Beyond Design Freedom

Providing a Set-Up For Material Modelisation within Kangeroo Physics

Florian Chéraud[1]
[1] *Ecole Nationale Supérieure d'Architecture Paris-Malaquais*
[1] *florian.cheraud@paris-malaquais.archi.fr*

Kangaroo Physics, a physical simulation engine, is amongst the most used form-finding tool with nearly 500 000 downloads. Mostly resorted to by users with moderate computation skills, it provides a simplified interface for an advanced simulation tool. It is a Particle Spring System relying on the Dynamic Relaxation method and offering a wide design space. Thanks to the visual scripting interface provided by Grasshopper, the user has access to a fixed set of physical ``goals'' and unitless variables, without having to work with more complex aspects of the Kangaroo physical model. This setup induces a disconnection between the user and the physical model with its variables. The goal of this research is to introduce, within the Grasshopper environment, a tensile parameter, the Young Modulus, into the Kangaroo model. Thus, while preserving the design freedom of the plug-in, a better understanding of the physical behaviour modelled in Kangaroo is offered to neophytes, as well as better control of material properties.

Keywords: *Kangaroo Physics, Tensile Parameter, Form-Finding, User Control*

INTRODUCTION

With the first Digital Turn in architecture (Carpo 2012), in the 1990s design processes have shifted from representation to simulation. Designers work in a whole new space, where topology, geometry, material and load case are equivalents parameters that can be programmed. The designer's role is therefore overturned: he/she no longer projects a finite shape, but he/she collaborates with the machine at different degrees, from adjusting parameters on an interface, to designing his own tools, no longer focusing on a finite shape but on a dynamic design process. With the massive development of digital tools and their democratisation, the first has recently become predominant, raising the issue of the designer's ability to understand and fully master his tools (Gaudillière 2020).

One of these processes is called form-finding. We can distinguish two approaches to form-finding which could be called: the "physical rationalisation" approach and the "shape freedom" approach. The first is to maximise the mechanical and geometric accuracy of the model, in order to establish the constructibility of a given object. In this case, a mechanical problem is defined ahead of the form-finding process. The second is part of a more prospective context and offers more formal variation possibilities. In this case, it is necessary to operate with as few constraints as possible, in order to generate a design space as large as possible. These two approaches co-exist and can correspond to the stages of the architecture project. However, the physical rationalisation approach is the most common in research today (Luo et al. 2018) (Shi et al. 2018). The technical complexity of new physical simulation models forces designers to rely on tools whose physical principles governing simulation are hidden in favour of a simplified interface (Gaudillière 2020)(Mueller and Brown 2017). Such software with a simplified interface are accessible without any in-depth technical knowledge, be it in the field of physics of in the field of computation.

With nearly 500,000 downloads [1], Kangaroo Physics is the most downloaded grasshopper plugin and one of the most widely used form-finding tool. Kangaroo is among those tool with a simplified interface, thanks to its integration into Grasshopper, a visual programming software relying on dynamically linked libraries (DLL).

The purpose of this paper is to question the role played by tools and their interfaces in the designer's practice, through an analysis of Kangaroo Physics. The user is offered a simplified interface, this simplification allowing for a wider design freedom by easing the resort to advanced simulation tools. This to the detriment of physical rationality set aside, because the technical principles and physical laws at stake during the simulation are overshadowed. The goal of this paper is therefore to explicit Kangaroo's functioning in a pedagogical objective.

First, a brief historical review will be made and a classification of the form-finding methods will be established. Then, a study of the functioning of Kangaroo is proposed, in order to to detect interface biases. Finally, a set of tools is developed, and applied to three case studies.

FORM-FINDING METHODS CLASSIFICATION

The form-finding process can be defined as follows: it is the search for a state of equilibrium of forces under given conditions and according to a given stress.

Until the beginning of the 20th century, form-finding was based on analog research processes. In his experiments with soap bubbles, Frei Otto (1969) explored the properties of matter to generate minimal surfaces. Before him, Antoni Gaudi (see Fig. 1a) used suspended chain models to prefigure vaults and arches shape. With analog research, the architect places himself in a physical relationship with material. This type of approach requires intuition. However, analog models show their limits when the design intention requires a high number of iterations, on complex geometries. Moreover, scaling remains an issue.

