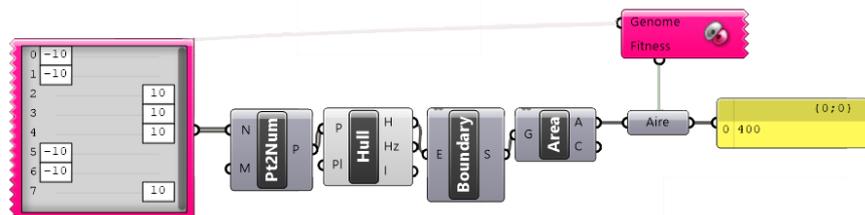


Session 3 : Outils pédagogiques

Architectures numériques : pratiques de la modélisation paramétrique



Les soutiens d'EduBIM 2019



Architectures numériques : pratiques de la modélisation paramétrique

JUTAND Alix ¹

¹ *Ecole Nationale Supérieure d'Architecture de Toulouse*

Résumé

Cet article présente un ensemble d'explorations et de réflexions menées à l'ENSA de Toulouse dans le cadre du séminaire Architectures Numériques. Il s'agit, au travers de cet enseignement, d'expérimenter les potentialités offertes par les outils numériques et de questionner leur utilisation par l'architecte à la fois dans son processus de création et dans la gestion de projet. L'article porte ainsi sur une expérimentation de la modélisation paramétrique menée au sein de ce dispositif pédagogique en tant qu'étudiante.

Mots clefs

Architectures numériques, modélisation paramétrique, conception, forme architecturale, algorithme évolutionnaire

1. Introduction

Le séminaire Architectures Numériques, enseignement de Master animé au sein de l'Ecole Nationale d'Architecture de Toulouse, porte sur l'utilisation des outils numériques d'aujourd'hui et de demain au service du projet architectural. Il permet aux étudiants d'explorer les nouveaux processus de conception et de partage d'information dans ce contexte technologique. Cet enseignement nous amène ainsi en tant qu'élèves architectes à nous questionner sur la transition numérique ainsi que son impact sur l'évolution de la création architecturale et de la profession.

La première partie de ce séminaire est organisée comme une phase de découverte et de pratique des différents outils logiciels pour la modélisation paramétrique et la pratique collaborative BIM. Ponctué de phases théoriques et d'analyse, elle tente de répondre aux interrogations que peut induire l'exploration du numérique et pratiques architecturales. C'est dans ce cadre que s'inscrit cet écrit.

Dans un second temps, l'enseignement accompagne les étudiants dans la rédaction d'un mémoire de Master présentant la réflexion de chacun sur une problématique propre, en lien avec les recherches et interrogations soulevées au long des premières phases.

Cet article présente donc ici quelques expérimentations menées dans le cadre de ce séminaire Architectures Numériques, en croisant pratique et analyse des outils de modélisation paramétrique au service de la création architecturale.

2. Interroger le numérique comme exploration des processus de conception de formes

La modélisation paramétrique, permise par la puissance de calcul des ordinateurs actuels, permet la création de formes à partir d'éléments/valeurs variables préétablis appelés paramètres dont dépend la définition de l'objet architectural. Ainsi la forme générée n'est pas fixée dans une géométrie figée, elle reste adaptable et évolutive tout au long du processus de conception.

Cette pratique a pour caractéristique de se présenter sous la forme d'une modélisation encodée, bâtie à partir d'un algorithme écrit par l'architecte lui-même. L'objet architectural est ensuite généré par exécution du code informatique.

Concrètement, on dispose d'une interface de codage graphique tel que le logiciel Grasshopper qui permet de réaliser des programmes à partir de blocs non modifiables (de type boîte noire), chacun encodé pour une fonction précise. A partir de ces fonctions, le concepteur monte un réseau d'opérations mathématiques qui donneront forme à l'objet architectural. Ce code génératif fixe est ensuite adaptable en fonction des différents besoins du projet par l'ajout de paramètres introduits en entrée qui autorisent une variation de la forme de l'objet sous contrainte. A chaque modification d'un paramètre, l'ordinateur recalcule la solution mathématique puis la transcrit dans l'interface graphique. Cette modélisation est donc très interactive et permet de visualiser instantanément la solution. Enfin, il est possible de figer la géométrie obtenue dans un certain état et de la récupérer sur n'importe quel logiciel de conception architecturale pour poursuivre la modélisation hors du contexte paramétrique.

