



HAL
open science

Heuristique du franchissement : conception d'un processus génératif pour une passerelle piétonne

Lionel Clot

► **To cite this version:**

Lionel Clot. Heuristique du franchissement : conception d'un processus génératif pour une passerelle piétonne. Architecture, aménagement de l'espace. 2017. dumas-02280368

HAL Id: dumas-02280368

<https://dumas.ccsd.cnrs.fr/dumas-02280368>

Submitted on 6 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

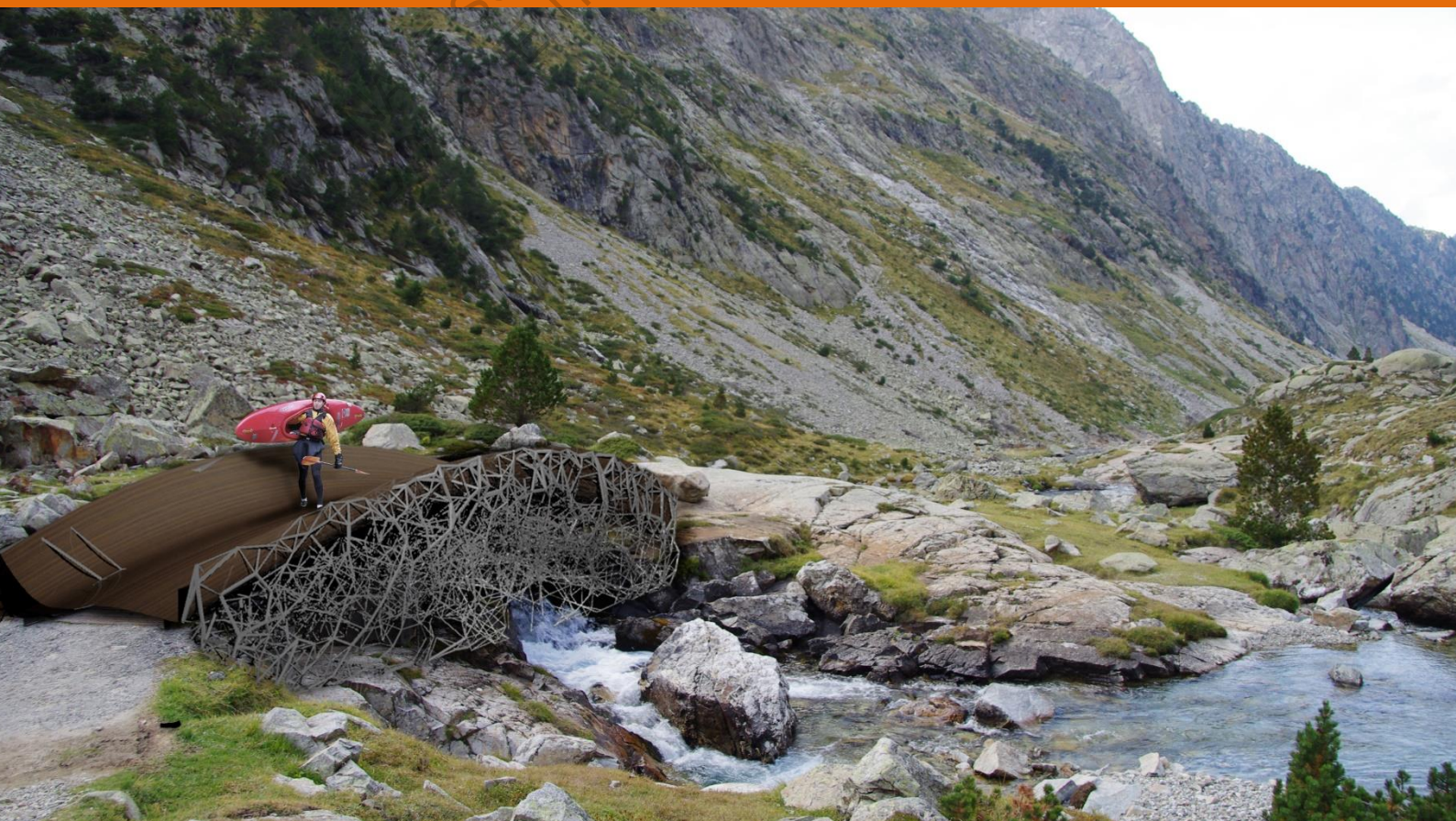
HEURISTIQUE DU FRANCHISSEMENT

Conception d'un processus génératif pour une passerelle piétonne.

Lionel CLOT

Université Fédérale Toulouse Midi-Pyrénées, Ecole Nationale Supérieure
d'Architecture de Toulouse, septembre 2017

Séminaire Architectures Numériques dirigé par Bernard Ferrière, Jean-Pierre
Goulette (encadrant) et Sandra Marquès (encadrant)





ECOLE NATIONALE SUPERIEURE D'ARCHITECTURE
DOCUMENT SOUMIS AU DROIT DE LA PROPRIÉTÉ INTELLECTUELLE

SOMMAIRE

Sommaire.....	3
Introduction :.....	4
I. Etat de l'art.....	8
I. 1. Evolution des outils.....	8
I.1.1. Le crayon	8
I.1.2. L'ordinateur.....	8
I.1.3. Le dessin en 3D	9
I.1.4. Le BIM	9
I.1.5. Le paramétrique	11
I.1.6. Le génératif.....	13
I.1.7. Les robots en architecture.....	21
I. 2. Modèle de calcul (Karamba3D).....	24
II. Développement.....	26
II. 1. Application à un élément simple.....	26
II. 2. Pont en coque à double courbure.....	29
II. 3. Processus génératifs.....	34
II.3.1. Optimisation topologique,.....	34
II.3.2. Recherche de forme : La conception d'une paroi de partition pour Airbus A320 38	
II.3.3. Optimisation des sections.....	39
II. 4. Création d'un algorithme génératif et formalisation	40
III. Comparaison et conclusion.....	60
Bibliographie.....	64
Table des illustrations.....	66
Lexique.....	68

INTRODUCTION :

L'architecture est pour moi la discipline de la synthèse : tant entre les personnes, qu'entre les processus, les matériaux, les coûts et le budget, les souhaits d'un maître d'ouvrage, les choix esthétiques. On reproche parfois aux formations à l'architecture françaises d'être trop artistiques, et pas assez techniques, notamment lorsqu'on les compare aux allemandes ou aux suisses par exemple. Celles-ci incluent dans les cursus une grande part de ce qui, en France, relève de l'ingénierie.

Pourtant, je considère que l'architecture inclut à la fois cette dimension artistique, non-mathématique, créative, relative du cerveau droit, à la pensée arborescente ; et la dimension logique, rigoureuse, absolue du cerveau gauche, à la pensée linéaire. C'est pourquoi j'ai choisi de suivre le double cursus mis en place il y a presque vingt ans par l'Ecole Nationale Supérieure d'Architecture de Toulouse et l'Institut National des Sciences Appliquées de Toulouse, afin d'acquérir ces deux modes de pensée.

Après avoir obtenu un Diplôme Universitaire de Technologie en Génie Civil à l'Université de Reims-Champagne-Ardenne, j'ai poursuivi mes études à l'ENSA de Toulouse. Arrivé en quatrième année, j'ai dû choisir un séminaire : il s'agit d'une sorte de spécialisation, d'une unité d'enseignement, qui a pour objectif la rédaction du présent mémoire.

J'ai donc choisi d'intégrer le séminaire « Architectures Numériques », encadré par Bernard Ferriès, Jean-Pierre Goulette et Sandra Marquès. Les encadrants ayant des parcours assez divers, en informatique, en ingénierie, en architecture, le séminaire est à leur image, ouvert, trans- et inter-disciplinaire ; donc le plus proche de ma définition de l'architecture¹.

Depuis toujours j'ai également une certaine attirance pour la robotique, que je considère comme un moyen de libérer l'homme d'un travail. Il s'agit d'une réflexion déjà menée par des philosophes comme Nietzsche ou Marx à la fin du XIXème siècle, lors d'une révolution industrielle.

¹ Nous retrouvons d'ailleurs souvent le terme d'architecte informatique pour les concepteurs-programmeurs dans les technologies de l'information.

« Du moment où le travail commence à être réparti, chacun entre dans un cercle d'activités déterminé et exclusif, qui lui est imposé et donc il ne peut s'évader ; il est chasseur, pêcheur, berger [...] et il doit le rester sous peine de perdre les moyens qui lui permettent de vivre. »¹

Le travail d'un ouvrier qui ne fait qu'appliquer est donc ici aliénant, et non plus libérateur comme il l'était jusqu'à présent et tel que Hegel le décrivait. Je considère donc que tout travail d'application pure ne laissant pas place à l'improvisation, l'imagination, l'intelligence humaine, ne doit pas être un travail effectué par un homme, mais par une machine. Cette machine apportera une haute précision, et des moyens nouveaux de faire, bien sûr avec de nouvelles contraintes. Ces contraintes sont cependant les mêmes que celles apportées par un simple nouvel outil comme un marteau ou un tournevis. De même pour un robot : elles font partie de la façon de penser un projet, de l'idée de celui-ci.

Ces robots constituent la 4^{ème} révolution industrielle telle que les fabricants de robots KUKA la décrivent. Elle a déjà eu lieu dans les secteurs de l'automobile et de l'aéronautique, et atteint progressivement la médecine. Malgré la forte industrialisation du bâtiment, avec la standardisation des éléments, elle n'a pas encore véritablement touché le milieu de la construction, qu'il s'agisse du bâtiment ou des travaux publics.

Il y a actuellement un essor de la conception assistée par ordinateur (CAO). Nous avons d'abord transposé la table de dessin sur un ordinateur avec les logiciels de DAO dans les années 80 et depuis, avec la CAO en 3D, puis le BIM, nous nous approchons d'une pré-construction, numériquement, du bâtiment. Mais cette dernière se limite à un savoir-faire déjà existant, la conception numérique peut aller plus loin, si elle devient algorithmique, nous retrouvons une façon de faire qui se libère de l'industrialisation, de la standardisation, nous retrouvons une succession de tâche, d'actions, qui mènent à l'objet. Bien que certains aient parfois l'impression que l'ordinateur soit aliénant, restreignant le concepteur, il s'agit de l'apprentissage d'un nouvel outil, permettant de nouvelles libertés de conception, nous redevenons artisans de la construction.

¹ « L'Idéologie allemande », Karl Marx, Friedrich Engels (1845), dans Philosophie, Karl Marx, Maximilien Rubel, éd. Gallimard, coll. « Folio », 1982 (ISBN 2-07-032839-2), p. 319

Quels sont donc ces nouvelles libertés, quels effets ont-elles sur le projet final, ou sur le processus qui mène à sa conception, à sa réalisation ?

De la même façon qu'un ébéniste pourra utiliser un tour, un ciseau, une scie, de la colle, etc. ; l'utilisateur extrudera, intersectera, surfacera, découpera... Mais il existe également une façon de concevoir relativement récente, il s'agit de la conception générative. « *Form ever follows function* »¹ (la forme découle toujours de la fonction) disait Louis Sullivan, qui a été maître de Frank Lloyd Wright. Bien qu'apparue près d'un siècle plus tard, la conception générative suit totalement ce précepte : la forme n'est qu'un résultat, une conséquence, elle n'est pas la cause du travail, elle devient presque secondaire, ce qui compte, c'est la fonction. Le concepteur, en conception générative conçoit donc le processus qui transforme la fonction en forme, en se libérant de tous les paramètres générant la forme, qui deviennent alors gérés par l'ordinateur, et en se concentrant sur la fonctionnalité du *dessin*.

Comment concevoir en conception générative ? Quel impact aurait-elle sur la conception et la réalisation d'une passerelle ?

La conception générative étant une discipline assez récente, il y a relativement peu de bibliographie à ce sujet, pourtant, des designers comme Joris Laarman avec son projet MX3D Bridge à Amsterdam l'appliquent à l'architecture. Les développeurs de logiciels s'intéressent donc à ce sujet, en particulier l'éditeur Autodesk avec leur projet de recherche *Dreamcatcher*, qui est à l'origine d'un grand nombre d'applications aux domaines variés, et donc à même de se pencher sur cette discipline trans-disciplinaire. Toujours dans le milieu de la construction, nous retrouverons également *Institute of Advanced Architecture of Catalunya* qui a imprimé un pont en avec une imprimante 3D à béton fibré en décembre 2016. Enfin, il y a actuellement des recherches en musicologie pour imiter des morceaux baroques, ou en art avec la peinture d'un nouveau « Rembrandt » en 2016 dans le cadre du projet *The Next Rembrandt* : des chercheurs ont analysé, et appris à un ordinateur en *deep learning* (une forme d'intelligence artificielle) à imiter le style de Rembrandt. Enfin, nous pouvons également citer en exemple la façon

¹Louis H. Sullivan *The Tall Office Building Artistically Considered*. Lippincott's magazine, 1896, p403-409.

dont les chercheurs de Google *Deepmind* ont simulé l'apprentissage de la marche avec des animaux virtuels... ainsi que les nombreuses recherches basées sur les réseaux de neurones.

La conception générative constitue une sorte de sérendipité provoquée : nous nous plaçons dans un cadre de perte de contrôle, nous ne cherchons qu'à obtenir un résultat fonctionnel, la forme importe peu. Nous sommes donc à même d'obtenir des résultats peut-être très classiques. En effet la théorie de l'évolution de Darwin stipule qu'avec le temps tout s'améliore et donc qu'un pont « classique », avec de beaux arcs, serait une bonne solution au problème posé « *franchissement de la rivière* ». Mais le résultat reste incertain : nous devons accepter une perte de contrôle pour que puisse s'exprimer pleinement le potentiel de l'inconnu.

Après avoir dressé un état de l'art et décrit les outils nécessaires à la conception générative ainsi que leurs limites, nous expliciterons les différentes formes que la conception générative peut prendre, de l'optimisation à la recherche de forme.

Nous procéderons ensuite par expérimentation en mettant à l'œuvre ces nouveaux outils dans un cas concret : la « conception » d'une passerelle d'une portée de 25m. Nous devons évidemment faire des choix limitant les possibilités à des formes réalistes, qui puissent être ensuite réalisées. Nous conclurons enfin sur cette étude de cas en comparant ce mode de conception avec les méthodes plus courantes, nous efforçant ainsi de mettre en lumière les impacts de la conception générative en termes de forme sur le projet.

I. ETAT DE L'ART

I. 1. EVOLUTION DES OUTILS

I.1.1. Le crayon

Depuis quelques années, nous observons en France et à l'international une nouvelle révolution de l'architecture. A la fin du siècle dernier, l'ordinateur est venu bouleverser un modèle de conception traditionnel : le papier et le crayon. Autrefois, nous dessinions plans, coupes, perspectives à la main. Bien sûr, cela avait un impact fort sur les formes des bâtiments, celles-ci restaient relativement simples en projection sur les plans habituels. Le dessin à la main impliquait une grande rigueur, mais également une grande liberté grâce à un outil presque instinctif. Enfants déjà nous gribouillons, dessinons dans la terre ou le sable - ou au grand dam de nos parents, sur les murs. Le crayon ou stylo est alors un moyen de garder une trace du mouvement de la main. Il est aussi un moyen de communication, puisqu'il sert avant tout à montrer.

Il y a des siècles, lorsque nous construisions par des moyens traditionnels, nous ne nous embarrassions pas de plans : la conception se faisait en même temps que la réalisation. Peu à peu, des méthodes mécaniques et industrielles se développant, le dessin a permis d'explicitier l'assemblage de tous les matériaux.

I.1.2. L'ordinateur

Peu à peu, le papier est devenu écran, et le crayon est devenu bouton. La main s'est posée sur la souris. Cela traduit aussi une évolution de la dimension du *dessin*¹. En effet, le papier admet une légère épaisseur : les traces effacées, les traits plus ou moins

¹ Nous utiliserons le mot *dessin*, en italique, pour définir une conception, une forme, un dessin, un modèle. Au final, nous lui donnons, lorsqu'il est en italique, le sens du mot anglais *design*, qui n'inclue pas de précision concernant une dimension. En effet, un dessin est nécessairement bidimensionnel, alors qu'un *design* – *dessin*, donc – peut être un concept plan, spatial, systémique...

énergiques... L'ordinateur ne permet plus cela, tout doit être parfait, identifié par des coordonnées. Mais cette épaisseur est remplacée par le temps : Nous pouvons annuler nos actions sans avoir à les effacer... il suffit de cliquer sur un bouton !

1.1.3. Le dessin en 3D

Le modèle géométrique¹

Sur l'ordinateur, toujours, des logiciels comme le célèbre Sketchup permettent d'esquisser des formes tridimensionnelles, en laissant toujours possible et aisée l'annulation de ses actions. Les lignes et surfaces du *dessin* peuvent cette fois devenir des formes plus complexes, tridimensionnelles.

1.1.4. Le BIM

Le modèle géométrique sémantique²

Plus que des lignes, surfaces, formes, il s'agit cette fois d'objets auxquels nous donnons un nom :

- Dessinons une ligne (figure 1). Elle est liée à un plan de construction, le rez-de-chaussée par exemple.
- Cette ligne est en fait un axe. (figure 2)
- L'axe porteur d'un mur de 2.73m de haut. (figure 3)
- Ce mur est composé de 20cm de béton à l'intérieur, d'un peu de peinture, et de tout un complexe isolant et de parement à l'extérieur. (Figure 4)
- Tous ces éléments sont eux-mêmes caractérisés : ils ont un cout, une résistance au feu, un pouvoir isolant (phonique, thermique, hydrique...). (Figure 5)

¹ ²Modèles du BIM tels qu'identifiés par Gilles Halin et Mohamed-Anis Gallas, « *Une approche pédagogique par les modèles pour la sensibilisation au concept de BIM (Maquette Numérique)* » In Jean-Pierre Goulette (dir), Bernard Ferriès, (dir) *SCAN'16 Toulouse, Séminaire de conception architecturale numérique: Mètre et paramètre, mesure et démesure du projet*, PUN – Editions universitaires de Lorraine, 2016

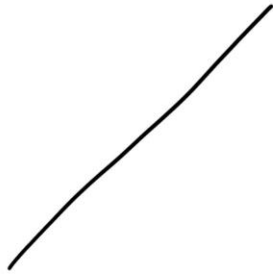


Figure 1 : Première étape : la ligne



Figure 2 : Deuxième étape : l'axe

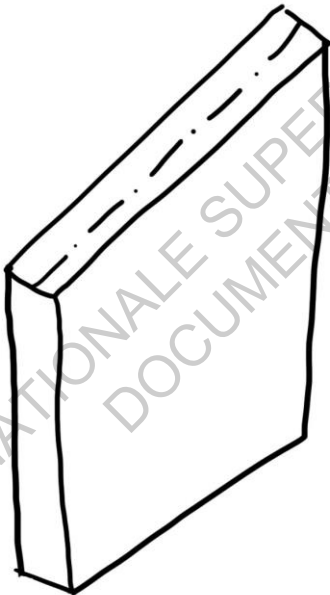


Figure 3 : Troisième étape : le mur

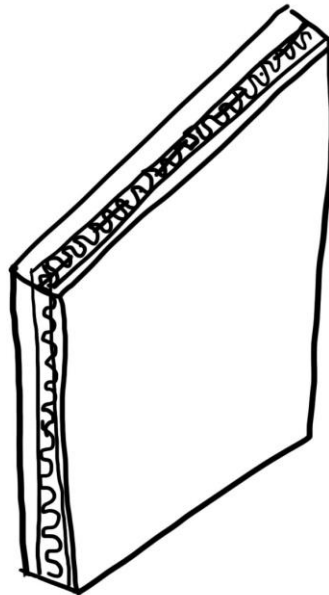


Figure 4 : Quatrième étape : le mur composite

Le modèle géométrique sémantique enrichi, et partagé¹

Ainsi, les formes, la géométrie deviennent des éléments enrichis en informations, nous parlons désormais de sémantique. Ce qui serait un parallélépipède dans Sketchup est en fait un mur. C'est le rôle du format libre IFC : la géométrie est sémantisée.²

1.1.5. Le paramétrique

Le modèle de processus³

En paramétrique, le concepteur ne conçoit plus tant un *dessin* que le *processus* qui le génère. Cela oblige à rationaliser, linéariser une idée⁴. Comme le papier permettait de transmettre une idée, le paramétrique oblige à l'écrire et à la décomposer.

