063-0606-22L : Computational Structural Design II

Structurally-informed Materialisation

Serban Bodea & Dr. Juney Lee

Spring Semester 2022

ETH zürich DARCH BRG


BRG
http://block.arch.ethz.ch
@blockresearchgroup


FIRST
STREET
GARDEN
IDEAS
CITY
MAY 28-30
FESTIVAL FOR
THE FUTURE
ideas-city.org


BRG

Lecturers

Serban Bodea

Dr. Juney Lee

Doctoral student instructors

Selina Bitting

Chaoyu Du

Teaching assistant

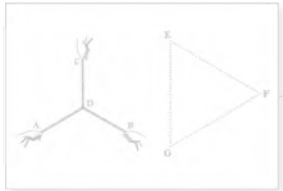
Artemis Maneka

Fall Semester

Spring Semester

Computational Structural Design I • Computational Graphic Statics

2D



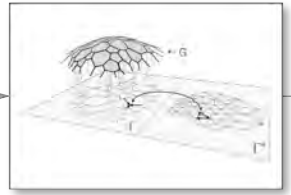
Module I
Basics of Graphic Statics

Module II
Interactive Graphic Statics I (single node)

Module III
Interactive Graphic Statics 2 (multi node)

Module IV
Algebraic Graph Statics

2.5D



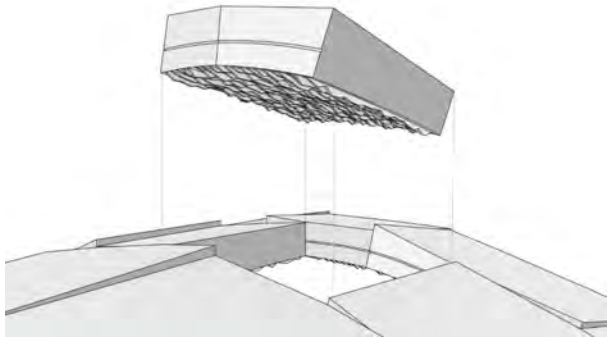
Module V
Thrust Network Analysis

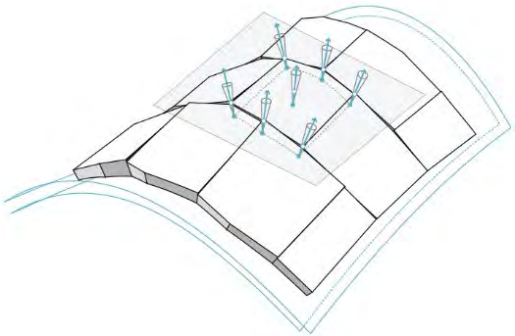
3D

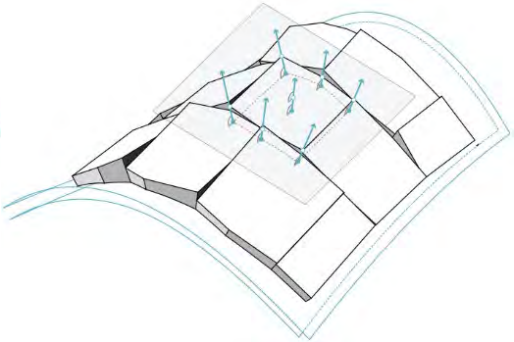
Module VI
3D Graphic Statics

Armadillo Vault, Venice Biennale 2016

Armadillo Vault, Venice Biennale 2016



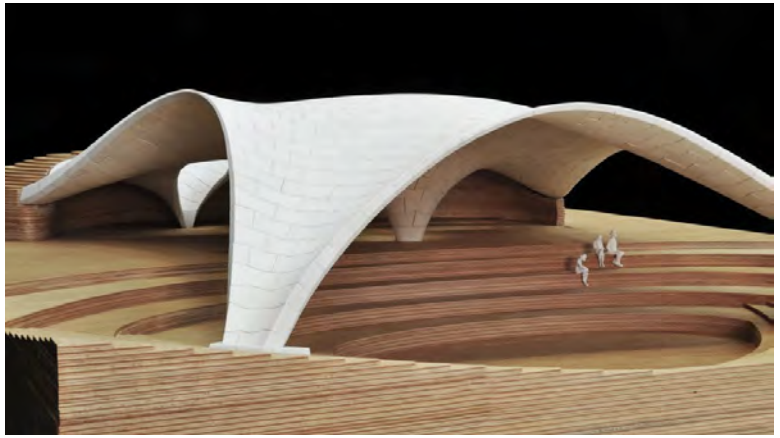


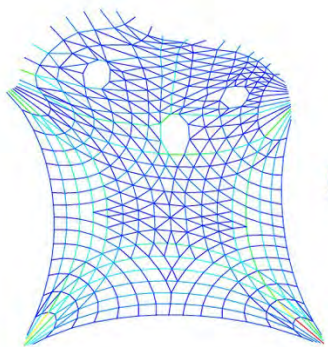


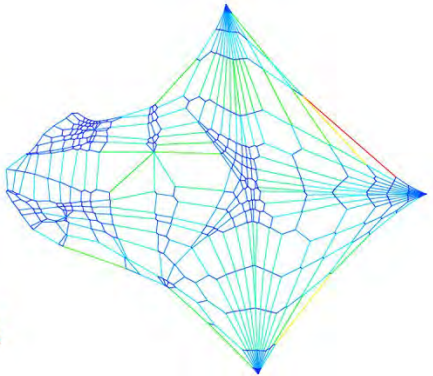
Armadillo Vault, Venice Biennale 2016

Armadillo Vault, Venice Biennale 2016

Module	Week	Hr.	Topic	Instructor	Assignment
I Form finding	Week 1 (2/24)	1	**Lecture 1**	**Form finding of Funicular Shell Structures**	**Dr. Juney Lee**	
		2	Tutorial 1	Form finding with compas-RV2	Dr. Juney Lee	
		3				
	Week 2 (3/3)	2		pseudocodes, Introduction to Jupyter Notebook, Introduction to the Python programming language		
		3	Work Session 2		Chaoyu Du	
II Geometry	Week 3 (3/10)	1	Tutorial 3	Coding in Python: data types, for-loops, conditionals	Serban Bodea and Chaoyu Du	Assignment 1 (individual)
		2	Work Session 3		Chaoyu Du	
		3				
	Week 4 (3/17)	1	Tutorial 4	Introduction to computational geometry in COMPAS	Serban Bodea and Chaoyu Du	
		2		Geometry and Class in COMPAS		
		3	Work Session 4		Chaoyu Du	
	Week 5 (3/24)			Seminar week		
III Materialisation	Week 6 (3/31)	1	**Lecture 2**	**Geometry, Rationalization & Materialization**	**Dr. Juney Lee**	
		2	Tutorial 5	Introduction to The Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du	
		3	Work Session 5		Dr. Juney Lee and Chaoyu Du	
	Week 7 (4/7)	1	Tutorial 6	Operations with the Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du	Assignment 2 (individual)
		2	Work Session 6		Dr. Juney Lee and Chaoyu Du	
		3				
	Week 8 (4/14)	1	Tutorial 7	Geometry rationalization and materialization methods for funicular structures: Case Studies	Dr. Juney Lee and Chaoyu Du	
		2	Work Session 7		Dr. Juney Lee	
		3				
	Week 9 (4/21)			Easter		
IV Fabrication	Week 10 (4/28)	1	**Lecture 3**	**Introduction to Computer Assisted Manufacturing for Architecture Engineering and Construction**	**Serban Bodea**	
		2	Tutorial 8	Subtractive Manufacturing methods	Serban Bodea	
		3	Work Session 8		Selina Bitting and Chaoyu Du	
	Week 11 (5/5)	1	Tutorial 9	Introduction to Hotwire Cutting	Chaoyu Du	Assignment 3 (in groups)
		2	Work Session 9		Chaoyu Du and Selina Bitting	
		3				
	Week 12 (5/12)	1	Tutorial 10	Introduction to Milling	Selina Bitting	
		2	Work Session 10		Selina Bitting and Chaoyu Du	
		3				
	Week 13 (5/19)	1	**Lecture 4**	**Guest lecture: Dr. Catherine De Wolf**		
		2		Group presentations of assignment 3		
		3				





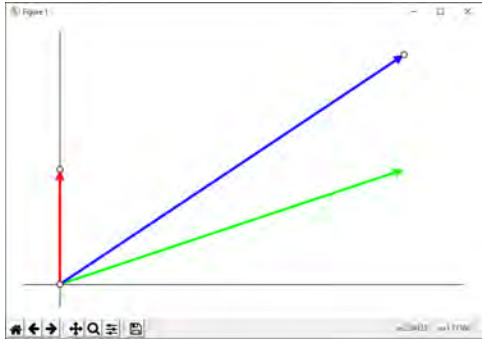


I	Form finding	Week 1 (2/24)	1	Lecture 1	Form finding of Funicular Shell Structures	Dr. Juney Lee
			2	Tutorial 1	Form finding with compas-RV2	Dr. Juney Lee
			3			
II	Geometry	Week 2 (3/3)	1	Tutorial 2	Procedural thiking: logic diagrams, flow charts, pseudocodes, Introduction to Jupyter Notebook, Introduction to the Python programming language	Serban Bodea and Chaoyu Du
			2			
			3	Work Session 2		Chaoyu Du
		Week 3 (3/10)	1	Tutorial 3	Coding in Python: data types, for-loops, conditionals	Serban Bodea and Chaoyu Du
			2	Work Session 3		Chaoyu Du
			3			
		Week 4 (3/17)	1	Tutorial 4	Introduction to computational geometry in COMPAS Geometry and Class in COMPAS	Serban Bodea and Chaoyu Du
			2			
			3	Work Session 4		Chaoyu Du
		Week 5 (3/24)	Spring break			
III	Materialisation	Week 6 (3/31)	1	Lecture 2	Geometry, Rationalization & Materialization	Dr. Juney Lee
			2	Tutorial 5	Introduction to The Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du
			3	Work Session 5		Dr. Juney Lee and Chaoyu Du
		Week 7 (4/7)	1	Tutorial 6	Operations with the Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du
			2	Work Session 6		Dr. Juney Lee and Chaoyu Du
			3			
		Week 8 (4/14)	1	Tutorial 7	Geometry rationalization and materialization methods for funicular structures: Case Studies	Dr. Juney Lee and Chaoyu Du
			2	Work Session 7		Dr. Juney Lee
			3			
		Week 9 (4/21)	Easter			
IV	Fabrication	Week 10 (4/28)	1	Lecture 3	Introduction to Computer Assisted Manufacturing for Architecture Engineering and Construction	Serban Bodea
			2	Tutorial 8	Subtractive Manufacturing methods	Serban Bodea
			3	Work Session 8		Selina Bitting and Chaoyu Du
		Week 11 (5/5)	1	Tutorial 9	Introduction to Hotwire Cutting	Chaoyu Du
			2	Work Session 9		Chaoyu Du and Selina Bitting
			3			
		Week 12 (5/12)	1	Tutorial 10	Introduction to Milling	Selina Bitting
			2	Work Session 10		Selina Bitting and Chaoyu Du
			3			
		Week 13 (5/19)	1	Lecture 4	Guest lecture: Dr. Catherine De Wolf	
			2		Group presentations of assignment 3	
			3			

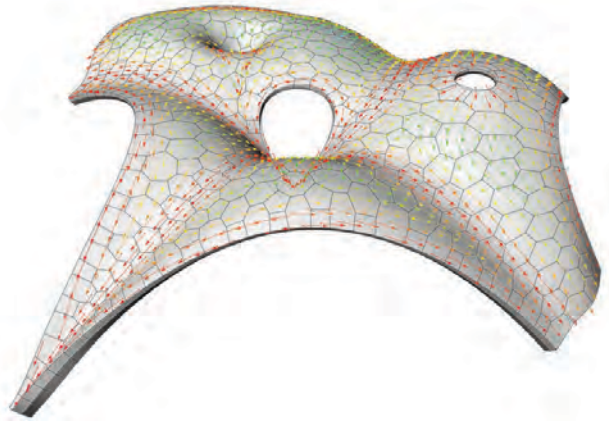
Assignment 1 (individual)

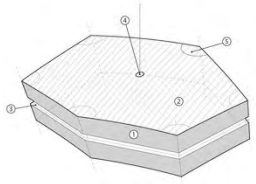
Assignment 2 (individual)

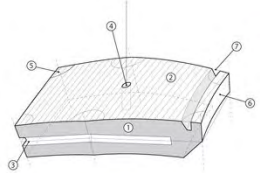
Assignment 3 (in groups)



Module	Week		Title	Topic	Instructor	Assignment
I Form finding	Week 1 (2/24)	1	Lecture 1	Form finding of Funicular Shell Structures	Dr. Juney Lee	
		2	Tutorial 1	Form finding with compas-RV2	Dr. Juney Lee	
		3				
II Geometry	Week 2 (3/3)	1	Tutorial 2	Procedural thiking: logic diagrams, flow charts, pseudocodes, Introduction to Jupyter Notebook, Introduction to the Python programming language	Serban Bodea and Chaoyu Du	
		2				
		3	Work Session 2		Chaoyu Du	
	Week 3 (3/10)	1	Tutorial 3	Coding in Python: data types, for-loops, conditionals	Serban Bodea and Chaoyu Du	Assignment 1 (individual)
		2	Work Session 3		Chaoyu Du	
		3				
	Week 4 (3/17)	1	Tutorial 4	Introduction to computational geometry in COMPAS Geometry and Class in COMPAS	Serban Bodea and Chaoyu Du	
		2				
		3	Work Session 4		Chaoyu Du	
	Week 5 (3/24)			Seminar week		
III Materialisation	Week 6 (3/31)	1	Lecture 2	Geometry, Rationalization & Materialization	Dr. Juney Lee	
		2	Tutorial 5	Introduction to The Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du	
		3	Work Session 5		Dr. Juney Lee and Chaoyu Du	
	Week 7 (4/7)	1	Tutorial 6	Operations with the Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du	Assignment 2 (individual)
		2	Work Session 6		Dr. Juney Lee and Chaoyu Du	
		3				
	Week 8 (4/14)	1	Tutorial 7	Geometry rationalization and materialization methods for funicular structures: Case Studies	Dr. Juney Lee and Chaoyu Du	
		2	Work Session 7		Dr. Juney Lee	
		3				
	Week 9 (4/21)			Easter		
IV Fabrication	Week 10 (4/28)			Architecture Engineering and Construction		
		2	Tutorial 8	Subtractive Manufacturing methods	Serban Bodea	
		3	Work Session 8		Selina Bitting and Chaoyu Du	
	Week 11 (5/5)	1	Tutorial 9	Introduction to Hotwire Cutting	Chaoyu Du	Assignment 3 (in groups)
		2	Work Session 9		Chaoyu Du and Selina Bitting	
		3				
	Week 12 (5/12)	1	Tutorial 10	Introduction to Milling	Selina Bitting	
		2	Work Session 10		Selina Bitting and Chaoyu Du	
		3				
	Week 13 (5/19)	1	Lecture 4	Guest lecture: Dr. Catherine De Wolf		
		2		Group presentations of assignment 3		
		3				





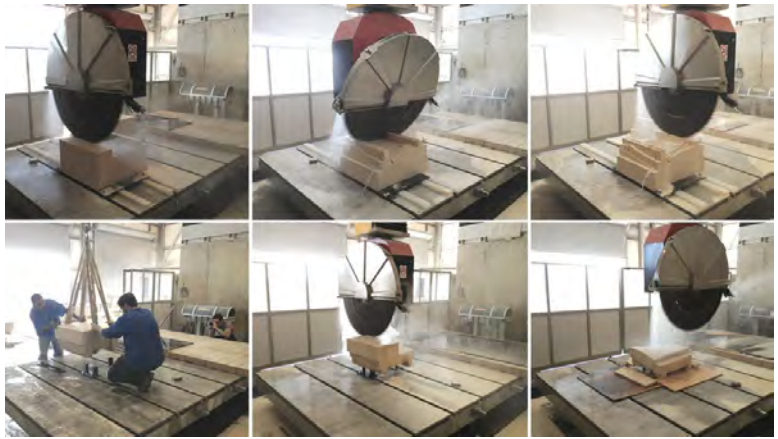


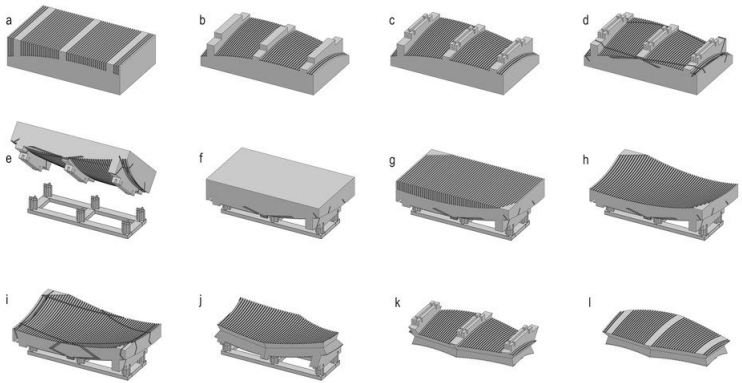
I	Form finding	Week 1 (2/24)	1	Lecture 1	Form finding of Funicular Shell Structures	Dr. Juney Lee
			2	Tutorial 1	Form finding with compas-RV2	Dr. Juney Lee
			3			
II	Geometry	Week 2 (3/3)	1	Tutorial 2	Procedural thiking: logic diagrams, flow charts, pseudocodes, Introduction to Jupyter Notebook, Introduction to the Python programming language	Serban Bodea and Chaoyu Du
			2			
			3	Work Session 2		Chaoyu Du
		Week 3 (3/10)	1	Tutorial 3	Coding in Python: data types, for-loops, conditionals	Serban Bodea and Chaoyu Du
			2	Work Session 3		Chaoyu Du
			3			
		Week 4 (3/17)	1	Tutorial 4	Introduction to computational geometry in COMPAS Geometry and Class in COMPAS	Serban Bodea and Chaoyu Du
			2			
			3	Work Session 4		Chaoyu Du
		Week 5 (3/24)		Seminar week		
III	Materialisation	Week 6 (3/31)	1	Lecture 2	Geometry, Rationalization & Materialization	Dr. Juney Lee
			2	Tutorial 5	Introduction to The Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du
			3	Work Session 5		Dr. Juney Lee and Chaoyu Du
		Week 7 (4/7)	1	Tutorial 6	Operations with the Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du
			2	Work Session 6		Dr. Juney Lee and Chaoyu Du
			3			
		Week 8 (4/14)	1	Tutorial 7	Geometry rationalization and materialization methods for funicular structures: Case Studies	Dr. Juney Lee and Chaoyu Du
			2	Work Session 7		Dr. Juney Lee
			3			
		Week 9 (4/21)		Easter		
IV	Fabrication	Week 10 (4/28)	1	Lecture 3	Introduction to Computer Assisted Manufacturing for Architecture Engineering and Construction	Serban Bodea
			2	Tutorial 8	Subtractive Manufacturing methods	Serban Bodea
			3	Work Session 8		Selina Bitting and Chaoyu Du
		Week 11 (5/5)	1	Tutorial 9	Introduction to Hotwire Cutting	Chaoyu Du
			2	Work Session 9		Chaoyu Du and Selina Bitting
			3			
		Week 12 (5/12)	1	Tutorial 10	Introduction to Milling	Selina Bitting
			2	Work Session 10		Selina Bitting and Chaoyu Du
			3			
		Week 13 (5/19)	1			
			2		Group presentations of assignment 3	
			3			

Assignment 1 (individual)

Assignment 2 (individual)

Assignment 3 (in groups)





I	Form finding	Week 1 (2/24)	1	Lecture 1	Form finding of Funicular Shell Structures	Dr. Juney Lee
			2	Tutorial 1	Form finding with compas-RV2	Dr. Juney Lee
			3			
II	Geometry	Week 2 (3/3)	1	Tutorial 2	Procedural thiking: logic diagrams, flow charts, pseudocodes, Introduction to Jupyter Notebook, Introduction to the Python programming language	Serban Bodea and Chaoyu Du
			2			
			3	Work Session 2		Chaoyu Du
		Week 3 (3/10)	1	Tutorial 3	Coding in Python: data types, for-loops, conditionals	Serban Bodea and Chaoyu Du
			2	Work Session 3		Chaoyu Du
			3			
		Week 4 (3/17)	1	Tutorial 4	Introduction to computational geometry in COMPAS Geometry and Class in COMPAS	Serban Bodea and Chaoyu Du
			2			
			3	Work Session 4		Chaoyu Du
		Week 5 (3/24)			Seminar week	
III	Materialisation	Week 6 (3/31)	1	Lecture 2	Geometry, Rationalization & Materialization	Dr. Juney Lee
			2	Tutorial 5	Introduction to The Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du
			3	Work Session 5		Dr. Juney Lee and Chaoyu Du
		Week 7 (4/7)	1	Tutorial 6	Operations with the Mesh Half-edge data structure	Dr. Juney Lee and Chaoyu Du
			2	Work Session 6		Dr. Juney Lee and Chaoyu Du
			3			
		Week 8 (4/14)	1	Tutorial 7	Geometry rationalization and materialization methods for funicular structures: Case Studies	Dr. Juney Lee and Chaoyu Du
			2	Work Session 7		Dr. Juney Lee
			3			
		Week 9 (4/21)			Easter	
IV	Fabrication	Week 10 (4/28)	1	Lecture 3	Introduction to Computer Assisted Manufacturing for Architecture Engineering and Construction	Serban Bodea
			2	Tutorial 8	Subtractive Manufacturing methods	Serban Bodea
			3	Work Session 8		Selina Bitting and Chaoyu Du
		Week 11 (5/5)	1	Tutorial 9	Introduction to Hotwire Cutting	Chaoyu Du
			2	Work Session 9		Chaoyu Du and Selina Bitting
			3			
		Week 12 (5/12)	1	Tutorial 10	Introduction to Milling	Selina Bitting
			2	Work Session 10		Selina Bitting and Chaoyu Du
			3			
		Week 13 (5/19)	1	Lecture 4	Guest lecture: Dr. Catherine De Wolf	
			2		Group presentations of assignment 3	
			3			

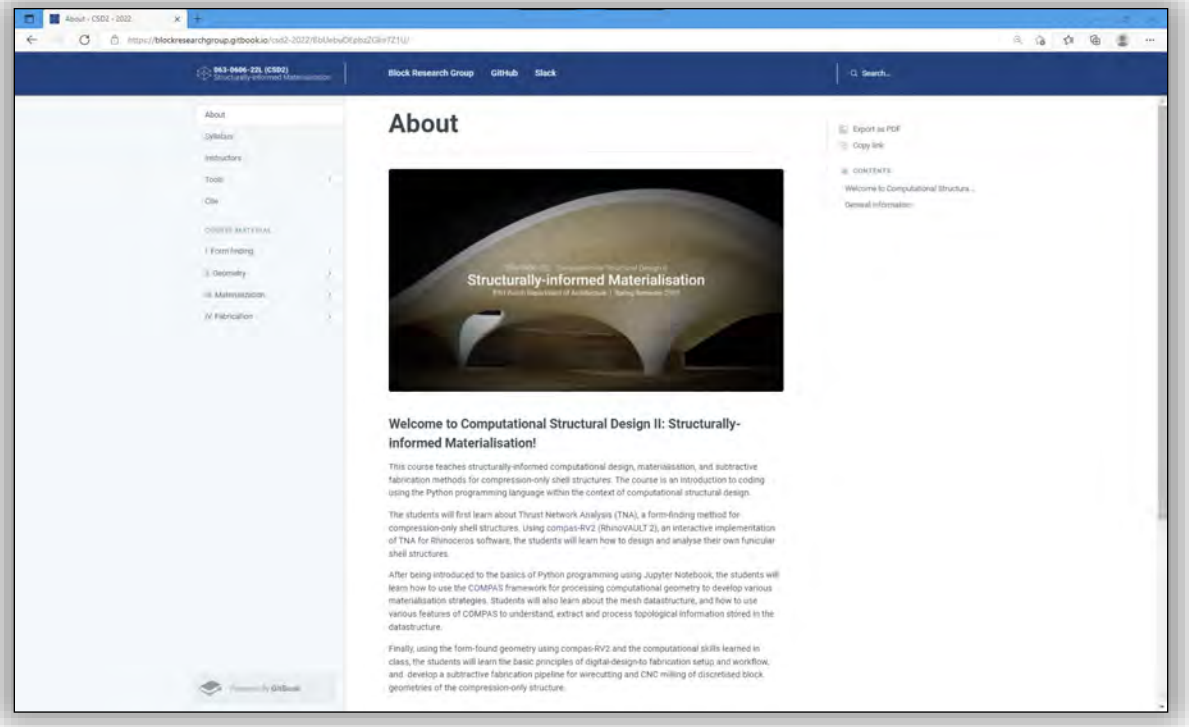
Assignment 1 (individual)

Assignment 2 (individual)

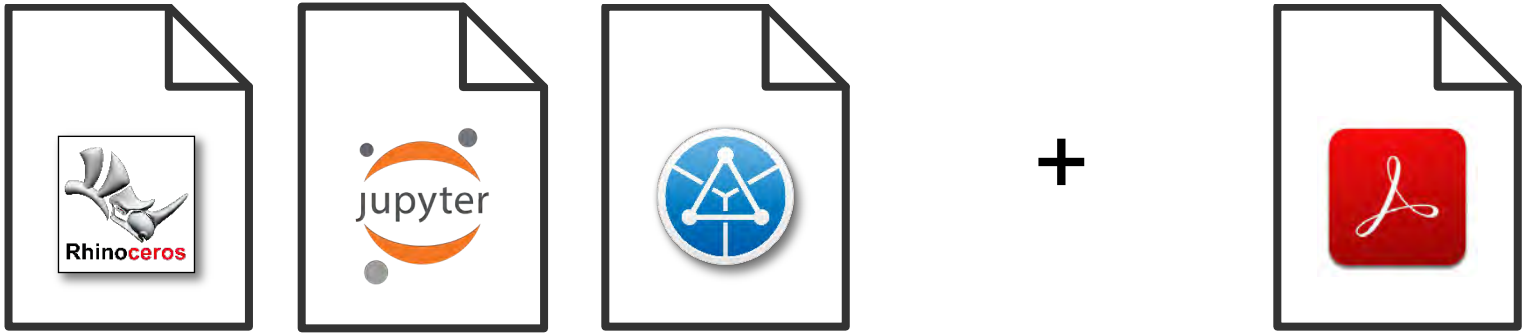
Assignment 3 (in groups)

ETH zürich

D BAUG



Course website ⟶ https://blockresearchgroup.gitbook.io/csd2-2022/



Skill

Understanding

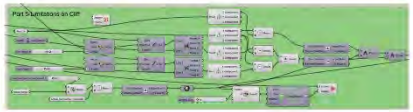
CSD1_FS2020 10/27/20 8:33 PM

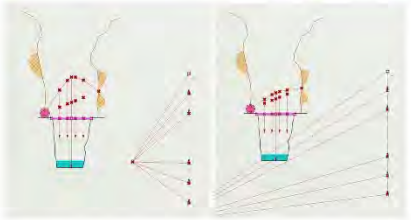
3.
The basic idea is to construct the limitations and let rhino show the warning by displaying a symbol at the footage if either of the 3 conditions is not satisfied.

1)To constrain the inclination on the left side: First, input the limit angles X, Y as number sliders and then use the "Rotate" components to translate them into vectors. Starting from the point A on force diagram, use "Line SDL" to draw two constrain lines based on those vectors and get two intersection points with the vertical line (let's name them as point "Bx" and point "By")in the force diagram. Deconstruct the two points and the top point which represents the first force line. By calculating those Y-coordinates (substrate them in pairs and then multiply the results and then to see if it positive or negative) one could easily inspect if the top point lies in between the two points-"Bx"and"By".

2)To examine if the horizontal components at the left footage: Input the Z value and compare it with the distance of point A to the vertical line. (Deconstruct the point A and calculate the difference between the X-components of point A and the vertical line)

3)To exclude the tension situation: one could simply limit the allowable value of inputted X and Y value: all must bigger than "0".





3

CSD1_HS2020 September 3, 2020 10:16 AM

3.
The maximum horizontal force is defined by a parallel line to the load line from the force diagram. This line moves to the left, according to the maximal load set in the „max Horizontal support" component. The angles, in wich the support can be, can be controlled with the two other components (angle 1 and angle 2).
The move component is defined by the motion in „Unit x" (because the line needs to be moved along the x axis) and a reverse component (so the vector turns negative).
Angle 1 and 2 are defined by a rotation.
Angle 1 is the rotation of the direction of the load-line.
Angle 2 is the the rotation from angle 1.
The direction and the size of the support forces are in this case A and B.

Is the support force A inside the green surface, so the direction of the support force is ok, for the defined angles of Angle 1 and Angle 2.
Is the pole inside the green surface, the direction and the size of the maximal horizontal component are ok for the set parameters.

Direction of the support force inside the set parameters, but the set horizontal component is too small (or the geometry of the bridge too shallow)

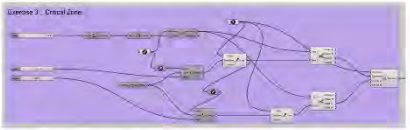
Direction of the support force inside the set parameters. horizontal component also inside the set parameters —> no cross at the pole

2

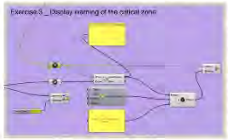
CSD1_FS2020 10/29/20 11:42 PM

The result of the division can not be a round number because in the way I proceeded it will always be an odd number, that's why a used the *Round* component. I then linked the result with *index* output of the *List Item* which is also connected to the *Panel* component of the force magnitudes.
I finally connected the final List to the existing *Amplitude* component. That way I managed to add the force in the middle of the bridge.

Part 3 - Modify the parametric model to provide solutions only in the feasible design space.



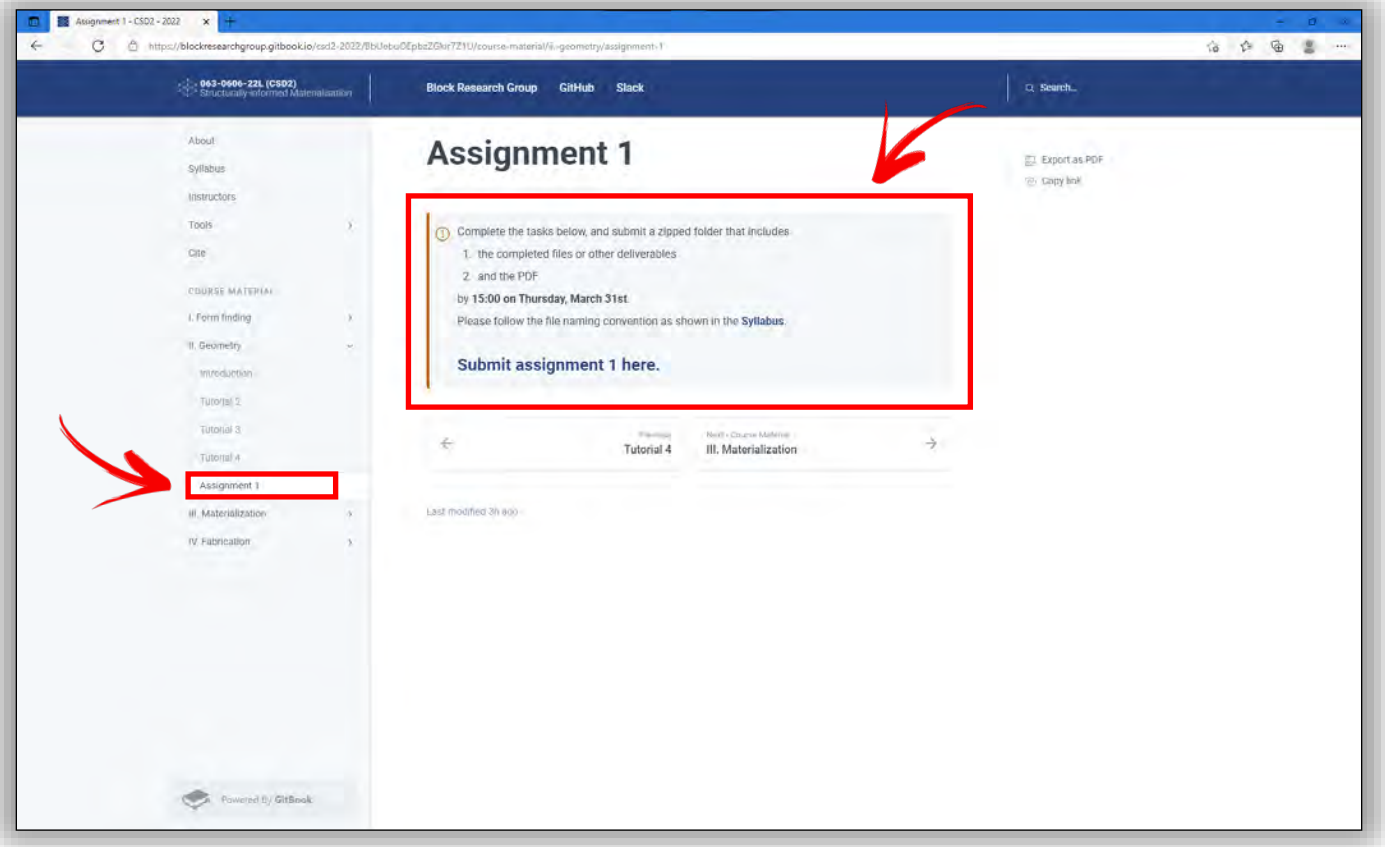
To create the asked limitation we start by adding the *Hmax*, *angle a* and *angle b* values with a number slider.
The Hmax number slider connected to the *Move* component will define the intensity of the movement of the geometry according to the chosen Hmax value. Same for the angles a and b, which are connected the the *Rotate* components. The two angles linked to the Z unit Axis define the rotation of the vector line.

