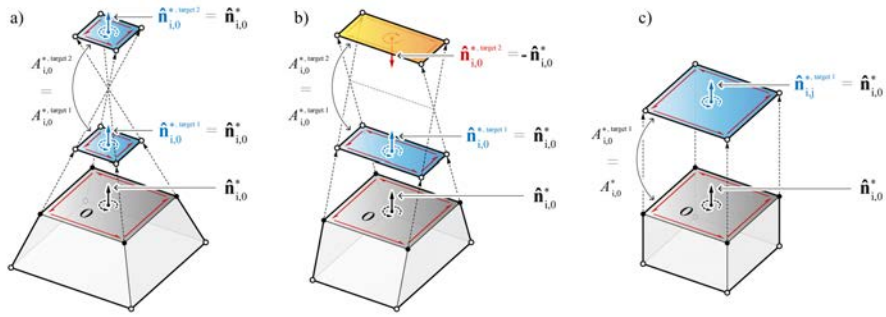


Figure 5.7: Possible solutions of face pull operation with target face areas, in both positive and negative domains: a) symmetric complex cell, where $\hat{\mathbf{n}}^*_{i,0}$ does not change on either side of the inflection point; b) partially symmetric complex cell, where $\hat{\mathbf{n}}^*_{i,0}$ changes in its orientation above the inflection line; and c) case where all of the trailing edges of $f^*_{i,0}$ are perpendicular to the face, and therefore moving $f^*_{i,0}$ along its normal does not change its area.

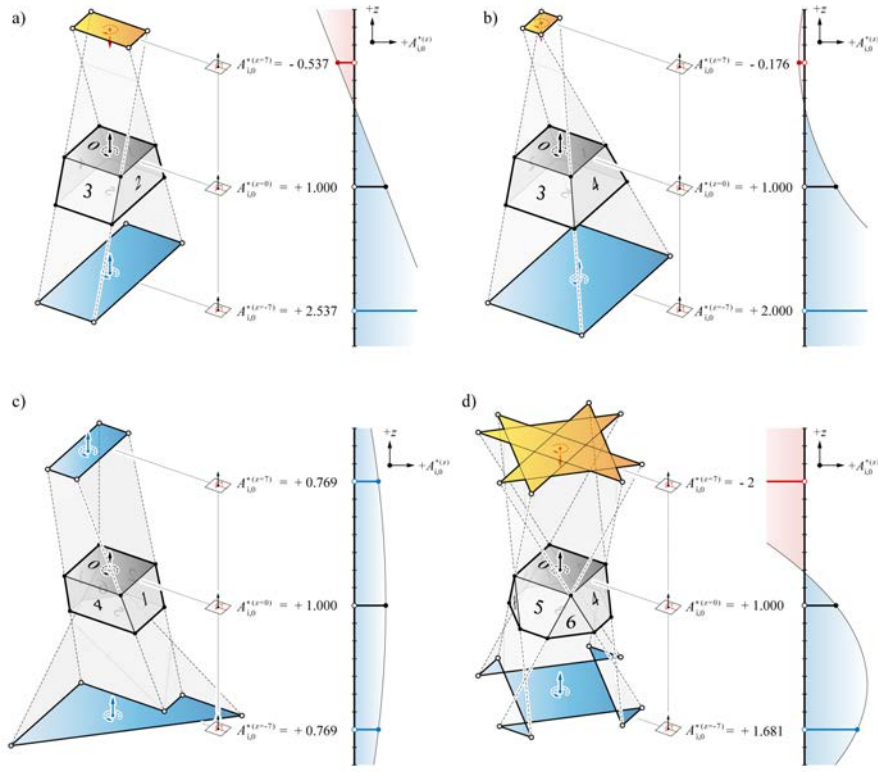


Figure 5.8: The relationship between the face area and its relative normal position z for various cell types: a) a partially symmetric cell where $A^*_{i,0}$ as a function of z is linear; b) a cell and $f^*_{i,0}$ with only 3-valent vertices; c) a cell and $f^*_{i,0}$ with a 4-valent vertex; and d) a cell and $f^*_{i,0}$ with only 4-valent vertices.

$+z$ direction is in the direction of the face normal. If the area decreases, then the $+z$ direction is in the opposite direction of the face normal. If there is no change in the area, the search is aborted as this suggest a case where all the trailing edges are perpendicular to the face.

Once the $+z$ direction has been determined, the initial search range $[a, b]$ can be defined, where a is the lower bound and b is the upper bound. b can be manually defined, or a multiple of $A^{*,target}_{i,j}$ can be used to have an initial range that is reasonably proportional to the difference between $A^{*,target}_{i,j}$ and $A^{*(z=0)}_{i,j}$ Then, the objective function is evaluated at c and d, where $c = b - (b - a)\phi$ and $d = a + (b - a)\phi$, and ϕ is the "Golden ratio":

$$\phi = \frac{1 + \sqrt{5}}{2} \tag{5.2}$$

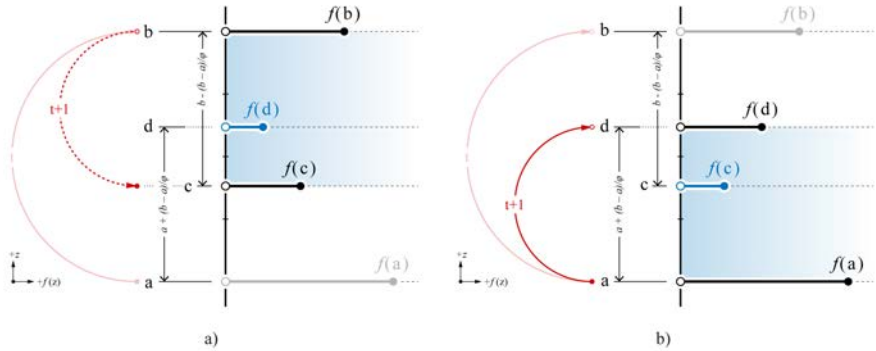


Figure 5.9: The Golden-section search method: a) if $f(d) < f(c)$, the range of search for the next time step $t+1$ is $[c, b]$; and b) if $f(d) > f(c)$, the range of search for the next time step $t+1$ is $[a, d]$.

If $f(d) < f(c)$ at step t, the range of search for the time next step $t+1$ is $[c, b]$. If $f(d) > f(c)$ at step t, the range of search for the next time step $t+1$ is $[a, d]$. This loop continues, until the difference between c and d is less than the tolerance entered.

As an input, the user can enter either a positive or negative value as the target area. A positive number will mean that the normal of the resulting face will remain in the same direction as its initial normal direction. A negative number will mean that the normal of the resulting face will be in the opposite direction as its initial normal direction. As shown in Figures 5.7-a and 5.8-c, it is possible that there may be two values of z that results in the target face area. In such cases, the smaller z value will be returned unless specified otherwise. This iterative procedure for the arearisation of a single face of a polyhedral cell is adopted from Press et al.'s formulation of Golden-section search method (2007).

This procedure is applied to $f^*_{i,0}$ of the polyhedral cell shown in Figure 5.10. The initial area of $f^*_{i,0}$ at $z = 0$ is 1. A target area $A^{*,target}_{i,0}$ of 2 is given. With an initial search range of $[0, 6]$ and tolerance of $1\mathrm{e}{-6}$, the final z value of 2.140 is output after 31 iterations, with $A^{*(t=31)}_{i,0} = 19.99972$.

Because arearisation algorithm allows a more precise control of the areas of individual faces of a cell, and therefore the magnitudes of the forces in the corresponding members in the form diagram, it can be used to explain the concept of static indeterminacy of spatial structures. It can also demonstrate that polyhedral reciprocal diagrams are not limited to the exploration of statically determinate spatial structures, but also various possible equilibrium solutions for statically indeterminate structures (Kilian and Ochsendorf, 2005; Block, 2005; Van Mele et al., 2012).

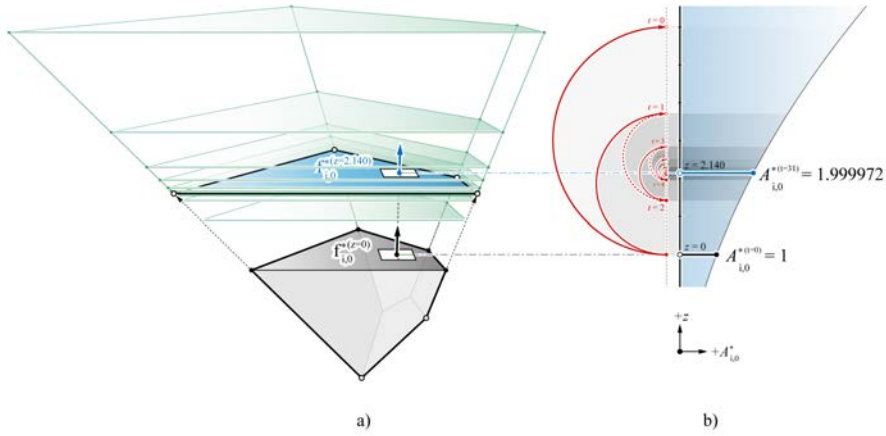


Figure 5.10: Arearisation of a face using the Golden-section search: a) visualisation of the face at each time step; and b) the plot of the function of $A^{*(z)}_{i,0}$ and the search process.

In an indeterminate structure, the distribution of forces among the members of the structure is highly dependent on the boundary conditions, imperfections of the building components and the tolerance accumulated during the assembly on site. Therefore, the actual internal stress state is unknown, difficult to predict and is sensitive to minor changes in the boundary conditions. Using polyhedral force diagrams and the arearisation algorithm, indeterminate states of equilibrium can be visualised and described. In the context of structural design, this indeterminacy can be exploited to explore and obtain different internal equilibrium states.

Figure 5.11-a shows a polyhedral cell for an elastic stress state of a five-bar structure with one applied load $\mathbf{f}_{i,5} = P$, where all of the bars have the same axial stiffness EA. However, if some bars were slightly shorter than intended due to some imperfections during the fabrication process, the tensile forces in those bars would be higher. For example in Figure 5.11-b, bars 1 and 4 are relatively shorter, and therefore the tension forces of $\mathbf{f}_{i,1}$ and $\mathbf{f}_{i,4}$ are slightly higher. On the other hand, if some bars are slightly longer than intended, their internal forces may go under compression while other bars compensate by carrying higher tensile forces (Figure 5.11-c, d, e).

In certain cases, asymmetric distributions of forces may cause some bars to carry no force at all, and the applied load P will be carried by only four of the five bars. Small changes in the boundary conditions, fabrication imperfection and construction tolerance can potentially cause significant changes in the internal stress distribution of indeterminate structures. In contrast to conventional numerical calculations and finite element analysis of structures, polyhedral reciprocal diagrams can visualise and describe different states of internal stress distribution. Consequently, it can be used not only for design

explorations, but also to explain and investigate static indeterminacy, which is traditionally a difficult concept to understand through just numbers or analysis data.

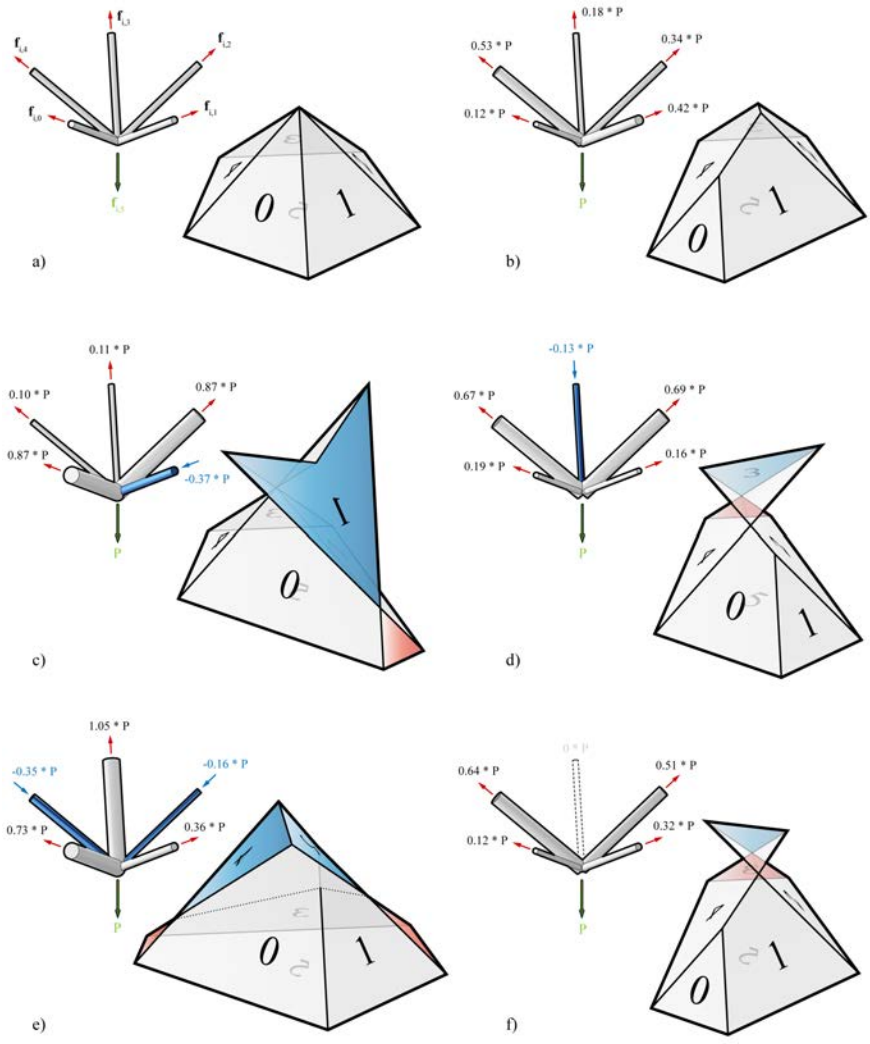


Figure 5.11: An indeterminate, single-node structure with five tensile members with one applied load $\mathbf{f}_{i,5}$ = P (the area of $A^*_{i,5}$ is 1 in all examples), can have different stress states depending on the boundary conditions: a) the initial, elastic stress state when all five bars are under the same amount of tensile force; b) a stress state where tensile forces $\mathbf{f}_{i,1}$ and $\mathbf{f}_{i,4}$ are much greater than the others; c) a stress state where $\mathbf{f}_{i,1}$ is now in compression, with $\mathbf{f}_{i,0}$ and $\mathbf{f}_{i,2}$ carrying most of the tensile loads; d) a stress state where $\mathbf{f}_{i,3}$ is now in compression, with $\mathbf{f}_{i,2}$ and $\mathbf{f}_{i,4}$ carrying most of the tensile loads; e) a stress state where $\mathbf{f}_{i,2}$ and $\mathbf{f}_{i,4}$ are in compression, with $\mathbf{f}_{i,0}$ and $\mathbf{f}_{i,3}$ carrying most of the tensile loads; and f) a stress state where $f^*_{i,3}$ is complex and an oriented area of 0, and therefore the magnitude of $\mathbf{f}_{i,3}$ is 0.

5.3.2.2 Arearisation of multiple faces

The arearisation of a single face uses the combination of the face pull operation and a minimisation procedure to determine the new location of a single face based on a given target area for that face. Because the orientations of the faces are fixed, it is ideal for controlling the magnitude of the force in a single member and exploring different equilibrium states for a group of members with fixed orientations. However, it is not ideal for scenarios where the areas of multiple faces need to be controlled simultaneously. Controlling the areas of multiple faces at once can be useful in addressing boundary conditions where the applied loads are known and the magnitudes of the reaction forces are specified in advance, or in a multi-cell polyhedron where multiple faces may need to be constrained at once.

Instead of pulling each face independently towards its target location, each face can be iteratively re-sized until all of the faces have reached their target areas. At each iteration, the faces are re-sized individually and then new vertices are computed for the cell. Re-sizing a face with a target area can be formulated as a polygon scaling problem. The area of each face can be determined using the procedure described in Section 4.2.6.2. The procedure is visualised in Figure 5.12 as a reference for the following series of equations. The area of the n-th sub-triangle $f^*_{i,j,n}$ is:

$$A^*_{i,j,n} = q \cdot A^*_{i,j} = \frac{1}{2} \cdot |\mathbf{r}_n| \cdot |\mathbf{r}_{n+1}| \cdot \sin\beta \tag{5.3}$$

where q is the ratio of the area of the n-th sub-triangle $A^*_{i,j,n}$ to the total area $A^*_{i,j}$, and β is the angle between $\mathbf{r}_n$ and $\mathbf{r}_{n+1}$ (Fig 5.12-b). If the face $f^*_{i,j}$ is scaled by a factor of s from $\mathbf{b}_{i,j}$ to satisfy the target face area of $A^{*,target}_{i,j}$, the new area of the n-th sub-triangle $f^*_{i,j,n}$ is:

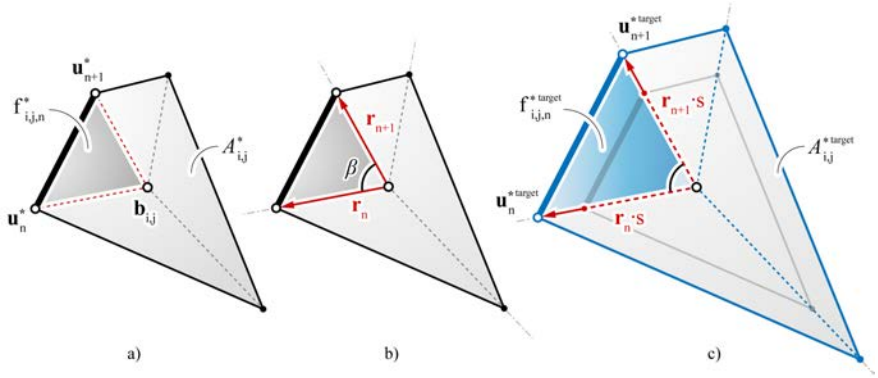


Figure 5.12: Face re-sizing with respect to a target area: a) face partitioned into sub-triangles; b) two new inner edges and the angle β that can be used to determine the area of each sub-triangle; and c) the new face now with the target area, scaled by factor of s.

$$\begin{aligned} A^{*,target}_{i,j,n} &= q \cdot A^{*,target}_{i,j} \\ &= \frac{1}{2} \cdot |\mathbf{r}_n \cdot s| \cdot |\mathbf{r}_{n+1} \cdot s| \cdot \sin\beta \\ &= \frac{1}{2} \cdot s^2 \cdot |\mathbf{r}_n| \cdot |\mathbf{r}_{n+1}| \cdot \sin\beta \end{aligned} \tag{5.4}$$

Solving for q from equations 5.3 and 5.4, the scale factor can be obtained as:

$$s = \sqrt{\frac{A^{*,target}_{i,j}}{A^{*}_{i,j}}} \tag{5.5}$$

In certain applications, it is possible for faces to have target areas of zero (see Section 4.2.9.2). Using the scale factor s from Equation 5.5, faces with target area of zero would collapse to a point. However, it is possible for a face with zero area to collapse into a point or and edge. It is also possible for a complex face to have an area of zero. It is generally assumed that the eventual shape of the faces with target areas of zero is unknown, and therefore the face should not be scaled to a point. Faces with target areas of zero are omitted during the iterative procedure.

Once the scale factor s has been determined for each face, each face can be scaled independently from one another at each time step, and the new coordinates for a vertex can be averaged to determine its new location. This procedure is equivalent to the planarisation algorithm, except that the faces are now scaled prior to being projected onto the target planes. The target plane can be constrained by target normals, or re-defined for each time step as the best-fit plane from its current vertex coordinates. Allowing faces to be adaptive to the new best-fit target planes allows the faces to reorient themselves to satisfy the given force distribution constraints, and therefore output a new form-found geometry of the structure.

In addition to force-driven form finding, the arearisation algorithm can also be used for the construction of polyhedral cells from a given set of force magnitudes and orientations (see Section 4.2.9.2). Because the orientations of the faces are fixed in this application, the construction of EGI and the unit cell with zero faces is necessary before arearisation can be applied.

Figure 5.13 shows construction sequences of polyhedral cells from their face normals and areas. The first column shows the equilibrated force vectors as spike models. The second column shows the corresponding EGIs with adjacency and cross-adjacency arcs. The third column shows the unit cell with their zero faces highlighted in orange. Finally, the last column shows the final geometry of the polyhedral cells with face areas that match the magnitudes of the corresponding force vectors.

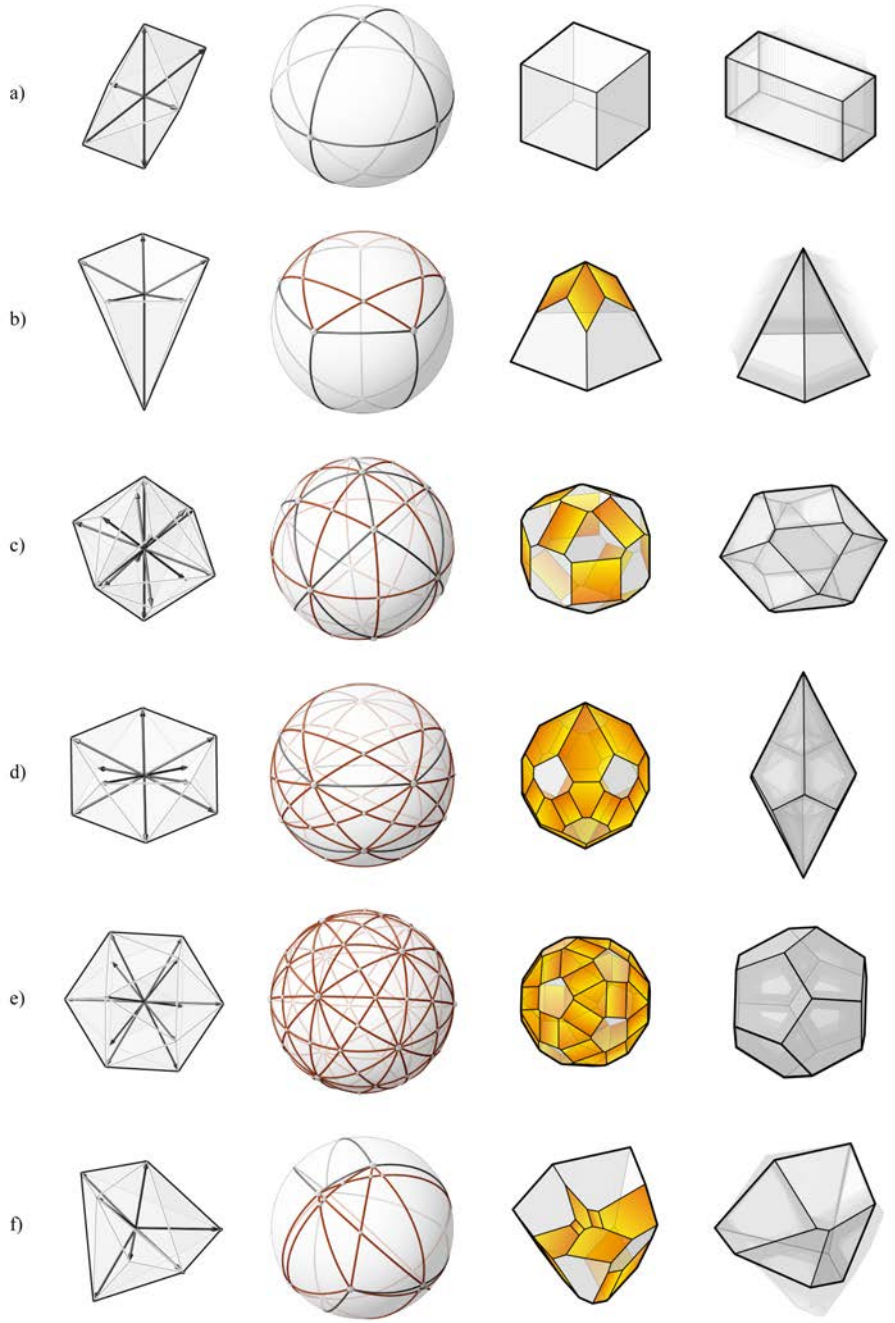


Figure 5.13: Construction of polyhedral cells from force magnitudes and orientations - the force vectors, the EGI, unit cell and the final geometry of the polyhedron (from left to right): a) rectangular box; b) pentagonal pyramid; c) tetradecahedron; d) pentagonal trapezohedron; e) dodecahedron; and f) irregular, asymmetric polyhedral cell.

5.3.3 Reciprocation

Another constraint that needs to be enforced is the perpendicularity of the members in the form diagram to the corresponding faces in the force diagram. The perpendicularity can be imposed through an iterative solver, similar to the method implemented for graphic statics applications in 2.5D (Rippmann et al., 2012) and in 3D (Akbarzadeh et al., 2015b). Detailed explanation of the procedure can be found in (Rippmann, 2016, pp. 120-125).

In `compas_3gs`, the perpendicularity between the polyhedral form and force diagrams is enforced by the reciprocation algorithm. The reciprocation algorithm adds several new layers, such as the weight factor, constraining edge lengths, and incorporating tensile members. Similar to the planarisation algorithm, basic geometric constraints can be enforced manually by the user at any point during the reciprocation algorithm (such as vertex fixity, edge fixity, face orientations, etc.).

Based on the convention established by Van Mele and Block (2014), the directed edge $e_{i \to j}$ of the form diagram and the corresponding directed edge $e^*_{i \to j}$ of the force diagram are parallel and have the same orientations for tensile forces or members; their unitised vectors are equal:

$$\hat{\mathbf{e}}_{i \to j} = \hat{\mathbf{e}}^*_{i \to j} \tag{5.6}$$

In 3D graphic statics, the edges of the form diagram are perpendicular to the corresponding faces of the force diagram. For a tensile force or member, the $\hat{\mathbf{e}}_{i \to j}$ of the form diagram and the unitised normal $\hat{\mathbf{n}}^*_{i,j}$ of the corresponding face (for multi-cell polyhedrons, defined as the unitised normal of the halfface of i-th cell that is adjacent to the j-th cell) are equal:

$$\hat{\mathbf{e}}_{i \to j} = \hat{\mathbf{n}}^*_{i,j} \tag{5.7}$$

In certain applications, it may be desirable to control how much the form and force diagrams are changing relatively to each other during the form-finding procedure. Before the iterative process, a target vector $\mathbf{t}_{i,j}$ can be determined by using the initial edge and face normal orientations, and a weight factor γ. Then, the edges of the form diagram and the faces of the force diagram re-orient themselves towards $\mathbf{t}_{i,j}$. The weighting factor γ is a value between 0 and 1, and it determines how much each diagram changes. For example, $\gamma = 1$ would adjust the edges of the form diagram only, whereas $\gamma = 0$ would reorient the faces of the force diagram only.

$$\mathbf{t}_{i,j} = \gamma \, \hat{\mathbf{e}}_{i \to j} + (1 - \gamma) \, \hat{\mathbf{n}}^*_{i,j} \tag{5.8}$$

Once the target vectors for each face and the corresponding edge have been determined, the force diagram can be adjusted by using the planarisation algorithm, using $\mathbf{t}_{i,j}$ as the target normal of the projection plane for each face. Before the edges of the form diagram can be reoriented, the current directions of $\hat{\mathbf{e}}_{i\to j}$ relative to $\hat{\mathbf{n}}^*_{i,j}$ need to be checked. This check determines whether or not the edge is in tension or compression, and therefore if the edge and the corresponding normal need to be parallel only, or parallel as well as having the same orientation. The relative direction of $\hat{\mathbf{e}}_{i\to j}$ and $\hat{\mathbf{n}}^*_{i,j}$ can be evaluated by their dot product. Using this dot product, the edge-to-face direction coefficient λ can be determined. A negative dot product means they are pointing in the opposite directions (λ=-1), while a positive dot product means they are pointing in the same direction (λ=1). With the edge to face normal direction coefficient λ, Equation 5.8 updates to:

$$\mathbf{t}_{i,j} = \gamma\, \hat{\mathbf{e}}_{i\to j} + (1-\gamma)\, \lambda\, \hat{\mathbf{n}}^*_{i,j} \tag{5.9}$$

For example in Figure 5.14-a, $\hat{\mathbf{e}}_{i\to j}$ and $\hat{\mathbf{n}}^*_{i,j}$ are generally pointing in the same direction, which would signify that $e_{i\to j}$ is in tension in this scenario. However, if $e_{i\to j}$ was in compression, $\hat{\mathbf{e}}_{i\to j}$ and $\hat{\mathbf{n}}^*_{i,j}$ would be pointing in opposite directions. The updated $\mathbf{t}_{i,j}$ can be used to re-orient $e_{i\to j}$. By scaling unitised $\hat{\mathbf{t}}_{i,j}$ by the length of the corresponding edge $||\mathbf{e}^{(t)}_{i\to j}||$ at time step t, new coordinates for the two vertices $\mathbf{v}^{(t+1)}_i$ and $\mathbf{v}^{(t+1)}_j$ of $e^{(t+1)}_{i\to j}$ can be computed:

$$\begin{aligned} \mathbf{v}^{(t+1)}_i &= \mathbf{v}^{(t)}_j - \hat{\mathbf{t}}_{i,j} \cdot ||\, \mathbf{e}^{(t)}_{i\to j}\, || \\ \mathbf{v}^{(t+1)}_j &= \mathbf{v}^{(t)}_i + \hat{\mathbf{t}}_{i,j} \cdot ||\, \mathbf{e}^{(t)}_{i\to j}\, || \end{aligned} \tag{5.10}$$

Edge length constraint can be imposed globally to all edges or locally for each edge by setting the lower bound ω_{lb} and upper bound ω_{ub}. If $||\mathbf{e}^{(t)}_{i\to j}||$ falls outside of the defined bounds, $||\mathbf{e}^{(t)}_{i\to j}||$ in can be replaced with either ω_{lb} or ω_{ub}. Once all the edges have been reoriented at each time step, there will be multiple new coordinates for each vertex. The centroid of the new coordinates for each vertex, is then the new vertex coordinate for that time step. The iterative process is repeated until a convergence tolerance is reached, i.e. when all of the edges of the form diagram are perpendicular to the corresponding faces of the force diagram within a specified maximum deviation value. One way to measure the deviation δ between $\hat{\mathbf{e}}_{i\to j}$ and $\hat{\mathbf{n}}^*_{i,j}$ is by evaluating their dot product and subtracting it from 1:

$$\delta = |\, 1 - |\, \hat{\mathbf{e}}_{i\to j} \cdot \hat{\mathbf{n}}^*_{i,j}\, |\, | \tag{5.11}$$

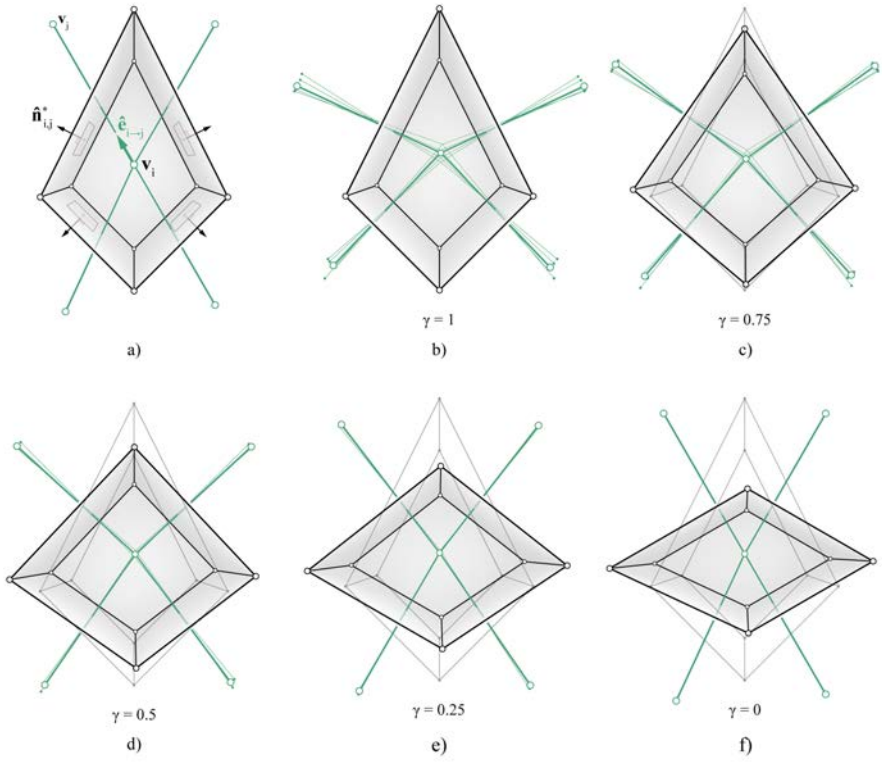


Figure 5.14: Reciprocation of a single prismatic polyhedral cell c_i^* with four edges in tension with varying values of γ (top view, with the faces that are parallel to the page removed): a) initial form and force diagrams; b) $\gamma = 1$, where only the form diagram is adjusted; c) $\gamma = 0.75$; d) $\gamma = 0.5$, where both c_i^* and the form diagram are adjusted equally; e) $\gamma = 0.25$; and f) $\gamma = 0$, where only c_i^* is adjusted.

