

École polytechnique de Louvain

Conception et calcul des structures composées d'éléments empilés de type droxels sur base d'une approche paramétrique

Auteur : **Grégoire GUILMIN**

Promoteur : **Pierre LATTEUR**

Lecteurs : **Igor BOUCKAERT, João PACHECO DE ALMEIDA, André
Stephan**

Année académique 2020–2021

Master [120] : ingénieur civil des constructions

Remerciements

Je tiens particulièrement à remercier mon promoteur de mémoire, le Professeur Pierre Latteur, École Polytechnique de Louvain (EPL), Université Catholique de Louvain (UCL) pour sa supervision et ses conseils. Mais également pour m'avoir fait confiance et laissé des libertés dans la conception du programme tout en étant présent pour m'orienter.

Ensuite et à l'avance, je remercie Igor Bouckaert, João Pacheco De Almeida et André Stephan pour la lecture de ce mémoire et leur présence en tant que jury à ma défense orale.

Enfin, je suis très reconnaissant envers mes parents qui ont passé du temps à la relecture et je remercie Chloé De Ketele pour ses questions pertinentes à propos du programme.

Table des matières

Remerciements	I
Table des matières	III
Avant-propos	1
Rhinoceros	1
Grasshopper	1
Droxel	1
Droxelisation	1
1. Introduction	3
1.1. Contexte général	3
1.2. Objectif de ce mémoire	3
2. Le Droxel	5
3. Principes utilisés pour l'algorithme	7
3.1. Processus de détection des sous-structures	12
4. Code Grasshopper	16
Flow Charts	18
4.1. Clusters	18
4.1.1. 01 – Base	20
4.1.2. 011 - Grid	21
4.1.3. 02 – Layer	23
4.1.4. 021 – Droxel	25
4.1.5. 022 – Position	27
4.1.6. 023 – Stability Check	28
4.1.7. 0231 – Step Sub	30
4.1.8. 0232 – Sub Edit	32
4.1.9. 02311 – Lowers	36
4.1.10. 02312 – Uppers	37
4.1.11. 023121 – Step Uppers	39
4.1.12. 02313 – Substructure	41
4.1.13. 02314 – Gravity Center	42
4.1.14. 02315 – Gravity Test	44
4.1.15. 02316 – Lowest Statement	46
4.1.16. 02317 – Sub Statement	47
5. Application du code Grasshopper avec des exemples	48
5.1. Exemple – A	50
5.2. Exemple – B	53
6. Droxels de changement de pente	58
6.1. Code Grasshopper pour la couche de transition	61
6.1.1. Cluster Pre-transition Droxel	62
6.1.2. Cluster Transition Droxel	63
7. Pistes d'amélioration du code Grasshopper	66
Droxel par droxel	66

Coefficient de frottement	66
8. Apporter une solution de droxelisation stable	70
8.1. Galapagos.....	70
8.1.1. Exemple C avec l'interface Galapagos	71
9. Conclusion.....	75
10. Table des figures.....	77
11. Table des images	78
12. Table des tableaux.....	79
13. Références bibliographiques	80
14. Annexes.....	81

Avant-propos

Les définitions suivantes sont des outils pour permettre au lecteur une bonne compréhension de ce mémoire.

Rhinoceros

« Rhinocéros 3D aussi communément appelé Rhino 3D ou Rhino est un logiciel de dessin 3D qui permet de modéliser une grande variété de formes – sans enfermer et limiter la créativité » [\[1\]](#)

Grasshopper

« Grasshopper (GH) est un plugin de Rhinoceros 3D. Un plugin (encore appelé module d'extension ou module externe) est un programme complétant les fonctionnalités d'un logiciel. Le plugin GH permet de créer des modèles paramétriques sur Rhinoceros grâce à de la programmation visuelle. Un modèle paramétrique définit une géométrie dynamique où la modification de valeurs des paramètres se répercute sur l'ensemble de celle-ci. » [\[2\]](#)

Droxel

« Le mot droxel est ici défini comme un mot-valise de drone et de voxel, tandis que le voxel désigne un pixel tridimensionnel. Un droxel est donc une unité élémentaire de maçonnerie 3D qui permet de façonner un nombre infini de conceptions structurelles, parfois sans utiliser d'échafaudage. » [\[3\]](#)

Droxelisation

« Transformer la structure en une combinaison d'éléments compatibles avec les drones, comme les droxels. Ce processus pourrait être appelé "Droxelisation". » [\[3\]](#)

1. Introduction

1.1. Contexte général

Le monde de la construction est en perpétuelle évolution mais certains aspects évoluent moins rapidement que d'autres. La main d'œuvre dans ce domaine reste encore à l'heure d'aujourd'hui indispensable. Dans la construction, les ouvriers sont les plus vulnérables face au risque d'accident. Mais le développement d'un autre secteur étroitement lié, celui des nouvelles technologies, est en train de changer notre manière de construire. Évidemment, les robots et les drones sont déjà mis à contribution sur chantier mais principalement avec l'assistance humaine. Alors oui, les machines facilitent la tâche de l'homme. Mais est-il possible de diminuer au maximum l'intervention de ce dernier sur les chantiers de construction ? Pour cela il faut repenser notre façon de construire et également réinventer nos éléments de construction.

Le Professeur Pierre Latteur de l'École Polytechnique de Louvain-la-Neuve a imaginé et s'est lancé depuis lors dans le domaine de la construction à l'aide de drones, et de ce fait dans l'étude d'éléments compatibles comme les droxels. Les droxels sont des éléments structurels empilables. Ils ont été développés pour répondre à des propriétés précises, comme une certaine tolérance de pose pour pallier aux erreurs de précisions des drones. Ce type de bloc permet de construire un grand nombre d'architectures en trois dimensions tout en assurant une bonne stabilité structurelle étant donné que la géométrie particulière du droxel autorise le contact avec jusqu'à dix autres blocs et donc de créer des entités rigides à part entière. Un atout non négligeable de l'utilisation des droxels est qu'un seul modèle géométrique est suffisant pour concevoir des murs en coin, des structures autoportantes, qui ne nécessitent pas de structure temporaire pour leur mise en œuvre, comme des arches ou des porte-à-faux.

Tous les avantages du droxel lui concèdent d'être également compatible pour des structures que tout un chacun pourrait construire lui-même de type abri de jardin, petit hangar, etc.

1.2. Objectif de ce mémoire

Pour réaliser des structures autoportantes en droxels, que ce soit par des drones ou bien par des hommes, il est nécessaire de calculer la position précise de chaque bloc. C'est parfaitement faisable pour des petites et simples structures. Néanmoins, lorsque l'architecture se complexifie, placer les droxels aux bons endroits pour assurer la stabilité structurelle de la construction est nettement moins évident. Automatiser le processus de transformation de quelle que structure que ce soit en droxels peut se faire à l'aide de programmes de calculs et de modélisation.

La finalité de ce mémoire est de concevoir un code interactif à l'aide du programme Grasshopper capable de droxéliser toute structure tout en assurant leur stabilité globale à chaque étape de la mise en place des droxels. Le plugin Grasshopper lié à Rhino ne demande pas des connaissances en programmation très élevées mais permet une programmation visuelle et paramétrique [5].

Ce code se base sur un algorithme présenté dans un article scientifique intitulé « Feasibility study for drone-based masonry construction of real-scale structures » rédigé par Sébastien Goessens¹, Caitlin Mueller² et Pierre Latteur¹ [3].

L'objectif est réel car un tel programme qui propose une solution stable à chaque instant de la construction, et pas seulement pour le résultat final, peut définir un ordre de pose pour chaque droxel et donner ensuite les instructions adéquates d'exécution aux drones. Droxeliser d'une telle manière signifierait créer une synergie entre l'architecture et le dimensionnement structurel. Digitaliser l'ensemble des processus de la construction permet une coordination et une coopération entre les étapes de la conception et ce jusqu'à la mise en œuvre sur chantier. Au final, ce processus optimise les coûts et offre un gain de temps depuis la conception jusqu'à la réalisation.

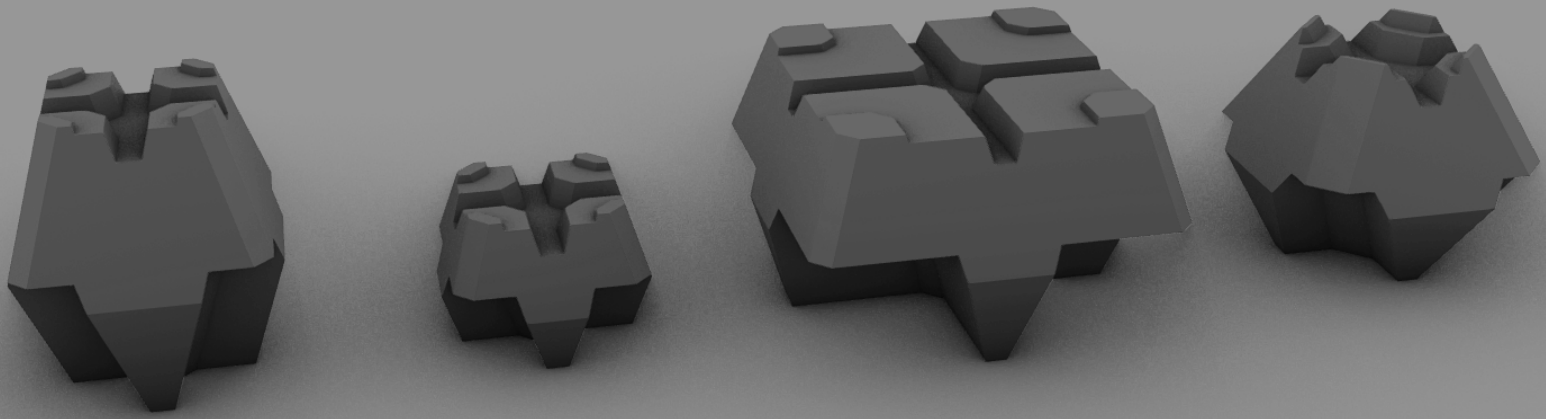
J'ai organisé travail qui suit de la manière suivante :

- Tout d'abord, je détaille la géométrie du droxel et j'explique les principes généraux utilisés dans mon code Grasshopper.
- Ensuite, je présente mon code Grasshopper en parcourant tous les clusters (portions de code) qui le composent.
- Puis je montre son application avec un exemple stable et un exemple instable.
- Je poursuis après avec une section sur les droxels de changement de pente qui permettent d'affiner la droxelisation.
- Pour finir, j'évoque des pistes d'améliorations concrètes et des idées d'ordre général sur le code.
- Après quoi j'expose et teste plus en détail une des pistes pour apporter une solution de droxelisation stable.