Si l'interface de codage graphique peut paraître intuitive elle reste quand même complexe dans son utilisation. En effet, le concepteur doit s'approprier le langage de l'outil pour parvenir aux résultats graphiques escomptés. De plus les blocs de codage étant fixes, on se trouve parfois limité dans l'expression mathématique d'une fonction sculpturale simple. En effet, le processus de création se fait par un ensemble de transformations identifiables à travers des verbes d'actions et de conditions qu'il faut ici traduire en langage mathématique.

Par ailleurs, le principe de boîtes noires à venir assembler est aussi contraignant et peut interroger sur la place laissée par le logiciel à l'architecte dans la génération de la forme. En d'autres mots, tout se passe comme si le concepteur était autorisé à créer dans la limite de ce que le logiciel l'autorise à faire. Tout en ouvrant des champs de conception révolutionnaires, le numérique contraint aussi le langage de l'architecte à celui du logiciel.

La force de ces logiciels paramétriques réside aussi dans la grande palette d'applications et de calculs pouvant y être intégrés. Sous Grasshopper des plugs-in de calcul de structure (Karamba) et de résolution de systèmes physiques (Kangaroo) permettent d'optimiser les formes créées et de résoudre des calculs géométriques très complexes : reprise de charges, comportement de la forme sous contraintes, création de maillages, pavages, etc. Le logiciel Grasshopper offre ainsi la possibilité d'intégrer un large panel de contraintes dans la définition de l'objet architectural. Le travail de conception n'est donc plus seulement axé sur des choix esthétiques ou logiques, des réflexions structurelles, physiques et de fabrication sont prises en compte. Avec cet outil numérique, l'architecte peut reprendre le contrôle sur l'objet qu'il dessine et donc sur le projet au sens large. En effet, il intègre dès la définition de la forme l'ensemble des éléments impactant le processus. Son travail se retrouve à nouveau au centre de la réflexion de l'équipe de conception et de la prise de décision.

Les modeleurs paramétriques sont aussi des assistants d'exception dans l'exploration de formes. Ainsi, des algorithmes dits évolutionnaires reproduisent le mode de fonctionnement des systèmes naturels en utilisant le langage de la génétique. Au lieu de résoudre analytiquement les équations complexes d'un problème à plusieurs variables, le logiciel utilise la notion de gène. Ainsi le solveur évolue vers la solution du système d'équations en évaluant la proximité de chaque état obtenu avec le résultat attendu.

Le processus de calcul se base sur les paramètres variables du système appelés gènes. Le logiciel lance une première génération aléatoire de génomes dans l'espace de résolution du problème posé et évalue leur adéquation avec la solution recherchée. A l'issue de ce premier tour, les génotypes des états les plus favorables sont croisés entre eux afin de converger vers la (ou une) solution du problème posé. Au cours du processus de sélection génétique, le programme développe aussi des mutations, c'est-à-dire des solutions hors du champ réduit de solutions définit au fur et à mesure de l'avancée du programme, afin de vérifier qu'une solution plus optimale n'aurait pas été écartée.

Si cette méthode de calcul est particulièrement puissante, elle démontre aussi par son processus certaines faiblesses. En effet, l'introduction de mutations est parfois insuffisante pour extraire l'algorithme de son processus de réduction de population. Une solution plus optimale peut alors être omise. De plus, dans le cas d'un problème à plusieurs solutions, le logiciel converge seulement vers l'une d'entre elles. Le choix de l'utilisateur entre ces différentes possibilités se trouve alors réduit. Enfin, les conditions de départ (valeurs d'entrée des paramètres) au début de l'exécution du programme influencent très fortement la solution finale. Le concepteur se doit donc d'être vigilant quant aux réponses apportées par cet outil et intervenir dans le paramétrage du processus de résolution pour obtenir les résultats désirés.

Par ailleurs, la recherche de solution est lente et doit être reconduite à chaque modification du programme. Lorsque l'analyse évolutionnaire se fait de façon multicritère, c'est-à-dire que la solution recherchée dépend de plus d'un seul gène, la simulation devient lourde. Il peut donc être utile d'utiliser les méthodes d'intelligence artificielle intégrées à Grasshopper. Le plug-in Crow, basé sur des réseaux de neurones artificiels, permet ainsi d'apprendre au programme à reconnaître par lui-même un certain type d'éléments et donc de réaliser une pré-sélection des solutions.