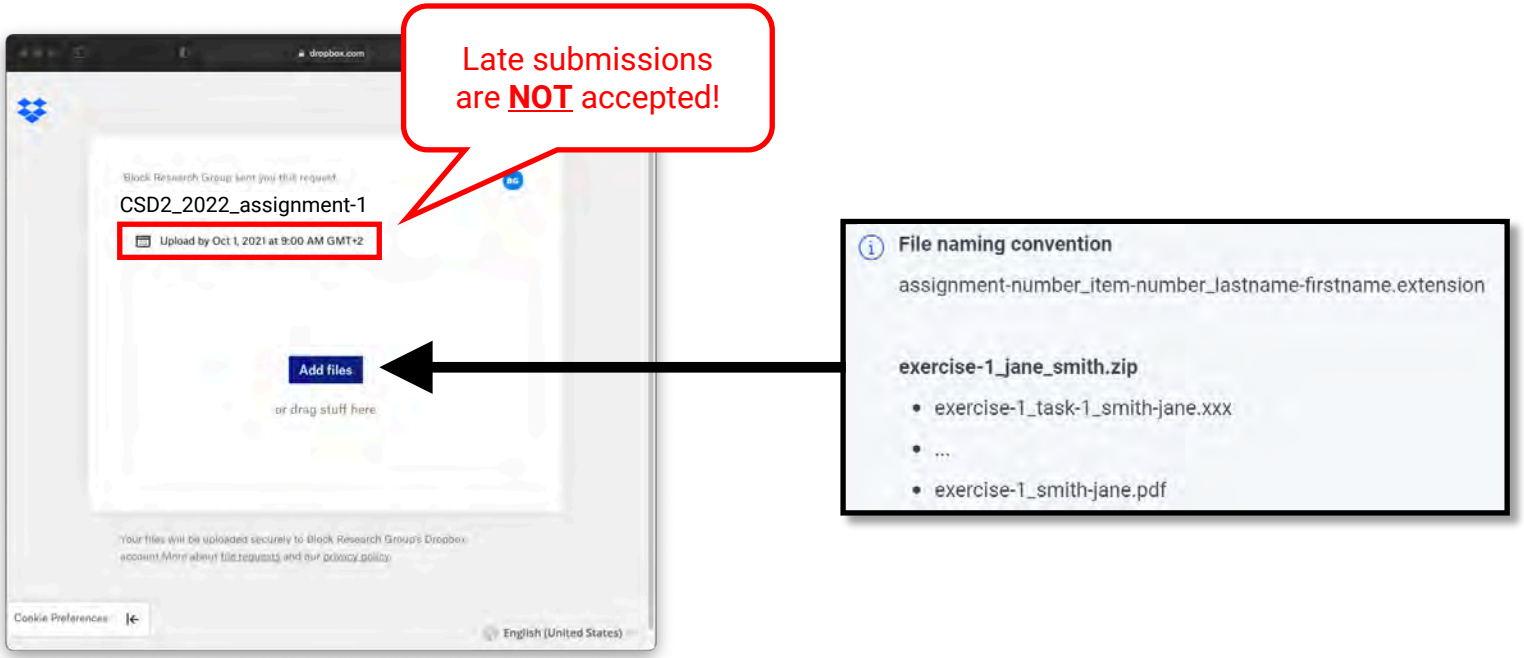


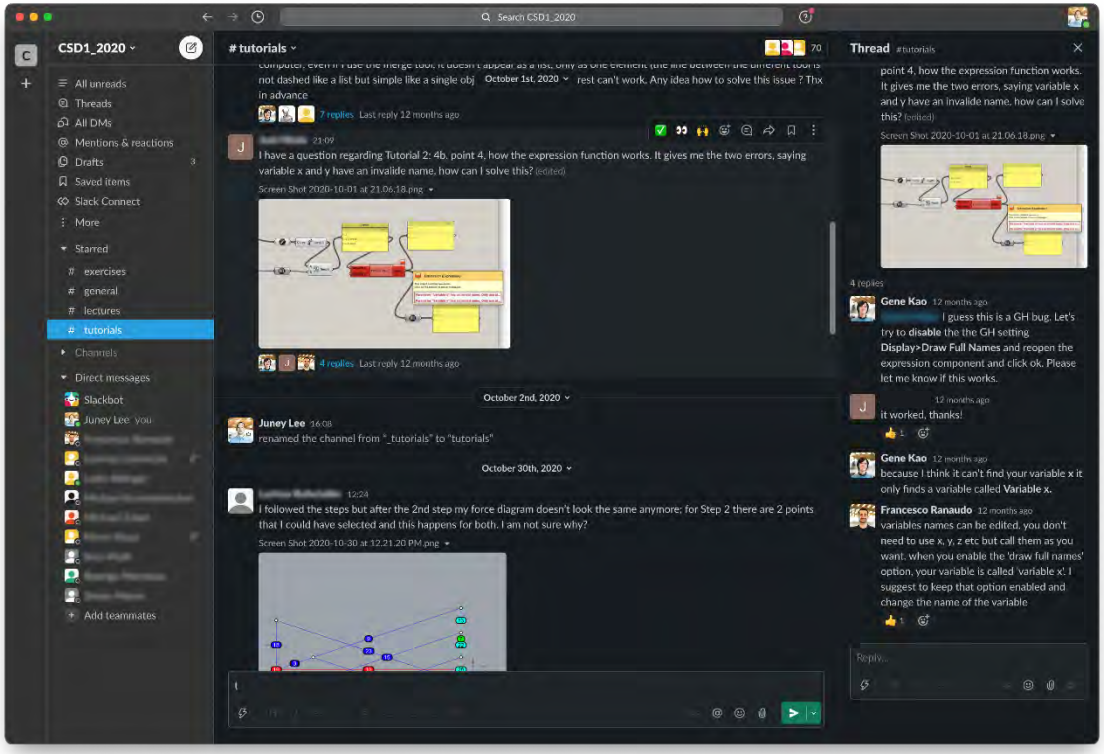
With the Line SDL component I created the vectors starting from the initial point in the force diagram which followed the defined direction of the *previous Rotate* component. The Line / Line component allowed me to create the intersecting points between the two vectors of the angles and the line defined by Hmax.
I finally used the *4 point Surface* component to link the the line defined by the Hmax and the two vector lines of the angles in order to get the feasible region.

In order to be informed when the anchors are not anymore in the feasible zone (out of the defined surface from the previous point) I created a warning display.

2


Assignment 1 - CSD2 - 2022
https://blockresearchgroup.gitbook.io/csd2-2022/8bUebu0EpbzZGkir7Z1U/course-material/ii.-geometry/assignment-1
063-0606-22L (CSD2)
Structurally-informed Materialization
Block Research Group
GitHub
Slack
Search...
About
Syllabus
Instructors
Tools
Cite
COURSE MATERIAL
I. Form finding
II. Geometry
Introduction
Tutorial 2
Tutorial 3
Tutorial 4
Assignment 1
III. Materialization
IV. Fabrication
Powered by GitBook
Assignment 1
Export as PDF
Copy link
Complete the tasks below, and submit a zipped folder that includes:
1. the completed files or other deliverables
2. and the PDF
by 15:00 on Thursday, March 31st.
Please follow the file naming convention as shown in the Syllabus.
Submit assignment 1 here.
Previous
Tutorial 4
Next - Course Material
III. Materialization
Last modified 3h ago


Late submissions are NOT accepted!
dropbox.com
CSD2_2022_assignment-1
Upload by Oct 1, 2021 at 9:00 AM GMT+2
Add files
or drag stuff here
Your files will be uploaded securely to Block Research Group's Dropbox
Cookie Preferences
English (United States)
File naming convention
assignment-number_item-number_lastname-firstname.extension
exercise-1_jane_smith.zip
exercise-1_task-1_smith-jane.xxx
...
exercise-1_smith-jane.pdf



Course Slack Workspace → Link provided on the course GitBook

Week 1 Thursday, February 24th

Hour 1	**15:45 – 15:50**	· Introduction
	15:50 – 16:30	· **Lecture 1 : Form finding of Funicular Shell Structures**
	16:30 – 16:45	· Break
Hour 2	**16:45 – 17:30**	· **Tutorial 1 : compas-RV2 (RhinoVAULT 2) – Part 1**
	17:30 – 17:45	· Break
Hour 3	**17:45 – 18:30**	· **Tutorial 1 : compas-RV2 (RhinoVAULT 2) – Part 2**

063-0606-22L : Computational Structural Design II

Structurally-informed Materialisation

Lecture 1

Form Finding of Funicular Shell Structures

Thursday, February 24th, 2022

Dr. Juney Lee

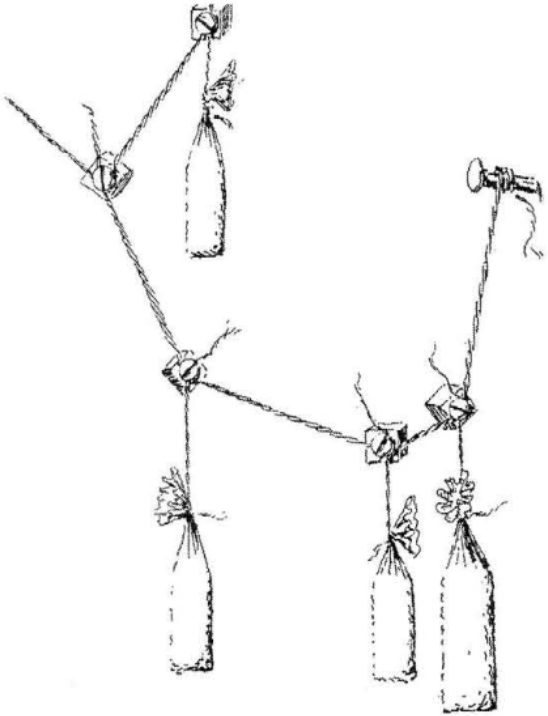
ETH zürich DARCH BRG

Form finding

“As hangs the flexible line, so but inverted will stand the rigid arch.”

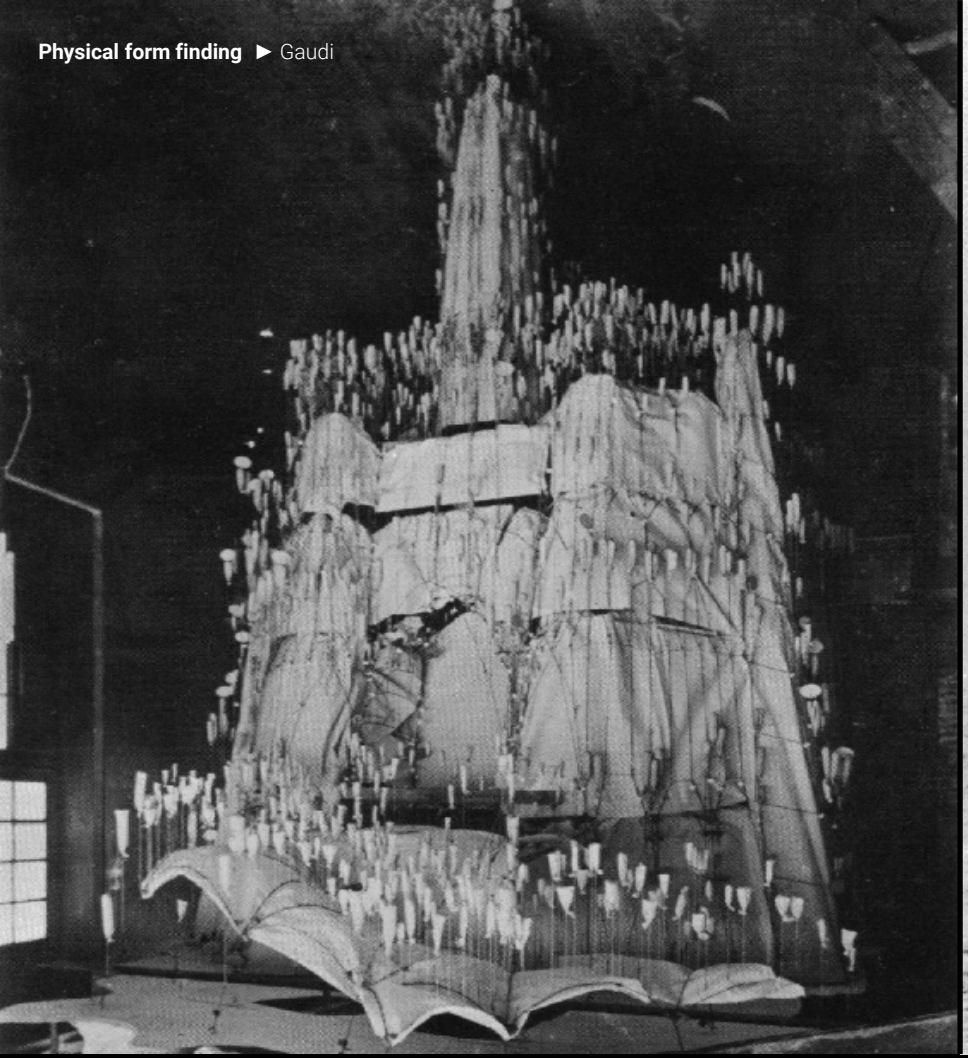
Hooke (1676)

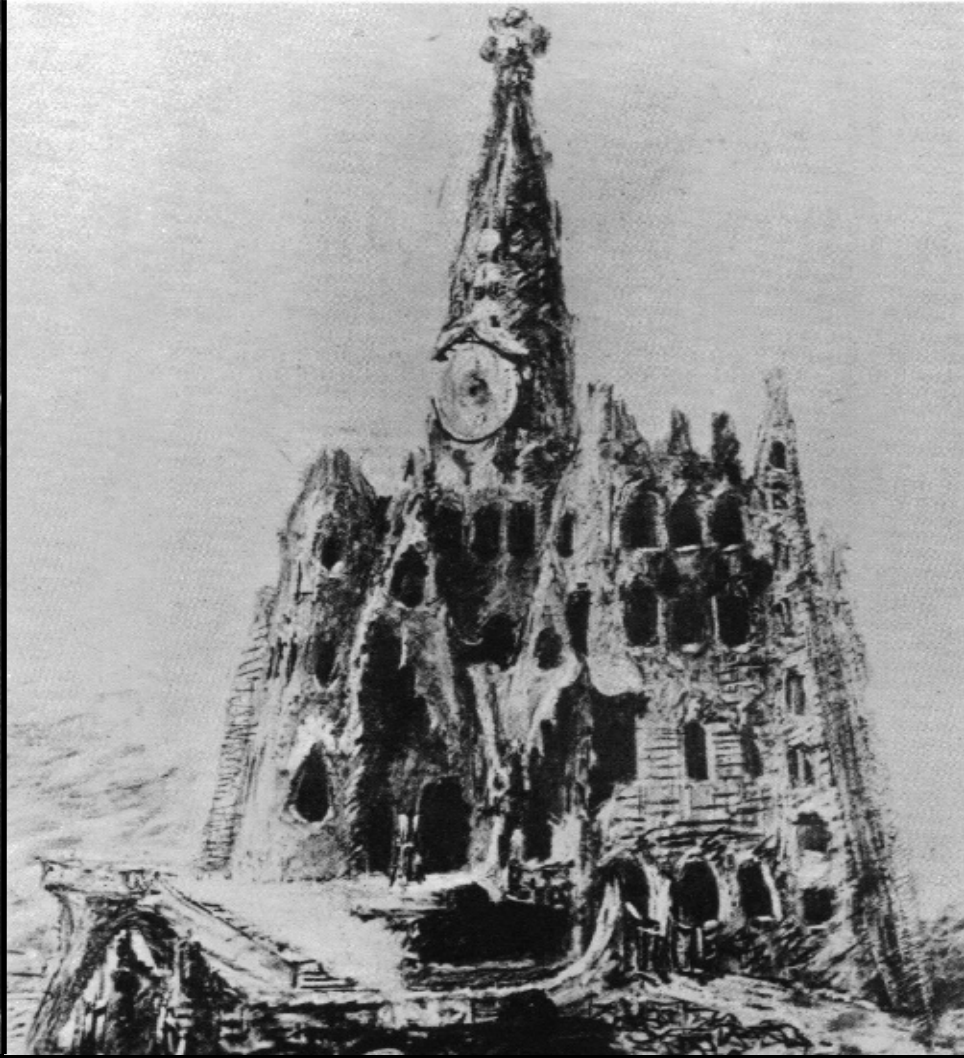
Credit: Prof. G. Wayne Brodland, University of Waterloo



Redesign of Gaudí's hanging model for the Colonia Güell | Images: Beotis (2007)

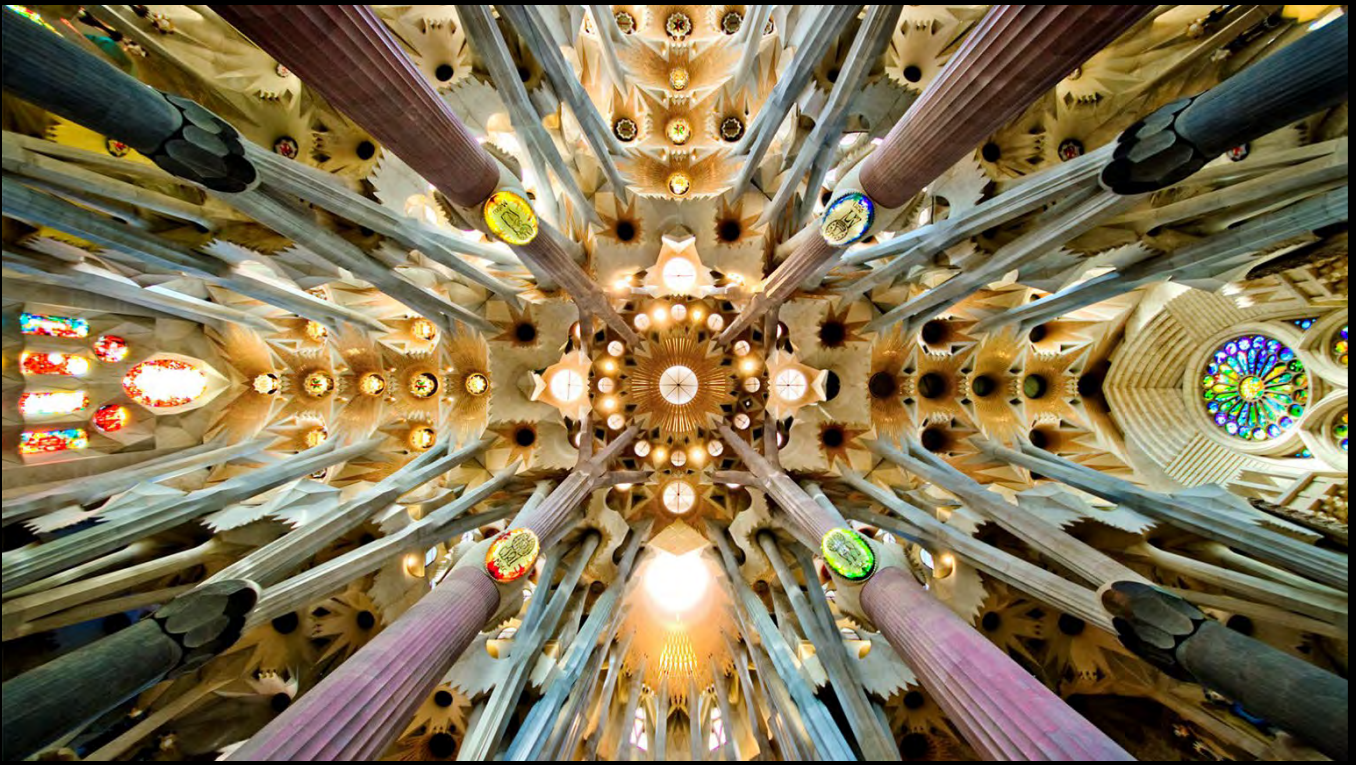
Physical form finding ► Gaudi



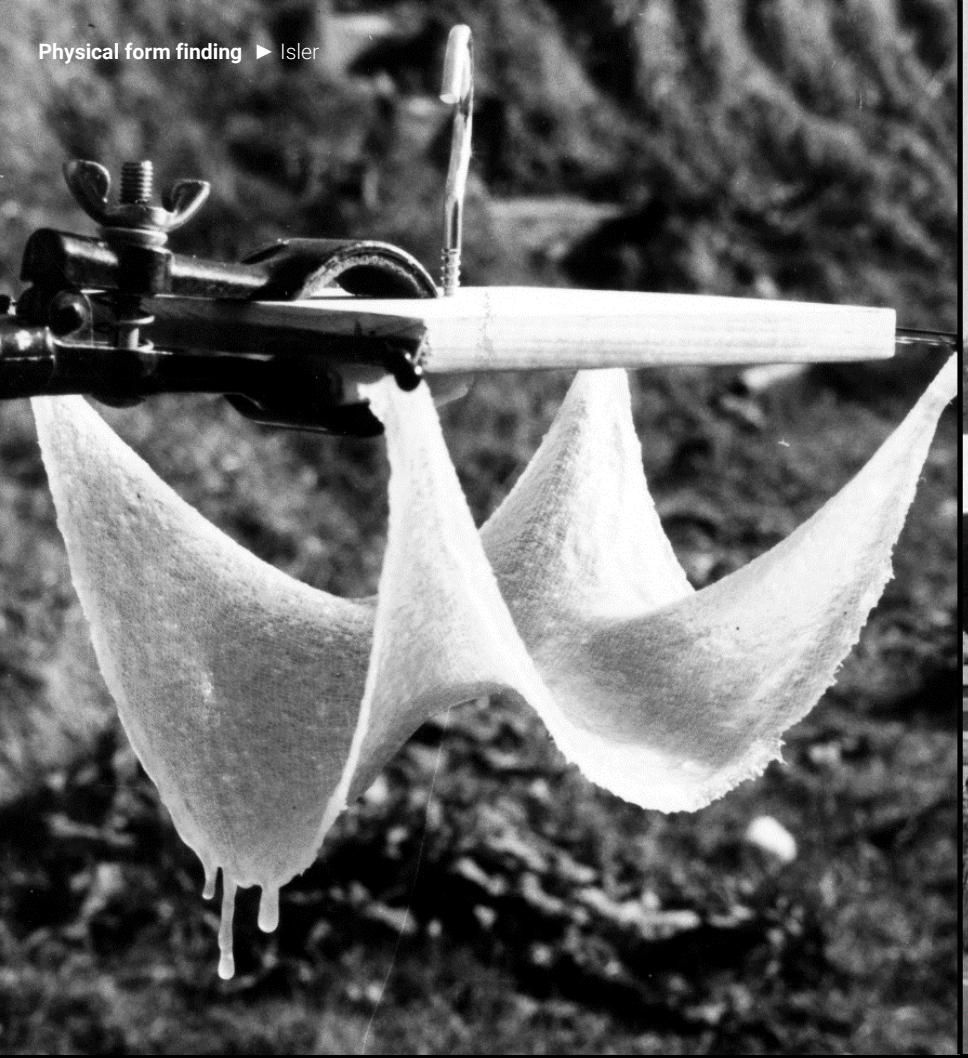


Gaudí's design of its exterior directly sketched on the inverted photograph of the hanging model | Images: Collins, 1963

Gaudí's Church of Colònia Güell, 1898

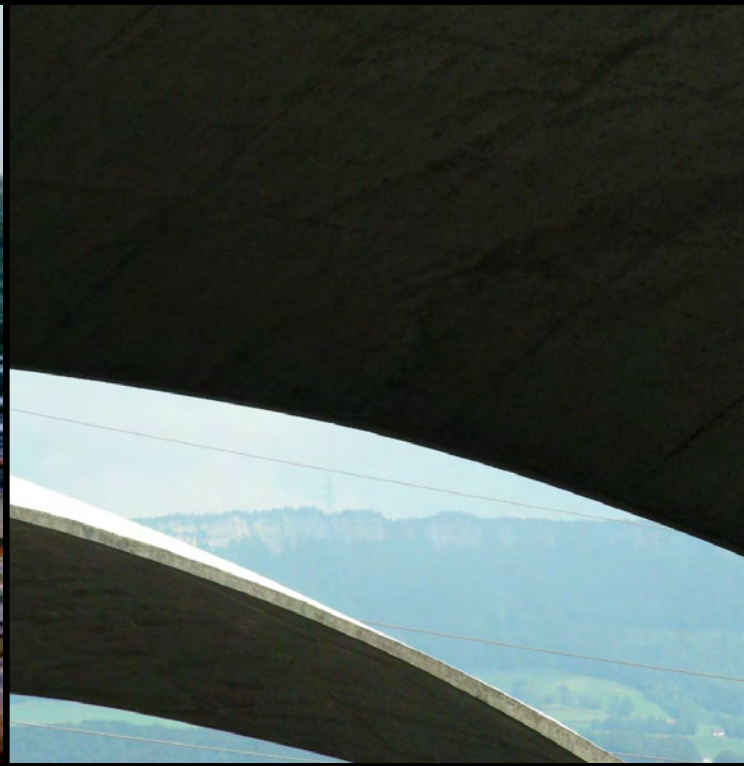


Gaudí's Sagrada Família | Images: Wikipedia



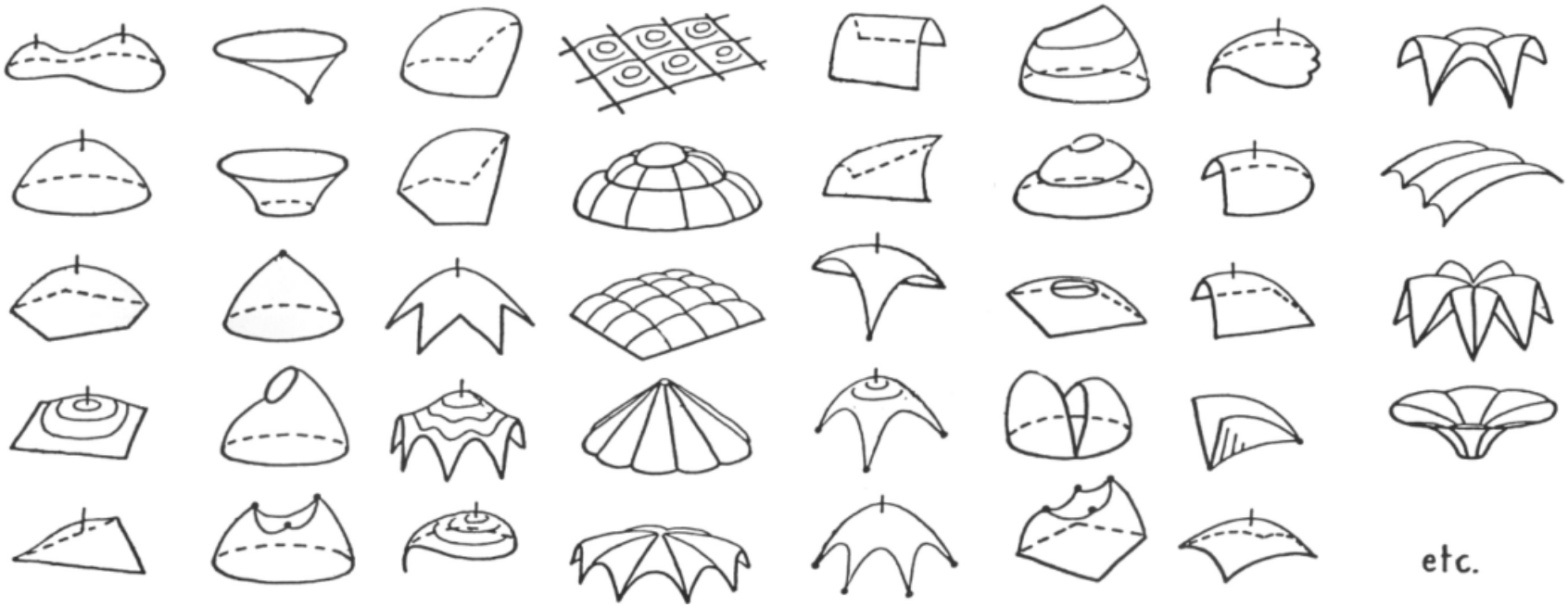
Physical Form Finding Models. | Isler Archive

Tankstelle Deitingen Süd, Schweiz, 1968, Ing.: Heinz Isler

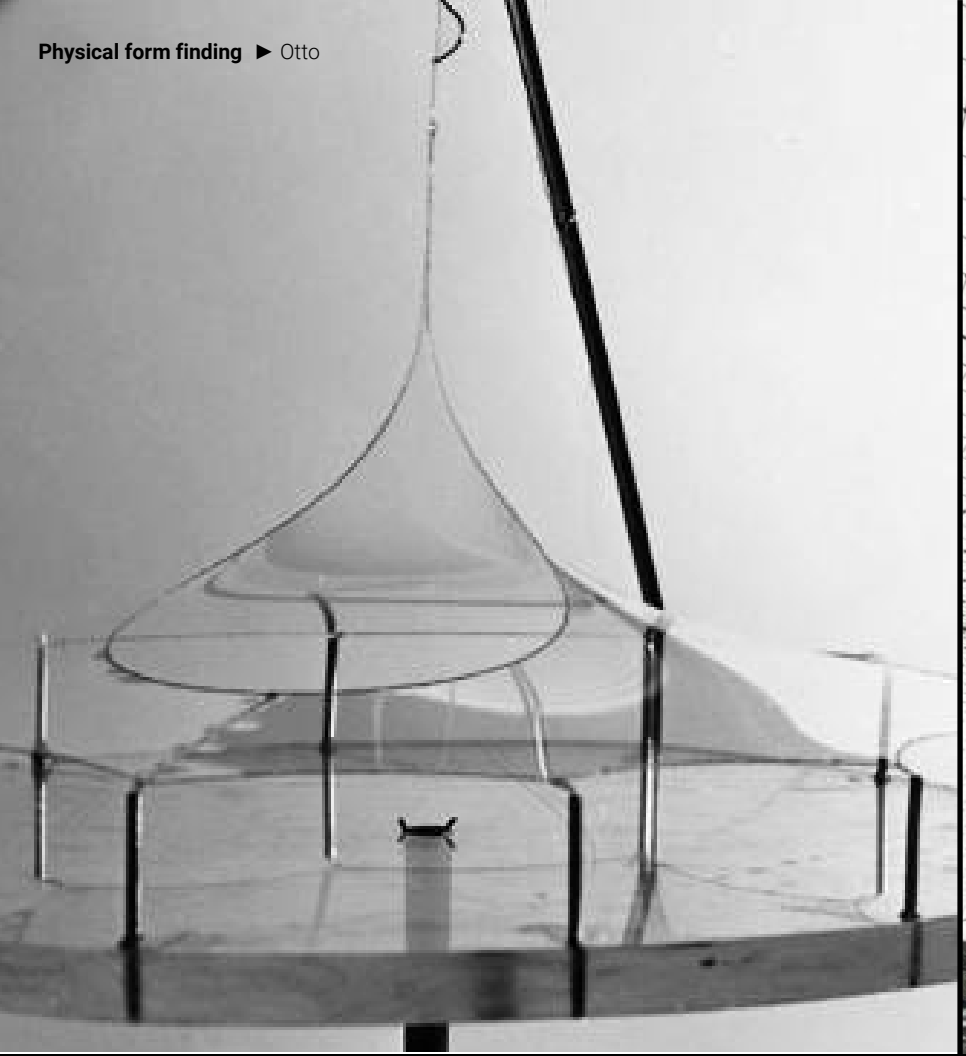


Tankstelle Deitingen Süd, Schweiz, 1968, Ing.: Heinz Isler

Tankstelle Deitingen Süd, Schweiz, 1968, Ing.: Heinz Isler



"New shapes for shells" Heinz Isler, First Congress of the International Association for Shell Structures (IASS) in 1959