Dans l'exemple de notre mur,

- il s'agit de créer deux points à partir de coordonnées. Nous relierons ensuite ces points par une ligne. (Figure 6)
- Nous créons ensuite un rectangle à partir de cette ligne. Et en extrudant ce rectangle, nous obtenons un parallélépipède rectangle. Enfin, nous le définissons comme étant un mur, c'est la sémantisation (Figure 7)

Mais le principe du paramétrique, c'est que quel que soit l'état d'avancement du processus de conception, il est toujours possible de modifier les paramètres d'origine, par exemple, ici, les coordonnées des points. (Figure 8)

On peut facilement comparer le paramétrique à une feuille de calcul Excel : si l'on fait les opérations mathématiques manuellement, nous obtenons à la fin un résultat. Mais si l'on écrit ces opérations dans les cellules, en faisant référence à d'autres cellules, le résultat s'affichera automatiquement. Et si l'on fait évoluer la valeur d'une des cellules, le résultat évolue également en conséquence !

¹ et ³ Modèles du BIM tels qu'identifiés par Gilles Halin et Mohamed-Anis Gallas, « Une approche pédagogique par les modèles pour la sensibilisation au concept de BIM (Maquette Numérique) » In Jean-Pierre Goulette (dir), Bernard Ferriès, (dir) SCAN'16 Toulouse, Séminaire de conception architecturale numérique: Mètre et paramètre, mesure et démesure du projet, PUN – Editions universitaires de Lorraine, 2016

² *Industry Foundation Classes*

⁴ Telle que définie par Gilles Deleuze dans Pierre André Boutang L'Abécédaire de Gilles Deleuze, 1996

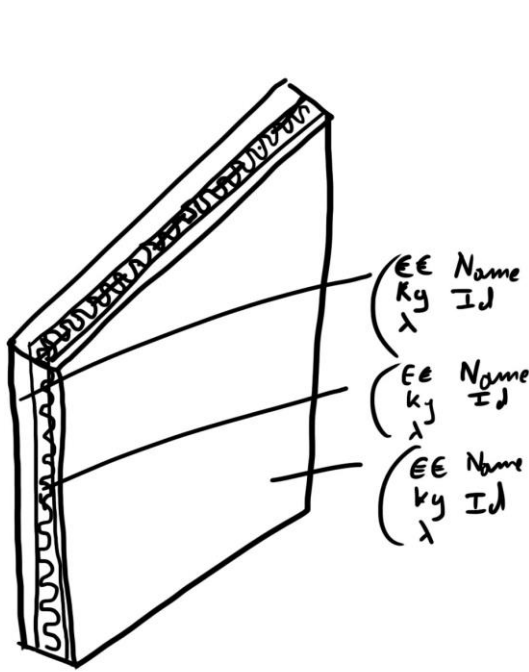


Figure 5 : Le mur enrichi

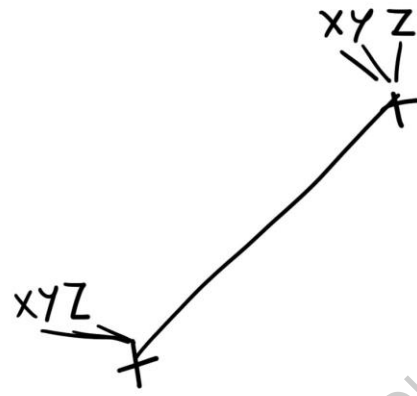


Figure 6 : Les paramètres

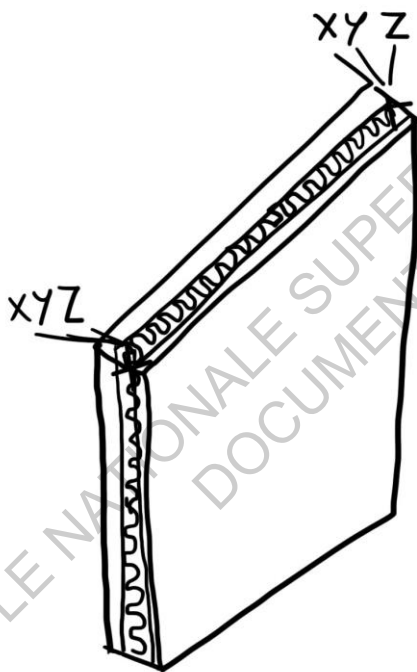


Figure 7 : Le mur paramétré

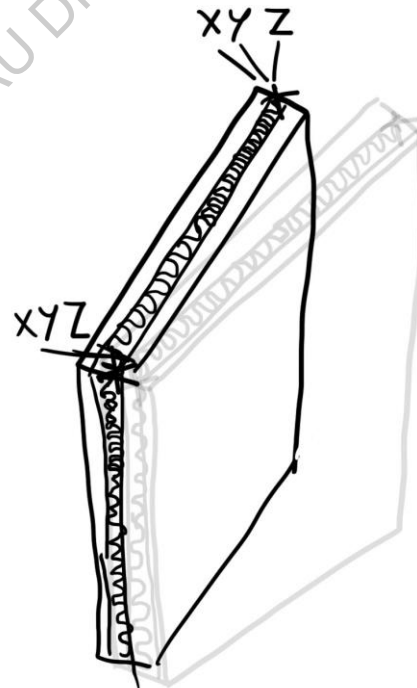


Figure 8 : Le mur avec les paramètres modifiés

1.1.6. Le génératif

« *Survival of the fittest* »¹

Le concepteur ne conçoit presque plus du tout le *dessin*, mais seulement le processus, et un système d'analyse.

Nous avons déjà décrit le processus pour le paramétrique, il est ici assez semblable. Mais comme pour le BIM, il va falloir nommer les éléments :

- Le résultat est appelé individu.
- Les paramètres, individuellement correspondent à des gènes.
- L'ensemble des paramètres constituent un génome.
- Le système d'analyse définit une aptitude (*fitness*).

L'objectif du génératif est de trouver la meilleure solution à un problème donnée non par une analyse cartésienne mais par heuristique.

Chaque gène ne permet pas en lui-même de résoudre le problème, mais c'est la combinaison de l'ensemble des gènes qui le permet. La pratique évolutionnaire ne permettant pas comme en mathématique d'obtenir la solution parfaite, mais simplement un suffisant. En trouvant la meilleure combinaison de gènes possibles nous définissons « l'aptitude », il s'agit de la capacité d'un individu à répondre au problème.

Ainsi, par l'intermédiaire du processus, le génome constitue un individu qui est placé dans un environnement. L'environnement regroupe l'ensemble des contraintes pour lesquelles chaque individu aura une certaine aptitude définie par ses gènes à survivre – c'est la « survie du plus apte ».

Face à un environnement les individus de la génération n sont plus ou moins aptes (donc proche de la solution). Le principe est ici de tuer, de supprimer, ou en tout cas de retirer du calcul les moins aptes (les moins bien notés), et faire se reproduire ensemble ceux qu'il reste. Leur descendance, la génération $n+1$ sera donc plus apte. Il suffit ensuite

¹ Herbert Spencer *Principles of Biology*, 1864 résumant Charles Darwin *L'origine des espèces* (sélection naturelle)

de reproduire ce processus un certain nombre de fois. À chaque génération, les individus seront plus aptes.

Le système d'analyse consiste ensuite à donner une note à chaque individu. Il s'agit d'un nombre réel. Dans notre étude de cas (voir partie 4.), cette note correspondra à la flèche au centre de notre objet.

Prenons un exemple :

Il y a trois gènes à l'origine de la génération d'un individu :

- La taille des pattes
- Le nombre de pattes.
- La taille du corps.

On applique à ce génome le processus qui constitue un individu. Jusque-là, tout est semblable au paramétrique.

Mais nous allons alors faire suivre un parcours à cet individu¹ et noter sa capacité à évoluer dans le parcours en fonction des valeurs de ses gènes. Entreront en compte, par exemple, la distance parcourue dans un laps de temps et l'énergie dépensée dans le processus.

Prenons 50 individus générés complètement aléatoirement. L'un possèdera par exemple une centaine de grosses pattes sur un tout petit corps, un autre une seule patte, un ressemblera à un teckel avec quatre pattes mais un corps de plusieurs mètres de long, une autre à un cheval avec quatre pattes assez bien proportionnées par rapport au corps, un autre à chien de berger, un autre à un insecte avec six petites pattes sur un petit corps, et encore 44 autres variantes. Nous leur fait faire le parcours et nous les notons. Nous observerons alors que certains auront été avantagés : le cheval et le chien de berger par exemple, mieux proportionnés. Au contraire, le teckel aura eu beaucoup de difficultés à avancer.

¹ Littéralement dans le cadre du projet *Deepmind* de Google

Voilà la fin de la première génération. Il s'agit maintenant de faire se reproduire entre eux les individus pour en obtenir de meilleurs, à nouveau 50. Nous pouvons nous douter que les plus mauvais ne donneront pas une descendance particulièrement bonne. Alors nous ne sélectionnerons que les meilleurs pour se reproduire entre eux. Et produire une seconde génération, qu'on fera ensuite se reproduire de la même façon, dans une seconde itération. Mais cela risque de produire ce que l'on appelle des *optima locaux*.

Tous les animaux que l'on trouve dans la nature sont des *optima locaux* : ils sont des solutions (individus) à un environnement (c'est-à-dire un système de notation) auxquels ils sont adaptés (fitness). Le cheval est adapté à son environnement, dans lequel il survit bien et avance bien. La fourmi également. Les deux reçoivent donc une bonne note. Mais si nous mélangeons le cheval et la mouche ensemble, nous risquons d'obtenir un individu étrange, qui ne sera plus du tout adapté à son environnement ! Nous ferons donc se reproduire le cheval avec un autre individu assez proche de lui génétiquement, quitte à ce que sa note soit plus faible. Ainsi, le cheval et l'âne feront ensemble une mule. Et cette mule pourra peut-être avoir la meilleure note, être le maximum.

La conception générative part du principe qu'avec un nombre infini d'individus, et un nombre infini d'itérations, nous obtiendrons le résultat parfait. Mais comme nous sommes des êtres finis dimensionnellement, nous ne pouvons pas nous permettre des infinis. Les mathématiciens chercheraient à obtenir ce résultat parfait par une méthode analytique. Nous, nous limiterons les individus et les itérations, en adaptant nos critères de reproduction, pour qu'après un temps raisonnable, nous obtenions un résultat acceptable. Il s'agit en quelque sorte du « nécessaire et suffisant ». Lorsque nous concevons un pont, il est à la fois nécessaire et suffisant qu'il résiste au passage des voitures, par exemple.

Grasshopper

C'est un plug-in du Logiciel Rhinocéros 3D. Il permet de faire de la programmation graphique, c'est-à-dire que plutôt que d'écrire de longues lignes de codes l'utilisateur place des blocs les uns après les autres. Chaque bloc a une fonction précise, à sa gauche, nous le nourrissons avec des données entrantes (input), qu'il interprète et offre à

disposition, à sa droite d'autres données sortantes (*output*). Dans l'exemple de notre bloc « *List item* », celui-ci prendra le n-ième (valeur de type nombre donnée par l'*input* « *item* ») élément de la liste (*input* « *list* »), *warp* étant ici inutilisé (si nous demandons le 11^{ème} élément d'une liste qui ne comporte que 10 élément, le bloc ne retournera aucune information si *warp* est défini sur *false*, et retournera le 1 élément si sa valeur est *true*), et le retournera dans l'*output* « *i* » (pour *item*).

Chaque bloc est organisé dans un onglet (*Params, Maths, Sets, Vector, Curve, Surface, Mesh, Intersect, Transform, Display*) et une sous partie. Ainsi, notre bloc *List item* se situe dans l'onglet « *Sets* » (« ensemble » en français) et la partie « *List* ».

Chaque onglet regroupe ainsi un ensemble de blocs avec des fonctions différentes, d'analyse mathématique, de gestion de liste, de création géométrique...

L'onglet *Params* renferme ce à quoi Galapagos peut principalement toucher : les « *sliders* ». Ceux-ci ressemblent fortement aux outils de gestions du volume sur nos ordinateurs : nous faisons se déplacer un point entre des bornes (de 0 à 100 souvent), le point prenant alors une valeur. Nous définissons donc le *slider* par ses bornes, et le nombre de décimales. Ces *sliders* sont donc des gènes avec un fonctionnement qui varie légèrement de ce qu'on observe en biologie : nos *sliders* renvoient un nombre réel, naturel ou entier. Dans la nature, les gènes fonctionnent plus comme des interrupteurs : soit oui, soit non – *true* ou *false* dans un langage de programmation. Ils sont cependant excessivement nombreux¹, et il est rare qu'il n'y ait qu'un seul gène codant pour un caractère^{2 3}. Dans Grasshopper, le fonctionnement est assez opposé : nous limitons au maximum le nombre de *sliders*/paramètres, puisqu'un trop grand nombre devient difficile à appréhender intellectuellement, ou même à concevoir !

¹ L'étude du poisson Tetraodon nigoviridis dont nous sommes relativement proches génétiquement par l'équipe du Pr. Jean Weissenbach fait état d'un nombre entre 28000 et 34000. * Nature Genetics, Volume 25 Number 2 Page 235 - 238 (2000)

² Nous pouvons faire l'analogie avec notre processus paramétrique : un gène ou ensemble de gène codant pour un caractère, correspond à un paramètre ou un ensemble de paramètres générant un caractère (courbure, hauteur, etc.). Nous avons ainsi l'habitude de nommer le paramètre par son caractère : le paramètre dictant le rayon d'un arc sera simplement appelé rayon de l'arc.

³ C'est le cas pour les fossettes, les taches de rousseurs, ou l'attachement des lobes d'oreilles. McKusick, Victor A. (25 June 1994). "Dimples, Facial". Online Mendelian Inheritance in Man. Johns Hopkins University. 126100.[permanent dead link] ; Genetics/Reproduction". ScienceNet – Life Science. Singapore Science Centre. Archived from the original on 2003-09-25.

Les données peuvent prendre de nombreuses formes : nombres, textes, *mesh*, surfaces, volumes, plans, points... qui sont organisées en listes. Ces listes qui sont-elles mêmes organisées au sein d'un arbre pour composer une liste de liste, et au-delà si nécessaire. Grasshopper ne fera pas de différence de nomenclature quel que soit le degré de la liste. Ainsi, chaque donnée peut-être retrouvée ou triée selon des coordonnées.

Galapagos

Ce bloc grasshopper fonctionne légèrement différemment : Celui-ci n'a pas de bornes output, mais deux pseudo-bornes d'input :

- *Fitness*, qui va chercher un nombre à l'*output* d'un autre bloc.
- *Genome*, qui fait varier un ou plusieurs *sliders*.

Il faut ensuite double cliquer sur le bloc pour accéder à un menu de paramétrage, qui permet de choisir si nous souhaitons minimiser ou maximiser le nombre entrant dans fitness, et de paramétrer le fonctionnement évolutionnaire ou recuit. Le fonctionnement évolutionnaire est ici celui qui nous intéresse, mais Galapagos permet aussi l'« *annealing simulation* ». « *annealing* » (recuit en français) étant une procédure issue de la métallurgie dont l'objectif est de faire doucement descendre la température du métal pour que les atomes migrent vers une structure cristalline (plus solide). Dans Galapagos, le processus est le même : les individus migrent vers le haut de la montagne, vers un Fitness plus élevé.

Pour l'algorithme évolutionniste, Galapagos propose plusieurs paramètres :

- *Max* : le nombre maximal d'individus dans une génération
- *Population* : le nombre visé d'individus dans une génération
- *Initial boost* : le multiplicateur pour le nombre d'individus de la première génération
- *Maintain* : le pourcentage d'individus maintenus d'une génération à la seconde.
- *Inbreeding* : pourcentage de gènes en commun entre deux individus se reproduisant. (tout en gardant en compte que le développeur, David Rutten, a choisi de maintenir une part de reproduction aléatoire)

Dans le second onglet, nous retrouvons les résultats de la simulation :

La partie haute est dédiée à l'étude des évolutions : l'abscisse correspond à l'indice de la génération d'individus, en ordonnée nous retrouverons l'aptitude, en rouge la solution optimale, puis en orange et jaune l'étendue des valeurs d'aptitude des autres individus de la génération. La moitié droite, bleutée, correspond à une sélection dont les résultats sont affichés dans la moitié basse.

Dans le tiers gauche de la partie basse, nous retrouvons une sorte de diagramme permettant d'évaluer la présence d'extrema locaux : les bonnes solutions avec des génomes proches se regroupent entre-elles, tandis que les autres restent plus dispersées.

Le tiers central est occupé par une représentation de l'ensemble génomique : chaque trait noir correspond à une courbe reliant chaque valeur d'un gène

Le tiers droit représente des sortes de chromosomes à un seul bras : chaque gène est coloré dans une différente teinte de vert, et plus ou moins large suivant sa valeur. Cependant, il apporte aussi une information majeure : la valeur de l'aptitude.



Figure 9 : Interface du solveur Galapagos, David Rutten

Le Fitness Landscape¹

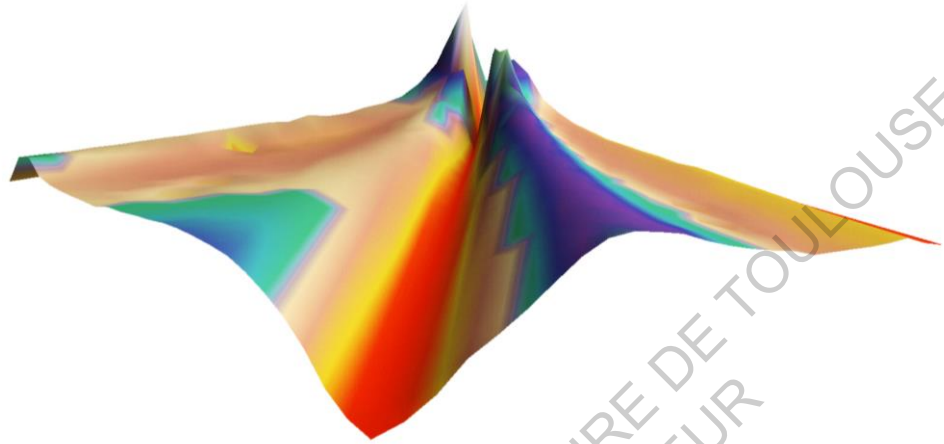


Figure 10 : Exemple d'un diagramme d'aptitude pour la coque à double courbure : nous observons deux optima très proches mais séparés par une vallée.

Normalement, c'est la sélection naturelle qui agit. Il s'agit en fait simplement de réguler qui va pouvoir se reproduire. Celle-ci implique des notions de stérilité, de mort infantile (où l'individu est trop peu apte pour survivre jusqu'à l'âge de reproduction). Elle fait donc uniquement rentrer en jeu l'aptitude.

Mais ici il y a également de la sélection artificielle, comme les hommes l'ont fait depuis des millénaires avec ce(ux) qu'ils consomment (fruits, bétail...) mais aussi ce(ux) qu'ils utilisent (chiens notamment). Il n'y a pas de sexe, de genre...

- La sélection évolutionniste, gérée intégralement par le module, par un savant mélange de :
 - Coalescence hybride *crossover coalescence* : les gènes de l'individu de la génération $n+1$ sont à moitié issu du « père », à moitié issu de la « mère »
 - Coalescence par mélange *blend coalescence* : les valeurs des gènes sont interpolées entre celles du « père » et de la « mère »

¹ Nous vous invitons à vous rendre directement à la partie *pont en coque à double courbure* pour un exemple applicatif.