Ce code est le fruit de mon travail et reste ouvert à toutes améliorations futures pour permettre plus globalement une concrétisation de la conception paramétrique des structures de type droxels.

¹ Université catholique de Louvain UCLouvain, École polytechnique de Louvain, génie civil et environnemental, Belgique.

² Massachusetts Institute of Technology MIT, Department of Architecture, Building Technology Program, United States of America.



2. Le Droxel

Le droxel a une forme particulière définie par huit paramètres (H , L , E , R , B , Cr , F et A) géométriques. Chaque combinaison différente de ces paramètres aboutit à une géométrie unique. La *Figure 1* montre comment ces huit facteurs dimensionnent le droxel. Il est important de bien les visualiser car l'emplacement des blocs est fonction de certains de ces paramètres.

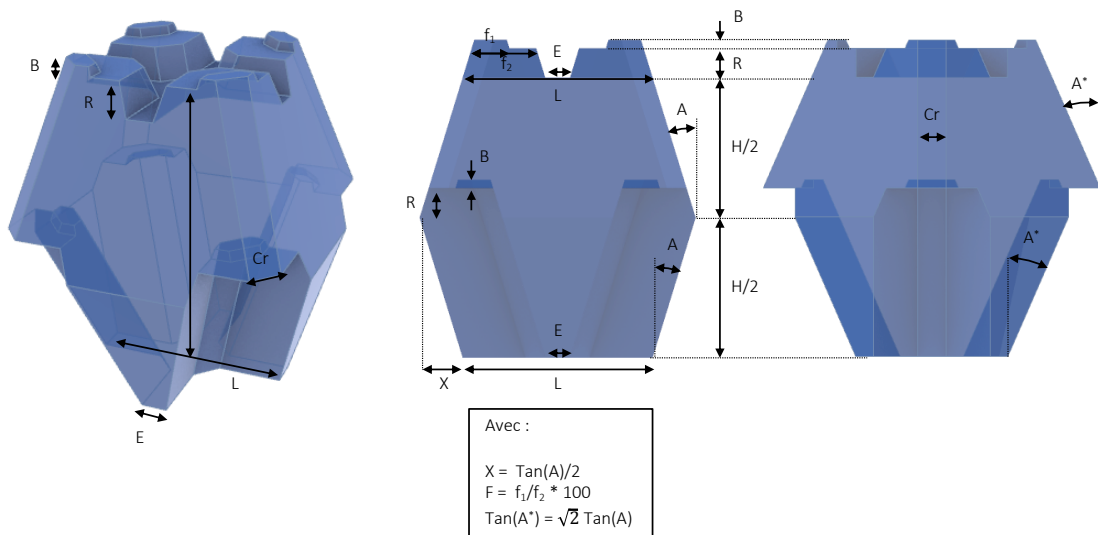


Figure 1 : Géométrie du droxel

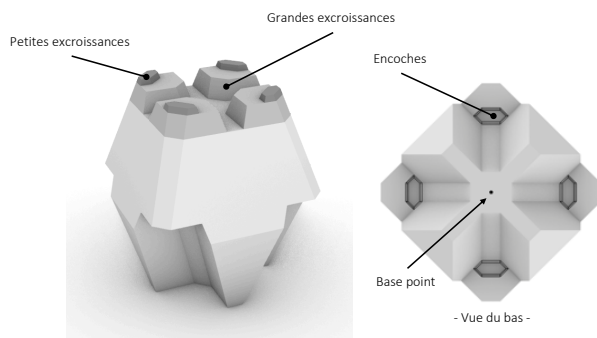


Figure 2 : Description du droxel

Les petites excroissances s'insèrent parfaitement dans les encoches pour permettre d'accrocher d'autres droxels en porte-à-faux. Le facteur F est la proportion de surface que prennent les petites sur les grandes excroissances (*Figure 2*). Le Base point est le point centré à la base du droxel. On utilisera ce point comme « carte d'identité » du droxel dans le code Grasshopper.

Une des manières de le construire est de déterminer les positions de tous ses points de contrôle dans un système de coordonnées cartésiennes où son Base point est utilisé comme repère orienté dans le

sens comme indiqué à la *Figure 3*. Les coordonnées de tous les points (*Figure 3*) sont fonction des huit paramètres géométriques et forment le quart du droxel. Le reste du bloc est développé par symétrie.

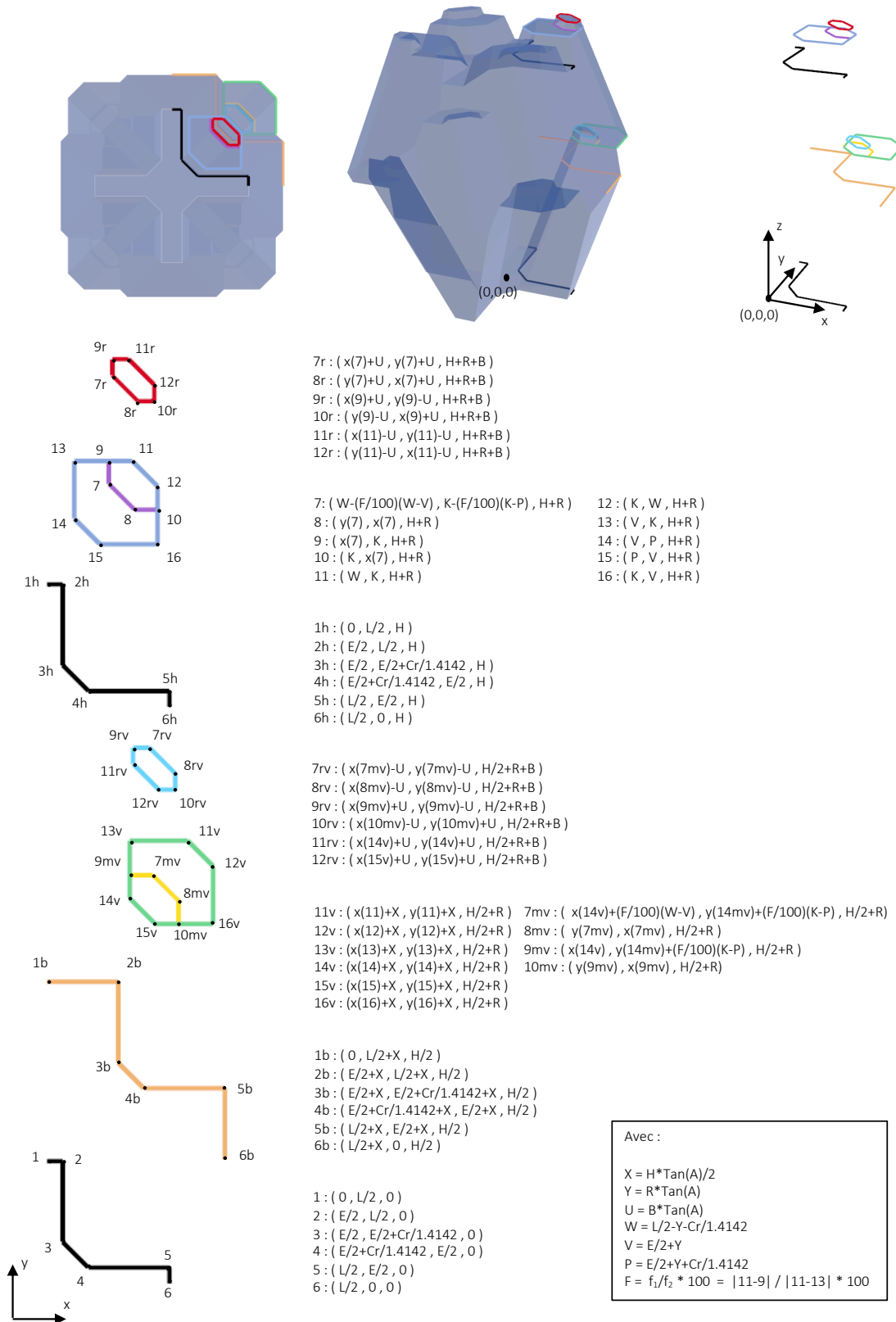


Figure 3 : Géométrie du droxel dans un système de coordonnées cartésiennes

Le droxel est un élément structurel empilable. Il peut s'empiler sur d'autres droxels selon 11 modes différents. Ces modes sont classés en trois catégories et une lettre est accordée pour chaque cas différent de la catégorie. Le [mode 1](#) est un droxel simplement déposé au-dessus d'un autre. Le [mode 2](#) est un droxel accroché en porte-à-faux sur un ou plusieurs droxels. Enfin, le [mode 3](#) est une combinaison des deux premiers. Il faut également remarquer que lorsqu'on travaille dans un repère orienté, un mode peut se diviser en plusieurs situations. Par exemple dans le [mode 2.a](#) le droxel inférieur peut se placer à quatre endroits différents (*Figure 4*).

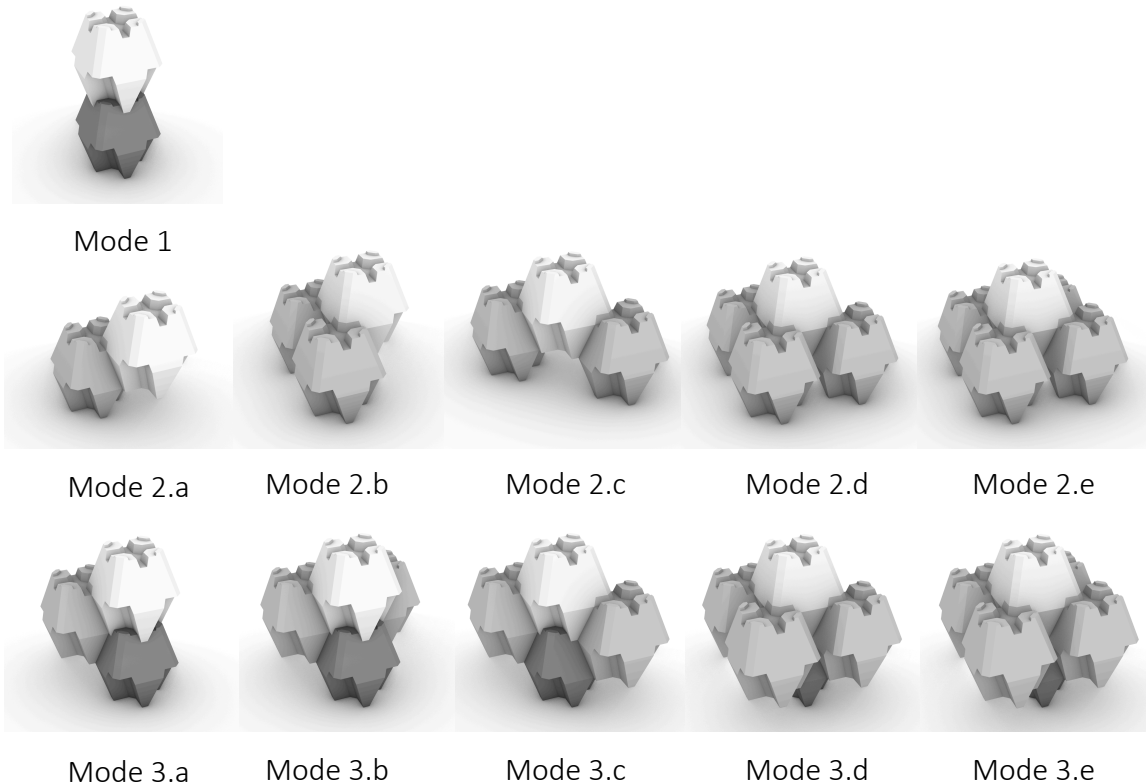


Figure 4 : Modes de pose d'un droxel

3. Principes utilisés pour l'algorithme

Avant d'en venir aux spécifications et au fonctionnement du programme Grasshopper, il est nécessaire d'expliquer sur quels principes et hypothèses il est élaboré.