Soap bubble experiments | Frei Otto

German Pavillion at Expo 1967 | Frei Otto

The Institute for Lightweight Structures (IL)
University of Stuttgart

Director: Frei Otto

Roof for Munich Olympic Stadium 1972

Montreal Pavilion measuring model, 1966

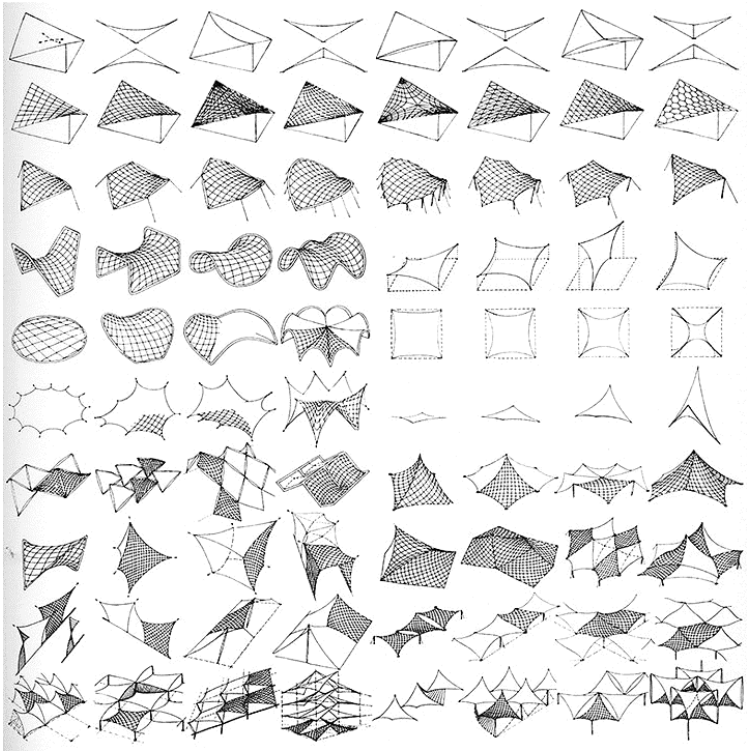
Munich Stadium measuring model, ca. 1970

Computational **form finding**

Form finding

"… forward process in which
parameters are explicitly/directly controlled
to find an 'optimal' geometry of a structure
which is in static equilibrium
with a design loading…"

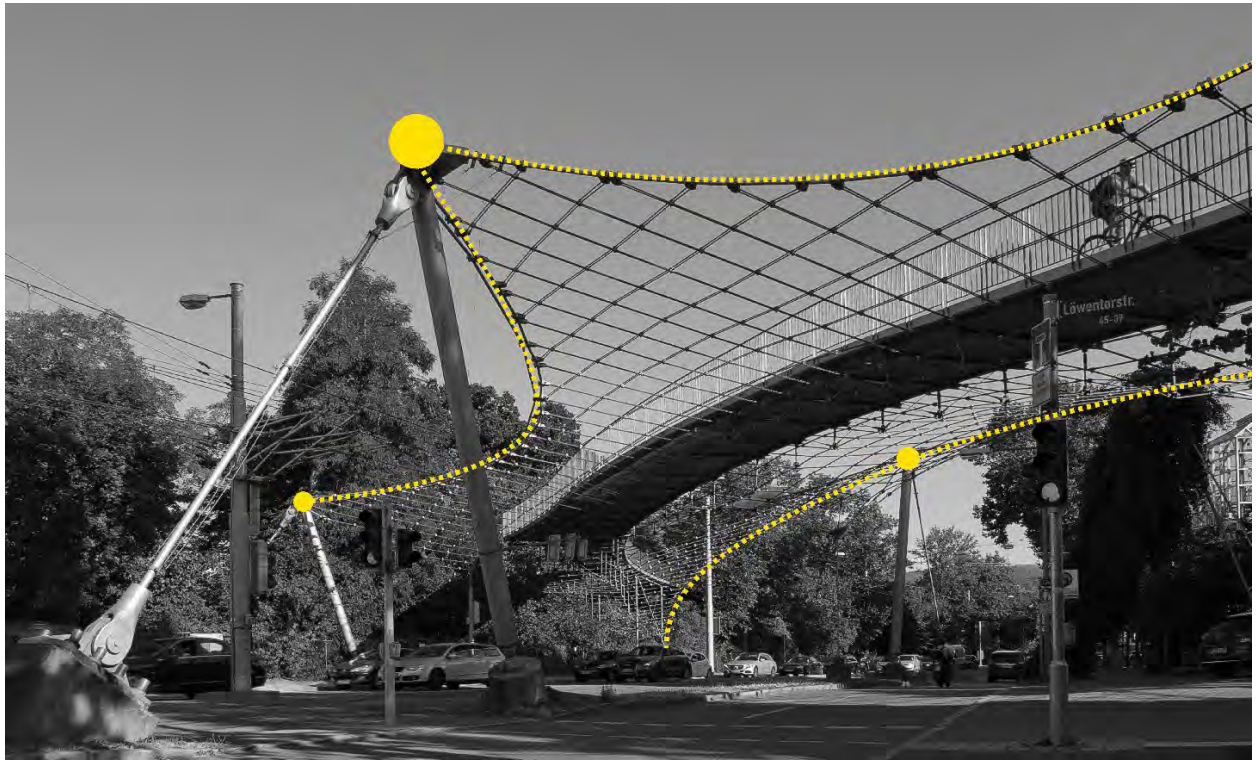
Adriaenssens et al. (2014)



Frei Otto. Casabella 301 1966: 35

Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

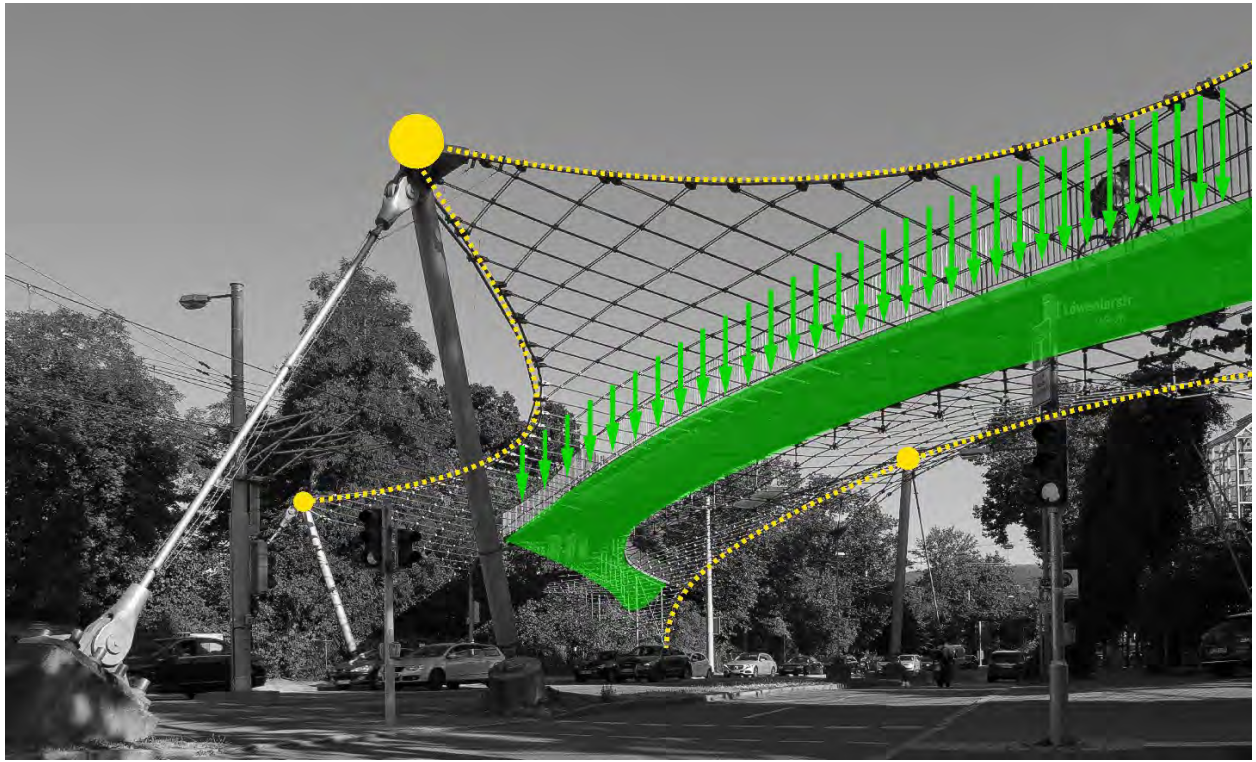
1. Defined boundary conditions



Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

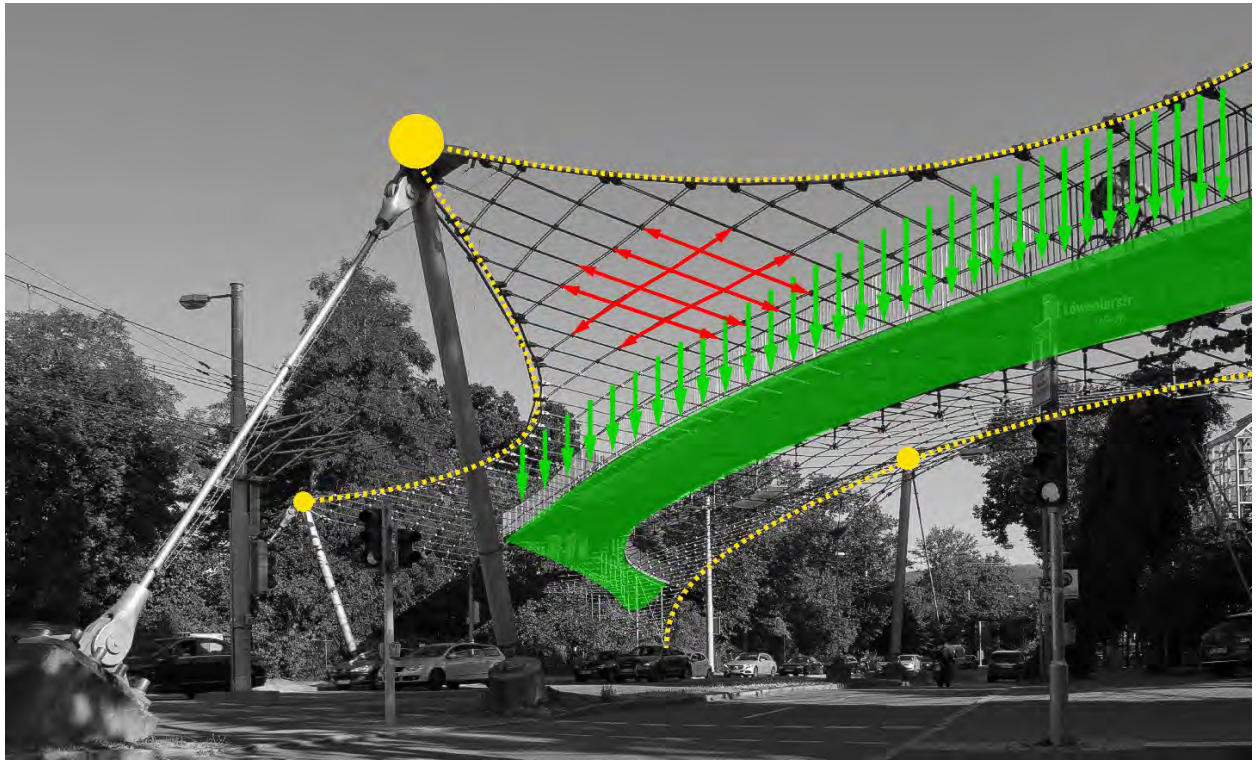
1. Defined boundary conditions

2. Defined loading case



Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

1. Defined boundary conditions
2. Defined loading case
3. Defined state of self stress



Schlaich Bergermann Partner, Lodzer Steg Stuttgart, 1992

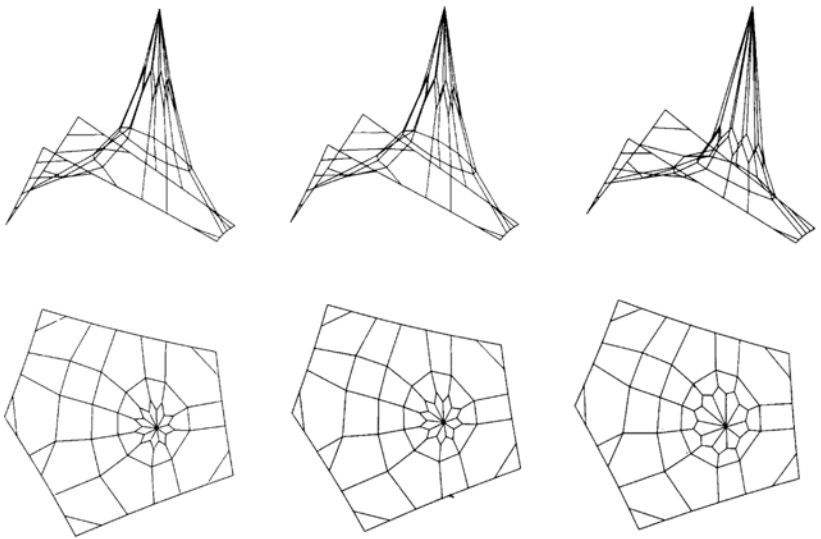


Fig. 7. Global changes in the force densities and lengthening of the radial branches.

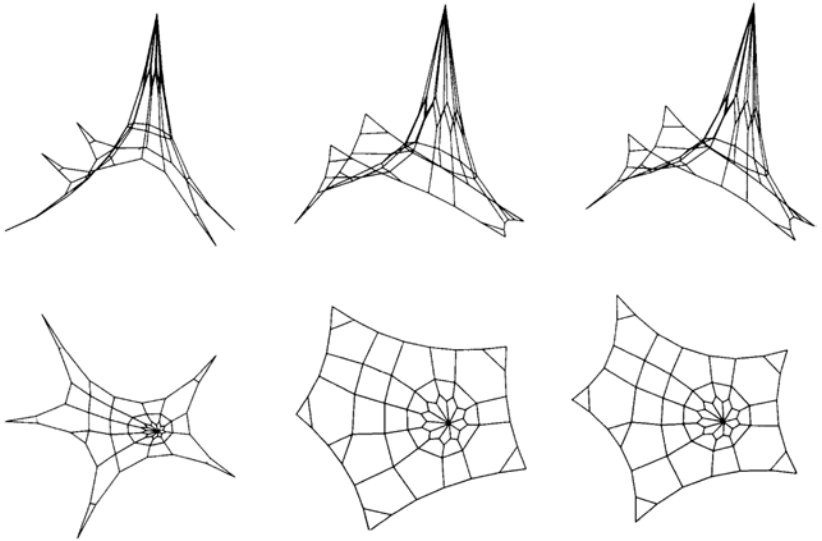


Fig. 8. Global changes in the force densities of the main cables.

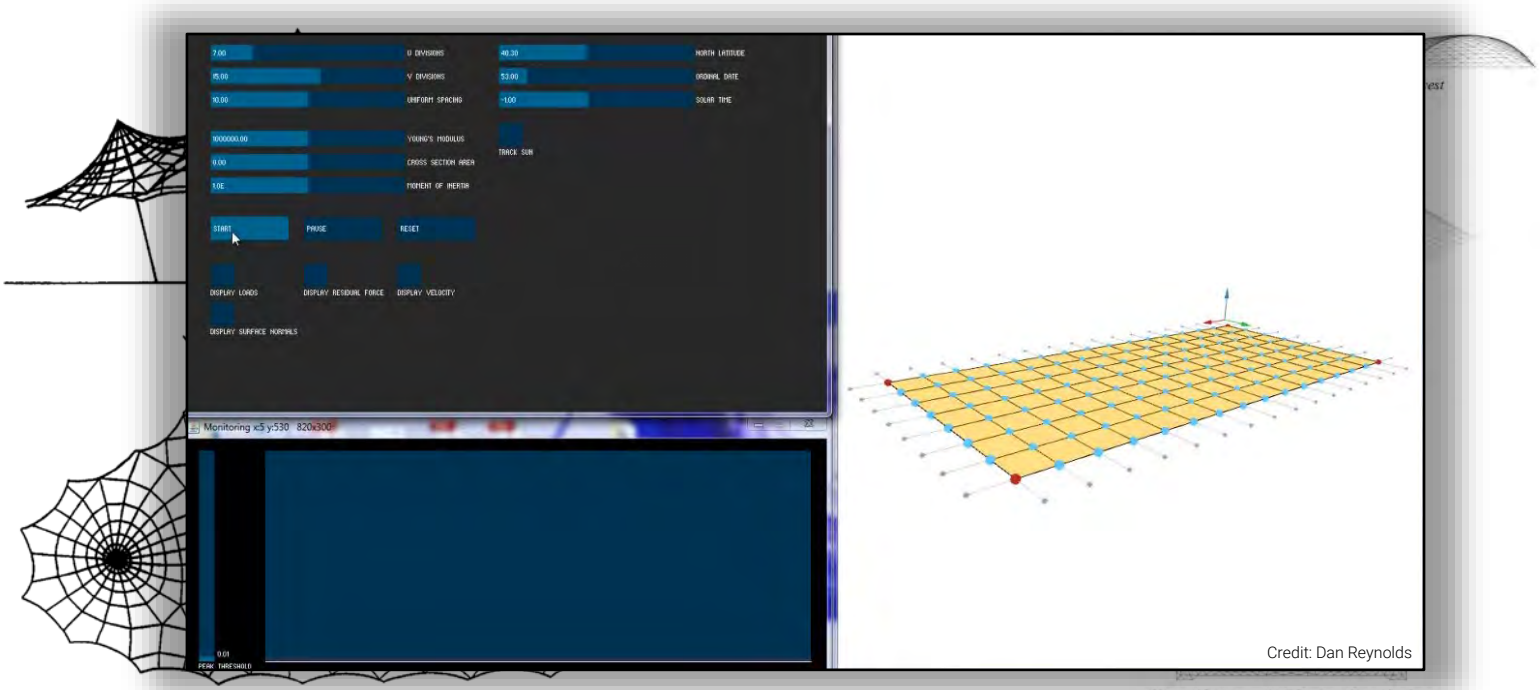


Figure 13. Outwards deflections due to loading

"Form-finding and analysis of prestressed nets and membranes" | Barnes (1988)

"The analytic and numerical definition of the geometry of the British Museum Great Court Roof" | Williams (2001)

Form finding with graphic statics

JOURNAL OF THE INTERNATIONAL ASSOCIATION FOR SHELL AND SPATIAL STRUCTURES: J. IASS

THRUST NETWORK ANALYSIS: A NEW METHODOLOGY FOR THREE-DIMENSIONAL EQUILIBRIUM

PHILIPPE BLOCK[1] AND JOHN OCHSENDORF[2]

[1]Research Assistant, ph_block@mit.edu, [2]Associate Professor, jao@mit.edu, Building Technology Program, Massachusetts Institute of Technology, Room 5-418, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

Editor's Note: The first author of this paper is one of the four winners of the 2007 Hangai Prize, awarded for outstanding papers that are submitted for presentation and publication at the annual IASS Symposium by younger members of the Association (under 30 years old). It is re-published here with permission of the editors of the proceedings of the IASS 2007 Symposium: Structural Architecture: Toward the Future Looking to the Past, held in December 2007 in Venice, Italy.

ABSTRACT

This paper presents a new methodology for generating compression-only vaulted surfaces and networks. The method finds possible funicular solutions under gravitational loading within a defined envelope. Using projective geometry, duality theory and linear optimization, it provides a graphical and intuitive method, adopting the same advantages of techniques such as graphic statics, but offering a viable extension to fully three-dimensional problems. The proposed method is applicable for the analysis of vaulted historical structures, specifically in unreinforced masonry, as well as the design of new vaulted structures. This paper introduces the method and shows examples of applications in both fields.

Keywords: *compression-only structures, unreinforced masonry vaults, funicular analysis, Thrust Network Analysis, reciprocal diagrams, form-finding, lower-bound analysis, structural optimization*

1. INTRODUCTION

Medieval vault builders created complex forms carefully balanced in compression. The structural properties of these sophisticated forms are still poorly understood because of a lack of appropriate analysis methods, i.e. methods relating stability and form. Understanding the mechanics of these vaulted structures leads to new insights for both analysis and design.

Thrust Line Analysis is a powerful graphical method for calculating the range of lower-bound equilibrium solutions of compression-only systems, such as unreinforced masonry structures (Fig. 1). It visualizes the stability of these structures and suggests possible collapse mechanisms [1]. Unfortunately, thrust line analysis is primarily suitable for 2-D cases and this limitation has prevented it from being used for the assessment of complex 3-D structures. While numerical methods

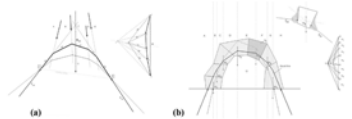


Figure 1. *(a) Two possible compression-only equilibrium shapes for a random set of loads, and (b) an interactive thrust-line application developed in [1]: the user can adopt the geometry by dragging control points and the structural feedback, in the form of a thrust-line, is updated in real-time. The magnitudes of the forces in the system are visualized in the accompanying funicular polygon (right).*

Vol. 48 (2007) No. 3 December n. 155

based on elastic solutions give one possible answer, they no longer suggest better form as was inherent to the more holistic graphical methods.

There is a real need for tools to better understand and visualize the stability of compression-only structures, such as historic unreinforced masonry structures, as well as design tools that suggest better form. Both problems are related to finding axial force structures in equilibrium acting only in compression or tension. Currently, graphic statics provides a holistic analysis and design tool for two-dimensional structures. With today's availability of powerful virtual 3-D and parametric environments, the following question arises: can a fully three-dimensional version of thrust-line analysis provide the same freedom to explore the infinite equilibrium solutions for a certain loading condition?

2. METHODOLOGY

The *Thrust-Network Analysis* method presented in this paper is inspired by O'Dwyer's work on funicular analysis of vaulted masonry structures [2]. It is extended by adding the concept of duality between geometry and the in-plane internal forces of networks [3].

2.1. Reciprocal figures

The duality between the geometry of a network and its internal forces is an old concept, first explained by Maxwell [4]. He called this relationship *reciprocal* and defined it as follows: *"Two plane figures are reciprocal when they consist of an equal number of lines, so that corresponding lines in the*

Figure 2. *The primal grid □ and dual grid □* are related by a reciprocal relationship. The equilibrium of a node in one of them is guaranteed by a closed polygon in the other and vice versa. The labelling was done to continue [6].*

two figures are parallel, and corresponding lines which converge to a point in one figure form a closed polygon in the other." This means that the equilibrium of a node in the first diagram is represented by a closed polygon in the second diagram and vice versa (Fig. 2). Graphic statics is based on this principle [5].

2.2. Assumptions

The proposed method produces funicular (compression-only) solutions for loading conditions where all loads are applied in the same direction, as is the case for gravitational loading. Since the solutions are compression-only this also means that the vaults can never curl back onto themselves, which would demand some elements to go into tension. The resulting three-dimensional networks can represent load paths throughout a structure. There is no constraint on the length of the branches

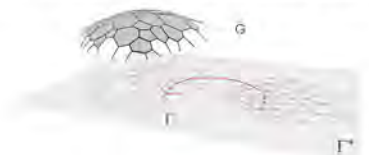


Figure 3. *Relationship between compression shell (G), its planar projection (primal grid Γ) and the reciprocal (funicular dual grid Γ*) to determine equilibrium.*

JOURNAL OF THE INTERNATIONAL ASSOCIATION FOR SHELL AND SPATIAL STRUCTURES: J. IASS

Figure 7. *(a) Possible thrust values for this given vault range from 21% to 32% of its total weight. (b) 3-D web and rib action with the forces mainly spanning between the ribs as represented in the dual grid (c).*

realm of lower-bound analysis. Put simply, if a compression-only network can be found that fits within the boundaries of a vault, then the vault will stand in compression. Note that if we want a solution without any tension (hinges) in the section, we want the solution to lie within the middle third of the section (defined by equation 6). This is a powerful concept for understanding the stability and proximity to collapse of such structures. Additional reading on this topic can be found in Heyman [10], O'Dwyer [2], Boothby [11] and Block et al. [1].

The method is particularly appropriate for historic masonry vaults because their self-weight is the dominant load. The range of possible equilibrium states, bounded by a minimum and maximum thrust, can be produced (Fig. 7a). We are interested in this range of thrust values because it provides the most useful characterization of the structural behavior of the vault. The minimum (or passive) thrust state represents the least amount this vault will push sideways onto its neighboring elements, as a function of its self-weight and shape. The maximum (or active) state of thrust on the other hand represents the largest horizontal force this vault can provide. So, this value demonstrates how much horizontal force this vault can safely take from its neighboring elements.

Figure 7b shows a solution with three-dimensional web action. The distribution of the horizontal components in the network is represented in its dual grid (Fig. 7c). More detailed applications of this methodology to masonry vaults can be found in [12].

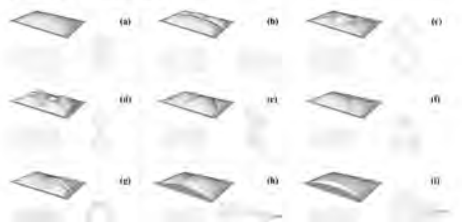
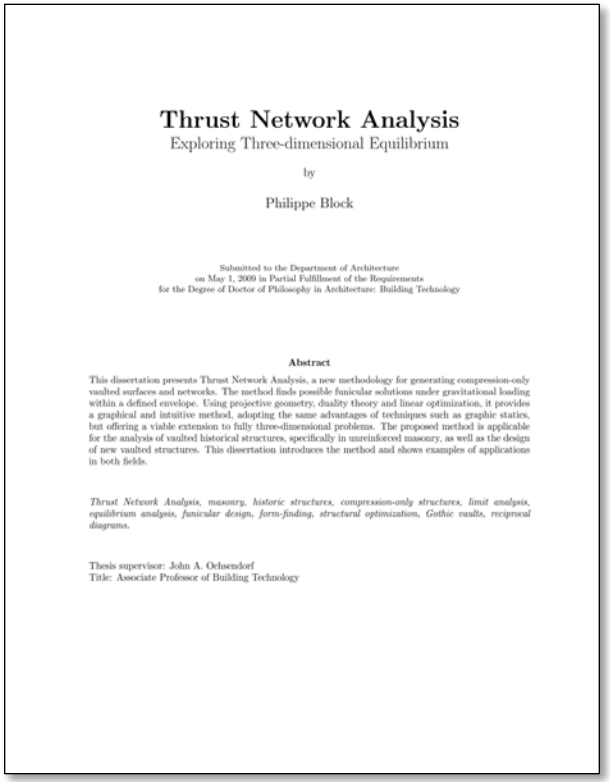


Figure 8. *A series of examples, starting from a regular rectangular grid, showing the relationship between the primal (red), dual grid and the corresponding solution.*



Thrust Network Analysis

Exploring Three-dimensional Equilibrium

by

Philippe Block

Submitted to the Department of Architecture
on May 1, 2009 in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Architecture: Building Technology

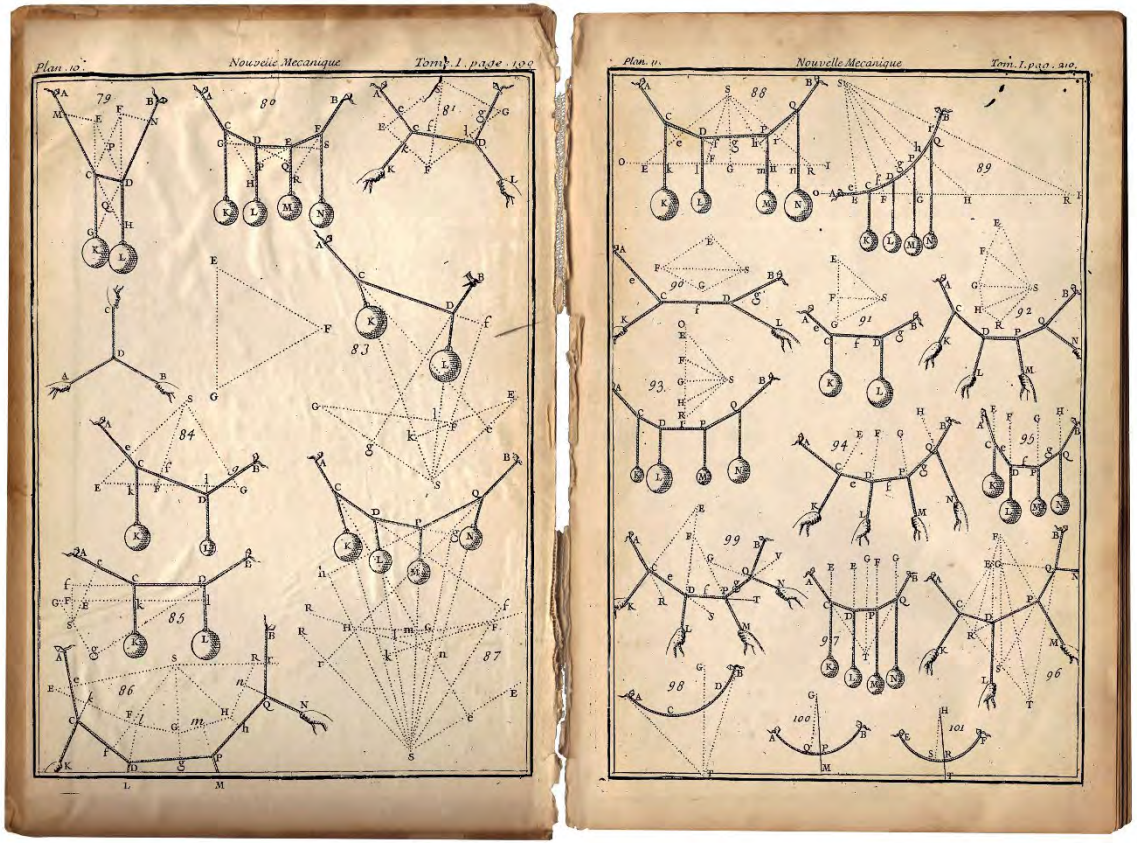
Abstract

This dissertation presents Thrust Network Analysis, a new methodology for generating compression-only vaulted surfaces and networks. The method finds possible funicular solutions under gravitational loading within a defined envelope. Using projective geometry, duality theory and linear optimization, it provides a graphical and intuitive method, adopting the same advantages of techniques such as graphic statics, but offering a viable extension to fully three-dimensional problems. The proposed method is applicable for the analysis of vaulted historical structures, specifically in unreinforced masonry, as well as the design of new vaulted structures. This dissertation introduces the method and shows examples of applications in both fields.