- Coalescence par mélange variant suivant l'aptitude relative : les valeurs des gènes sont interpolées suivant l'aptitude des parents entre celles du « père » et de la « mère »
- La sélection artificielle : l'homme derrière la machine a la liberté de ré-instituer un individu d'une génération précédente s'il le considère « prometteur ». Le processus n'est donc pas « bête et méchant »

Imaginons un terrain de montagne, à l'échelle des Pyrénées. Celui-ci est peuplé d'animaux dont l'objectif est de monter le plus haut possible (d'être le plus apte), et dont les gènes peuvent être rapportés en coordonnées X et Y. Les Pyrénées peuvent alors être une sorte de diagramme d'aptitude : chaque montagne correspond à un optimum vers lequel les individus de la vallée tenteront de monter. Mais il y a de très nombreuses montagnes, donc autant de relativement bonnes solutions. Mais les animaux qui vivent sur la côte basque auront un génome trop différents de ceux qui vivent proche de l'Aneto (ou du Vignemale si nous voulons rester chauvins.), ils tenteront donc de grimper sur la montagne la plus proche.

Mais si celle-ci est liée par un col assez peu marqué à une autre montagne un peu plus haute, il est probable que les individus migrent sur celle-ci. Cependant, si elle est séparée par une vallée large et profonde, ils n'iront jamais dessus. Si cette vallée est très étroite, nous pouvons imaginer que les animaux « sauteront » sur la montagne d'à côté : les individus de la génération n+1 ont toujours un génome légèrement différent de celui de leurs parents : ils peuvent donc naître en plaine vallée, comme sur la montagne d'à côté, ce qui compte, c'est que les coordonnées X et Y (en quelque sorte les coordonnées GPS) soient proches. Nous pouvons bien évidemment, lors de la simulation de reproduction, faire varier cet aléa génomique, mais cela veut dire risquer de faire naître de nombreux individus très peu aptes.

De la même façon, des individus vivant sur des montagnes différentes ne devraient pas trop se reproduire entre eux : ils sont trop différents. Si nous imaginons un hybride entre une baleine et une fourmi, nous nous doutons bien que l'enfant ne sera pas apte à grand-chose et relativement difforme. Ici, c'est le même problème, si deux individus de

deux montagnes différentes se reproduisent, l'enfant risque de naître en pleine vallée, et donc d'être « tué » par le solveur évolutionnaire qui ne veut garder que les plus aptes.

A contrario, si les individus ne se reproduisent qu'entre eux, avec une très faible « différence génomique », ils ne risquent pas de sauter par-dessus le ravin et arriver sur la montagne d'à côté, ils resteront sur leur montagne, fiers de leur territoire.

La grande problématique des solveurs évolutionnaires est donc de définir cette incestuosité, cette proximité génomique acceptable. Mais il s'agit aussi de définir le nombre d'individus : s'il y a une montagne très fine mais très haute, comme une tour, il y a peu de chances qu'un individu y grimpe.

Il faut donc augmenter ce nombre de chances : soit en ajoutant des individus à chacune des générations, soit en augmentant le nombre de génération ; dans tous les cas cela mobilise des ressources de calculs à nos ordinateurs et nous souhaitons les abaisser au maximum. Nous rappelons en effet que l'objectif de la conception générative est d'obtenir le meilleur suffisant.

1.1.7. Les robots en architecture

A l'heure actuelle, les robots n'ont qu'une très faible place en architecture. Tout juste se limite-t-on à des machines qui automatisent une partie du travail. La perceuse en est l'exemple le plus rudimentaire, mais CAT (Caterpillar) développe également désormais des guidages automatiques de l'élévation et du dévers pour ses niveleuses. L'architecture en est à un stade d'évolution assez semblable à celui de l'agriculture.

La troisième révolution industrielle, expliquée par Jeremy Rifkin, c'est-à-dire celle des technologies de l'information et de la communication n'a pas véritablement touché le milieu de l'architecture. Nous construisons toujours « sur plan », manuellement. Malgré quelques évolutions, il reste très rare de voir des tablettes sur un chantier, les prises de notes se font même majoritairement toujours sur papier¹, la quasi-totalité de la numérisation du travail se fait en dehors du chantier, au « bureau », qu'il s'agisse de conception ou de gestion du chantier.

¹ Constructys, *Etude d'impact de la transition numérique sur le secteur de la construction*, Juin 2017

Ce sont actuellement le second-œuvre et les travaux publics qui mènent ces transformations, puisqu'ils arrivent à standardiser leurs actions, les rendre véritablement répétitives. On tombe alors effectivement dans la division du travail que nous critiquions en introduction, mais celle-ci est particulièrement efficiente et rentable. Peut-être cette révolution n'a-t-elle justement pas véritablement touché ce milieu parce qu'elle ne s'est pas accompagnée du même potentiel de développement économique que les précédentes ?

Cela justifierait-il l'absence de robots, alors qu'on en trouve dans l'automobile et l'aéronautique ?

L'industrie dite « 4.0 », celle de l'automatisation a fortement touché ces deux derniers milieux, en particulier grâce à la soudure : il s'agit d'une technique exigeante en terme de précision, et dont l'efficacité est critique : une mauvaise soudure peut remettre en cause l'intégrité physique de toute une voiture, et que dire d'une fusée. Au contraire, l'assemblage des tableaux de bord se fait toujours souvent manuellement : les câbles ne sont pas toujours au même endroit, il faudrait donc que les robots sachent véritablement les trouver, les attraper, puis les accrocher ensemble, ce qui demande des capacités techniques phénoménales. Alors que les bambins jouent à mettre des cubes dans des trous en forme de carré, il s'agit d'un travail très complexe pour un robot.

Il lui faut comprendre ce qu'est le cube, en tant qu'objet spatial mais aussi comme outil, comme sujet ou finalité.

Il lui faut comprendre ce qu'est le trou.

Il lui faut appréhender physiquement ces deux entités, les reconnaître.

Il lui faut être capable d'attraper le cube.

Il lui faut sentir le cube, s'il l'a attrapé.

Il lui faut savoir diriger, orienter le cube.

Il lui faut savoir s'il a bien effectué la tâche, c'est-à-dire savoir que le cube est dans le trou, alors qu'il ne le voit plus.

L'Association for Robots in Architecture, fondée en 2010 par Sigrid Brell-Cokcan et Hoannes Braumann a justement pour objectif de développer la présence de ces robots dans notre discipline.

Nous sommes libres d'estimer à quel point 2010 est tardif pour ce genre d'association, souvent à l'origine de transformation dans le milieu. Nous pouvons comparer cette date à celle de la fondation de BuildingSMART en 1995, association à l'origine d'une grande partie de l'interopérabilité logicielle dans nos logiciels métiers, et en particulier du format libre IFC.

Le BIM commence tout juste à devenir un élément obligatoire dans les concours et les appels d'offres, plus de vingt ans après que le concept ait véritablement été imaginé, les premières expériences ayant été faites il y a une quinzaine d'année.

Les premières expériences avec des robots se déroulent actuellement. Faudra-t-il donc attendre vingt ans avant que les robots ne se généralisent ?

A notre sens, il suffira de justifier les bénéfices pécuniaires pour que les robots apparaissent. En effet, il faut que la réticence innée au changement soit outrepassée par de réels intérêts. Le gain de productivité du BIM, estimé à environ 20% semble être suffisant pour qu'une réelle dynamique puisse se mettre en place. Il faudrait donc savoir estimer le gain de productivité avec des robots pour qu'ils deviennent communs.

Il existe pourtant déjà tout une nébuleuse d'outils (et d'outils pour robots) plus ou moins robotisés en architecture :

- Le scanner 3D (pour nuage de points et photogrammétrie) type Leica C10
- Les drones (en particulier pour la prise de vues (quad ou hexacopters notamment))
- Robots industriels pour interaction homme-machine type KUKA LBR iiwa
- Robots industriels de grande dimension capables de manipuler plus d'une tonne, comme le KUKA KR 1000 titan
- Les exosquelettes (Eiffage avec Bioservo Technologies AB par exemple)
- Les buses d'impression 3D (plastique, béton, métaux)
- Les embouts de manipulation (pour pliage, assemblage, presse, fil diamanté)

I. 2. MODELE DE CALCUL (KARAMBA3D)

Il s'agit ici de définir l'aptitude (fitness) du modèle, en l'analysant, et en précisant quelles sont nos souhaits. Nous appliquons ces calculs à des systèmes « standard », par exemple un treillis, un *bow string*, un pont à haubans, etc. Nous pourrions envisager de quantifier les coûts de ces structures, et comparer des matériaux.

Le calcul statique s'appuiera sur le plug-in Karamba3D.

Le principal module de calcul que nous utiliserons est « AnalyseTh1 (Karamba) », qui s'appuie sur des calculs du premier ordre, et l'hypothèse des petites déformations. Celle-ci provient de la mécanique des milieux continus, et j'ai la chance d'avoir pu suivre un module sur celles-ci à l'INSA de Toulouse, par Thomas de Larrard.

Ces calculs font appel à des tenseurs, objets mathématiques exprimés dans un espace vectoriel (dans notre cas, l'« espace » ou le « plan »), pour nous, il s'agira donc de matrices.

Le tenseur des déformations, caractérise la variation de distance entre un point M du système et le point M', du système auquel nous aurons appliqué des forces.

En posant $\vec{v} = \overline{MM'}$ le vecteur de déplacement, et en utilisant le tenseur des déformations de Green-Lagrangé, qui permet de visualiser facilement les déformations que nous nommerons ϵ , nous obtenons :

$$\epsilon = \frac{1}{2} \left(\frac{\partial \dot{v}^T}{\partial M} \cdot \frac{\partial v}{\partial M} + \frac{\partial v}{\partial M} + \frac{\partial v^T}{\partial M} \right)$$

Si nous pose l'hypothèse que le système ne subit que de petites déformations, c'est-à-dire que nous sommes dans l'élasticité d'un matériau commun en construction (par exemple de l'acier, si nous utilisons du caoutchouc, cette hypothèse ne serait pas valable.), alors nous négligeons les termes d'ordre 2

Nous utiliserons également le tenseur des contraintes, caractérisant la réaction du matériau face aux forces qui lui sont appliquées : en un point et un plan de coupe, nous obtiendrons des valeurs pour la direction longitudinale, et les tangentes. Nous obtenons

également des rotations, c'est-à-dire des moments de flexion (ou de torsion). Le tenseur des contraintes s'exprime alors ainsi :

$$\bar{\sigma}(P) = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix}_{x,y,z} = (C(M, \vec{x}) \quad C(P, \vec{y}) \quad C(P, \vec{z}))$$

Il faudra diagonaliser cette matrice en $\begin{pmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{pmatrix}_{\epsilon_1, \epsilon_2, \epsilon_3}$

qui nous donne ainsi les contraintes maximales suivant les directions principales ϵ_i .

Nous utiliserons la contrainte équivalente de Von-Mises: définie par $T(P) = \frac{1}{2} \sqrt{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2}$

Exprimé en MPa, nous pouvons la comparer à la résistance élastique. Celle-ci est de l'ordre de 40MPa pour le plastique ABS, fréquemment utilisé en impression 3D, et de 235MPa pour un acier commun, de 2MPa pour un béton ordinaire, et de 3.5MPa pour un béton fibré.

La contrainte équivalente de Von-Mises permet d'approximer la contrainte de rupture du matériau: Celui-ci supportera en effet différemment une contrainte tangentielle et une contrainte axiale.

Nous n'utiliserons ainsi trois critères :

- 1) Le critère de poids : limitant le poids total du pont
- 2) Le critère de plasticité (Von-Mises)
- 3) Le critère de déformation (par exemple 1/200 de la portée)

Il faudra alors choisir un de ces trois critères comme celui qui devra être minimisé, les deux autres servant alors plutôt de pénalité: Par exemple, si la résistance de Von Mises est dépassée par exemple, nous aurons une flèche de 1000 mètres de plus que ce qu'aurait calculé seul Karamba. Ainsi, Galapagos cherchera l'individu avec la flèche la plus faible possible, respectant également la non-plasticité du matériau.

II. DEVELOPPEMENT

II. 1. APPLICATION A UN ELEMENT SIMPLE

Nous prenons ici un exemple que nous essayons de prendre le plus simple possible : Imaginons un petit gave pyrénéen coulant sur un plateau. Sur les rives, la roche est dure et permet de s'y appuyer sans glisser. Nous pourrions tailler des pierres facilement, tout est à disposition. Il faudra franchir les 8 mètres qui séparent chacune des rives, en empilant 10 pierres de même section. Nous nous doutons qu'il faudra faire une forme courbe, puisqu'à plat elles ne résisteront pas. Mais quelle forme devrions-nous donner ? Nous en essayons 5, dans l'ordre de leur apparition historique : un arc simple (style roman), l'ogive (période gothique), l'anse de panier (style Tudor), l'hyperbole (trouvée chez Antoni Gaudi, et la chaînette. Nous laissons ainsi le soin au module Galápagos de déterminer les meilleurs gènes pour minimiser la flèche au centre.

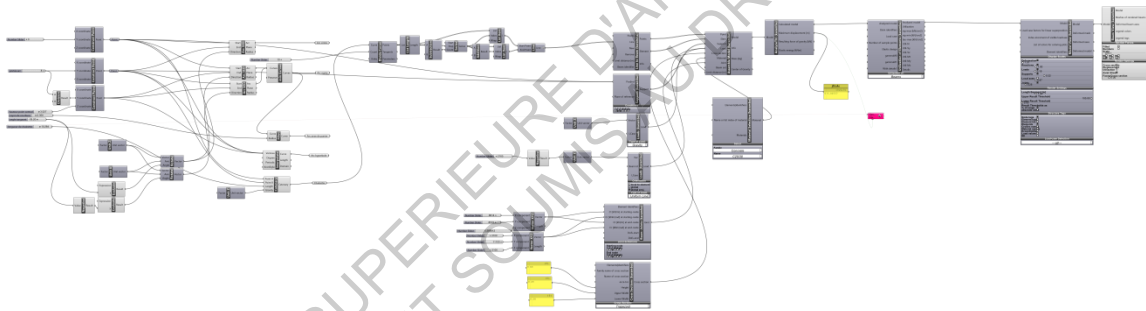


Figure 11 : Vue d'ensemble d'un algorithme Grasshopper, composé de blocs et de liens : de gauche à droite : options de génération des arcs, analyse du premier ordre par Karamba, module Galapagos, représentation des sollicitations

Ci-dessus, en figure 11, la définition Grasshopper :

- à l'extrême gauche, nous trouvons les paramètres à l'origine des différentes formes, puis
- une série de blocs permettant de dessiner les courbes. Celles-ci sont ensuite divisées en 10 portions.
- Le module d'analyse structurelle : chaque portion de courbe est à l'origine d'une pierre de section carrée de 20cm, les supports limitent fortement les déformations sans les supprimer, et nous ajoutons une force verticale correspondant au poids propre de chacune des pierres, ainsi qu'une force

uniformément répartie verticale de 2MN/m : celle-ci approxime l'effet d'une haute construction au-dessus de ce pont.

- Le bloc du module Galapagos, en rose, qui fait varier les paramètres pour créer des individus. La flèche est donnée par le module d'analyse structurelle, et Galapagos tente de la diminuer au maximum.

Voici les résultats¹ pour chaque courbe :

- Chaînette : 2.55 cm
1 gène : longueur de la chaînette
- Hyperbole : 1.53 cm
1 gène : hauteur du point central
- Arc anse de panier : 1.52 cm
3 gènes : angle de la tangente avec la verticale aux extrémités, hauteur du point de fin de cercle, rayon de courbure de congé.
- Arc en ogive : 2.23 cm
2 gènes : hauteur du point central, rayon de courbure
- Arc simple : 3.90 cm
1 gène : angle de la tangente avec la verticale aux extrémités

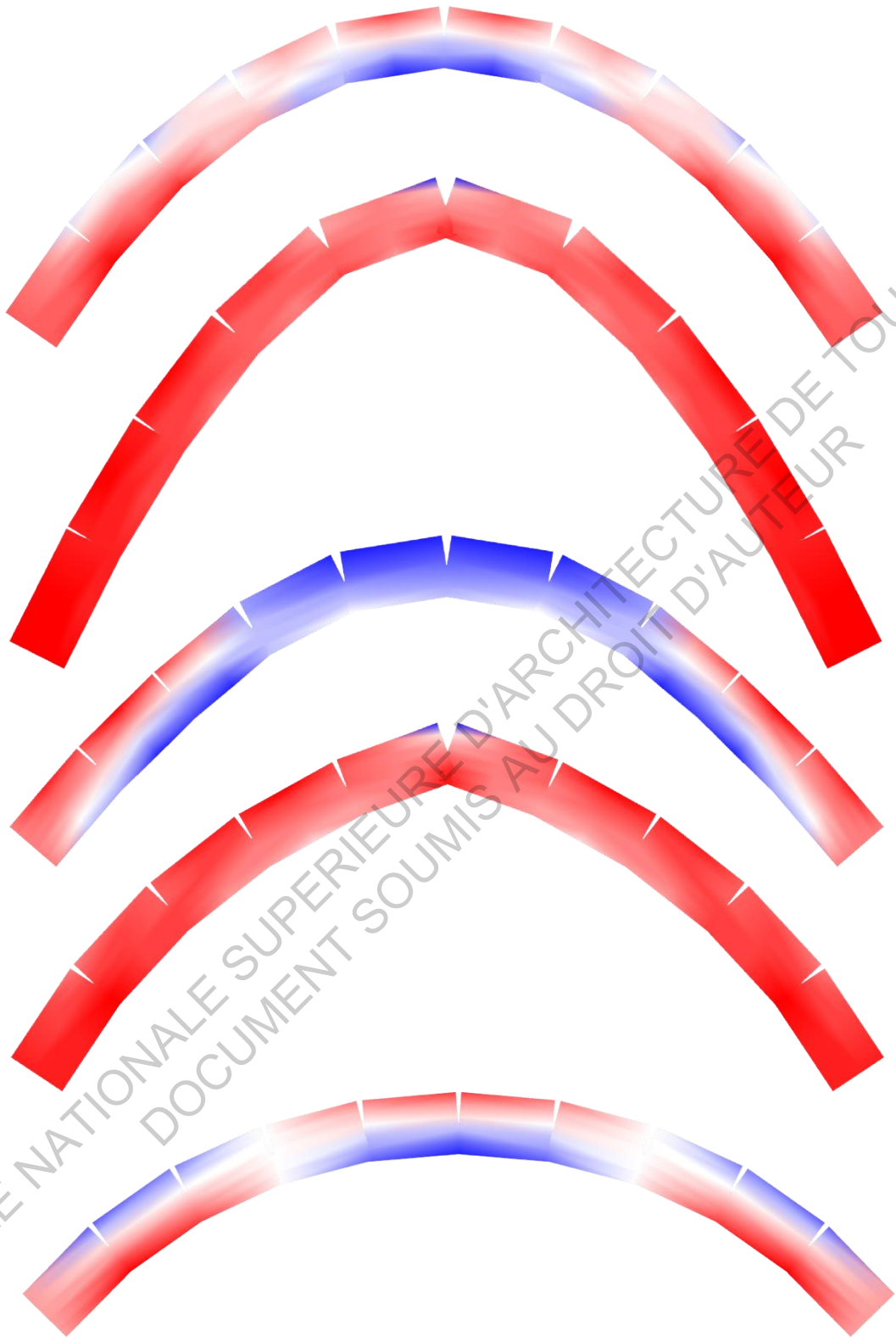
On observe ainsi que conformément aux attendus, les arcs simples et en ogive ne sont pas les solutions les plus optimales. La forme d'anse de panier a elle fortement dérivé de ce qu'on trouve dans l'architecture anglaise puisque les rayons de courbure au niveau des appuis sont particulièrement grands (de l'ordre de plusieurs mètres, contre habituellement quelques dizaines de centimètres), ils s'approchent ainsi fortement de ce qu'on obtient avec une hyperbole.

La forme en chaînette est-elle un peu décevante avec sa flèche relativement élevée : en effet, il est fréquent de concevoir des structures comme des *gridshells* en laissant tomber une forme, comme un drap tenu à chaque angle par une personne. Cette forme est ensuite retournée : la théorie voulant qu'une forme générée par une force opposée à la gravité résiste particulièrement bien à une vraie pesanteur.