Les constructions géométriques que le code Grasshopper développe, que ce soient des courbes, des volumes ou même des plans, sont orientés initialement selon le repère général du fichier Rhino. Les droxels auront toujours la même orientation : les directions – AA – sont orientées selon les axes x et y (*Figure 5*).

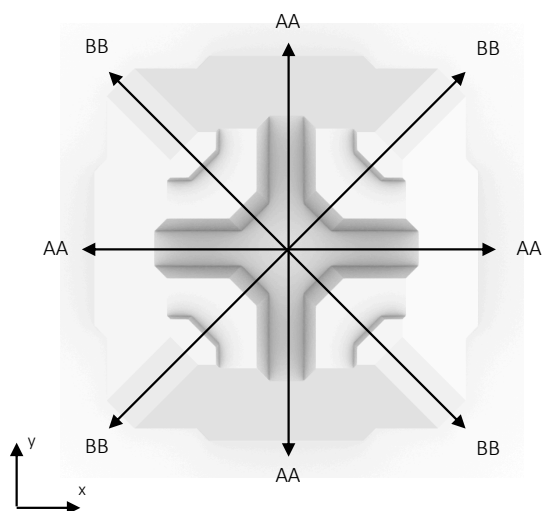


Figure 5 : Directions principales d'un droxel (vue du dessus)

L'orientation des droxels sera toujours la même et ne variera pas en fonction de la forme de la structure à droxeler. L'utilisateur peut construire n'importe quelle géométrie en trois dimensions mais il doit faire le choix de l'orientation par rapport au repère général de Rhino tout en connaissant l'orientation des droxels.

Un droxel peut être en contact avec dix autres droxels. On appelle « voisins inférieurs » ceux qui lui sont inférieurs et « voisins supérieurs » ceux qui lui sont supérieurs. Les voisins de chaque catégorie sont numérotés de 1 à 5 (Figure 6).

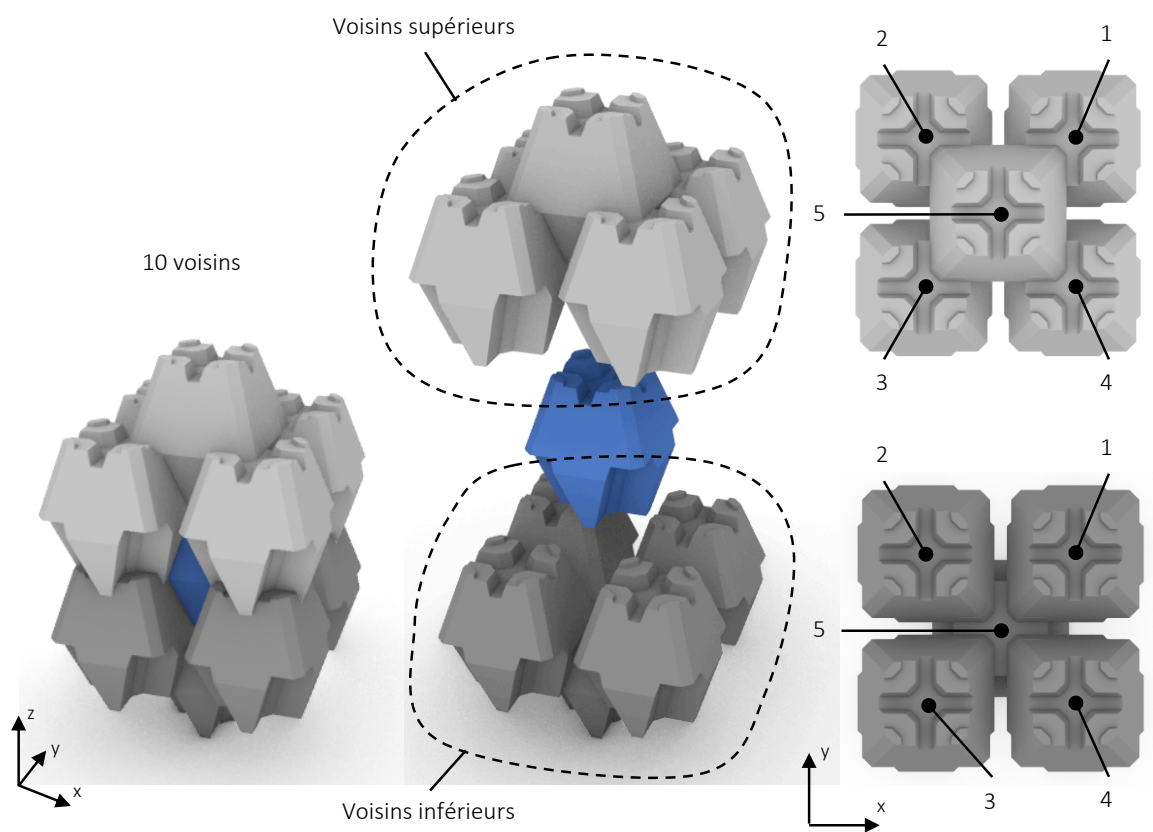


Figure 6 : Voisins d'un droxel

L'étude de la stabilité des constructions en droxels n'est pas évidente car ce ne sont pas des types de structures que nous avons l'habitude d'étudier dans le génie civil. Il n'est pas possible d'utiliser la théorie des poutres ou encore complètement apparenter les droxels à la maçonnerie traditionnelle par exemple. Ce sont des éléments structuraux qui s'empilent et qui s'accrochent entre eux. Une structure en droxels se fragmente en sous-ensembles. Un sous-ensemble se comporte comme une entité individuelle dans la structure. Il peut se désolidariser du reste de la structure s'il n'est pas en équilibre. Une manière de l'expliquer plus simplement, est de s'imaginer que lorsqu'on soulève un droxel d'une

structure à la verticale, la sous-structure sera constituée de tous les droxels qui se décrochent de celle-ci. La constitution des sous-ensembles n'est pas fixe, c'est à dire qu'un droxel peut appartenir à plusieurs sous-structures et une sous-structure peut en inclure d'autres. Mais un sous-ensemble n'est pas systématiquement une sous-structure. Une sous-structure pourrait être définie comme étant un groupe composé d'un ou plusieurs droxels dont les éléments les plus bas ne sont pas accrochés.

Toute l'analyse réalisée par le programme Grasshopper se base sur ces « sous-structures » ou « Substructures »³. Un algorithme sur les sous-structures est proposé dans la revue scientifique « Feasibility study for drone-based masonry construction of real-scale structures » [3].

Dans ce travail, l'analyse de la stabilité des structures ne se concentre que sur la stabilité globale et non locale. On ne parlera donc pas de contrainte ni de déformation mais bien de forces et d'équilibre. Le programme se focalise sur l'équilibre des structures en droxels lors de leur mise en œuvre. C'est à dire qu'on suppose que les structures ne sont soumises qu'aux charges dues au poids propre des blocs. On néglige toute charge extérieure au système comme les charges d'exploitation, de vent, les charges sismiques, etc...

On désire profiter du gros avantage de ce type d'éléments structurels, le fait qu'ils soient autoportants. Lors de la mise en œuvre il faut donc qu'ils soient à l'équilibre à chaque étape. Par chaque étape on sous-entend à chaque ajout d'un nouveau droxel.

Les blocs s'empilent du haut vers le bas les uns sur les autres. Il ne peut donc y avoir de droxel « flottant », sans voisins inférieurs. On émet l'hypothèse que tous les droxels appartenant à la même couche sont posés au même moment. On peut voir à la *Figure 7* comment sont définies les couches.

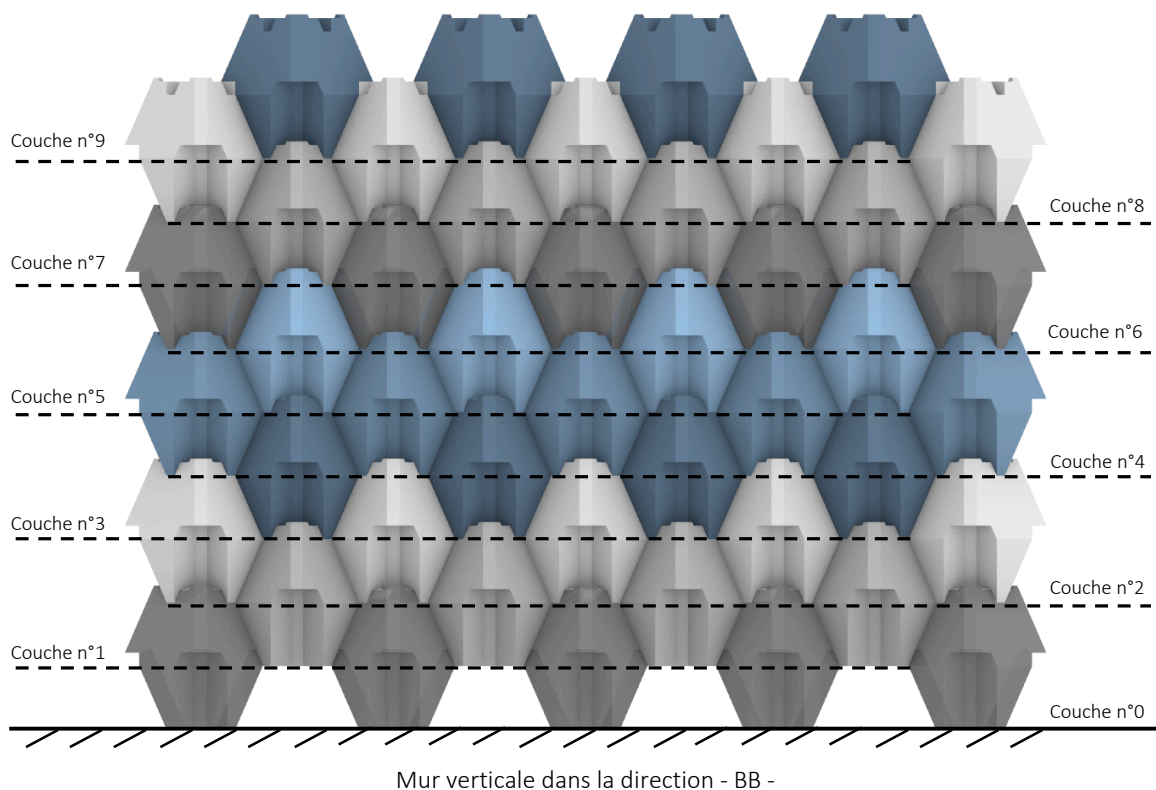


Figure 7 : Couche de droxels

³ On utilise la traduction anglaise dans le code Grasshopper.