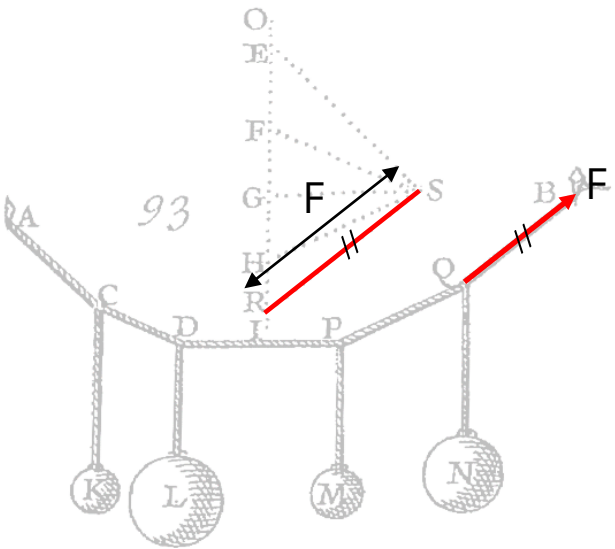
Thrust Network Analysis, masonry, historic structures, compression-only structures, limit analysis, equilibrium analysis, funicular design, form-finding, structural optimization, Gothic vaults, reciprocal diagrams.

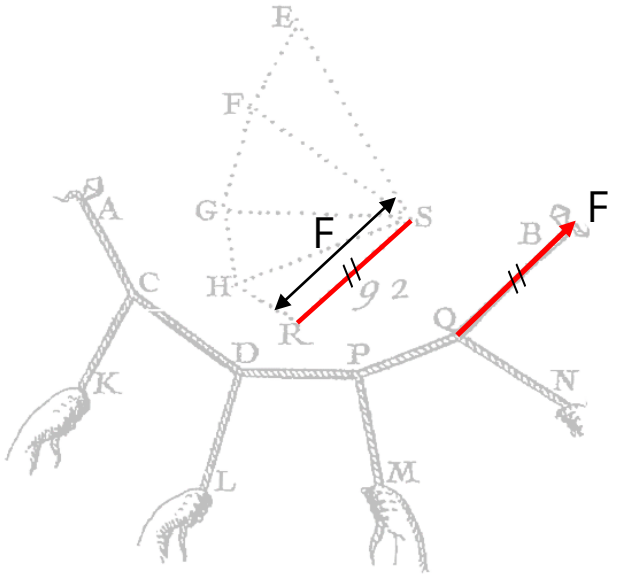
Thesis supervisor: John A. Ochsendorf
Title: Associate Professor of Building Technology

"Thrust Network Analysis" | Block (2009)



"Nouvelle mécanique ou Statique" | Varignon (1725)





Varignon (1725)

Stress

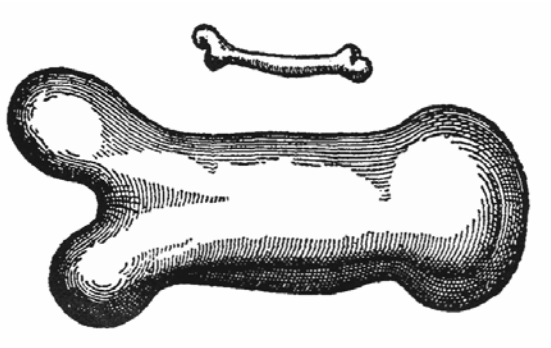
↓

Material failure

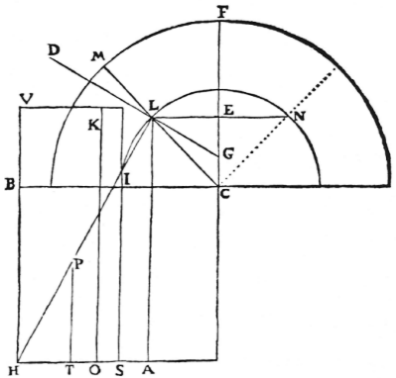
Stability

↓

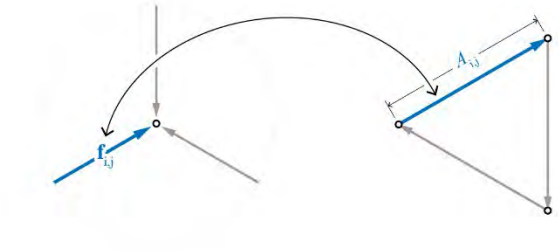
Geometry



Galilei (1638)

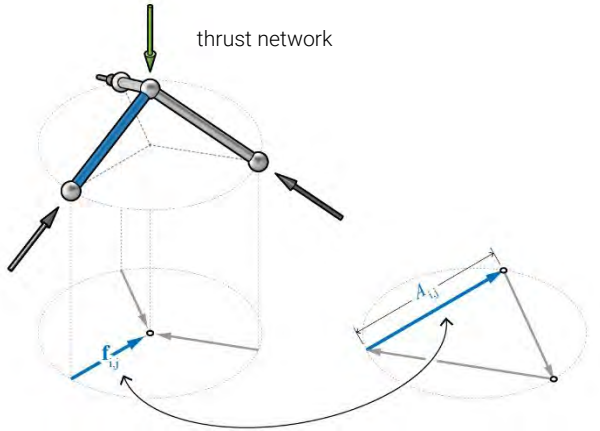


de Lahire (1695)



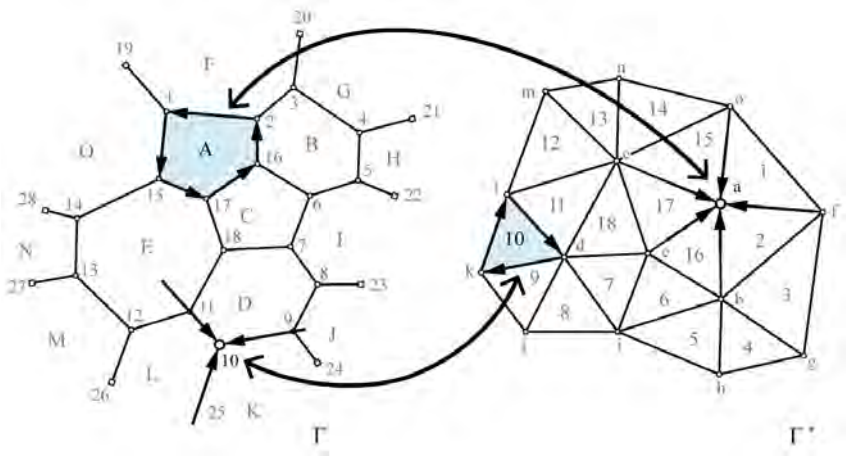
form diagram force diagram

2D

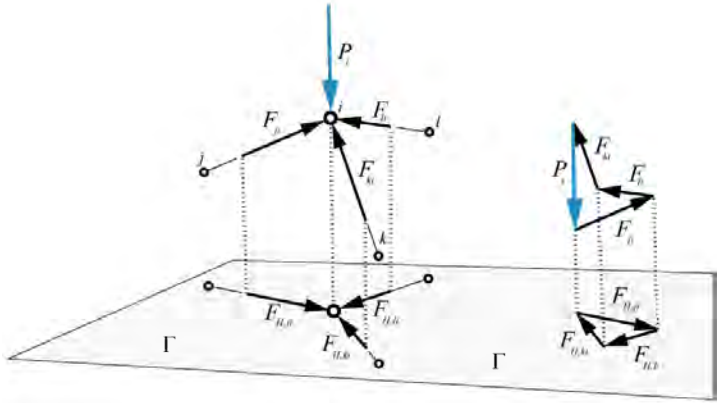


form diagram force diagram

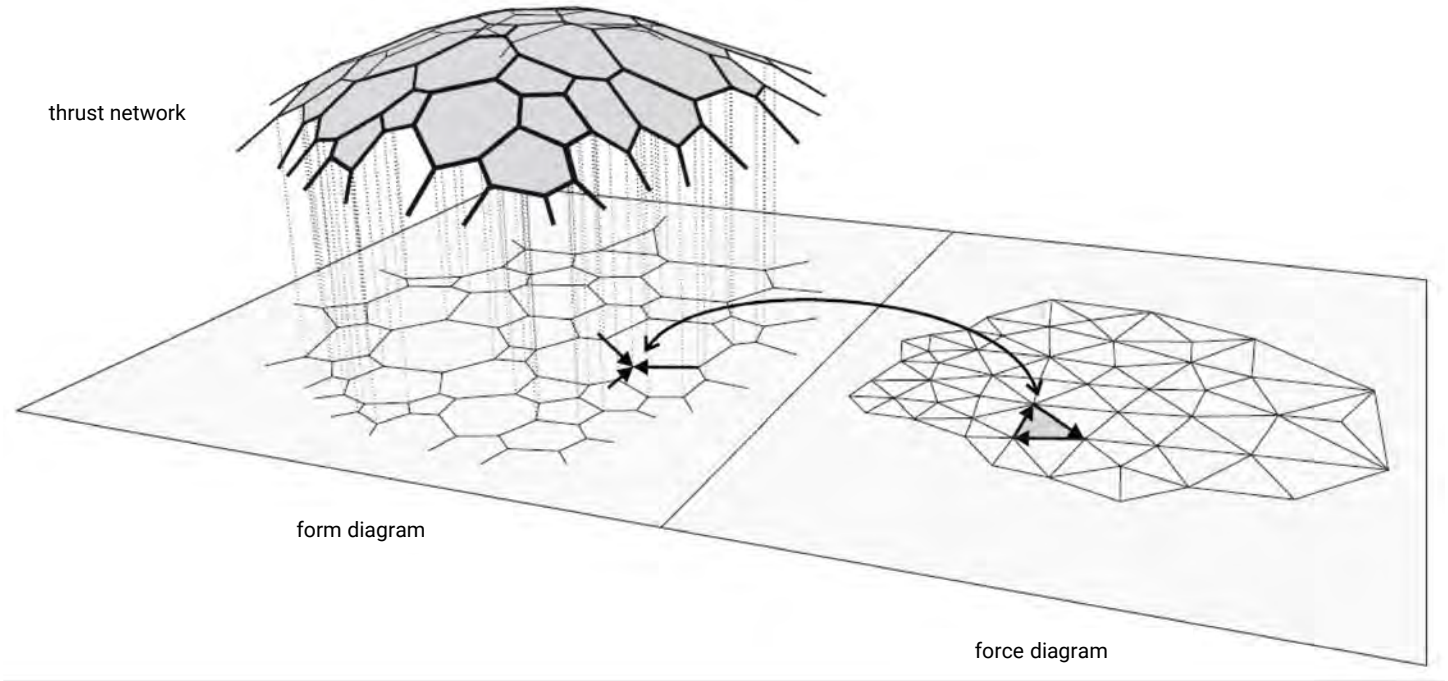
2.5D



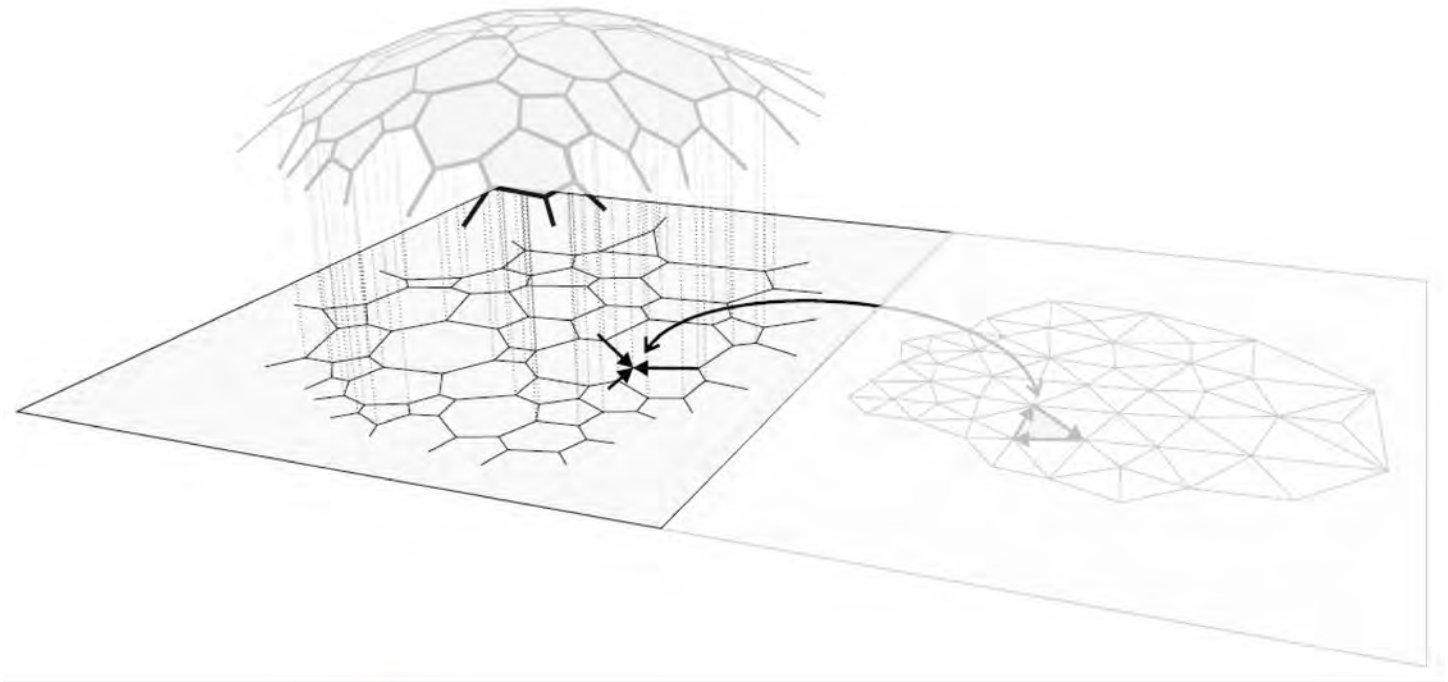
Form Diagram



Force Diagram

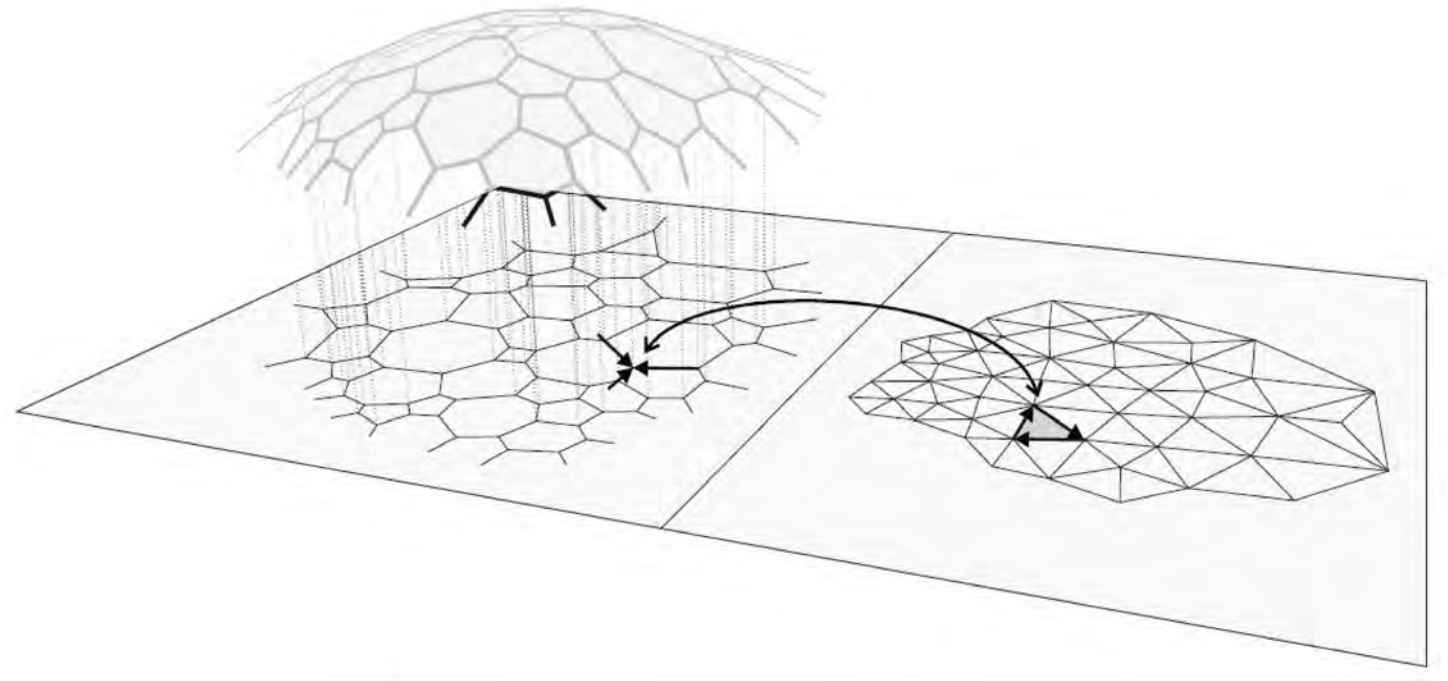


"Thrust Network Analysis" | Block (2007)



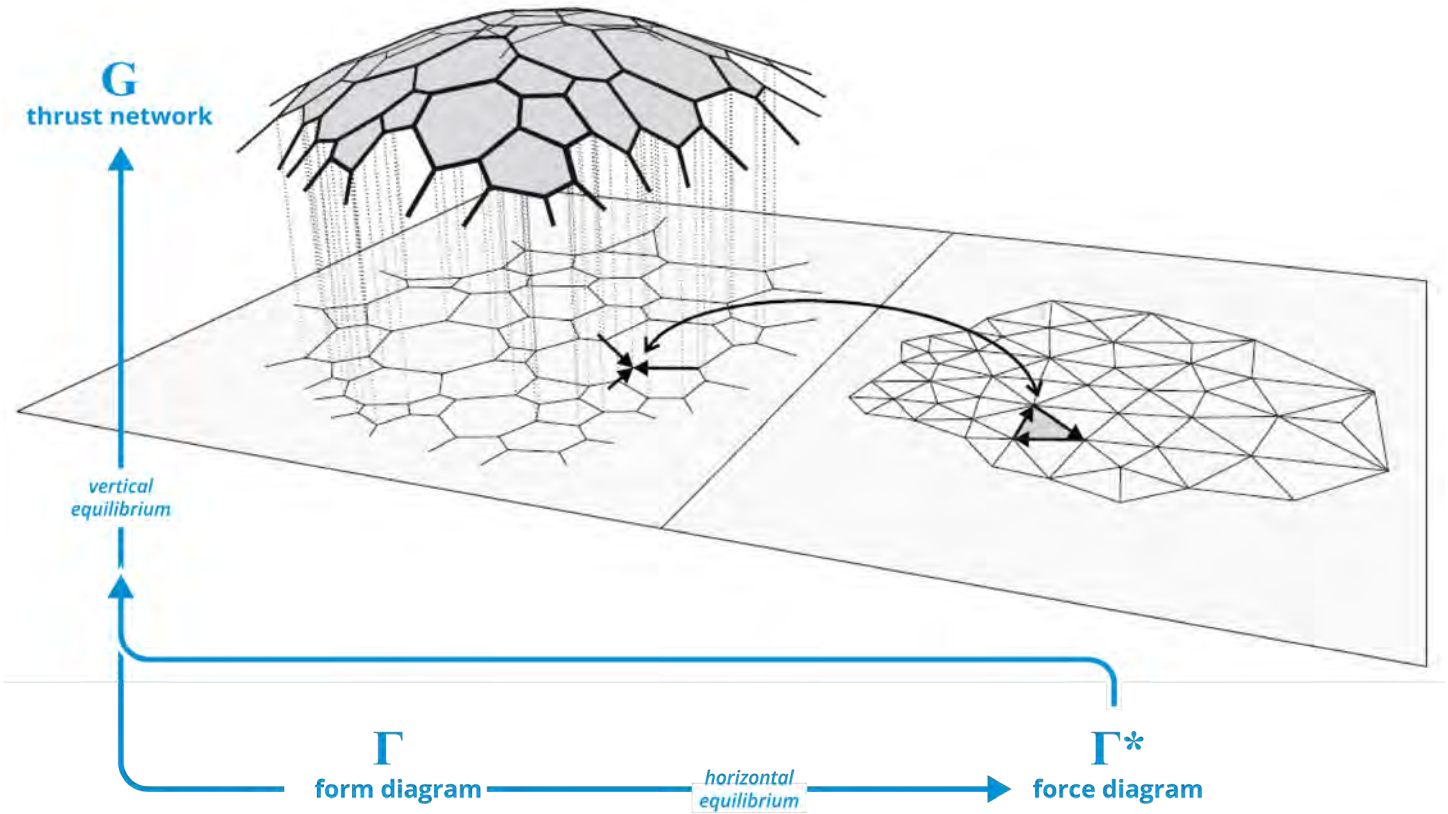
Γ
form diagram

"Thrust Network Analysis" | Block (2007)

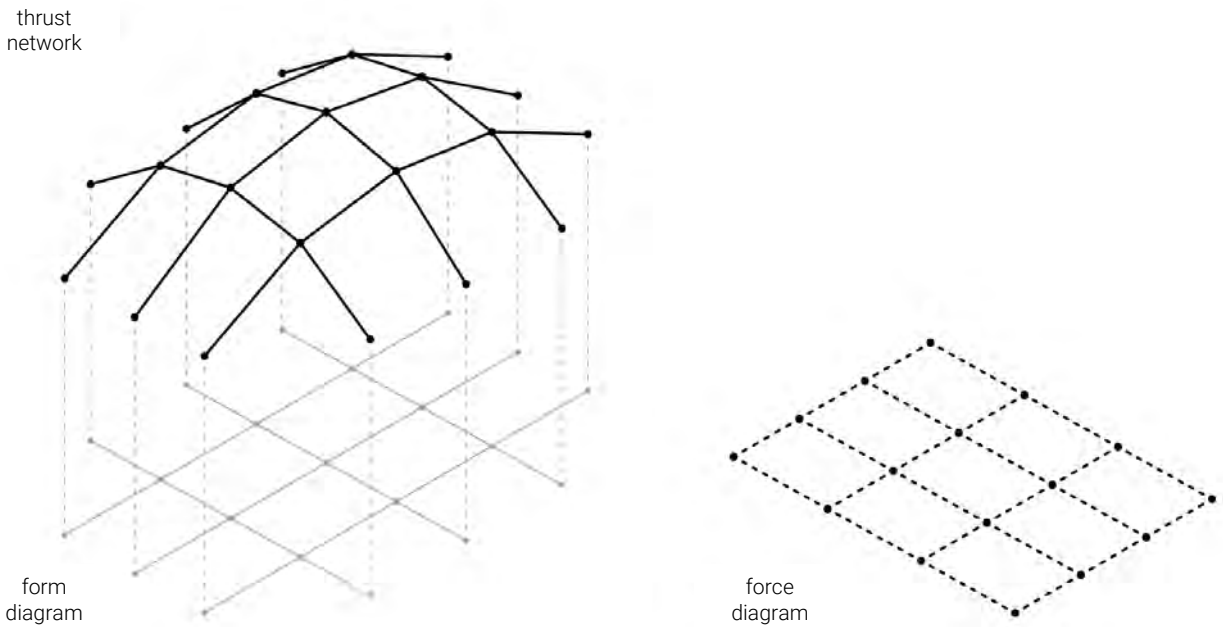


Γ form diagram — *horizontal equilibrium* → Γ^* force diagram

"Thrust Network Analysis" | Block (2007)



"Thrust Network Analysis" | Block (2007)

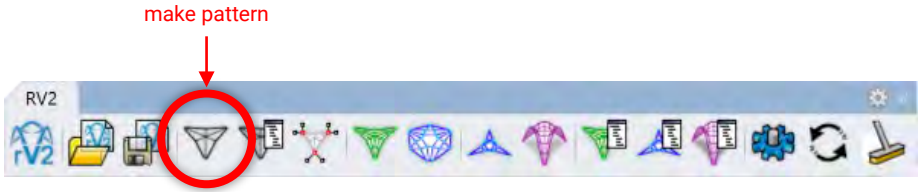


"Funicular Shell Design" | Rippmann (2018)