Ici, la complexité du programme devient un frein évident à l'utilisation de ces méthodes numériques. En effet, un temps non négligeable de formation et de pratique est nécessaire avant de pouvoir maîtriser ces outils. Il est donc compréhensible que de nombreux architectes restent réticents à l'appropriation de ce type de modélisation. Le concepteur qui n'a pas une connaissance approfondie du modéleur peut rapidement se retrouver limité dans sa création et c'est ce qui sans doute effraie avant tout les sceptiques. Pour dépasser ce sentiment de dépossession de sa création, le concepteur doit prendre le temps d'évaluer sa pratique du numérique et comprendre les codes de développement, afin de gérer son algorithme de façon efficace et cohérente en lien avec son processus de création.

3. Expérimentation de la modélisation paramétrique sous Grasshopper

L'objectif de cet exercice était à la fois de me familiariser avec la modélisation paramétrique de Grasshopper et d'expérimenter les différents outils de calcul du logiciel.

Dans le cadre de mes explorations architecturales de l'atelier de projet architectural du semestre, j'en était venue à la question de l'optimisation d'une forme et de son système structurel. En effet, mon travail portait sur une architecture d'école mobile pensée comme un kit devant être facilement transportable.

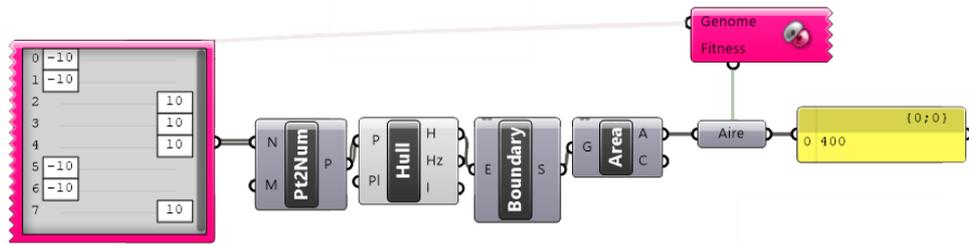
Après avoir suivi l'initiation au paramétrique menée au cours de ce séminaire Architectures Numériques, j'ai voulu expérimenter l'optimisation de formes à partir d'un ensemble de contraintes définies en entrée, à savoir dans le cadre de mon projet, et en première approche, obtenir une surface au sol maximale, limiter les dimensions des éléments structurels ainsi que le poids de la structure.

Cette partie se propose de décrire les expérimentations menées au cours de cet exercice.

Définition des contraintes morphologiques

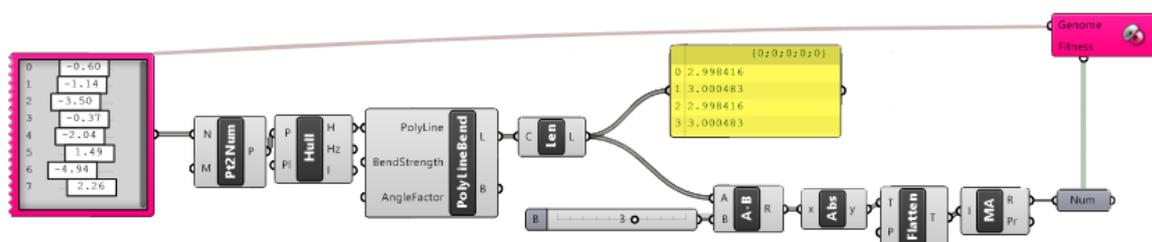
Maximisation de la surface au sol

On suppose une géométrie (plan de l'école) définie à partir de 4 points d'abscisse et d'ordonnée comprises entre -10 et 10. A l'aide d'un algorithme évolutionnaire monocritère on cherche à déterminer la forme qui maximise la surface au sol. Après exécution du programme on obtient comme attendu le carré maximal autorisé d'aire égale à 400.