L'algorithme doit donc, en même temps que droxéliser, analyser la stabilité globale. Dès qu'on ajoute des droxels pour réaliser la structure désirée, on se retrouve avec une nouvelle structure différente de la précédente. Lorsqu'on pose une couche additionnelle sur les précédentes, ce sont de nouvelles sous-structures qui apparaissent, de nouvelles résultantes de forces.

Les forces qui interviennent dans un système de droxels sont leur poids propre. Donc, lorsqu'il s'agit de calculer la position du centre de gravité d'une sous-structure, ce n'est rien d'autre que le point d'application de la résultante des forces de gravité de tous les droxels qui la composent.

Le code Grasshopper utilise la même géométrie pour tous les droxels et on émet l'hypothèse qu'ils possèdent la même masse, étant donné que la masse n'est pas paramétrable dans le programme développé. Chaque élément subit une force verticale de gravité identique. Les coordonnées du point d'application de la résultante d'une sous-structure sont les coordonnées pondérées des centres de gravité de chaque droxel.

La zone d'appui d'une sous-structure est la surface de contact qu'elle a avec tous les éléments externes, c'est-à-dire le sol ou d'autres droxels. On suppose que les droxels de la première couche, en contact avec le sol, sont encastrés⁴.

Enfin, il existe deux catégories pour décrire la relation qu'un droxel a avec ses voisins inférieurs. On dit qu'un droxel est « hooked » (accroché) s'il ne bascule pas lorsqu'il subit une force dont la projection verticale du point d'application est en dehors de la zone d'appui du droxel. A l'inverse, si le point d'application est en dehors et que le droxel se renverse, le droxel est « not hooked » (pas accroché).⁵ La « relation » est en soit le mode de pose défini précédemment (Figure 4).

On classe les modes de pose dans les deux catégories de la façon suivante : les [mode 2, 3.c, 3.d et 3.e](#) font partie de la catégorie « hooked »⁶ et les droxels de [mode 1, 3.a et 3.b](#) sont « not hooked ». Les modes « hooked » sont considérés comme tels car une trop grande translation verticale serait nécessaire pour renverser le droxel.

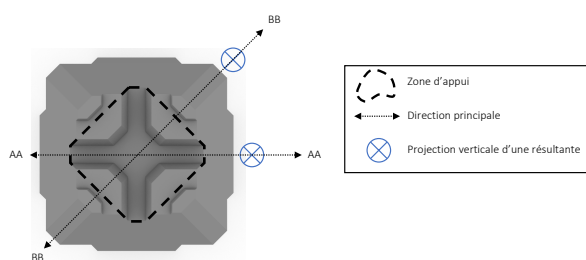


Figure 8 : Exemple de projection verticale de résultante en dehors de la zone d'appui du mode 1

L'instabilité du [mode 1](#) est soit une rotation soit la combinaison d'une translation verticale et d'une rotation (Figure 9). Lorsque la projection verticale de la résultante des forces croise la direction – AA – en dehors de la zone d'appui le droxel bascule (Figure 8). Pour toutes les autres positions (hors zone d'appui) le droxel va traduire verticalement afin de s'extraire des excroissances du droxel inférieur tout en effectuant une rotation.

⁴ Cette condition peut rapidement être changée dans le code Grasshopper.

⁵ Encore une fois la terminologie anglaise est préférée pour la programmation.

⁶ Dans la section « Pistes d'amélioration » les modes 2.a et 2.b peuvent être « not hooked ».

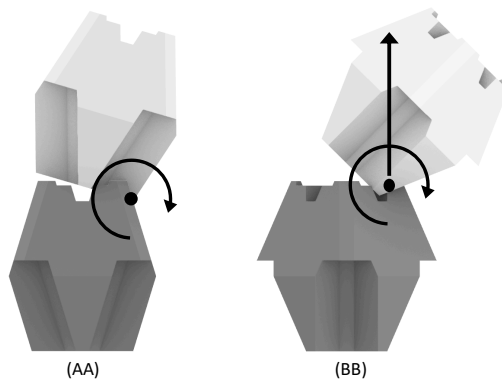


Figure 9 : Mouvement d'instabilité d'un droxel de mode 1 dans les directions -AA- et -BB-

Un droxel de [mode 3.a](#) ou [3.b](#) présente également une instabilité quand la projection verticale de la résultante du poids propre de la sous-structure à laquelle il appartient est hors de sa zone d'appui (Figure 10).

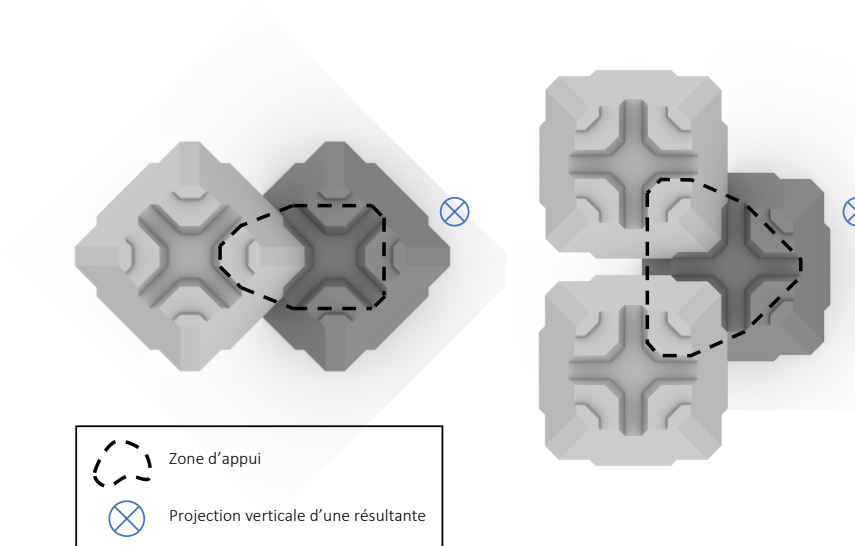


Figure 10 : Exemple de projection verticale de résultante en dehors de la zone d'appui du mode 3.a et 3.b

Le mouvement du droxel est régi par un cycle commençant par une translation verticale puis une rotation toutes deux infinitésimales et qui se répète jusqu'au basculement du droxel. Il se crée une surface de glissement entre l'encoche du droxel supérieur et la petite excroissance de son droxel inférieur (Voir à la Figure 11 pour le [mode 3.a](#)).

On considère ces modes instables contrairement au [mode 2.a](#) et [2.b](#) car la rotation tend à amplifier le soulèvement tandis que le droxel est bloqué vers le bas.