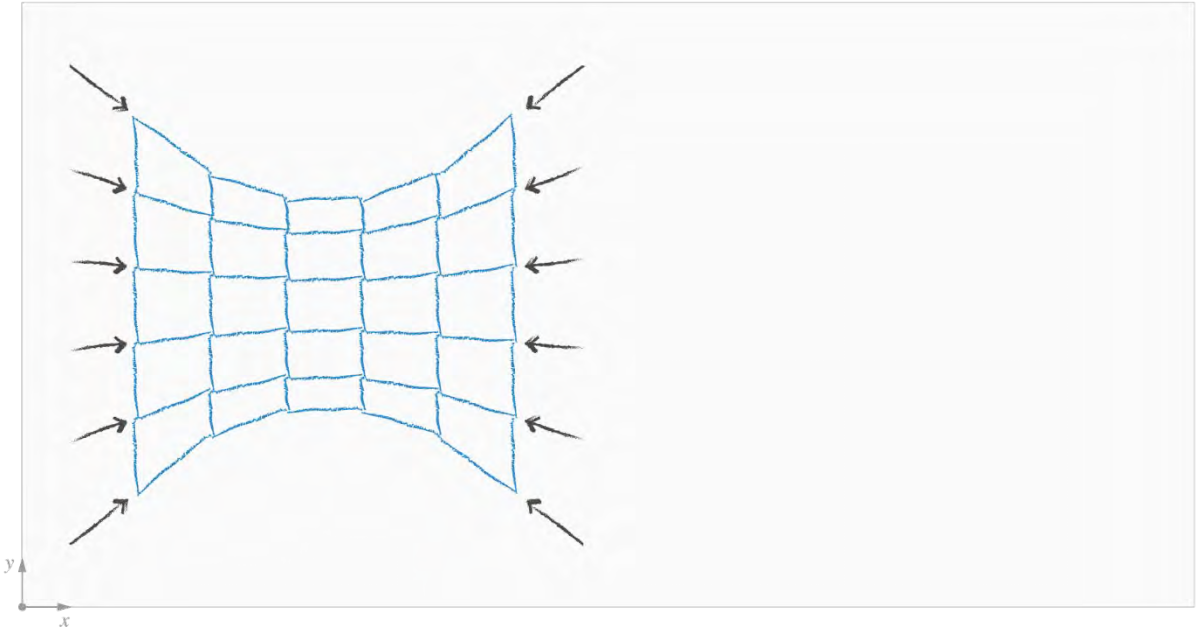
RV2 workflow

TNA steps

1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



1. Pattern

2. Boundary conditions

3. Form diagram

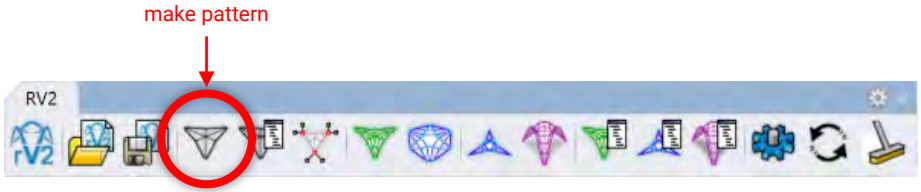
4. Dual diagram

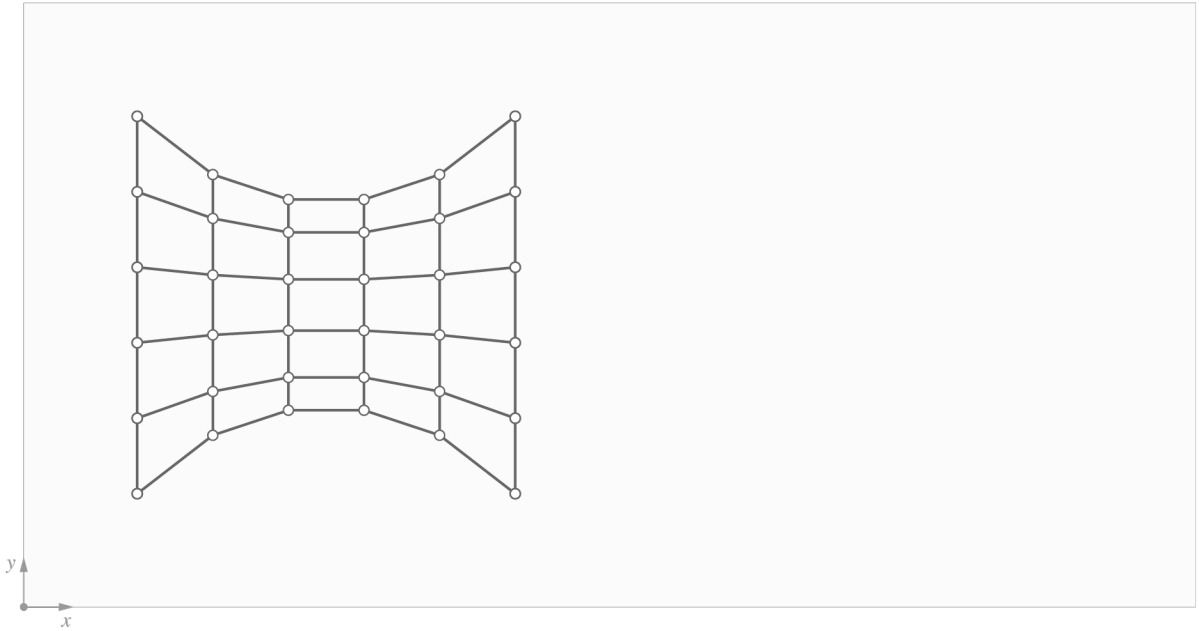
5. Horizontal equilibrium

6. Force diagram

7. Vertical equilibrium

8. Thrust diagram





1. Pattern

2. Boundary conditions

3. Form diagram

4. Dual diagram

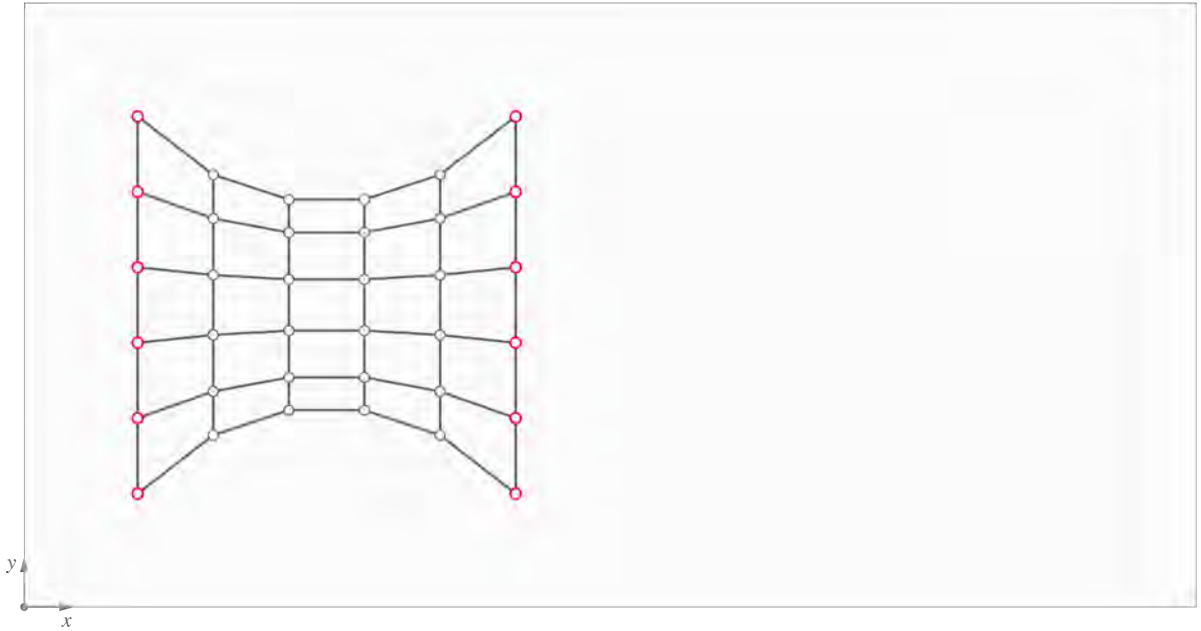
5. Horizontal equilibrium

6. Force diagram

7. Vertical equilibrium

8. Thrust diagram

define boundary conditions



1. Pattern

2. Boundary conditions

3. Form diagram

4. Dual diagram

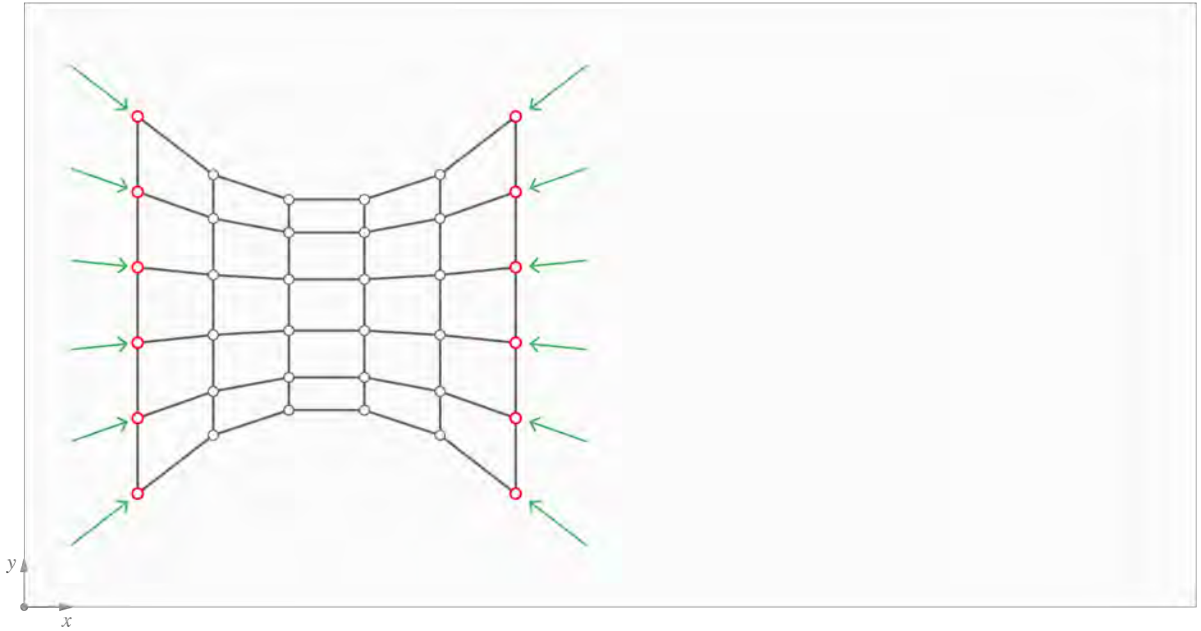
5. Horizontal equilibrium

6. Force diagram

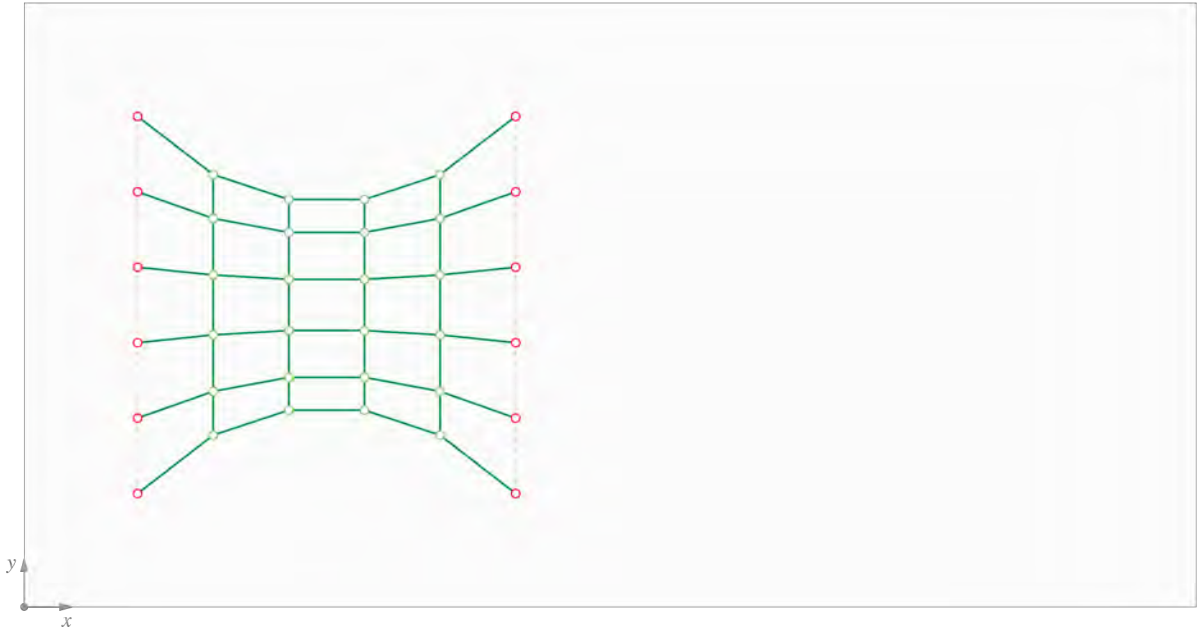
7. Vertical equilibrium

8. Thrust diagram

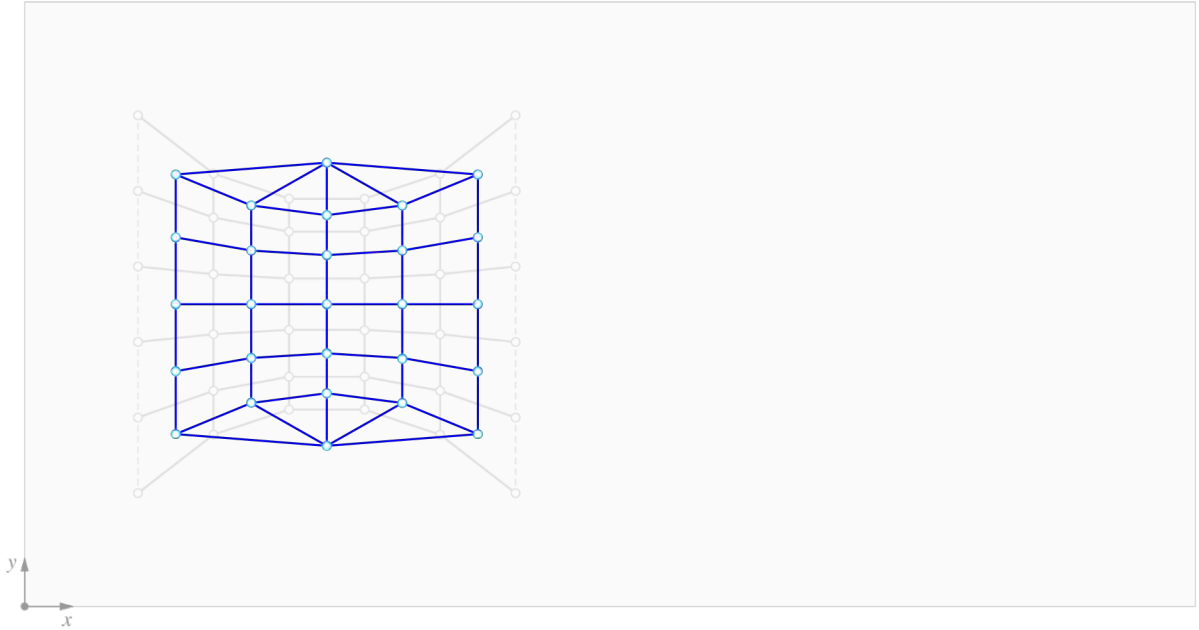
define boundary conditions



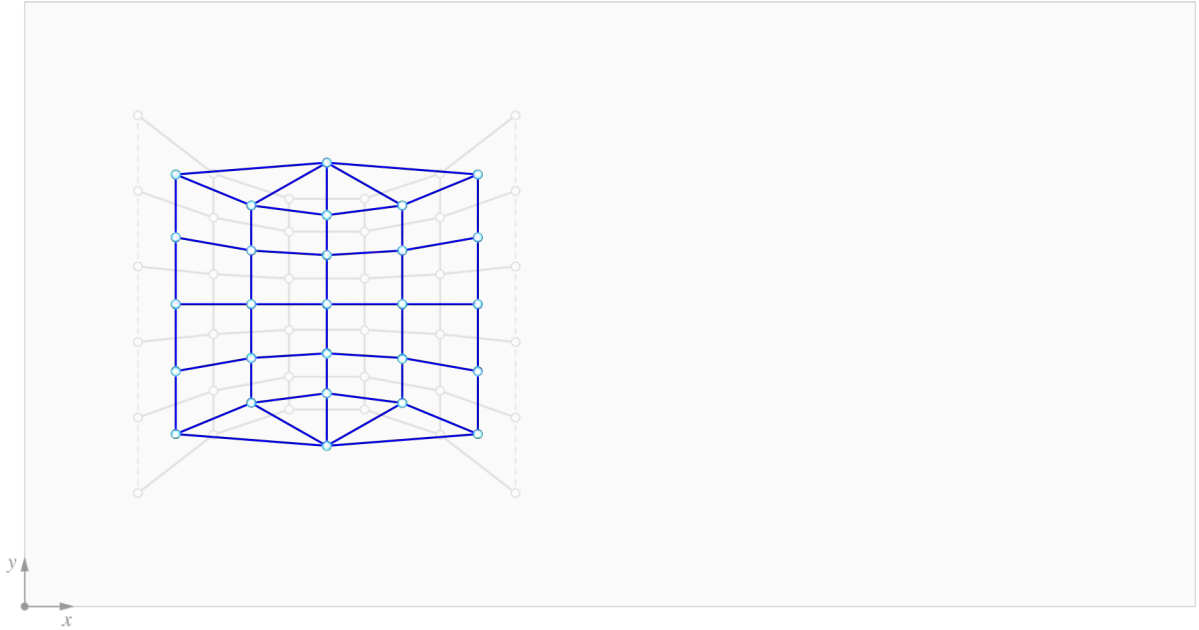
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



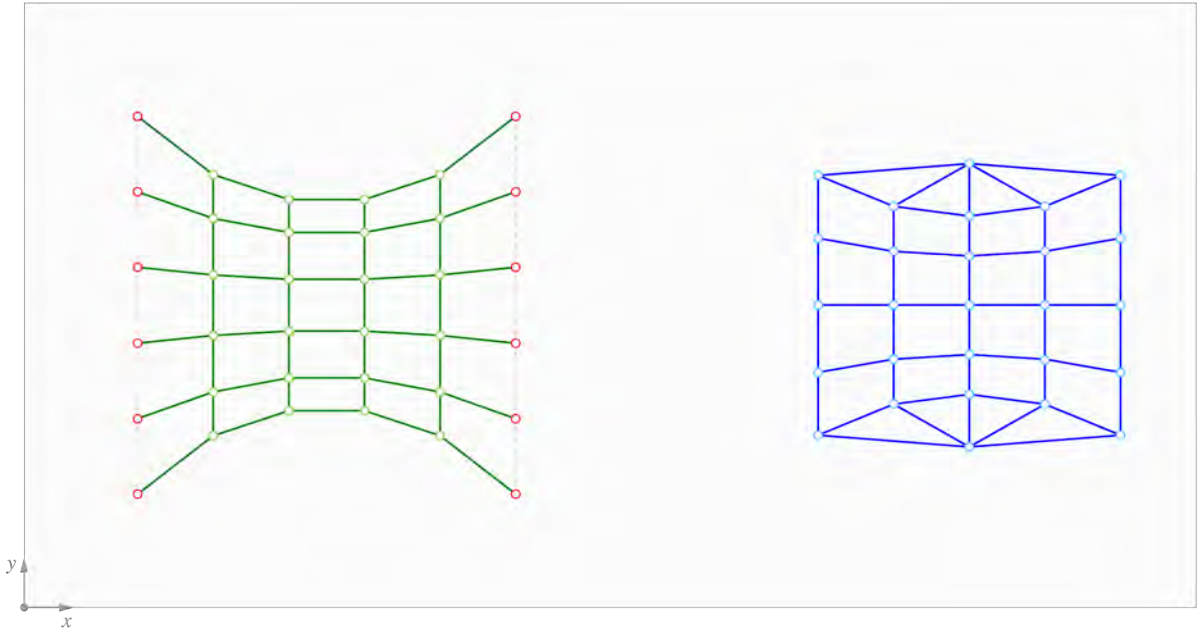
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

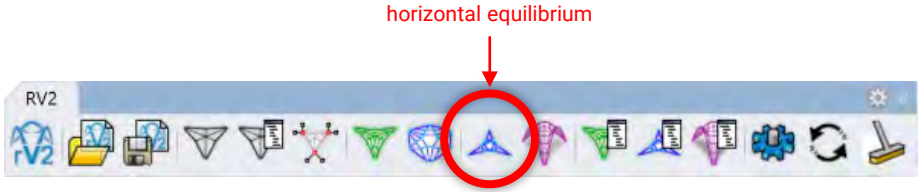


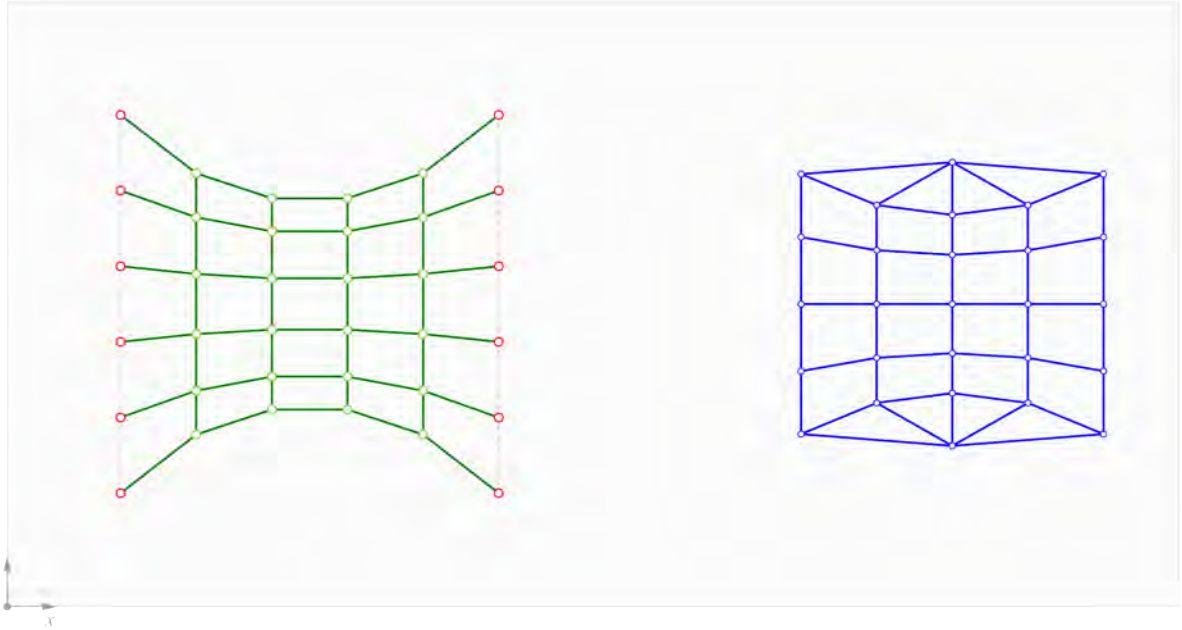
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



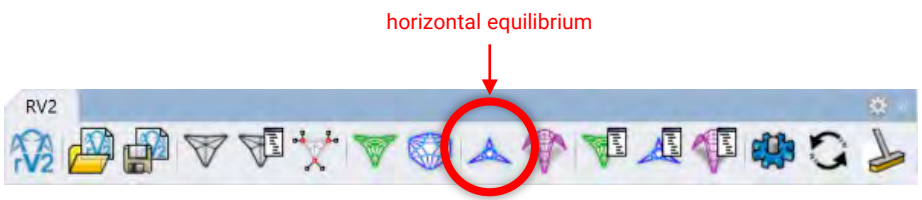
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

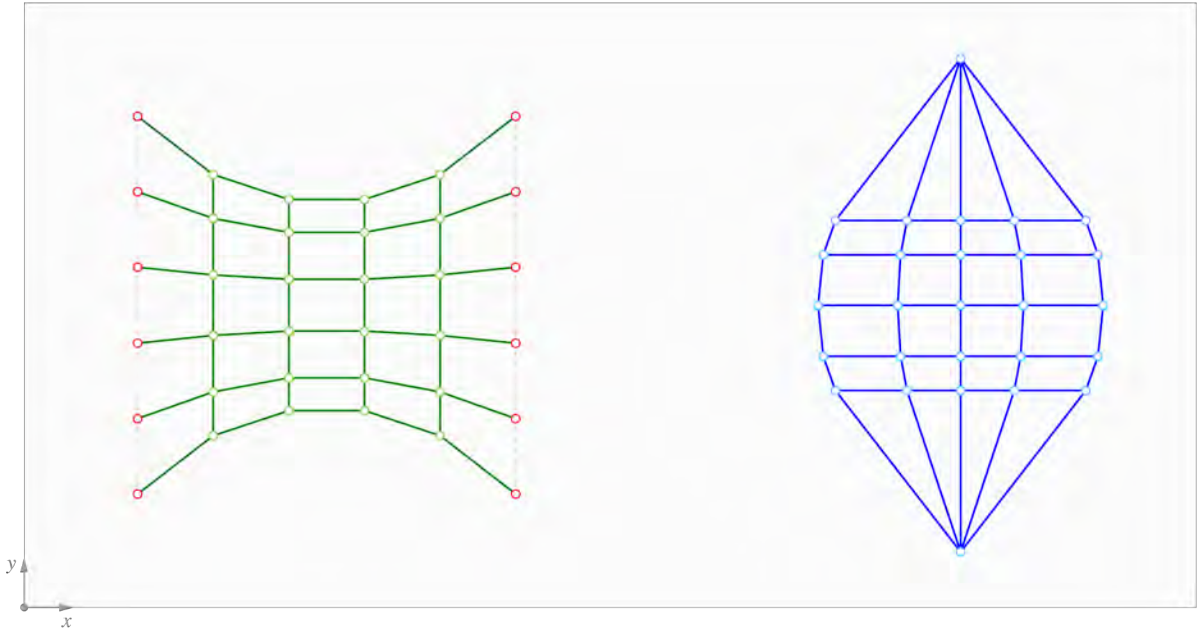
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



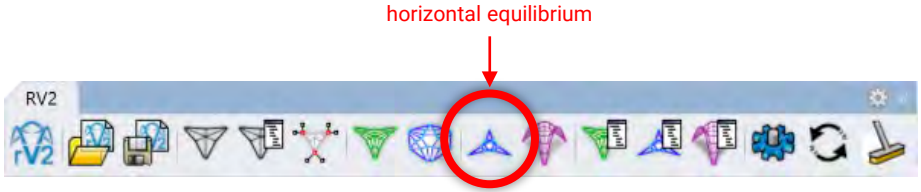


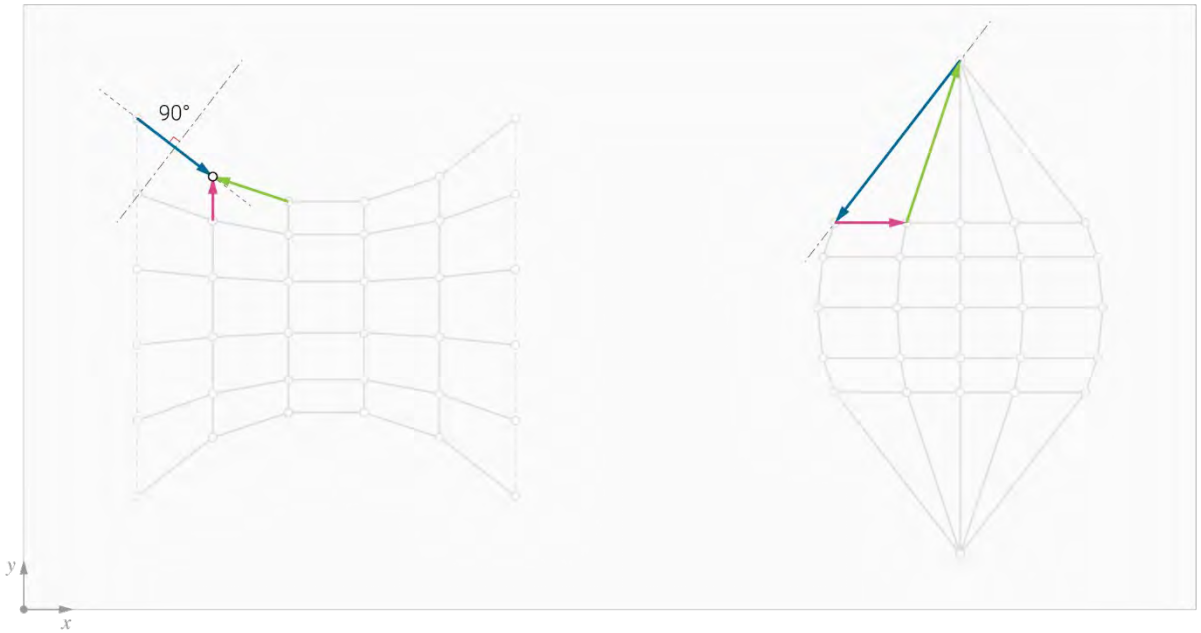
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



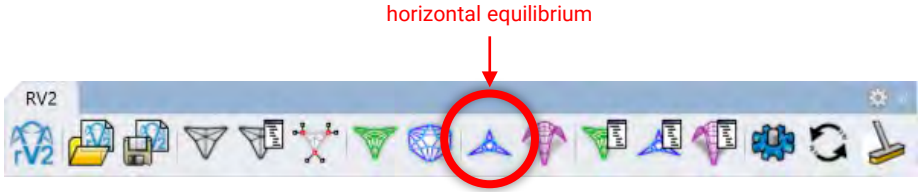


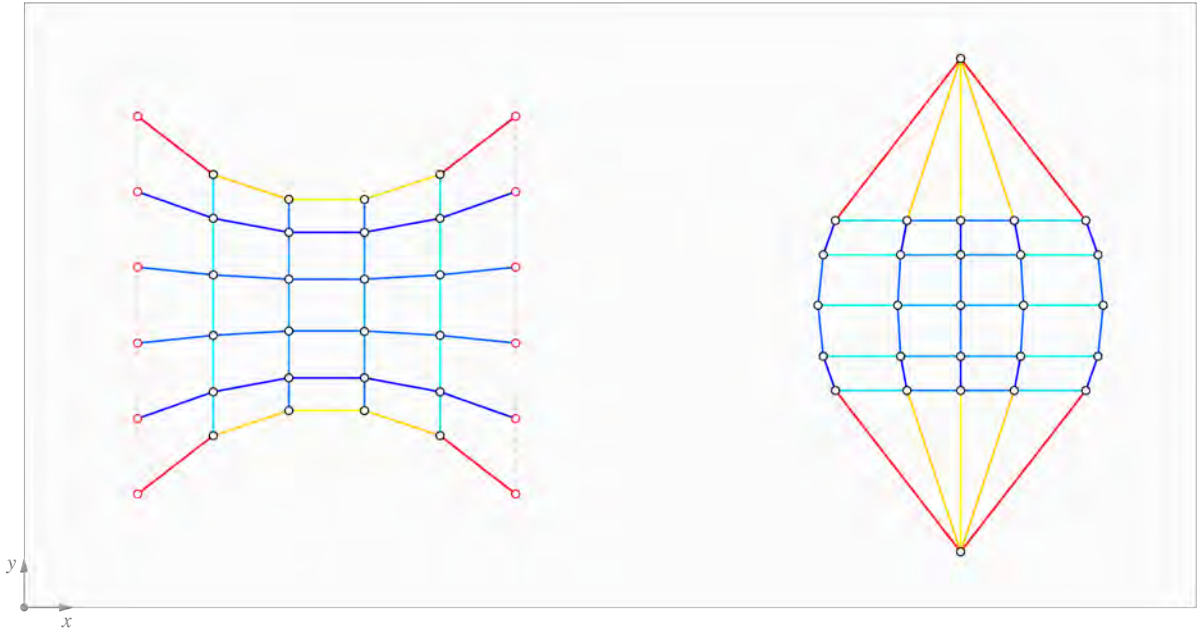
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



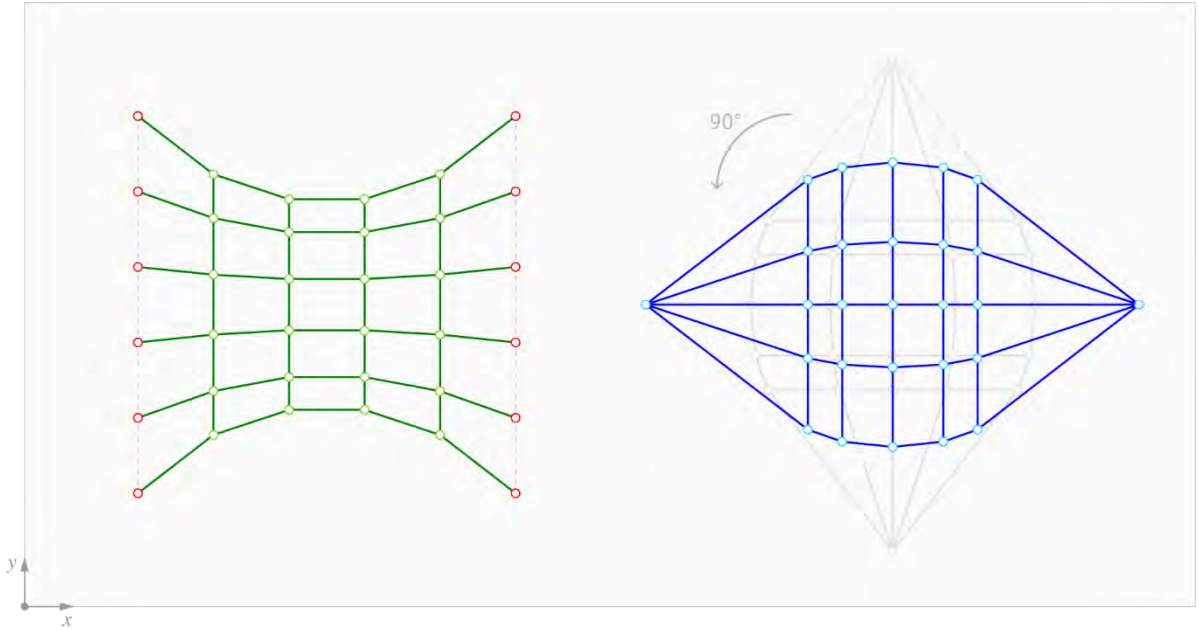


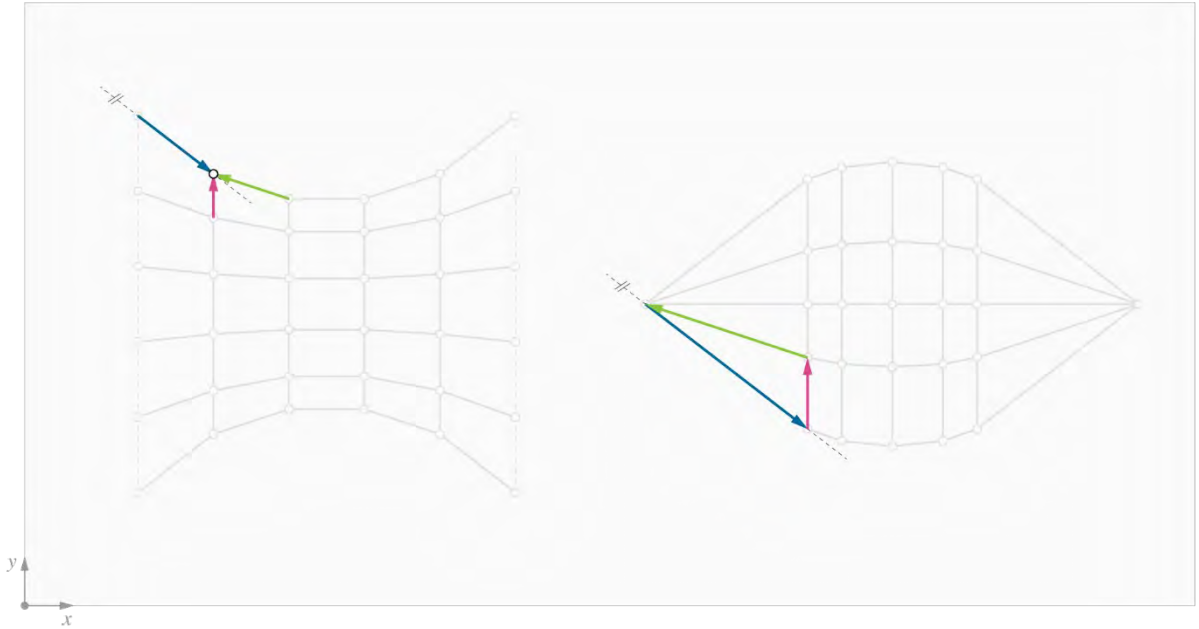
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