If $\hat{\mathbf{e}}_{i \to j}$ and $\hat{\mathbf{n}}_{i,j}^*$ are parallel and in the same direction, the dot product will be 1 which would make $\delta = 0$. At the end of each iteration, δ can be compared to the value of tolerance parameter to either proceed with or terminate the algorithm.

5.4 CAD integration

This section presents **`compas_3gs.rhino`**, which is a CAD helper package for **`compas_3gs`**. It provides a user interface for processing, visualising and interacting with **`compas_3gs`** datastructures and functionality in the interactive CAD environment of Rhinoceros (Robert Mcneel & Associates, 1993). The overall set up of the computational design environment using **`compas_3gs`** is described. Key features of the three main components of **`compas_3gs.rhino`**, wrappers, display and control, are also presented.

5.4.1 Setup

The **compas_3gs** package of the COMPAS framework is implemented independent of the functionality provided by existing CAD software. However, CAD tools are still necessary for constructing and manipulating geometry, applying constraints interactively, making custom user interfaces and visualising script results. While the geometric computation is handled by **compas_3gs** in the background, existing CAD software can be used as the medium for processing, visualising and interacting with datastructures and geometrical objects (Figure 5.15).

The CAD software that was chosen for this dissertation was Rhinoceros, mainly due to its wide usage in architecture, an established user base and a large ecosystem of additional plugins and packages. The built-in visualisation and control functions of Rhinoceros are exploited to create an interactive environment that is specifically calibrated for 3D graphic statics design applications. CAD helper packages for other software can be developed in the future.

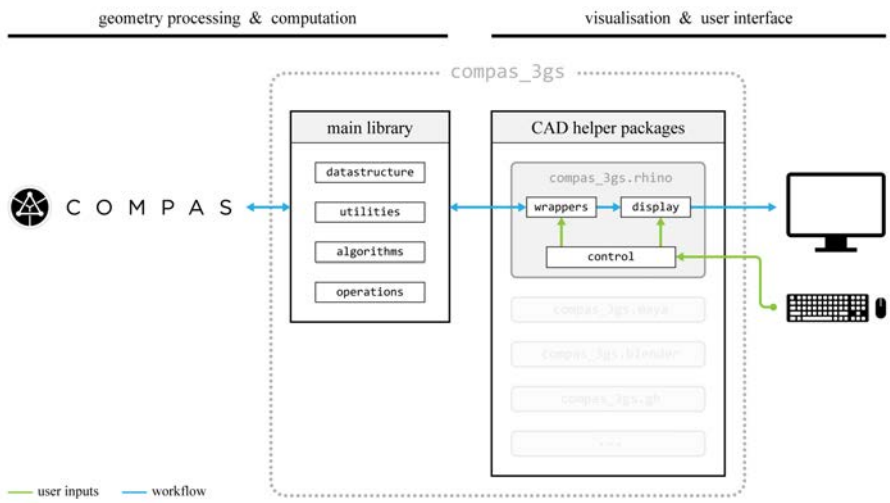


Figure 5.15: Graphical overview showing how **compas_3gs.rhino** is used as the visualisation and user interface for **compas_3gs**.

5.4.2 Wrappers

COMPAS and its additional packages, including **compas_3gs**, are implemented primarily in Python and designed to be used on different platforms and in combination with external software and various scientific and non-scientific Python libraries. The datastructures, utility functions, operations

and algorithms of **compas_3gs** are independent of CAD software. However, in order for the scripts to run and be visualised within existing CAD software, the contents of **compas_3gs** need to be wrapped with "wrapper" functions using the API of the CAD software. In addition to Python language, **compas_3gs.rhino** uses RhinoScript [1] and RhinoCommon API [2] for visualisation, the user interface features and exchange of data between **compas_3gs** and Rhinoceros. The components of **compas_3gs** can be accessed manually and executed using script editors in Rhinoceros, such as the built-in IronPython editor or the Python editor in Rhinocero's Grasshopper. The users can also create custom toolbars for Rhinoceros that compile specific combinations of **compas_3gs** commands and functions that are tailored for the design problem at hand.

5.4.3 Control

In 3D graphic statics, the polyhedral force diagram often contains multiple cells that are overlapping and on top of one another. Through the **compas_3gs.rhino** drawing functionalities, the polyhedral form and force diagrams are visualised as native Rhinoceros objects. Once drawn, these objects can be selected using the built-in Rhinoceros select functions. However, in a dense cluster of cells, selecting a desired interior cell can be cumbersome and difficult since the pointer of the mouse is subject to tolerance of the Rhinoceros viewport. Furthermore, it may not be obvious to the user which node in the form diagram the selected cell corresponds to. By using Rhinoceros' built-in **DrawForeground** functionality, selection of various geometric objects can be enhanced; before the mouse is clicked, the objects where the mouse pointer is hovering on is dynamically highlighted (Figure 5.16). In addition, the corresponding elements in either the form or the force diagram are also highlighted with matching colours. Taking advantage of the simple dynamic drawing functions, the selection process is enhanced, and understanding the relationship between the corresponding elements in the form and force diagrams is also improved without having to constantly redraw the objects.

5.4.4 Display

This part of **compas_3gs.rhino** includes visualisation strategies for improving the user interaction and experience. Various dynamic visualisations and display modes facilitate the filtration and communication of various visual information to the user in a clear and understandable manner.

[1] McNeel's RhinoScript Wiki: https://wiki.mcneel.com/developer/rhinoscript
[2] RhinoCommon API documentation: https://developer.rhino3d.com/api/RhinoCommon

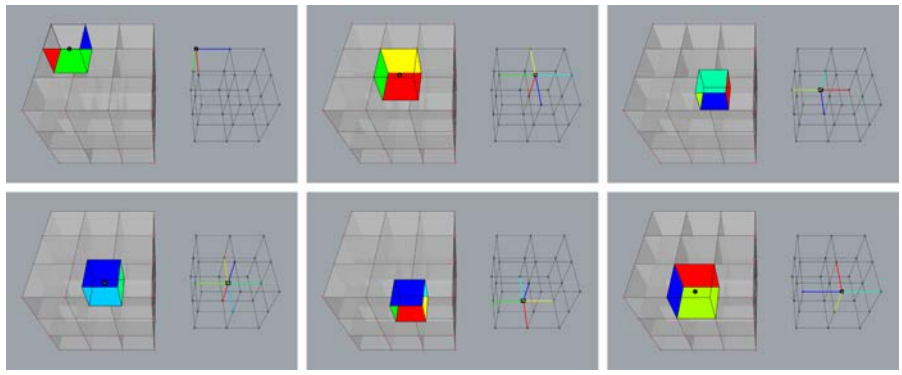


Figure 5.16: Dynamic display of **`volmesh`** elements and the corresponding elements in the **`network`** as the mouse pointer is hovered over them.

Dynamic drawing

`OnDynamicDraw` functions of Rhinoceros can also be used to dynamically visualise meaningful information and data during the operations. For example, during the face pulling operation, simply changing the geometry without any other information makes it difficult for the user to know what the quantitative consequence of the operation is. During this operation, the area of the face can be dynamically displayed as a number (Figure 5.17). In addition, faces could also be colour coded, such that it is possible to visually observe how much the area of the face is changing relative to the other faces. For example in Figure 5.17, a rainbow colour map is used, where the blue end of the spectrum represents a smaller face area compared to the other faces, and the red end of the spectrum represents a relatively larger face area. The face areas can also be dynamically visualised as scaled force vectors at the centre of the polyhedral cell, so that the face areas can be inspected by a number, colour and as a length of a vector.

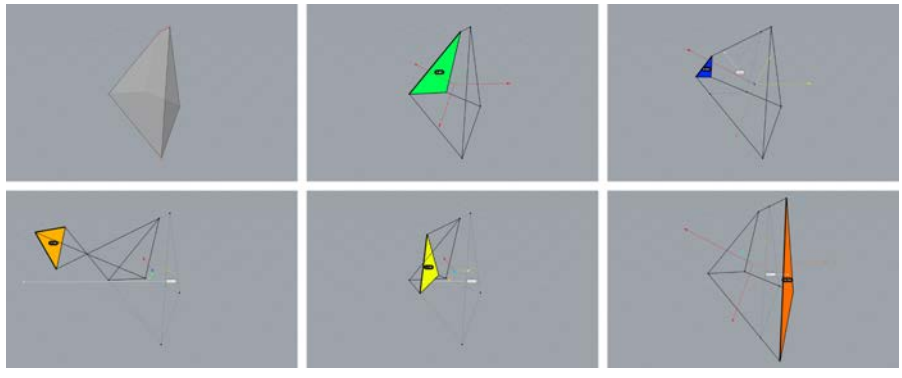


Figure 5.17: Dynamic visualisation of the face area information during the face pull operation.

By using the **`DisplayConduit`** functionality of Rhinoceros, temporary solutions at each step of the iterative solvers can be dynamically visualised. As opposed to having the final results just appear on the screen, visualisation of the individual steps enhances the visual understanding of both the process and the solution. This is drastically different from conventional, "black-box" structural design or analysis tools, which are based on numerical solvers that are difficult for users to understand without an advanced knowledge or experience.

Based on user feedback of similar implementations of computational graphic statics using iterative geometric solvers, the dynamic visualisation of the iterative solving process was appreciated as an invaluable feature in helping them better understand the relationship between the geometry of the form and force diagrams (Rippmann, 2016, p. 254). The dynamic drawing and selection functionalities can be combined to enhance the user interaction and visualisation of the results as well as the solving procedures (Figure 5.18).

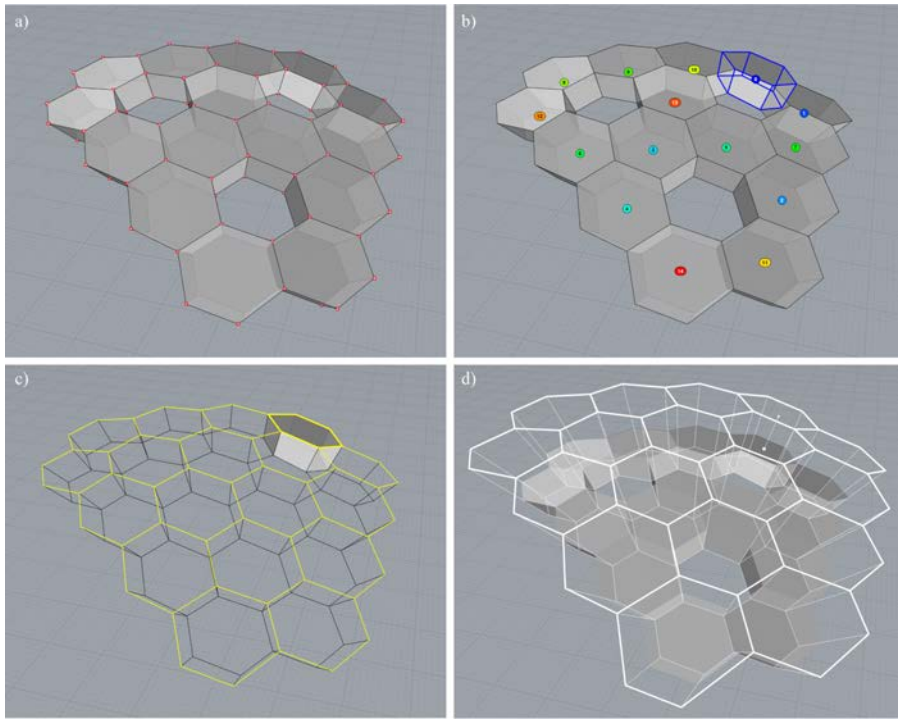


Figure 5.18: Various dynamic drawing helper functions to enhance the face pull operation on a **`volmesh`**: a) a normal display mode of a **`volmesh`**; b) dynamically drawings for visually assisting the selection of cells; c) dynamic highlighting of faces and its dependent faces; and d) **`DrawForeground`** function for previewing the solution.

Static display modes

Different static display modes may be necessary depending on the design problem at hand. By default, the faces of the force diagrams are displayed as a set of meshes, and the corresponding form diagram is displayed using points and lines. If the global force distribution needs to be assessed, the magnitudes of the forces in the structure can be displayed through pipes with different sizes, as scaled force vectors at each of the nodes, or by simply using a consistent colour scale on both form and force diagrams. The unified diagram with a slider for the scale factor α, can be used to visualise force diagrams in a way that is easier to visually understand and discern (Figure 5.19).

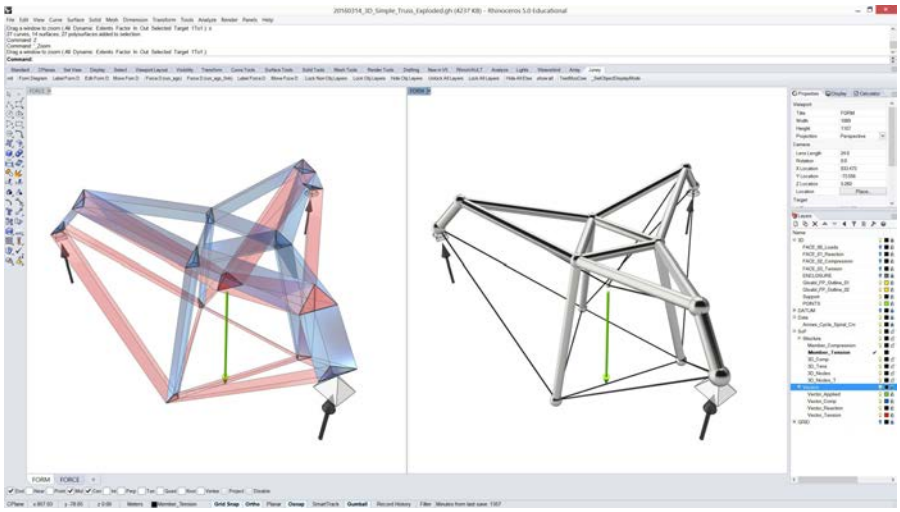


Figure 5.19: A screenshot of the form diagram and the scaled unified diagram in Rhinoceros.

5.5 Summary

This chapter presented **compas_3gs**, an implementation of the computational framework. Overall organisation of the library, implementation of different datastructures in 3D graphic statics context, and technical details of some of the key features and algorithms were described. **compas_3gs.rhino** was also presented, which is a helper package that enables **compas_3gs** and its functionalities to be used and applied within the interactive design environment of Rhinoceros. The online documentation of **compas_3gs** was also introduced, which contains detailed information about the functionalities of the library, the source codes, tutorials, examples and general reference material for users, researchers and potential contributors. Using the presented functionalities of **compas_3gs**, the new structural design potentials of the computational framework will be demonstrated in the next Part of the dissertation.

Part III

Applications

6 Addressing boundary conditions

This chapter presents a method for constructing global force polyhedrons that incorporates various boundary condition constraints of a design problem, using the EGI and the arearisation algorithm (see Sections 4.2.9 and 5.3.2.2, respectively).

Some of the contents of this chapter are based on the following publication by the author:

- *Area-controlled construction of global force polyhedra* (Lee et al., 2017)
- *Disjointed force polyhedra* (Lee et al., 2018)

6.1 Goals

The establishment of the resultant force and the global equilibrium is one of the most fundamental principles of graphic statics. In 2D graphic statics, the resolution of global equilibrium for a given boundary condition (i.e. magnitudes and locations of external loads and reaction forces) are typically addressed using trial funicular construction. The procedures of this technique are well documented in numerous publications (Allen and Zalewski, 2009; Bow, 1873; Wolfe, 1921).

Akbarzadeh et al. (2015a) introduced a three-dimensional equivalent of the trial funicular method for constructing the global force polyhedron, $\Lambda^{\perp}$, for a given set of boundary conditions. By translating well established concepts and procedural techniques from 2D to 3D graphic statics, it was shown how the geometry of a $\Lambda^{\perp}$ can be constructed. However, this method is strictly dependent on geometric procedures, which results in a few key limitations.

First, the method is only applicable for determinate boundary conditions consisting of three support locations and three applied loads. It has not yet been shown how the $\Lambda^{\perp}$ for an indeterminate system can be constructed (i.e. mismatching number of supports and applied loads, asymmetric support and applied load locations), and how the magnitudes of the external forces can be controlled and logically distributed during the construction process.

Furthermore, procedural method for constructing $\Lambda^{\perp}$ does not take into account any specific magnitudes of external forces. Given a set of applied loads with specific magnitudes, the procedural method will need an additional step to ensure that the face areas of $\Lambda^{\perp}$ match the magnitudes of corresponding force.

Finally, procedural geometric construction of $\Lambda^{\perp}$ means that any change in the boundary conditions requires a complete reconstruction of the polyhedral geometry. Although the step-by-step procedure is necessary and important for teaching the fundamental principles of polyhedral reciprocal diagrams, repeated procedural reconstruction is cumbersome and inconvenient during early stages of design when the boundary conditions may not yet be finalised and multiple scenarios can still be investigated.

By using the EGI (see Section 4.2.4) and the arearisation algorithm (see Section 5.3.2), the construction of $\Lambda^{\perp}$ can be automated while maintaining the flexible datastructure to negate the need for constant re-drawing. Building upon the previous implementations that introduced various methods for computing the geometry of polyhedral cells from its face areas and orientations, the examples presented in this section demonstrates the capacity of reciprocal polyhedral diagrams to address more quantitative, force-driven boundary constraints.

6.2 Setup

6.2.1 Basic approach

The method for constructing the geometry of a polyhedral cell from a system of force vectors in equilibrium is provided in Section 4.2.9. The main steps of the method is summarised in Figure 6.1. The method for constructing polyhedral cells using EGI described in Section 4.2.9 assumes that the lines of action all of the force vectors are concurrent, or intersecting at one point in space. For a boundary condition where all of the externals forces are concurrent at point **p** (Figure 6.1-a), the method can be directly used to construct the $\Lambda^{\perp}$.

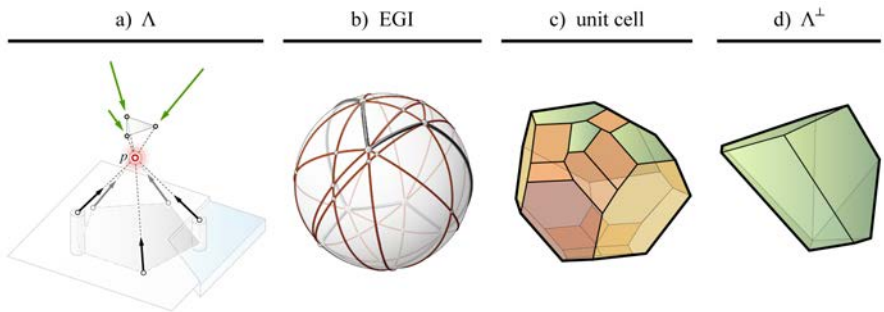


Figure 6.1: Procedure for constructing the global force polyhedron, $\Lambda^{\perp}$, for a concurrent system of external forces: a) given boundary condition, with a common intersection point p of all the external forces; b) the corresponding EGI; c) the unit cell with zero faces; and d) the global force polyhedron, $\Lambda^{\perp}$.

6.2.2 Concurrent cases

In both 2D and 3D graphic statics applications, the global equilibrium of external forces is established using the line of action l of the resultant forces; one resultant force, R, representing the externally applied loads, and one "anti-resultant" force, $-R$, representing the reaction forces. The anti-resultant has the same magnitude as the resultant but opposite in direction. Typically, the magnitudes and directions of the externally applied loads are known, and they can be used to determine the magnitude and location of R by procedurally constructing a funicular polygon or polyhedron (Akbarzadeh et al., 2016), or simply computing the weighted average of the force vectors. The lines of action l of R and $-R$ are coincident in space. For a boundary condition where all of the external forces intersect at a single point **p**, lines of action l of R and $-R$ always passes through point **p** (Figure 6.2-a). For such cases, the forces intersecting at **p** can be used to construct the EGI and then the $\Lambda^{\perp}$ as described in previous sections.

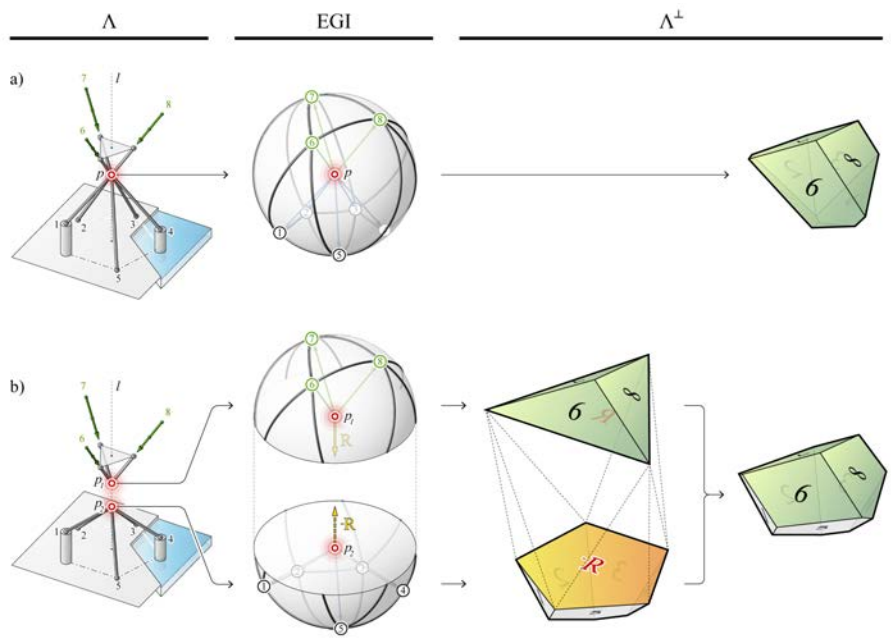


Figure 6.2: a) For a concurrent configuration of forces, a single EGI can describe the $\Lambda^{\perp}$. b) For a partially concurrent configuration of forces, where the lines of action l of R and $-R$ are coincident, the global equilibrium can be represented by two hemispheres of a single EGI. Top hemisphere describes the polyhedral cell representing the equilibrium of the applied loads and the resultant, R, while the bottom hemisphere describes the polyhedral cell representing the equilibrium of the reaction forces and the anti-resultant, $-R$.

6.2.3 Partially concurrent cases

It is also possible for partially concurrent system of external forces to be in global equilibrium. In a partially concurrent configuration, the external forces may not intersect at a single point, but the lines of action l of R and $-R$ may still be coincident. For the example shown in Figure 6.2-b, the externally applied loads intersect at point $\mathbf{p}_1$, while the reaction forces intersect at point $\mathbf{p}_2$. For such partially concurrent cases, the forces at two points $\mathbf{p}_1$ and $\mathbf{p}_2$ can be represented by two halves of a single EGI, where the two points are the origins of the two hemispheres. The top hemisphere then is the dual of the tetrahedral cell representing the equilibrium of the three applied loads and the resultant R. The bottom hemisphere is the dual of the six-sided cell representing the equilibrium of the five reaction forces and the anti-resultant $-R$. Because R and $-R$ are coincident in space with the same magnitude but opposite directions, the two polyhedral cells can be interpreted as a pair of disjointed cells (see Section 4.4). If the two hemispheres were interpreted as a single EGI such that points $\mathbf{p}_1$ and $\mathbf{p}_2$ are coincident, the same procedure used to construct the $\Lambda^{\perp}$ using a single EGI for concurrent cases can be used equivalently for partially concurrent cases.

A parallel configuration of forces can be interpreted as a special case of a partially concurrent configuration of forces, where the lines of action of some or all of the external forces intersect at a common point at infinity. Consider the series of scenarios shown in Figure 6.3. In these examples, the externally applied forces are concurrent and intersect at point $\mathbf{p}_1$ on the line of action l. Reaction forces are also concurrent and intersect at point $\mathbf{p}_2$ on the line of action l. In this series of scenarios, $\mathbf{p}_1$ remains fixed while $\mathbf{p}_2$ is gradually raised until all of the reaction forces are vertical and parallel from one another (Figure 6.3-f).

From the EGI of each scenario, it can be observed that the vertices on the bottom hemisphere converge toward the south pole of the EGI as the reaction forces become more vertical. It is also evident that if the locations of the reaction forces remain unchanged, the topology of the EGI does not change regardless of where $\mathbf{p}_2$ is. Therefore, any of five EGIs can be used to construct the initial unit cell of any other scenario. In general, to construct the EGI representing a partially concurrent configuration, either $\mathbf{p}_1$ or $\mathbf{p}_2$ can be used as the origin of the EGI. For cases where $\mathbf{p}_1$ or $\mathbf{p}_2$ fall far outside of the area of interest such as in a scenario consisting of a parallel configuration of forces, the centroid of the locations of the external forces can be used as the origin of the EGI. Using an arbitrary point like the centroid may result in an EGI that correctly describes the topology of the $\Lambda^{\perp}$, but the vertices may not initially be in correct positions. The target normal for each vertex of the EGI is typically defined by the vector from the origin to the vertex. However, if the forces have specified target magnitudes or orientations, they can be imposed during the final arearisation step of the construction process while maintaining the correct topological structure of the polyhedral cell.

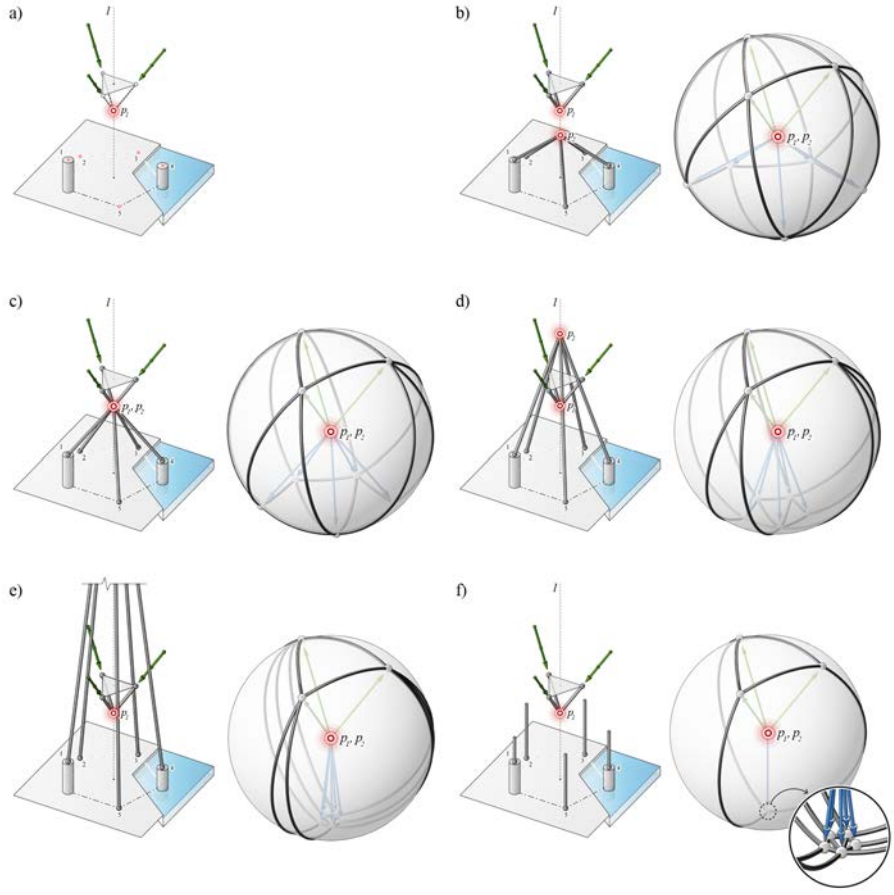


Figure 6.3: For partially concurrent configuration of forces, if the locations of the external forces are fixed, the topology of the EGI is the same regardless of the position of $\mathbf{p}_1$ or $\mathbf{p}_2$ on the line of action: a) base scenario with three applied loads which are concurrent; b), c), d), e) scenarios with varying positions of $\mathbf{p}_2$ and the corresponding EGI; and f) a scenario parallel reaction forces, where the topology of the EGI remains unchanged. Cross adjacency arcs are not shown on the EGIs for visual clarity.

6.2.4 Non-concurrent cases

In a non-concurrent configuration of forces, the forces do not intersect at a common point. In general, a system of non-concurrent forces cannot be deduced to a single resultant force without an introducing and additional "resultant couple" (Akbarzadeh et al., 2015a). Therefore, the polyhedral cell representing the equilibrium of a non-concurrent configuration of forces cannot be closed with a single resultant face, as shown in Figure 6.2-b. As presented, the method of constructing $\Lambda^{\perp}$ using EGI requires that, either: all of the external forces are concurrent through a single point; or, they are partially concurrent with a maximum of two concurrent points on the line

of action of the resultant forces, where all of the externally applied loads are concurrent and all of the reaction forces are concurrent. Non-concurrent configurations of forces are not addressed in this implementation.

6.3 Results

This section presents several design examples demonstrating how the the polyhedral cell construction method using the EGI can be used to construct $\Lambda^{\perp}$ during early stages of design while incorporating various boundary condition constraints.

6.3.1 Simple examples

This subsection presents two types of examples for a simple loading scenario, where various boundary condition constraints result in different geometries of the $\Lambda^{\perp}$. The first set of examples will use fixed support locations, and the second set of examples will use fixed prescribed support force magnitudes.

Consider the loading scenario in Figure 6.4. There are three applied point loads, and five pinned supports. The magnitudes, orientations and the locations of the applied loads (forces 1, 2 and 3) are fixed and are not allowed to change in any of the examples. The supports are always coplanar, and their positions can be either fixed or unconstrained on that plane.

Each applied load will represent one unit force, and the magnitudes of the forces at the supports are assigned and measured proportionally to this unit force. For each example, a table showing which constraints were used as input is shown. Empty slot in the table means no constraint was assigned. In the form diagram for each example, force vectors are shown with dots at 0.5 increments for visual reference.

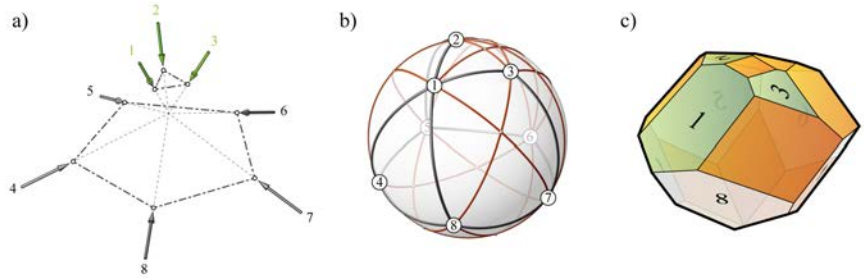


Figure 6.4: a) The loading scenario, with three non-vertical point loads and five supports (the hidden lines are shown for clarity only, and does not necessarily imply any specific structural topology); b) the EGI of the loading scenario; and c) the unit polyhedron with zero-area faces shown in orange, and the applied load faces shown in green.

6.3.1.1 Constraining support locations

In the first set of examples, the locations of the external loads are constrained, and therefore all face orientations are constrained to remain the same. In addition to the orientation constraints, assigning specific target areas for all faces will potentially result in an over-constrained scenario where the forces may not sum to zero and therefore a polyhedron may not exist that satisfies the input constraints. Therefore, only some of the faces are given specific target areas. Because the face orientations are constrained, only the magnitudes of the forces at the supports can change; the support locations do not change in these examples.