On observe ainsi sur la figure 12 les structures déformées et leurs sollicitations.

¹ Hypothèses de simplification : les pierres sont encastrées entre-elles à leurs extrémités. Leur résistance est équivalente à celle d'un béton C25/30 non-armé. Elles ont une résistance en traction. Les mouvements aux appuis sont nuls en Z et fortement limités en X et en Y, les rotations sur appuis sont limitées également.

ECOLE NATIONALE SUPERIEURE D'ARCHITECTURE DE TOULOUSE
DOCUMENT SOUMIS AU DROIT D'AUTEUR



II. 2. PONT EN COQUE A DOUBLE COURBURE

Afin d'expliciter la notion de *fitness landscape*, nous allons prendre un exemple résolu par une méthode analytique. Nous définirons 2 gènes, qui permettront avec la déformation au centre d'obtenir 2 variables et un résultat, pouvant être représentés dans un diagramme tridimensionnel.

Il faut donc générer suffisamment de surfaces différentes pour avoir un diagramme relativement lisse. Nous commencerons donc par 16 valeurs possibles pour chacun des gènes. Ainsi, nous obtiendrons 256 individus. Si nous représentions cela dans une image, nous aurions alors 16 pixels dans chaque dimension, et une couleur permettant d'estimer la flèche. Mais bien que plus réaliste, cela n'est pas très visuel. Nous représenterons donc les choses dans un espace tridimensionnel, en ajoutant malgré tout la couleur, puisque ce que vous lisez actuellement est soit imprimé soit un PDF, et ne permet donc pas de bien appréhender les profondeurs.

Nous créons donc un pont, considéré comme une coque par le logiciel : il est d'une épaisseur mince mais non négligeable. Il a donc un poids propre, et doit supporter une charge surfacique. Nous cherchons ici à définir la forme idéale de cette coque.

Une fois qu'on a fixé la largeur du pont à 4m et sa portée à 24m¹, nous pouvons paramétrer deux gènes : il s'agira des altitudes

- Du point au milieu de la courbe longitudinale passant par le centre
- Des points au milieu des courbes longitudinales passant par le coté

En effectuant un loft sur ces courbes (de telle sorte que la courbe qui couperait le pont en son milieu soit une hyperbole) Ainsi, nous obtenons en vue du dessus un rectangle de 4 × 20 m, et en 3D une surface à double courbure ressemblant à une selle de cheval.

¹ Nous nous rapprochons ainsi de l'exemple d'Alberto Pugnale, Tomas Mendez Echenagucia et Mario sassone dans « Shell structures for architecture », au chapitre 18

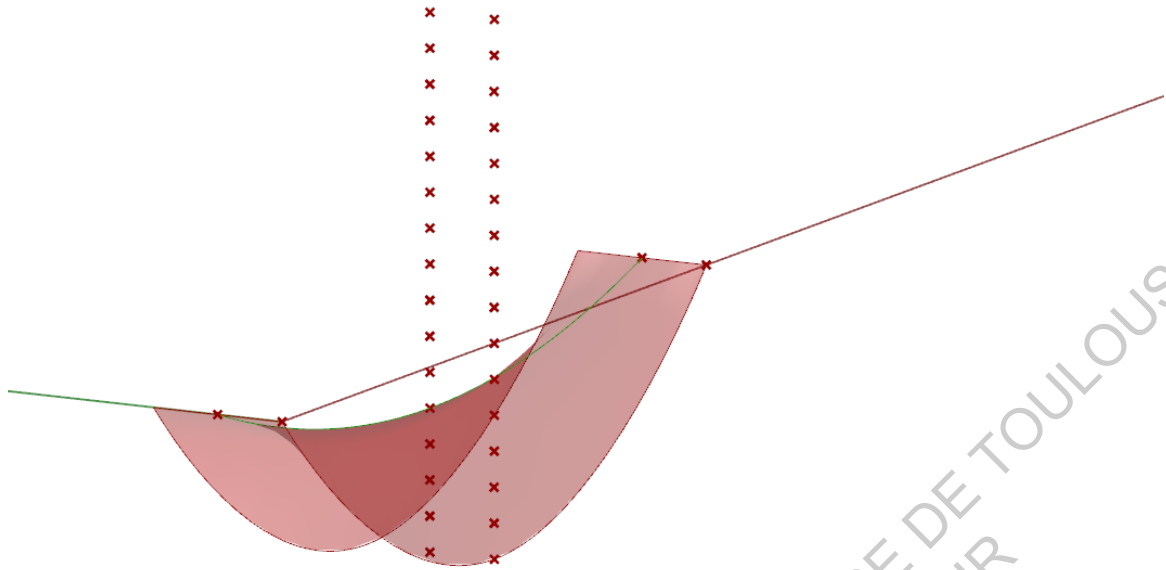


Figure 13 : Altitudes possibles et surface résultant d'un exemple

L'intérêt de cet exemple va être de savoir quelle serait la forme idéale pour minimiser la flèche au centre. Si nous admettons 16 altitudes possibles pour chacun des gènes, cela représente 256 solutions possibles. Ce nombre est suffisamment grand pour nécessiter une gestion rigoureuse des listes des gènes.

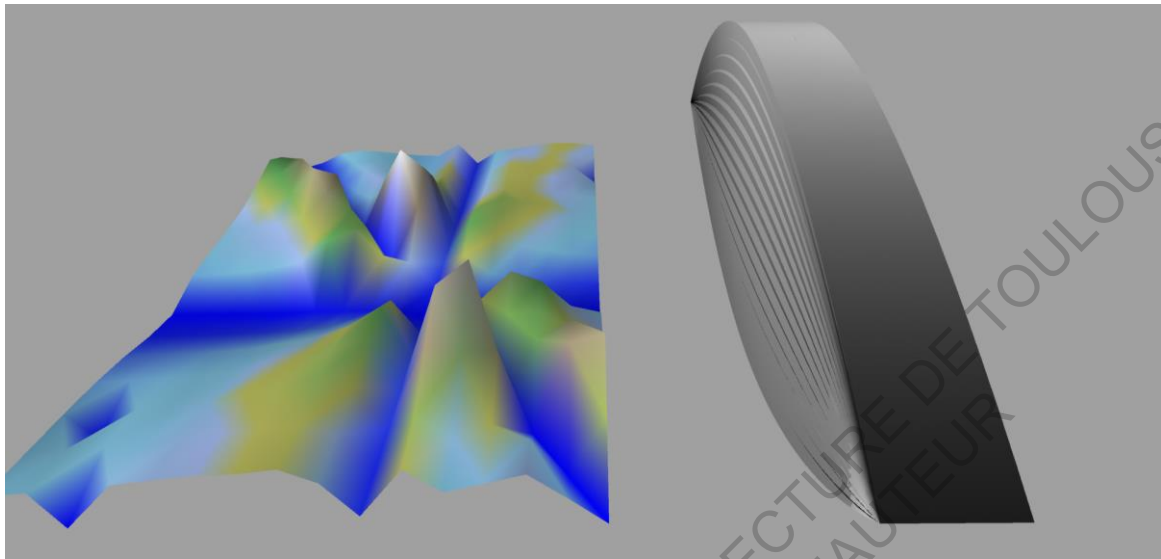
Après avoir généré les courbes génératrices pour qu'elles soient hyperboliques, nous effectuons le loft, qui nous donne la surface que nous voyions sur la figure. Celle-ci est transformée en maillage, qui peut être interprété par notre plug-in de modélisation statique après une dernière étape de transformation en coque (*shell*)

Nous définissons les appuis aux extrémités, et appliquons une charge surfacique au maillage afin d'en approximer le poids.

Elément, charge, appuis composent ainsi un système que nous analysons avec une théorie du premier ordre. Nous ne cherchons pas une grande précision, étant donné le nombre d'individus à analyser, il vaut mieux que le calcul se fasse rapidement. Sur mon ordinateur, il aura fallu 13 secondes. Le module effectuant l'analyse nous donne directement la flèche maximale. Comme nous souhaitons visualiser les solutions les moins sujettes au fléchissement¹, nous inversons les résultats scalaires pour que les

¹ En général, dès qu'il y a compression, c'est l'élançement de l'élément, donc la flèche qui devient caractéristique, puisqu'elle générerait un flambement, qui ne permettrait plus à l'ensemble de la section de travailler en compression, répartissant donc les charges inégalement.

flèches ahurissantes donnent un nombre nul, alors que les flèches très faibles donneront un nombre élevé



Cela permet de créer un *fitness landscape*, c'est-à-dire un diagramme d'aptitude qui nous permet de représenter la flèche en z de chacun des individus selon ses gènes. Il permet de représenter visuellement l'ensemble des monts et vallées des résultats possibles des individus. Ainsi, puisque le diagramme est généré à partir d'un grand nombre d'individus, il constitue lui-même l'ensemble des possibilités génomiques possibles pour un individu.

Cela me permet d'explicitier le problème des minimas et maximas locaux en conception générative : L'objectif est d'obtenir un individu qui arrivera à monter le plus haut possible sur un mont, ce qui signifiera que sa flèche la plus faible possible. Mais s'il y a trop de montagnes, nous obtenons un paysage qui deviendra complexe pour l'algorithme génératif. En effet, celui-ci a pour objectif principal de créer des individus qui monteront haut. Mais si un individu sur une montagne se reproduit avec un individu sur une autre montagne, l'individu de la génération suivante risque de naître en pleine vallée et donc d'avoir une très forte flèche ! Il faut donc faire en sorte que les individus soient relativement incestueux¹, pour se reproduire majoritairement entre individus sur la même montagne.

¹ C'est-à-dire qu'ils se reproduisent avec d'autres individus partageant un génome semblable.



Figure 15 : Les sept meilleures solutions pour le pont à double courbure

Réciproquement, si nous limitons trop l'algorithme génératif pour que les individus ne se reproduisent qu'entre individus sur la même montagne, nous risquons de passer à côté d'une grande montagne qui serait entourée de profondes vallées. Ce qui est assez proche du modèle que nous observons ici : la solution la plus à gauche sur la figure 15 est très proche des pires solutions (complètement plat, ou alors avec une simple courbure), et pourtant, c'est la meilleure.

Les sept exemples de la figure 15 permettent de bien expliciter le problème des optimums locaux dont nous parlons plus tôt. Ces sept résultats, bien qu'ayant un génome très différent, ont des flèches assez semblables. Nous pourrions comparer cela à des vaches, chevaux, chèvres et moutons : ils sont très différents, mais broutent tous suffisamment bien de l'herbe pour avoir évolué et s'être reproduits jusqu'à aujourd'hui...

Il y a donc ici une intervention humaine : nous ne choisirions sûrement pas la coque uniquement pour sa flèche, mais aussi pour des raisons bien moins conceptualisables : l'esthétique, la praticité... Celles-ci pourraient donc mener le concepteur à choisir la 5^{ème} meilleure solution, qui permettrait de ne pas avoir besoin de garde-corps, et permettrait le passage d'un bateau sous celle-ci. Il y a donc une nouvelle fois une phase de sélection non-naturelle : celle qui a lieu pendant les générations d'individus qui fait revenir d'anciens individus, mais aussi celle qui a lieu a posteriori et qui impliquera le choix d'un individu non idéal, mais optimal.

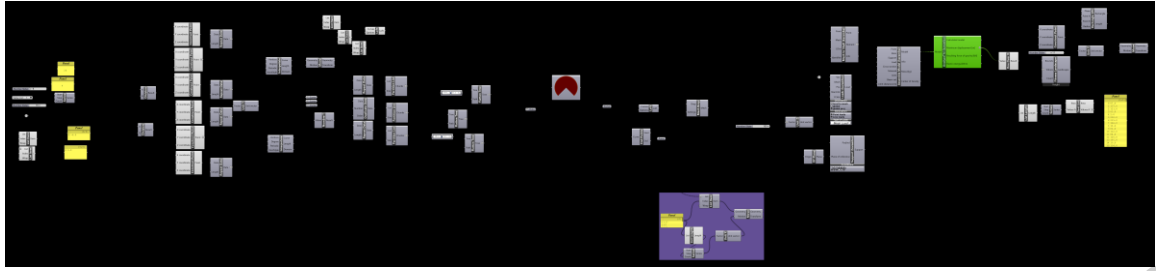


Figure 16 : Définition Grasshopper : la moitié gauche compose le loft, la moitié droite l'analyse.

Sur la figure 16, nous retrouvons la définition Grasshopper qui permet de générer les formes. Elle prend une forme semblable à celle des arcs. Il s'agit d'une succession de blocs, de gauche à droite, comme un sens de lecture. Mais cette axe horizontal correspond ainsi à une sorte de dimension temporelle, plus nous avançons dans le temps, plus le projet évolue.

On commence donc, à gauche, par définir les différentes valeurs possibles pour les deux paramètres définissant les hauteurs à mi portée au centre, et sur le côté. Nous créons 16 valeurs pour chaque paramètre, et donc créons une sorte de liste-tableau à 16 valeurs en horizontal, 16 valeurs en vertical, donc 256 variantes possibles.

Ces paramètres définiront donc 256 courbes différentes, puis 256 surfaces issues du loft de ces courbes.

Après avoir enfin analysé structurellement ces 256 surfaces, transformées en coques, nous les représentons dans le *fitness landscape* (figure 14).

II. 3. PROCESSUS GENERATIFS

II.3.1. Optimisation topologique,

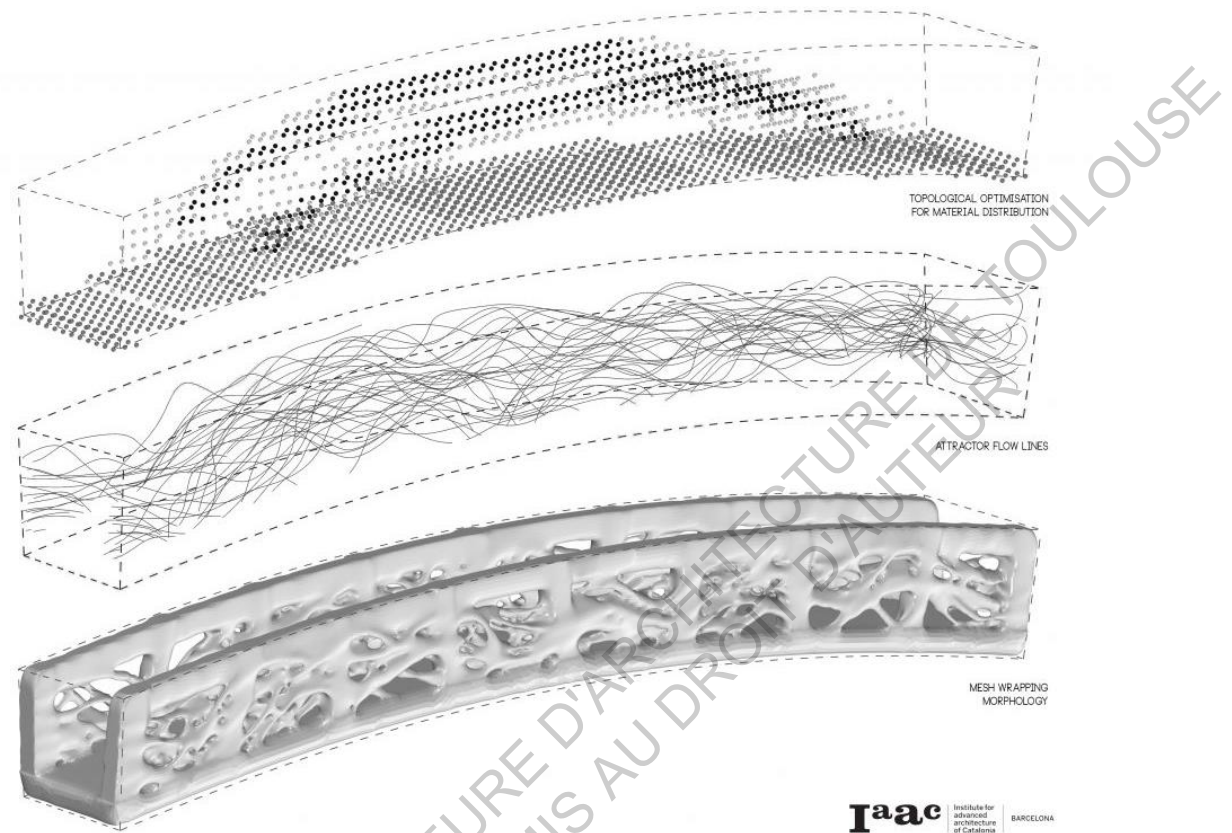


Figure 17 : Pont imprimé en 3D par l'IAAC en décembre 2016 : diagramme du processus de développement

Prenons l'exemple du pont imprimé en 3D par l'Institut pour l'Architecture Avancée de Catalogne (IAAC) en décembre 2016. Nous pouvons supposer leurs étapes de conception :

1. Génération d'une forme extérieure « contour », peuplée de points
2. Optimisation topologique (il s'agit d'une forme de conception évolutionniste) pour déterminer les points nécessitant le plus de matière
3. Dessin aléatoire de courbe à destination structurelle
4. Division du pont en 8 parties pour rendre la fabrication plus aisée, création d'un sol épais, ainsi que contours de garde-corps.
5. Association d'une épaisseur à chaque ligne selon la proximité avec les points d'attraction topologique, créant ainsi de nombreux *mesh*
6. Union et relaxation du *mesh* pour lisser les contours.

Ces contours évoquent ainsi les dessins trabéculaires de nos os.

Ceux-ci ont une conception assez semblable : chaque courbe correspond à un trabécule. Les trabécules, dont le nom vient du latin *trabes*, c'est-à-dire poutre, sont des filaments osseux qui, lorsqu'ils se lient composent la partie spongieuse des os. Les os sont en effet composés d'une sorte de coque externe, assez épaisse, appelée os compact, et au niveau des articulations (épiphyse) d'une partie spongieuse qui renferme la moelle rouge. Chez les oiseaux, le corps principal de l'os est également rempli par cette structure.

On peut la comparer à l'âme de nos poutrelles métalliques en I (IPE, IPN, HEA, etc.) : elle est accolée à une bordure pleine et épaisse, la semelle – ou la partie extérieure de l'os pour l'oiseau – qui est sollicitée en compression ou traction simple. Cette âme permet de maintenir écartées les semelles, et donc de maintenir un grand moment quadratique (inertie). Il faut donc que cette partie soit indéformable. Cependant, elle reste malgré tout assez peu sollicitée structurellement, comparativement aux semelles, c'est pourquoi il faut qu'elle soit la plus légère possible.

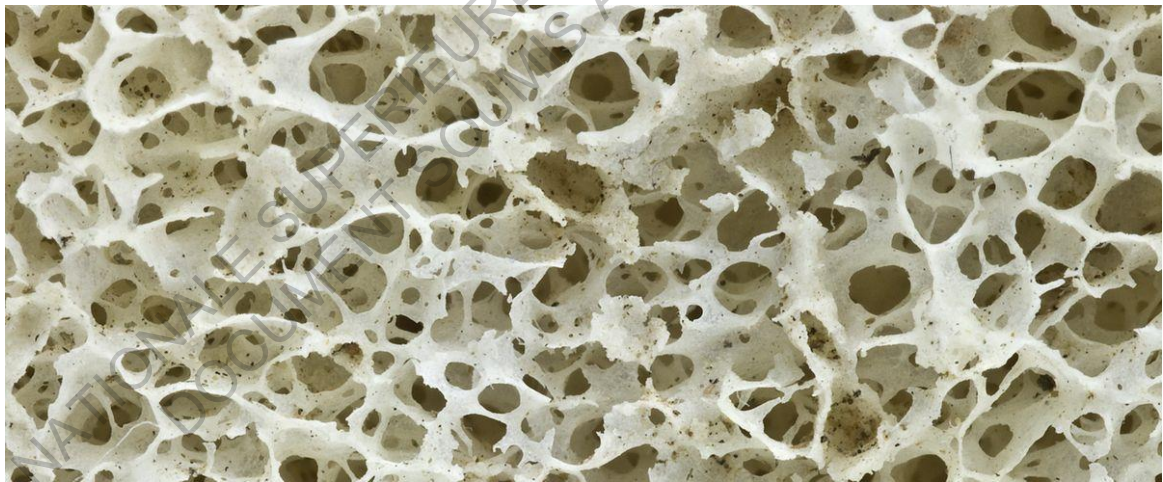


Figure 18 : Coupe macroscopique d'un os de hanche humain, Patrick Siemer

L'outil principalement utilisé est ici l'optimisation topologique : il s'agit de définir une sorte de forme « contour », soutenue en certains endroits définis par des appuis, et à laquelle nous appliquons des forces : poussée sur un garde-corps, charge d'exploitation, et plus rarement dilatation thermique. Le logiciel va alors remplir cette forme par des cubes, appelés voxels. Ceux-ci peuvent être comparés à des pixels, simulés dans une troisième dimension. Ils adoptent une fonction assez semblable aux intégrales,

notamment utilisés en mécanique des milieux continus pour les simulations structurelles¹, nous utilisons des volumes élémentaires. Ceux-ci permettent d'évaluer d'un point de vue macroscopique² le comportement de la matière.