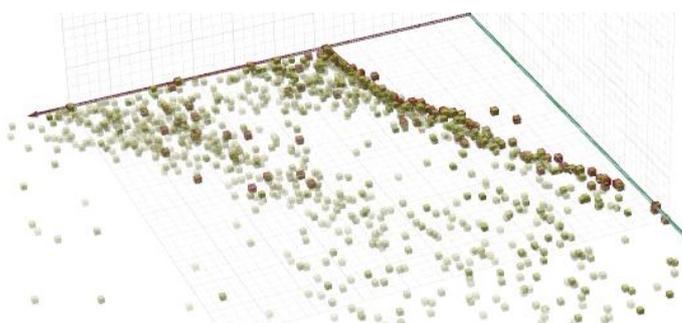


Minimisation de la distance entre points

A présent on considère les points définis en première partie comme les points d'ancrage des poteaux de la structure porteuse de cette école. Pour que la structure soit facilement transportable nous limitons arbitrairement cette distance à 3m. Après calcul d'optimisation nous obtenons bien un parallélogramme dont chaque côté approxime les 3m.



Maintenant nous décidons de combiner ces deux contraintes opposées dans le programme évolutionnaire. Il s'agit maintenant d'un programme multicritère. Nous obtenons alors le graphique suivant, propre au plug-in Octopus qui donne différentes solutions évaluées par le programme. Le concepteur peut alors visualiser et choisir une forme générée qui répond au mieux aux deux contraintes imposées.

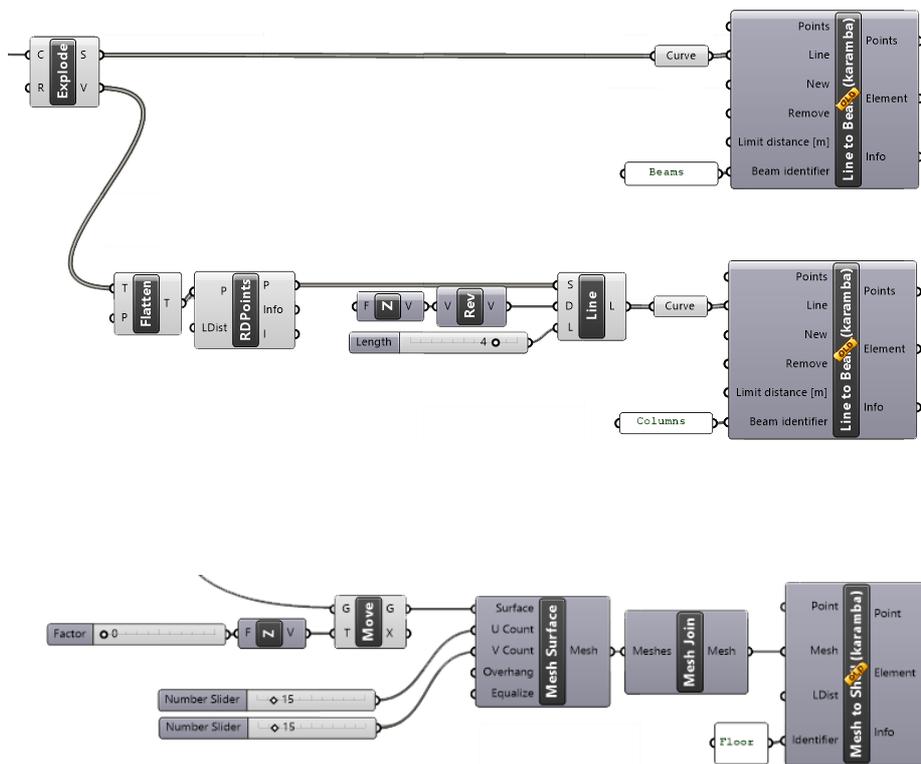


Cependant on constate qu'il est difficile ici de trouver, au milieu des solutions proposées, la forme optimale qui serait théoriquement un carré de 3m de côté. Il s'agit donc de générer un ensemble de formes répondant plus ou moins aux critères fixés plutôt que d'obtenir la solution exacte du problème.