In the example in Figure 6.5-a, all faces corresponding to the support reaction forces were given a target area of 1. However, the converged solution does not result in a polyhedron with the correct face areas, suggesting that indeed the inputs caused the problem to be over-constrained, and either the target area or the orientation criteria for some of the faces need to be unconstrained. In Figure 6.5-b and c, fewer faces are given prescribed areas, and the solution converges much more accurately as expected.

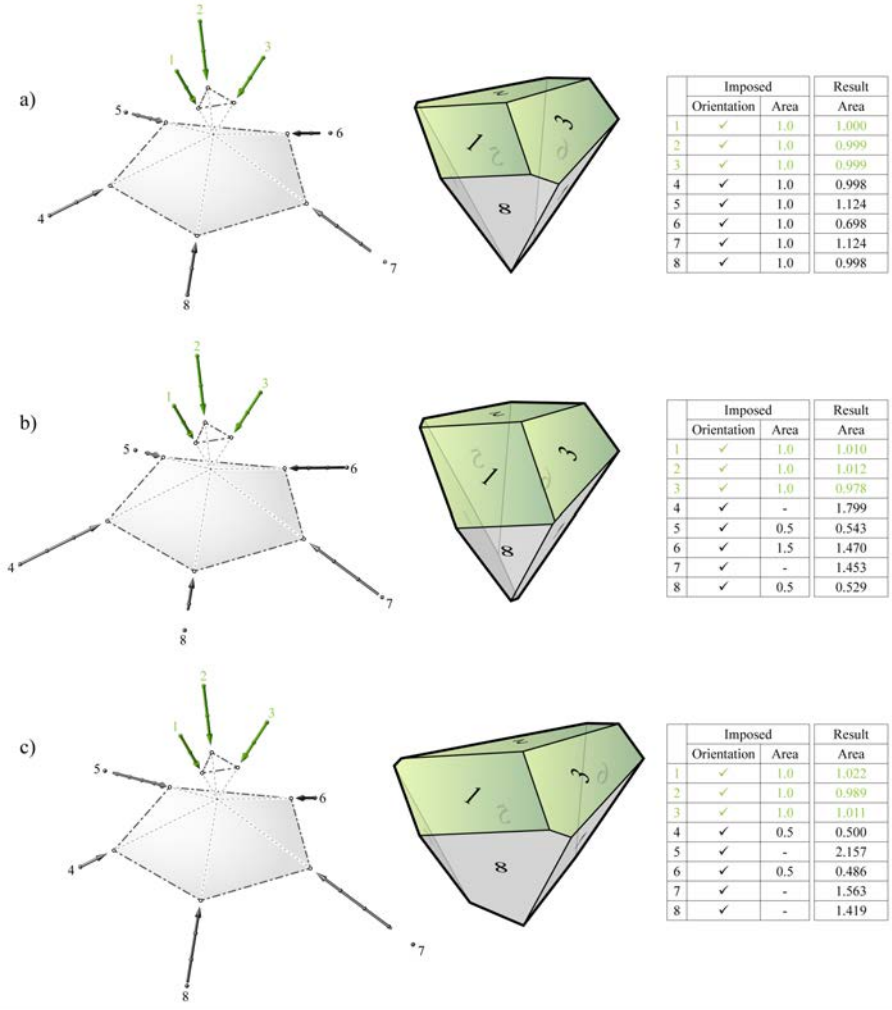
6.3.1.2 Constraining support force magnitudes

In some structural design applications, the magnitude of the reaction force at a support may be more important than its location or orientation. In the second set of examples in Figure 6.6, the orientation of the reaction forces at the supports are allowed to change freely. Unlike the examples in Figure 6.5, all support faces can be assigned target areas, and the correct orientations of the faces will be found that satisfies the target area constraints. In these examples, the resulting polyhedron satisfies the target area constraints, but the supports are relocated to new positions.

6.3.2 Design scenario

The presented construction method can be particularly useful in computing the geometry of $\Lambda^{\perp}$ during early stages of design, when the only information known to the user might be the location and magnitudes of the external forces and basic support conditions and constraints. This section presents a simple design scenario, where various boundary condition constraints result in different geometries of the $\Lambda^{\perp}$.

Suppose that a shell structure with five supports is being considered for a hypothetical site shown in Figure 6.7-a. The client has requested for two of the supports to be on piers 1 and 4. For the first iteration, a solution was found where all of the supports had the same magnitude of reaction forces (Figure 6.7-b). However, it became evident that the two piers may be susceptible to high horizontal thrusts. By limiting the horizontal thrusts



a)

	Imposed		Result
	Orientation	Area	Area
1	✓	1.0	1.000
2	✓	1.0	0.999
3	✓	1.0	0.999
4	✓	1.0	0.998
5	✓	1.0	1.124
6	✓	1.0	0.698
7	✓	1.0	1.124
8	✓	1.0	0.998

b)

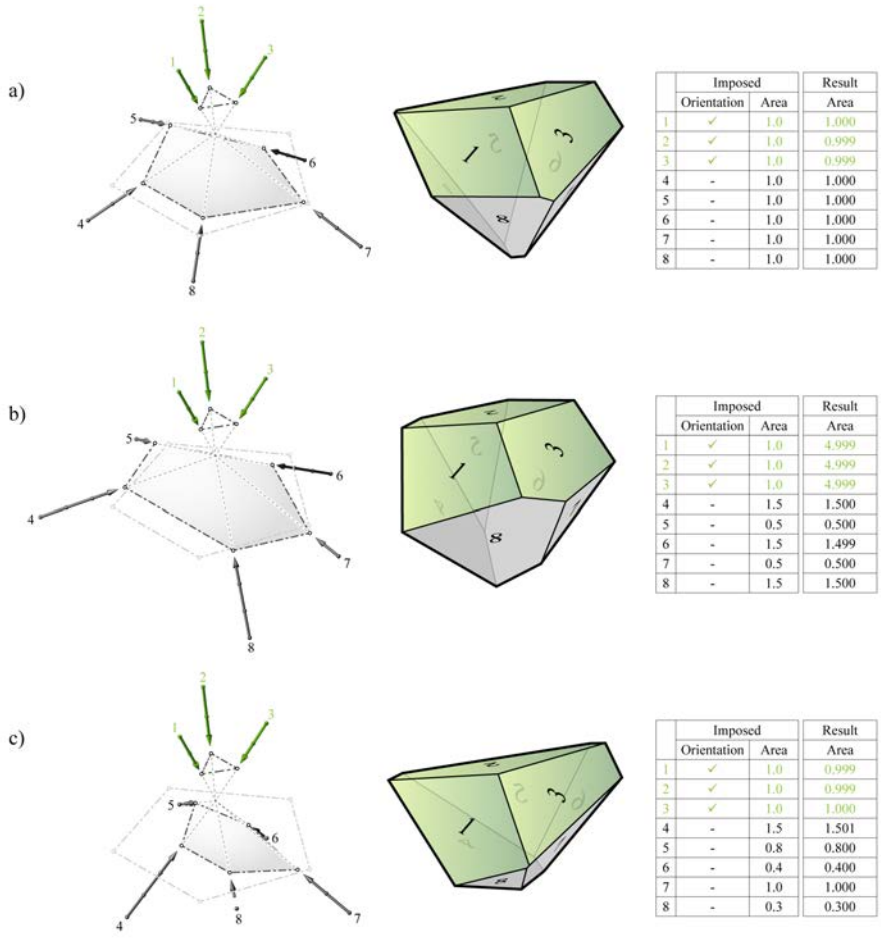
	Imposed		Result
	Orientation	Area	Area
1	✓	1.0	1.010
2	✓	1.0	1.012
3	✓	1.0	0.978
4	✓	-	1.799
5	✓	0.5	0.543
6	✓	1.5	1.470
7	✓	-	1.453
8	✓	0.5	0.529

c)

	Imposed		Result
	Orientation	Area	Area
1	✓	1.0	1.022
2	✓	1.0	0.989
3	✓	1.0	1.011
4	✓	0.5	0.500
5	✓	-	2.157
6	✓	0.5	0.486
7	✓	-	1.563
8	✓	-	1.419

Figure 6.5: Examples where the orientation of the support faces are not allowed to change.

on piers 1 and 4 to not exceed a certain amount, a much shallower $\Lambda^{\perp}$ was found (Figure 6.7-c). As a last constraint, the client requests that none of the supports land on the water front area. By constraining the supports 2, 3, and 5 to stay clear of this zone marked by the red lines shown in Figure 6.7-d, while satisfying the force constraints imposed in Figure 6.7-a and b, a new $\Lambda^{\perp}$ found. Once a $\Lambda^{\perp}$ has been found that meets the main boundary condition criteria, the designer can then proceed to applying various subdivision and transformations to $\Lambda^{\perp}$ to explore more specific forms and topologies of the structure (presented in the next chapter), knowing that the required boundary condition criteria have already been satisfied.



	Imposed		Result
	Orientation	Area	Area
1	✓	1.0	1.000
2	✓	1.0	0.999
3	✓	1.0	0.999
4	-	1.0	1.000
5	-	1.0	1.000
6	-	1.0	1.000
7	-	1.0	1.000
8	-	1.0	1.000

	Imposed		Result
	Orientation	Area	Area
1	✓	1.0	4.999
2	✓	1.0	4.999
3	✓	1.0	4.999
4	-	1.5	1.500
5	-	0.5	0.500
6	-	1.5	1.499
7	-	0.5	0.500
8	-	1.5	1.500

	Imposed		Result
	Orientation	Area	Area
1	✓	1.0	0.999
2	✓	1.0	0.999
3	✓	1.0	1.000
4	-	1.5	1.501
5	-	0.8	0.800
6	-	0.4	0.400
7	-	1.0	1.000
8	-	0.3	0.300

Figure 6.6: Examples where the orientations of the support faces are free to change, and target areas can be assigned for all faces.

6.4 Outlook

This chapter presented how the EGI and the arearisation algorithm can be used to construct a global force polyhedron, $\Lambda^{\perp}$, from a given set of boundary condition constraints. There are a couple of key directions for further research and improvements.

First, it is uncertain how the arearisation algorithm will behave when all of the external forces are parallel, and therefore all of the target normals are parallel. This means that during the iterative geometric computation, the

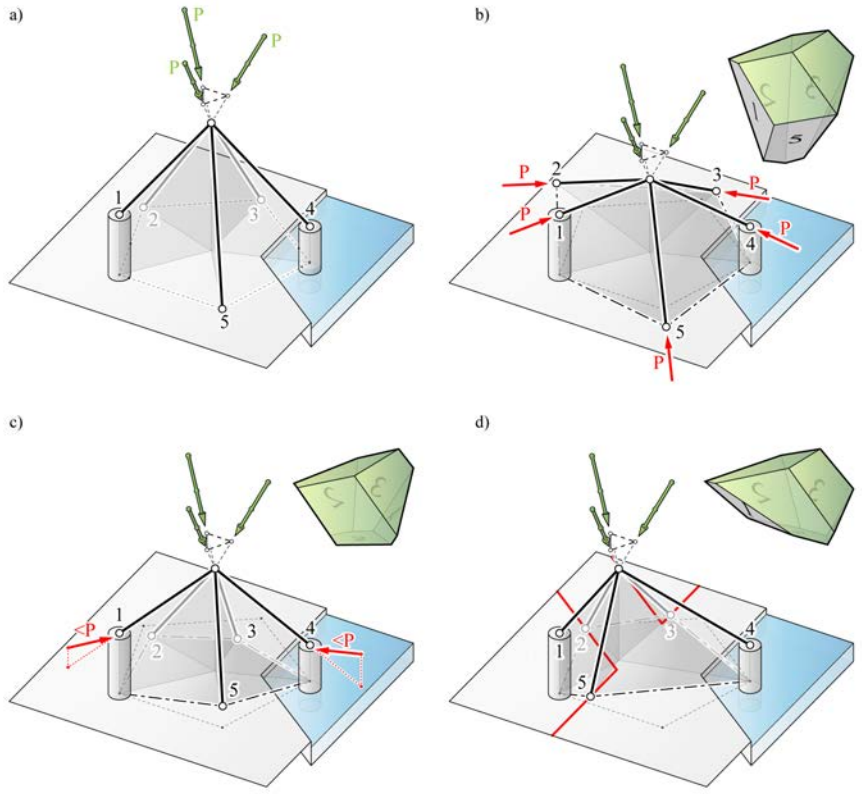


Figure 6.7: a) hypothetical site for a shell-like structure with five supports; b) $\Lambda^{\perp}$ where all the reaction force magnitudes are equal; c) $\Lambda^{\perp}$ where all the reaction force magnitudes are equal, but the horizontal components of the reactions at piers 1 and 4 are limited to stay below a desired amount; and d) imposing constraints from both b) and c), while keeping all of the supports clear of the water front area.

edges between a pair of faces can move freely on the plane since the two corresponding target planes are coplanar and do not have an intersection. As a result, there needs to be an additional process for enforcing the necessary constraints for the edges during the construction of flat polyhedral cells.

In addition, resolving non-concurrent configurations of forces remains challenging. Further steps will need to be investigated to address "resultant couple" forces in order to generalise the presented method to address a wider range of loading conditions.

7 Generating new topologies

This chapter presents form-finding explorations through geometric transformations and modifications of polyhedral force diagrams. First, the setup and workflow of the methodology are presented, describing how various polyhedral transformations and manipulations available through `compas_3gs` can be applied in different combinations for this particular design application. Second, 3D design examples are presented where the spatial topologies are generated solely by manipulating the geometry of force diagrams. Because 2D equilibrium is a special cases of 3D equilibrium, the proposed methodology of this chapter is also applicable for 2D design applications. Finally, the chapter concludes by discussing potential future applications of the presented methodology, particularly in combination with data driven design and machine learning techniques.

Contents of this chapter are based on the following publication by the author:

- *Form-finding explorations through geometric manipulations of force polyhedrons* (Lee et al., 2016)
- *Disjointed force polyhedra* (Lee et al., 2018)

7.1 Goals

In a conventional "top-down" design workflow, a structure is typically sketched by hand, translated into a digital model, then analysed using a finite element software to check its equilibrium. Even if an equilibrium of a spatial structure is achieved, it is difficult to make modifications to the geometry in an interactive manner; a substantial re-modelling and re-analysis are required for any design changes. Subsequently, it is difficult to explore new spatial typologies efficiently, and the resulting spatial structures are often arrays or accumulation of known, two-dimensional solutions.

The goal of this chapter is to propose a new structural design paradigm, where solutions can be generated through transformations of the geometry of force diagrams. Because forms are generated as a result of force diagrams, the entire structure can be easily controlled and modified without breaking its spatial equilibrium. Generating the structural form by manipulating the force diagram also means that the structural typology is an unknown at the start, resulting in feasible designs that are not biased towards known solutions or predefined typologies. Built-in equilibrium constraint of all the solutions means that the user can focus on designing and exploring, and less

on analysing. The design examples in this chapter will demonstrate how the generative nature of this approach can lead to the emergence and discovery of new typologies without any biases towards conventional solutions.

7.2 Setup

First step of the proposed methodology is to formulate the boundary conditions by defining the magnitudes and locations of the applied forces, and the location of the supports. The global force polyhedron Λ can be constructed using the procedure described in Section 4.2.9.2. For an indeterminate system of external forces, the user will need to determine an initial distribution of reaction forces to start the transformative process, which can be modified later. From a set of polyhedral transformations and modifications, either a vertex, face or cell—individually or in groups—can be chosen to apply the geometric operations. These operations can be applied iteratively in any order, until a desired design criteria is reached. After each operation, the reciprocation algorithm can be used to form-find the geometry of the equilibrated structure. The described workflow is summarised in the flowchart in Figure 7.1. Figure 7.2 shows some of the geometric operations that are available in `compas_3gs`, which can be used for this design application.

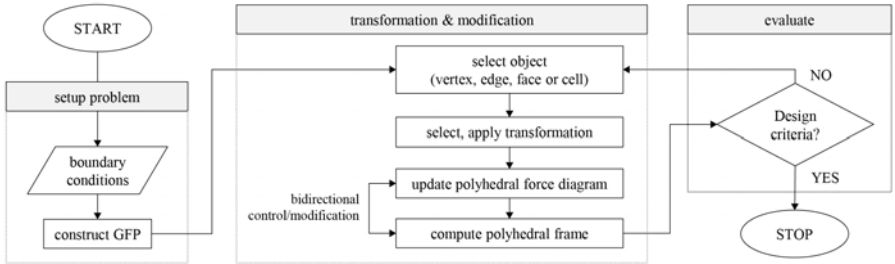


Figure 7.1: Flowchart of the form-finding exploration workflow.

7.3 Results

This section presents variations for three design scenarios generated using the presented method: a horizontally spanning structure with one point load and three vertical (two roller and one pinned) supports (Figure 7.3); a vertical cantilever structure with three pinned supports and a horizontal point load (Figure 7.4); and a bay of a large roof with a single tower support at the centre and a suspended roof (Figure 7.5). The same operations can be applied to prismatic polyhedral cells to explore two-dimensional structures. Figure 7.6 shows how the same operations can be used to generate a two-dimensional, discrete Michell truss.

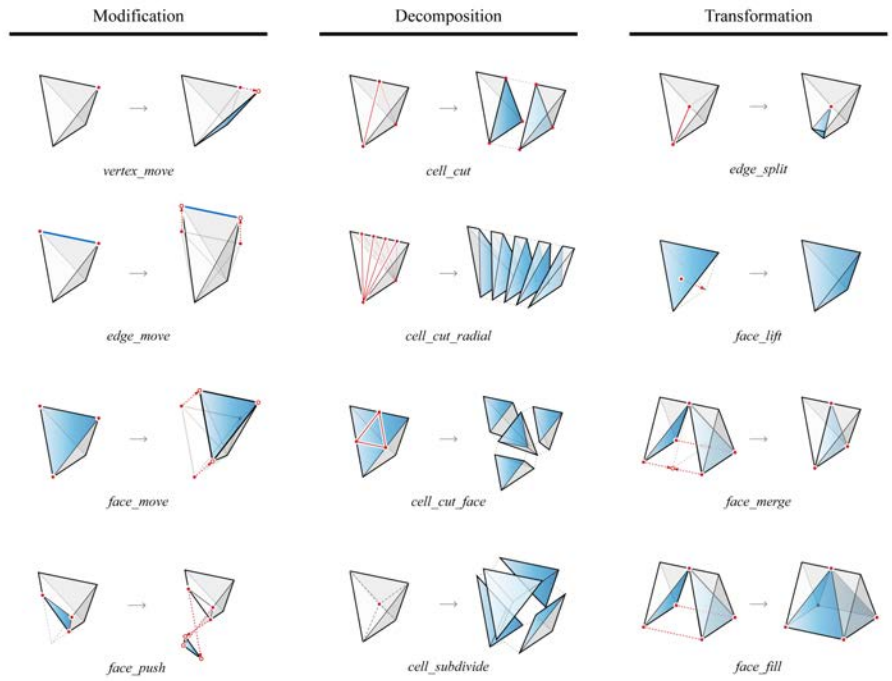


Figure 7.2: Three categories of polyhedral edits: modification, decomposition and transformation.

7.4 Outlook

Shape grammars (Stiny and Gips, 1972; Stiny, 2008) is a rule-based computational design methodology where the generation and transformation of shapes can be automated to explore a wide range of diverse designs. Because of its generative property, new and unexpected design typologies can emerge, allowing designers to explore new design spaces in an unbiased manner. The ability to embed intelligence to the grammar rules allows shape grammars to be applied in architectural design for a variety of purposes: analysis and generation of geometric patterns (Stiny, 2008); qualitative investigation of architectural styles (Stiny and Mitchell, 1978; Koning and Eizenberg, 1981); and function and performance-aware design explorations (Mitchell, 1991).

Shape grammars can also be very powerful in engineering applications as well. Within the context of structural design, shape grammars can be used to generate new structural typologies that are not only feasible but also unique in its visual aesthetics. Furthermore, when combined with optimisation procedures such as shape annealing (Reddy and Cagan, 1995; Shea et al., 1997), the generation of designs can be guided by a performance criteria (Cagan and Mitchell, 1993; Shea and Cagan, 1999). The form-based grammar

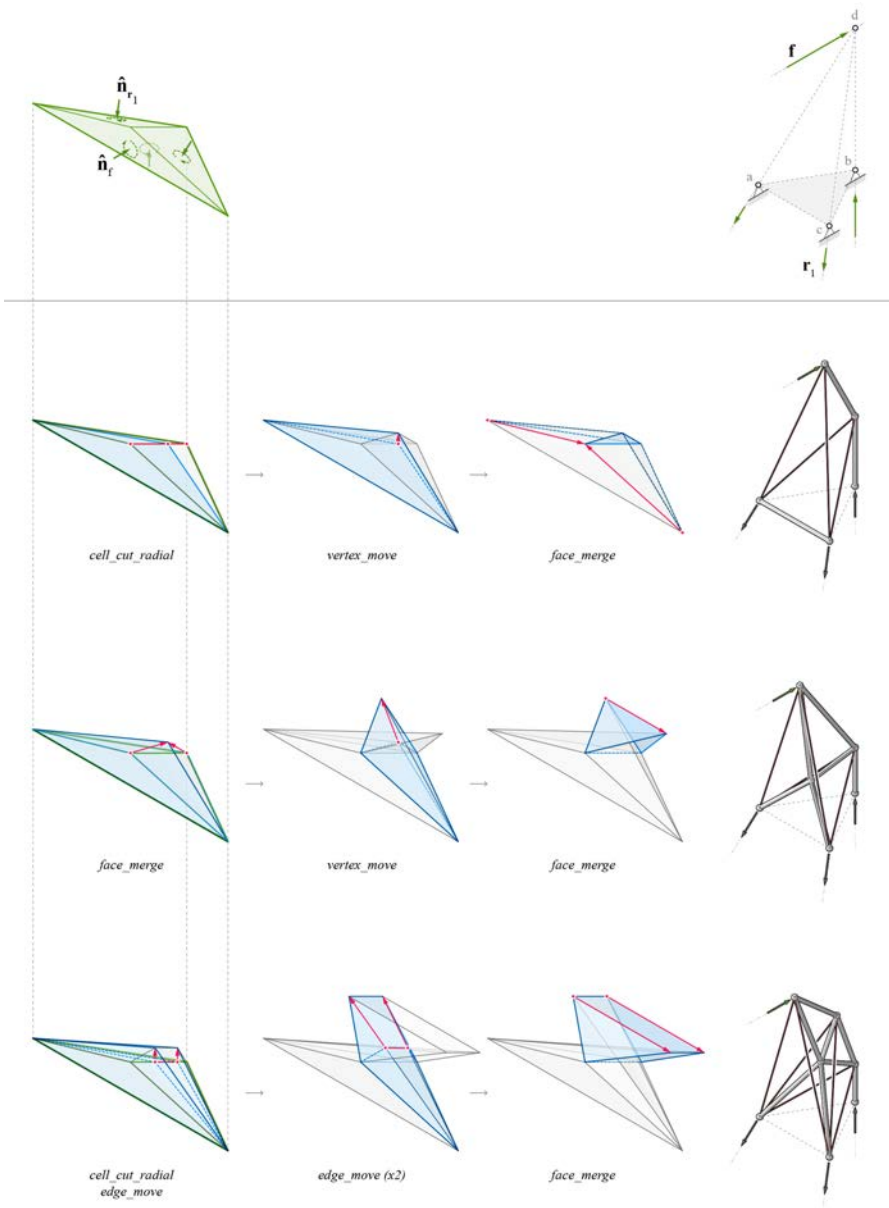


Figure 7.3: Design example 1: a vertical cantilever with horizontal point load.

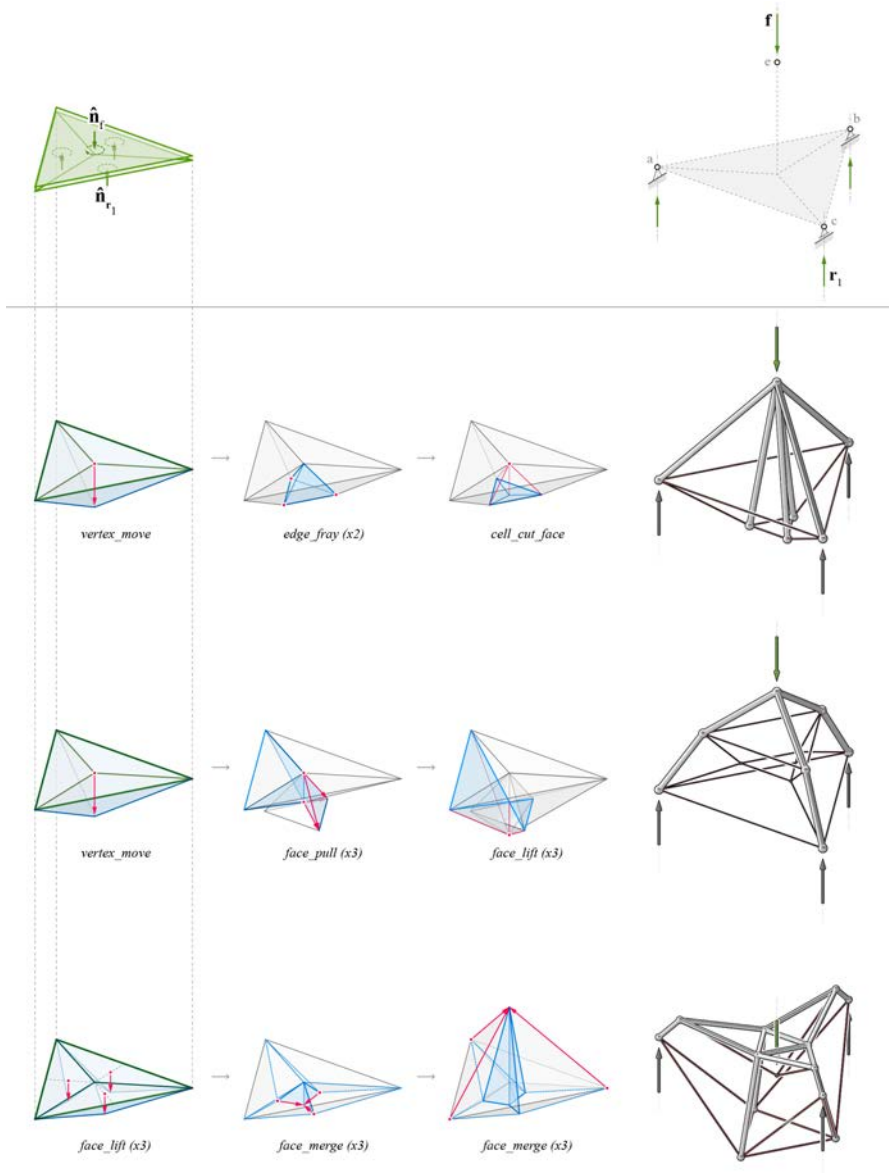


Figure 7.4: Design example 2: a horizontally spanning structure with three supports.

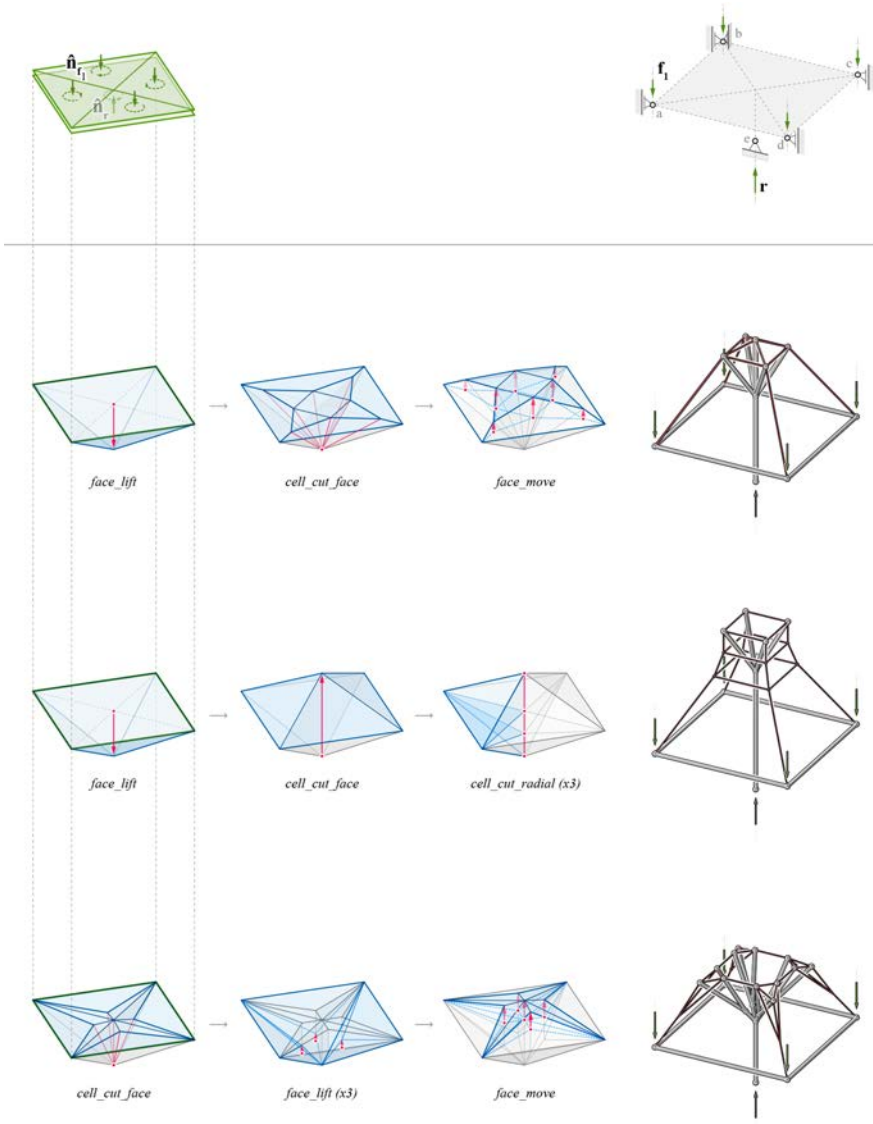


Figure 7.5: Design example 3: a bay of a suspended roof structure with a central tower.