Since the end of the 1960s, the rise of computer science has made it possible to develop form-finding methods in the theoretically unlimited design space offered by computers (see Fig 1b). Block and Veenendaal define three families of algorithmic form-finding methods (Adriaenssens et al. 2014, table 10.1 p.116) :

- Stiffness Matrix Methods (such as Natural Shape Finding (1974))
- Geometric Stiffness Method (such as Force Density Method (1971), Thrust Network Analysis (2007))
- Dynamic Equilibrium Methods (such as Dynamic Relaxation (1984), Particule Spring System (2005))

These methods can then be classified into two categories: static problem methods (stiffness matrix methods) and dynamic problem methods (dynamic equilibrium method) (see Fig. 2).

The first category includes methods that require a rigorous description of boundary conditions (geometry, topology, material, loads), the simulation is dependent on the material and on solid geometry, which is difficult to reconcile with any prospective approach. Methods such as the FEM consume a lot of computing power and are intended to evaluate a given solution. In short, it is a knowledge-based process, while the objective of the form-finding process is precisely the opposite; to produce variety. Al-

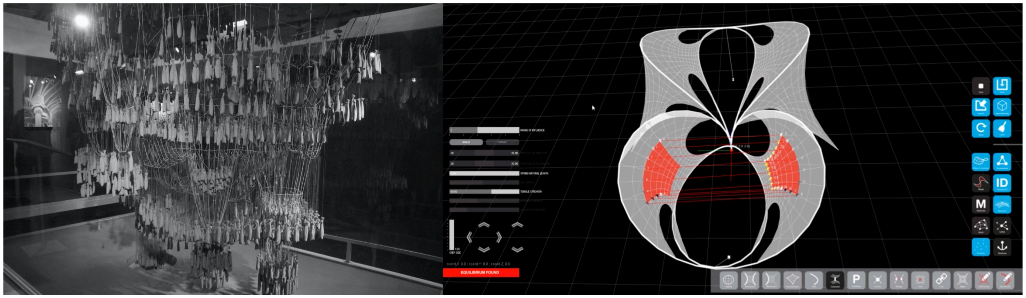


Figure 1
1a. (left) A model with suspended chains, Gaudi, 1889 [2]. 1b. (right) ElasticSpace ITKE interface, a screenshot of the video [3].

though the debate remains open, we chose to exclude these methods from form-finding as is defined in this research.

The second category includes methods making it much easier to perform interactive deformations and manipulate complex interactions with only a few equations and parameters. In fact, these methods are well suited to generate visually correct simulations. For example, the Dynamic Relaxation Method is based on the resolution of the balance of forces to reach the static state of a structure. In addition, this method requires the least computing power and produces the most diverse results(Veenendaal and Block 2012, section 2.5). The most recent process (2005) based on this latter method is the particle-spring system (PSS). In a PSS, each object is discrete, that is, each object must be readable as a set of points. Each point is characterised by its position, mass and velocity. The particle system's evolution is calculated using springs; understand the repulsive, attractive or neutral relationship between two particles in the system. It is therefore possible to build interactive models with parameters modifying the geometry in real-time. Kangaroo Physics belongs to this category (see Fig 2). Thus, the surface becomes a mesh, i.e. a set of points and lines. However, a particle spring-system cannot simulate complex physical phenomenons such as flexion, given that in an environment that was originally designed to simulate hulls and minimal surfaces, structures operate with normal forces (compression/traction) only.

Based on the PSS method, several design tools have been developed in the past years. CADenary is a software developed by Axel Kilian (MIT), in 2005, based on Gaudi's catenaries system and offering a configurable tool for the generation of compression-only vaults (Kilian and Ochsendorf 2005). Regarding complex structures involving hybrid dynamic bending and tension, in 2013 Achim Menges developed SPRINGform (Ahlquist et al. 2013). Evy Slabbinck's recent research at the University of Stuttgart has led to the development of a real-time recursive topol-