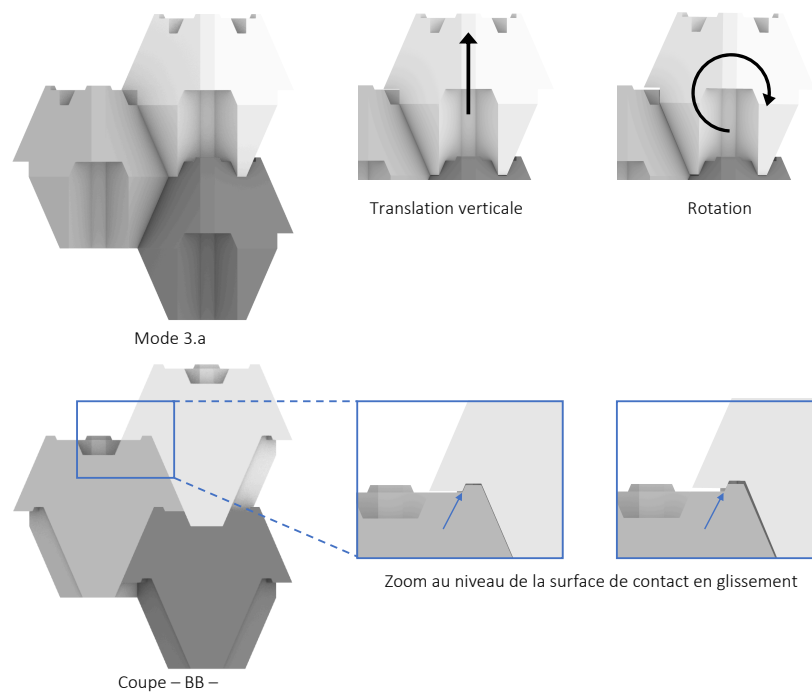


Figure 11 : Mouvement de l'instabilité du mode 3.a

3.1. Processus de détection des sous-structures

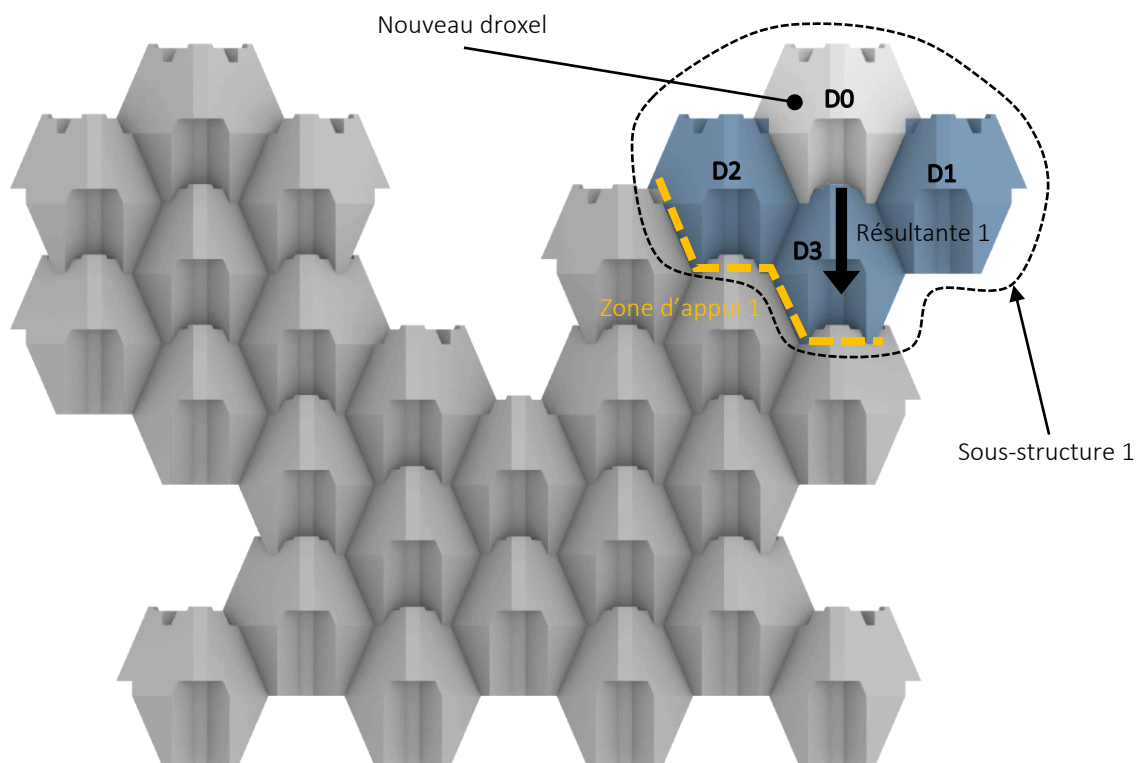
La marche à suivre générale est établie pour détecter les sous-structures instables créées suite à l'ajout d'un nouveau droxel dans une structure droxelisée déjà existante [3].

Processus de détection des sous-structures instables :

1. Sélectionner le nouveau droxel.
2. Ajouter les voisins inférieurs du nouveau droxel.
3. Ajouter les voisins supérieurs des droxels du groupe et leurs propres voisins supérieurs et ainsi de suite jusqu'à la nouvelle couche.
4. Déterminer le droxel situé au plus bas du groupe.
5. Analyser le mode de pose du droxel le plus bas du groupe.
6. Si c'est le mode 1, 3a ou 3b, il s'agit d'une sous-structure sinon ce n'est pas encore une sous-structure.
 - a. Si le groupe est une sous-structure,
 - i. Calculer le point d'application du poids résultant des droxels de la sous-structure (son centre de gravité).
 - ii. Déterminer la zone d'appui de la sous-structure.
 - iii. Tester si la projection du centre de gravité est incluse dans la zone d'appui.
 - iii-1. Si la projection du centre de gravité n'est pas incluse dans la zone d'appui, la sous-structure est instable. Le nouveau droxel crée une instabilité.
 - iii-2. Si la projection du centre de gravité est incluse dans la zone d'appui, continuer en recommençant le processus à l'étape 1 avec le droxel situé au plus bas de la sous-structure et non plus le nouveau droxel.
 - b. Si le groupe n'est pas une sous-structure, continuer en recommençant le processus à l'étape 1 avec le droxel situé au plus bas du groupe et non plus le nouveau droxel.

Appliquons ce processus sur un exemple avec une structure simple (*Figure 12*).

1. Sélectionner le nouveau droxel D0.
2. Ajouter les droxels D1, D2 et D3.
3. –
4. Déterminer le droxel situé au plus bas du groupe (D3).
5. Analyser le mode de pose du droxel D3.
6. C'est le [mode 3.a](#).
 - a. Le groupe est une sous-structure (sous-structure 1),
 - i. Calculer la résultante 1.
 - ii. Déterminer la zone d'appui de la sous-structure (zone d'appui 1).
 - iii. Tester si la projection du centre de gravité est incluse dans la zone d'appui.
 - iii-1. –
 - iii-2. La projection de la résultante 1 est incluse dans la zone d'appui 1, continuer en recommençant le processus à l'étape 1 avec le droxel D3.
 - b. –

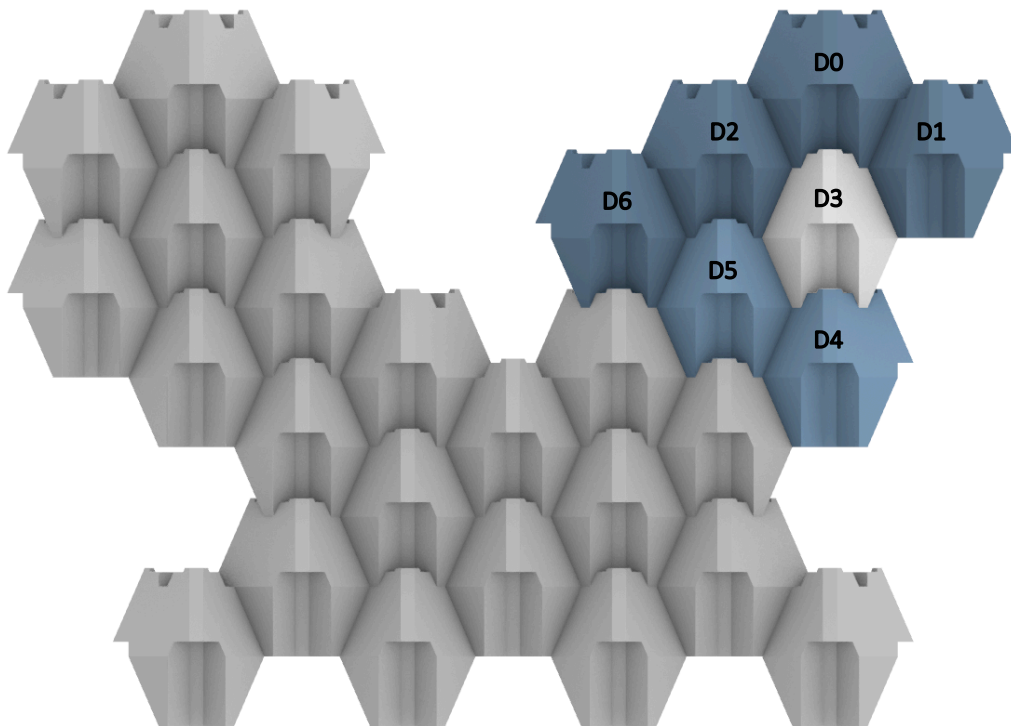


Structure dans le plan vertical de direction - BB -

Figure 12 : Application du processus de détection (1)

On recommence le processus avec le droxel D3 (*Figure 13*) :

1. Sélectionner le nouveau droxel D3.
2. Ajouter les droxels D4 et D5.
3. Ajouter les droxels D6, D1, D2 et D0.
4. Déterminer le droxel situé au plus bas du groupe (D4).
5. Analyser le mode de pose du droxel D4.
6. Si c'est le [mode 2.a.](#)
 - a. -
 - i. -
 - ii. -
 - iii. -
 - iii-1. -
 - iii-2. -
 - b. Le groupe n'est pas une sous-structure, continuer en recommençant le processus à l'étape 1 avec le droxel D4.



Structure dans le plan vertical de direction - BB -

Figure 13 : Application du processus de détection (2)

On recommence le processus avec le droxel D4 (Figure 14) :

1. Sélectionner le nouveau droxel D4.
2. Ajouter le droxel D7.
3. Ajouter les droxels D8, D5, D3, D6, D1, D2 et D0.
4. Déterminer le droxel situé au plus bas du groupe (D7).
5. Analyser le mode de pose du droxel D7.
6. C'est le [mode 3.a.](#)
 - a. Le groupe est une sous-structure (sous-structure 2),
 - i. Calculer la résultante 2.
 - ii. Déterminer la zone d'appui de la sous-structure (zone d'appui 2).
 - iii. Tester si la projection du centre de gravité est incluse dans la zone d'appui.
 - iii-1. La projection de la résultante 2 n'est pas incluse dans la zone d'appui 2, la sous-structure 2 est instable. Le nouveau droxel D0 crée une instabilité.
 - iii-2. -
 - b. -

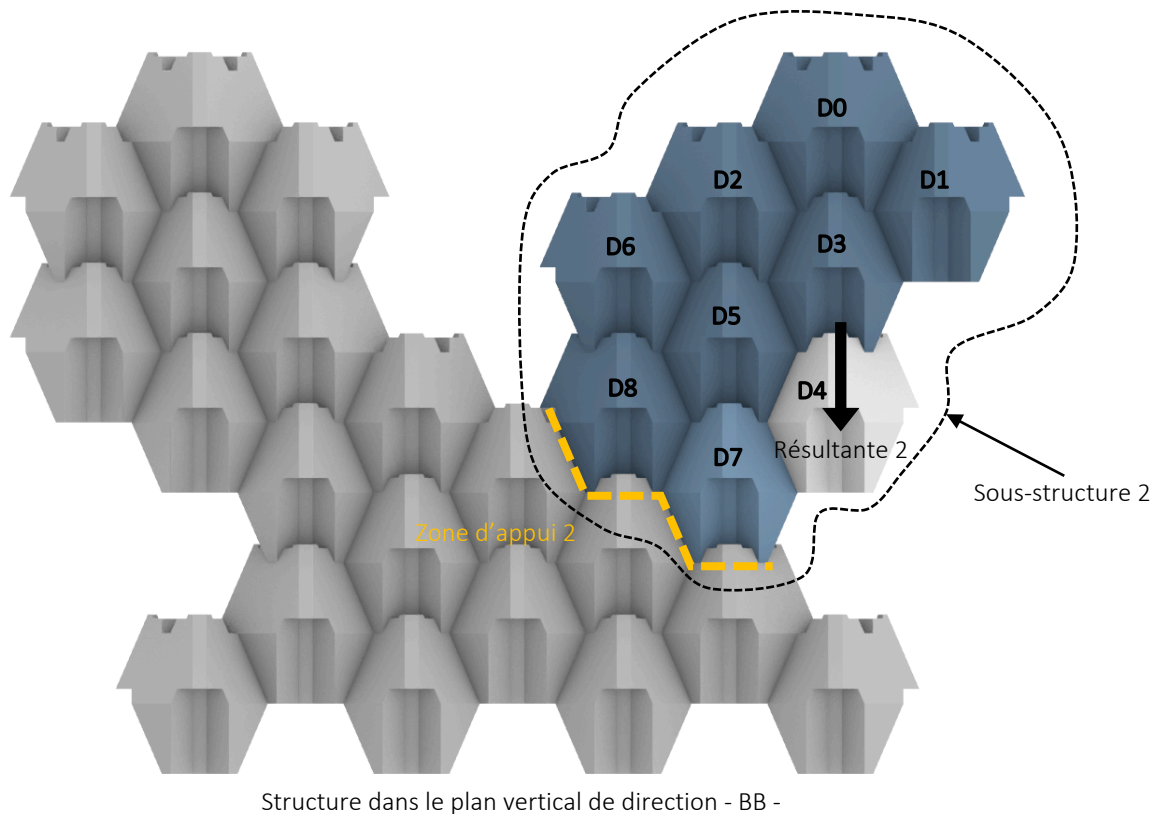


Figure 14 : Application du processus de détection (3)

Le processus de détection de sous-structures instables est adapté dans l'algorithme Grasshopper pour des structures plus complexes en trois dimensions et non pas dans un seul plan comme dans l'exemple précédent. Il y a souvent plusieurs droxels « situés au plus bas » dans une sous-structure, plusieurs sous-structures générées par un droxel, etc.