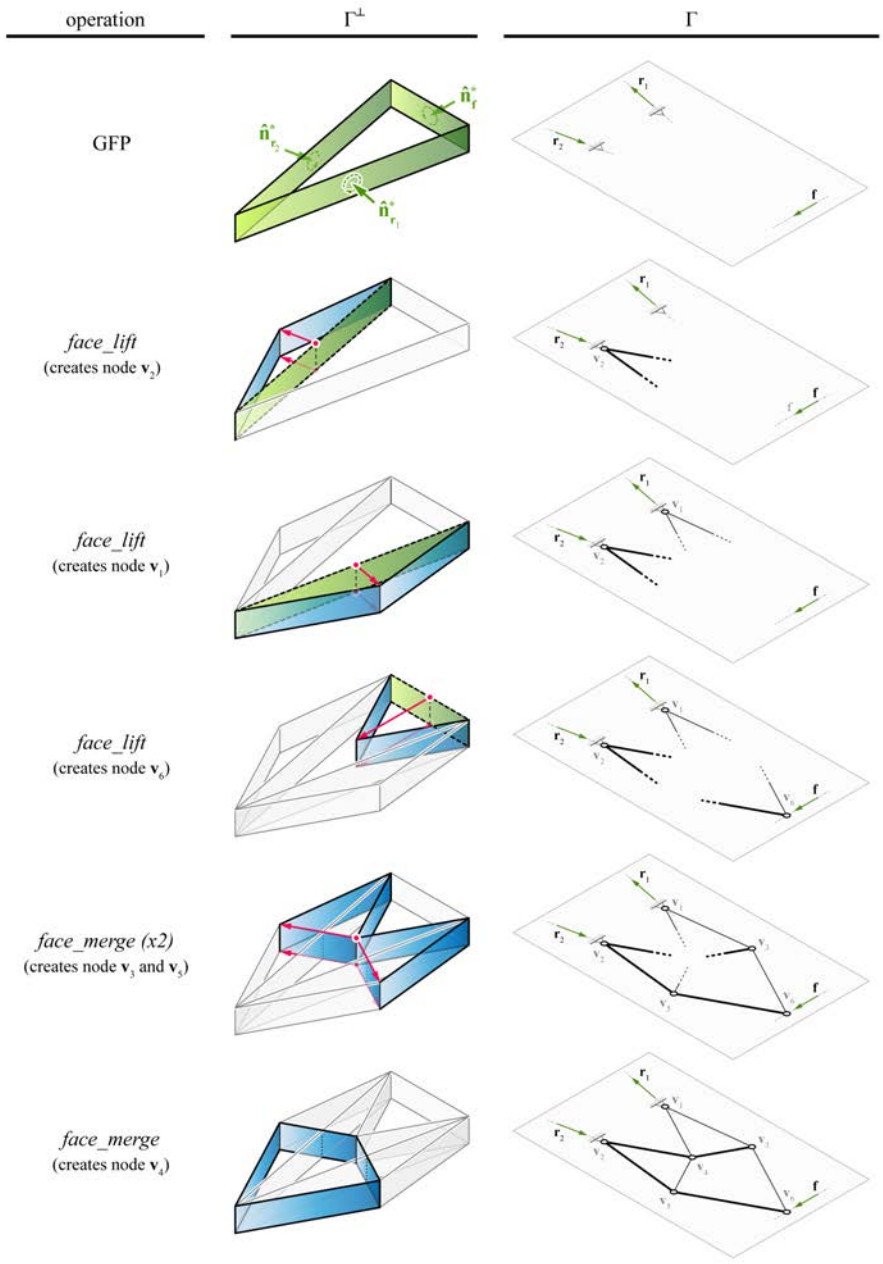


Figure 7.6: Sequence of manipulations to generate a discrete Michell truss.

rules either introduce new members or group existing ones. During each iteration, rules are randomly chosen and applied. The rule is only accepted if the resulting structure performs better than the one from the previous iteration. This evaluation is performed with external finite element software at every step. The procedure continues until the rate of improvement from iteration to iteration has fallen under a desired convergence rate. Using different combinations of rules, different topologies can emerge for the same given design problem (Figure 7.7).

Rather than modifying the geometry of the structure which requires a feasibility and equilibrium check with a finite element software at every iteration, grammar rules can be formulated and applied to the geometry of the force diagram (Figure 7.8). If the external forces are in equilibrium (which can be verified by constructing the global force polygon or polyhedron), a series of subdivisions or transformations of each polygon or polyhedral cell of the force diagram will result in a structure that is already in equilibrium (Akbarzadeh et al., 2014, 2015c; Lee et al., 2016). As a result, equilibrium solutions to a given design problem with an established boundary conditions can be generated rapidly and automatically (Lee et al., 2016).

By the virtue of the existence of the form and force diagrams, the performance criteria (internal axial forces) can be easily calculated through simple multiplications without any external finite element software. Evaluation criteria such as the structure's total load path can be used to sort the results, and only show the user feasible and realistic solutions to a problem. From a selection of designs that are new in its typology and also efficient, the user can first explore the design space visually before choosing an optimal solution from a set of possible designs (Balling, 1999; Mueller, 2014). In this way, new typologies can emerge without the bias of the user, and subsequently lead to new design discoveries.

Although a large number of solutions can be generated automatically, the randomness of rule applications can often lead to inherently bad solutions. And because each solution has a different topology (different number of nodes and members), it is difficult to perform any optimisation procedures during the rule applications to guide the evolution of the solutions. Aforementioned shape annealing could be applied at each iteration, but

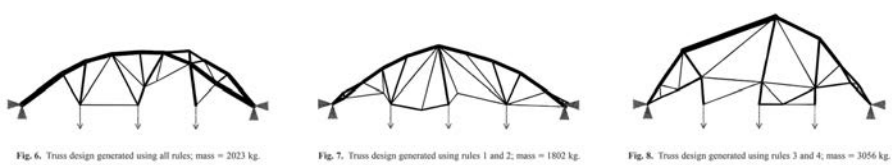


Figure 7.7: Generation of truss designs using shape, size and topology transformation rules in combination with shape annealing (Shea and Cagan, 1999).

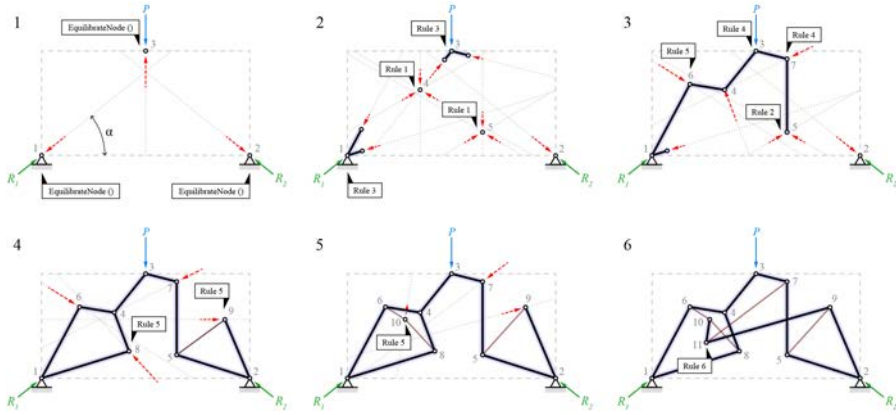


Figure 7.8: Automatic generation of an equilibrium structure for a simple loading case, using graphic statics and shape grammars (Lee et al., 2016).

when exploration and discovery of new typologies is the objective, optimally guided generation of a single solution may not be of priority. Only after tens of thousands of solutions have been generated, can they be sorted using a performance criteria metric for any meaningful, comparative evaluations. Such brute force method can be inefficient and unproductive for more complex design problems. In addition, any small change in the boundary conditions will require an entirely new set of solutions to be generated and sorted. The computational setup also does not learn or store any knowledge from the previous generations or sorting results, meaning it is ultimately a trial-and-error process where any meaningful information gathered from one problem cannot be used for any other problems.

Rather than sorting the raw data of the results for every design problem, the data could be used to extract certain patterns or sequence of grammar rules that tend to result in good solutions. For a human, it is impossible to extract any meaningful patterns, if there are any, from such a large and complex data set. In order to analyse and draw any meaningful conclusions from large data of multi-layered information, machine learning techniques such as sentiment analysis can be used. Sentiment analysis, or also referred to as opinion mining, is a natural language processing (NLP) technique for computational analysis of text, and systematically identifying, extracting and quantifying the underlying subjective information from the text (Liu, 2015). Sentiment analysis is widely applied in internet marketing and customer service sectors to analyse the voice of the customer through the online reviews, surveys and social media data. By analysing the choice and the sequencing of words, and the grammar that holds the text together, the overall subjective tone or attitude of the text can be quantitatively evaluated. The analysis can also happen at different levels of granularity within the text: document level, sentence level or aspect level. With the advent of machine learning and

deep learning techniques, various computational approaches and methods for sentimental analysis have been proposed (Goldberg, 2016; Zhang et al., 2018).

Within the context of structural design with shape grammars, the structures can be considered as the "document" and the grammar rules and their parameters as the "vocabulary" (Shea, 1997). From the large number of solutions that can be rapidly generated, the machine learning algorithms can analyse the sequence of rules and detect any patterns as they relate to the structural performance or efficiency of the eventual solution. As with sentimental analysis of texts, the analysis can also happen at various levels within the structure; it could be global patterns that influence the overall behaviour of the structure, or it could be at the local level of a single node or a neighbourhood of nodes.

With the transformations presented in this chapter formulated as grammar rules, the rule history for each solution can be analysed and evaluated using the similar algorithms used in sentimental analysis applications. In addition, specific combinations and sequencing of grammar rules can be formulated to address particular design problems. A catalogue of good recipe of structural grammar rules could be useful for designers in developing new design aesthetics and styles, but also in gaining new insights as to which combination and ordering of grammar rules tends to improve the performance of structures.

8 Exploring non-polyhedral structures

This chapter presents force-driven design applications through a more precise control of the face areas of force diagrams. It demonstrates how 3D graphic statics can incorporate more quantitative constraints. Using cell networks, the examples presented in this chapter showcase new structural typologies that are not realisable with methods that are based on multi-cell polyhedrons. First, the goals and computational setup of the methodology are described. Then, a wide range of examples are presented, each showcasing a new design opportunity: improvement of visualisation; new structural design typologies; addressing boundary conditions; and interactive force-driven design. Finally, the chapter concludes by discussing potential future applications particularly within the context of structural engineering, and discuss how the presented methodology can offer key advantages over conventional tools.

Contents of this chapter are based on the following publication by the author:

- *Disjointed force polyhedra* (Lee et al., 2018)

8.1 Goals

The goal of this chapter is to demonstrate how controlling the face areas of force diagrams (and therefore the force magnitudes in the spatial structure) expands the range of 3D graphic statics applications. In contrast to the previous chapter where the primary objective was to generate new structural shapes and discover new topologies, this chapter focuses on addressing more realistic, force-driven constraints that are commonly used in structural engineering practice.

8.2 Setup

While geometric transformations of $\Gamma^{\perp}$ are ideal for initial form generation and explorations, $\Psi^{\perp}$ provides a means for designers to interactively incorporate force-driven constraints. The unique features of both $\Gamma^{\perp}$ and $\Psi^{\perp}$ can be incorporated into one coherent workflow.

First, the boundary conditions of a design problem are clearly identified by the designer, from which the initial $\Lambda^{\perp}$ can be generated using the EGI and the arearisation algorithms (Figure 8.1-a). The designer then can proceed to subdividing or transforming the $\Lambda^{\perp}$ (Figure 8.1-c). Using the

reciprocation algorithm, the designer can interactively generate Γ, and explore the geometry of the structure in real-time (Figure 8.1-d). During this interactive exploration, the designer can set various form-driven constraints such as node locations, fixities, edge lengths and orientations (Figure 8.1-b).

Once a general form and topology of Γ have been determined, the designer can proceed to disjointing $\Gamma^{\perp}$ into $\Psi^{\perp}$ using the EGI and arearisation algorithms (Figure 8.1-e), which converts Γ to Ψ. Because the internal force in every member is already known from $\Gamma^{\perp}$, the construction of the initial $\Psi^{\perp}$ is straightforward. The designer can now set various force-driven constraints such as target force magnitudes or orientations for specific members of the structure (Figure 8.1-f). At this point, the designer can still apply any form-driven constraints to Ψ, or polyhedral transformations to the cells of $\Psi^{\perp}$.

Because Ψ and $\Psi^{\perp}$ have an interdependent relationship, the geometry of Ψ needs to be updated at each iteration as each of the cells adapts to the new force constraints (Figure 8.1-h). At the end of each iteration, the two corresponding faces of a member may not necessarily be parallel or have the same areas. For members that do not have target member forces or orientations, the average of the two contact face normals and areas are used as targets for the next iteration.

The iteration is terminated when the desired tolerance or a designated iteration count has been reached. The edges of Ψ and the corresponding cell faces of $\Psi^{\perp}$ should now be close to being perpendicular to one another, unless the form and force constraints input by the designer in the previous steps caused the polyhedral reciprocal diagrams to be over-constrained and a solution satisfying all of the constraints could not be found. In this case, the designer will need to consider eliminating some of the constraints.

As the final step, the designer has the option to visualise the unified diagram, $\Psi^{\perp}(\alpha)$, which aids in understanding the force magnitudes and distributions relative to Ψ (Figure 8.1-i). The designer can also go back to the previous steps to continue the design exploration.

8.3 Results

Using cell networks, new 3D graphic statics applications are now possible. This section presents examples that demonstrate these new potentials.

8.3.1 Improved visualisation for 3D graphic statics

Complicated self-stressed structures, such as the Jessen icosahedral tensegrity (Figure 8.2-a), are used commonly in literature to demonstrate the need for "zero-volume cells" and "zero bars" to construct a complete $\Gamma^{\perp}$ (Figure 8.2-b) (McRobie, 2016b,a; Konstantatou and McRobie, 2016). All of

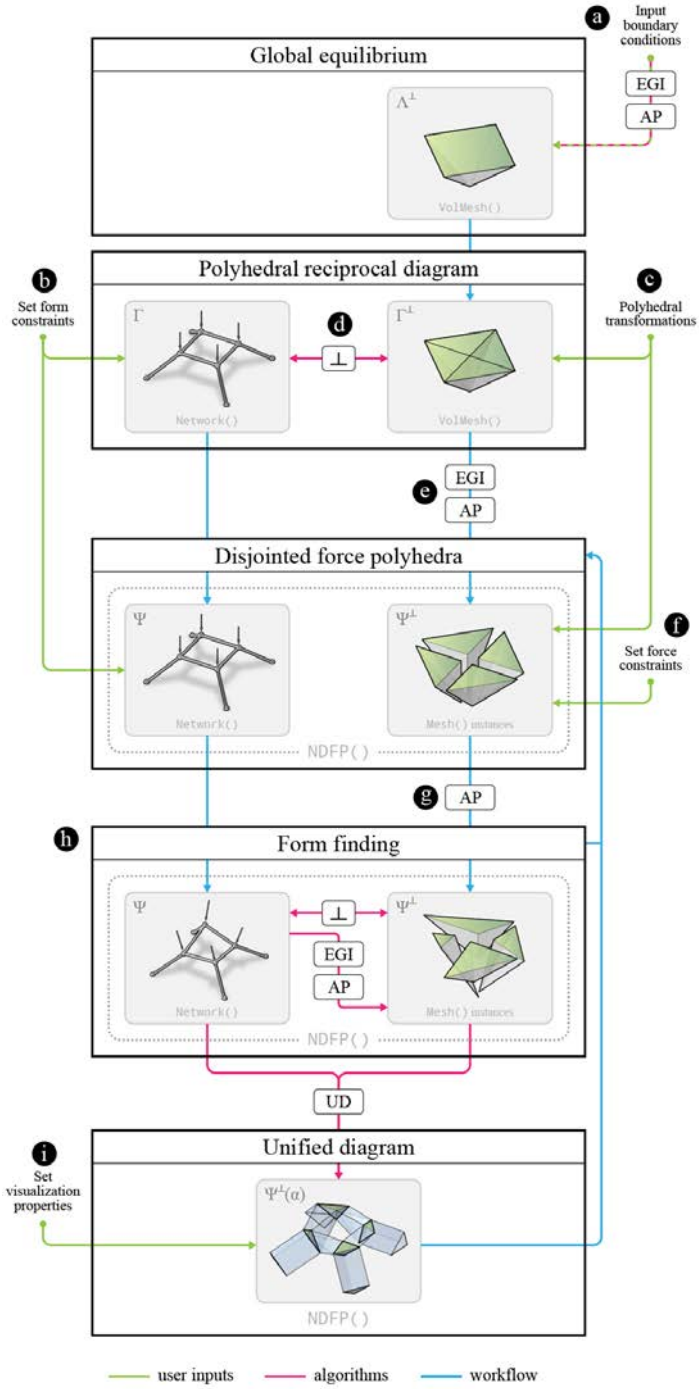


Figure 8.1: Overview of the main steps of the force-driven design workflow using cell networks.

the complex faces of the central, "zero-volume cell" highlighted in Figure 8.2-b have areas of zero. While necessary for constructing a complete $\Gamma^{\perp}$, such overlapping elements only make $\Gamma^{\perp}$ more difficult to read and understand. For a designer who is interested in rapidly exploring the design space using $\Gamma^{\perp}$, clusters of complex polyhedral cells with additional "zero-volume cells" and "zero bars" can be distracting and confusing.

The geometric properties of the Jessen icosahedron is well known (Jessen, 1967; Fuller, 1975; Robert William Burkhardt, 2008), and since tensegrity structures are self-stressed and self-equilibrated, the construction of $\Psi^{\perp}$ is simple and straightforward once the vertex locations have been determined. $\Psi^{\perp}(\alpha)$ shown in Figure 8.2-c is drawn with the same α as the $\Gamma^{\perp}(\alpha)$ in Figure 8.2-b, but without the "zero-volume cells" and "zero bars" and therefore reducing significant amount of visual clutter.

With the equilibrium of the external loads and reactions always being verified by $\Lambda^{\perp}$ and the individual polyhedral cells being generated and visualised per node-by-node basis, $\Psi^{\perp}$ can be constructed for any structure in static equilibrium with minimal number of complex cells and without any additional fictitious nodes or prisms. This example shows that $\Psi^{\perp}(\alpha)$ can be used as a simplified and improved visualisation alternative to a complete but often more complicated $\Gamma^{\perp}(\alpha)$.

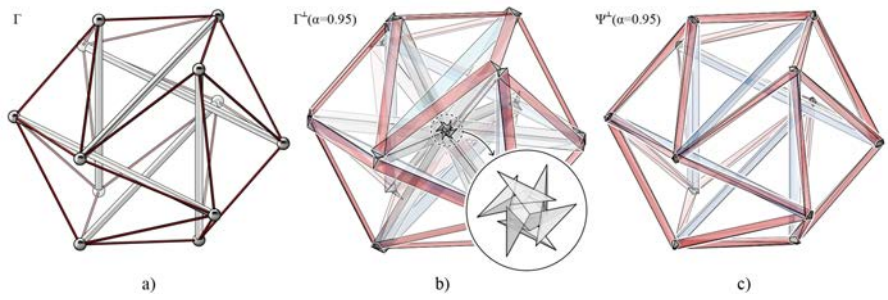


Figure 8.2: a) Γ of a Jessen icosahedral tensegrity structure; b) the complete $\Gamma^{\perp}(\alpha)$ using complex cells and "zero bars" (after McRobie (2016b)); and c) $\Psi^{\perp}(\alpha)$, without the use of any complex cells, "zero-volume cells" or "zero bars."

8.3.2 New structural typologies

In this section, new structural typologies are presented, for which $\Gamma^{\perp}$ cannot be constructed despite the static equilibrium of the structure. It is shown how $\Psi^{\perp}$ can be constructed for such structures and how they can be used in context of 3D graphic statics.

8.3.2.1 2D-3D combined structures

Using prismatic polyhedral cells with virtual faces to represent 2D nodes in equilibrium as described in Section 4.2.7.2, $\Psi^{\perp}$ for structures with both 2D and 3D nodes can now be constructed. Figure 8.3 shows a twisting arch bridge, with 2D nodes along the arch, and point loads applied to the deck. Combination of 2D and 3D nodes in the same structure means that the members that form a face of Ψ are not necessarily planar, and thereby allows incorporation of twisted faces and features into the structure.

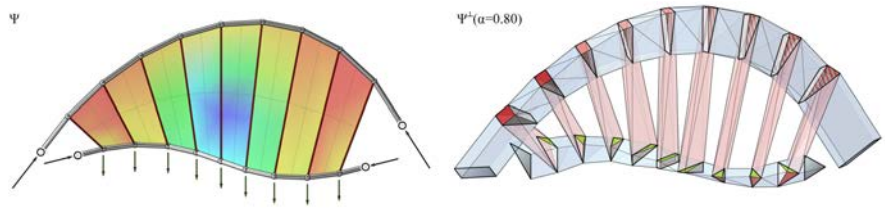


Figure 8.3: Twisting arch bridge with non-planar faces, and the corresponding $\Psi^{\perp}(\alpha)$.

8.3.2.2 Overlapping structures

A complete $\Gamma^{\perp}$ can be constructed only if Γ is a planar graph (Van Mele and Block, 2014). In 2D, this means that a viable form diagram can be rearranged on the plane so that none of the edges are crossing another edge or traversing over a face. In 3D, this means that a viable form diagram Γ can be untangled in space so that none of the edges are crossing a face. Otherwise, a topological dual of that diagram does not exist, and the corresponding $\Gamma^{\perp}$ cannot be constructed.

The example in Figure 8.4 is a layered shell structure with overlapping vertical support elements. Although this structure is in equilibrium, the complete $\Gamma^{\perp}$ cannot be constructed. However, a cell network $\Psi^{\perp}$ can be constructed to represent the equilibrium of this structure, where the individual cells represent the local equilibrium at each of the nodes while being constrained at the periphery by the global force polyhedron representing the external forces.

8.3.2.3 Non-polyhedral structures

The dual and reciprocal relationship between Γ and $\Gamma^{\perp}$ means that both diagrams are polyhedral in their geometric properties. Subsequently, any structure generated through subdivisions or transformations of $\Gamma^{\perp}$ has subspaces that are also polyhedral. The spatial tree structure shown in Figure 8.5 was generated by node-by-node transformations or polyhedral cells, which result in an equilibrated and yet non-polyhedral form diagram, Ψ. $\Psi^{\perp}$ allows exploration of equilibrated structures that do not have polyhedral

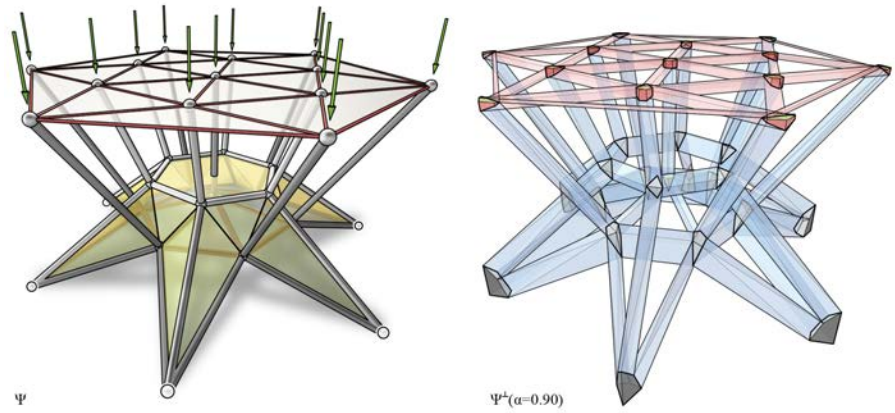


Figure 8.4: A layered and self-overlapping shell structure supporting a flat surface that is uniformly loaded.

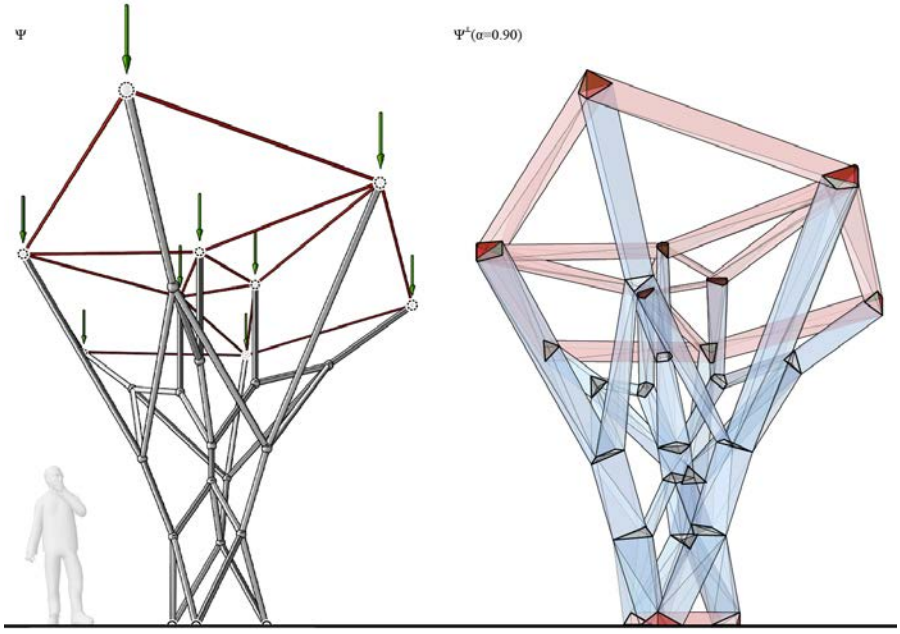


Figure 8.5: Tree structure with non-polyhedral subspaces, as a result of additive, node-by-node transformations of $\Psi^{\perp}$.

geometries, and investigation of free-form designs that are more organic in their aesthetic is now possible. Furthermore, any force equilibrium of structures generated with other form-finding methods such as force density method and TNA can be translated into $\Psi^{\perp}$.

8.3.3 Force-driven design

This section presents how polyhedral cells and cell networks in combination with the arearisation algorithm, can be used to incorporate more force-driven constraints, especially with regards to the boundary conditions.

8.3.3.1 Placing point loads anywhere

One of the main limitations of previous graphic statics applications is that the external loads must be applied at the periphery of the structure, meaning there cannot be any "inner leaves" (Van Mele and Block, 2014). This is also true for 3D graphic statics, where any inner leaves or crossing members mean that a topological dual does not exist. With $\Psi^{\perp}$, point loads can be placed anywhere in the structure.

Figure 8.6-a shows an indeterminate truss with four horizontally restrained supports (1-4) and one pinned support at the bottom (5). If a point load were to be applied to the inner, central node of the structure, a complete $\Gamma^{\perp}$ cannot be constructed as there does not exist a topological dual for such configuration of edges. However, in Figure 8.6-b, a point load of $0.25P$ is applied to the central node at an arbitrary angle, from which a new Ψ and $\Psi^{\perp}$ were found. The new added freedom to place loads anywhere in the structure, allows investigation of irregular loading scenarios, asymmetric loading conditions, and potentially incorporate self-weight loads into the structure.

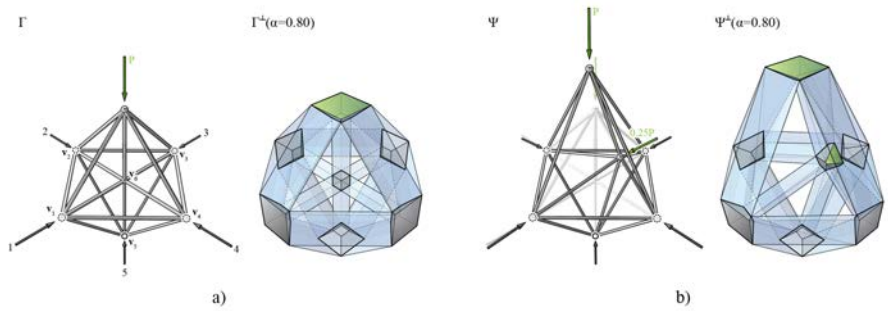


Figure 8.6: a) An indeterminate truss with four horizontally restrained supports (1-4) and one pinned support (5), and the corresponding $\Gamma^{\perp}(\alpha)$; and b) the same truss with a point load applied to the interior node, and the corresponding Ψ and $\Psi^{\perp}(\alpha)$.

8.3.3.2 Tributary area

While polyhedral subdivisions and transformations allow generation of intricate spatial structures, the resulting distribution of applied loads on the structure does not end up representing realistic loading scenarios.

Consider the spatial tree structure shown in Γ of Figure 8.7-a, which is designed to support a triangle-shaped roof that weighs P. Suppose that the points on top and the bases of the structures are finalised by the design team, and are fixed for the remainder of the design exploration. As shown in Γ of Figure 8.7-a, the distribution of applied forces as a result of polyhedral transformations of $\Gamma^{\perp}$ often do not correctly reflect the true tributary areas of the structure. With the top and the base points fixed, and the correct distribution of applied loads imposed, the new shape of the design is found (Figure 8.7-b).

A designer is typically concerned with rapid shape explorations during early stages of design. As the design is gradually finalised and a specific topology of the structure is chosen, the designer can begin to adjust the design load case of the initial $\Gamma^{\perp}$ to a more correctly calibrated loading scenario to continue developing the design towards a more realistic version of the initial concept.

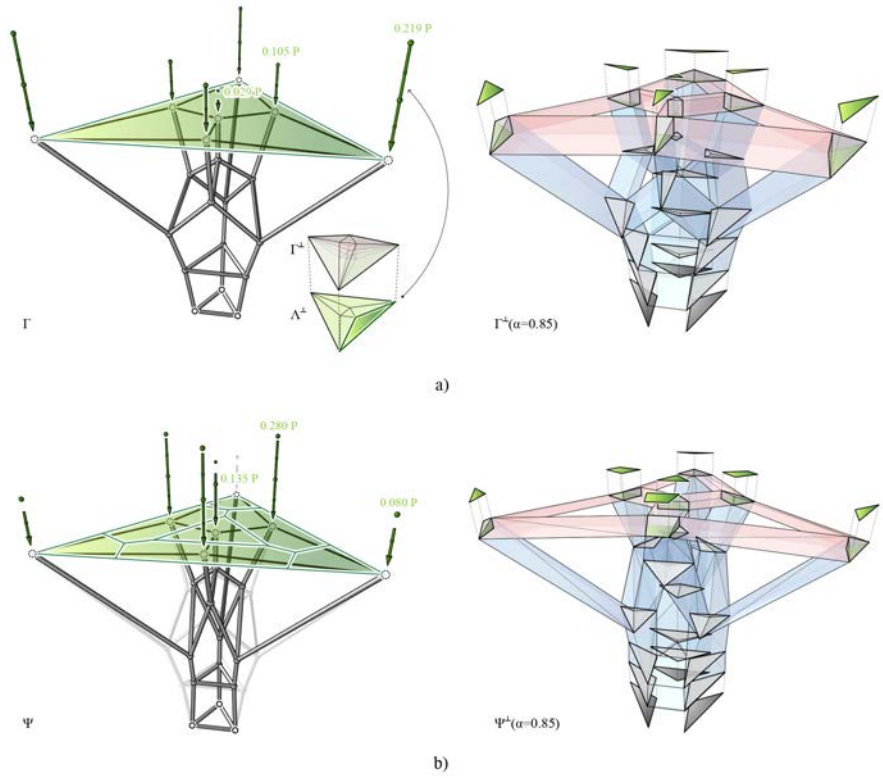


Figure 8.7: a) A tree structure generated through polyhedral subdivisions and transformations of $\Gamma^{\perp}$, with a distribution of applied loads that do not reflect the correct tributary areas; and b) the same structure with correct distribution of applied loads according to the actual tributary areas, and the subsequently form-found, new tree structure.

8.3.3.3 Interactive force-driven design

Cell networks can also be used in an interactive modelling environment and allow designers to explore various spatial structures based on specific force-driven constraints in real-time. Consider a vertically loaded column in Figure 8.8-a, with a fully pinned support at the bottom and a horizontally restrained support at the top. Any combination of edges in Ψ or cell faces of $\Psi^{\perp}$ can be selected to input specific target force magnitudes.

In general, Ψ for a given force distribution is not always unique, and is subject to certain geometric constraints such as maximum and minimum length of edges allowed, and node location constraints during the form finding process. Consequently, a design problem can become over-constrained and an equilibrium solution may not be found that satisfies all of the input constraints. In such cases, the converged solution then provides the designer with the closest solution given the input constraints, and indicates where certain constraints can be removed or modified. The presented framework allows designers to interactively set different combinations of constraints to explore various equilibrium solutions.

8.4 Outlook

With the ability to control the face areas, and therefore the force magnitudes within the structure, polyhedral reciprocal diagrams can potentially be utilised for numerous other engineering applications. One particular area of relevance is strut-and-tie modelling (STM).

STM is a method of shear analysis of deep concrete structures in bending by constructing an equivalent truss. In practice, it is primarily used for designing discontinuity regions (often referred to as D-regions in literature) in reinforced and prestressed concrete structures to determine the layout scheme for reinforcements. D-regions are parts within the concrete block where the geometrical discontinuities cause strain distributions to be non-linear.

By constructing a triangulated truss structure within the boundaries of the concrete mass or block, one possible load path is assumed, from which the designer can begin to understand the complicated internal force flow as a simplified series of compression struts and tension ties. By calculating the forces in the tension ties, the reinforcements can be sized and laid out along the tensile members. STM is based on the concept of lower-bound, theory of plasticity; the capacity of the equivalent, internal truss that is in equilibrium with the applied loads, will be greater than or equal to the collapse capacity of the concrete mass. It is a well-established method in structural engineering of concrete structures. STM standards and guidelines can be found in most major building codes.