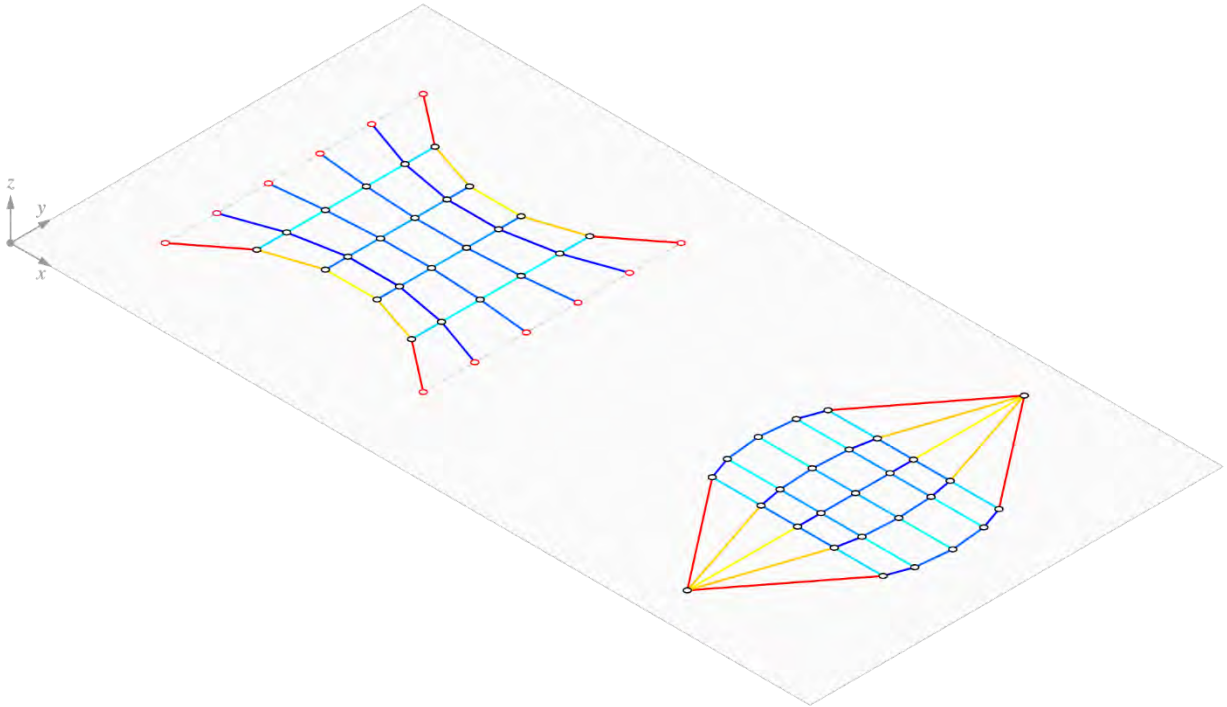




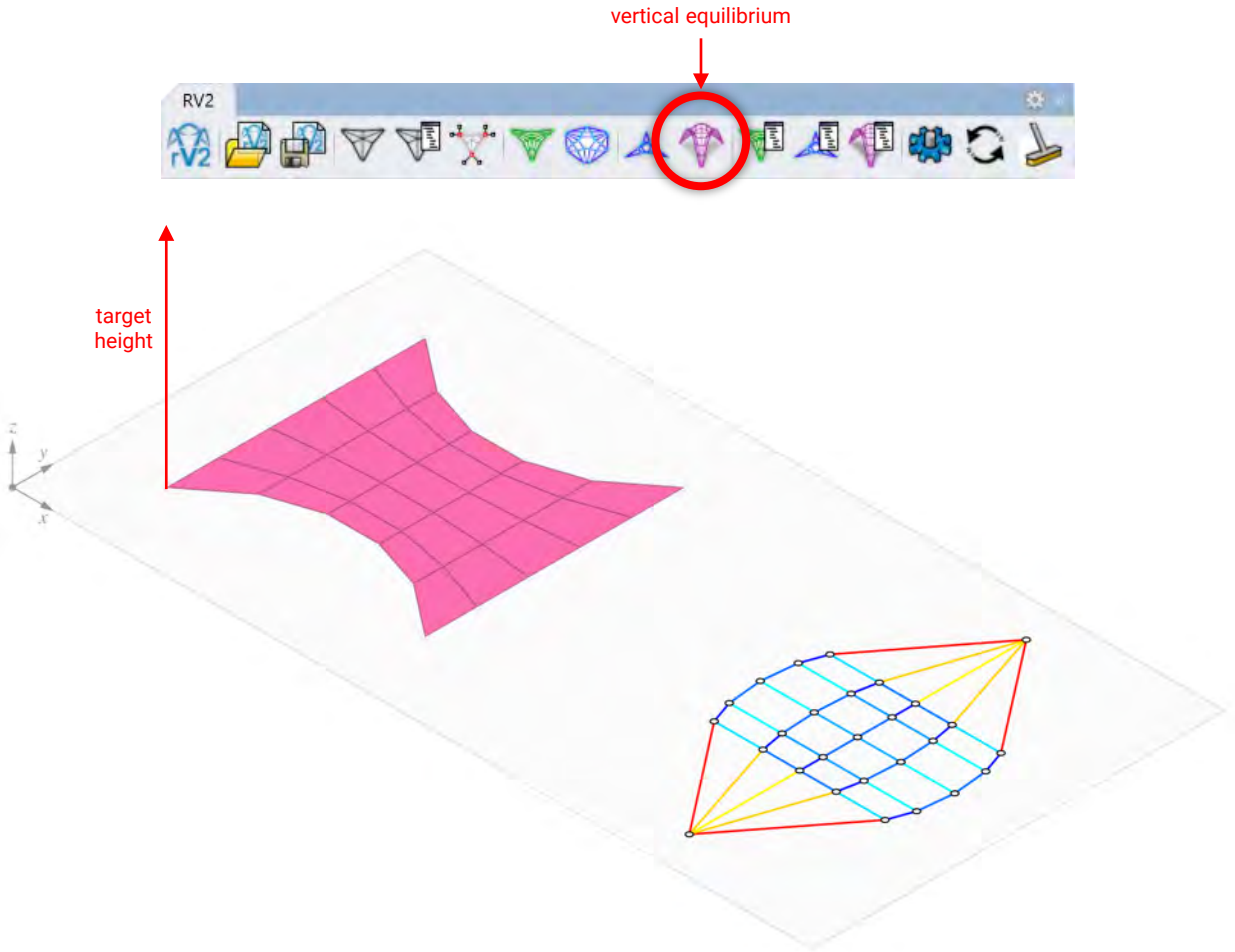
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



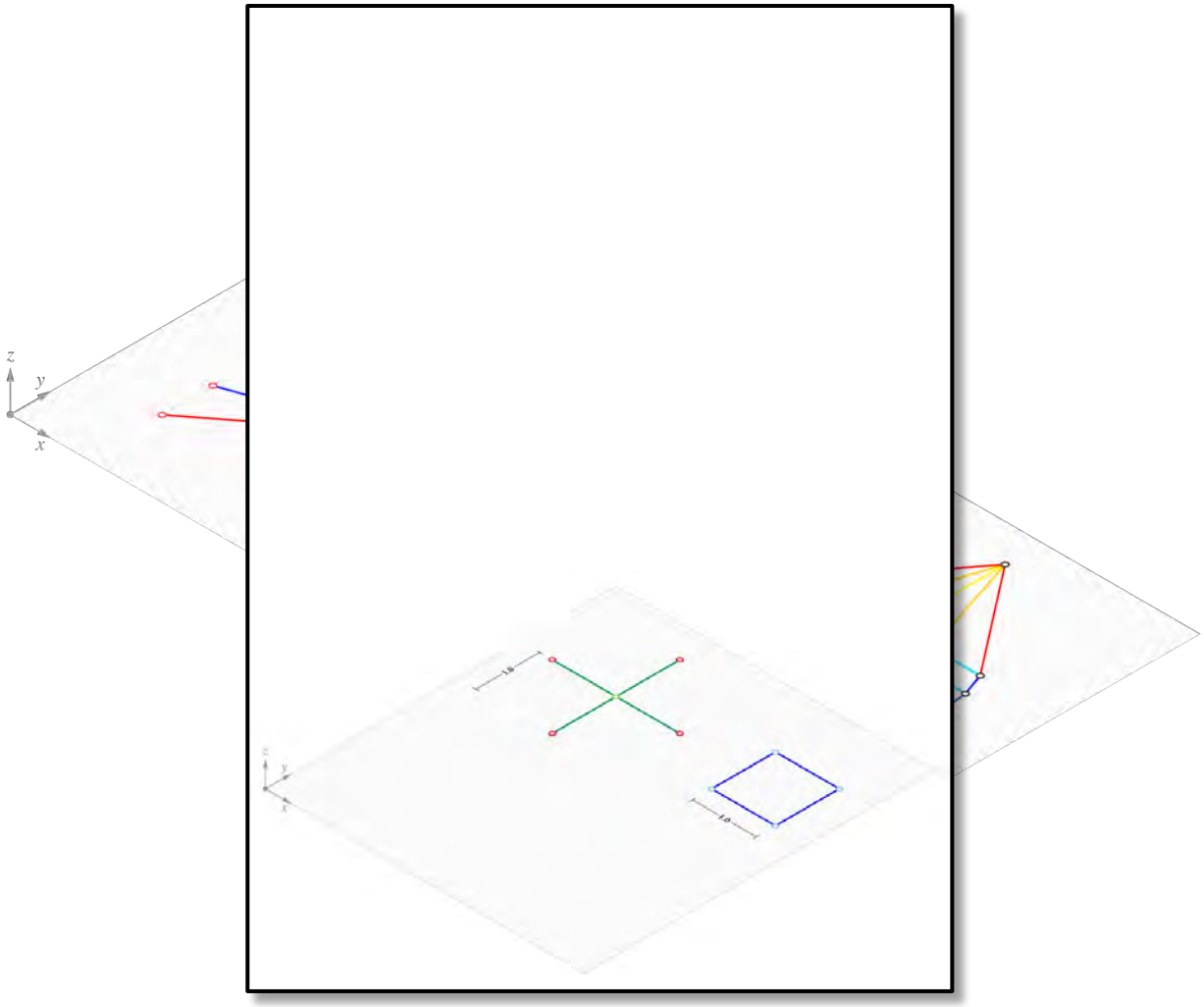




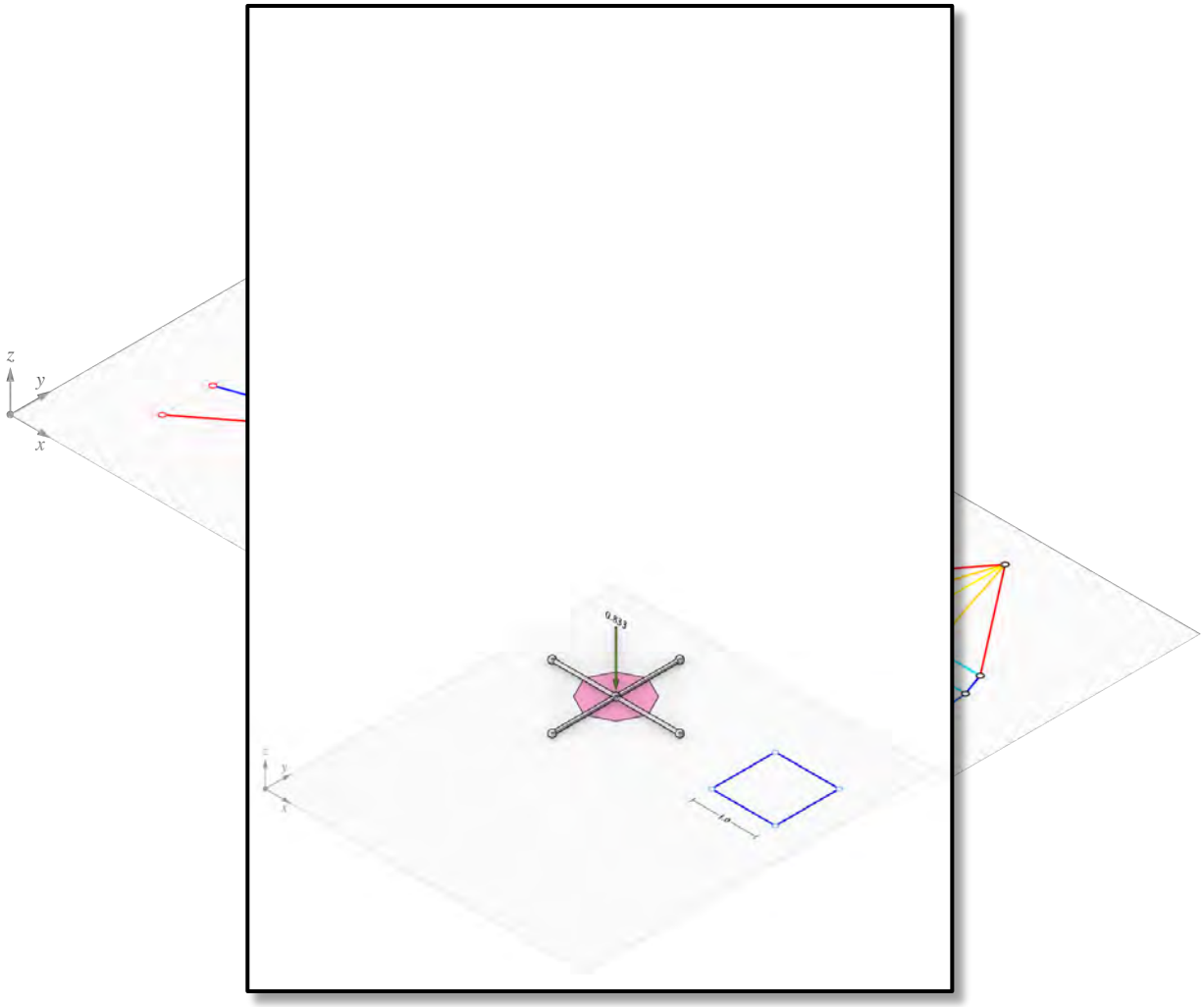
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



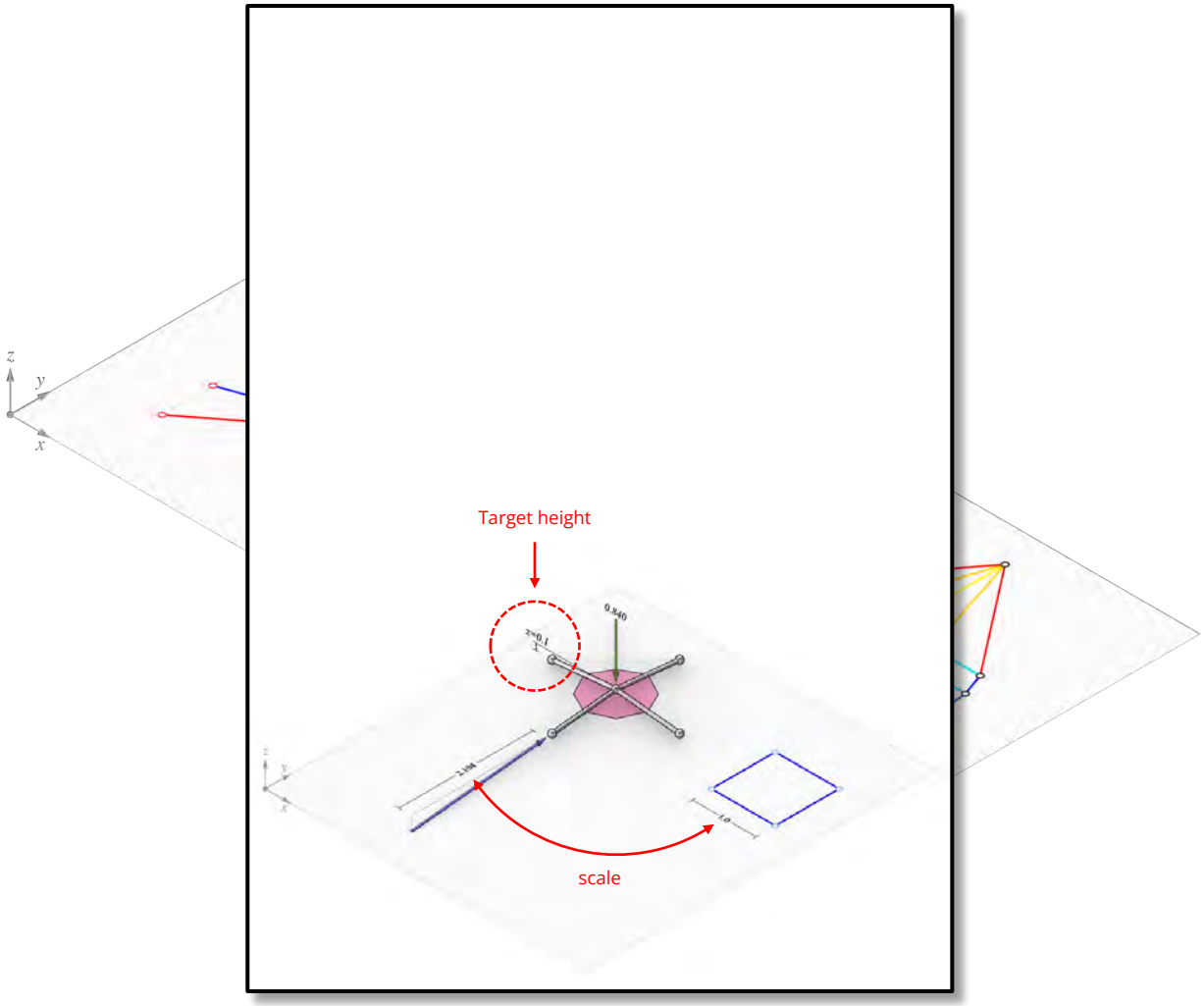
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



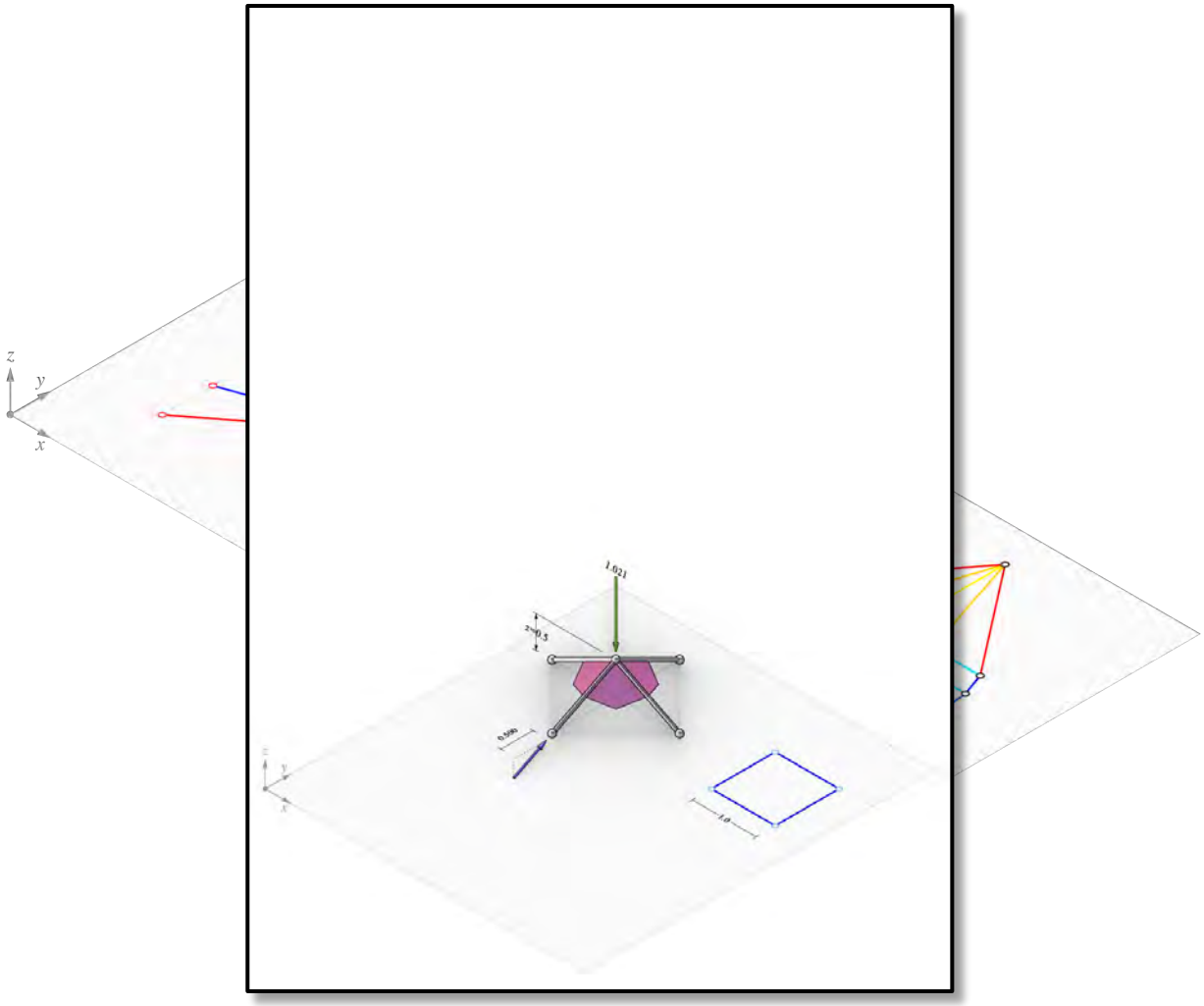
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



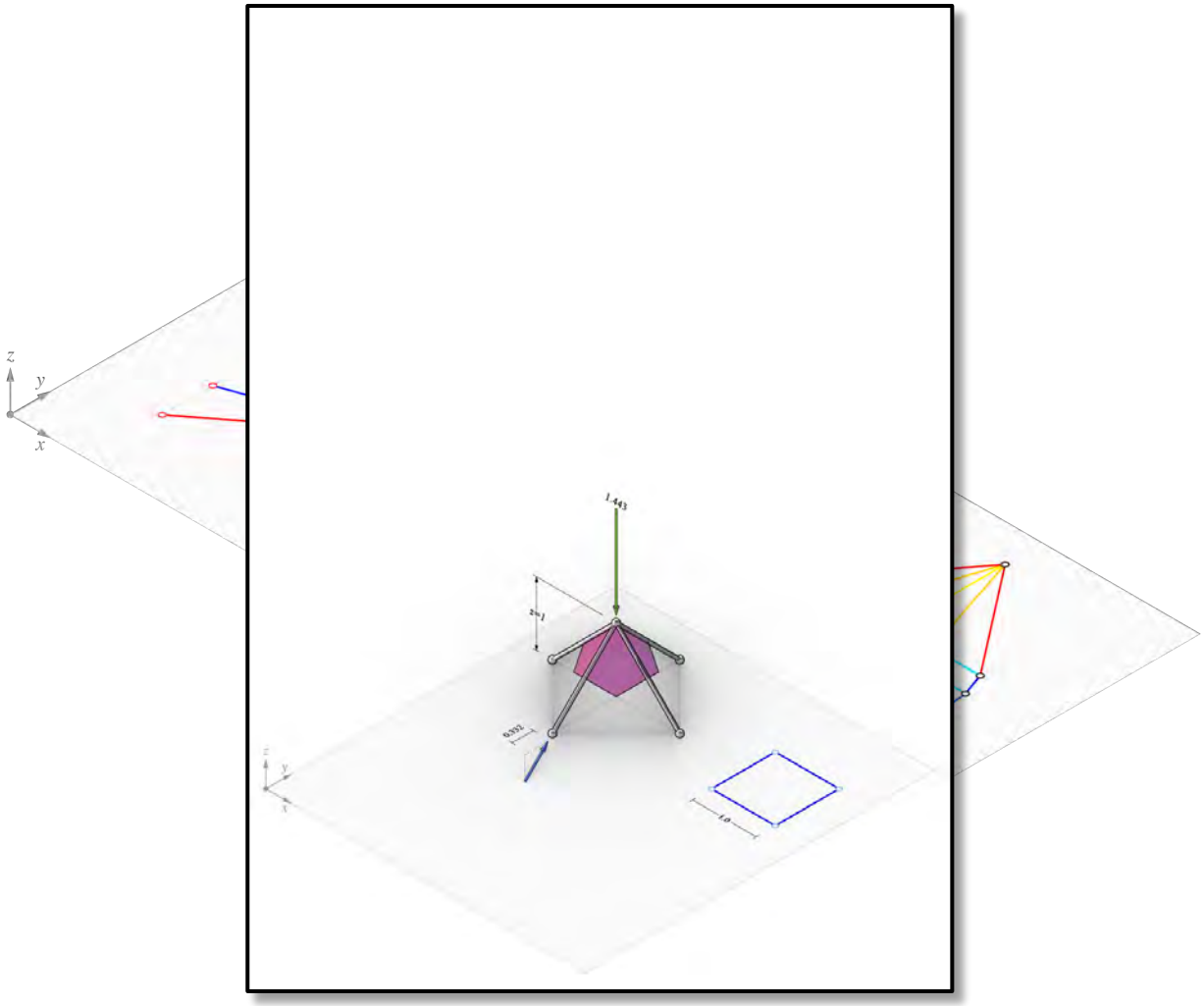
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



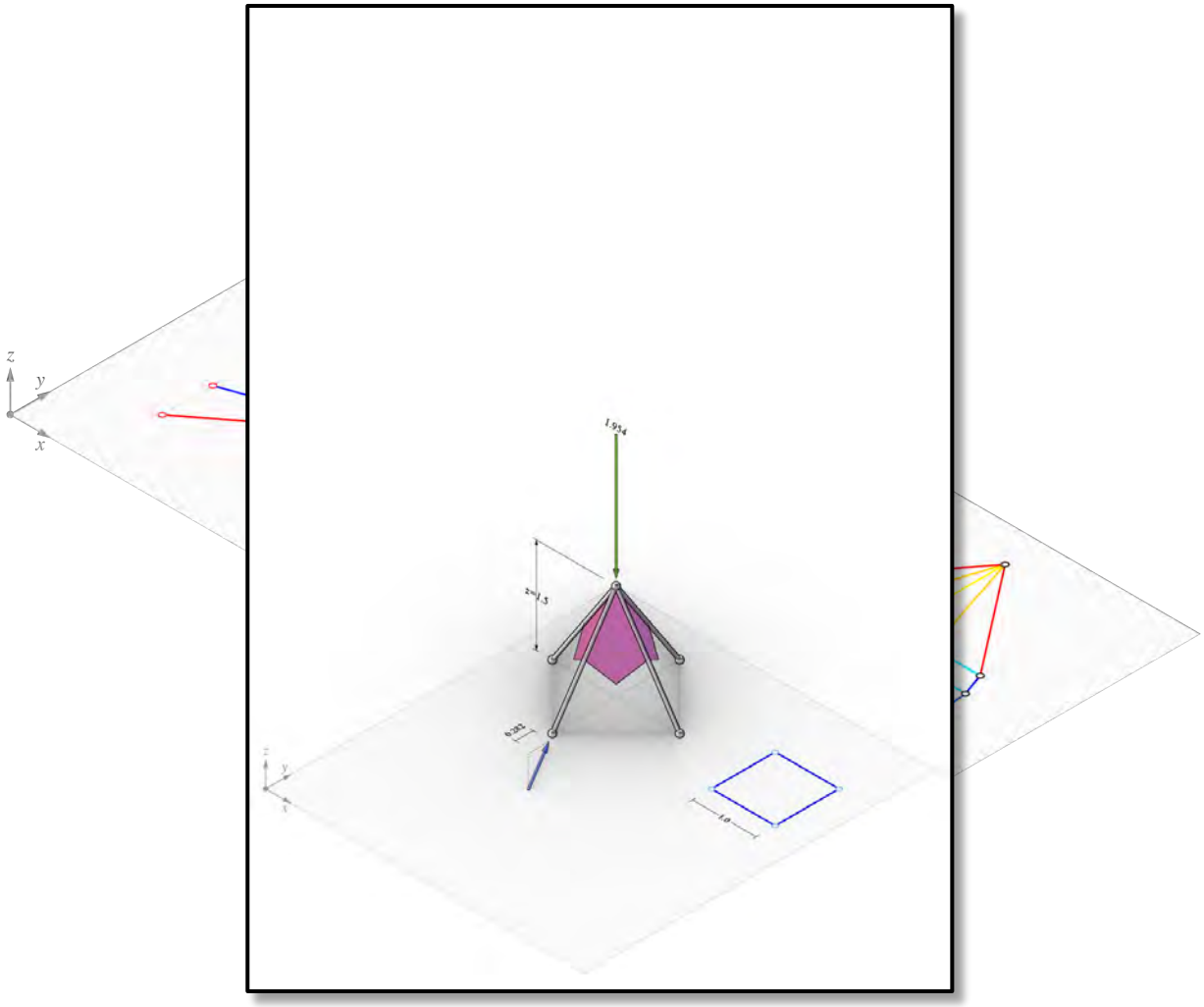
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



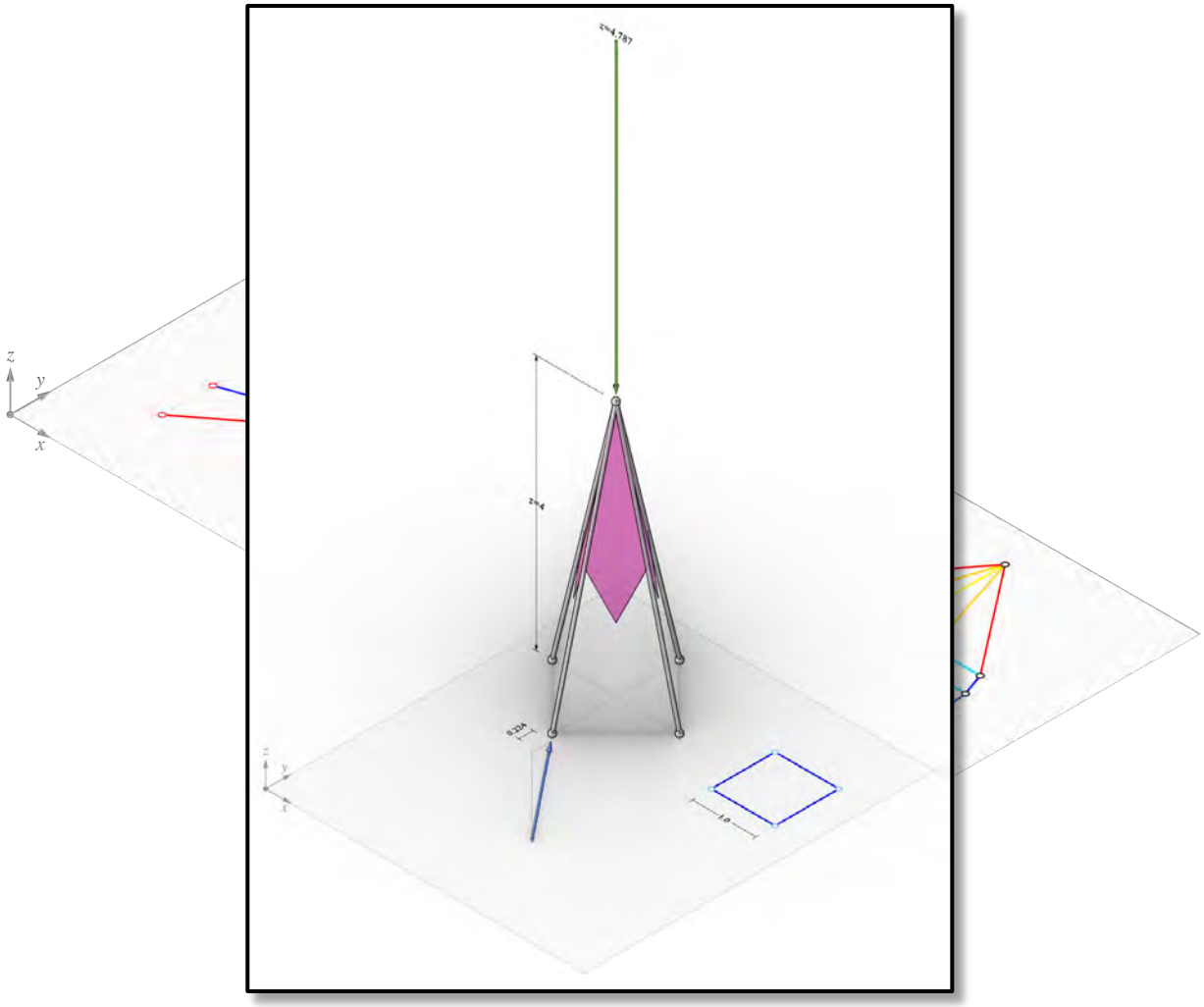
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



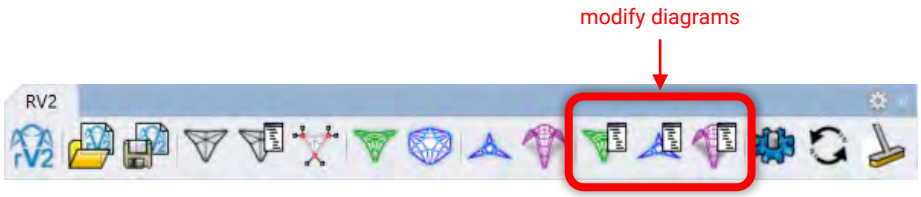
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