En « voxelisant » le *dessin*, nous obtenons donc une approximation du volume en de nombreux petits cubes liés entre eux. Certains reçoivent des forces, certains sont considérés liés à des appuis, et d'autres les transmettent simplement. La simulation physique permet ainsi de connaître les sollicitations en chaque point du pont, et dessine en quelque sorte des lignes de force sur la forme. Nous effectuons ainsi une première itération avec une résolution grossière (par exemple des cubes de 50 centimètres de côté dans le cadre de notre pont), puis nous augmentons la résolution en divisant chaque voxels en n^3 voxels. Donc très fréquemment en 8 (2^3) ou 27 (3^3) autres voxels.

On obtient donc une sorte d'image tridimensionnelle pixélisée qui représente les sollicitations de chaque point de la matière (par un gradient de couleur).

En créant une sorte de sphère autour de chaque centre de voxel, dont le diamètre est proportionnel à la sollicitation du voxel, et en liant ces sphères entre-elles en lissant les contours, nous obtenons une forme optimale.

Dans le cadre du projet de l'IAAC, l'optimisation topologique sert à faire varier les sections des courbes qui traversent chacun des voxels, ils n'ont donc pas utilisé uniquement l'optimisation topologique.

¹ Par exemple, calculer l'aire sous une courbe se fait par une intégrale : Il s'agit de la somme des rectangles que l'on peut « caser » sous la courbe : plus ces rectangles sont étroits, plus nous nous approcherons de la réalité.

² C'est-à-dire qu'on ne prend pas en compte les déformations à l'échelle moléculaire, de telle-sorte qu'on puisse considérer la matière dans ce volume élémentaire comme homogène et de préférence isotrope.

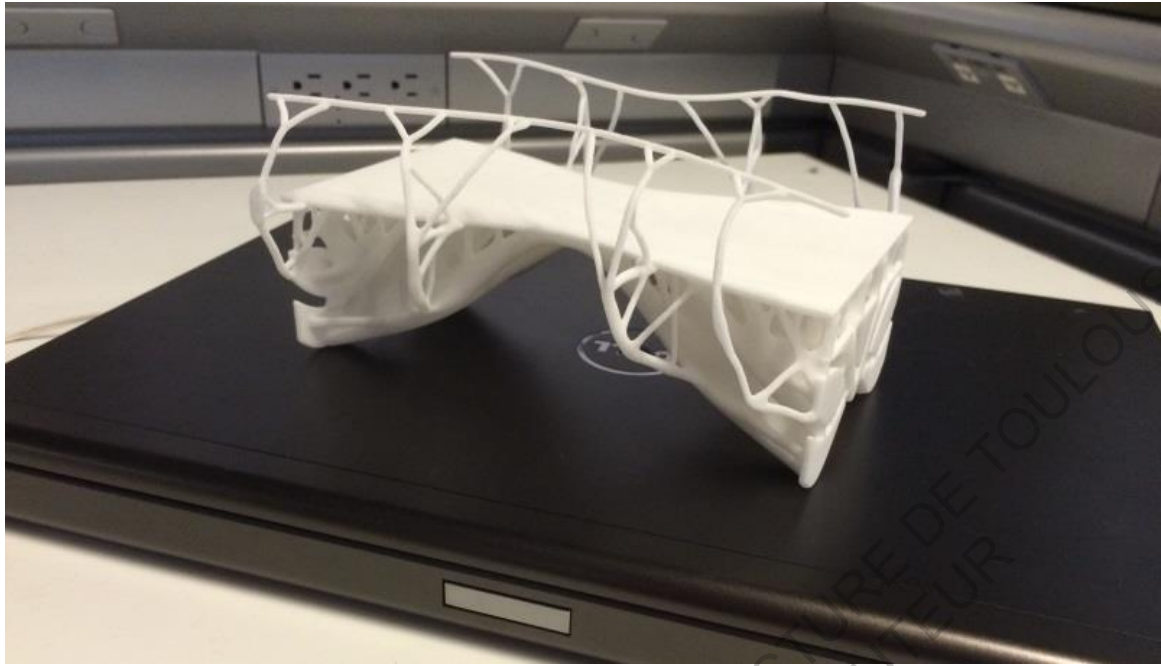


Figure 19 : Maquette d'un pont créé par optimisation topologique. Projet Autodesk Dreamcatcher

Le projet de recherche *Dreamcatcher* d'Autodesk a pour objectif l'exploration de ces méthodes de conception générative. Avant de s'allier à Joris Laarman pour le pont MX3D bridge qui devrait être imprimé en 3D, en acier inoxydable 316L par des robots à Amsterdam dans le courant de l'année, les chercheurs produisaient par eux-mêmes certains *dessins*. La figure 19 représente la maquette (imprimée en 3D) d'un pont généré uniquement par optimisation topologique. Pour la générer, les chercheurs appliquent une forme uniforme sur le dessus, créant un tablier, et appliquent des forces de poussée sur des lignes qu'on appellerait garde-corps. A chaque extrémité du pont, nous trouverons des supports. Ici, les chercheurs n'ont utilisé que l'optimisation topologique, nous pouvons donc remarquer l'aspect biomorphique que prend le pont généré. Cette forme semblable à ce qu'on trouve dans la nature nous permet de remarquer que si elle produit ce genre de formes, c'est qu'elles sont optimales !

11.3.2. Recherche de forme: La conception d'une paroi de partition pour Airbus A320



Figure 20 : Paroi de partition imprimée en 3D (Airbus / Autodesk Dreamcatcher)

A nouveau, nous retrouvons un exemple issu du projet *Dreamcatcher* d'Autodesk, cette fois en partenariat avec Airbus. Il s'agissait de concevoir une cloison de séparation pour les avions Airbus A320. Ici, un solveur génératif a imité la croissance d'organismes monocellulaires (myxomycètes) pour *dessiner* la structure interne de cette cloison. Une fois *dessinée*, celle-ci est analysée avec la simulation d'une force de 9G. La méthode générative est ensuite la même : le solveur fait varier les paramètres à l'origine de la croissance des myxomycètes, crée des individus, et des générations. L'objet final ainsi obtenu, imprimé en 3D a permis de réduire le poids de chacun des cloisons de séparation de 45%, tout en étant 8% plus résistante que celles utilisées habituellement.

II.3.3. Optimisation des sections

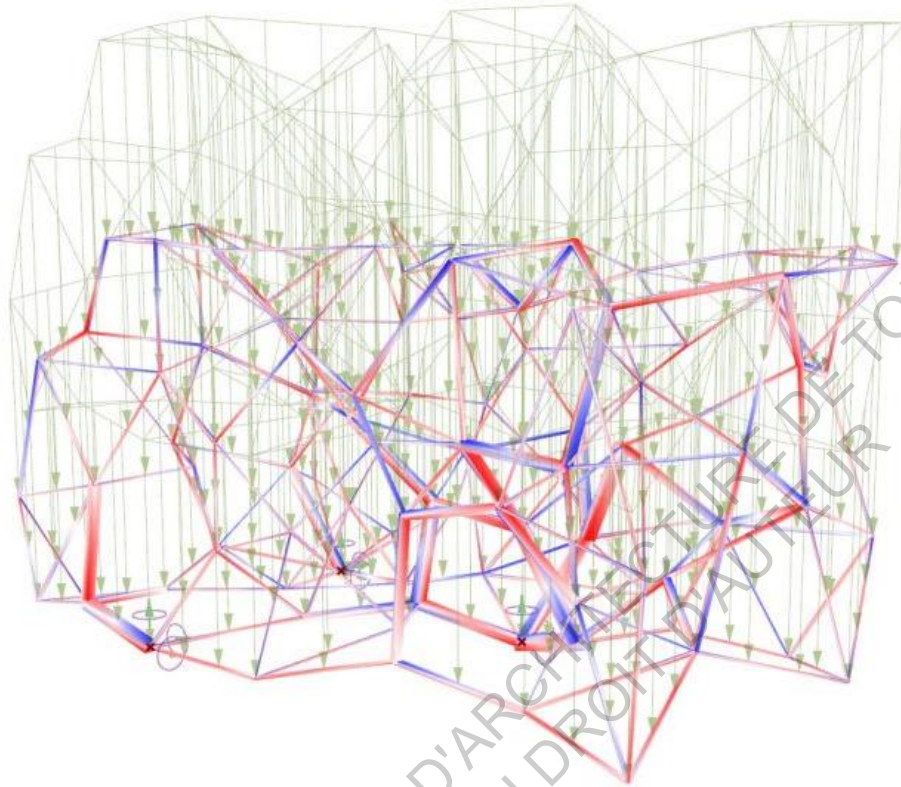


Figure 21 : Exemple d'application du Cross section optimizer de Karamba, Clemens Preisinger, Jungwo Park

Il s'agit ici d'un cas un peu plus standard, au processus plus itératif que génératif : en effet, le solveur ne crée pas de nombreux individus, mais seulement un par génération, qu'il aura obtenu par une méthode analytique. A chaque génération, les sections (par exemple rectangulaire, en I, en H, en tube) de chaque poutre/bielle sont modifiées d'abord en fonction des sollicitations et optionnellement des déformations, en respect de l'Eurocode 3 (structures en acier). Ainsi, après un nombre de génération généralement assez restreint (autour de 5 pour des modèles simples, jusqu'à 20 dans des cas très complexes), nous obtiendrons un modèle bien optimisé.

II. 4. CREATION D'UN ALGORITHME GENERATIF ET FORMALISATION

Nous allons désormais nous inspirer des méthodes que nous avons découvertes et appliquées plus tôt afin de créer une passerelle piétonne. Nous allons donc créer un pont de 24 mètres de portée, avec un peu plus de quatre mètres de largeur utile.

Mais il faut pour cela commencer par s'interroger sur ce qu'est fondamentalement une passerelle. Nous cherchons ici à « mettre à plat », « théoriser » la passerelle, afin de pouvoir la concevoir. Mais nous souhaitons en même temps élargir au maximum cette définition, afin de ne pas tomber dans l'imitation de ce qui existe déjà. La conception générative est un modèle nouveau, nous escomptons donc des résultats nouveaux et inattendus. Pas de simples arches pleines, de formes trop simples, ou de formes trop biomorphiques.

Nous n'utiliserons donc pas d'optimisation topologique. En effet, le concepteur n'aurait qu'une faible place dans ce procédé: il ne serait qu'un transcritteur qui convertirait des données « réelles » (charges, portées, masse en pourcentage...) en données interprétables par l'ordinateur qui créerait plus ou moins automatiquement un modèle. Ce serait la solution de facilité. Nous explorerons donc d'autres méthodes.

Cette passerelle sera définie par une section au niveau des appuis, formellement relativement libre, et une section à mi-travée. Le solveur aura alors pour rôle de créer une sorte de forme/contour adaptée à un matériau relativement faible.

Ce matériau sera ensuite remplacé par la suite par un autre plus résistant, qui n'aura donc pas besoin du même volume pour avoir des réactions et comportements physiques semblables. De la même façon qu'en tranchant un os on découvre une autre structure trabéculaire très aérée, nous allons transformer notre coque solide en contour imaginaire d'une structure composée de milliers d'éléments évoquant des rhizomes, des neurones...

Ainsi, nous obtiendrons au final une structure aérienne, presque creuse, apte à être appropriée par la faune et la flore (on espère que les oiseaux viendront nicher aussi sous ce pont-là !), tout en étant optimale à nos yeux humains et à nos attentes normatives.

Le processus étant quelques peu complexe, il est divisé en onze parties :

- Paramètres
- Courbes
- Surface
- Analyse (Karamba)
- Heuristique (Galapagos)
- Remplissage par sphères (Kangaroo)
- Génération de « poutres »
- Analyse (Karamba)
- Optimisation des sections (Karamba)
- Jointoiment des sections pour imiter une structure trabéculaire et augmenter la résistance aux moments fléchissant (Exoskeleton)
- Export, rendus

Paramètres

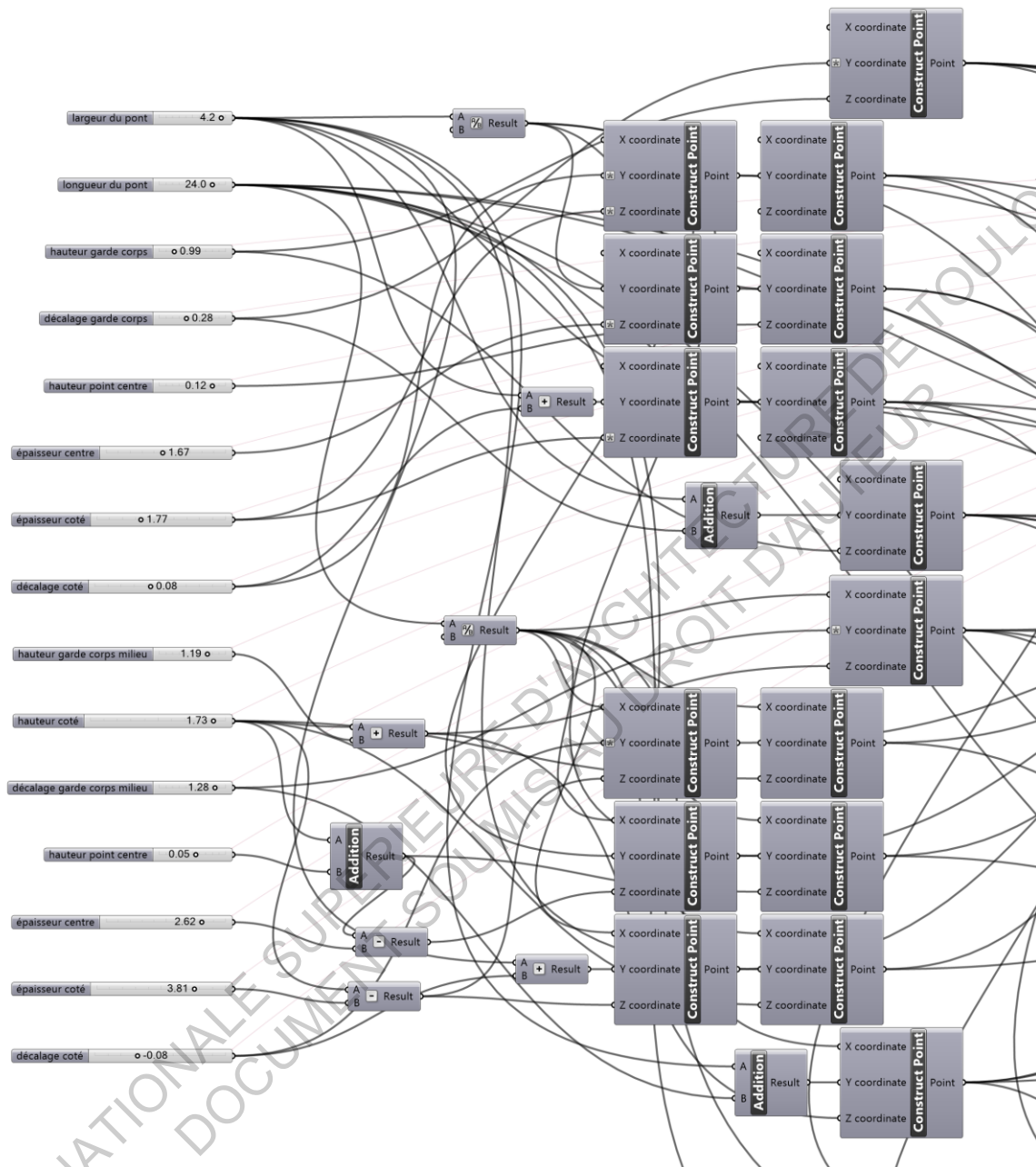


Figure 22 : Paramètres et génération des points: la moitié haute sert à dessiner la courbe au niveau des appuis, la moitié basse à mi-travée

Nous commençons par quelques paramètres non-modifiables par le logiciel :

- La largeur du pont
- La longueur du pont

Tout le reste constitue des paramètres modifiables par le solveur évolutionnaire. Ils sont divisés en deux parties, que l'on devine grâce à la symétrie suivant l'axe central de l'image de la figure 22 :

La partie haute permet de gérer la forme de la courbe générant la forme au niveau des appuis, et la partie basse la courbe générant la forme à mi-portée. Nous détaillerons ces points dans la partie suivante

On devine en arrière-plan les courbes rouges qui indiquent les *sliders* sur lesquels le solveur peut agir.

Points de contrôle des courbes aux appuis

Points touchés	Nom du paramètre	Borne basse	Borne haute	Valeur choisie
0, 4	Hauteur du garde-corps	0.90	1.30	0.99
0, 4	Décalage du garde-corps suivant l'axe y	-0.50	1.41	0.28
2	Hauteur du point au centre	-0.20	0.20	0.12
6	Épaisseur au centre	0.30	3.20	1.69
5, 7	Épaisseur sur le côté	0.10	5.00	4.90
5, 7	Décalage du point d'épaisseur de côté suivant l'axe y	-0.60	1.00	0.08

La courbe passant par les points étant relativement lisse, une borne basse de 90cm de haut du garde-corps permettait d'assurer toujours une hauteur réelle de 1m, donc le respect des normes NF P01-012 et XP P98-405.

La hauteur du point au centre a été limitée pour garder une certaine accessibilité PMR tout en permettant une plus grande liberté formelle, et de s'approcher de la

meilleure solution que l'algorithme génératif avait obtenu dans la partie pont en coque à double courbure.

Il eût été possible, étant donné les valeurs des bornes pour les décalages, d'obtenir une courbe qui s'intersecterait elle-même. Cependant, celle-ci donnerait une géométrie certainement inapte, et le solveur l'éviterait de lui-même.

On remarque également qu'aucune des valeurs choisie par l'algorithme génératif n'est un extrême, ce qui signifie que la forme finale obtenue constitue un réel optima, c'est-à-dire que l'individu se situe en haut d'une montagne, et non pas aussi haut qu'il le pouvait sur une montagne tronquée.