4. Code Grasshopper

Le code Grasshopper transforme une géométrie réalisée dans Rhino en une structure composée de droxels.

Le canevas principal (*Figure 15*) est l'interface principale dans laquelle l'utilisateur peut définir les paramètres d'entrée qui sont sous forme de curseurs. Tous ce qui est paramétrable par l'utilisateur est regroupé dans les groupes verts du canevas principal. Dès que l'on rentre dans un cluster, on change de canevas. Un cluster est un objet Grasshopper qui rassemble d'autres objets (composant, cluster, ...). Il réalise un processus sur base de paramètres d'entrée (inputs) et renvoie les résultats de sortie (outputs).

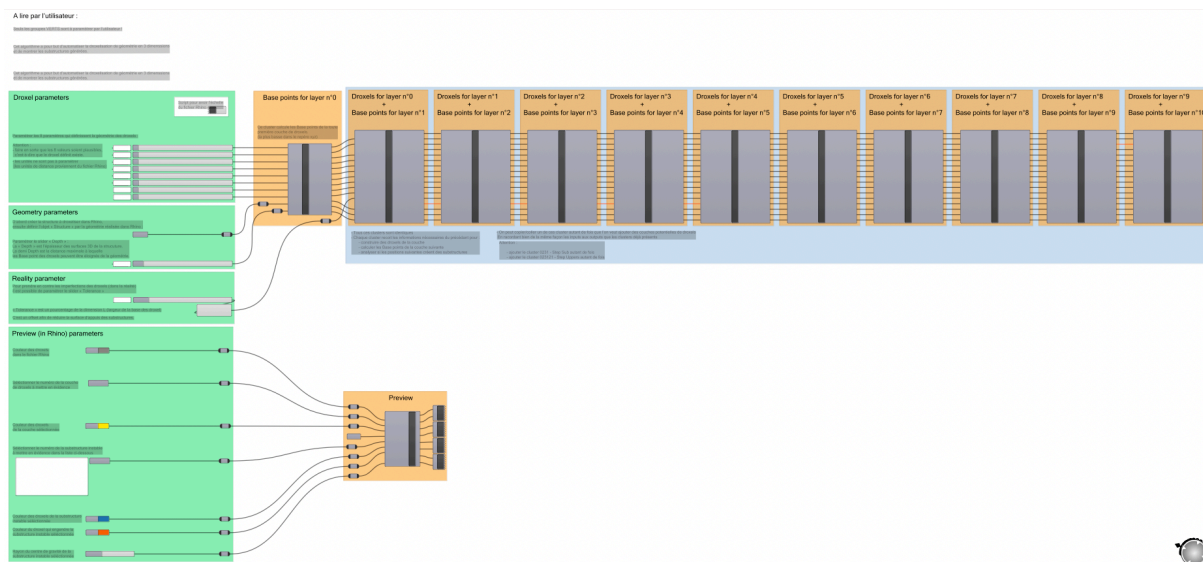


Figure 15 : Canevas principal

L'utilisateur entre dans un premier temps les huit paramètres des droxels qu'il souhaite. Les facteurs de distance ont les mêmes unités que l'échelle du fichier Rhino, souvent en millimètres ou en mètres. L'échelle (scale) est détectée par le code Grasshopper à l'aide d'un petit script Python.

Ensuite il entre la géométrie de la structure du fichier Rhino sous forme de *Brep* qui est un type d'objet Grasshopper signifiant « Boundary representation ».

Au lieu d'entrer une structure avec une épaisseur, on externalise un paramètre *Depth*. En effet, si on veut construire un mur plan en droxel, son épaisseur ne sera pas régulière à cause de la géométrie particulière des blocs. Il faudrait définir un paramètre pour savoir de combien les droxels peuvent déborder de la géométrie initiale de la structure. On simplifie avec la *Depth* : qui est le double de la distance à laquelle les Base points des droxels peuvent se déporter de la structure.

Un dernier paramètre optionnel est un facteur de tolérance qui concerne l'analyse de la stabilité globale. Il réduit la surface d'appui des sous-structures pour tenir compte d'une marge d'erreur lorsque le résultat est comparé à la réalité.

Une fois le code compilé, l'utilisateur obtient une visualisation 3D des résultats de la droxelisation dans Rhino. L'utilisateur peut mettre en évidence une couche de droxels mais le plus important est qu'il peut également voir les sous-structures instables si le programme en a détectées lors de la droxelisation.

Toute la complexité de la droxelisation avec Grasshopper est de gérer correctement des « arbres de données ». Associer un droxel à un nom ou à un numéro est impossible. Un droxel est représenté par l'indice de son Base point dans une liste et plus globalement dans une arborescence de données.

« Grasshopper n'utilise, contrairement à un environnement de programmation, aucun nom d'objet pour définir un objet. Cela peut sembler trivial, mais c'est l'une des différences les plus fondamentales par rapport à un environnement de modélisation traditionnelle. Dans Grasshopper, l'objet ou les objets sont placés dans une liste. Les différentes listes de données sont organisées dans une arborescence de données où chaque branche et contenu de données de la branche ont un numéro d'index. L'accès à l'objet est donc plus problématique que dans un environnement de script. Grasshopper dispose de divers outils pour remédier à ce problème. Ces outils prennent en charge l'édition et la sélection du contenu des listes et l'édition de l'arborescence des données. La connaissance de ces techniques est essentielle pour une utilisation efficace de Grasshopper. » [6]

Plusieurs outils sont disponibles pour gérer correctement les listes. Les plus utilisés sont les options « Flatten Tree » et « Graft Tree ». L'outil *Flatten* est utilisé pour restructurer radicalement un arbre complexe avec des sous-listes en une seule liste contenant tous leurs composants. A l'inverse, avec l'outil *Graft*, une liste est complexifiée en une arborescence de sous-listes.

Dans Grasshopper nous utilisons des composants (Figure 16) qui permettent de supprimer les doublons de points géométriques qui seraient répétés plusieurs fois afin de minimiser la taille des listes avec lesquelles nous travaillons.

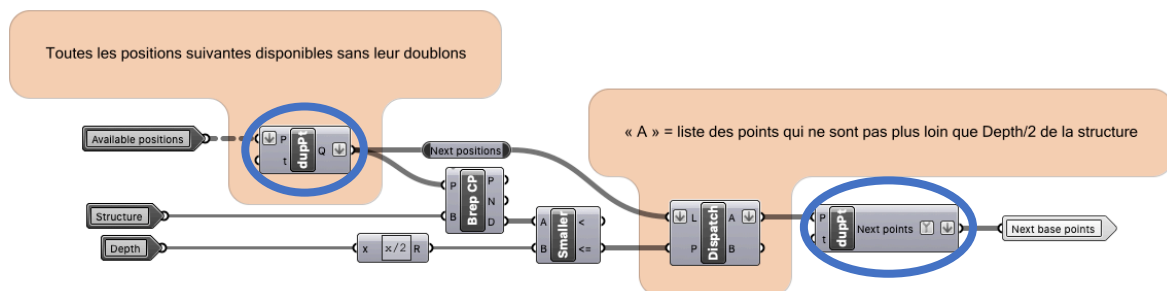


Figure 16 : Composant Grasshopper pour supprimer les doublons

Un composant « Group », important également, peut être utilisé. Il crée un groupe de constructions géométriques et est employé pour regrouper les Base points d'un même ensemble afin de travailler avec l'ensemble et non pas avec chaque élément individuel.

Flow Charts

Dans le but d'expliquer le plus limpide possible les décisions et les différents processus de l'algorithme, des flow charts⁷ (Figure 17) sont schématisés pour chaque cluster. Les flow charts sont des représentations simplifiées du langage Grasshopper pour comprendre les imbrications de tous les clusters de l'algorithme. Un flow chart permet de comprendre les liens entre les clusters, les processus, les prises de décisions et les données en entrée (Inputs) et données produites (Outputs). Dans les flow charts des clusters, nous faisons le choix de ne pas montrer tous les composants présents dans le code Grasshopper mais bien les éléments et les liens principaux qui sont essentiels à la compréhension des clusters. Lorsqu'un « Process » est encadré en pointillé celui-ci est schématisé en dessous du flow chart.

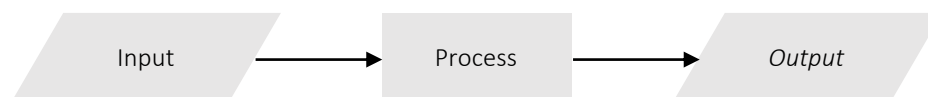


Figure 17 : Légende d'un flow chart

4.1. Clusters

Un cluster est en réalité une portion de code qui est rassemblé dans un objet Grasshopper. Un nom leur est attribué pour permettre à l'utilisateur de bien s'y retrouver dans le code.

Exemple : **021 – Droxel**, signifie que le cluster se nomme Droxel et le nombre signifie que c'est le premier cluster (**1**) appartenant au deuxième cluster (**2**) du canevas principal (**0**).

On présente tous les clusters classés de façon hiérarchique dans la table suivante :

- 01 – Base
 - 011 – Grid
- 02 – Layer (x10)
 - 021 – Droxel
 - 022 – Position
 - 023 – Stability Check
 - 0231 – Step Sub (x10)
 - 02311 – Lower
 - 02312 – Uppers
 - 023121 – Step Uppers (x10)
 - 02313 – Substructure
 - 02314 – Gravity Center
 - 02315 – Gravity Test
 - 02316 – Lowest Statement
 - 02317 – Sub Statement
 - 0232 – Sub Edit

- 03 – Preview

Le cluster **023 – Stability Check** organise ses clusters selon le même schéma des étapes du **processus de recherche des sous-structures instables**. Le parcours des données dans le code Grasshopper est

⁷ Il est recommandé au lecteur de suivre les flow charts en parallèle des explications.

représenté à l'aide du flow chart général du programme qui montre également la corrélation entre tous les clusters à la *Figure 18*.