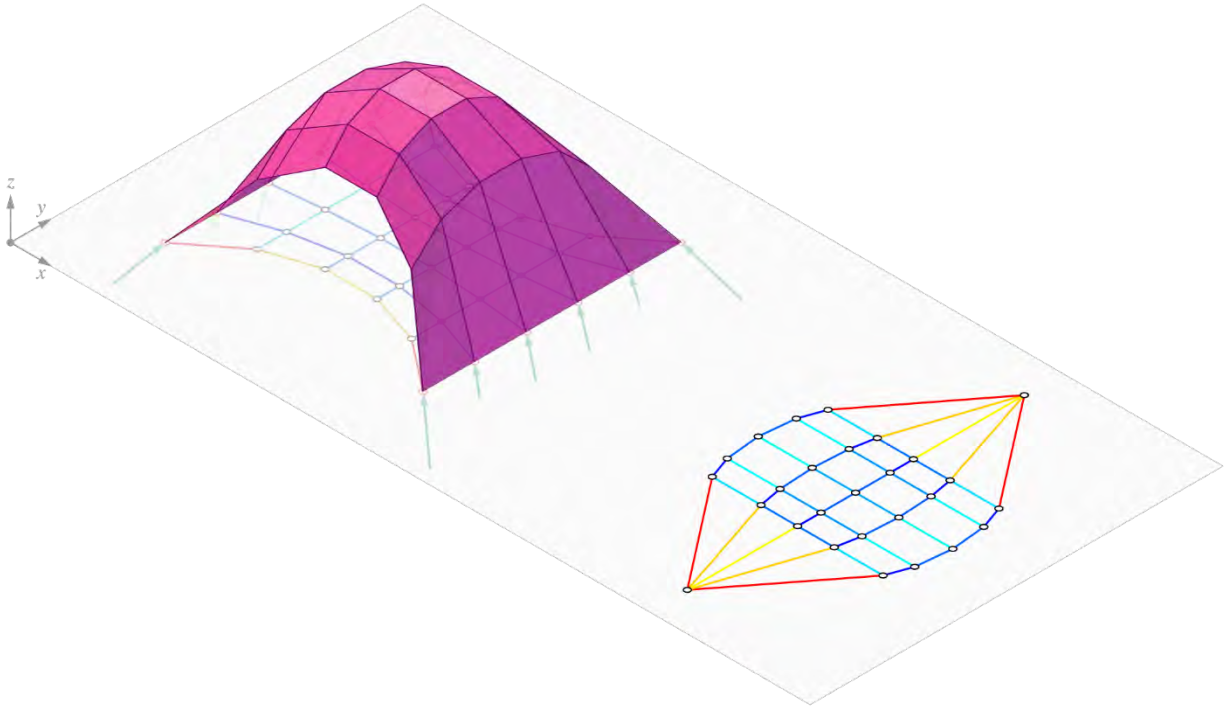


1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

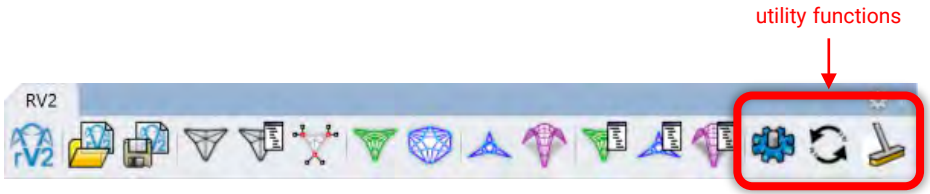


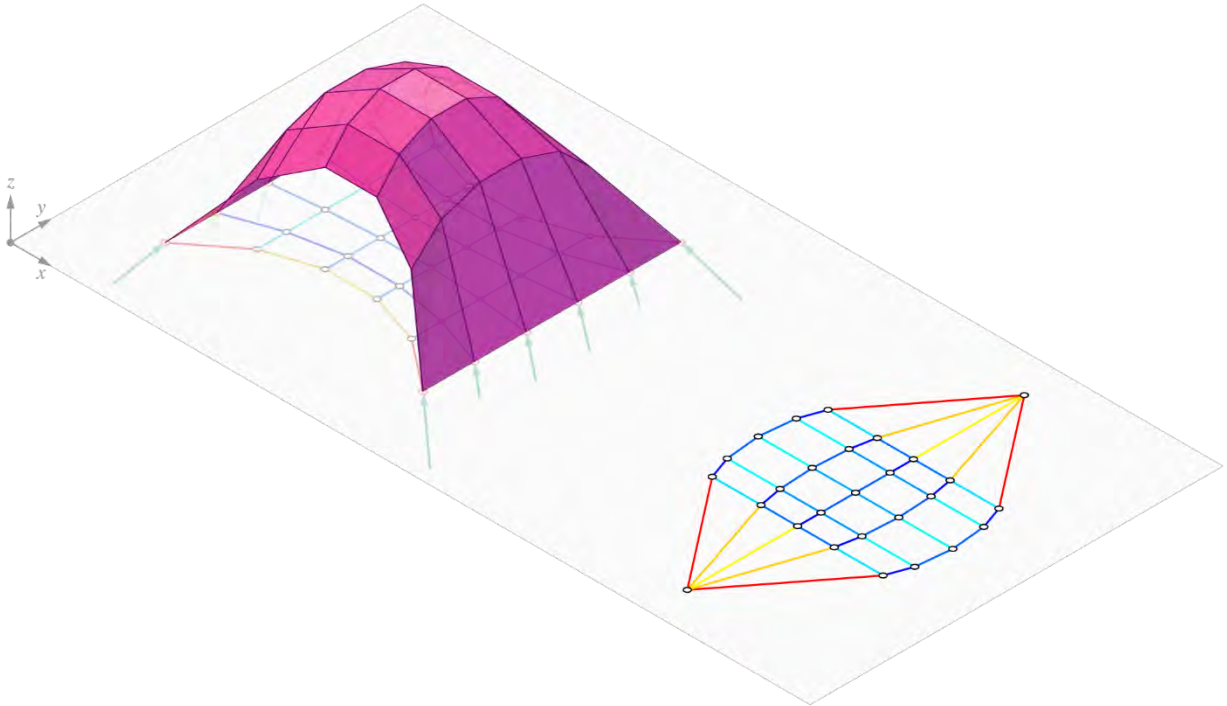
1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram

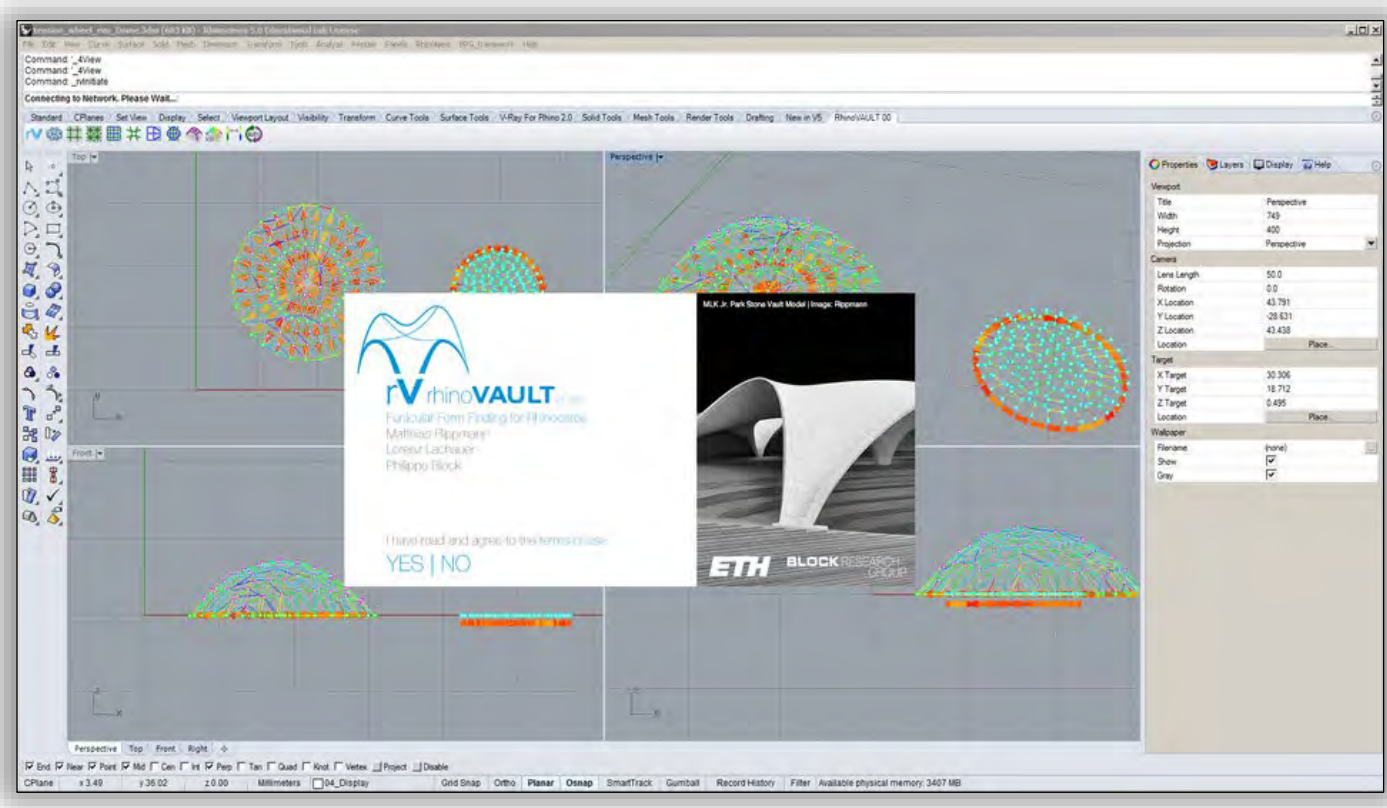




1. Pattern
2. Boundary conditions
3. Form diagram
4. Dual diagram
5. Horizontal equilibrium
6. Force diagram
7. Vertical equilibrium
8. Thrust diagram



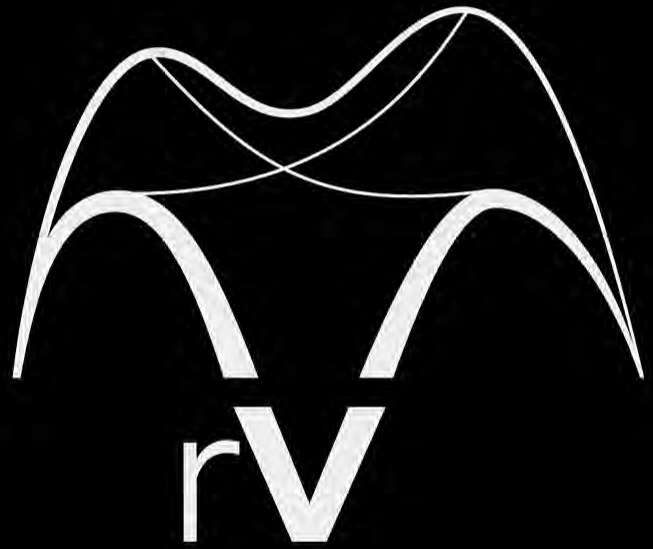




rhinoVAULT | Rippmann, Lachauer & Block (2012)

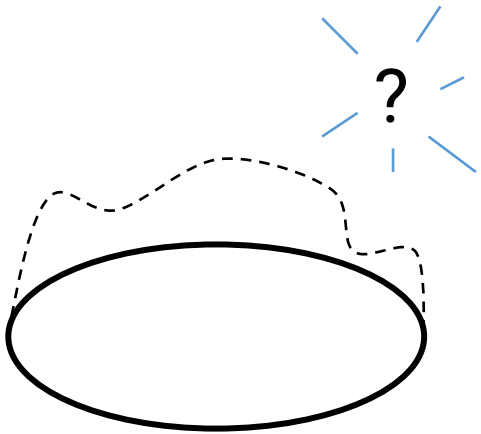
rhinoVAULT

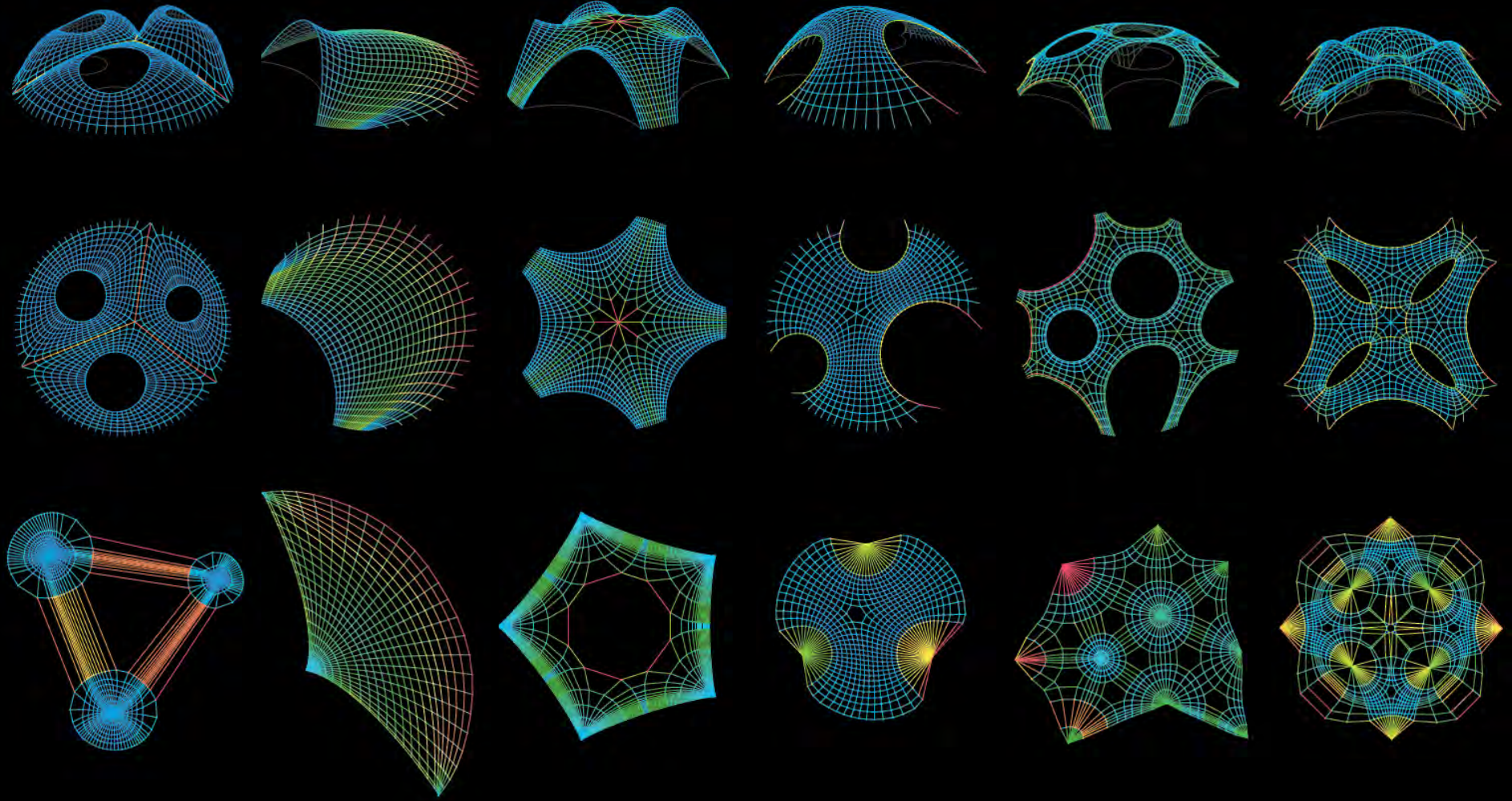
MATTHIAS RIPPMANN
LORENZ LACHAUER
PHILIPPE BLOCK

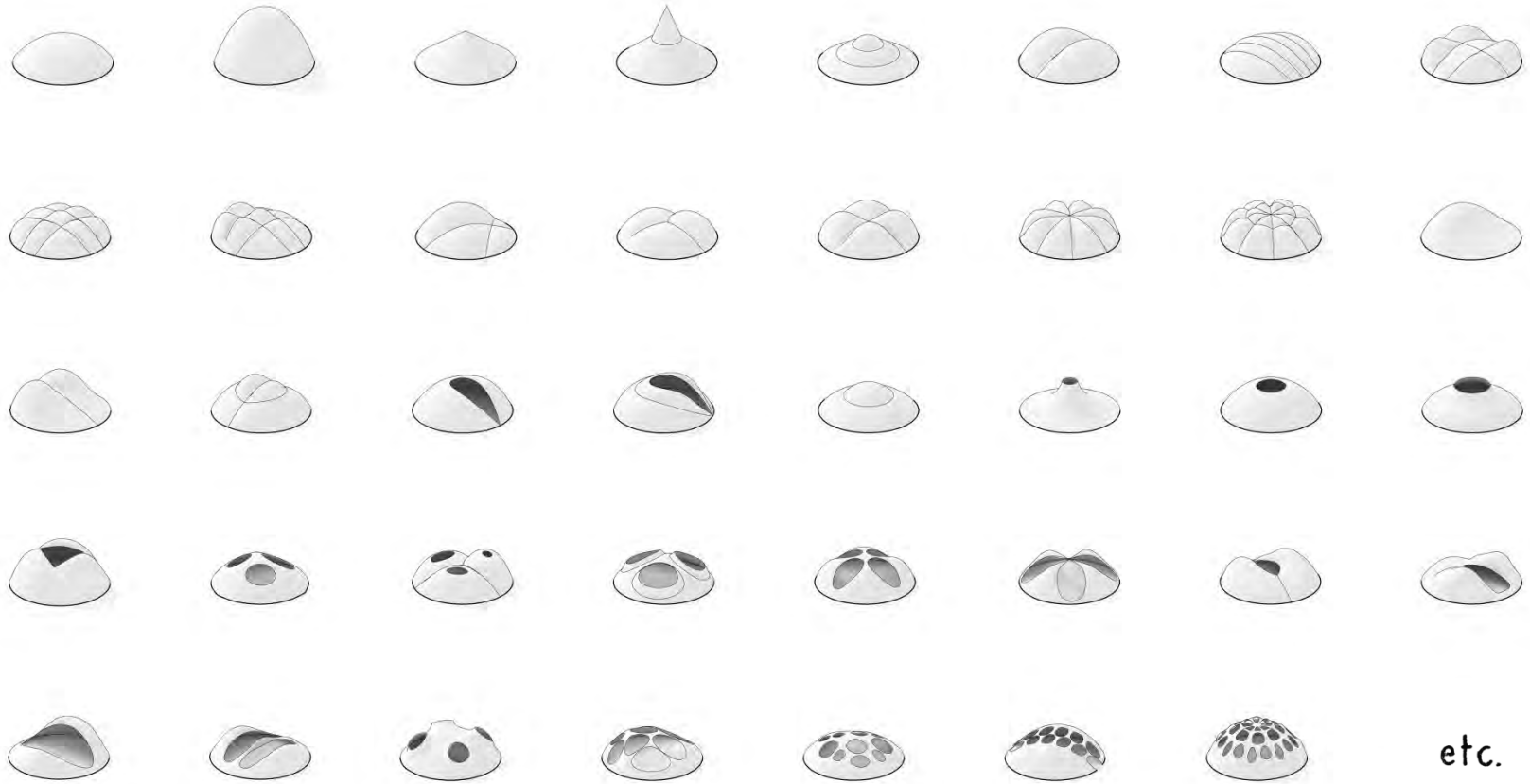


ETH Zürich

BLOCK RESEARCH GROUP

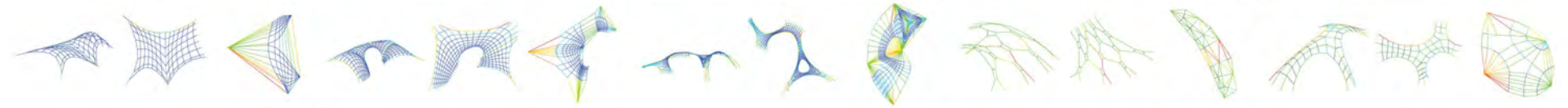


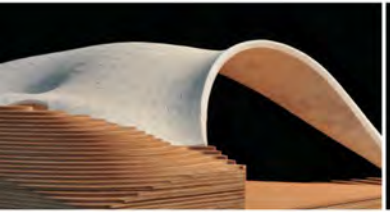



etc.

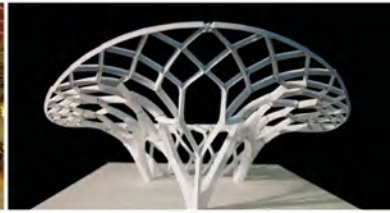


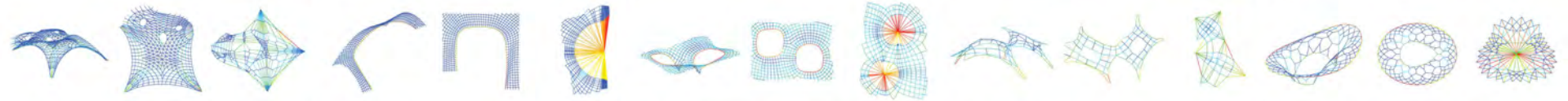






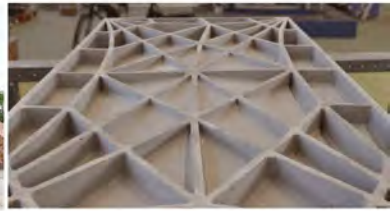


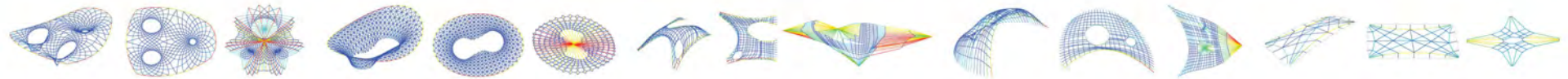


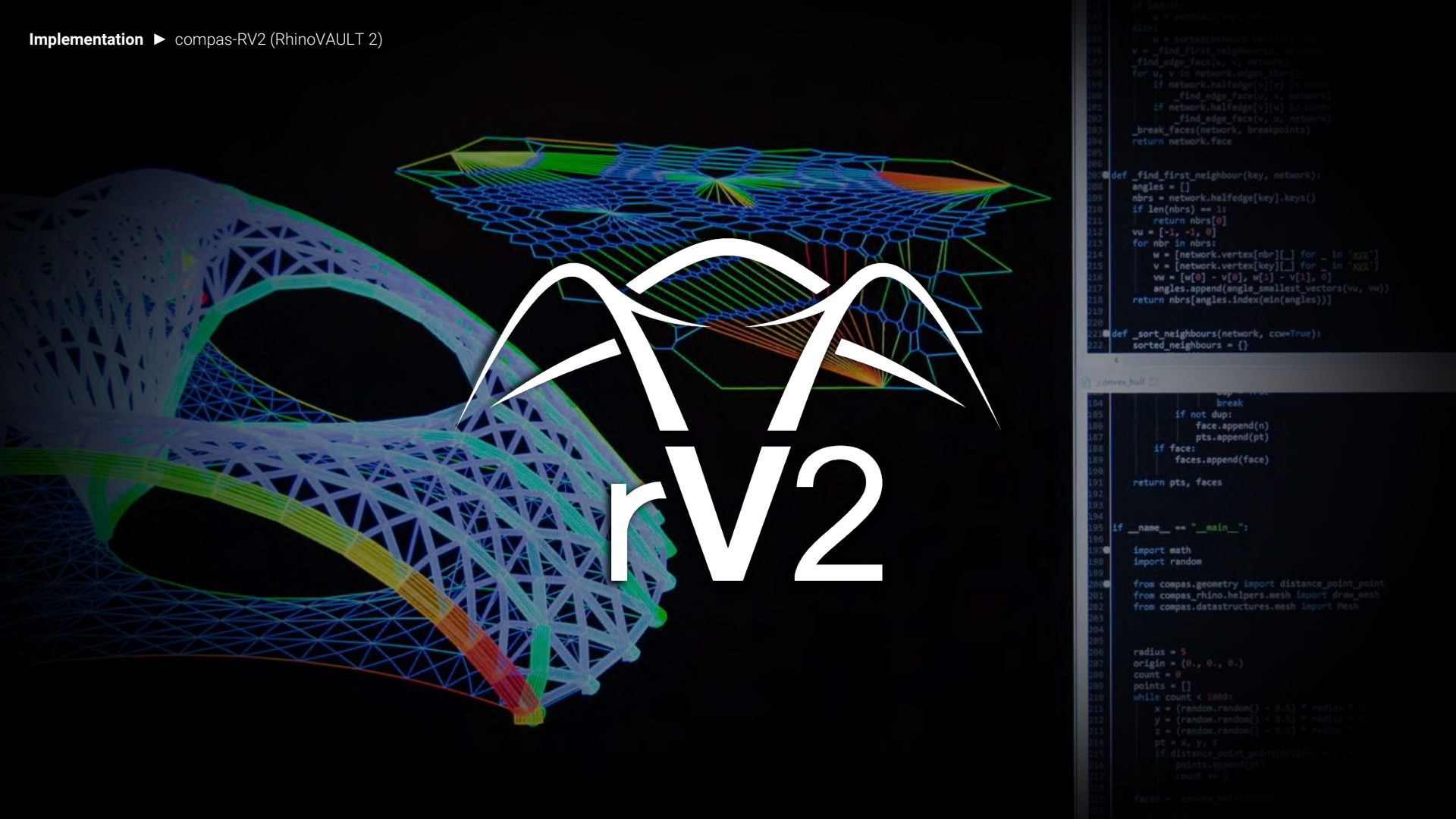






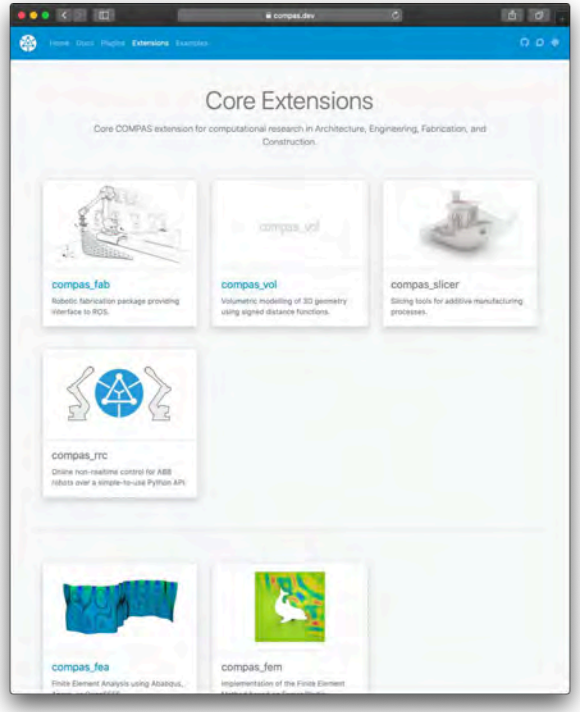


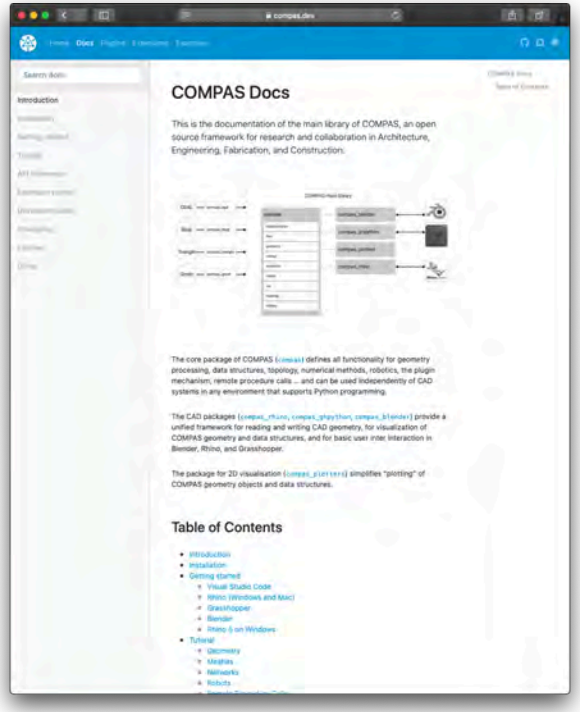



rV2

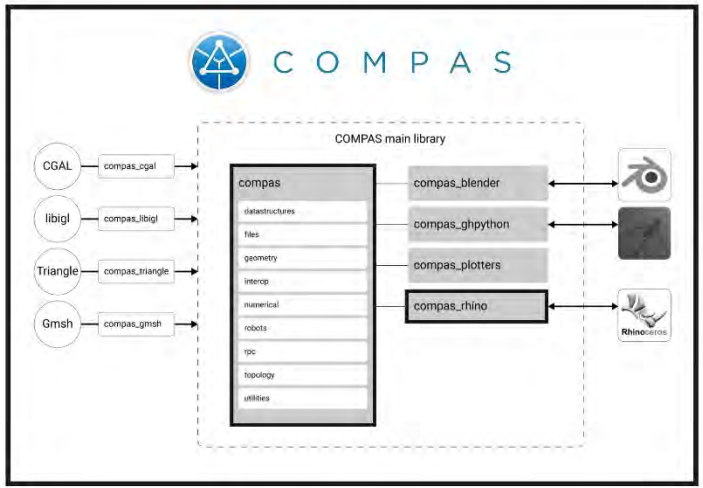
COMPAS

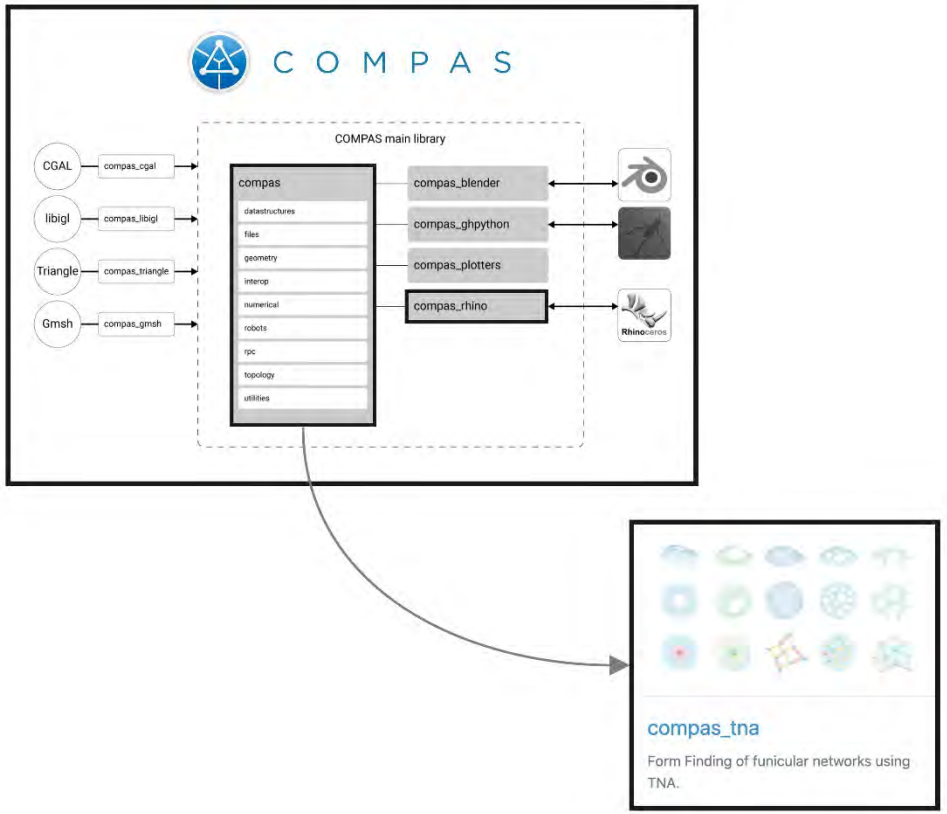
Open-source, Python-based framework for computational research and collaboration in architecture, engineering and digital fabrication