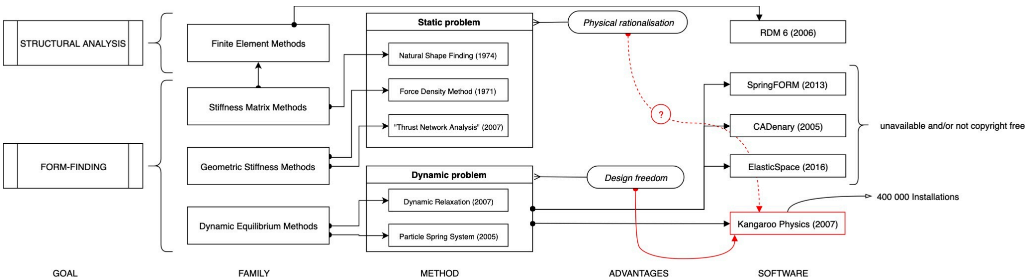


Figure 2
Available form-finding tools.

ogy algorithm: ElasticSpace [3] (Suzuki et al. 2017). All these solutions have a simplified interface (see Fig. 1b). Kangaroo Physics relies on the same form-finding method but is the only released and free tool today. K2Engineering is a grasshopper plugin using Kangaroo Physics library, developed by Cecilia Brandt. Oriented toward physical rationalisation, the plugin includes similar adjustments as the ones proposed in this paper. Although this plugin has been used in some works (Melville et al. 2017)(Bonavia et al. 2019), no article details how this plugin works, and it hasn't been released, although the source code is available[4]. The author has not been able to get the plugin working at the time of writing this article. The existence of K2Engineering shows that there is a flaw since there was a need to make this plugin, but this flaw is not demonstrated anywhere, even though it is very important for reducing the gap between the user, and the physical model and its variable. The goal of this article is to reduce this gap, first by showing its existence, and then by developing a clear and appropriable approach to reduce it.

Despite Kangaroo Physics' many advantages, several issues remain in the form-finding developed in it. It relies on known methods without showing their full complexity and potential. The source code of Kangaroo has been published on Github by Daniel Piker [5], and online documentation about the methods involved are available online[6]. Using these resources, we will develop the functioning of Kangaroo Physics (a Particle-Spring System) in order to highlight this gap, then we will propose a solution that will be applied to a set of three case studies in order to validate the principle, validate design freedom, and validate the results.

ANALYSIS

Particle-Spring System Functioning

Hooke Law (Hooke 1678). In a Particle-Spring System, a spring is calculated according to Hooke's law:

$$F_s = k \cdot x \tag{1}$$

Dynamic relaxation. The name dynamic relaxation appears for the first time in an article by A. S. Day and J. R. Otter (1960), where the authors sought to model the movement of the swell. They replace the equations of continuity and motion of fluid mechanics with those of elasticity (Hooke's law) and structural mechanics dynamics (Newton's Laws).

This method simplifies the solution of a non-linear equation system into an iterative and linear explicit calculation. The proposed static solution is therefore the result of a damped dynamic process. The dynamic behaviour itself does not matter because it is the static state that is sought. The dynamic relaxation iterative model is explained by Cyril Douthe (2007, part 3.2.2 p.83).

Kangaroo Operating Mode

A dynamic relaxation algorithm follows specific steps (Adriaenssens et al. 2014, table 10.1 p.116), presented in the following parts, for a generic tensioned surface. The input is a mesh, its vertices and edges become nodes and connections in kangaroo. The output is a set of curves, that can be assembled into a mesh.

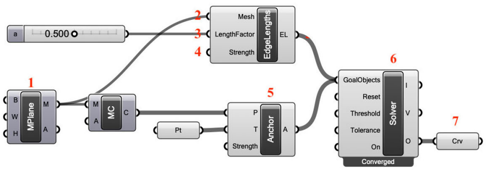


Figure 3
Algorithm for simulating a generic tensioned surface in Grasshopper with Kangaroo Physics.

Definitions of the initial conditions. Initial geometry (Li), rest geometry (L0), and load cases (see Fig. 3).

- 1: A mesh composed of vertices and edges
- 2: The initial length of the mesh edges gives Li for each edge
- 3: The length factor, a
- 4: "Strength" corresponds to the stiffness of the connection, k

We therefore retrieve, for each interaction L of the system, Hooke's law (equation 1):

$$F = k(a \cdot L_i - L_i) \quad (2)$$

With $L_0 = a \cdot L_i$

- 5: Definition of the anchorage points, here the 4 corners of the mesh.