Points de contrôle de la courbe à mi-travée

Points touchés	Nom du paramètre	Borne basse	Borne haute	Valeur choisie
8, 12	Hauteur du garde-corps	0.90	1.30	1.19
8, 9, 10, 11, 12	Hauteur du coté	1.00	2.00	1.73
8, 12	Décalage du garde-corps suivant l'axe y	-0.40	1.60	1.28
14	Hauteur du point au centre	-0.50	0.50	0.05
14	Épaisseur au centre	0.30	3.20	2.62
13, 15	Épaisseur sur le coté	0.10	5.00	3.81
13, 15	Décalage du point d'épaisseur de côté suivant l'axe y	-0.60	1.00	0.08

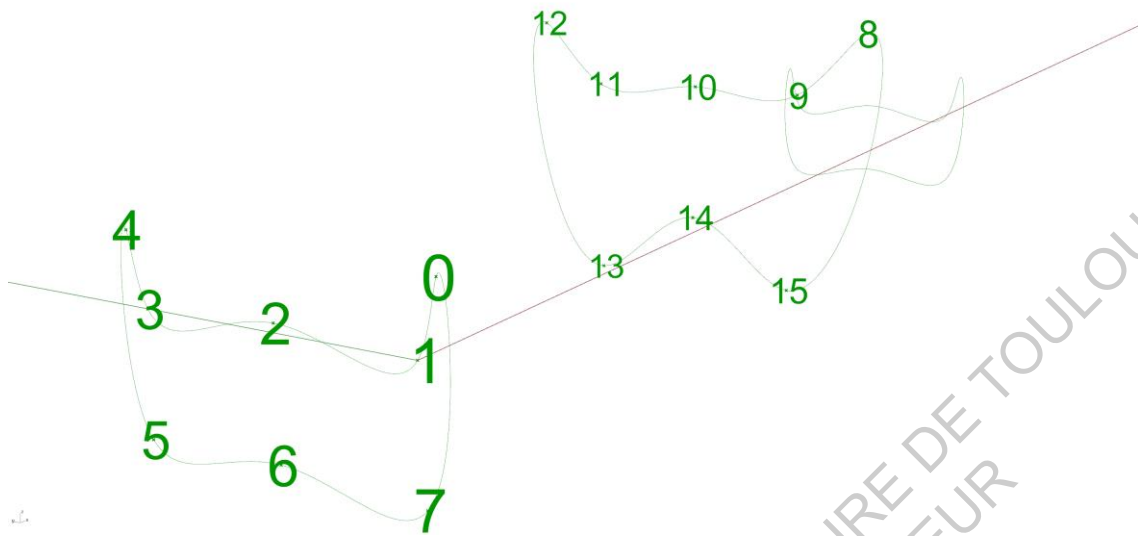


Figure 23 : Numérotation des points et courbes génératrices

Courbes et surface

Comme nous avons observé dans l'application à l'élément simple que la forme hyperbolique était optimale dans le cas des arches, nous l'utilisons ici pour générer toutes les courbes. Celles-ci sont ensuite « loftées » : c'est-à-dire qu'on peut les comparer aux membrures des bateaux : elles constituent la structure transversale. Le loft permettant de créer la surface qu'on appellerait coque. Le fonctionnement de l'outil loft et de l'outil *interpolate* est analogue : d'une dimension, on passe à deux avec l'*interpolate*, puis de deux à trois avec le loft, avec la même méthode de liaison des points par hyperboles

Surface

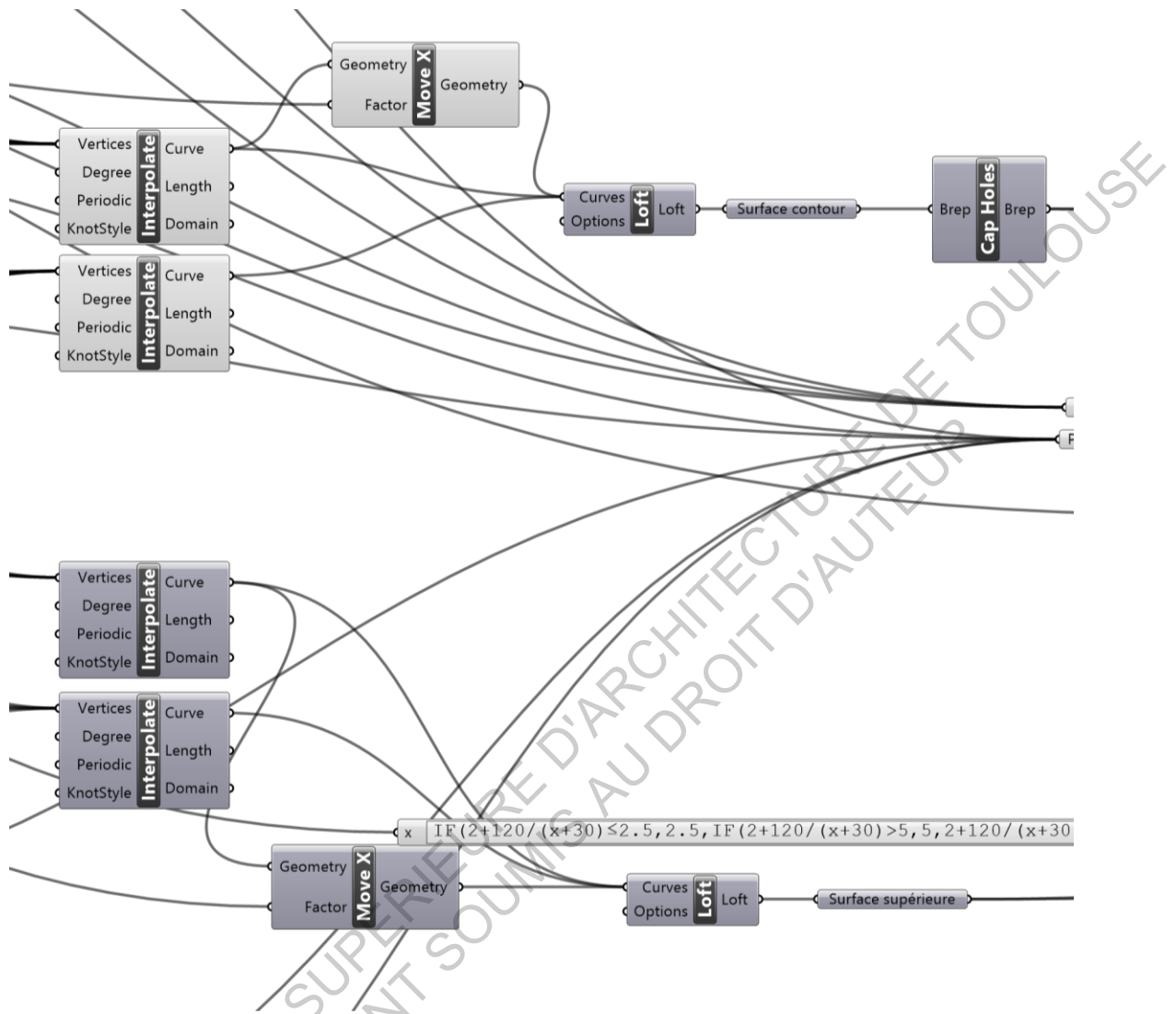


Figure 24 : Blocs permettant la création des courbes, puis des surfaces

Nous observons dans cette partie les blocs permettant les générations des courbes, lorsqu'ils sont groupés par deux, celui du dessus permet de générer la courbe par un appui, courbe ensuite décalée de la portée du pont suivant l'axe x. Celui du bas est utilisé pour la courbe à mi-portée. Afin de transformé le loft en un volume, on utilise l'outil *cap holes* pour le refermer, on obtient alors un *brep*.¹

¹ De l'anglais *boundary representation*, c'est-à-dire représentation géométrique par les contours

Analyse (Karamba)

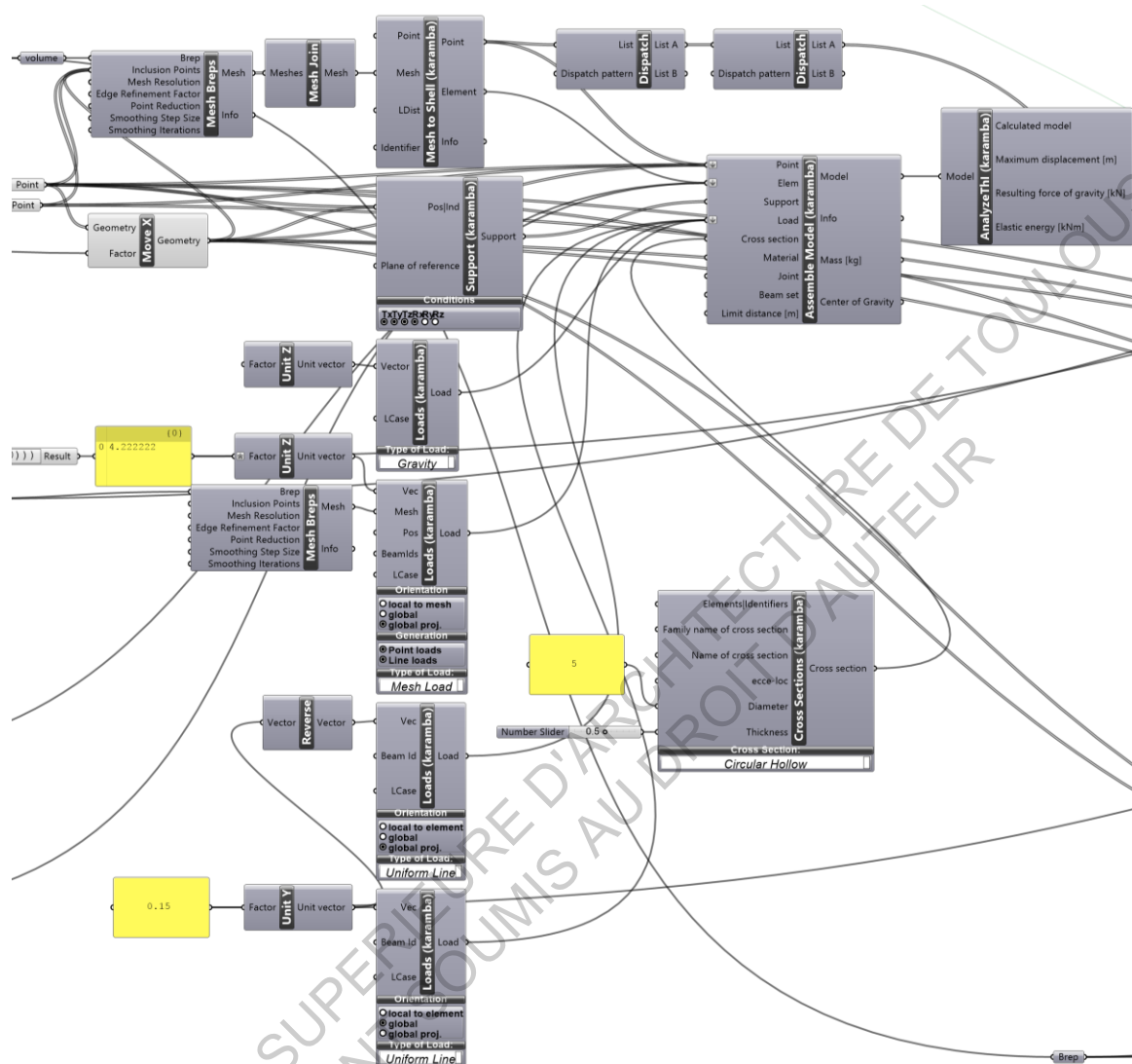


Figure 25 : Première phase d'analyse structurelle

Sur la figure 24, nous pouvons observer une longue formule de calcul, celle-ci est en effet issue d'une norme qui indique que plus la passerelle est grande, plus la charge d'exploitation doit être considérée grande, avec une formule de calcul la bornant entre 2.5 et 5 kN/m². Pour le reste, l'algorithme est assez semblable à ce que nous avons vu précédemment : nous définissons des charges (permanentes (*Gravity*) et d'exploitation (*Mesh Load*)), dont des charges de poussées sur les garde-corps (*Uniform Line*), des supports (semi-encastrement : seuls les rotations autour des axes Y et Z sont autorisés, c'est-à-dire qu'on interdit au pont le roulis.).

Après s'être assuré que les points de support faisaient bien parti du *mesh* issue du volume (et en forçant cette appartenance), puis de la coque, nous ajoutons ces éléments au solveur :

- L'élément, c'est-à-dire la coque
- Les 8 supports (points 1, 3, 5, 7 et 9, 11, 13, 15)
- Un poids propre : gravité
- Une charge d'exploitation dont nous parlons plus tôt
- Deux charges de poussées de 0.15kN/ml sur le haut des garde-corps.

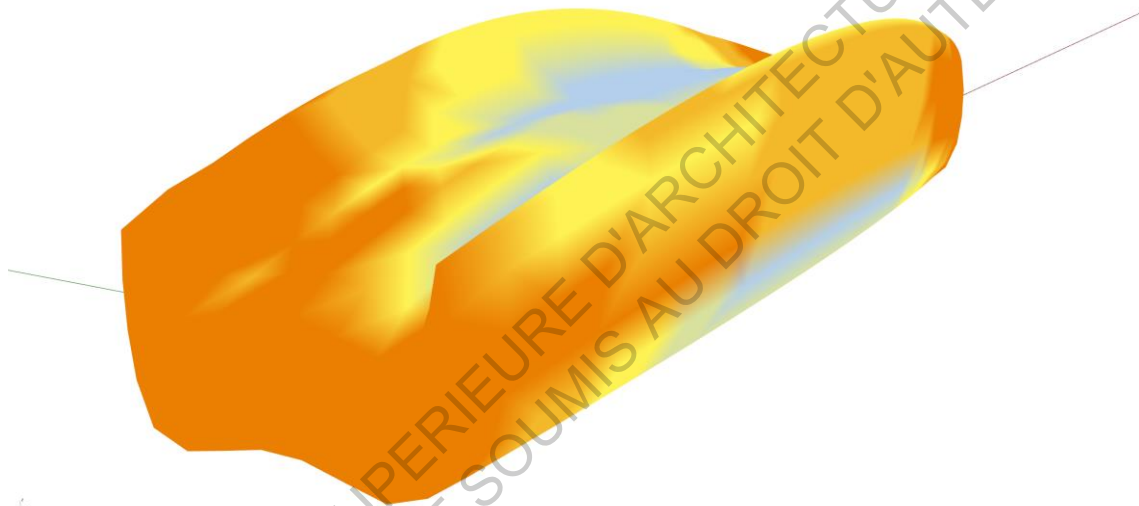


Figure 26 : Mesh de la passerelle en phase 1, représentation des déformations

Après avoir effectué une analyse du modèle avec Karamba, on obtient toutes les réactions d'appuis, ainsi que les contraintes de Von-mises, ou comme sur la figure 26, les déformations, qui sont évidemment plus fortes à mi-travée.

Heuristique (Galapagos)

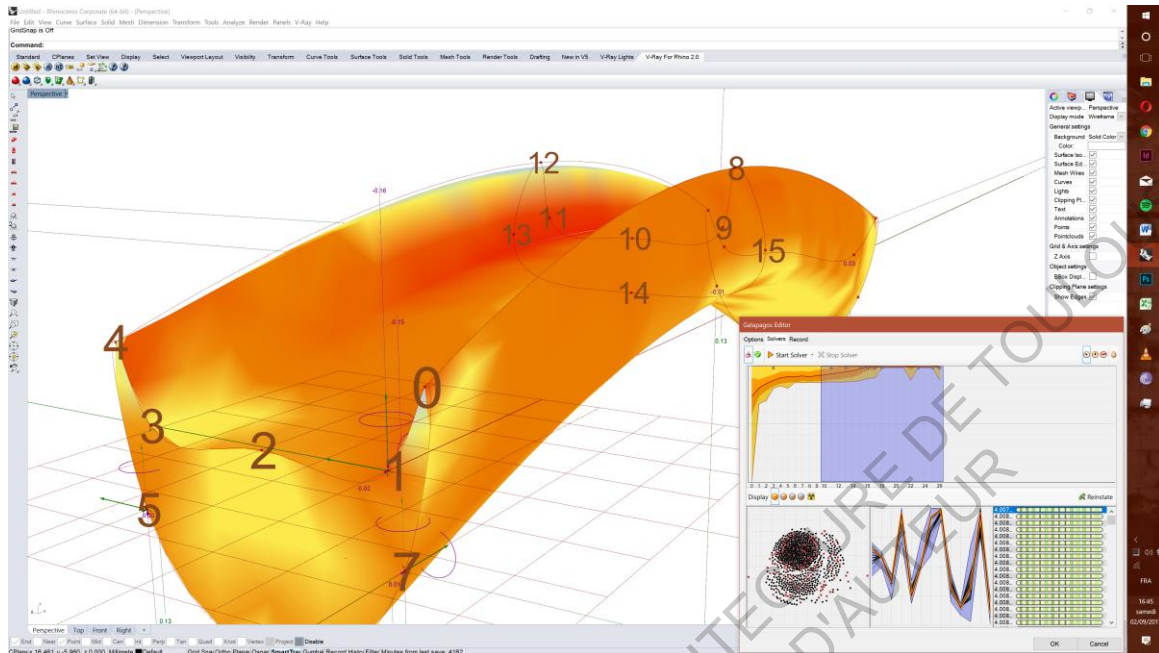


Figure 27 : Interface Galapagos pendant la résolution

Sur la figure 27, nous pouvons observer à quoi ressemble notre interface de travail avec Grasshopper et Galapagos. Le fonctionnement du solveur génératif (en bas à droite) a déjà été expliqué en détail dans la partie 1.1.6, mais nous en rappelons le fonctionnement, avec l'observation et l'analyse de ces résultats :

- Dans l'interface de Rhinocéros, nous observons l'ensemble des individus proposés par le solveur, ils ont ici été colorés en fonction des contraintes de Von Mises, qui permettent de représenter visuellement la contrainte interne à la matière en prenant en compte les sollicitations longitudinales et tangentielle.
- Dans la partie haute du solveur de Galapagos, nous retrouvons un graphique présentant l'aptitude des individus des différentes générations. Ici, la génération 26 a été atteinte après environ 300 minutes de calculs. Les graphiques ressemblent souvent à des logarithmes : à chaque génération les individus devraient être meilleurs qu'à la génération précédente, mais plus l'indice de la génération sera élevé, moins les individus auront progressé

Il faut donc définir le seuil du suffisant¹. Ici, une durée de calcul de 300 minutes semblerait presque être trop courte, on aurait pu aller jusqu'à 500 minutes pour s'assurer de la validité de cette affirmation.

- Dans la partie basse du solveur de Galapagos, on peut apprécier plus en détail le génome des individus. On peut notamment se faire une idée du fitness landscape en bas à gauche. Il y a beaucoup trop de gènes pour que la comparaison aux Pyrénées que nous avons précédemment soit très juste², mais on peut simplifier le problème en approximant que les individus semblent graviter autour d'un point. Puis, après une sorte de ravin, ils continuent à graviter autour de celui-ci, en se regroupant ensemble. Il y a ensuite quelques autres-points « renégats » flottant plus loin, que le solveur a certainement du tuer d'une génération à l'autre.

Le résultat de cette simulation est légèrement différent de la forme que nous avons plus tôt et que nous continuerons à utiliser plus tard puisque la gestion des supports était légèrement différente.

¹ Ce seuil peut-être très variable suivant la complexité et l'escarpement du *fitness landscape* et du génome. Par exemple, dans le cas de notre étude de forme pour une arche, en 5 générations un résultat tout à fait acceptable est obtenu. Au contraire, si on demande à un algorithme d'apprendre à conduire, ce nombre sera bien plus grand.

² Le réalisateur Christophe Nolan s'est attaché à représenter un espace en 4 à 5 dimensions dans le film *Interstellar*, 2014. Nous aurions ici 14 dimensions (13 gènes et une aptitude).