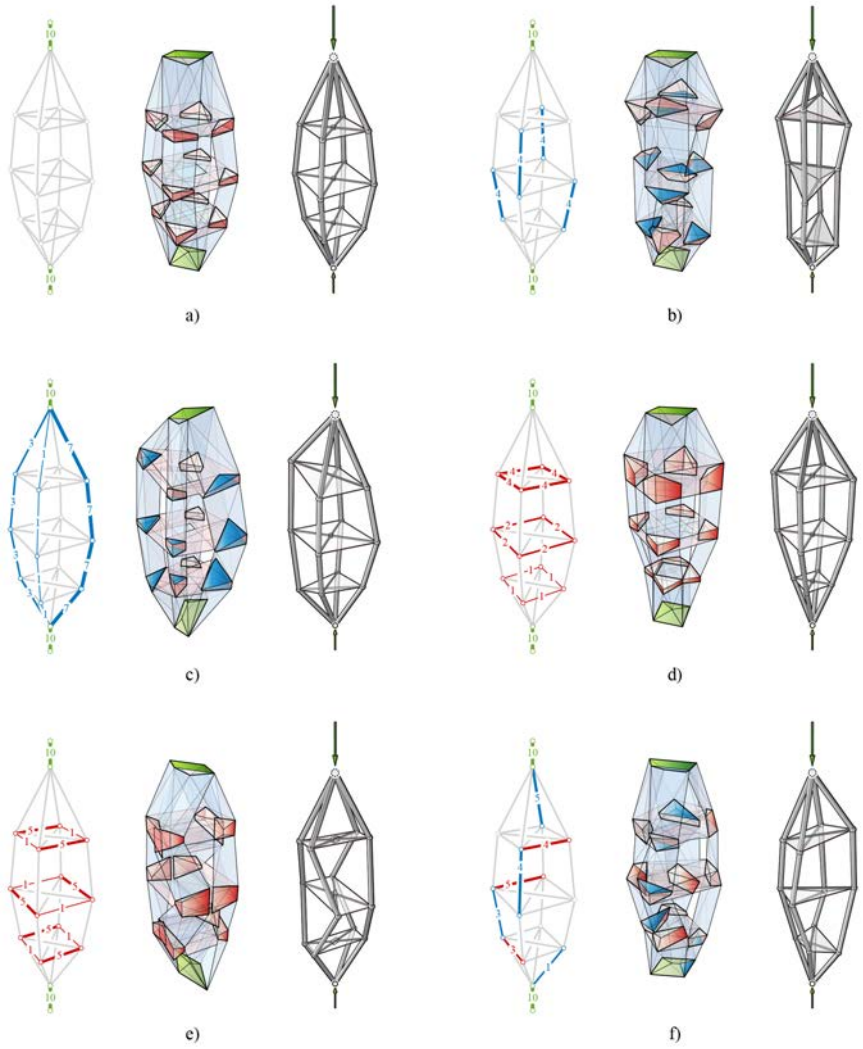


Figure 8.8: Interactive design explorations of a vertically loaded column using various combinations of target force magnitudes in specific members of the structure. Each example shows: the initial structure with the target force magnitudes; the corresponding $\Psi^{\perp}(\alpha)$; and the form-found Ψ.

Although the building codes provide detailed rules and guidelines on how rebars should be laid out, the method of generating the initial truss model is not clearly established in practice. It is generally dependent on the experience and intuition of the engineer. From built projects and known solutions, an engineer can develop his or her own a set of rule-of-thumb techniques which can be used to make an educated guess on the

internal force flow of the concrete blocks (Schlaich and Schäfer, 1991). The engineer then typically sketches out the equivalent truss model by hand. Alternatively, the internal force flow can be approximated through topology optimisation or stress field analysis. Although a more accurate internal force flow can be approximated, the conversion of pixel or vector-field data to a discretised truss model still requires human interpretation (Beghini et al., 2013) or an entirely separate computational procedure. Even after a logical interpretation has been made, the static equilibrium of the truss model is not always guaranteed. Most importantly, if some tensile members of the strut-and-tie model is over-stressed and an alternative truss model needs to be investigated, the entire process must be restarted.

Graphic statics can be particularly useful in constructing suitable strut-and-tie models for generating reinforcement layouts, as demonstrated by Alic and Persson (2018) for 2D cases. For 3D STM problems where the connectivity interpretation can be much more challenging, the algorithms and various functionalities of **`compas_3gs`** can facilitate the engineers in

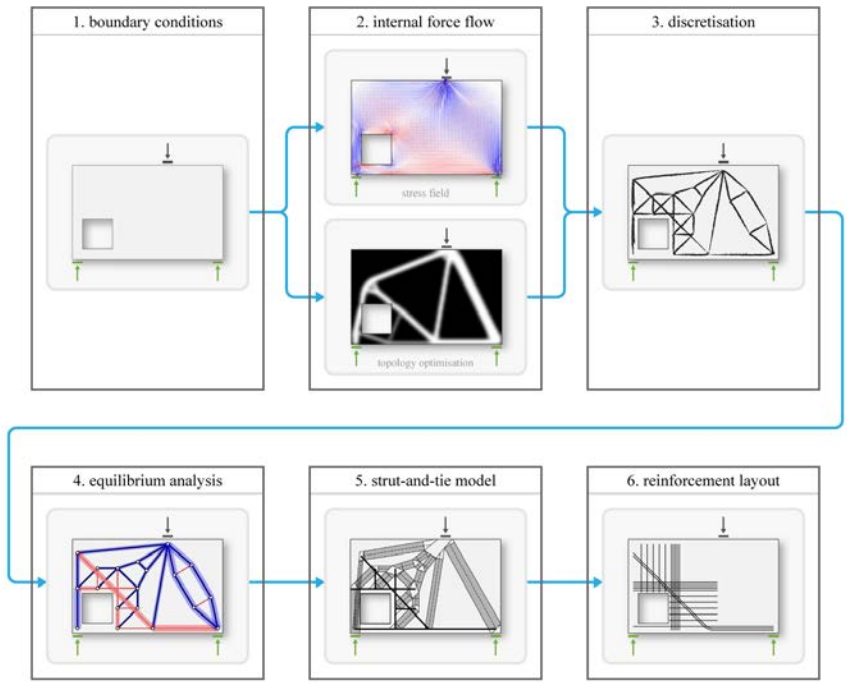


Figure 8.9: Conventional workflow for 2D strut-and-tie modelling: 1) boundary conditions including the extents of the concrete mass and the location and magnitudes of the externally applied loads; 2) generation of the internal force flow patterns using stress field analysis with a FE software or topology optimisation; 3) interpretation of the internal force flow patterns as a discretised truss model; 4) equilibrium analysis of the truss model; 5) calculation of the forces in the tension members; and 6) sizing and lay out of the reinforcement.

ways conventional FE methods cannot. In 3D, the setup of the problem is generally the same as the 2D case. The formulation of the boundary conditions consists of volumes and volumetric voids as opposed to 2D polygons and voids (Figure 8.10-1). If topology optimisation is used to interpret the force flow, the result is made up of voxels as opposed to the pixel information for 2D cases (Figure 8.10-2). This voxel information can be used to generate the initial truss model.

Rather than manually guessing the node and member locations, a skeletonisation algorithm can be developed in the future that automates this process (Figure 8.10-a). Discretisation of pixel or voxel clusters into lines or curves is a well-researched topic in medical imaging (i.e. tractogrphy of X-ray image information for tracking brain's white matter bundles for neurosurgical planning) and computer graphics (i.e. handwriting recognition). Within the context of STM, this process is in principle the same as discretising pixel or voxel information as a result of topology optimisation. Various strategies have been proposed regarding this procedure in the literature, with more recent application to 3D problems by Cuillière et al. (2018).

The result of skeletonisation can be considered as the initial form diagram of the truss model, which is not guaranteed to be in equilibrium or necessarily polyhedral in its geometry (Figure 8.10-3). From this initial form diagram, the cell network $\Psi^{\perp}$ can be generated using the EGI for each node (Figure 8.10-b). The initial $\Psi^{\perp}$ can then be modified with the appropriate boundary condition constraints (i.e. applied and reaction force locations and magnitudes) (Figure 8.10-c). Reciprocation of the modified $\Psi^{\perp}$ then yields an updated form diagram Ψ which is now in equilibrium.

The modified Ψ which is now in equilibrium, can be used to analyse the forces within the tensile members. Another benefit of using 3D graphic statics for STM applications is that the forces can be constrained within the truss model. If some of the tensile members are over-stressed, the form diagram can be easily modified or transformed to reduce the stresses in specific areas of the structure by enforcing maximum force magnitude constraints, or by altering the geometry of the structure to diffuse the stress concentrations through subdivision or transformations of the force diagram. In this way, cell networks can be used to equilibrate discretised truss models in space while enforcing necessary boundary condition and other force-driven constraints.

In addition to having direct relevance and applicability for STM and reinforcement design for concrete structures, the equilibrated spatial structures generated through this workflow can also be used as the structure itself. The concrete block can be considered as the design domain, where the topology optimisation and skeletonisation generate the initial topology of the form diagram, with which the designer can use as a good starting place for further design explorations.

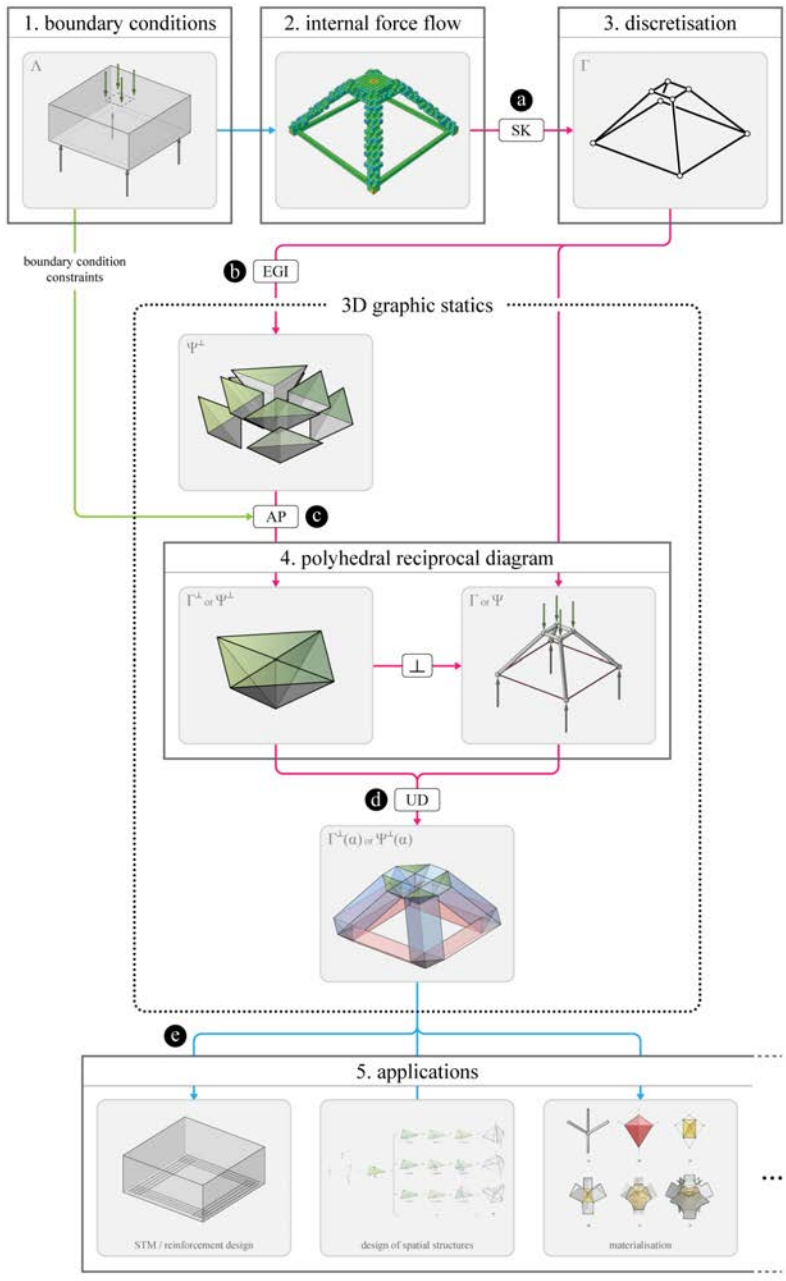


Figure 8.10: Potential workflow for 3D strut-and-tie modelling: 1) boundary conditions in space; 2) voxel data from 3D topology optimisation (image source: (Shobeiri and Ahmadi-Nedushan, 2017)); 3) initial form diagram; 4) polyhedral reciprocal diagram; and 5) further applications.

9 Materialising 3D graphic statics

This chapter presents fabrication-related applications of 3D graphic statics. These applications are demonstrated through the form finding and fabrication design of the MycoTree project for the 2017 Seoul Biennale of Architecture and Urbanism. As the main designer and developer of the structural geometry and the fabrication process, the author was able to test, verify and improve various functionality and features of the presented computational framework. First, the motivations behind the project and the use of cultivated building materials are discussed. Second, the design and development of the structural geometry is presented. Additionally, detailed description of the fabrication geometry of the moulds and various components of the project will be provided. This chapter concludes by making conjectures on other potential applications.

MycoTree was exhibited in Seoul, Korea from September, 2017 until April, 2018 as the centrepiece of the "Beyond Mining – Urban Growth" exhibition at the 2017 Seoul Biennale of Architecture and Urbanism curated by Hyungmin Pai and Alejandro Zaera-Polo. MycoTree is a result of collaboration between the following partners:

- **Sustainable Construction, Karlsruhe Institute of Technology** (material research)

  Karsten Schlesier, Felix Heisel, Dirk Hebel

- **Block Research Group, ETH Zürich** (structural geometry, form finding and fabrication design)

  Juney Lee, Matthias Rippmann, Tomás Méndez Echenagucia, Andrew Liew, Noelle Paulson, Tom Van Mele, Philippe Block

- **Alternative Construction Materials, Future Cities Laboratory, Singapore ETH Centre** (material testing and development)

  Nazanin Saeidi, Alireza Javadian, Adi Reza Nugroho, Robbi Zidna Ilman, Erlambang Adjidarma, Ronaldiaz Hartantyo, Hokie Christian, Orion Tan, Sheng Yu, Kelly Cooper, Dirk Hebel

- **Mycotech, PT Miko Bahtera Nusantara, Indonesia** (production partner)

Contents of this chapter have been published in (Hebel et al., 2017; Heisel et al., 2017, 2018)

9.1 Goals

The goals of this chapter is to demonstrate how the inherent, polyhedral geometric properties of polyhedral force diagrams can be exploited to enhance the fabrication design and process. The three specific goals highlighted in this chapter are as follows.

The first goal is to enable explorations of branching, funicular structures in space beyond arches and domes. Such compression-only structures drastically reduce the internal forces, and therefore the stresses on the materials. As a result, weak materials can be used in full-scale, structural applications.

The second goal is to bring to light the importance of calibrating polyhedral reciprocal diagrams to realistic boundary conditions. As discussed in Section 8.3.3.2, a structure that is form-found through polyhedral reciprocal diagrams is only in equilibrium with loads of specific magnitudes that are applied at specific locations. These load magnitudes and locations typically do not coincide with the actual tributary area of the structure. Such boundary condition criteria must be taken into account to ensure that the computationally form-found geometry will sufficiently stand in reality with realistic loads.

The third and last goal is to take advantage of the planar geometry of the polyhedral form and force diagrams and directly use it to facilitate the fabrication and production process. In contrast to 3D printing and milling operations which can be time consuming and expensive with increasing scale of projects, polyhedral geometric processing allows realisation of complex spatial structures using modest sheet materials and relatively inexpensive 2-axis CNC and laser cutting.

Ultimately, the practical application of the presented computational framework for 3D graphic statics provides insights into the challenge of translating academic, pavilion-scale projects to real building applications. Scaling up of digital fabrication is not entirely dependent on high-tech machines, but can also be facilitated by high-tech geometry that allows realisation of complex geometry through comparatively cheap materials and processes.

9.2 MycoTree

This section describes the motivation, design of structural geometry, development of fabrication process and on-site assembly of the MycoTree project.

9.2.1 Motivation

As populations and urban aspirations grow, so does the demand for materials and resources to support them. Although such demands were once

satisfied by local and regional hinterlands, they are becoming increasingly global in scale and reach. This phenomenon has generated material flows that are trans-continental in scope and has profound consequences for the sustainability, functioning, sense of ownership and identity of future cities. However, the focus of the global construction industry on only a few established building materials results in a high consumption of specific and limited natural resources (Frondel et al., 2007). In this respect, the built environment of the future city demands the utilisation of new resources.

The 21st century will face a radical paradigm shift in how materials are produced for the construction of future habitats. The linear concept of "produce, use, and discard" has proven itself unsustainable in the face of scarce resources and exponentially growing urban populations (Hebel et al., 2014). Instead, to achieve a cycle of production, use, and re-use, we must explore alternative materials and approaches to construction. Materials that were previously considered unwanted and low-strength may present possibilities to end this undesirable state of affairs.

MycoTree, the prototypological structure described in this chapter, consists of organic building components made from agricultural waste, which are stabilised only by the grown matrix of mushroom mycelium. Its geometry was designed using 3D graphic statics, utilising compression-only form to enable the weak material to perform structurally. The resulting expressive and provocative structure demonstrates that the combination of newly developed building materials—which can be effectively grown locally or even on-site—with informed structural design and engineering may bring about the changes that are desperately needed.

9.2.1.1 Depleting finite resources

Over the last few centuries, the global construction industry has grown increasingly more dependent on mined resources. Today, steel-reinforced concrete is the most produced construction material on an industrial scale worldwide, with more than 50% of all man-made objects containing cement-bearing materials (Scrivener, 2015). Mining, in contrast to cultivating, has a significant disadvantage: excavated materials do not regenerate themselves. The aggregates used in a concrete mix, i.e. sand and gravel, are forever lost and transformed without the possibility to recover them in their original form. In the case of sand, the ruthless exploitation of this natural resource already shows dramatic consequences.

Sand is the product of millions of years of natural decomposition of rocks from mountainous areas that is flushed through streams and rivers into the oceans. Yet, the ever-growing building industry mines this natural resource at an alarmingly unsustainable rate; today, twice as much sand is being consumed as it is produced naturally in the same amount of time (Milliman and Syvitski, 1992). The global market for sand is estimated at 15 to 30 billion

tonnes per year, with a value of more than 70 billion US Dollars (SRF TV, 2014; Peduzzi, 2014). Drastic forms of sand mining are appearing all around the globe: the sands of North African beaches are being illegally harvested, with one out of two beaches in Morocco affected; rivers are being dredged and ocean floors scraped; and landmasses are collapsing and islands are eroding away. The consequences reach far beyond the actual mining areas and leave behind devastating traces. Water levels fall in dredged rivers of India, Thailand, and Cambodia, destroying traditional settlements and their modes of life. Dredging seafloors not only harms the local ecosystems, but also has lasting effects on distant sea regions with sediments suspended in the ocean currents. Marine sand mining has irreversible ramifications that will haunt generations to come. Sand shortages lead to increasing illegal mining practices and trade in developing countries (Delestrac, 2014).

Other finite minerals and metals in the earth's crust, are equally affected by today's mining-centric standards and practices of the building industry. In 2005, the German Federal Ministry of Economic Affairs and Energy (BMWi) prognosticated that the global reserves of lead, zinc and tin will be depleted in less than 25 years with then-current rate of mining and consumption (Frondel et al., 2007). Even copper and iron deposits are in danger of depletion if today's 'take, make and dispose' mentality continues (Reller and Graedel, 2009). At the same time, global material extraction has more than doubled in the past 30 years, and is estimated to continue rising: from 65 billion metric tons of raw materials entering the economic system in 2010 to about 82 billion metric tons in 2020 (WU Global Material Flows Database, 2015).

In recent decades, the building industry has reacted to this increasing scarcity of mined resources by aiming for a reduction of construction material use through higher efficiency. An alternative pathway is being presented by the concept of circular economy, a framework 'that is restorative and regenerative by design and aims to keep products, components, and materials at their highest utility and value at all times, distinguishing between technical and biological cycles' (Ellen MacArthur Foundation, 2012). By closing the loop, the circular economy aims to recirculate material resources that are currently being diverted to and lost in landfills, oceans or through incineration.

9.2.1.2 Towards circular material application

In regard to the building industry, the circular economy's design imperative relates to a key requirement: building components must be designed, manufactured, and constructed so that all components used can be disassembled, separated and cleanly returned into their respective technical or biological material cycles to maximise the recovery and reuse of construction materials after a building's life cycle. This may necessitate the redesign of standard building components, the development of innovative new joining systems or

an avoidance of glues and un-recyclable composites. The past 40 years have seen the development of various design guidelines and manuals in an effort to move towards a more sustainable building environment and practice. Notable examples include Walter R. Stahel's *Performance Economy* (Stahel, 1982), Werner Sobek's *Triple Zero* (Sobek and Trumpf, 2008) guidelines or William McDonough and Michael Braungart's *Cradle to Cradle* (McDonough and Braungart, 2002). In these guidelines, materials are considered as borrowed goods, which the client only uses for a certain amount of time and then gives back to the construction market or the natural environment to be reused, recycled, or decomposed.

The development of today's high-performance materials and their assembly are still largely based on the linear economic model, typically aiming to decrease material use by increasing the component's strength. A common result are irreversible (and often non-recyclable) material composites. The paradigm shift towards a circular economy thus not only requires a rethinking of material selection and assembly, but also a new approach towards their structural application, which fully activates the potential of circular recycled or cultivated materials despite their sometimes more challenging material properties. MycoTree addresses both of these aspects by activating a weak material through informed structural design. The utilised new generation of cultivated building materials based on mycelium as a self-assembling glue certainly adapts the above described circular economic model; a buildings that is built from organically grown materials, also can be composted after its initial use and become the source for a new cultivation and building cycle.

9.2.2 Mycelium

Mycelium is the root network of fungi, a fast-growing matrix that can act as a natural binder. It consists of individual hyphae, which grow from mycelium fungal strain spores and consume feedstock containing carbon and nitrogen (Carlile, 1995). Digesting plant-based waste products, such as sawdust, mycelium's dense network binds the substrate into a structurally adequate material.

The advantages of such materials as a potential alternative to traditional building materials are significant. As mycelium-bound building components are organic in matter, they can simply be composted after their original use. Furthermore, mycelium-bound building components can act to reverse carbon emissions through the absorption of carbon. On the other hand, the production of mycelium-bound materials, which is based on locally produced wood and agricultural waste products could address a wide range of economic, environmental and socio-cultural issues. Mycelium can transform low-strength waste products into a high impact, affordable and sustainable material, which could be used in the construction sector.

9.2.2.1 Cultivation

For the MycoTree, the mycelium strain *Ganoderma Lucidum* (*G. lucidum*) of *Basidiomycetes*, commonly known as Lingzi mushroom was chosen. This particular strain grows quickly and sturdily in a typical tropical climate. Material engineering and production of building elements for the MycoTree were conducted in collaboration with the company Mycotech in Indonesia.

The use of agricultural waste products was investigated as an alternative to substrates consisting of waste products from wood, as commonly used in the production of mycelium-bound materials. The recipe of the mycelium-bound material was developed regarding the composition of its substrate as well as the conditions of the mycelium growth aiming at maximising its compressive strength and rigidity while minimising its growth and production time. Waste from sugarcane and the tapioca starch production, specifically the roots of *Manihot esculenta* (cassava), were selected as basis for the production of the final MycoTree components. A mycelium-bound material grown under the same conditions on a mix of woodchips and sawdust of the plant *Albizia Chinensis* wood species (which is widely available in Indonesia) was used as a reference for comparison. The substrate mixtures were inoculated using pre-grown spawns cultivated on corn.

Since mycelium-bound materials have only been introduced recently, no international standard currently exists that prescribes the production and testing methods for such materials. Therefore a modified production method was developed at FCL Singapore using inputs from agricultural mushroom farms (Stamets, 2011) and previous published works (Travaglini et al., 2013; Lelivelt et al., 2015; Pelletier et al., 2017). The process of producing mycelium-bound material can be summarised as follows:

1. Fiberising agricultural waste products into smaller pieces;
2. mixing the substrates with further supplementary nutrition;
3. adjusting the water content to 60-65% in the substrate mixture;
4. autoclaving the substrate in 121°C for 30 minutes to avoid contamination by other competing microorganisms;
5. inoculating the sterilised substrate with *G. Lucidum* spawn;
6. incubating the inoculated substrate in climate conditions of 28°C and 80% relative humidity for 8-12 days for initial colonisation;
7. crushing the colonised substrate and placing them into moulds for further incubation in the same climate conditions for another 6-9 days;
8. removing the incubation moulds and exposing the mycelium-bound material to air for chitinous skin development;
9. transferring the fully grown specimens into a ventilated oven with the temperature of 80°C for 24-48 hours in order to reduce the moisture content to less than 10% and consequently preventing further growth.

Figure 9.1: Production of the mycelium components: a) collection and processing of substrate such as agricultural waste or saw dust; b) packaging of substrates in bags and inoculation of mushroom hyphae; c) sterile environment to keep out competing bacteria or mushroom spores, and sufficient watering of the samples; d) monitoring of ideal conditions in terms of humidity and temperature; e) incubation in bag logs; f) breaking down the colonised substrates; g) transferring the broken down substrates into the moulds; and h) final growth in a humidity controlled bags.

9.2.2.2 Testing

Mechanical properties of the various mycelium-bound materials including compressive stress (σ) and modulus of elasticity (E) were determined under static compressive loads using a Shimadzu 100 kN Universal Testing Machine (UTM) and cubic specimens of 50 x 50 x 50 mm. Tests were conducted according to ASTM D3574, a standard for testing flexible cellular materials (Travaglini et al., 2013).

Several images were taken using Scanning Electron Microscopy (SEM) to ascertain the cellular structure of the mycelium hyphae, growth patterns and the porosity of the newly developed mycelium-bound materials. A JEOL 5410 SEM was used at a setting of 1.0 kV. The samples were coated with gold-palladium to prevent a charging effect. Compressive stress at 5% deformation of the specimens was recorded during testing as it was defined as the allowable stress limit in the structural design of MycoTree components.

Table 1 displays a comparison of density, stress at 5% deformation and modulus of elasticity of two sets of mycelium-bound composite samples with two different substrate compositions: (i) sawdust and woodchips, and (ii) sugarcane and waste of cassava roots. While the first mixture represents a commonly used substrate for mycelium cultivation, the second substrate mixture was specifically developed for the components of the MycoTree.

The compressive stress that developed in mycelium-bound samples grown on (i) woodchips and sawdust at 5% deformation amounts to an average of 0.17 MPa with an average elastic modulus of 3.97 MPa. The mycelium-bound materials based on (ii) sugarcane and waste of cassava roots developed an average compressive stress of 0.61 MPa at 5% deformation with an average elastic modulus of 22.70 MPa, hence achieving superior material properties.

a)

Substrate composition	Density (kg/m³)	Average elastic modulus (MPa)	Average compressive stress at 5 % deformation (MPa)
woodchip and sawdust	420	3.97	0.17
sugarcane and cassava roots	440	22.70	0.61

b)

Figure 9.2: Compression test of a mycelium sample: a) mushroom mycelium is a relatively weak material with a very low bending and tensile capacity. However, tests showed that the compressive strength of the material is sufficient for certain structural applications; b) physical and mechanical properties of mycelium-bound material samples with different substrate compositions.

9.2.3 Strength through geometry

The advent of advanced computational tools, in conjunction with state-of-the-art research in recycled or cultivated materials, has the potential to significantly improve the relevance and applicability of humble, often low-strength materials in designing of efficient and expressive structures. A material's weakness can be turned into a strength by strategically placing them in an appropriately designed structural geometry. Through informed structural design methods, compression-only structures can empower weak materials to become load-bearing elements at an architectural scale. Compression-only structures significantly reduce the amount of internal stresses, and subsequently reduce the amount of material required to carry the applied loads.

In developing contexts where resources, infrastructure and funds are often inadequate in supply, designing with locally available materials can be a solution for the global scarcity of conventional building materials, by capitalising upon what is usually abundantly available: manpower. When building with locally-sourced materials like soil-pressed tiles, workers from the community can be trained to make and install the tiles, as in the SUDU - Sustainable Urban Dwelling Unit in Addis Ababa, Ethiopia a collaboration between the Ethiopian Institute of Architecture, Building Construction and City Development EiABC, Dirk E. Hebel and Philippe Block (Hebel et al., 2015). Shell structures for the ceilings and the roofs with innovative design elements such as stabilising fins, drastically reduce the need to import and transport expensive, engineered materials because the compression-only structure has been designed for weak materials with low tensile capacity.

In a more urban context, building with recycled materials reduces the rate at which finite resources are being depleted. For the 2015 IDEAS CITY Festival in New York City, the teams of ETH Zurich's Professorship Hebel and the BRG designed and constructed a temporary vault structure built from compressed tetra-pack panels. The shape of the vault, which spanned a neighbourhood park between two buildings in lower Manhattan, followed the flow of forces and was optimised so that the stresses in the structure were predominantly compressive, enabling a considerably weak product—essentially made from discarded beverage containers—to act as a structural material. Following the festival, the panels and other materials used were recycled or reused (Heisel, 2015). Research in development and application of cultivated materials is the next step in these collaborative investigations.

As we learn more about their properties and undiscovered potentials, materials that were previously deemed undesirable because they were perceived as vernacular (i.e. bamboo) or even as hazardous waste (i.e. mycelium) present new possibilities for applications in efficient, compression-only structures. The MycoTree at the 2017 Seoul Biennale of Architecture and Urbanism in Korea was built to test, validate and showcase this potential.

Figure 9.3: Previous collaborations between the KIT and the BRG, showcasing compression-only shell structures from low-strength materials: (left) the Sustainable Urban Dwelling Unit in Addis Ababa, Ethiopia; and (right) and WasteVault – the ETH Pavilion at the 2015 IDEAS CITY Festival in New York City.

9.2.4 Structural design

The development of engineered materials, such as concrete or steel, is largely focused on making the materials stronger by increasing their allowable stresses. Mycelium-bound materials offer significant ecological advantages over established engineered materials on the one hand, but comparably low structural strength on the other. To build with materials that are weak in tension and bending, the use of good geometry is essential to guarantee static equilibrium of the structure through contact only, i.e., through compression.

The MycoTree is designed to be a compression-only structure, where all of the mycelium components are carrying only compression forces. The geometry of MycoTree is intended to show that spatial funicular structure are not limited to arches or vaults, but also can be branching tree structures. Achieving stability through geometry rather than material strength opens up the possibility of using weak materials such as mycelium in structural applications. While the mycelium tree portion of the structure is a metaphor for a load-bearing component of a building, the bamboo grid at the top of the structure represents a floor or a ceiling of a building that applies load to the mycelium structure. The grid is primarily active in tension and is intended to resolve and internalise the horizontal forces at the top of the tree. Although the corners of the grid are cantilevering, the grid is designed to behave conceptually like a rigid slab.