Definition of convergence parameters. The solver component (see Fig. 3).

- 6: The threshold indicates the moment at which the simulation will be stopped and at which the algorithm will have converged. By default this threshold is 10-15, or one femtometer. By raising this threshold, we accelerate the convergence of the simulation, at the expense of its accuracy.

Four other entries can be found on this component: "GoalObjects" - is the sum of forces, "Reset" - to which a reset button for the simulation is connected, "Tolerance" - the threshold for which 2 points will be merged, "On" - To which a switch to turn the solver off is connected.

Definition of the masses at each node. By default each node has a mass equal to 1. Since the mass acts on the inertia, it also acts on the convergence speed of the algorithm.

Convergence loop. At each iteration and until the model converges.

Calculation of residual forces $\vec{R_n^t}$. The "GoalObjects" entry is the sum of the residual forces for each node.

Calculation of speeds $V_n^{t + \vec{\Delta} t/2}$. (Douthe 2007, part 3.2.2 p.83)

$$V_n^{t+\vec{\Delta} t/2} = V_n^{t-\vec{\Delta} t/2} + \frac{\Delta t}{m_n} \cdot \vec{R_n^t} \quad (3)$$

Calculation of positions $X_n^{t \vec{+} \Delta t}$. (Douthe 2007, part 3.2.2 p.83)

$$X_n^{t+\vec{\Delta} t/2} = \vec{X_n^t} + \Delta t \cdot V_n^{t+\vec{\Delta} t/2} \quad (4)$$

The algorithm will stop when the threshold is reached, we will then have the static state of the geometry.

Static state of the geometry (L). (see Fig. 3).

- 7: Once the static state of the geometry is reached, the PSS gives 3 different types of output: Iterations (I) number of iterations required until the model converges, Vertices (V) for the points, and Output Geometry (O) for the edges.

Verifying Newton's Laws

Kangaroo Physics uses Newton's first law to calculate the system. The third law is de facto validated because the PSS are based on the interaction between two points. The force between these two points acts in the direction of the line. It is added to one point and subtracted from the other.

In Kangaroo (see Fig. 4), by default, each particle has a unit mass. Newton's second law can be used to define a mass for a particule. (In version 1 of Kangaroo, we could define the mass of an object with a dedicated component, which is missing in version 2).

We therefore use the following expression:

$$\left(\frac{m_t}{n}\right) g = p \quad (5)$$

with:

- m_t : total mass of the object
- n : number of nodes in the discrete object
- g : acceleration of terrestrial gravity
- p : object weight in N

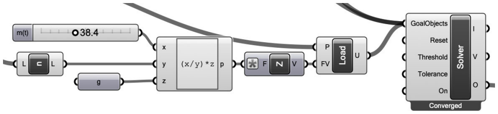


Figure 4
Weight / mass definition in Kangaroo Physics.

Young Modulus

The longitudinal modulus of elasticity, also called Young's modulus, is the constant that connects the stress (as normal stress, traction or compression) and uniaxial deformation (proportional deformation) of a given material. This constancy measures the stiffness of this material. It is therefore interesting to compare it with Hooke's law, to understand to what extent the latter approximates physical phenomena.

For a specimen of any material of module E, we have:

$$E = \frac{F \cdot L_0}{S \cdot \Delta L} \quad (6)$$

With:

- L_0 : resting length of the specimen
- ΔL : relative elongation
- F : force exerted by the material when it is elongated by ΔL
- S : section perpendicular to the force F

We thus find Hooke's law back:

$$F = \left(\frac{E \cdot S}{L_0}\right) \cdot \Delta L \equiv F = k \cdot x \quad (7)$$

$$[N] = \frac{[Pa][m^2]}{m}[a\,\text{dim}\,.]$$

Hooke's Law describes only the linear part of the behaviour of matter (see Fig. 5). We also observe that this behaviour varies greatly depending on the material (from 7.50.10-7 GPa to 1200 GPa [7]). It thus appears relatively complex to control physically rational behaviour, in Kangaroo as it is. Nevertheless, with equation 7, we understand that the modulus of elasticity can be introduced into Kangaroo. By in producing the Modulus of Elasticity, we open the way to the implementation of bending or buckling (and other properties) as it is linked to the Modulus of Elasticity, however it is not the purpose of this paper.