Remplissage par sphères (Kangaroo)

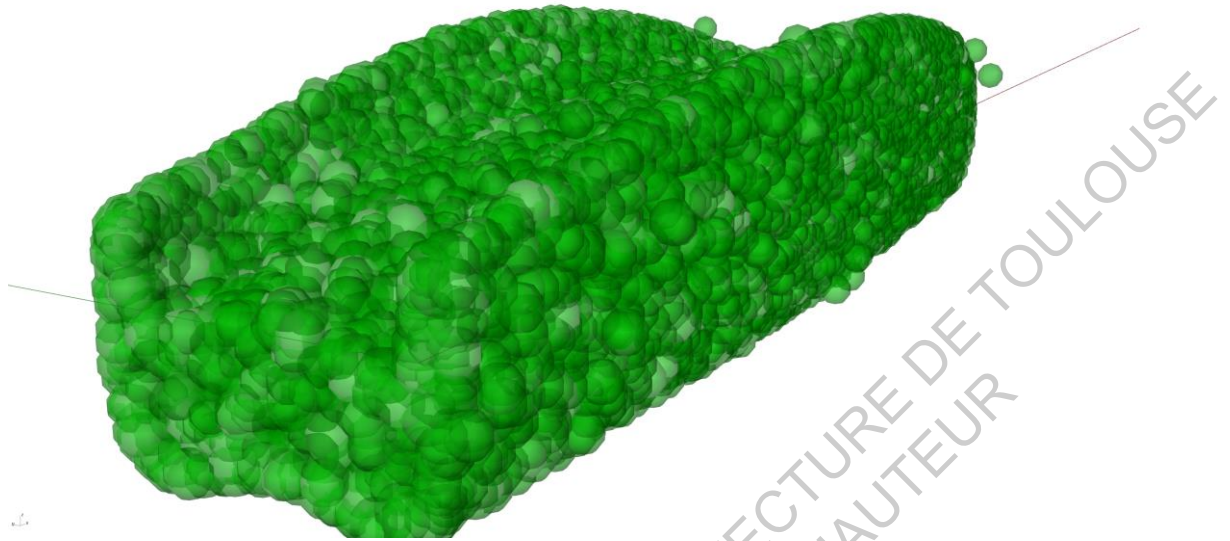


Figure 28 : Deuxième phase : remplissage par des sphères

Ici, c'est le module Kangaroo qui s'occupe de la simulation : il cherche à remplir le volume généré précédemment avec des boules dont on définit le diamètre. Il s'agit d'une simulation dynamique. Bien que celle-ci permette en théorie une simulation en temps réelle, elle se veut réaliste, mais le comportement reste approximatif. En effet, les simulations physiques sont moins rigoureuses que les simulations statiques, pour des raisons évidentes de mobilisation de logiciel. De plus, les théories à l'origine ne viennent pas des mêmes domaines : la mécanique des milieux continus pour la statique, et de la physique pour la mécanique. Ici nous cherchons surtout à remplir le volume de façon relativement uniforme. Nous aurions pu recourir à l'utilisation d'un voronoï, mais cette méthode permet plus de rigueur avec un écartement plus constant et un bon remplissage des extrémités, ou des zones étroites comme les garde-corps.

Dans l'exemple de la figure 28, le module remplit la coque avec des sphères de 40cm de diamètre. Chaque étape de simulation prend environ 4 secondes avec cette dimension. Cette simulation doit normalement s'arrêter lorsque les déplacements entre chaque phase ne dépassent pas un centimètre, c'est le seuil du suffisant de Kangaroo.

Génération de « poutres »

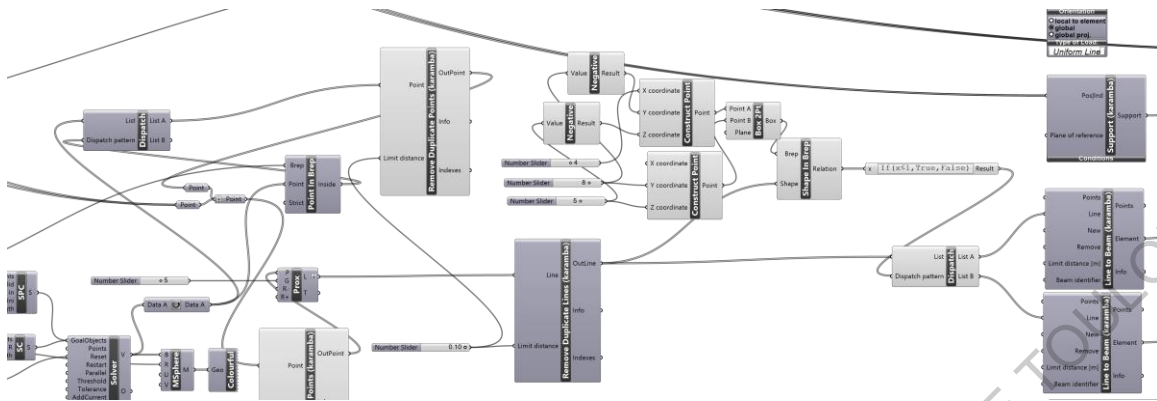


Figure 29 : Passage de centre de sphères à des lignes avec triage.

Nous utilisons d'abord un barrage de données (*data dam*) permettant d'empêcher les calculs des phases suivantes tant que la phase mécanique n'est pas terminée : celle-ci prend environ 4 secondes, mais la phase d'analyse en prend environ 20, et l'optimisation des sections près de 2 minutes, tandis que la relaxation des maillages, c'est-à-dire l'arrondissement des contours peut prendre jusqu'à 20 minutes. Ici nous nous concentrons seulement sur la simulation physique, le barrage nous permet donc de laisser l'aval à l'abri de ce flot de données et donc à la simulation physique de prendre le temps d'arriver à une solution acceptable.

Nous effectuons une extraction des points des centres des sphères, vérifions l'appartenance à la coque et excluons ceux qui sont en dehors. Nous ajoutons également les points correspondant aux appuis, afin de s'assurer que des poutres auront les appuis comme origine.

Des lignes sont ensuite tracées entre ces points, puis on vérifie qu'elles-mêmes ne sortent pas de la coque en excluant à nouveau celles qui sont en dehors. Enfin, celles-ci sont « nourries » dans le bloc « *line to beam* » afin d'être considérées/sémantisées comme des poutres.

Cette construction géométrique est très semblable à ce que nous avons précédemment à la seule différence que nous n'avons pas de *shell* (coque) mais des *beams* (poutres), pour le reste, nous avons les mêmes chargements, mais nous définissons également le matériau comme étant de l'acier S355, il s'agit d'un acier courant en construction et avec d'assez hautes résistantes.

Après avoir défini la section de chacune de ces poutres comme étant circulaire pleine de 2cm, nous précisons que la liaison entre ces poutres est un encastrement.

Nous commençons à avoir un modèle assez complexe puisque nous obtenons environ 20000 poutres (pour des sphères de 60cm) aux géométries toutes différentes.

ECOLE NATIONALE SUPERIEURE D'ARCHITECTURE DE TOULOUSE
DOCUMENT SOUMIS AU DROIT D'AUTEUR

Optimisation des sections (Karamba)

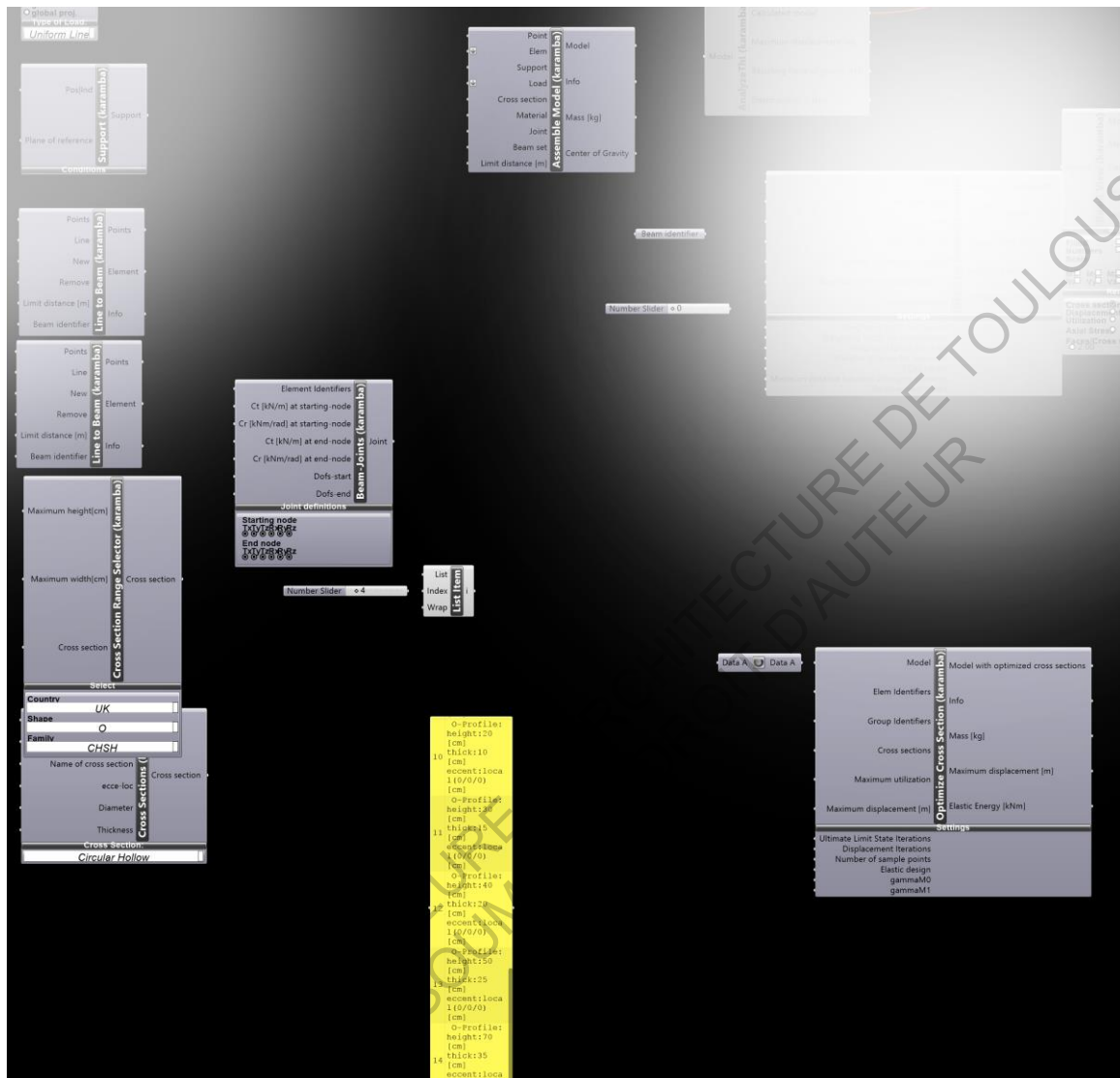


Figure 31 : Optimisation des sections

Il y a cette fois-ci un nouveau module : *Optimize Cross Section* : celui-ci évalue de manière itérative quelle serait la meilleure section pour chacune des nombreuses poutres, parmi une liste que nous donnons : celle-ci permettrait, si l'on choisit une construction par assemblage, de rationaliser en partie le modèle. Ainsi, on pourrait même faire construire la passerelle par des hommes : il y aurait des portions (les liaisons entre barres) imprimées en 3D par des méthodes plus habituelles (soit directement, soit coulées dans des moules contre-forme en plâtre/terre dont une cire perdue (forme) aurait été imprimée aisément : cette méthode-là ne requiert aucune matière spécifique. Il faudrait enfin souder ou assembler les barres et les liaisons entre elles.

Jointoiment des sections pour imiter une structure trabéculaire et augmenter la résistance aux moments fléchissant (Exoskeleton et Intralattice)

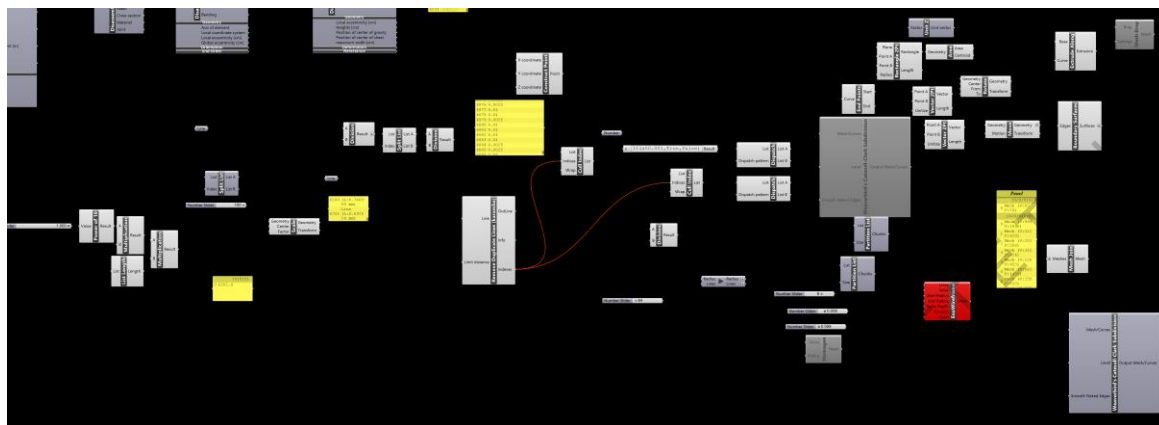


Figure 32 : Jointoiment des sections

Cette dernière partie de l'algorithme de Grasshopper est un peu plus brouillonne, puisque nous effectuons plusieurs fois la même chose. Il s'agit d'utiliser la méthode des *Marching cubes* pour arrondir les angles des maillages : celle-ci est assez risquée et admet une faible marge d'erreur aléatoire (moins de 0.5 %), cependant, comme nous avons 20000 poutres avec des sphères de 60cm, la loi de Murphy s'applique, « Tout ce qui est susceptible de mal tourner tournera nécessairement mal ».

L'algorithme commence par décomposer le modèle géométrique de Karamba pour extraire les axes de chacune des poutres et les sections associées. Ces deux éléments sont ensuite légèrement filtrés pour limiter les erreurs (dus parfois à la proximité géométrique) puis utilisées par les modules *Exoskeleton* et *Intralattice*, qui sont basées sur la même méthode.

Ces modules vont lier entre elles les poutres en augmentant la densité de matières aux angles, ce qui permettra un bon transfert des moments fléchissant d'une poutre à l'autre : cela permet au final de limiter les déformations de la passerelle

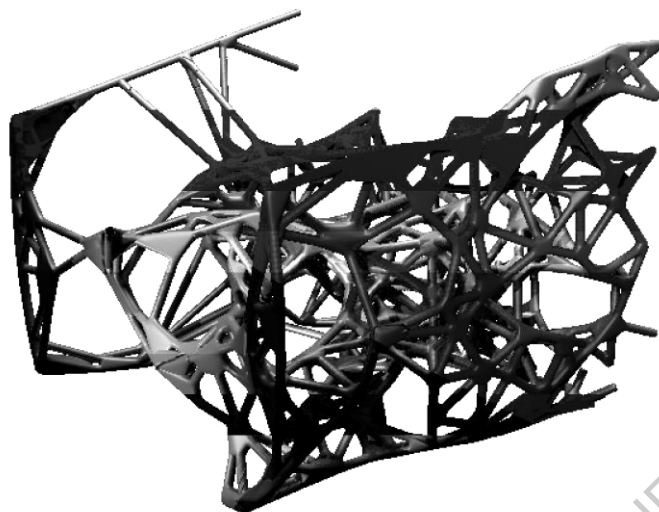


Figure 33 : Portion de passerelle

Sur la portion de passerelle de la figure 33, on peut voir l'effet de ces deux modules : les angles sont arrondis, mais les sections en parties courantes sont maintenues.

Afin d'estimer réellement le comportement de la passerelle, avant de la réaliser, il faudrait considérer le maillage obtenu comme une coque, et simuler à nouveau les déformations sur celle-ci, ainsi que les contraintes internes. Cependant, étant donné que pour seulement un quart de la portée le *mesh* comprend plusieurs millions de facettes¹ après réduction, et que l'algorithme à l'origine du jointolement n'a pas permis d'effectuer cette opérations sur la totalité des poutres

¹ Lorsqu'elles sont de type *quads* (quadrilatère déformé), environ deux fois plus si le maillage est triangulé

Export, rendus

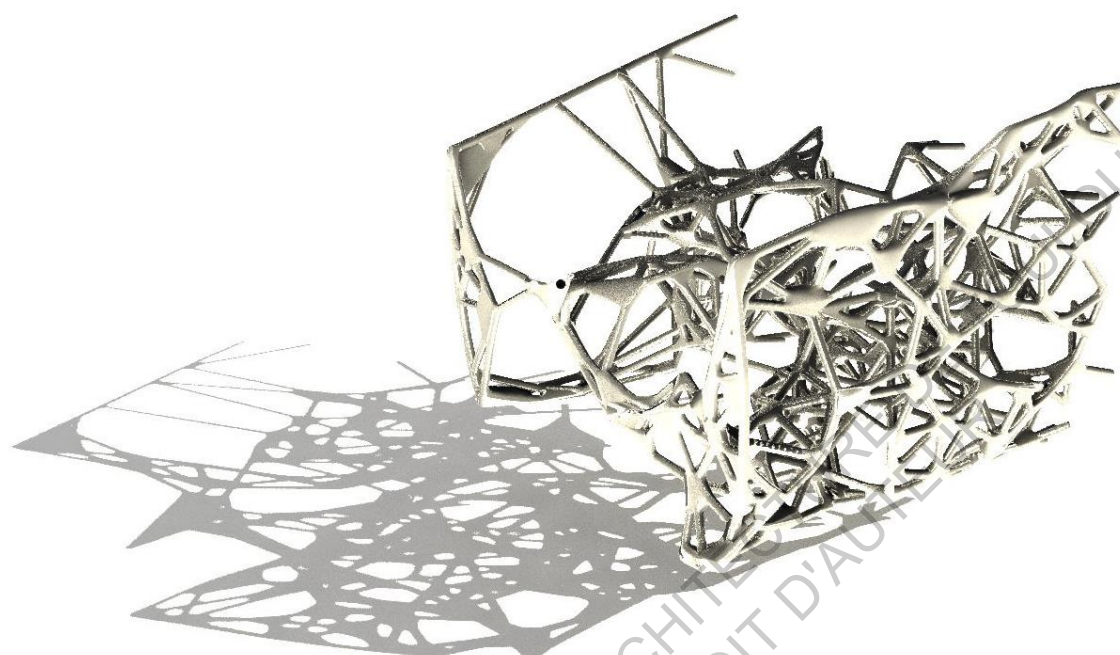


Figure 34 : Rendu 3D par lancer de rayons sur le modèle de la figure 32

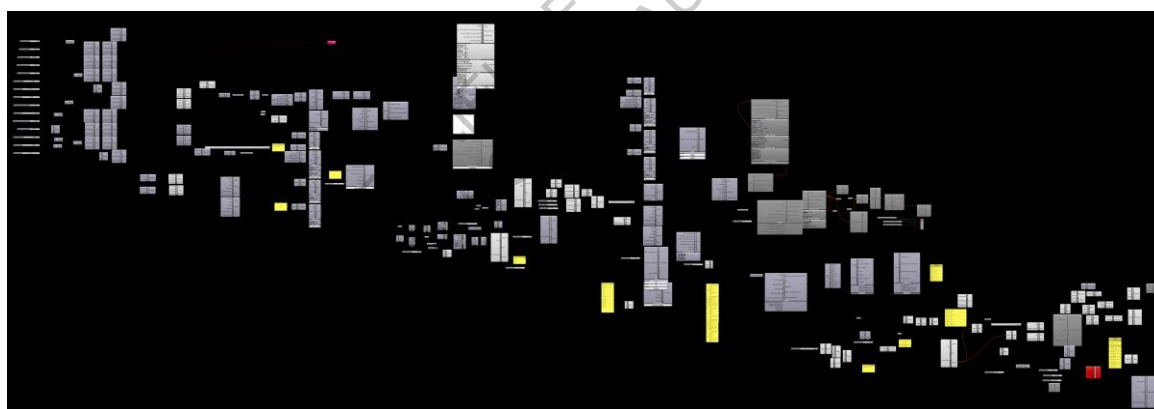


Figure 35 : Vue d'ensemble de l'algorithme de génération de passerelle

Une fois le maillage final obtenu dans Grasshopper, il suffit de le « cuire » (*bake*), il s'agit d'une opération qui transforme le maillage paramétrique en un réel objet géométrique. On pouvait voir dans l'interface Rhino, sur la figure 27, que le fichier s'appelait « *Untitled* » : en effet, il n'a pas été enregistré puisqu'il ne contient absolument aucune donnée, tout avait été paramétré dans Grasshopper. Ainsi, la *cuisson* du maillage nous permet d'avoir un réel objet géométrique, très facilement exportable puisque les maillages sont des objets géométriques assez normés (en particulier s'ils sont triangulés,

les *quads* sont un peu plus rares). Simplement composés de facettes triangulaires définies par des lignes, et orientées, ils permettent de représenter facilement une grande majorité de formes géométriques, du tétraèdre au lapin de Stanford.¹

Ainsi, nous pouvons enfin exporter ce module géométrique dans à peu près n'importe quel logiciel de rendu (les essais ont été concluants avec 3DS MAX, Maxwell, V-Ray, Artlantis et Twinmotion), Pour obtenir des résultats semblables à la figure 34, puis effectuer des incrustations comme dans les photos sur les pages de garde et de couverture.