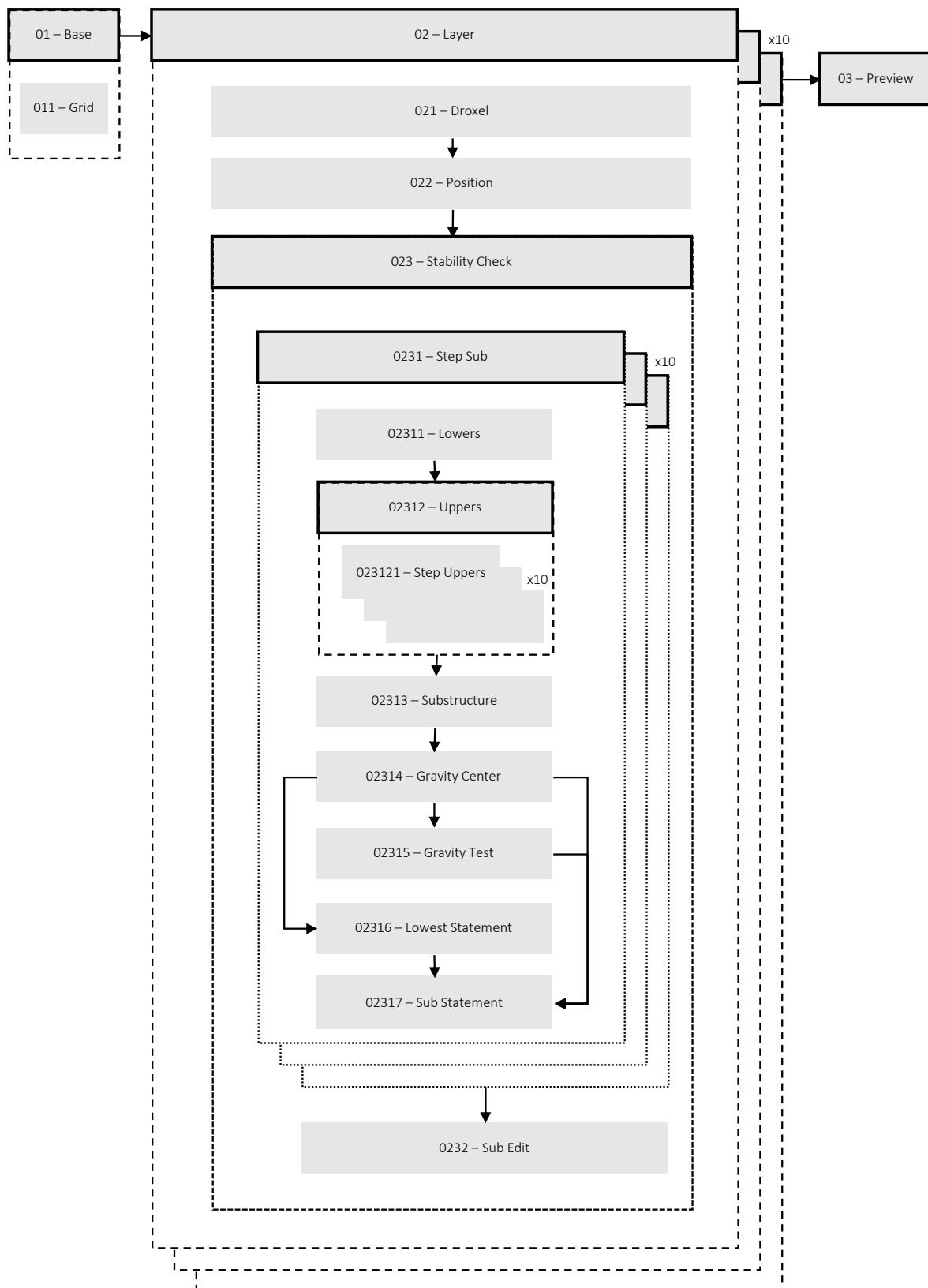
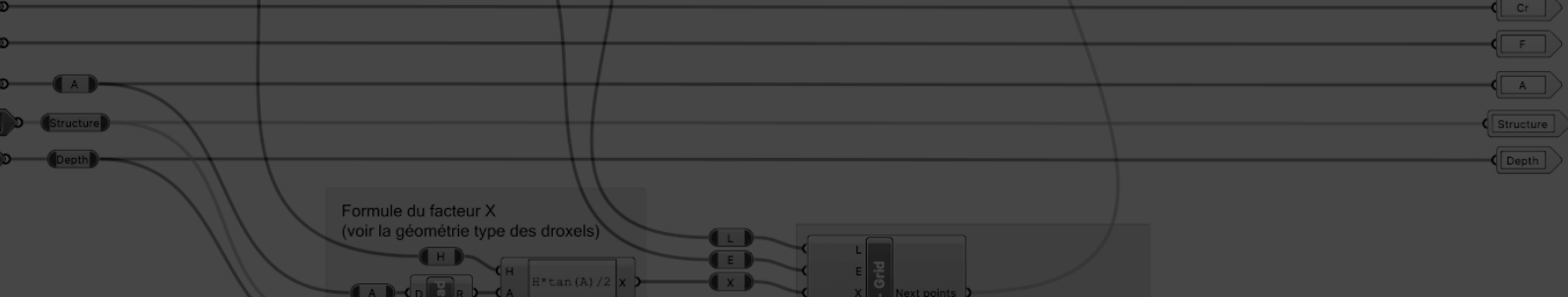


Figure 18 : Corrélation des clusters



4.1.1. 01 – Base

Le **cluster 01 – Base** ne contient qu’un seul autre cluster (*Figure 19*). Il faut mentionner que la plupart des paramètres d’entrée (*Tableau 1* et *Tableau 2*) ne sont pas utilisés car ce cluster est présent pour éclaircir le canevas principal.

Inputs	[Unité] / type d’objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
R	[Scale]	Paramètre géométrique du droxel.
B	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
F	[%]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.
Structure	Brep	Géométrie de la structure à droxeler.
Depth	[Scale]	Épaisseur de la structure à droxeler.

Tableau 1 : Inputs du cluster 01 - Base

Outputs	[Unité] / type d’objet Gh	Description
Layer of point	Group of points	Groupe contenant les Base points de la première couche.
Base points	List of points	Liste contenant les Base points de la première couche.
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
R	[Scale]	Paramètre géométrique du droxel.
B	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
F	[%]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.
Structure	Brep	Géométrie de la structure à droxeler.
Depth	[Scale]	Épaisseur de la structure à droxeler.

Tableau 2 : Outputs du cluster 01 - Base

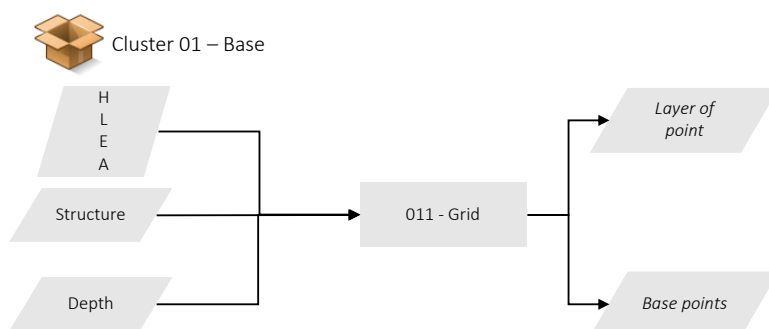
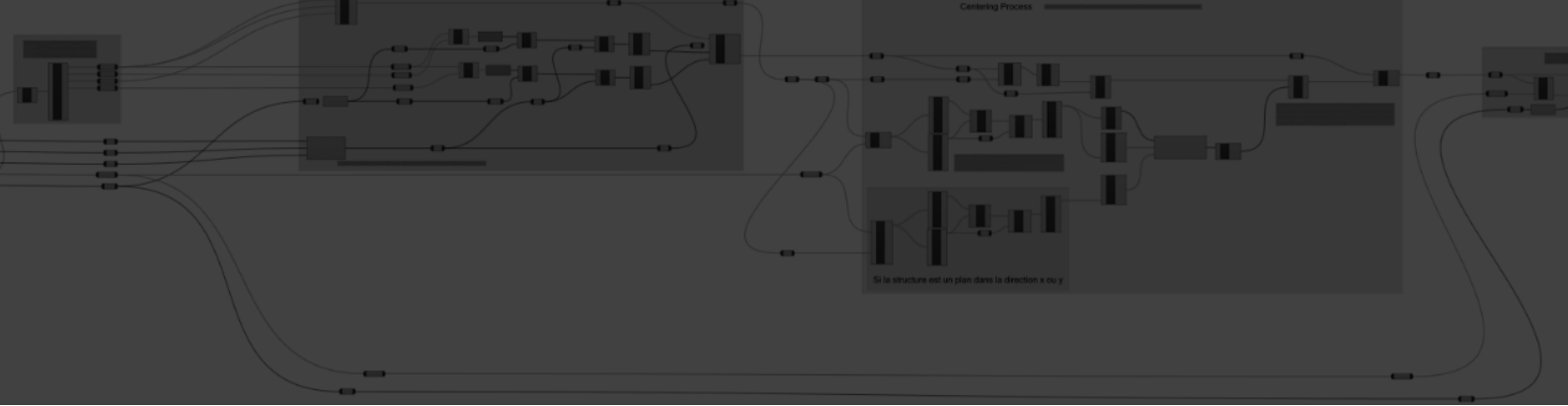


Figure 19 : Flow chart du cluster 01 – Base



4.1.2. 011 - Grid

Le **cluster 011 – Grid** crée la grille de la première couche (n°0) de droxels. On calcule la position des premiers Base points de la structure. Les droxels qui se trouvent à la même hauteur ont comme positions possibles les points d'un maillage constitué de cellules carrées de côté égal à $L+E+2X$ (Figure 20).

Il faut situer la structure dans l'espace pour ne pas créer une grille de dimension infinie et réaliser des calculs inutiles pouvant être évités. En construisant la plus petite boîte rectangulaire qui inclut la structure on peut en extraire les coordonnées des coins de celle-ci et dès lors créer une grille de taille appropriée.

La taille des cellules étant déjà imposée, il ne manque que le nombre de cellules dans les directions x et y. On prend le premier nombre entier plus grand que deux fois la dimension de la boîte selon respectivement x et y additionné au double de la *Depth*, le tout divisé par la largeur des cellules (Figure 20).

Cette valeur est surévaluée car il faut s'assurer de proposer assez de positions étant donné que la grille est ensuite déplacée pour être centrée par rapport au centre de la courbe créée par l'intersection entre le bas de la boîte et la structure.