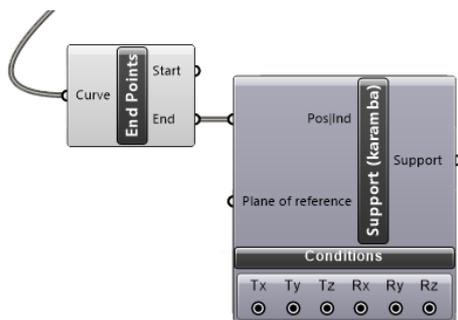
Modélisation des éléments structurels à l'aide de l'outil Karamba

La deuxième partie de l'exercice se concentre sur la modélisation du système structurel de la forme définie en première partie, afin de prendre en compte par la suite ces contraintes structurelles réelles dans le processus de conception de l'objet architectural. De plus, cette partie était pour ma part une opportunité pour explorer les possibilités du plug-in de calcul structurel Karamba non développé au cours du séminaire.

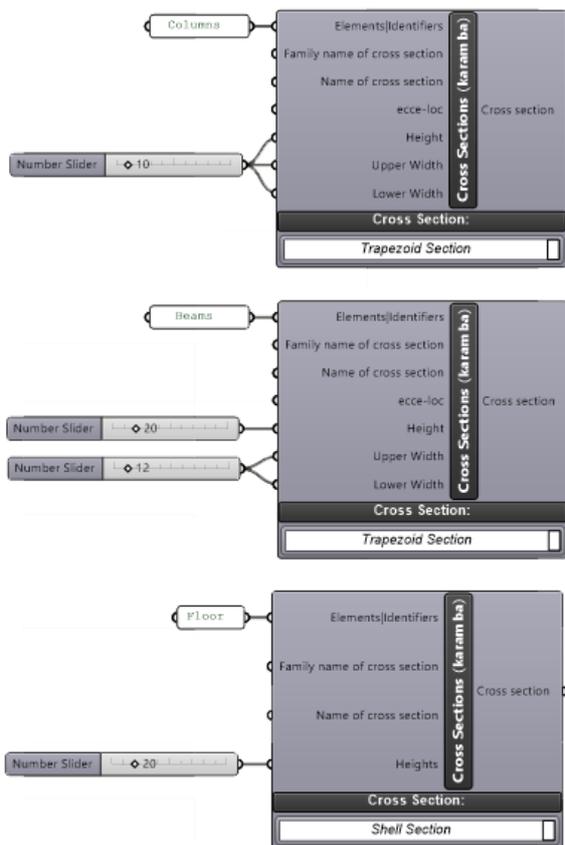
Définition des poteaux, poutres et plancher.



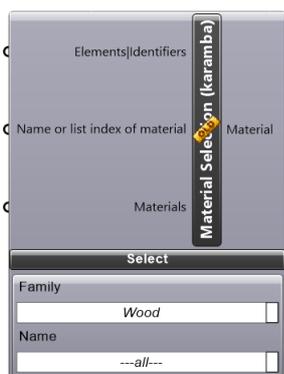
Détermination des conditions d'appui du système constructif.
On considère ici des encastresments en pied de poteaux.



Définition des sections des éléments structurels.

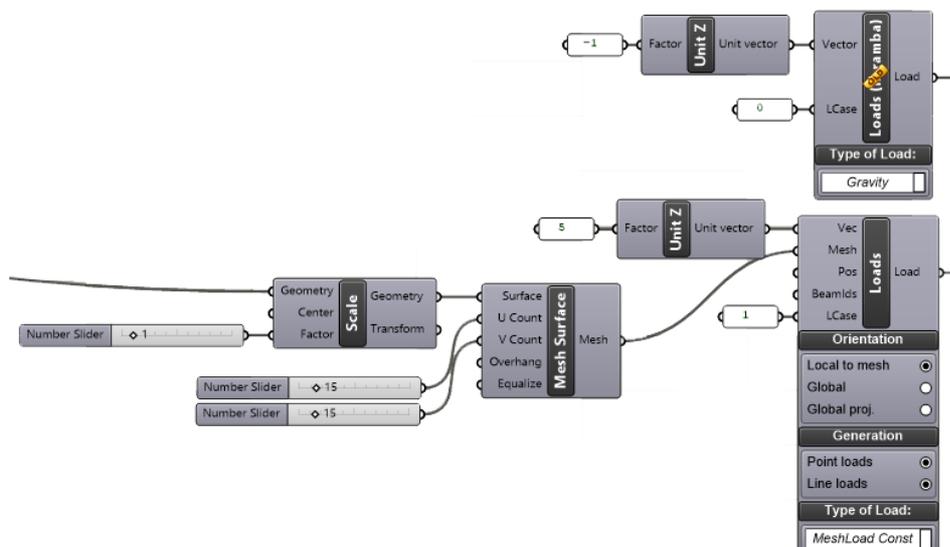


Choix du matériau de construction.