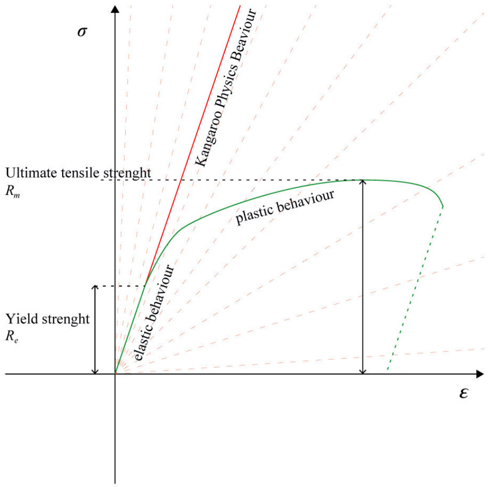


Figure 5
Kangaroo Physics behaviour v. given material behaviour.

PROPOSED RESOLUTION

In this section we define four components, following the introduction of Young's modulus into Kangaroo.

Rational Stiffness

With equation 7, Young's modulus can be linked to Hooke's law. A physical rationalisation can therefore be introduced in Kangaroo Physics. Thus, we have:

$$\left(\frac{E \cdot S}{L_0}\right) = k \quad (8)$$

This is translated into Kangaroo (see Fig. 6). For better intelligibility, the diameter in millimetres is the variable. It is converted to meters and gives section S (see asterisk) with $S = (d/2)^2\Pi$

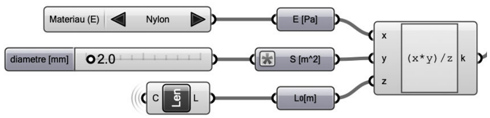


Figure 6
Rational stiffness in Grasshopper's environment.

Yield Strength

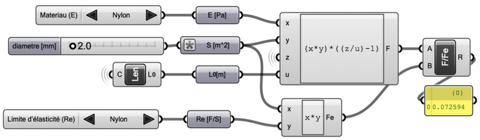


Figure 7
Yield strength in the Grasshopper environment.

Following what has been shown earlier in the paper, we are attempting to visualise the moment where a connection exceeds its yield strength in Kangaroo (see Fig. 7).

First of all, the yield strength is a stress expressed in Pa, so:

$$R_e = \frac{F_e}{S} \equiv F_e = S \cdot R_e \tag{9}$$

Moreover, by reusing equation 6, we have: $F = E \cdot S\left(\frac{L}{L_0} - 1\right)$

We visualise the elasticity state of a connection by comparing the force F in this connection with the maximum force F_e,

- if $\frac{F}{F_e} \in [0; 1]$ then the connection is in elastic behaviour, the simulation is correct.
- if $\frac{F}{F_e} > 1$ then the material is in plastic behaviour, the simulation is incorrect.

Moreover, the mechanical resistance limit noted R_m is also a constraint, it can be visualised with the same method.

Support Reaction

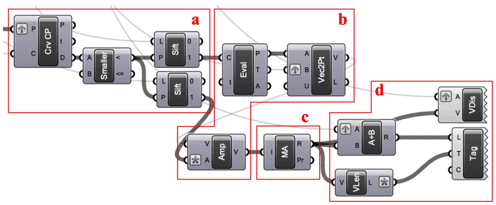


Figure 8
Support reaction in Grasshopper's environment.

In fact, the support reaction P at point A is equal to the sum of the vectors of the forces exerted on this support. In Grasshopper (see Fig. 8), the tool is divided into 4 parts: extraction of the connections related to the supports and the forces associated with these connections (a), definition of vectors having as amplitude the values of forces previously extracted (b), for each support, sum of the vectors (c), then display of the reaction vectors to the supports and their value in kN (d)

Rational Prestressing

By default, prestress is defined by a ratio between L_i and L_0, and prestress is characterised in Newton.

By picking up equation 6, we find: $L_0 = L_i - \frac{F \cdot L_i}{E \cdot S}$. Figure 9 shows this equation in the Grasshopper environnement.