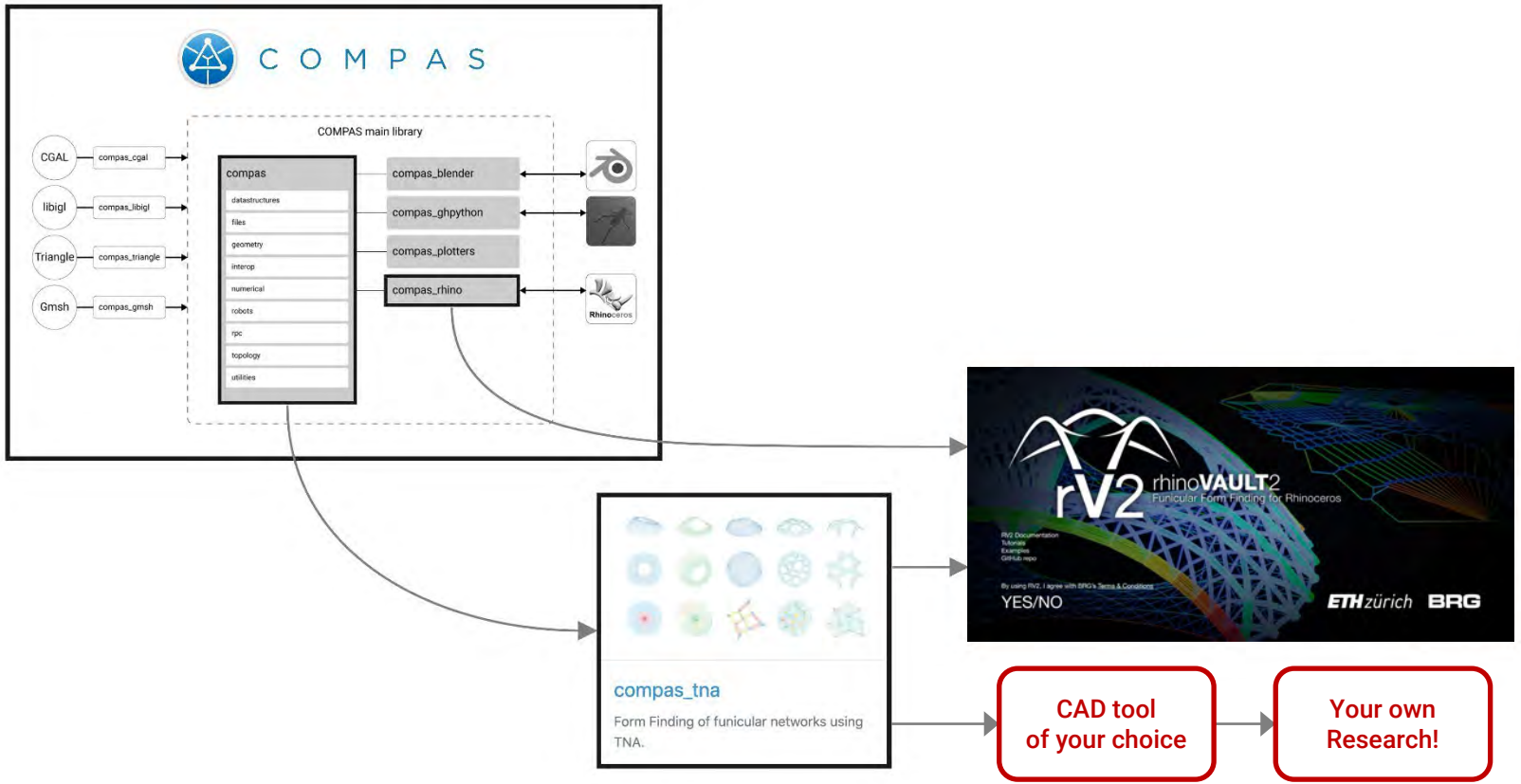


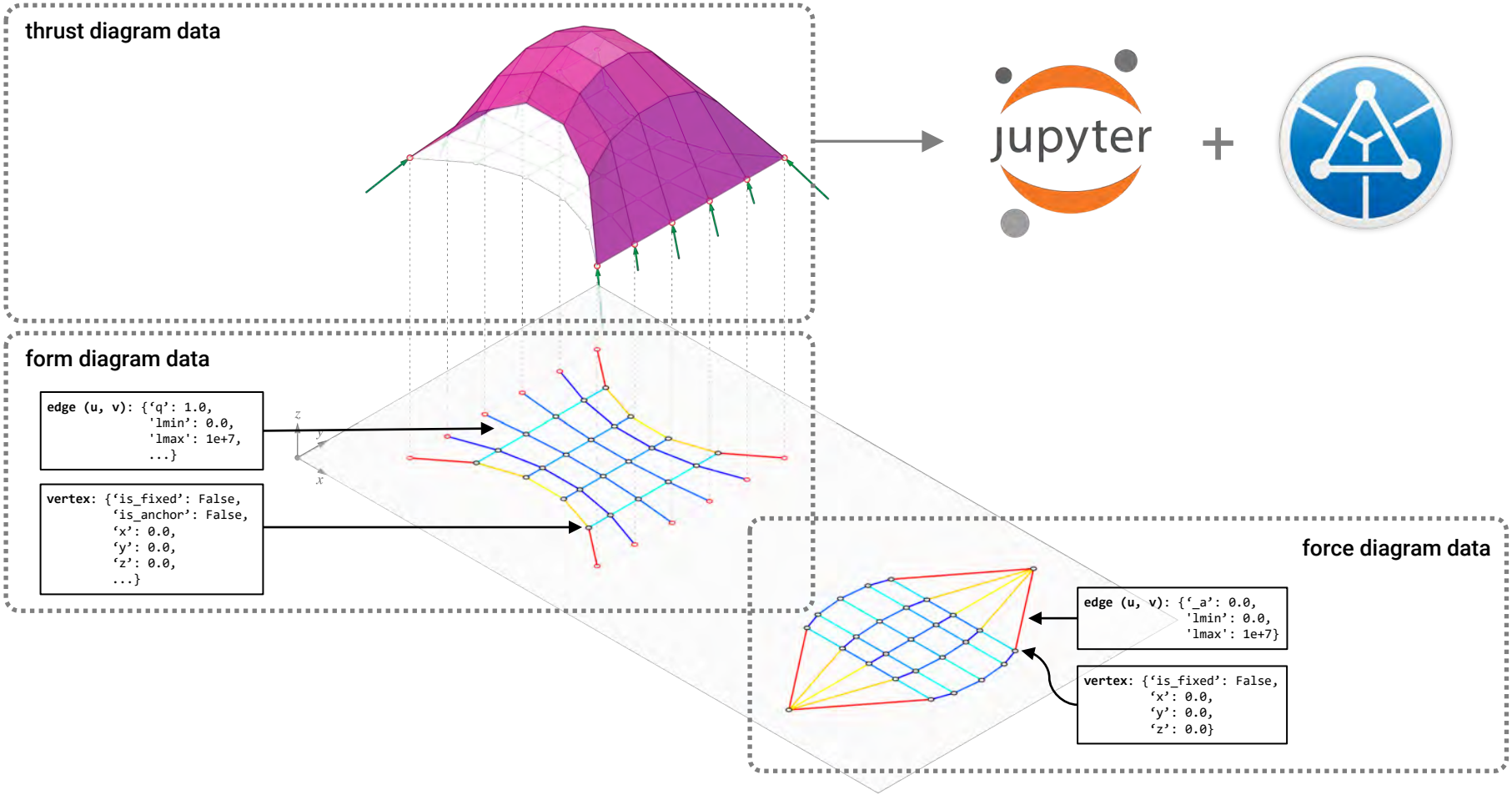


https://compas.dev


COMPAS
COMPAS main library
CGAL
compas_cgal
libigl
compas_libigl
Triangle
compas_triangle
Gmsh
compas_gmsh
compas
datastructures
files
geometry
intercop
numerical
robots
rpc
topology
utilities
compas_blender
compas_ghpython
compas_plotters
compas_rhino
Rhinoceros


COMPAS
COMPAS main library
CGAL
compas_cgal
libigl
compas_libigl
Triangle
compas_triangle
Gmsh
compas_gmsh
compas
datastructures
files
geometry
interop
numerical
robots
rpc
topology
utilities
compas_blender
compas_ghpython
compas_plotters
compas_rhino
Rhinoceros
compas_tna
Form Finding of funicular networks using TNA.


COMPAS
COMPAS main library
CGAL
compas_cgal
libigl
compas_libigl
Triangle
compas_triangle
Gmsh
compas_gmsh
compas
datastructures
files
geometry
interop
numerical
robots
rpc
topology
utilities
compas_blender
compas_ghpython
compas_plotters
compas_rhino
Rhinoceros
rV2
rhinoVAULT2
Funicular Form Finding for Rhinoceros
YES/NO
ETH zürich
BRG
compas_tna
Form Finding of funicular networks using TNA.
CAD tool of your choice
Your own Research!


thrust diagram data
jupyter
+
form diagram data
edge (u, v): {'q': 1.0,
'lmin': 0.0,
'lmax': 1e+7,
...}
vertex: {'is_fixed': False,
'is_anchor': False,
'x': 0.0,
'y': 0.0,
'z': 0.0,
...}
force diagram data
edge (u, v): {'_a': 0.0,
'lmin': 0.0,
'lmax': 1e+7}
vertex: {'is_fixed': False,
'x': 0.0,
'y': 0.0,
'z': 0.0}

Universal Computing “Turing” Machine

… a machine capable of computing anything that is computable by executing instructions (program) stored on tape (memory), allowing the machine to be programmable …

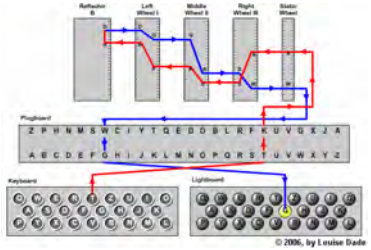
“On Computable Numbers” | Alan Turing (1936)

The Imitation Game (2014)

ATTACKATDAWN

Encryption machine (“Enigma”)

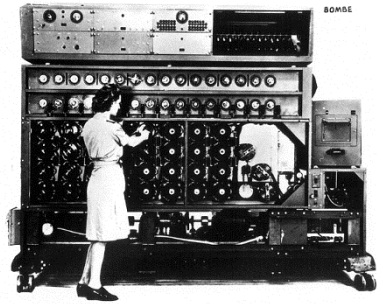
”Scrambler” using rotor configurations



Encryption key

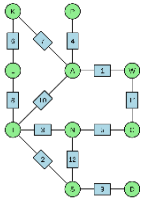
Algorithm

WSNPNLKLSTCS



Decryption machine

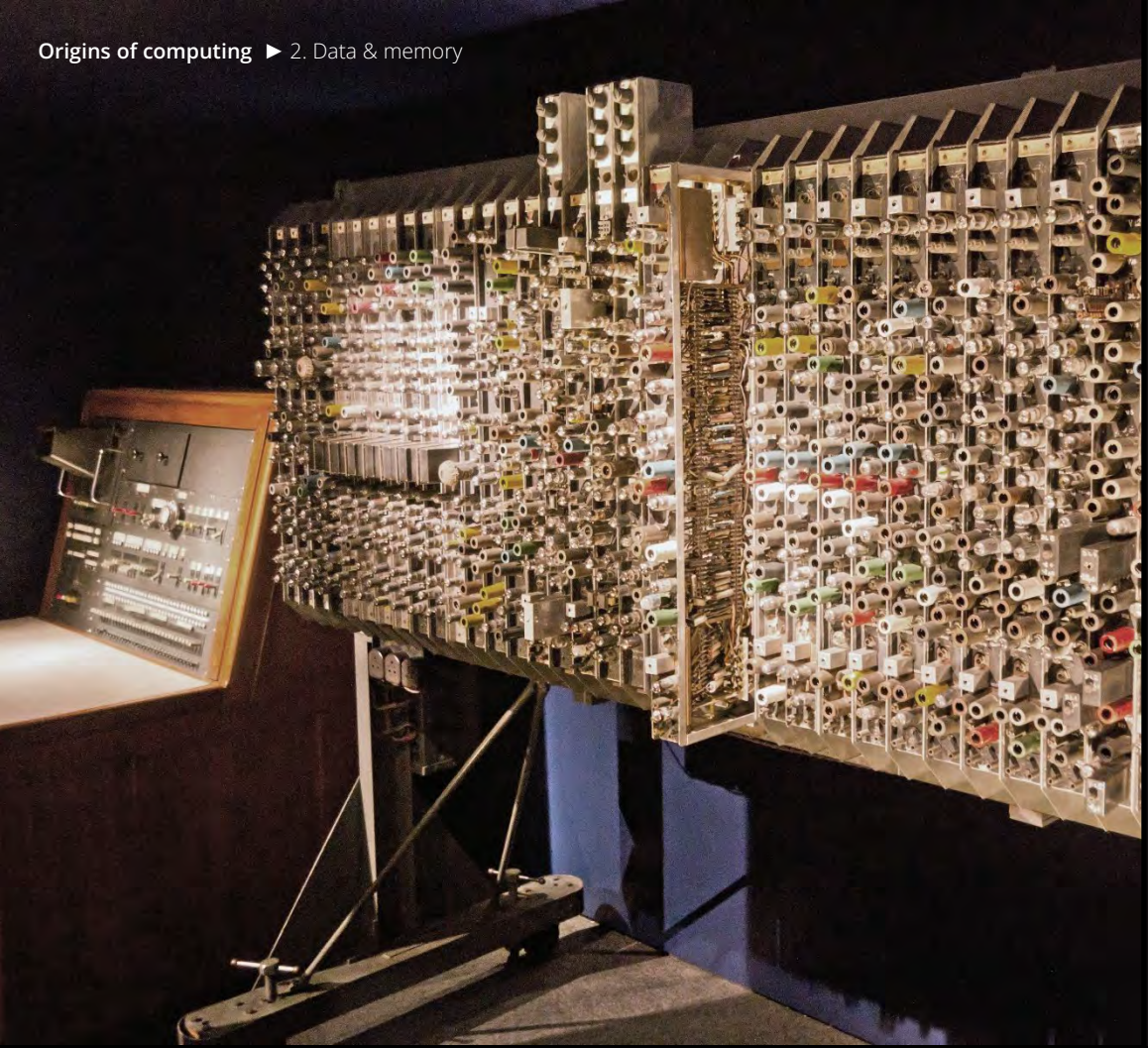
Multiple reverse-engineered Enigma machines



Ciphertext	W	S	N	P	N	L	K	L	S	T	C	S
Plaintext "crib"	A	T	T	A	C	K	A	T	D	A	W	N
Message position	1	2	3	4	5	6	7	8	9	10	11	12
Upper drum setting	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Middle drum setting	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Lower drum setting	A	B	C	D	E	F	G	H	I	J	K	L

Decryption key

Reverse-algorithm



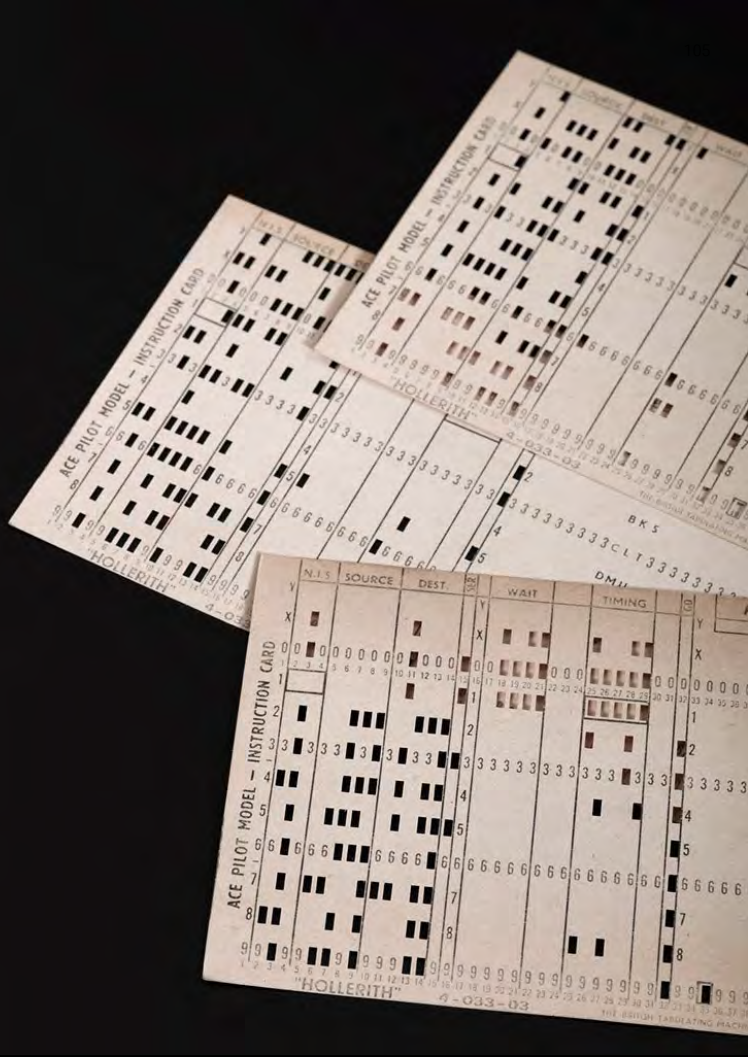
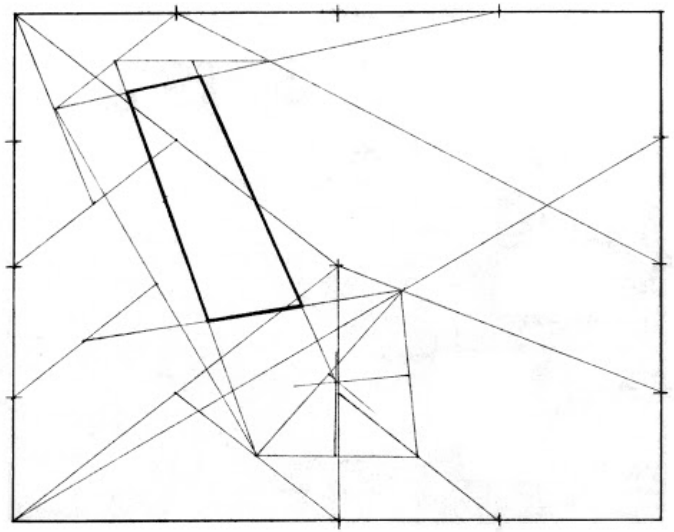


Photo: Antoine Taveneaux

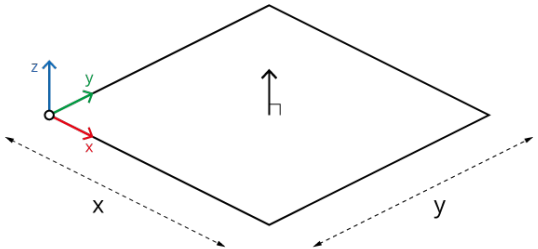
Photo: Science Museum London



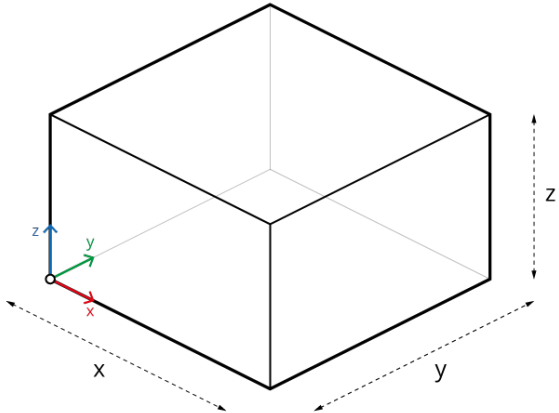
A quadrangle which is formed and enclosed by four lines, the first of which is drawn from a point halfway between a point halfway between the center of the wall and the upper left corner and the midpoint of the left side and the upper left corner to a point halfway between the midpoint of the top side and the upper right corner, the second line from a point halfway between the start of the first line and a point halfway between the midpoint of the top side and the upper left corner to a point halfway between a point halfway between the center of the wall and the lower left corner and the midpoint of the bottom side, the third line from a point halfway between a point halfway between the start of the first line and the end of the second line and a point halfway between the midpoint of the left side and the lower left corner to a point which is on an axis between the lower left corner to a point halfway between the midpoint of the right side and the upper right corner where a line drawn from the center of the wall to a point halfway between the midpoint of the right side and the lower right corner would cross that axis, the fourth line from a point equidistant from the end of the third line, the end of the second line and a point halfway between a point halfway between the center of the wall and the midpoint of the bottom side and a point halfway between the midpoint of the bottom side and the lower right corner to a point halfway between the start of the second line and a point where a line would cross the first line if it were drawn from the midpoint of the right side to a point halfway between the midpoint of the top side and the upper left corner.

– Sol LeWitt, 1974

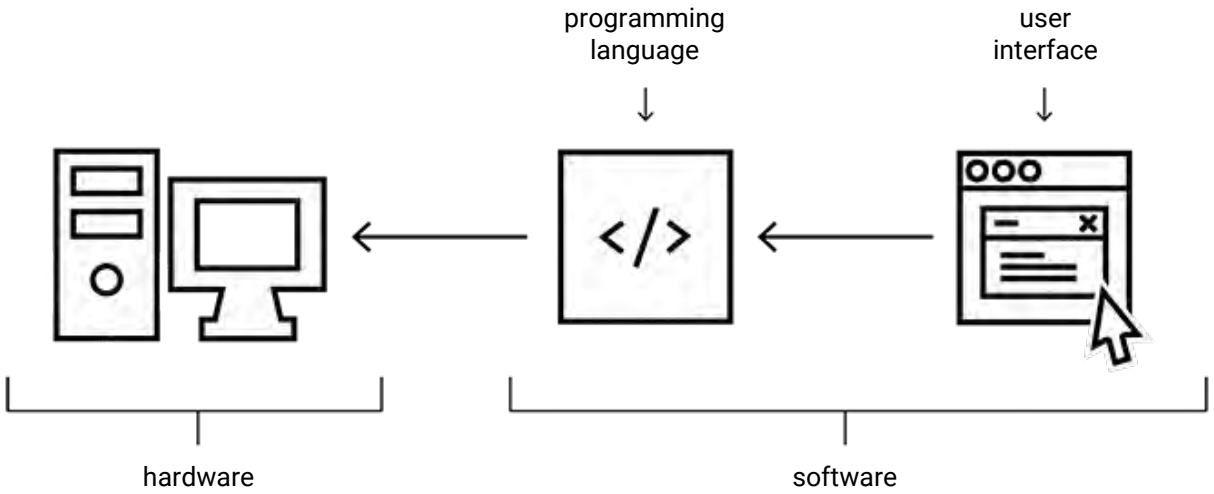
1. Choose a starting point

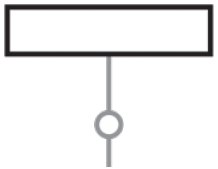


2. Draw a rectangle with dimensions X and Y on the xy plane

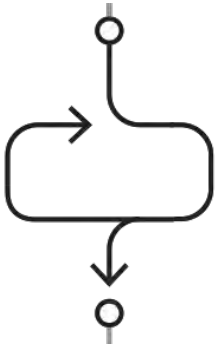


3. Extrude the rectangle in the z-axis with amount z

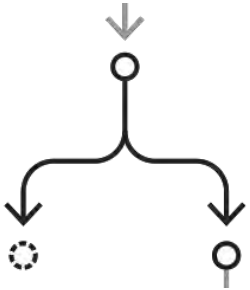

programming
language
user
interface
hardware
software



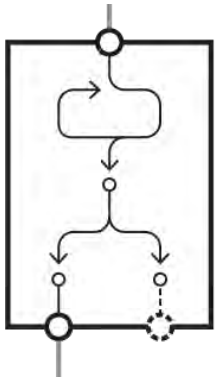
1. variables



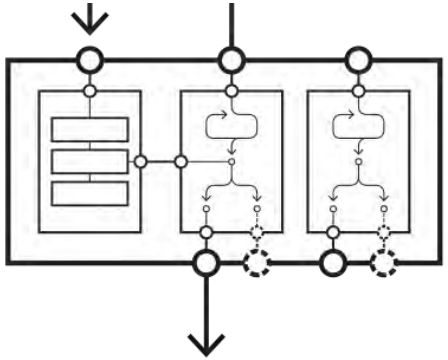
2. loops



3. conditional

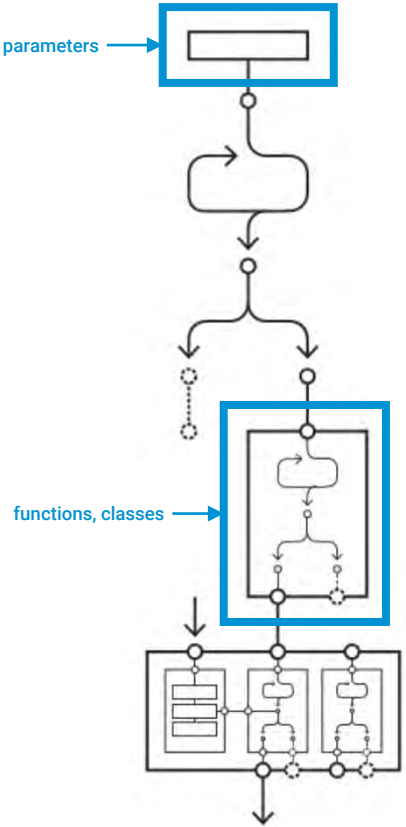


4. functions



5. objects

Adapted from Danil Nagy



1. variable(s)

```
x = 1
y = 'hello world'
myList = [1, 'b', 3]
myDict = {'a': 1, 'b': 2}
```

2. Loops

```
for i in range(10):
    # do something 10 times

for element in myList:
    # do something for all elements
```

3. Conditional

```
if x > 10 and y > 10:
    # do something
elif x > 5 or y < 5:
    # do something else
else:
    # do something else
```

4. Functions

```
def myFunction(input1, intput2):
    # do something
    return output

c = myFunction(a, b)
```

5. Object

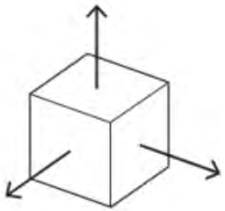
```
class MyClass:
    def __init__(self, input1):
        self.localVar = input1
    def myMethod(self, input2):
        # do something
        return output

classInstance = MyClass(a)
c = classInstance.myMethod(b)
```

procedural programming ("scripting")

object-oriented programming ("OOP")

Adapted from Danil Nagy

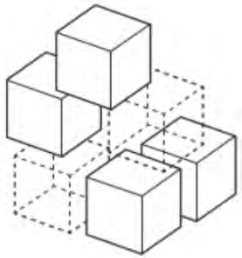


Morphological

- Continuous measures

+ Good top-down control over design

- Can usually only generate simple design spaces

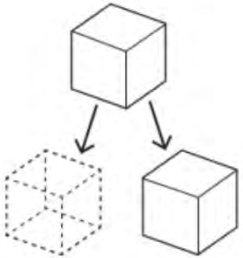


State-change

- Choices, categories

+ can create discontinuous design spaces
+ control over individual elements

- Many inputs required (each element needs to be controlled separately)

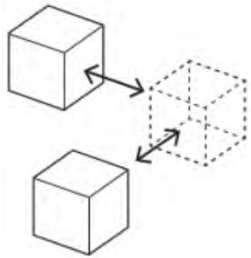


Rule-based

- L-system, shape grammars, 1D cellular automata

+ reduced number of inputs (abstraction of inputs into rule sets)
+ can create complexity

- Only top-down control
- can't control individual behaviour
- can't create emergence
- Potentially redundant or incomplete design space



Behavioral

- object-oriented, agent-based behavior models (dynamic)

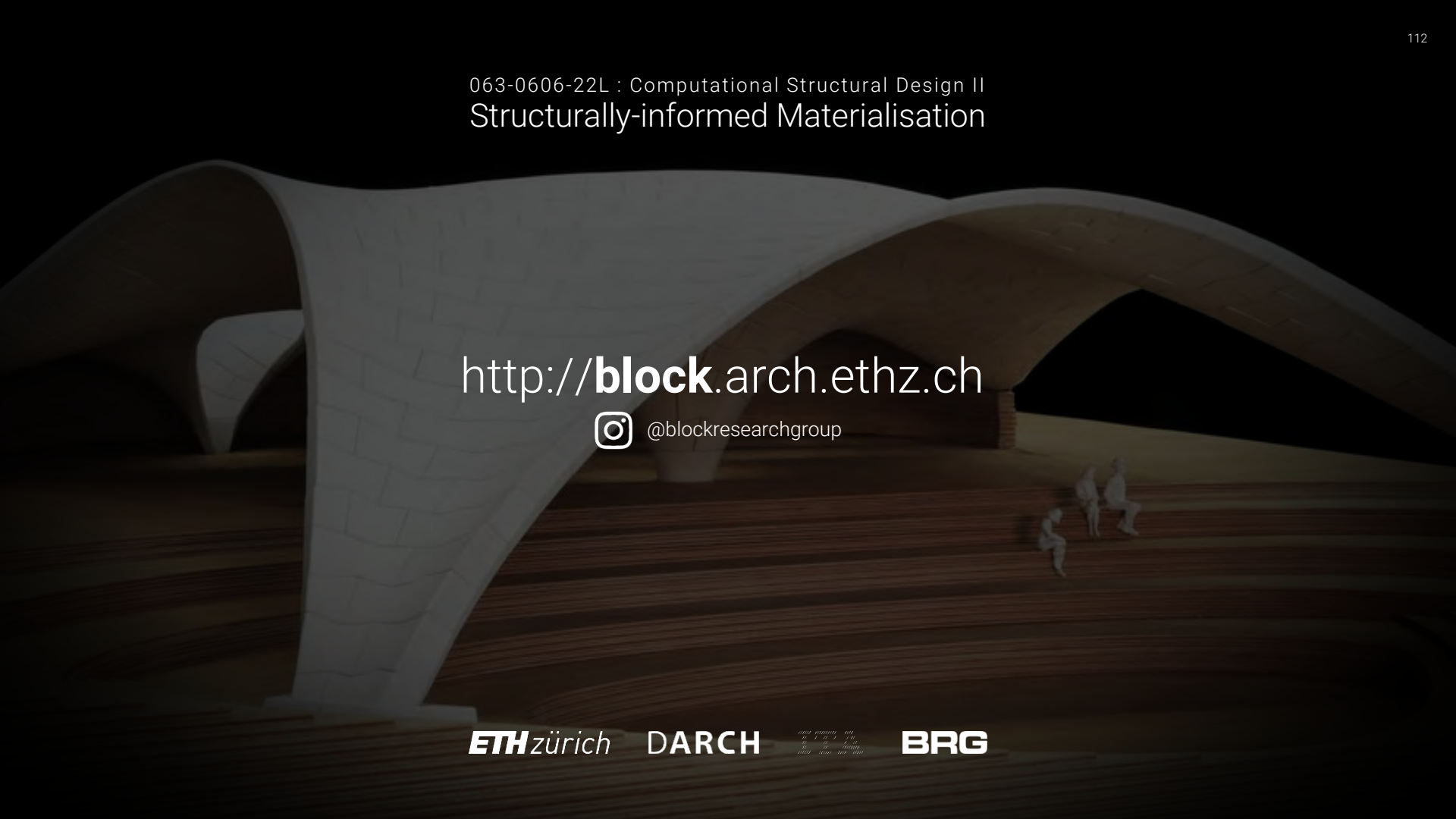
+ reduced number of inputs (abstraction of inputs into agent behaviors)
+ can lead to emergence

- Little intuitive control over macro design
- Potentially redundant or incomplete design space

Adapted from Danil Nagy

063-0606-22L : Computational Structural Design II

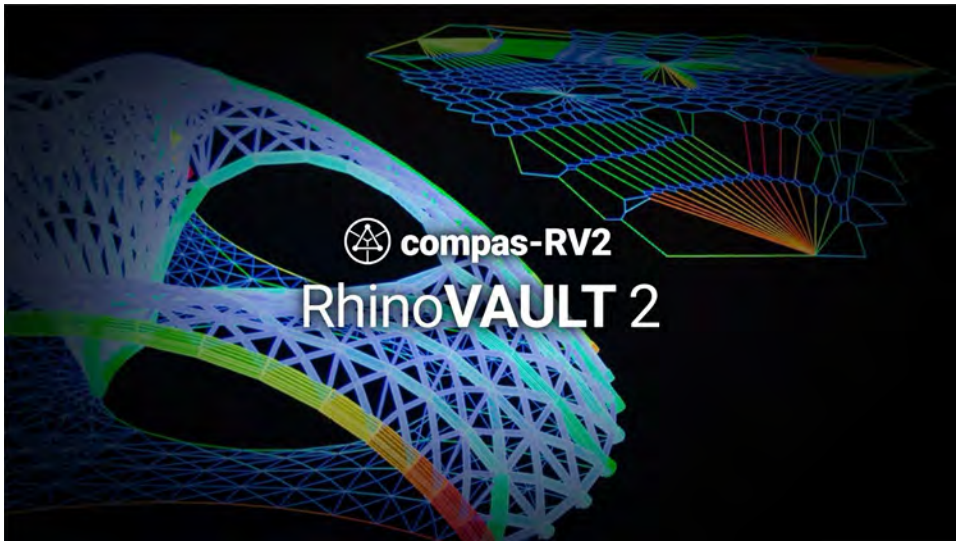
Structurally-informed Materialisation



http://**block**.arch.ethz.ch

@blockresearchgroup

ETH zürich DARCH BRG



Version 1.4.7

Course website ⟶ https://github.com/BlockResearchGroup/compas-RV2/releases/tag/v1.4.7