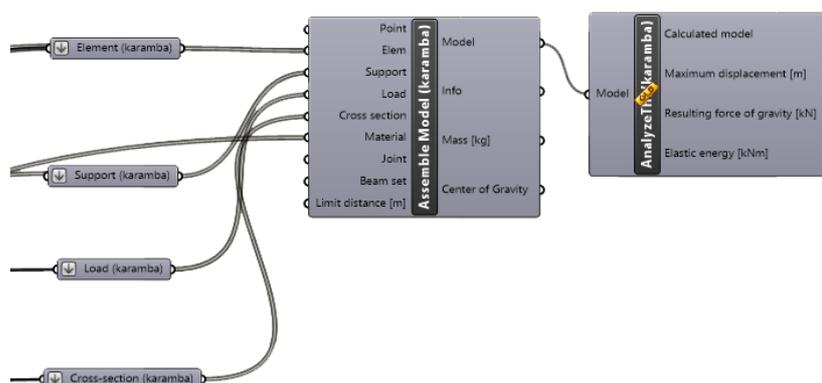


Définition des charges extérieures appliquées au système.

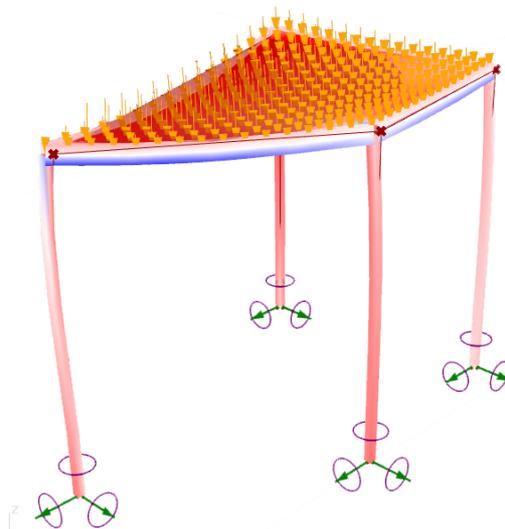
Ici sont considérés le poids de la structure ainsi qu'une charge de valeur 5 appliquée sur l'ensemble de la dalle



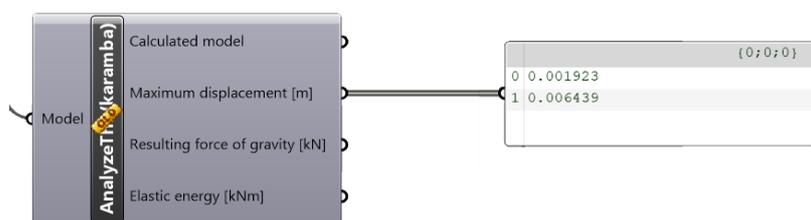
Compilation du modèle et calcul numérique de la structure.



On obtient le résultat graphique suivant qui permet de vérifier visuellement le bon comportement de la structure.



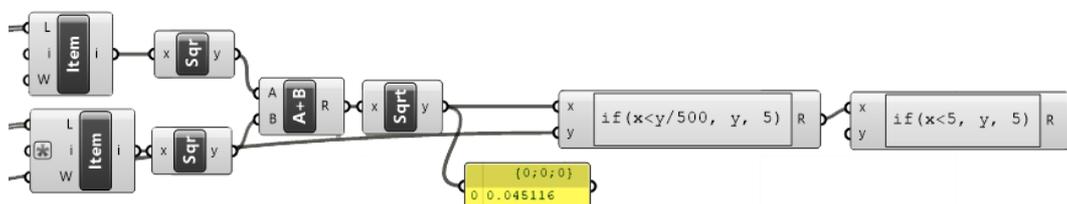
Le logiciel permet aussi d'obtenir la valeur numérique du déplacement maximal de la structure.