9.2.4.1 Tree structure

The geometry of the tree structure was generated through polyhedral transformations (Lee et al., 2016) of $\Gamma^{\perp}$. Figure 9.7 are some of the initial design options. Ultimately, the form finding of the structural geometry was guided by several key constraints. First, in order to minimise the geometric complexity while maximising fabricability, all nodes were limited to a valency of four, meaning there are no more than four mycelium elements

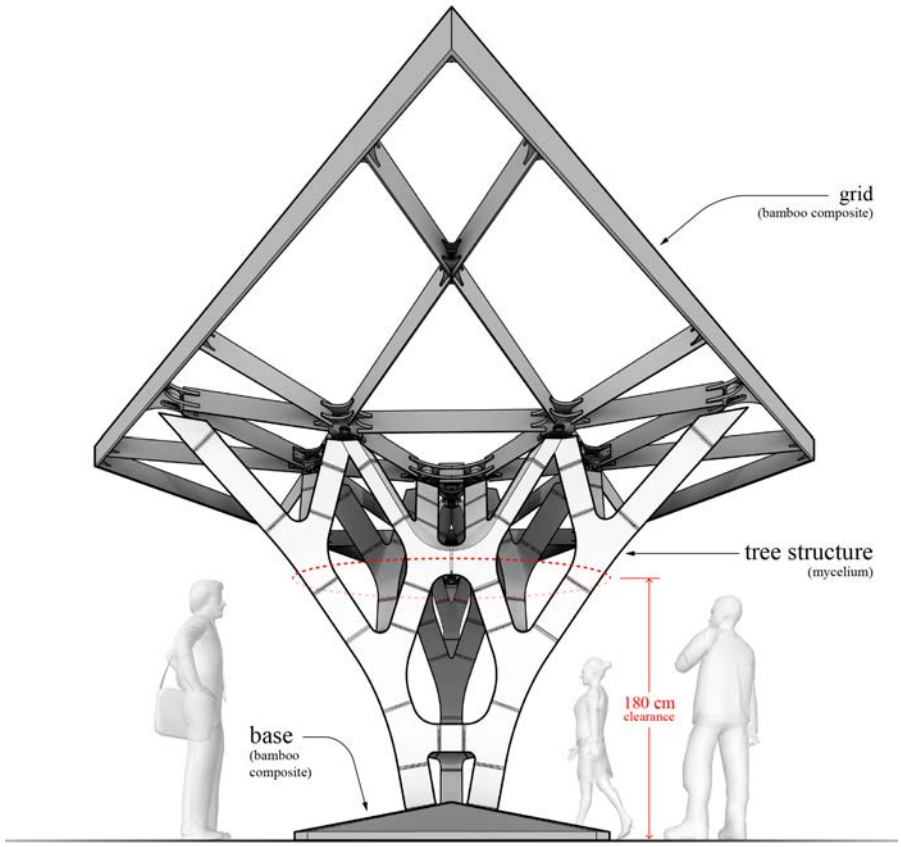


Figure 9.4: Perspective drawing of the completed structure.

coming together at any of the nodes. Second, the angle between any two linear mycelium members was constrained to be larger than 30 degrees. In addition, the centre-to-centre distances between any pair of nodes were constrained to be at least 40 cm such that smooth transitions could be made between any two directly adjacent nodes. Lastly, the maximum length of any linear mycelium member was limited to 60 cm in order to avoid potential buckling.

Figure 9.5 and 9.6 summarises the steps of the form finding process. $\Gamma^{\perp}$ of step 8 in Figure 9.6 shows the exploded view of the cells representing the nodes at the top of the structure. The top cells as illustrated represent the compression only forces in the entire structure, with external horizontal forces applying compression to the structure. These external horizontal forces can also be internalised by converting the top members into tensile members.

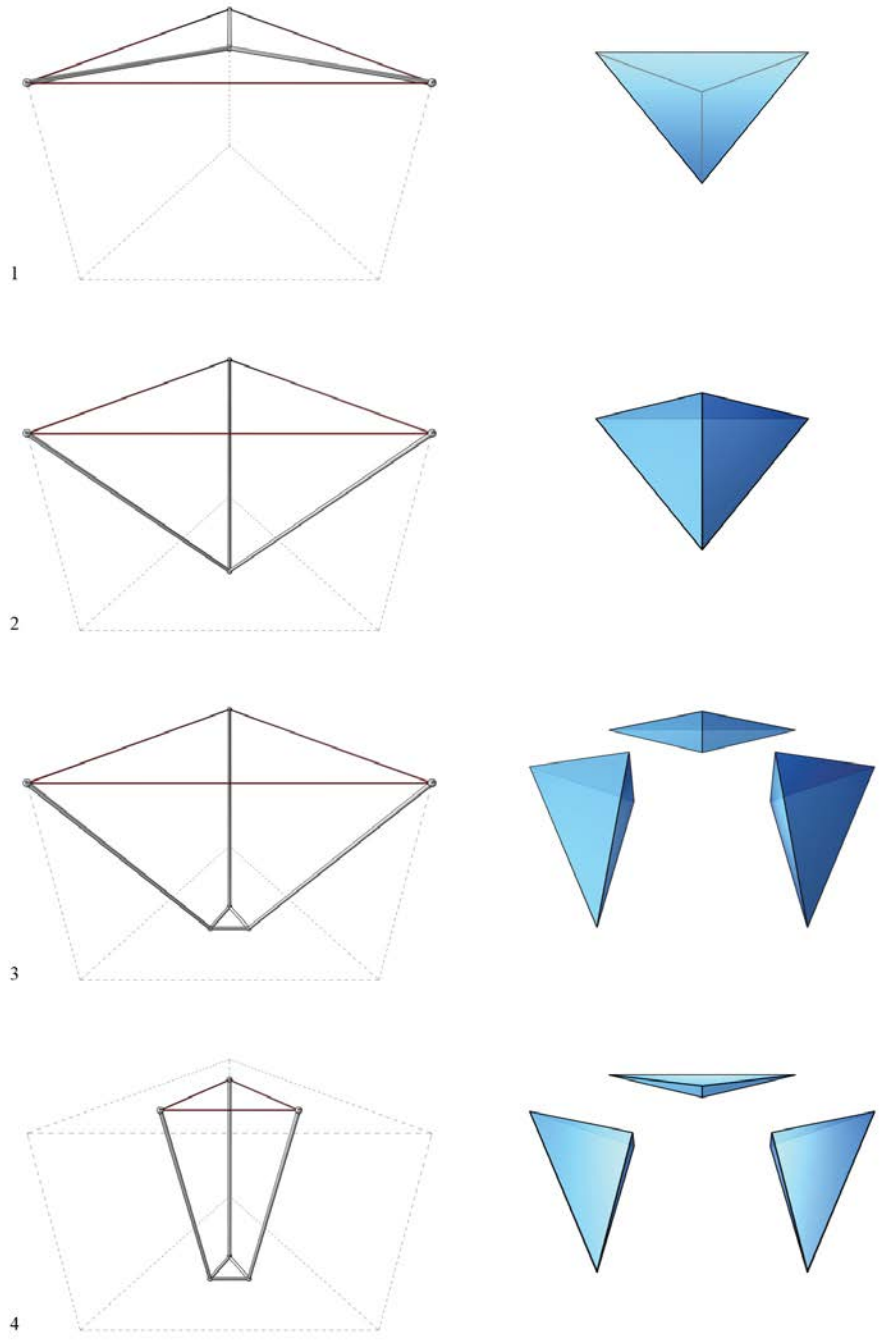


Figure 9.5: Steps 1 through 4 of the form finding process.

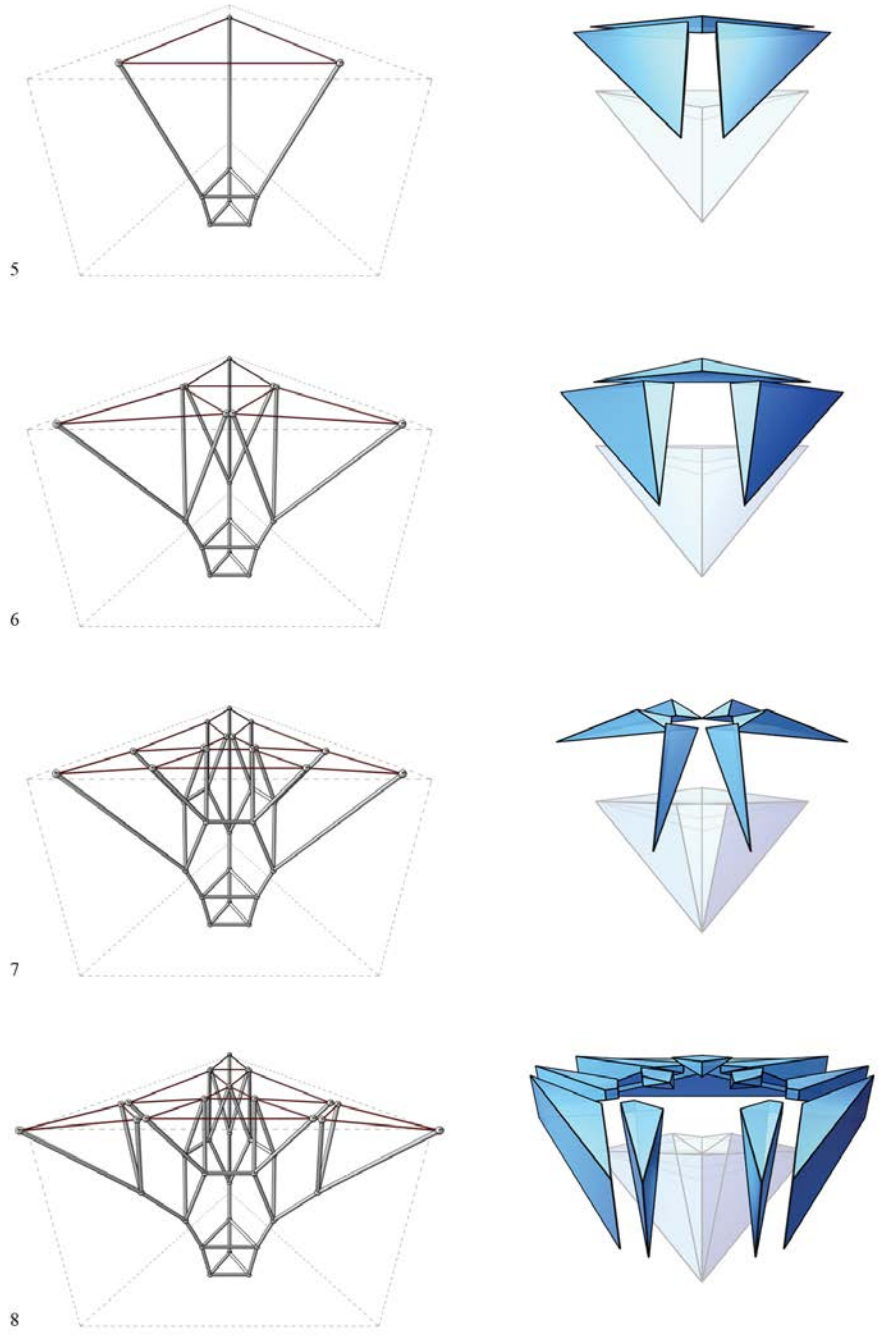


Figure 9.6: Steps 5 through 8 of the form finding process.

Figure 9.7: Alternate design options of the MycoTree.

9.2.4.2 Grid structure

The polyhedral form diagrams are in static equilibrium if and only if specific loads are applied at specific locations of the structure. Since the built structure is not designed to have any additional applied loads other than the weight of the grid, it is important to distribute the weight of the grid down to the structure with the correct proportions to ensure static equilibrium of the entire structure. The applied design loads for MycoTree is defined by the top faces of the global force polyhedron, $\Lambda^{\perp}$ (Figure 9.8-a1). Each of these faces have different areas, and therefore represent applied point loads with varying magnitudes (Figure 9.8-a2). Furthermore, the areas of these faces are not necessarily equal to the actual tributary area of the corresponding node in the form diagram.

Especially for this project, where the extents of the square grid does not match the triangular footprint of the top of the tree structure, the discrepancy between the design loading and the actual loading is greater. The face areas of $\Lambda^{\perp}$ can be optimised to match and correspond to more realistic loading conditions, which would result in a Γ with different geometry. Conversely, the footprint of the grid could be modified to provide the required tributary areas for each of the nodes. However, the dimensional restrictions of the exhibition space were the governing factors of the overall design; the room had a footprint of just 4 meters by 4 meters in plan; a head height clearance of 180 cm was to be maintained all around the base of the structure; and it was aesthetically important to spread the outermost branches as much as possible.

Figure 9.8-b1 shows the square grid, with the tributary area shown for each of the top nodes of the tree structure. If the grid structure is treated as a solid "slab" without any openings, the nodes towards the south side of the structure will be more loaded as expected (Figure 9.8-b2). Instead of a solid slab, the grid is designed to be made of composite bamboo strips (Figure 9.8-c1). Bamboo strips are light and have high tensile capacity, which are ideal properties for the top members that internalise the horizontal forces at the top of the structure. Because the weight of the grid was intended to act as the

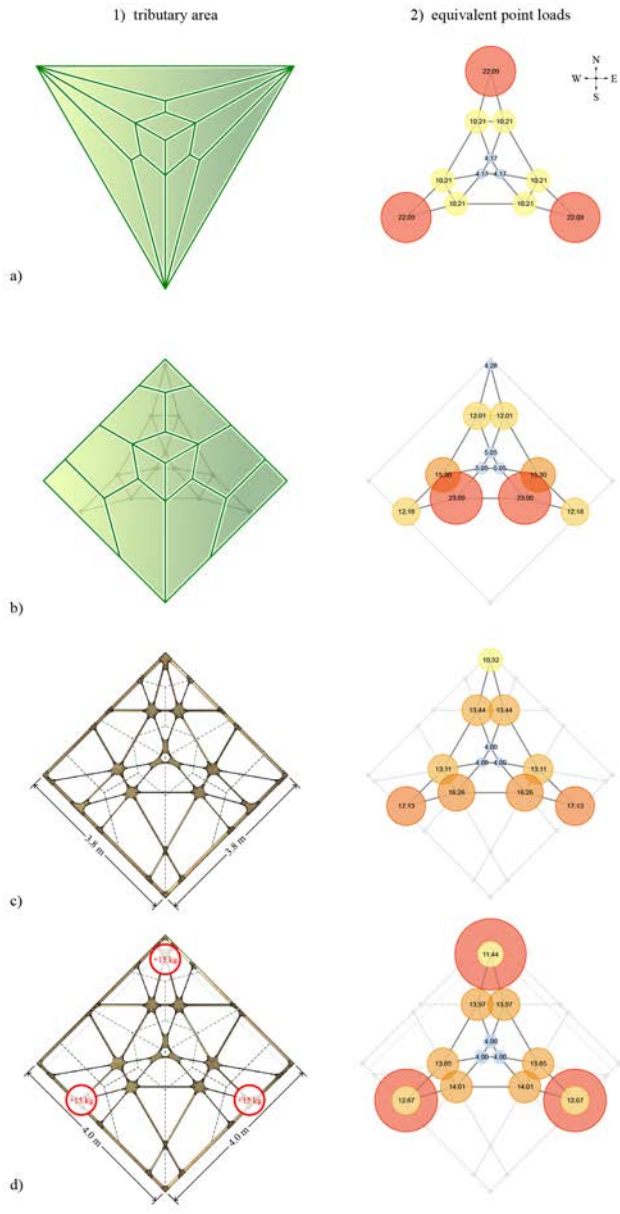


Figure 9.8: Calibrating the applied loads (2) on the structure to match the initial design load distribution (a1) from the polyhedral force diagram: a) the top face of the global force polyhedron (1), and the equivalent applied loads on the structure (2); b) the tributary area of the grid if treated as a solid "slab" (1), and the equivalent load distribution on the structure (2); c) the initial grid design with a plan dimension of 3.8 x 3.8 meters, and the equivalent load distribution on the structure (2) where the south side of the structure is more heavily loaded; d) adjusted grid design with a plan dimension of 4 x 4 meters (1) with additional weights added at the three outermost pickup points, and the equivalent load distribution (2) that is more proportionally closer to the design load distribution (a2).

main applied load of the structure, the lightness of the bamboo composite strips actually does not sufficiently provide enough loads, especially around the periphery of the structure (Figure 9.8-c2).

Within the boundaries of the exhibition space, the footprint of the grid can be extended in the northwest and north east directions by 20 cm, to distribute more of the weight of the grid towards the north side of the structure. Ultimately, the weight of the grid was not sufficient to provide the necessary amount of applied loads at the three corners of the tree structure. Additional weights of 15 kilograms at these three points were added above the grid structure to achieve the correct distribution of the applied forces (Figure 9.8-d1, 2).

9.2.5 Fabrication

One of the most rewarding benefits of using 3D graphic statics to design spatial structures is that the geometry of the structure is polyhedral by construction. Without the need for additional optimisation processes, the massing of the complex nodes of the tree structure can be developed through polyhedral transformations that only use planar and flat surfaces (Figure 9.9-b, c, d). As a result, the geometry of the moulds for the nodes can be materialised using only developable surfaces which can be laser or CNC cut from readily available and easy-to-recycle sheet materials.

For this project, transparent acrylic sheets were used as the main enclosure of the mould. Although a more environmentally responsible material is preferred, transparent plastic was chosen mainly due to the need for the visual monitoring of the mycelium mixture during the growth process. The mycelium mixture is prone to contamination during growth, and the earlier it is detected, the sooner a new batch of growth can be started. However, as the understanding of the material improves in the future, this contamination can be better contained and monitored through other, non-visual means, and transparent plastic will no longer be needed. In order to reduce the number of mechanical connections and adhesives, teeth-and-slit joints are used for the walls of the mould for efficient assembly and clean expressions of the curved edges after demoulding (Figure 9.9-f). For the straight linear elements, the same procedure can be used to generate the moulds.

Once the mould has been assembled, it can be filled with the mycelium mixture, which requires incremental compacting and additional filling as the mixture gradually densifies. Due to the irregular geometry of the nodes, there is no one, single opening from which the mould could be filled and compacted. A mechanism that compacts the mixture from all of the openings simultaneously had to be designed (Figure 9.10). Because it was not feasible to industrially manufacture such a device or mechanism, an ad hoc solution using simple kit of modest materials (laser and CNC cut plywood, wooden

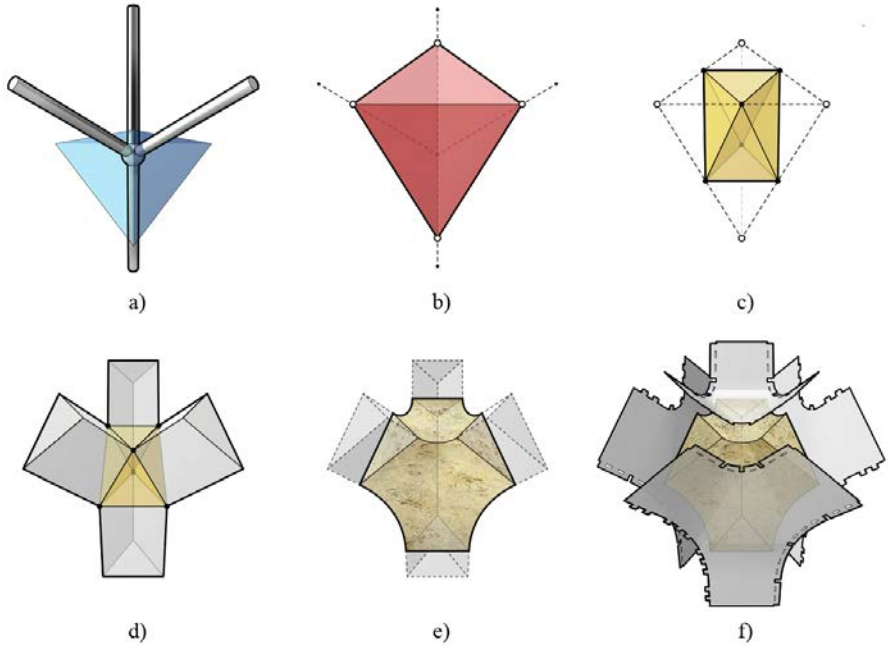


Figure 9.9: Development of the mould geometry: a) the orientation of linear members at a node; b) convex hull of points that are equidistant along each of the members from the centre of the node; c) dual transformation of the convex hull; d) extrusion of the dual hull faces along the orientation of the corresponding members; e) filleting of the edges; and f) the subsequent, developable surfaces of the node mould.

dowels, strings, etc) was needed. The simplicity of the mechanism and the constituent parts was important also because the workers at the rural mycelium production facilities had limited experience and resources with regards to digital fabrication. Using a set of embedded pipes held together by a small 3D printed joint in the centre of the mould, a string can be placed through the pipes and tied into a knot inside the 3D printed joint. The loose end of each strand of the strings are then tied to a "turnbuckle." As this turnbuckle is twisted, the string shortens toward the centre of the mould. The turnbuckle then presses against the push pipe, which subsequently presses the push plates inward and compact the mycelium mixture inside.

Because the main enclosure of the mould is made of thin sheet materials, additional reinforcements and stiffeners are needed to prevent bulging or busting of the enclosure. As mycelium mixture is compacted repeatedly over time, more mycelium mixture has to be put into the mould to reach the ultimate target shape of the component. Once the mycelium is densely compacted, and the end plates have reached their final target locations inside the mould, the entire mould is placed inside sealed bags where the final growth phase takes place.

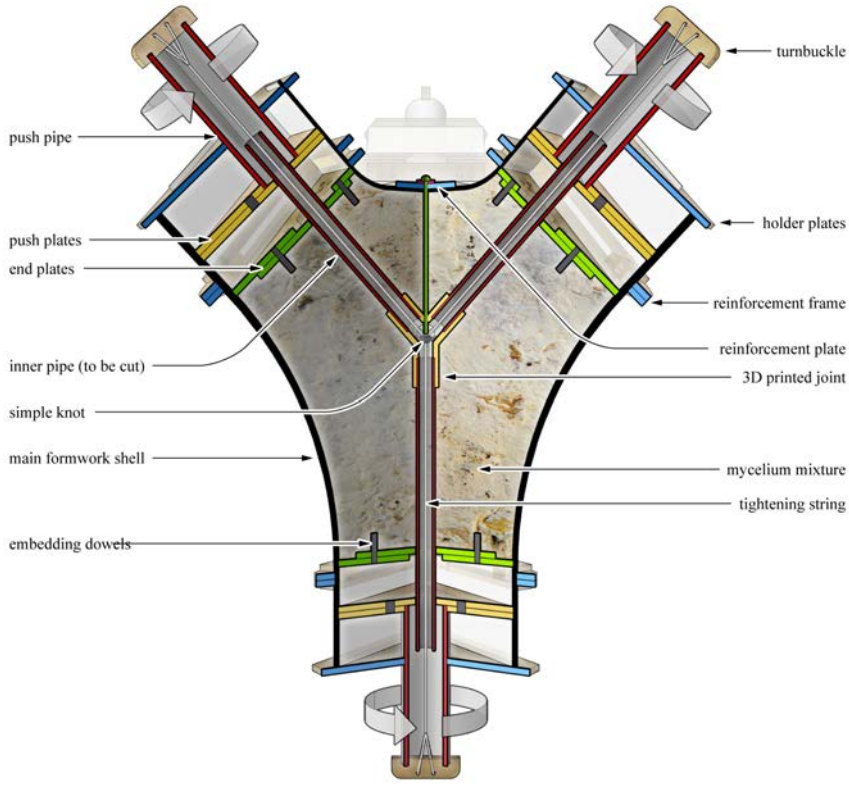


Figure 9.10: Cross section view of the mould and the compacting mechanism.

While the mycelium components are designed to carry their self-weight plus the weight of the grid through compression, the grid on top is acting primarily in tension; it holds the branching linear mycelium elements at the top together and prevents them from falling over. For this reason the grid is also constructed out of the bamboo composite material, utilising CNC-cut, 8 mm thick bamboo composite boards. For ease of assembly and for minimisation of steel connections, all joints of the grid are designed with slit-and-slot connections that can be easily assembled on-site and secured using simple locking elements and wooden dowels.

9.2.6 Assembly

MycoTree took seven days to assemble on site. All of the elements were pre-fabricated in Singapore and Indonesia, and were shipped to the exhibition location a week prior to the opening of the Biennale. The bamboo grid was first constructed on the floor of the space. The grid was then hoisted up to the target height, and held in place with crates and boxes from underneath. Once

Figure 9.11: On-site assembly of the MycoTree.

the base was located on the floor, the mycelium components were pieced together starting from the base of the structure. Because the structure is in equilibrium only when every component is in place and the weight of the bamboo grid has been activated, temporary scaffolding was used throughout the assembly. Boxes, crates and straps were used as bespoke supports as needed during the assembly.

To ensure clean connections and even load transfers from one element to another in the final structure, the ends of each individual mycelium-bound components are capped with plates of bamboo composite material. These plates are also used for compacting the mycelium mixture into the moulds, and are eventually secured to the mycelium-bound components with dowels. This bamboo composite material has been developed and optimised over the past five years at FCL Singapore in order to withstand high tensile and bending forces (Hebel et al., 2014).

9.2.7 Data and facts

The MycoTree supports a four by four meter bamboo grid at a height of three meter over ground. It consists of 36 linear members of maximum 60 cm in length as well as 15 nodal elements of mycelium-bound material. The bamboo grid weighs approximately 134 kg in total, the overall weight of mycelium-bound members amounts to approximately 182 kg.

The triangular sections of the mycelium-bound members are sized for a limit stress value of 0.1 MPa. The self-weight of the structure was considered with a safety factor of 1.35 during design. The structure can withstand an accidental horizontal point load of 0.7 kN at a height of 1.27 m (arm height) from the ground at a single node. However, because the structure is designed to be primarily in compression-only, any type of horizontal loading should generally be avoided.

9.3 Outlook

The inherent geometric properties of polyhedral force diagrams can be exploited for informed design of structural geometry, as well as fabrication geometry. The MycoTree project has shown how the geometry of the polyhedral cells can be used to develop the mould geometry of the joint and bar elements of the spatially branching structure. The polyhedral properties of structures form-found through 3D graphic statics can also be useful for designing claddings of freeform roofs and facades, where the planarity constraint is enforced as a byproduct of the form-finding process. A free-form surface which can be assembled with flat sheet materials such as glass panels, will significantly improve the efficiency of fabrication and construction.

The example shown in Figure 9.15 demonstrates this potential. Suppose that a design for a roof structure generated through polyhedral subdivisions is being considered. Through polyhedral transformations of $\Gamma^{\perp}$, various equilibrium structures can be explored, while the faces of Γ are inherently constrained to be planar. This built-in planarity property of Γ is ideal for both fabrication and construction of the geometry without the need of any additional optimisation processes.

The roof will be placed on six existing columns, therefore no horizontal reactions are allowed at the base of the roof. Rather than using straight cable ties to counterbalance the horizontal thrusts, the designers want to explore a more integrated cable-net-like design to maximise the visual and spatial experience from below. Furthermore, the designers want to consider using two main cables that have constant force throughout their length. Constant force members in trusses and various other structures are beneficial in that it allows a single cross section to be used, and the material is utilised to its full capacity throughout its length (Allen and Zalewski, 2009). Using a cell network for the cable-net, the force distribution can be controlled more precisely; the cell faces corresponding to the perimeter cables of the structure are constrained to have the same areas. At the same time, the geometry of the primary compression structure above can be constrained to remain purely polyhedral with planar faces, which is more ideal for fabrication-driven constraints.

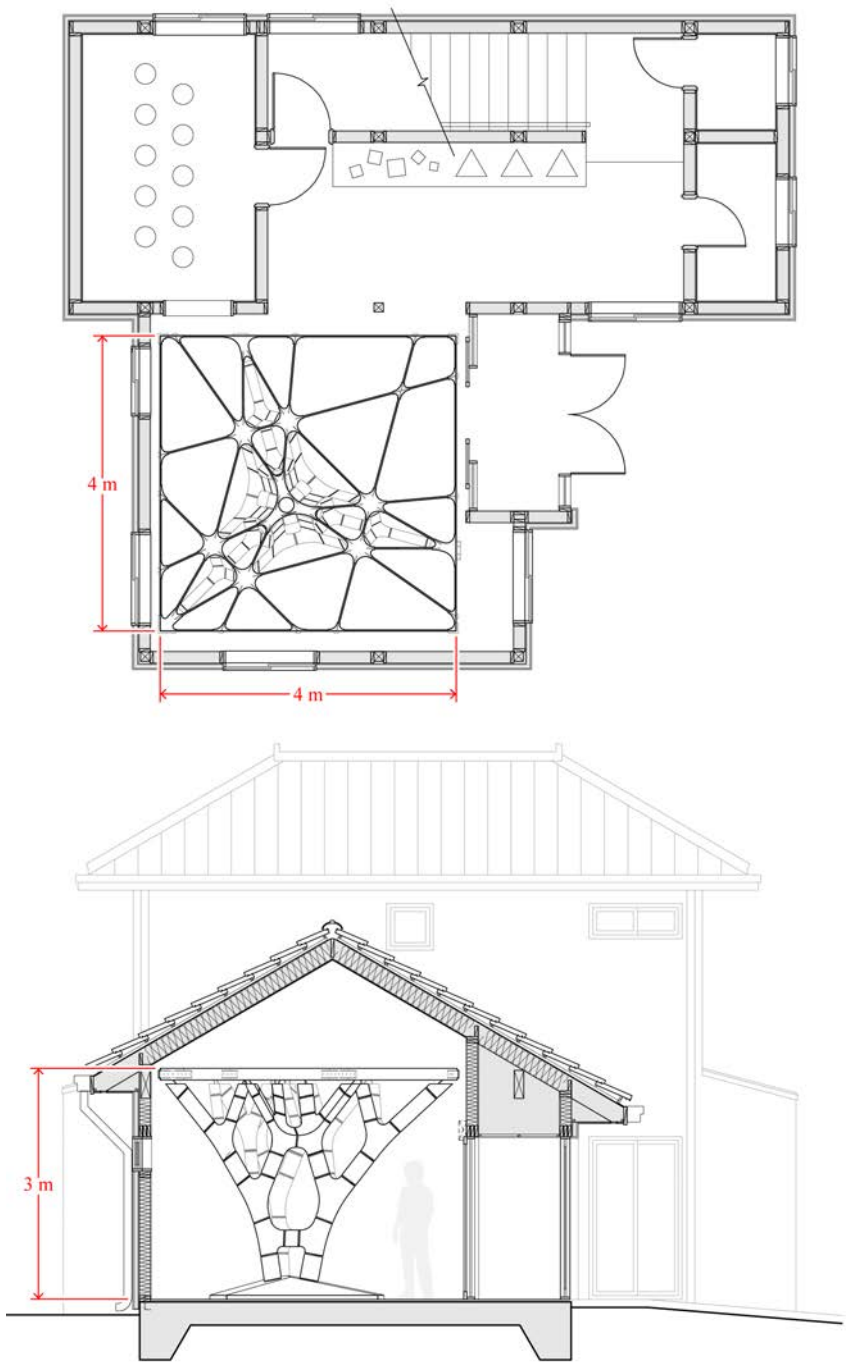


Figure 9.12: Plan and section of the completed exhibition space. MycoTree is roughly three meters tall, and four meters by four meters in plan.