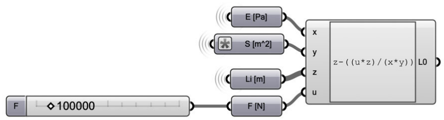


Figure 9
Rational prestressing in the Grasshopper environment.

CASES STUDIES

The data used in this section are from MathWeb [7]. The aim here is to validate our set of tools with three study cases.

Validation of the Principle

We are studying the elongation of nylon specimen, comparing traditional calculation and with our set of tools in Kangaroo.

Young modulus $E = 3GPa$, initial length $L_i = 1m$, section $S = 3,14160.10^{-6}m^2$, applied force ($m = 1kg$) $F = 9,80665N$

Manual calculation. Using equation 6, we find the value of L:

$$L = L_i + \frac{F \cdot L_i}{E \cdot S} \tag{10}$$

$L = 1,001041m$

Kangaroo Physics. By using the tools shown earlier in the paper, Kangaroo displays the same result : $L = 1,001041m$.This first experience confirms that Kangaroo is able to simulate material behaviour with our tools.

Validation of the Design Freedom

We are seeking to reproduce a classic form-finding case implementing our tools.

The simulation takes 46ms to compute, meaning there is a 0.046-second delay between the modification of a parameter by the user, and the visual feedback (see Fig. 10a). As a comparison, the initial Kangaroo simulation presented in part 3.2. requires 6ms. The simulation, therefore, remains usable in real-time. The user does not lose the flexibility of use, as a result of the addition of rationalisation tools. Thus the wide design space specific to Kangaroo is maintained. The variability of results enabled by this algorithm is shown in figure 10b. Initial geometry and prestress are the fixed parameters, but other parameters can be a source of variation: material, connection cross-section, weight, etc.

Validation of the Results

We are seeking to reproduce the conditions of the form-finding method developed by four Chinese engineers to optimise the cable network corresponding to the roof of the Suzhou Industrial Park Stadium (Shi et al. 2018). This article is recent and the subject of study, a tension cable network, can be modelled with Kangaroo Physics. The initial conditions are set out on page 7 of the reference article.

All cables have a mechanical strength limit (R_m) of $1,670MPa$, a density (d) of $7,85.10^3 kg/m^3$, a modulus of elasticity (E, Young's modulus) of $1,60.10^5 MPa$.

Definition of load cases applied to the nodes of the central hoop: $\dfrac{l_{h\infty p} \cdot S \cdot d}{nb_{nodes}} = 2173,41N$

Here the parameters are: height of peripheral nodes (anchoring), prestress level of each cable, section of each cable. We record the orders of magnitude of the parameters in Figures 8 to 12 of the referenced article: cable section: from 0.006 to 0.0150 m2 and node elevation: from 0 to 30 000 mm.

Having read this information, we can build a model with the system developed in case study 2. The development and application of a genetic algorithm is not the subject of this paper, so this part will not be covered. The objective function is therefore given page 5 of the reference article:

$$Objv(x) = \sum d_j \cdot S_j \cdot l_j \qquad (11)$$

By using the tools shown earlier in the paper, an algorithm is constructed in Grasshopper.

The figure above (see Fig. 11) shows 3 forms generated by the user's search. The details of the force per connection in kN (a) and the mechanical strength limit F/F_m (b) are given.

Using their genetic algorithm to have a bespoke dimensioning on each cable of the system, Luo, B., Ding, M., Han, L. and Guo, Z. get similar results. This, therefore, demonstrates that the proposed components for Kangaroo Physics produce valid results. The proposed method opens the understanding of complex form-finding cases, such as the design of a stadium roof, to neophytes.

CONCLUSION

Using Grasshopper's programming environment, we have integrated the elasticity parameter (Young's modulus) into Kangaroo's particle system. The new model is integrated into the form-finding process and projects four layers in real-time: the force F in kN per connection, the ratio between the force F and the yield strength limit force F_e, the ratio between the force F and the mechanical strength limit force F_m, and the reaction to the supports. Three new parameters are a source of shape variation: the material (Young's modulus E, yield strength R_e, mechanical strength limit R_m), but also the weight in Newton, and the prestress in Newton.