Retour aux contraintes morphologiques

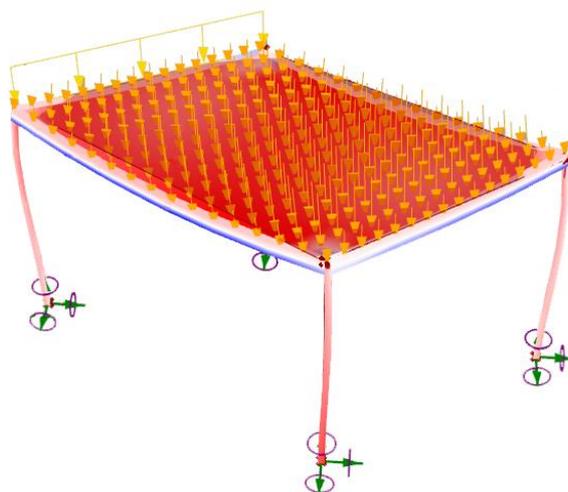
On vient maintenant limiter le déplacement maximal de la structure à $1/500$ de la portée la plus grande et réintroduire alors la portée la plus grande autorisée par le calcul structurel dans le calcul d'optimisation de la première partie de l'exercice. Ainsi au lieu de limiter la portée des poutres à une longueur arbitraire de 5m, les poutres doivent maintenant se limiter en longueur au minimum entre la valeur autorisée par le calcul structurel et la valeur de 5m pour permettre un transport facile des éléments.

Pour cela le logiciel autorise l'utilisateur à entrer des expressions mathématiques ce qui permet de contourner les limitations induites par les blocs de fonctions prédéfinis.

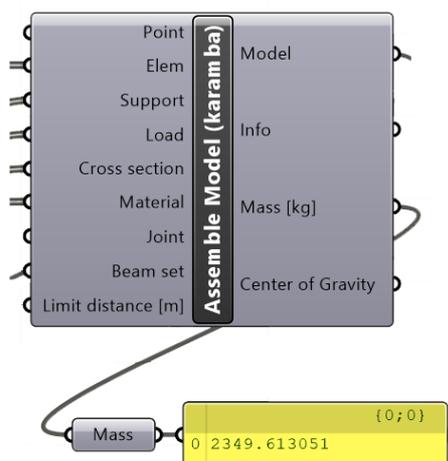


A présent l'algorithme créé cherche la forme optimale permettant à la fois d'obtenir une surface au sol la plus grande possible tout en limitant les portées des poutres telles que leur déformation soit inférieure à un déplacement de $L/500$ (contrainte structurelle) et que leur longueur soit inférieure à 5m (contrainte de transportabilité de la structure). Si on relance le calcul du code ainsi créé, l'exécution est beaucoup plus longue. En effet, à chaque itération d'optimisation, l'algorithme recalcule le modèle structurel de l'objet.

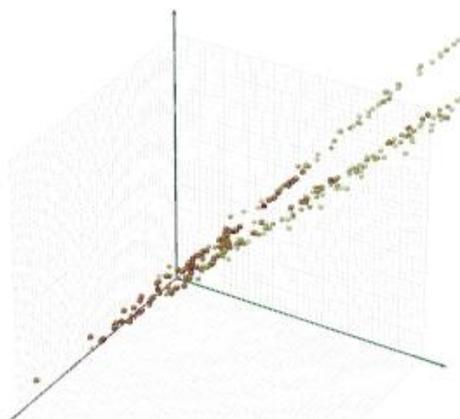
Après quelques instants, on obtient par exemple ce résultat intéressant qui répond aux critères demandés : les longueurs de poutres vont de 3,75m à 5,3m pour une surface au sol de 16,7m, le déplacement maximal de la structure vaut $0,008\text{m} < 0,01\text{m} = L/500.1$



Par ailleurs, la génération de la forme étant totalement paramétrique nous pouvons à tout instant décider de changer les dimensions des éléments structurels, les valeurs de forces appliquées au système, les contraintes imposées ou bien le nombre de points d'entrée définissant la forme générée. Par exemple nous décidons d'ajouter une nouvelle contrainte : limiter le poids de la structure pour une maintenance facilitée. Nous obtenons cette valeur très facilement grâce au bloc « model » de Karamba.