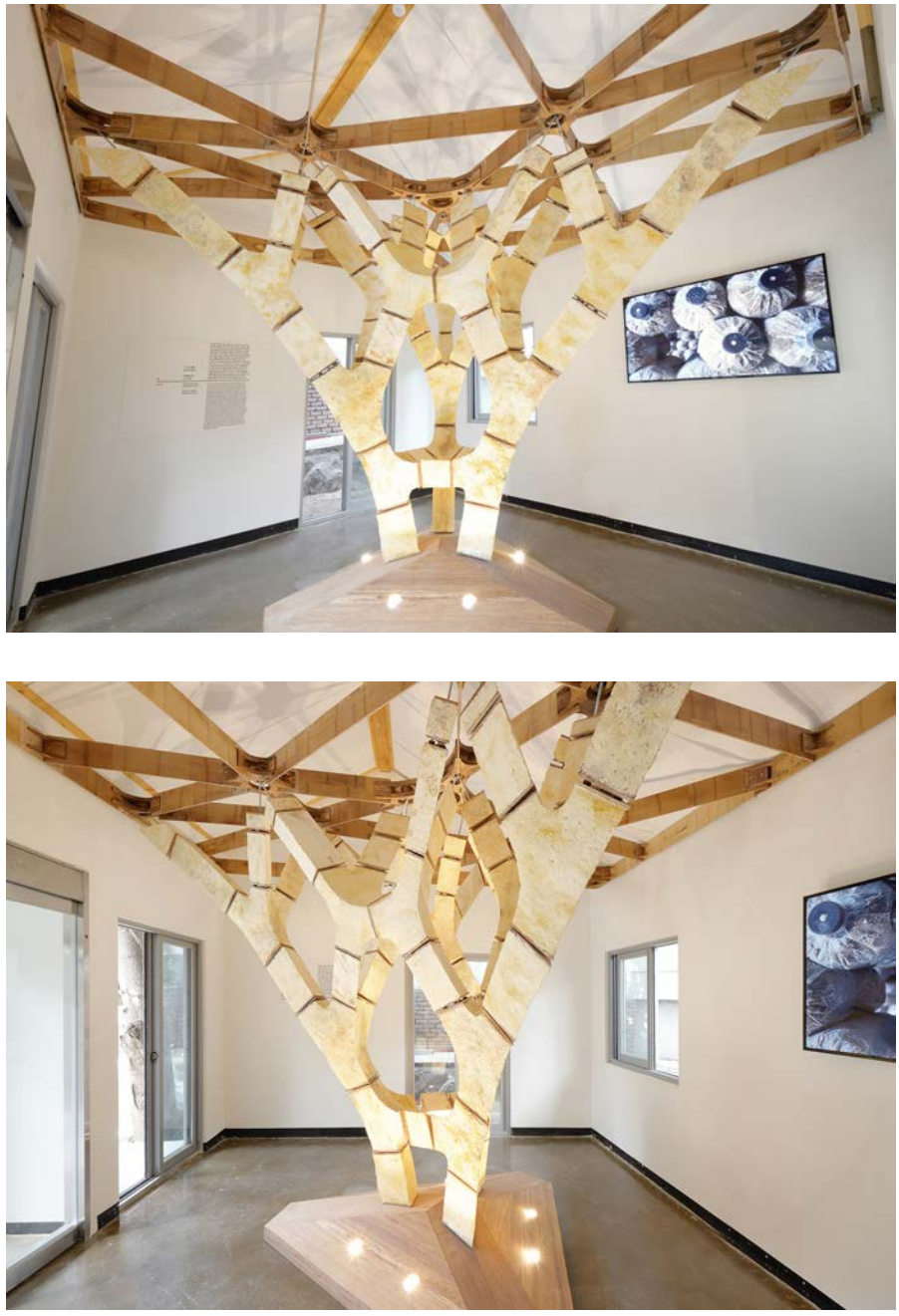


Figure 9.13: Completed MycoTree in the exhibition space.

Figure 9.14: The rest of the "Beyond Mining - Urban Growth" exhibition.

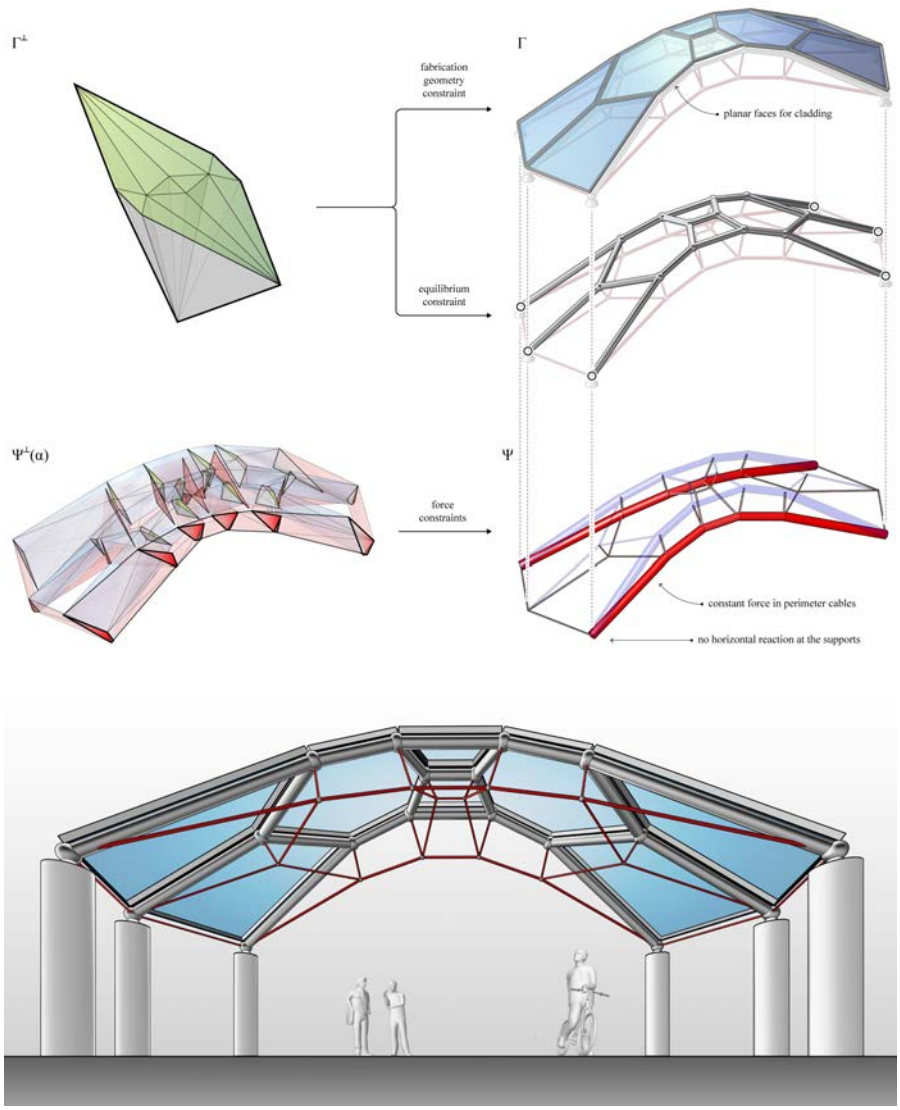
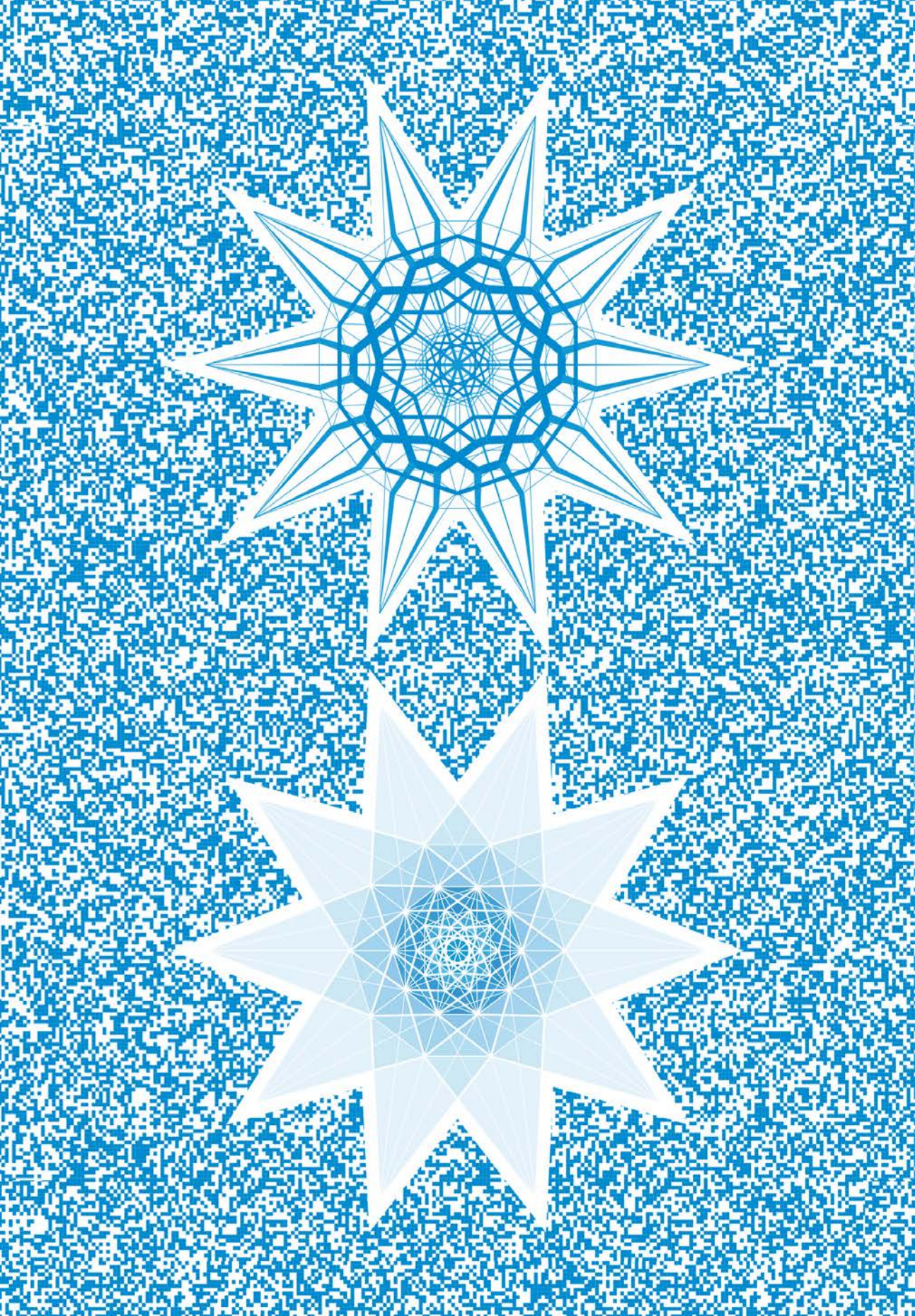


Figure 9.15: Combinatorial application of polyhedral design methods for a roof structure with no horizontal reactions. Transformation of $\Gamma^{\perp}$ is used to generate the polyhedral geometry of the structure and the outer layer of the roof, where the equilibrium and fabrication constraints are dominant. For the cable-net, $\Psi^{\perp}$ is used to enforce a constant-force constraint on the two main cables.

The designer's ability to maintain and impose the built-in polyhedral constraints to either the entire structure or only specific parts of the structure as needed, is a unique feature that can potentially be applied to constrained form finding and fabrication-aware design explorations. 3D graphic statics is primarily a computational form-finding tool, which visualises spatial structures as a network of lines without any thickness or materiality. In future research, it is important to start investigating how the inherent properties of polyhedral form and force diagrams can positively impact the fabrication geometry and process, so that the complex spatial structures can be materialised and assembled in a feasible manner and ultimately result in a meaningful structure.



Part IV

Conclusions

10 Conclusions

This dissertation presented a new computational framework for 3D graphic statics. The framework was implemented as an open-source computational library, **`compas_3gs`**, which makes 3D graphic statics available to a wide audience for design, analysis, research and educational purposes. The chapters of this dissertation presented: the motivation for this doctoral research; the practical relevance and originality of the work; a review of the relevant literature and state of the art; new datastructures and operations reinforced by theoretical foundations; and new 3D graphic statics applications that are now possible with the framework.

This final chapter presents the concluding statements and discussions of the dissertation. It gives a summary of the unique contributions made, while reflecting back on the initial problem statements outlined in Chapter 3. The practical relevance of the contributions to architecture, structural engineering, computational geometry and digital fabrication are discussed. It also identifies limitations of the presented research and speculates on directions for future research. The chapter concludes with final reflections and closing remarks.

10.1 Contributions

The specific contributions of this dissertation as they relate to the problem statements outlined in Chapter 3, are as follows.

1. **Addressing realistic boundary conditions**

    - Introduction of the Extended Gaussian Image (EGI) as the topological mapping of a spatial system of forces
    - Computational implementation of the EGI, and demonstration of its usefulness in understanding, processing and manipulating the topology and geometry of single polyhedral cells
    - Iterative method for computing the geometry of global force polyhedrons using the EGI and area-targeted planarisation algorithms
    - Integration of reaction force location and magnitude constraints during the construction of the global force polyhedron

2. **Topology generation method beyond subdivision**

   - Formulation of multi-cell polyhedrons, a datastructure based on halffaces for computational modelling and representation of polyhedral form and force diagrams
   - Clarification of the topological structure and organisation of polyhedral force diagrams as they relate to compression and tension forces in the corresponding form diagram
   - Expansion of geometric and topological transformation operations beyond subdivision for multi-cell polyhedrons (manipulations and additive transformations)
   - Various algorithms for global manipulation of multi-cell polyhedrons (planarisation, reciprocation, arearisation and constructors)
   - A new workflow and design pipeline for unbiased generation of new structural topologies

3. **Limitation to polyhedral forms**

   - Theoretical background for cell networks, which contain a collection of polyhedral cells that individually represent the local equilibrium of a node, but together represent the global equilibrium of the entire structure
   - Formulation of cell network datastructure, a hybrid datastructure that can represent the equilibrium of non-polyhedral structures
   - Use of prismatic polyhedral cells for the interpretation of 2D equilibrium, which: 1) extends the concepts and methods presented in the framework to 2D as well as 3D structures; and 2) allows exploration of structures with 3D and 2D nodes
   - A technique for calibrating polyhedral form diagrams to address realistic loading cases (tributary area loads)
   - The user's ability to place point loads at any node of the structure
   - A design workflow and pipeline for interactive force design, where polyhedral face areas (internal force magnitudes of the members in the structure) are explicitly constrained and controlled
   - Introduction of non-polyhedral structural typologies that can be explored with cell networks: 2D and 3D combined structures; structures with overlapping members; and structures with non-polyhedral geometry

4. **Materialisation of spatial structures**

   - Validation of 3D graphic statics as a tool for developing compression-only spatial structure, enabling weak organic materials to be used structurally at an architectural scale
   - Technique for materialising and developing fabrication geometry from the polyhedral force diagram, using affordable, flat sheet materials and 2D CNC cutting

5. **Insufficient user control and visualisation**

   - An algorithm for visualising the unified diagrams for multi-cell polyhedrons and cell networks
   - Techniques for exploiting built-in Rhinoceros visualisation and dynamic drawing functionalities to enhance the graphical user interface for 3D graphic statics applications

6. **Lack of computational library for 3D graphic statics**

   - Computational implementation of the presented framework in an open-source library, `compas_3gs`
   - A detailed online documentation of the library, including: installation instructions; summaries of the related theoretical background using texts, diagrams and examples; a forum and reference guide for future contributors and collaborators; and tutorials showing how to use various components of the framework for specific design or research objectives

10.2 Discussions

This section discusses the practical relevance and the potential impact of the contributions from the perspectives of the relevant disciplines: architecture, structural engineering, computational geometry, and digital fabrication.

10.2.1 Contributions related to 3D graphic statics

When graphic statics was first introduced during the 1800s, the tools required to construct reciprocal diagrams were simple: sheets of paper, a ruler and a pencil. What made graphic statics a revolutionary tool at the time is how a simple set of tools can be used to analyse complex structures which would otherwise require tedious, manual calculations. With the revival of graphic statics through the help of computation, the tools needed to construct interactive polyhedral reciprocal diagrams in 3D

require much more than just a CAD software. **compas_3gs** is intended to be the computational "paper," "ruler," and "pencil" for 3D graphic statics applications. As 3D graphic statics is one of the emerging areas of research in computational graphic statics, **compas_3gs** allows researchers within the graphic statics community to start using 3D graphic statics ideas and methods without having to implement an entire computational framework from scratch. Establishment of a common computational language also allows researchers to focus more resources and time on developing new application ideas rather than reinventing the wheel. The common platform allows researchers to exchange ideas in a more fluid manner, and open up opportunities for smoother collaborations regardless of the institutional affiliation, professional background or personal experience.

10.2.2 Contributions related to architectural design

For architectural designers, **compas_3gs** and presented design methodologies can be useful for exploring structurally-informed geometries during early stages of design. With the explicit control of the geometry of both the form and the internal forces of a structure, architects have the ability to generate and investigate a wide range of equilibrium structures more rapidly without having to analyse each variation. The ability to control the geometry of spatial structures is an important step forward in allowing architects to sketch and model in full 3D. A pipeline that allows the generation of the structural geometry solely from the geometry of the force diagram means that new design typologies may emerge, leading to new discoveries and ideas that would be difficult to conceive with conventional tools and the designer's biases. The manipulation and transformation functionalities of **compas_3gs** also allow architects to modify and refine structural geometries in an intuitive and interactive manner. Graphical representation of the forces and strategic visualisation of the polyhedral force diagrams have the potential of improving the architect's understanding of how the structure is behaving, and providing insights on how certain design changes have an impact on the structural performance.

10.2.3 Contributions related to structural engineering

The FE software used by structural engineers today is a powerful analysis tool, so long as the geometry of the structure is defined and provided as the input. The presented computational framework offers structural engineers new methods of addressing certain problems within the discipline that would otherwise be difficult with conventional tools. The ability to precisely control the face areas during the construction and transformations of force diagrams means that the internal forces of the structure can be controlled more carefully by the engineer. Area-controlled construction of global force polyhedra provides an explicit method of investigating different

reaction force distributions while taking into account various load cases and boundary condition constraints. Generalised cell networks enable engineers to more precisely control the geometry of polyhedral as well as non-polyhedral structures, while force-driven constraints are still imposed. As a result, some of the challenging problems in existing engineering frameworks such as topology generation and manipulation in strut-and-tie models, and discretisation of topology optimisation results can be addressed. Generalised cell networks also allow polyhedral 3D graphic statics to be implemented in combination with other numerical or form-finding methods. Whereas conventional FE results communicated through charts, graphs and colour scales can only inform whether a structure is stable or not, graphical representations of structural equilibrium can provide an entirely different way for engineers to visualise and understand a structure's behaviour. The geometry based understanding and design of structures can ultimately provide new insights that make the engineers smarter.

10.2.4 Contributions related to digital fabrication

The reciprocal diagrams in graphic statics traditionally refer to a form and a force diagram that have a reciprocal relationship between them. However, the diagrams themselves have no material information within them. The lines of the form diagram is literally the geometry of the structure, while the geometry of the force diagram (edges in 2D and faces in 3D) represents the internal forces of the structure. As a result, the materialisation of the form diagram is typically a separate process that is implemented after the form-finding process. Through the design, fabrication and construction of MycoTree, this dissertation has demonstrated that the geometry of the force diagram in combination with the geometry of the form diagram can be directly used to generate the massing and a logical fabrication geometry of the structure. Due to the inherent planar properties of the polyhedral force diagram, it opens up many possibilities for digital fabrication applications based on polyhedral reciprocal diagrams that would typically require an entire separate optimisation or modelling procedure.

10.2.5 Contributions related to computational geometry

The language of graphic statics is fundamentally based on geometry. As a corollary, the language of computational graphic statics is inherently based on theories and applications of computational geometry. The development of a computational 3D graphic statics framework hence requires a robust and flexible computational core for processing geometries of polyhedrons and networks of polyhedrons. The 3D graphic statics related functionalities and algorithms presented in this dissertation are related to and based on some of the fundamental concepts of computational geometry. Unlike meshes, volumetric meshes are not as well understood and researched. The presented

volumetric mesh datastructure and its wide range of utility functions, transformation operations and algorithms can be adapted for computational geometry applications to start building a body of research for volumetric meshes. While the sophistication or in-depth investigation of each of the computational geometry topics are not within the scope of this dissertation, the implementation of those concepts in the context of 3D graphic statics can nevertheless communicate the main ideas and principles used in a widely-used programming language that is legible and understandable by researchers from computational geometry. Consequently, **`compas_3gs`** and its online documentation are intended to be the start of a closer dialogue between graphic statics and computational geometry communities. Researchers from computational geometry can not only use, test and implement the contents of **`compas_3gs`**, but also make contributions of their own to improve the computational base for the framework.

10.3 Limitations and future work

This dissertation presented a computational design framework for 3D graphic statics and new applications that are now possible using the various components of the framework. While the benefits and advantages of the presented method have been highlighted, there are also limitations that require further research and investigation. They are summarised as follows.

1. **Constraining cell networks**

   Cell networks allow force-driven design explorations that are freed from any global geometric and topological constraints. The topology of the individual cells can change independently from one another. However, the explicit dual and reciprocal relationship is lost. Because there is no longer an explicit geometric constraint that governs the entire force diagram, sufficient constraints need to be imposed manually by the user to obtain stable solutions. While these constraints can be used to explore different equilibrium solutions, it may not be obvious to first-time users what reasonable domains are for various parameters and constraints. In future investigations using cell networks, a better means of informing and guiding the user regarding the control and enforcement of constraints will help address solutions that are over- or under-constrained.

2. **Guided generative design**

   The new generative design workflow presented in Chapter 7 allows users to sculpt equilibrium structures in space with a significant amount of control and design freedom. The examples presented in this dissertation applied arbitrary sequences and combinations of polyhedral transformations, as the primary objective was to showcase

how a wide range of new topologies can be generated and explored in an unbiased manner. In future research, the sequencing of different polyhedral transformations can be investigated, to determine which combinations of transformations tend to result in more efficient and better-performing structures.

3. **Identifying meaningful structures to investigate**

In future research, it will be important to identify structural typologies where the use of 3D graphic statics would be meaningful, and where it would not. One of the unique features of 3D graphic statics is the form-finding and exploration of spatial structures with multiple layers (i.e. cellular structures, spatial trusses). For example, a compression-only branching structure is an efficient typology where 3D graphic statics design approach naturally applies and is advantageous over other known approaches. On the contrary, a complex spatial structure with a large number of compression and tension members may be generated using 3D graphic statics, but may have limited merit in practice due to difficulties related to assembly of the structure and the maintenance of its highly indeterminate behaviour. By narrowing down the types of spatial structures where the application of 3D graphic statics design methods has clear merit and advantages over other known approaches, the efforts of future researchers can be more focused and impactful.

4. **Assembly and construction**

Compression-only spatial structure have an invaluable benefit in that it can drastically reduce the internal stresses. As a result, the overall amount of material used can be reduced, and unconventional and organic materials can be used structurally at architectural scales. As with any designs generated using graphic statics methods, the structure is a pin-jointed truss that is only in equilibrium once the entire geometry is complete and a specific load case has been applied. Each node is essentially a mechanism in space, which requires perfect alignments of the members carrying specific magnitudes of internal forces. This means that at any point during the assembly or construction process, the structure is most likely to be unstable. Consequently, the construction of spatial structures would require a complicated organisation and sequencing of scaffolding to support the structure until the last, "key stone" piece is inserted. Spatial structures that have multiple layers of elements on top of each other, would require an even more complicated assembly procedure. Structures with both compression and tension elements would also require close monitoring of the internal stresses in the tensile members, as a slight imbalance of force distribution could cause the entire structure to become unstable. Additionally, geometric tolerance during the

construction is an important factor that could drastically influence the overall behaviour of the structure. In future research, it will be important to address these construction related issues to demonstrate that novel structures generated with computational graphic statics can also be built in a feasible manner.

5. **Display and control mechanisms**

   Despite the presented techniques and methods to enhance the visualisation of polyhedral geometries, the 2D monitors of modern computers still remain as major obstacles. Regardless of how well a graphical user interface is designed and implemented, the human designer still interacts with the computational environment with a mouse, which is restricted to 2D movements on the plane of the monitor. Innovation in the medium of visualisation, and incorporation of technologies like virtual reality (VR) and augmented reality (AR) technologies, are future areas of research that could drastically enhance and revolutionise how humans can interact with spatial computational objects.

6. **Self weight**

   One of the the fundamental limitations of graphic statics is that the external forces must be applied at the extremities of the structure. This means that the self-weight of the structure cannot be taken into account. While the negligence of self-weight can be permissible for small structures and pavilions, any larger scale applications would need to take into account the self-weight of the structure. Although this dissertation has demonstrated how point loads can be applied anywhere within the structure using cell networks, a sub-processes for computing the self-weight of each member at every step would be needed.

10.4 Final reflections

At the 2014 International Association for Shell and Spatial Structures (IASS) Symposium in Brasilia, there were only five presentations in the Working Group 15, the session dedicated to graphic statics. During the summer of 2015, a few months before the start of this research, the number of presentations at the Graphic Statics session of the IASS Symposium in Amsterdam grew to eighteen. The overwhelming growth of interest in the field of graphic statics resulted in there being two separate sessions for graphic statics for the first time in history of the IASS. By the 2016 IASS Symposium in Tokyo, there were only six presentations during the graphic statics session. However, all of them were based on 3D applications of graphic statics.

The presented research and the development of the computational framework were motivated and inspired by this rapid emergence of 3D graphic statics. Its invaluable benefits in providing an interactive and intuitive means of designing spatial structures is unprecedented, with much more opportunities yet to be discovered. **`compas_3gs`** is intended to be a common platform and computational language through which future researchers can conduct their own research and experiments, communicate with one another, exchange their latest findings and collaborate on joint research projects. It is not meant to be an end result or a finalised computational package. Rather, it is meant to be the starting point and the initial basic kit of parts for 3D graphic statics, which works out of the box for users of all experience levels and background. The ultimate hope of this research is to witness the gradual increase of contributions to **`compas_3gs`** from various individuals and institutions over time, which will help expand, diversify and enrich the field of 3D graphic statics well after the completion of this dissertation.

References

Adriaenssens, S., S. Devoldere, L. Ney, and I. Strauven (2010). *Laurent Ney - Shaping Forces*. A+Editions and Bozar Books.

Akbarzadeh, M. (2016). *3D Graphical Statics using 3D Reciprocal Diagrams*. Ph. D. thesis, ETH Zürich, Zürich, Switzerland.

Akbarzadeh, M., J. Lee, T. Van Mele, and P. Block (2017). 3D graphic statics: combined compression and tension equilibrium. unpublished journal paper.

Akbarzadeh, M., M. Mahnia, R. Taherian, and A. H. Tabrizi (2017). Prefab concrete polyhedral frame: Materializing 3D graphic statics. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*.

Akbarzadeh, M., T. Van Mele, and P. Block (2014). Compression-only form finding through finite subdivision of the external force polygon. In *Proceedings of the 2014 International Association for Shell and Spatial Structures (IASS) Symposium*, Brasilia, Brazil.

Akbarzadeh, M., T. Van Mele, and P. Block (2015a). 3D graphic statics: Geometric construction of global equilibrium. In *Proceedings of the 2015 International Association for Shell and Spatial Structures (IASS) Symposium*, Amsterdam, The Netherlands.

Akbarzadeh, M., T. Van Mele, and P. Block (2015b). On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams. *Computer-Aided Design 63*, 118–128.

Akbarzadeh, M., T. Van Mele, and P. Block (2015c). Three-dimensional compression form finding through subdivision. In *Proceedings of the 2015 International Association for Shell and Spatial Structures (IASS) Symposium*, Amsterdam, The Netherlands.

Akbarzadeh, M., T. Van Mele, and P. Block (2016). Three-dimensional graphic statics: Initial explorations with polyhedral form and force diagrams. *International Journal of Space Structures 31*(2-4), 217–226.

Alexandrov, A. (2005). *Convex Polyhedra* (1st ed.). Springer Monographs in Mathematics. Springer-Verlag Berlin Heidelberg.

Alic, V. and K. Persson (2018). Generating initial reinforcement layouts using graphic statics. In *Proceedings of the 2018 International Association for Shell and Spatial Structures (IASS) Symposium*, Boston, USA.

Alic, V. and D. Åkesson (2017). Bi-directional algebraic graphic statics. *Computer-Aided Design 93*, 26 – 37.

Allen, E. and W. Zalewski (2009, September). *Form and Forces: Designing Efficient and Expressive Structures*. New York: Wiley.

Alling, N. and N. Greenleaf (1969). Klein surfaces and real algebraic function fields. *Bulletin of the American Mathematical Society 75*, 627–888.

Alumbaugh, T. J. and X. Jiao (2005). Compact array-based mesh data structures. In B. W. Hanks (Ed.), *Proceedings of the 14th International Meshing Roundtable*, pp. 485–503. Springer Berlin Heidelberg.

Athanasopoulos, G. S., M. Akbarzadeh, and A. McRobie (2018). Graphic statics: Constrained form finding for parallel system of forces using Corsican sum. In *Proceedings of the 2018 International Association for Shell and Spatial Structures (IASS) Symposium*, Boston, USA.

Athanasopoulos, G. S. and A. McRobie (2017). Graphic statics applied on gridshell roofs. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg, Germany.

Autodesk Inc. (2018). Maya. Computer software. https://www.autodesk.eu/products/maya/.

Bærentzen, J., M. Misztal, and K. Wełnicka (2012). Converting skeletal structures to quad dominant meshes. *Computers Graphics 36*(5), 555 – 561.

Baker, W. F. (1992). Energy-based design of lateral systems. *Structural Engineering International 2*(2), 99–102.

Baker, W. F., L. L. Beghini, A. Mazurek, J. Carrion, and A. Beghini (2013). Maxwell's reciprocal diagrams and discrete Michell frames. *Structural and Multidisciplinary Optimization 48*(2), 267–277.

Balling, R. (1999). Design by shopping: A new paradigm. In *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization*, Buffalo, New York, pp. 295–297.

Barnes, M. (1977). *Form-finding and analysis of tension space structures by dynamic relaxation*. Ph. D. thesis, City University, London, United Kingdom.

Baumgart, B. G. (1975). A polyhedron representation for computer vision. In *Proceedings of the May 19-22, 1975, National Computer Conference and Exposition*, AFIPS '75, New York, NY, USA, pp. 589–596. ACM.

Beghini, L. L., J. Carrion, A. Beghini, A. Mazurek, and W. F. Baker (2013). Structural optimization using graphic statics. *Structural and Multidisciplinary Optimization 49*(3), 351–366.

Bentley, J. L. and T. A. Ottmann (1979). Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers C-28*(9), 643–647.

Block, P. (2005). Equilibrium systems: Studies in masonry structure. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Block, P., T. Ciblac, and J. Ochsendorf (2006). Real-time limit analysis of vaulted masonry buildings. *Computers and Structures 84*(29), 1841–1852.

Block, P. and J. Ochsendorf (2007). Thrust network analysis: A new methodology for three-dimensional equilibrium. *Journal of the International Association for Shell and Spatial Structures 48*(3), 167–173.

Block Research Group (2010). eQUILIBRIUM. http://block.arch.ethz.ch/equilibrium.

Bolhassani, M., M. Akbarzadeh, M. Mahnia, and R. Taherian (2018). On structural behavior of a funicular concrete polyhedral frame designed by 3d graphic statics. *Structures 14*, 56 – 68.

Bolhassani, M., A. Tabatabaei Ghomi, A. Nejur, and M. Akbarzadeh (2018). Structural behavior of a cast-in-place funicular polyhedral concrete: Applied 3d graphic statics. In *Proceedings of the 2018 International Association for Shell and Spatial Structures (IASS) Symposium*, Boston, USA.

Bouaziz, S., M. Deuss, Y. Schwartzburg, T. Weise, and M. Pauly (2012). Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum 31*(5), 1657–1667.

Bow, R. H. (1873). *Economics of construction in relation to framed structures*. London: Spon.

Boy, W. (1903). Über die curvatura integra und die topologie geschlossener flächen. (mit 24 figuren im text). *Mathematische Annalen 57*, 151–184.

Cagan, J. and W. J. Mitchell (1993). Optimally directed shape generation by shape annealing. *Environment and Planning B: Planning and Design 20*(1), 5–12.

Carlile, M. J. (1995). *The Success of the Hypha and Mycelium*, pp. 3–19. Dordrecht: Springer Netherlands.

Cleveland, W. S. and R. McGill (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association 79*(387), 531–554.

Computer Graphics Group Aachen (2016, November). Concepts of openvolumemesh. http://www.openvolumemesh.org/Documentation/OpenVolumeMesh-Doc-Latest/concepts.html.

Conway, J., H. Burgiel, and C. Goodman-Strauss (2008). *The Symmetries of Things*. Ak Peters Series. Taylor & Francis.

Cox, H., R. Bisplinghoff, and W. Hemp (1965). *The Design of Structures of Least Weight*. International series of monographs on aeronautics and astronautics: Solid and structural mechanics. Pergamon Press.

Coxeter, H. (1973). *Regular Polytopes*. Dover books on advanced mathematics. Dover Publications.

Crapo, H. (1979). Structural rigidity. *Structural Topology* (1), 26–45, 73.

Crapo, H. and W. Whiteley (1993). Plane self stresses and projected polyhedra I: The basic pattern. *Structural Topology 20*, 55 – 78.

Crapo, H. and W. Whiteley (1994). Spaces of stresses, projections and parallel drawings for spherical polyhedra. *Beiträge zur Algebra und Geometrie 35*(2), 259–281.