The three study cases demonstrate both the accuracy and the interest of the proposed model at different scales and complexity degrees. It opens the way to a new level of accuracy in form-finding with

Kangaroo for untrained users, by expressing the full technical complexity at stake during the simulation by Kangaroo. Beyond expliciting Kangaroo's functioning in a pedagogical objective, and implementing physical properties through the Young Modulus, this research is a starting point to adding more physical properties.

In our quick paced technological era, digital tools and the algorithms they rely on tend to be displayed via simplified interface leading designers to partly losing control of their production. This is an attempt to give back control to designers, by reducing the gap between the user and the physical model with its variables. This research aims at reconciling the physical rationalisation approach with the shape freedom approach, by maintaining advantages coming from both, in order for neophytes to better grasp the place of computational tools in the design process.

ACKNOWLEDGEMENT

This work was conducted during my master's degree at the Ecole Nationale Supérieure d'Architecture Paris Malaquais as part of the Transitions seminar[8]. I would like to thank Nadja Gaudillière in particular for her support, as well as Thierry Ciblac, Loïc Couton, and Robert Le Roy.

REFERENCES

Adriaenssens, S, Block, P, Veenendaal, D and Williams, C (eds) 2014, *Shell Structures for Architecture: Form Finding and Optimization*, Routledge, London ; New York

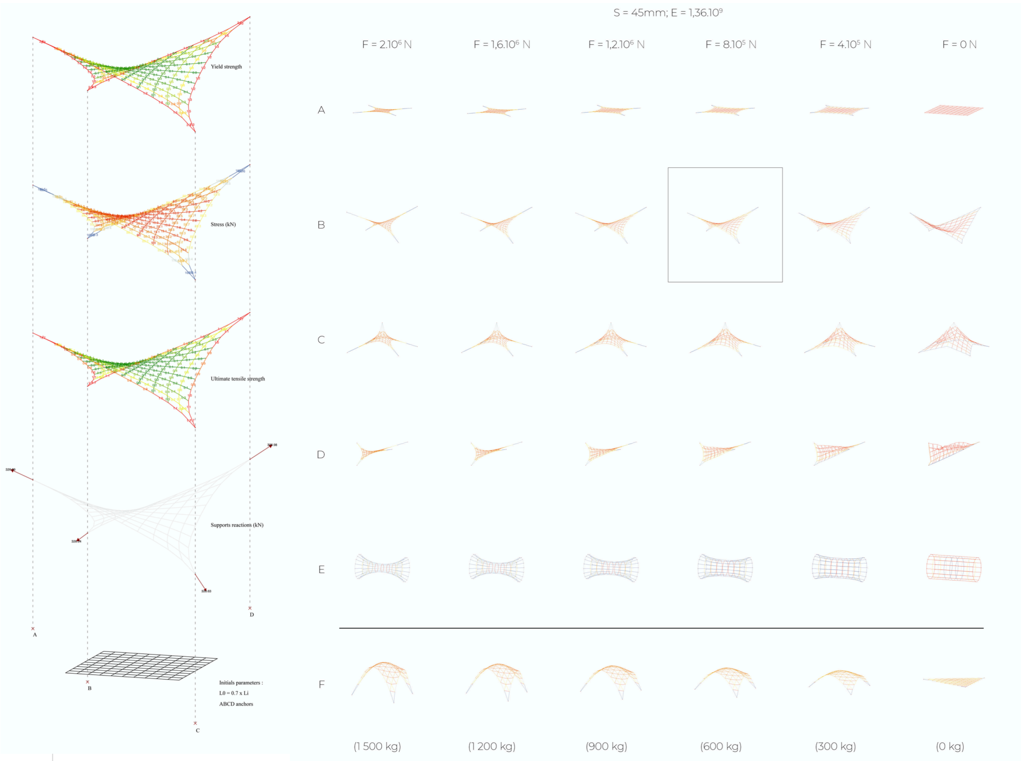
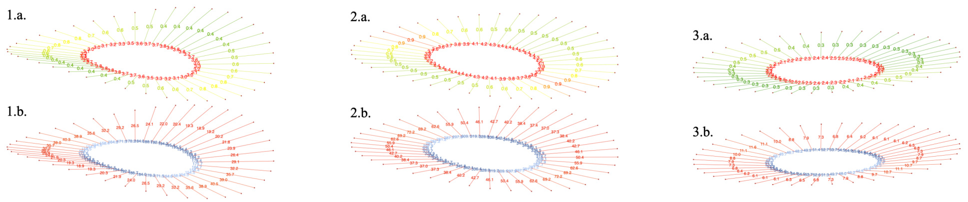