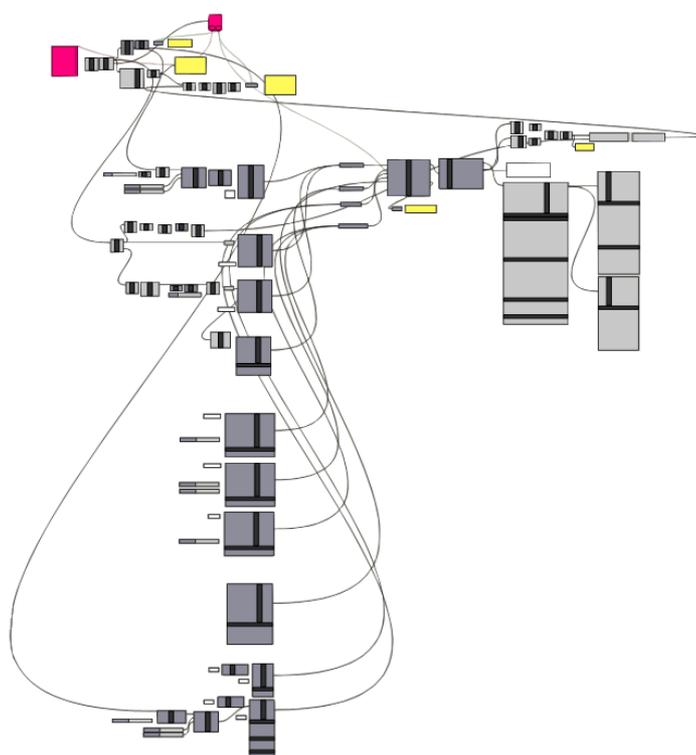


Le graph des résultats de l'algorithme évolutionnaire multicritère est maintenant à trois dimensions (3 critères). L'utilisateur peut donc sélectionner, à partir de ce nuage de solutions, la forme qui répond le mieux aux contraintes qu'il a imposées.



Nous pouvons, à travers ce court exemple, mesurer la force d'un logiciel comme Grasshopper dans la définition d'un objet architectural. Le concepteur peut non seulement créer une forme paramétrique qu'il pourra modifier en fonction de différents critères, mais il peut directement utiliser le code du logiciel pour résoudre des problèmes d'optimisation simples à formuler mais très complexes à résoudre par les méthodes de calcul analytique. L'architecte peut aussi intégrer des contraintes techniques à son modèle pour évaluer la faisabilité de son projet et faciliter les échanges avec les bureaux d'ingénierie. Le programme devient un fichier de collaboration entre les différentes parties prenantes du projet dès la génération de forme.

Bien sûr on peut toujours critiquer la complexité et la relative longueur d'exécution d'un tel programme, mais au vu des possibilités qu'il offre ces critiques restent mineures.



La problématique qui se pose ici plus sérieusement et celle du « lâcher prise » qu'expérimente le concepteur lors de la génération de forme. Après avoir défini les paramètres de génération et imposé des contraintes, l'architecte se trouve quelque temps dépossédé de l'objet qu'il construit. Le programme en prend momentanément le contrôle avant de présenter au travers de l'interface

graphique des formes sorties comme par magie. L'utilisateur doit donc être conscient de l'impact du code dans la génération de son projet pour pouvoir l'extraire de cette démarche paramétrique et lui donner une signification.

4. Conclusion

L'exemple développé, couplant modélisation paramétrique et méthodes numériques de calcul, illustre ici la puissance des outils numériques et leurs potentialités pour la création architecturale. Par l'utilisation pertinente de ces méthodes de travail, la conception permet ainsi d'intégrer un ensemble d'éléments et de contraintes dès la définition de la forme en offrant, tout au long du processus de création, la possibilité d'évolution de l'objet architectural à l'aide de paramètres variables.

Avec ces outils numériques, le travail de l'architecte peut intégrer des dimensions de projet plus larges en remettant le processus de morphogenèse architecturale au centre de la réflexion de l'équipe de conception et de la prise de décision. D'autant plus qu'il est possible aujourd'hui, à l'aide des formats IFC, d'exporter par la suite le fichier vers les logiciels BIM. Se dessine alors une nouvelle méthode de gestion projet en favorisant une collaboration entre acteurs autour d'un objet unique partagé tout au long du processus de conception du projet, en considérant aussi une modélisation paramétrique de son architecture.