Cremona, L. (1872). *Le figure reciproche nella statica grafica*. Milano: Tipografia di G. Bernardoni.

Cremona, L. (1890). *Graphical Statics: Two Treatises on the Graphical Calculus and Reciprocal Figures in Graphical Statics*. Oxford: Clarendon Press.

Cuillière, J.-C., V. François, and A. Nana (2018). Automatic construction of structural CAD models from 3d topology optimization. *Computer-Aided Design and Applications 15*(1), 107–121.

Culmann, K. (1864). *Die Graphische Statik* (1st ed.). Zürich: Verlag Meyer und Zeller.

Culmann, K. (1875). *Die Graphische Statik* (2nd ed.). Zürich: Verlag Meyer und Zeller.

Damiand, G. (2018). Linear cell complex. In *CGAL User and Reference Manual* (4.12 ed.). CGAL Editorial Board.

De Floriani, L. and A. Hui (2003). A scalable data structure for three-dimensional non-manifold objects. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, Aire-la-Ville, Switzerland, Switzerland, pp. 72–82. Eurographics Association.

De Floriani, L. and A. Hui (2005). Data structures for simplicial complexes: an analysis and a comparison. In *Proceedings of the third Eurographics symposium on Geometry processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association: Eurographics Association.

De Floriani, L. and A. Hui (2007). Shape representations based on simplicial and cell complexes. In D. Schmalstieg and J. Bittner (Eds.), *Eurographics 2007 - State of the Art Reports*. The Eurographics Association.

Delestrac, D. (2014). Sand wars. Documentary.

Demaine, E. D. and J. O'Rourke (2007). *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*, Chapter 23, pp. 339–357. Cambridge University Press.

Deng, B., S. Bouaziz, M. Deuss, A. Kaspar, Y. Schwartzburg, and M. Pauly (2015). Interactive design exploration for constrained meshes. *Computer-Aided Design 61*, 13 – 23.

Deuss, M., A. H. Deleuran, S. Bouaziz, B. Deng, D. Piker, and M. Pauly (2015). Shapeop—a robust and extensible geometric modelling paradigm. In R. M. Thomsen, M. Tamke, C. Gengnagel, B. Faircloth, and F. Scheurer (Eds.), *Modelling Behaviour: Design Modelling Symposium 2015*, pp. 505–515. Cham: Springer International Publishing.

D'Acunto, P., J.-P. Jasienski, P. O. Ohlbrock, and C. Fivet (2017). Vector-based 3d graphic statics: Transformations of force diagrams. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg, Germany.

Eigensatz, M., M. Kilian, A. Schiftner, N. J. Mitra, H. Pottmann, and M. Pauly (2010, July). Paneling architectural freeform surfaces. *ACM Trans. Graph. 29*(4), 45:1–45:10.

Ellen MacArthur Foundation (2012, January). Towards the circular economy 1: Economic and business rationale for an accelerated transition.

Ewert, S. (2002). *Brückensysteme*. Berlin: Ernst & Sohn.

Föppl, A. O. (1892). *Das Fachwerk im Raume. Leipzig Verlag von B.G.* Teubner.

Frondel, M., P. Grösche, D. Huchtermann, A. Oberheitmann, J. Peters, G. Angerer, C. Sartorius, P. Buchholz, S. Röhling, and M. Wagner (2007). *Trends der Angebots- und Nachfragesituation bei mineralischen Rohstoffen: Endbericht. Forschungsprojekt Nr. 09/05 des Bundesministeriums für Wirtschaft und Technologie (BMWi)*. RWI - Leibniz-Institut für Wirtschaftsforschung.

Fuller, B. (1975). *Synergetics : explorations in the geometry of thinking*. New York: Macmillan.

Garimella, R. V. (2002). Mesh data structure selection for mesh generation and FEA applications. *International Journal for Numerical Methods in Engineering 55*(4), 451–478.

Garner, C. W. L. (1967). Regular skew polyhedra in hyperbolic three-space. *Canadian Journal of Mathematics 19*, 1179–1186.

Ghomi, A. T., M. Bolhassan, A. Nejur, and M. Akbarzadeh (2018). Effect of subdivision of force diagrams on the local buckling, load-path and material use of founded forms. In *Proceedings of the 2018 International Association for Shell and Spatial Structures (IASS) Symposium*, Boston, USA.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research 57*(1), 345–420.

Google LLC (2018). Google Scholar. https://scholar.google.com/.

Greenwold, S. and E. Allen (2003). Active Statics. http://acg.media.mit.edu/people/simong/statics.

Gritzmann, P. and A. Hufnagel (1999). On the algorithmic complexity of Minkowski's reconstruction theorem. *Journal of the London Mathematical Society 59*, 1081–1100.

Grünbaum, B. (2003). *Convex polytopes* (2nd ed.). Graduate texts in mathematics. New York, Berlin, London: Springer.

Grünbaum, B. (2007). Graphs of polyhedra; polyhedra as graphs. *Discrete Mathematics 445-463*(307), 3–5.

Hablicsek, M., M. Akbarzadeh, and Y. Guo (2019). Algebraic 3d graphic statics: Reciprocal constructions. *Computer-Aided Design 108*, 30 – 41.

Hansen, S. R. and G. N. Vanderplaats (1990). Approximation method for configuration optimization of trusses. *AIAA Journal 28*(1), 161–168.

Harman, P. M. (2001). *The Natural Philosophy of James Clerk Maxwell*. Cambridge University Press.

Hartz, C., A. Mazurek, M. Miki, T. Zegard, T. Mitchell, and W. F. Baker (2017). The application of 2d and 3d graphic statics in design. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg, Germany.

Hebel, D., P. Block, F. Heisel, and T. Méndez Echenagucia (2017). Beyond mining - urban growth: The architectural innovation of cultivated resources through appropriate engineering. In J. S. A. Alejandro Zaera-Polo (Ed.), *Imminent Commons: The Expanded City*. Actar Publishers. Seoul Biennale of Architecture and Urbanism 2017 - Exhibition Catalogue.

Hebel, D., A. Javadian, F. Heisel, K. Schlesier, D. Griebel, and M. Wielopolski (2014). Process-controlled optimization of the tensile strength of bamboo fiber composites for structural applications. *67*(December 2014), 125 – 131.

Hebel, D., M. Wisniewska, and F. Heisel (2014). *Building from Waste: Recovered Materials in Architecture and Construction*. Walter de Gruyter GmbH.

Hebel, D. E., M. Moges, and Z. Gray (2015). *SUDU – Research and Manual (2 Volumes)*. Berlin, Germany: Ruby Press.

Heisel, F. (2015). Waste vault - the eth zürich pavilion at the ideas city festival in new york city. *FCL Magazine Special Issue Constructing Alternatives - Research Projects 2012-2015 - Assistant Professorship Dirk E. Hebel*, 68–77.

Heisel, F., K. Schlesier, J. Lee, M. Rippmann, N. Saeidi, A. Javadian, P. Block, and D. Hebel (2018). Design, cultivation and application of load-bearing mycelium components. *International Journal of Sustainable Energy Development 6*(1), 296–303.

Heisel, F., K. Schlesier, J. Lee, M. Rippmann, N. Saeidi, A. Javadian, D. E. Hebel, and P. Block (2017). Design of a load-bearing mycelium structure through informed structural engineering. In *World Congress on Sustainable Technologies (WCST 2017)*. Taylor Francis.

Hohenwarter, M., M. Borcherds, G. Ancsin, B. Bencze, M. Blossier, A. Delobelle, C. Denizet, J. Éliás, A. Fekete, L. Gál, Z. Konečný, Z. Kovács, S. Lizelfelner, B. Parisse, and G. Sturr (2002). GeoGebra. http://www.geogebra.org.

Horn, B. K. P. (1984). Extended Gaussian images. *Proceedings of the IEEE 72*(2), 1671–1686.

Ikeuchi, K. (1981). Recognition of 3-d objects using the extended gaussian image. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, BC, Canada, pp. 595–600.

Jasienski, J.-P., P. D'acunto, P. O. Ohlbrock, and C. Fivet (2016). Vector-based 3d graphic statics (part ii): Construction of force diagrams. In *Proceedings of the 2016 International Association for Shell and Spatial Structures (IASS) Symposium*, Tokyo, Japan.

Jessen, B. (1967). Orthogonal icosahedron. *Nordisk Matematisk Tidsskrift* (15), 90–96.

Jiang, C., C. Tang, J. Wang, J. Wallner, and H. Pottmann (2015). Freeform honeycomb structures and lobel frames. In *ACM SIGGRAPH 2015 Posters*, SIGGRAPH '15, New York, NY, USA, pp. 75:1–75:1. ACM.

Kántor, S. (2003). On the volume of unbounded polyhedra in the hyperbolic space. *Beiträge zur Algebra und Geometrie 44*(1), 145–154.

Kern, W. F. and J. R. Bland (1948). Prism. In *Solid Mensuration with Proof* (2nd ed.)., Chapter 13, pp. 28–32. New York: Wiley and Sons.

Kiefer, J. (1953). Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society 4*(3), 502–506.

Kilian, A. and J. Ochsendorf (2005). Particle-spring systems for structural form finding. *Journal of the International Association for Shell and Spatial Structures 46*(147), 77–84.

Koning, H. and J. Eizenberg (1981). The language of the Prairie: Frank Lloyd Wright's Prairie Houses. *Environment and Planning B: Planning and Design 8*(3), 295–323.

Konstantatou, M., P. D'Acunto, and A. McRobie (2018). Polarities in structural analysis and design: n-dimensional graphic statics and structural transformations. *International Journal of Solids and Structures*.

Konstantatou, M. and A. McRobie (2016). Reciprocal constructions using conic sections poncelet duality. In *Proceedings of the 2016 International Association for Shell and Spatial Structures (IASS) Symposium*, Tokyo, Japan.

Konstantatou, M. and A. Mcrobie (2018). Graphic statics for optimal trusses geometry-based structural optimisation. In *Proceedings of the 2018 International Association for Shell and Spatial Structures (IASS) Symposium*, Boston, USA.

Kremer, M., D. Bommes, and L. Kobbelt (2013). OpenVolumeMesh – a versatile index-based data structure for 3D polytopal complexes. In X. Jiao and J.-C. Weill (Eds.), *Proceedings of the 21st International Meshing Roundtable*, pp. 531–548. Springer Berlin Heidelberg.

Kurrer, K.-E. (2008). *The history of the theory of structures : from arch analysis to computational mechanics*. Berlin: Ernst Sohn.

Lachand-Robert, T. and É. Oudet (2005). Minimizing within convex bodies using a convex hull method. *SIAM Journal on Optimization 16*(2), 368–379.

Lachauer, L. (2015). *Interactive Equilibrium Modelling - A new approach to the computer-aided exploration of structures in architecture*. Ph. D. thesis, ETH Zürich.

Lachauer, L. and P. Block (2014). Interactive equilibrium modelling. *International Journal of Space Structures 29*(1), 25–37.

Lachauer, L. and T. Kotnik (2010). Geometry of structural form. In C. Ceccato, L. Hesselgren, M. Pauly, H. Pottmann, and J. Wallner (Eds.), *Advances in Architectural Geometry 2010*, pp. 193–203. Vienna: Springer.

Lakatos, I., J. Worrall, and E. Zahar (1976). *Proofs and Refutations: The Logic of Mathematical Discovery*. Cambridge University Press.

Lamé, G. and E. Clapeyron (1828). Mémoire sur la construction des polygones funiculaires. *Journal du génie civil I*, 496–504.

Lee, J., C. Mueller, and C. Fivet (2016). Automatic generation of diverse equilibrium structures through shape grammars and graphic statics. *International Journal of Space Structures 31*(2-4), 147–164.

Lee, J., T. Van Mele, and P. Block (2016). Form-finding explorations through geometric transformations and modifications of force polyhedrons. In *Proceedings of the 2016 International Association for Shell and Spatial Structures (IASS) Symposium*, Tokyo, Japan.

Lee, J., T. Van Mele, and P. Block (2017). Construction of polyhedra with target force magnitudes. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg.

Lee, J., T. Van Mele, and P. Block (2018). Disjointed force polyhedra. *Computer-Aided Design 99*, 11 – 28.

Lelivelt, R., G. Lindner, P. Teuffel, and H. Lamers (2015). The production process and compressive strength of mycelium-based materials. In *First International Conference on Bio-based Building Materials. 22-25 June 2015, Clermont-Ferrand, France*, pp. 1–6.

Linkwitz, K. and H. J. Schek (1971). Einige bemerkungen zur berechnung von vorgespannten seilnetzkonstruktionen. *Ingenieur-Archiv 40*(3), 145–158.

Little, J. J. (1983). An iterative method for reconstructing convex polyhedra from external guassian images. In *Proceedings of the 1983 Association for the Advancement of Artificial Intelligence (AAAI)*, pp. 247–250.

Liu, B. (2015). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Liu, Y., H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang (2006, July). Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25*(3), 681–689.

Lyusternik, L. A. (1963). *Convex Figures and Polyhedra*. New York: Dover Publications.

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions On Graphics (Tog) 5*(2), 110–141.

Maxwell, J. C. (1864). On reciprocal figures and diagrams of forces. *Philosophical Magazine Series 4 27*(182), 250–261.

Maxwell, J. C. (1870). On reciprocal figures, frames and diagrams of forces. *Transactions of the Royal Society of Edinburgh 26*(1), 1–40.

Mazurek, A., W. F. Baker, and C. Tort (2011). Geometrical aspects of optimum truss like structures. *Structural and Multidisciplinary Optimization 43*(2), 231–242.

McDonough, W. and M. Braungart (2002). *Cradle to Cradle: Remaking the Way We Make Things*. New York, USA: North Point Press.

McRobie, A. (2016a). Maxwell and Rankine reciprocal diagrams via Minkowski sums for two-dimensional and three-dimensional trusses under load. *International Journal of Space Structures 31*(2-4), 203–216.

McRobie, A. (2016b). Rankine reciprocals with zero bars. https://www.researchgate.net/publication/315027343_Rankine_Reciprocals_with_Zero_Bars.

McRobie, A. (2017a). The geometry of structural equilibrium. *Royal Society Open Science 4*(3).

McRobie, A. (2017b). Graphic analysis of 3d frames: Clifford algebra and Rankine incompleteness. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg, Germany.

McRobie, A., W. Baker, T. Mitchell, and M. Konstantatou (2016). Mechanisms and states of self-stress of planar trusses using graphic statics, part ii: Applications and extensions. *International Journal of Space Structures 31*(2-4), 102–111.

McRobie, A., M. Konstantatou, G. Athanasopoulos, and L. Hannigan (2017). Graphic kinematics, visual virtual work and elastographics. *Royal Society Open Science 4*(5).

McRobie, A. and C. Williams (2018). Discontinuous maxwell–rankine stress functions for space frames. *International Journal of Space Structures 33*(1), 35–47.

Menges, A. (2012). *Material Computation: Higher Integration in Morphogenetic Design*. Architectural Design March/April 2012, Profile No 216. Wiley.

Miki, M., T. Igarashi, and P. Block (2015). Direct application of airy stress functions to nurbs patches for computing compression shells. In *Proceedings of the 2015 International Association for Shell and Spatial Structures (IASS) Symposium*, Amsterdam, The Netherlands.

Milliman, J. and J. Syvitski (1992). Geomorphic/tectonic control of sediment discharge to the ocean: The importance of small mountainous rivers. *The Journal of Geology*, 525–44.

Minkowski, H. (1897). Allgemeine lehrsätze über die convexen polyeder. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 198–219.

Mitchell, T., W. Baker, A. McRobie, and A. Mazurek (2016). Mechanisms and states of self-stress of planar trusses using graphic statics, part i: The fundamental theorem of linear algebra and the airy stress function. *International Journal of Space Structures 31*(2-4), 85–101.

Mitchell, W. (1991). Functional grammars: An introduction. In *Proceedings of the 1991 ACADIA Conference*, Troy, New York, pp. 167–176.

Moni, S. (1990, June). A closed-form solution for the reconstruction of a convex polyhedron from its extended gaussian image. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, Volume 1, pp. 223–226.

Mueller, C. (2014). *Computational Exploration of the Structural Design Space*. Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Muttoni, A., J. Schwartz, and B. Thürlimann (1997). *Design of Concrete Structures with Stress Fields*. Birkhäuser Basel.

Nejur, A. and M. Akbarzadeh (2018). Constrained manipulation of polyhedral systems. In *Proceedings of the 2018 International Association for Shell and Spatial Structures (IASS) Symposium*, Boston, USA.

Nielsen, T. H., M. Akbarzadeh, and P. Goltermann (2017). Addressing buckling of compression members using subdivision of force diagrams. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg, Germany.

O'Rourke, J. (2010). Flat zipper-unfolding pairs for platonic solids. *CoRR abs/1010.2450*. http://arxiv.org/abs/1010.2450.

O'Rourke, J. (2011). Convex polyhedra realizing given face areas. *CoRR abs/1101.0823*. http://arxiv.org/abs/1101.0823.

Oxman, N. (2010). *Material-based design computation*. Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Peduzzi, P. (2014). Sand, rarer than one thinks. *Environmental Development 11*, 208–18.

Pelletier, M., G. Holt, J. Wanjura, A. Lara, A. Tapia-Carillo, G. McIntyre, and E. Bayer (2017). Research paper. *Industrial Crops Products 95*(Complete), 342–347.

Poncelet, J. V. (1822). *Traité des propriétés projectives des figures*. Bachelier.

Poranne, R., R. Chen, and C. Gotsman (2015). On linear spaces of polyhedral meshes. *IEEE Trans. Vis. Comput. Graph. 21*(5), 652–662.

Poranne, R., E. Ovreiu, and C. Gotsman (2013). Interactive planarization and optimization of 3D meshes. *Computer Graphics Forum 32*(1), 152–163.

Pottmann, H., C. Jiang, M. Höbinger, J. Wang, P. Bompas, and J. Wallner (2015). Cell packing structures. *Computer-Aided Design 60*, 70–83. Special issue on Material Ecology.

Pottmann, H., Y. Liu, J. Wallner, A. Bobenko, and W. Wang (2007). Geometry of multi-layer freeform structures for architecture. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA. ACM.

Pottmann, H., A. Schiftner, P. Bo, H. Schmiedhofer, W. Wang, N. Baldassini, and J. Wallner (2008, August). Freeform surfaces from single curved panels. *ACM Trans. Graph. 27*(3), 76:1–76:10.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (2007). Golden section search in one dimension. In *Numerical Recipes: The Art of Scientific Computing* (3 ed.)., Chapter 10.2. New York: Cambridge University Press.

Rahami, H., A. Kaveh, and Y. Gholipour (2008). Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures 30*(9), 2360 – 2369.

Rankine, W. (1858). *A manual of applied mechanics*. London: Griffin.

Rankine, W. (1864). Principle of the equilibrium of polyhedral frames. *Philosophical Magazine Series 4 27*(180), 92.

Reddy, G. M. and J. Cagan (1995, March). Optimally directed truss topology generation using shape annealing. *Journal of Mechanical Design 117*(1), 206–209.

Reeves, D., V. Bhooshan, and S. Bhooshan (2016). Freeform developable spatial structures david. In *Proceedings of the 2016 International Association for Shell and Spatial Structures (IASS) Symposium*, Tokyo, Japan.

Reller, A. and T. Graedel (2009). How long will it last? Infographic, NewScientist.

Remacle, J.-F. and M. S. Shephard (2003). An algorithm oriented mesh database. *International Journal for Numerical Methods in Engineering 58*(2), 349–374.

Rippmann, M. (2016). *Funicular Shell Design: Geometric Approaches to Form Finding and Fabrication of Discrete Funicular Structures*. Ph. D. thesis, ETH Zürich, Zürich.

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

Roach, J. and J. Wright (1986). Spherical dual images: A 3D representation method for solid objects that combines dual space and Gaussian spheres. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, Volume 3, pp. 1087–1092.

Robert Mcneel & Associates (1993). Rhinoceros: NURBS modelling for Windows. Computer software. http://www.rhino3d.com.

Robert William Burkhardt, J. (2008, September). A practical guide to tensegrity design. Accessed: November 2, 2017.

Saha, P. K., G. Borgefors, and G. Sanniti di Baja (2016). A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters 76*(C), 3–12.

Samyn, P., P. Latteur, and W. De Wilde (1999). Design of classical trusses by the use of morphological indicators. *Unknown Journal*. ASCE - AMERICAN SOCIETY OF CIVIL ENGINEERS, JOURNAL OF STRUCTURAL ENGINEERING (manuscrit envoyé en novembre 1999, 31 pages et figures).

Schiftner, A. and J. Balzer (2010). Statics-sensitive layout of planar quadrilateral meshes. In C. Ceccato, L. Hesselgren, M. Pauly, H. Pottmann, and J. Wallner (Eds.), *Advances in Architectural Geometry 2010*, pp. 221–236. Vienna: Springer.

Schlaich, J. and K. Schäfer (1991). Design and detailing of structural concrete using strut-and-tie models. *The Structural Engineer 69*(6), 113.

Schlaich, M. and G. Anagnostou (1990). Stress fields for nodes of strut-and-tie models. *Journal of Structural Engineering 116*(1), 13–23.

Schlegel, V. (1883). *Theorie der homogen zusammengesetzten Raumgebilde*. Number nos. 14-22 in Nova acta Academiae Caesareae Leopoldino-Carolinae Germanicae Naturae Curiosorum. Für die Akademie in commission bei W. Engelmann in Leipzig.

Schmidt, D. and C. Heckendorf (2017). ngram: Fast n-gram tokenization. R package version 3.0.4.

Scholz, E. (1989). *Symmetrie Gruppe Dualität: Zur Beziehung zwischen theoretischer Mathematik und Anwendungen in Kristallographie und Baustatik des 19. Jahrhunderts*. Science Networks. Historical Studies. Birkhäuser Basel.

Schrems, M. and T. Kotnik (2013). On the extension of graphic statics into the third dimension. In P. J. da Sousa Cruz (Ed.), *Structures and Architecture: Concepts, Applications and Challenges: Proceedings of the Second International Conference on Structures and Architecture*, Guimarães, Portugal, pp. 1736–1742. CRC Press.

Scrivener, K. (2015). Lecture at the chair of Marc Angelil, ETH Zürich.

Shea, K. (1997). *Essays of discrete structures: Purposeful design of grammatical structures by directed stochastic search*. Ph. D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

Shea, K. and J. Cagan (1999). Languages and semantics of grammatical discrete structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 13*(4), 241–251.

Shea, K., J. Cagan, and S. J. Fenves (1997). A shape annealing approach to optimal truss design with dynamic grouping of members. *Accepted Manuscripts Journal of Mechanical Design 119*(3), 388–394.

Shobeiri, V. and B. Ahmadi-Nedushan (2017). Bi-directional evolutionary structural optimization for strut-and-tie modelling of three-dimensional structural concrete. *Engineering Optimization 49*(12), 2055–2078.

Sobek, W. and H. Trumpf (2008, March). Sustainable tall buildings – some introductory remarks. In *CTBUH 8th World Congress*, Dubai, UAE.

Sondericker, J. (1903). *Graphic Statics with Applications to Trusses, Beams and Arches*. J.Wiley & Sons.

SRF TV (2014). Eco spezial: Sand – ein milliardengeschäft. TV Documentary.

Srinivasan, V., E. Mandal, and E. Akleman (2005). Solidifying wireframes. In R. Sarhangi and R. V. Moody (Eds.), *Renaissance Banff: Mathematics, Music, Art, Culture*, Southwestern College, Winfield, Kansas, pp. 203–210. Bridges Conference.

Stahel, W. (1982). Product-life factor. Technical report, Houston Area Research Center (HARC), The Woodlands, TX, USA.

Stamets, P. (2011). *Growing Gourmet and Medicinal Mushrooms*. Gardening/Mycology. Ten Speed Press.

Stevin, S. (1586). *De Beghinselen des Waterwichts*. Raphelengius, Franciscus.

Stiny, G. (2008). *Shape: Talking About Seeing and Doing*. The MIT Press.

Stiny, G. and J. Gips (1972). Shape grammars and the generative specification of painting and sculpture. In *International Federation for Information Processing Congress 71*, pp. 1460–1465.

Stiny, G. and W. J. Mitchell (1978). The Palladian grammar. *Environment and Planning B: Planning and Design 5*(1), 5–18.

Tang, C., X. Sun, A. Gomes, J. Wallner, and H. Pottmann (2014, July). Form-finding with polyhedral meshes made simple. *ACM Transactions on Graphics 33*(4), 70:1–70:9.

The CGAL Project (2018). *CGAL User and Reference Manual* (4.12 ed.). CGAL Editorial Board.

Travaglini, S., J. Noble, P. Ross, and C. Dharan (2013). 'mycology matrix composites. In *Proceedings of the American Society for Composites – 28th Technical Conference*, State College, PA, USA.

Vaitkus, M. and T. Várady (2015). A general framework for constrained mesh parameterization. In *Proceedings of the 31st Spring Conference on Computer Graphics*, SCCG '15, New York, NY, USA, pp. 15–21. ACM.

Van Mele, T. and P. Block (2014). Algebraic graph statics. *Computer-Aided Design 53*, 104–116.

Van Mele, T., L. Lachauer, M. Rippmann, , and P. Block (2012). Geometry-based understanding of structures. *Journal of the International Association for Shell and Spatial Structures 53*(174), 285–295.

Van Mele, T., A. Liew, T. M. Echenagucia, M. Rippmann, et al. (2017). compas: A framework for computational research in architecture and structures. http://compas-dev.github.io/compas/.

Van Steirteghem, J., P. Samyn, and W. P. De Wilde (2002). The use of morphological indicators in the optimisation of structures. In C. Brebbia and W. P. De Wilde (Eds.), *High Performance Structures and Composites*, Ashurst Lodge, Southampton, UK, pp. 431–441. WIT Press.

Varignon, P. (1725). *Nouvelle mécanique ou statique: ouvrage posthume*. Paris: C. Jombert.

Vouga, E., M. Höbinger, J. Wallner, and H. Pottmann (2012). Design of self-supporting surfaces. *ACM Trans. Graph. 31*(4), 87:1–87:11.

Weiler, K. J. (1988). Radial edge structure: A topological representation for nonmanifold geometric boundary modeling. In M. J. Wozny (Ed.), *Geometric Modeling for CAD Applications*, Rensselaerville, NY, USA, pp. 3–36.

Wenninger, M. J. (2012). *Spherical Models*. Dover Publications.

Wolfe, W. S. (1921). *Graphical Analysis: A Text Book on Graphic Statics*. New York: McGraw-Hill Book Co. Inc.

WU Global Material Flows Database (2015). Global material extraction by material category, 1980-2013.

Xu, Z. and M. Suk (1995). Representation and reconstruction of polygons and polyhedra using hierarchical extended gaussian images. *Annals of Mathematics and Artificial Intelligence 13*(3), 377–399.

Zadravec, M., A. Schiftner, and J. Wallner (2010). Designing quad-dominant meshes with planar faces. *Computer Graphics Forum 29*(5), 1671–1679.

Zalewski, W. and E. Allen (1998, February). *Shaping Structures: Statics*. New York: Wiley.

Zanni, G. and G. R. Pennock (2009). A unified graphical approach to the static analysis of axially loaded structures. *Mechanism and Machine Theory 44*(12), 2187 – 2203.

Zhang, L., S. Wang, and B. Liu (2018). Deep learning for sentiment analysis: A survey. *CoRR abs/1801.07883*. http://arxiv.org/abs/1801.07883.

List of Figures

Relevant publications by the author

Lee, J., C. Mueller, and C. Fivet (2016). Automatic generation of diverse equilibrium structures through shape grammars and graphic statics. *International Journal of Space Structures 31(2-4)*, 147–164.

Lee, J., T. Van Mele, and P. Block (2016). Form-finding explorations through geometric transformations and modifications of force polyhedrons. In *Proceedings of the 2016 International Association for Shell and Spatial Structures (IASS) Symposium*, Tokyo, Japan.

Lee, J., T. Van Mele, and P. Block (2017). Construction of polyhedra with target force magnitudes. In *Proceedings of the 2017 International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg, Germany.

Heisel, F., K. Schlesier, J. Lee, M. Rippmann, N. Saeidi, A. Javadian, D. E. Hebel, and P. Block (2017). Design of a load-bearing mycelium structure through informed structural engineering. In *World Congress on Sustainable Technologies (WCST 2017)*. Taylor Francis.

Lee, J., T. Van Mele, and P. Block (2018). Disjointed force polyhedra. *Computer-Aided Design 99*, 11 – 28.

Heisel, F., K. Schlesier, J. Lee, M. Rippmann, N. Saeidi, A. Javadian, P. Block, and D. Hebel (2018). Design, cultivation and application of load-bearing mycelium components. *International Journal of Sustainable Energy Development 6(1)*, 296–303.

About the author

Juney Lee | 이준 | 李儁

www.juney-lee.com

Education

2015 - 2018 **Doctor of Sciences**
ETH Zurich, Switzerland

2014 - 2015 **Master of Engineering in Civil and Environmental Engineering**
Massachusetts Institute of Technology, USA

2010 - 2014 **Master of Architecture**
Massachusetts Institute of Technology, USA

2003 - 2007 **Bachelor of Arts in Architecture**
University of California at Berkeley, USA

Minor in Structural Engineering
University of California at Berkeley, USA

Experience

2015 - **Block Research Group**
Zurich, Switzerland

2014 **CBT Architects**
Boston, MA, USA

2013 **TriPyramid Structures**
Westford, MA, USA

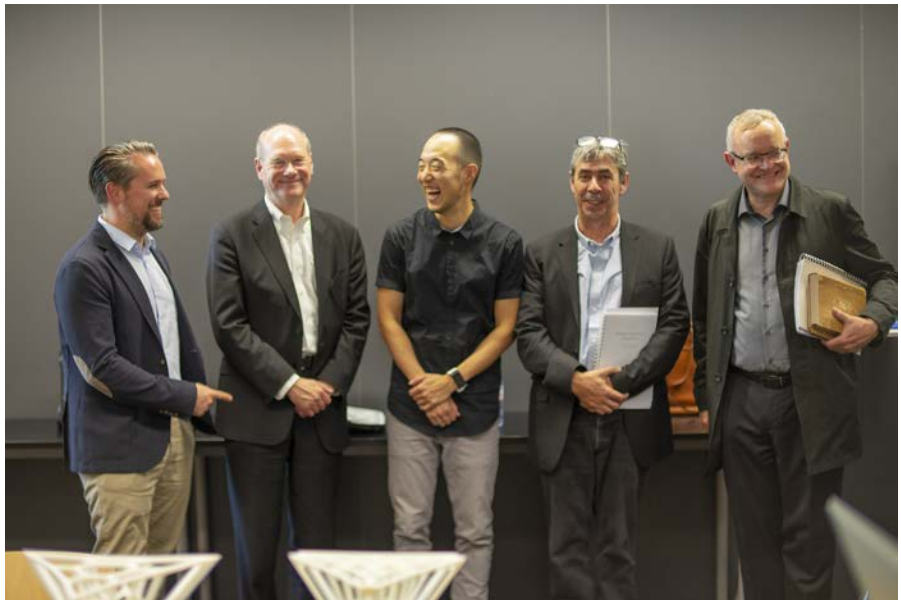
2012 **Skidmore, Owings & Merrill**
Chicago, IL, USA

2009 - 2010 **Grimshaw Architects**
New York, NY, USA

2008 - 2009 **Skidmore, Owings & Merrill**
San Francisco, CA, USA

2007 - 2008 **EndresWare Architects & Engineers**
Berkeley, CA, USA

Doctoral examination committee



From left to right: Prof. Dr. Philippe Block, Hon. D.Sc. William F. Baker, the author, Prof. Allan McRobie, Prof. Dr. Stefan Holzer. (Photo by Matthias Rippmann)

Supervisor

Prof. Dr. Philippe Block	ETH Zurich

Co-advisors

Hon. D.Sc. William F. Baker	Skidmore, Owings & Merrill LLP
Prof. Allan McRobie	University of Cambridge

Head of the committee

Prof. Dr. Stefan Holzer	ETH Zurich