Figure 10
10a. Isometric view of the physical rationalisation layers. 10b. Form-Finding with case study 2's algorithm: Two-dimensional population.



1. *Δz = 25 m (double courbure); F = 25 000 N; S = 0,030 m²; MASS = 772 094 kg*

2. *Δz = 12,5 m (double courbure); F = 5 000 N; S = 0,047 m²; MASS = 1 165 200 kg*

3. *Δz = 30 m (coplanar nodes); F = 25 000 N; S = 0,013 m²; MASS = 326 406 kg*

Figure 11
Three individuals with details of the #F# and #F_m# layers.

Ahlquist, S, College, T and Menges, A 2013, 'Frameworks for Computational Design of Textile Micro-Architectures and Material Behavior in Forming Complex Force-Active Structures', *ACADIA*, 33, pp. 281-292

Bin, L, Ming-min, D, Li-feng, H and Zheng-xing, G 2018, 'Structural Optimization of Spoke Single-Layer Cable-Net Structures Based on a Genetic Algorithm', *Journal of Aerospace Engineering*, 31(3), pp. 917-929

Bonavia, E, Vecvagare, S, Körner, A, Wood, D and Knippers, J 2019 'Point load Actuated Adaptive Gridshells: Methods for initial design and modeling', *Proceedings of IASS Annual Symposia*, pp. 1-8

Brown, N and Mueller, C 2017, 'Designing With Data: Moving Beyond The Design Space Catalog', *ACADIA*, 37, pp. 154-163

Carpo, M 2012, *The Digital Turn in Architecture 1992-2012: AD Reader.*, John Wiley \& Sons, Chichester

Douthe, C 2007, *Etude de structures élancées précontraintes en matériaux composites, application à la comception des gridshells*, Ph.D. Thesis, ENPC

Frei, O 1969, *Tensile Structures Volume 2 Design, Structure, and Calculation of Buildings of Cables, Nets, and Membranes*, M.I.T. Press

Gaudillière, N 2020 'Evolutionary Tools and the Practice of Architecture: from Appropriated Typology to Becoming the Black Boxes of CAAD', *Proceedings of the 2020 BIODIG Conference*

Hooke, R 1678, *Lectures de Potentia Restitutiva, Or of Spring Explaining the Power of Springing Bodies*, John Martyn

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

Melville, S, Brandt, C and Harding, J 2017, 'Calibrated modelling of form-active structures', *IABSE Symposium Report*, 108, pp. 155-156

Otter, J and Day, A 1960, 'Tidal computations', *The Engineer*

Shi, JX, Wu, Z, Tsukimoto, S and Shimoda, M 2018, 'Design optimization of cable–membrane structures for form-finding and stiffness maximization', *Composite Structures*, 192, pp. 528-536

Suzuki, S, Slabbinck, E and Knippers, J 2017 'A comparative overview of generative approaches for computational form-finding of bending-active tensile structures', *IASS 2017*

Veenendaal, D and Block, P 2012, 'An overview and comparison of structural form finding methods for general networks', *International Journal of Solids and Structures*, 49(26), pp. 3741-3753

[1] https://www.food4rhino.com/app/kangaroo-physics

[2] http://adventures-of-kk.blogspot.com/2012/05/day-8-bike-tour-gaudi.html

[3] https://vimeo.com/169692252

[4] https://github.com/CecilieBrandt/K2Engineering

[5] https://github.com/Dan-Piker/K2Goals

[6] https://docs.google.com/document/d/1X-tW7r7tfC9duICi7XyI9wmPkGQUPIm_8sj7bqMvTXs/edit?usp=embed_facebook

[7] http://www.matweb.com/index.aspx

[8] https://seminairemateriaux.wordpress.com