L'export au format .STL (stéréolithographie) est également possible afin de pouvoir imprimer le modèle en 3D, sur des petites imprimantes. On utiliserait alors une réduction au 200^{ème} afin d'avoir un objet d'une douzaine de centimètres, suffisamment grand pour pouvoir l'observer en détail, mais suffisamment petit pour limiter les couts de construction.

Enfin, nous pouvons réutiliser le modèle avec le module pour Grasshopper Kuka | PRC (*parametric robot control*) qui permet de simuler l'action et de préprogrammer des bras robotisés KUKA. Il faut pour cela que l'école en tant qu'institution adhère à l'*Association for Robots in Architecture*, pour que tous les étudiants et professeurs puissent apprendre et enseigner. Nous rejoindrions alors l'IAAC de Barcelone, l'école d'architecture d'Aarhus au Danemark, l'Université de Cardiff, d'Innsbruck, d'Aachen, d'Harvard, l'institut Polytechnique de Milan... Il faudrait alors inclure dans le programme pour la nouvelle école des espaces adaptés à ces robots, mais cela nécessite une certaine vision à accompagner !

¹ Créé par Greg Turk et Marc Levoy à l'Université de Stanford en 1994, il comprend à l'origine 69451 triangles issues du scan d'un lapin en céramique, il est devenu un modèle standard pour les tests d'algorithmes géométriques. Source : Greg Turk, Marc Levoy , 1994. Zippered Polygon Meshes from Range Images In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '94*, 1994. ACM New York, NY, Etats-Unis d'Amérique.

III. COMPARAISON ET CONCLUSION

« Ce que l'on conçoit bien s'énonce clairement,
Et les mots pour le dire arrivent aisément. »¹

Il me semble que la réciproque à ces célèbres vers de Nicolas Boileau est également vraie : ce qui s'énonce clairement se conçoit bien. Ce mémoire m'a forcé à vulgariser la conception générative, en faisant autant d'analogies possibles pour en faciliter la compréhension. Il s'agissait d'un bon exercice, puisqu'à chaque fois que quelqu'un me demandait le sujet du mémoire, il me fallait l'expliquer en quelques mots. J'ai ainsi appris à la résumer comme étant :

Le mode de conception où l'ordinateur est un étudiant en première année à l'école d'architecture qui apprend à faire du projet, il tâtonne sans jamais avoir de mode d'emploi. Le concepteur fait essayer à l'ordinateur quelques dizaines de possibilités à chaque fois, qui les mélange entre elles pour en produire d'autres, et répète l'opération jusqu'à ce qu'il finisse par y en avoir une qui lui convienne.

J'ai ainsi pu aborder, peut-être longuement, les préalables à cet outils, qu'ils soient chronologiques ou conceptuels, puis expliciter avec quelques exemples ce que peut faire la conception générative, avant d'entrer dans le vif du sujet avec la conception de cette passerelle. Celle-ci a été un véritable challenge pour moi – et pour le processeur et la RAM de mon ordinateur – puisqu'il me fallait apprendre, moi aussi, par l'expérience. Je n'ai malheureusement pas pu aller jusqu'au bout la conception de la passerelle, l'étape de « relaxation » du *mesh* étant basée sur la méthode des *marching cubes* (écrite en C# dans les deux modules *exoskeleton* et *intralattice*), que les développeurs décrivent comme *non-bulletproof*, c'est-à-dire qu'elle reste imparfaite et peut produire des erreurs. Ainsi, dans le dernier rendu, nous trouvons juste un cinquième du pont, mais déjà près de 2000 poutres. Si je descendais les boules de 60cm de diamètre à 40cm de diamètre, j'obtiendrais alors près de 20000 poutres sur la totalité de la passerelle.

¹ Nicolas Boileau, *l'Art Poétique*, 1674

Je pense donc continuer l'expérience en dehors du cadre du mémoire, très certainement dans d'autres logiciels, dits métiers, utilisant des méthodes plus robustes pour le calcul géométrique et éprouvées. Il me restera donc quelques étapes : d'abord le remplissage de la passerelle pourrait être fait non pas uniformément mais avec des boules plus grosses en bas, et plus petites au-dessus afin d'assurer la résistance au poinçonnement que je n'ai pas pris en compte dans le groupe d'analyse structurelle. Le maillage supérieur serait alors plus dense, ce qui devrait également apporter une certaine qualité esthétique. Il restera ensuite une dernière étape d'optimisation du poids après avoir vérifié la résistance structurelle du *mesh* relaxé : en effet, les joints seront plus épais, et la structure plus lourde, mais également moins soumise à des moments de flexion, il y aurait donc peut-être quelques ajustements à faire dans ce troisième calcul structurel. Enfin, nous pourrions imprimer en 3D à échelle réduite notre passerelle afin d'éprouver la faisabilité. L'étape utopique étant alors de l'imprimer à l'échelle réelle.

Pourtant, nous risquons de nous confronter à un blocage normatif, de confiance, de certification des logiciels de calculs. Pourtant, tout est déjà calculé par ordinateur, il est très exceptionnel de faire encore des calculs structurels à la main. De plus, ceux-ci correspondent plutôt au travail à la chaîne que critiquait Marx que nous citions en introduction : l'ingénieur structure produit des notes de calcul qui ne sont qu'un enchaînement de vérifications. Cependant, les normes ne sont plus suffisantes pour vérifier des modèles complexes : en effet, elles sont partiellement issues de l'expérience. Certaines valeurs de calculs sont « arbitraires » et ont été données par les chercheurs à l'origine de la rédaction de ces normes, elles ne sont donc pas issues exclusivement de calculs mathématico-physiques ; et ne sont à ce titre pas absolues. Comme toutes normes, celles-ci sont faites pour répondre à la quasi-totalité des cas, mais il reste un flou lorsqu'on dépasse leur cadre. Habituellement, celui-ci est compensé par des coefficients de sécurité plus élevés, mais dans notre cas ceux-ci ne suffisent plus.

Comment effectuer des calculs de résistance et de déformation de structures aux formes qu'on ne saurait nommer spécifiquement, sans simplement caractériser leur apparence biomorphique ?

Il s'agit ici d'un problème particulièrement caractéristique dans le cadre des structures béton armé : c'est en effet un matériau anisotrope et non homogène, même à

échelle macroscopique. Et que dire de l'échelle microscopique ! Il ne s'agit que d'un agrégat de granulats à la géométrie et aux résistances mécaniques peu contrôlées, liées par la réaction chimique d'hydratation de silicates et aluminates en une pâte visqueuse presque cristallisée d'ettringite et de portlandite, auquel nous ajoutons des adjuvants pétrochimiques et des fibres minérales ou végétales ? Il semble en effet impossible de pouvoir caractériser un jour le béton de façon réellement physique, comme nous sommes capables de le faire avec les métaux.

Il risque donc de falloir accepter notre incapacité à théoriser le comportement du béton de façon parfaitement rigoureuse. Nous savons le comprendre parce que nous avons créé des modèles que nous suivons et que nous avons appris à connaître.

Les chercheurs du projet MX3D Bridge à Amsterdam ont été confrontés à ce problème, bien que le pont soit réalisé à partir d'un alliage d'acier particulièrement résistant et que les calculs aient été basés sur la mécanique des milieux continus et les codes en vigueur, les autorités ont finalement décidé que le pont ne serait rendu accessible aux passants que s'il passait un test de résistance physique. C'est-à-dire que malgré toute la technologie impliquée dans le processus, la décision finale reste de charger un maximum le pont, en espérant que ça tienne.

Est-ce là une évolution ou une régression ? Est-ce que cela signifie se décharger de toute la complexité bureaucratique et normative pour se concentrer sur l'objectif final, simple, c'est-à-dire la réponse au programme, aux usages ? Ou au contraire, est-ce que cela signifie que l'on régresse à une construction antique ou médiévale ou l'on reconnaît la puissance de la nature, la simplicité de la vérification, mais aussi notre ignorance face au comportement physique ? Ou alors faudrait-il poursuivre nos recherches pour savoir mieux caractériser les matériaux, au risque de renforcer l'industrie au détriment de l'artisanat, pour obtenir un savoir absolu ?

Le numérique était la plus grande révolution du siècle dernier. L'intelligence artificielle sera-t-elle la plus grande du XXIème siècle ? Quelle place aura le concepteur lorsque l'ordinateur aura appris à faire plus que suivre l'algorithme, mais à le créer lui-même ? Aura-t-on une civilisation semblable à celle décrite par le film Wall-E, où les humains ne sont que des consommateurs, seront-ils augmentés, cyborgs, ou seront-ils dépassés et en voie de disparition ?

A plus courte échéance, quelle sera l'avenir pour les ouvriers rendus obsolètes ? Il me semble qu'il y a là la justification au salaire universel : tout être vivant apporte au monde et doit pouvoir vivre de ce qu'il fait. Pourtant, si toutes les sources de revenus financiers deviennent informatisées ou mécanisées, les seuls bénéficiaires deviendraient alors les propriétaires de ces outils, ne laissant alors que ce qui reste en marge aux autres, c'est l'augmentation des inégalités sociales.

Au-delà de toutes ces projections à long-terme, ces perspectives d'évolutions à plusieurs générations parfois pessimistes, nous pouvons déjà estimer l'impact de la conception générative à l'échelle de notre génération. Elle pourra apprendre après chaque exemple, conçue par elle-même ou par une autre. L'intelligence artificielle saura créer d'autres intelligences artificielles. Ainsi, nous pourrions concevoir plus facilement, nous n'aurons plus à nous encombrer de spécificités techniques, il nous suffira de décrire ce que l'on souhaite, de créer une tendance pour que le *dessin* s'oriente vers celui-ci.

Nous avons effectué une expérience sur une passerelle, puisqu'il s'agissait d'un objet à la forme assez libre, mais il est (ou sera bientôt) également possible d'appliquer la conception générative à des éléments bien plus habituels : le dessin de voiries, de mobilier (urbain ou d'intérieur), mais nous pouvons aller même jusqu'au *dessin* d'appartements ou de maisons, le *dessin* de l'immeuble dans son entièreté, en optimisant vues, couts, exposition, acoustique, positionnement des commerces, accessibilité, surface de circulations...

BIBLIOGRAPHIE

AFNOR. *Eurocode 1 - Actions sur les structures*, NF EN 1991-1-1, Mars 2003.

AFNOR. *Eurocode 3 : Calcul des structures en acier*, NF EN 1993-1-1, Octobre 2005.

AFNOR. *Règles de sécurité relatives aux dimensions des garde-corps et rampes d'escalier*, NF P 01-012, Juillet 1988

AFNOR. *Barrières de sécurité routières - Garde-corps pour ponts et ouvrages de génie civil - Conception, fabrication, mise en œuvre*, XP P98-405, Avril 1998

Aidan Kurtz, Yunlong Tang, Prof. Fiona Zhao, *Intralattice V0.7.6 User documentation*, [en ligne], 2014 disponible sur <<http://intralattice.com/userdocs/>> (consulté le 26/08/2017)

Alberto Pugnale, Tomas Mendez Echenagucia, Mario sassone, Computational morphogenesis In : Sigrid Adriaenssens (dir) Philippe Block (dir), Diederik Veenendaal (dir), et Chris Williams (dir) *Shell structures for architecture*, Routledge, 2014

Autodesk, Airbus *Reimagining the futur of travel*, [en ligne] 2016 disponible sur <<http://www.autodesk.com/customer-stories/airbus>> (consulté le 03/04/2017)

ArchiLab *Naturaliser l'architecture*, Frac Centre, 2014

Hartmut Bohnacker, Julia Laub, *Generative Design: Visualize, Program, and Create with Processing*, Princeton Architectural Press, 2012

Pierre André Boutang, *L'Abécédaire de Gilles Deleuze*, Editions Montparnasse, 1996

Jean-Pierre Goulette (dir), Bernard Ferriès, (dir) *SCAN'16 Toulouse, Séminaire de conception architecturale numérique: Mètre et paramètre, mesure et démesure du projet*, PUN Editions universitaires de Lorraine, 2016

Bruce Hemingway *Evolutionary Computation* [en ligne] disponible sur <<https://courses.cs.washington.edu/courses/cse466/05sp/pdfs/lectures/10-EvolutionaryComputation.pdf>> (consulté le 03/09/2017)

Emile Littré, *Dictionnaire de la langue française*, 1877

Karl Marx, Friedrich Engels, L'Idéologie allemande, in *Philosophie*, Karl Marx, Maximilien Rubel, éd. Gallimard, coll. « Folio », 1982

Panagiotis Michalatos, Millipede March 2014, [en ligne] 2014 disponible sur <<http://www.sawapan.eu>> (consulté le 17/06/2017)

Victor A. McKusick, Dimples, Facial In: *Online Mendelian Inheritance in Man*, Johns Hopkins University, 1994

Clemens Preisinger, *Karamaba User Manual for Version 1.2.2*, [en ligne] octobre 2016 disponible sur <<http://www.karamba3d.com/news/new-manual/>> (consulté le 20/08/2017)

David Rutten *Evolutionary Principles applied to Problem Solving*, [en ligne] 2010, disponible sur <<http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles>> (consulté le 03/04/2017)

Louis H. Sullivan, *The Tall Office Building Artistically Considered*, Lippincott's magazine, 1896, p403–409.

Herbert Spencer, *Principles of Biology*, 1864

Greg Turk, Marc Levoy, Zippered Polygon Meshes from Range Images In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '94*, 1994. ACM New York, NY, Etats-Unis d'Amérique.

Vinod Srinivasan, Esan Mandal and Ergun Akleman, *Solidifying Wireframes*, Texas A&M University

Pr. Jean Weissenbach In: *Nature Genetics*, Volume 25 Number 2 Pages 235-238, 2000

TABLE DES ILLUSTRATIONS

Figure 1 : Première étape : la ligne	10
Figure 2 : Deuxième étape : l'axe	10
Figure 3 : Troisième étape : le mur	10
Figure 4 : Quatrième étape : le mur composite.....	10
Figure 5 : Le mur enrichi	12
Figure 6 : Les paramètres	12
Figure 7 : Le mur paramétré.....	12
Figure 8 : Le mur avec les paramètres modifiés	12
Figure 9 : Interface du solveur Galapagos, David Rutten	18
Figure 10 : Exemple d'un diagramme d'aptitude pour la coque à double courbure : nous observons deux optima très proches mais séparés par une vallée.....	19
Figure 11 : Vue d'ensemble d'un algorithme Grasshopper, composé de blocs et de liens : de gauche à droite : options de génération des arcs, analyse du premier ordre par Karamba, module Galapagos, représentation des sollicitations.....	26
Figure 12 : Résultats optimaux pour (de haut en bas) la chaînette, l'hyperbole, l'anse de panier, l'ogive, et l'arc de cercle. En rouge la compression maximale, en bleu la traction maximale.....	28
Figure 13 : Altitudes possibles et surface résultant d'un exemple	30
Figure 14 : Diagramme d'aptitude et ensemble des 256 solutions superposées.....	31
Figure 15 : Les sept meilleures solutions pour le pont à double courbure.....	32
Figure 16 : Définition Grasshopper : la moitié gauche compose le loft, la moitié droite l'analyse.....	33
Figure 17 : Pont imprimé en 3D par l'IAAC en décembre 2016 : diagramme du processus de développement.....	34
Figure 18 : Coupe macroscopique d'un os de hanche humain, Patrick Siemer	35

Figure 19 : Maquette d'un pont créé par optimisation topologique. Projet Autodesk Dreamcatcher.....	37
Figure 20 : Paroi de partition imprimée en 3D (Airbus / Autodesk Dreamcatcher)	38
Figure 21 : Exemple d'application du Cross section optimizer de Karamba, Clemens Preisinger, Jungwo Park.....	39
Figure 22 : Paramètres et génération des points : la moitié haute sert à dessiner la courbe au niveau des appuis, la moitié basse à mi-travée.....	42
Figure 23 : Numérotation des points et courbes génératrices.....	45
Figure 24 : Blocs permettant la création des courbes, puis des surfaces.....	46
Figure 25 : Première phase d'analyse structurelle	47
Figure 26 : Mesh de la passerelle en phase 1, représentation des déformations	48
Figure 27 : Interface Galapagos pendant la résolution.....	49
Figure 28 : Deuxième phase : remplissage par des sphères	51
Figure 29 : Passage de centre de sphères à des lignes avec triage.....	52
Figure 30 : Troisième phase : construction du modèle géométrique du pont.....	53
Figure 31 : Optimisation des sections.....	55
Figure 32 : Jointoiement des sections	56
Figure 33 : Portion de passerelle	57
Figure 34 : Rendu 3D par lancer de rayons sur le modèle de la figure 32.....	58
Figure 35 : Vue d'ensemble de l'algorithme de génération de passerelle	58

LEXIQUE

4

4^{ème} dimension : Généralement, le temps.

B

Biomorphisme : « Caractère d'une œuvre d'art de tendance non figurative dont les formes rappellent celles du monde organique. »¹

C

Courbure : Déformation impliquant une dimension supplémentaire pour décrire l'objet. *Une feuille de papier roulée doit être définie dans l'espace.* Si la courbure est double, il n'est pas nécessaire d'avoir une dimension supplémentaire, mais celle-ci devient rigide. *Les chips type pingles sont à double courbure.*

D

Définition : Fichier Grasshopper regroupant l'ensemble des blocs et des paramètres. C'est en quelque sorte le plan d'assemblage. Il s'agit d'un algorithme

¹ Larousse en ligne

E

Espace : plan extrudé. Non-courbé. Défini par trois axes.

Exoskeleton : module de jointolement de lignes en tubes puis en un *mesh*.

F

Fitness Landscape : diagramme d'aptitude. Chaque point sur ce diagramme correspond à un individu, dont l'aptitude est représentée sur l'axe vertical, X et Y représentant les valeurs de deux gènes.

G

Galapagos : Solveur évolutionnaire inclus dans Grasshopper

Grasshopper : Plug-in pour programmation visuel inclus dans Rhinocéros

H

Heuristique : Qui consiste ou qui tend à trouver.

I

Isotrope : dont les résistances mécaniques sont équivalentes quelle que soit la direction.

K

Kangaroo : module de simulation physique

Karamba : module d'analyse statique

L

Loft : surface passant par des courbes génératrices.

M

Maillage : surface quadrillée ou triangulée en portions planes.

Mesh n.m. : voir maillage.

P

Plan : surface non-limitée sans courbure. Toujours défini dans un repère orthonormé à deux axes.

R

Rhinocéros : Logiciel de modélisation géométrique 3D

Robot : Être mécanique à commande numérique

S

Surface : plan limité pouvant admettre une ou deux courbures.

T

T, grandeur, direction supplémentaire représentant le temps.

Trabécule, n. f., du latin *trabes*, poutre, « petits prolongements de substance osseuse qui, entre-croisés, limitent les cavités médullaires du tissu spongieux dans le voisinage du canal des os longs. »¹

U

U, grandeur : direction principale d'une surface

V

V, grandeur : direction secondaire d'une surface

Voxel : Equivalent tri-dimensionnel du pixel.

X

X, grandeur : direction principale d'un objet linéaire.

Y

Y, grandeur : direction secondaire d'un objet plan

Z

Z, grandeur : direction verticale d'un objet spatial.

¹ Dictionnaire de la langue française (1872-1877), Emile Littré

Mémoire de séminaire : conditions de consultation

Ce document est protégé par le droit d'auteur (art. L. 112-1 du Code de la propriété intellectuelle).

L'auteur du document accorde les droits d'usages suivants :

	OUI	NON
Diffusion numérique limitée aux étudiants de l'ENSA Toulouse		
Diffusion numérique limitée au réseau Archirès		
Diffusion internet (Dumas)		
Exposition		