Cette étape est effectuée par le processus « Centering Grid » (Figure 20). Cette manœuvre est d'autant plus utile lorsque la structure à droxéliser est d'une géométrie régulière (cube, sphère, etc.), afin de disposer les droxels de manière équilibrée.

La dernière étape est de ne garder que les positions situées à une certaine distance maximale de la structure. L'utilisateur a paramétré cette distance par le biais du facteur *Depth*. *Depth* étant l'épaisseur de la structure, la distance maximale est la moitié de *Depth*.

Inputs	[Unité] / type d'objet Gh	Description
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$\frac{H}{2} * \tan(A)$
Structure	Brep	Géométrie de la structure à droxéliser.
Depth	[Scale]	Épaisseur de la structure à droxéliser.

Tableau 3 : Inputs du cluster 011 – Grid

Outputs	[Unité] / type d'objet Gh	Description
Next points	List of points	Liste qui contient les Base points de la première couche.

Tableau 4 : Outputs du cluster 011 – Grid



Cluster 012 - Grid

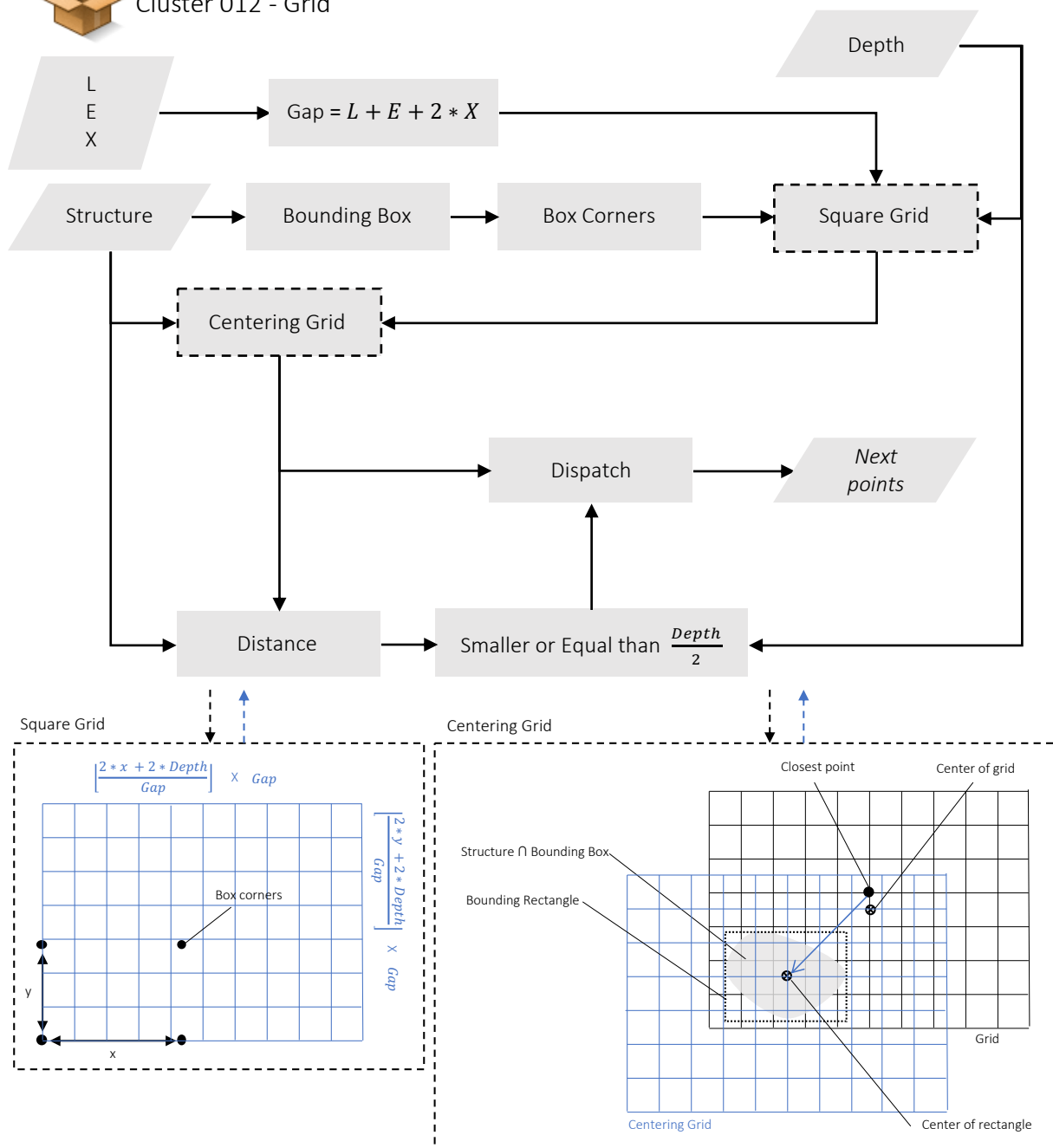
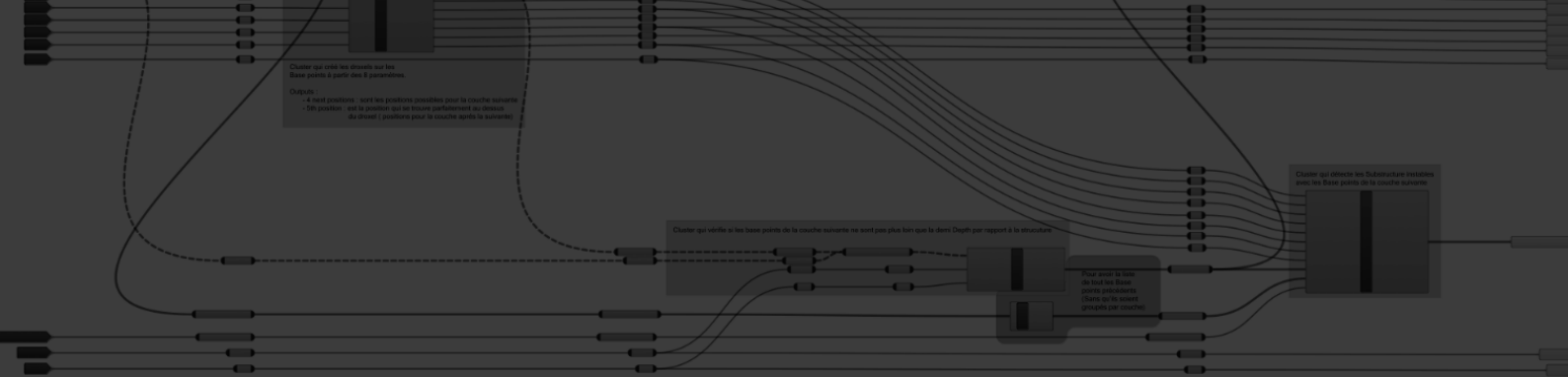


Figure 20 : Flow chart du cluster 011 – Grid

« Dispatch » est un composant Grasshopper qui permet de dispatcher une liste dans deux listes cibles selon un pattern composé de 0 et de 1 ou de valeurs booléennes. Dans le cas présent on teste si une valeur (distance entre les points et la Structure) est plus petite ou égale à une autre ($Depth/2$).

« Bounding Box » est un composant Grasshopper qui construit la plus petite boîte selon les directions principales du repère de Rhino contenant une géométrie en trois dimensions.



4.1.3. 02 – Layer

Le cluster **02 – Layer** construit les couches de droxels. Son premier objectif est de former les droxels sur les Base points grâce au cluster **021 – Droxel**. Le second est de préparer les Base points pour la couche suivante. Le cluster **02 - Layer** obtient les prochains Base points avec l'aide du cluster **022 – Position** et analyse leur influence sur la stabilité avec le cluster **023 – Stability Check** (Figure 22).

Grasshopper ne permet pas de compiler un cluster en boucle. Ses outputs ne peuvent être en même temps ses inputs. On répète alors le cluster **02 – Layer** un nombre prédéfini de fois. Le nombre maximum de couches de droxels qui pourront être construites. Dix clusters sont présents dans l'algorithme ce qui signifie que les structures ne peuvent être droxelisées avec plus de dix couches de droxels. Tous les clusters sont en série. Chacun transmet ses données au suivant et ainsi de suite. L'ensemble des données est récolté par le cluster **03 – Preview** au dernier cluster **02 – Layer** pour fournir un rendu 3D dans Rhino.

Si le code venait à être modifié pour permettre un plus grand nombre de couches, il suffirait de copier-coller le cluster **02 – Layer** le nombre de fois souhaité et de les relier de façon similaire à ceux déjà présents (Figure 21).

Deux autres clusters (**0231 – Step Sub** et **023121 – Step Uppers**) qui permettent de voyager entre les différentes couches sont également présents dix fois et devraient donc être répétés le nombre de fois nécessaire.

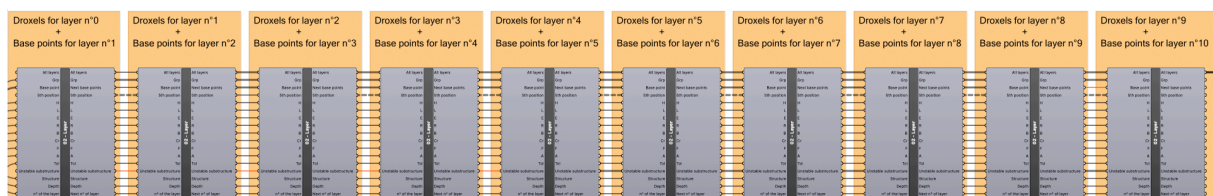


Figure 21 : Cluster 02 – Layer

Inputs	[Unité] / type d'objet Gh	Description
All layers	List of groups of Breps	Liste qui contient l'ensemble des droxels précédents regroupés par couche.
Grp	List of groups of points	Liste qui contient l'ensemble des Base points précédents regroupés par couche.
Base points	List of points	Liste qui contient les Base points de la couche.
5th position	List of points	Liste qui contient la position du voisin supérieur n°5 de chaque droxel de la couche précédente.
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
R	[Scale]	Paramètre géométrique du droxel.
B	[Scale]	Paramètre géométrique du droxel.

Cr	[Scale]	Paramètre géométrique du droxel.
F	[%]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.
Unstable substructures	List of groups of points	Liste qui contient les groupes de Base points des substructures instables précédentes.
Structure	Brep	Géométrie de la structure à droxeler.
Depth	[Scale]	Épaisseur de la structure à droxeler.

Tableau 5 : Inputs du cluster 02 - Layer

Outputs	[Unité] / type d'objet Gh	Description
All layers	List of groups of Breps	Liste qui contient l'ensemble des droxels regroupés par couche.
Grp	List of groups of points	Liste qui contient l'ensemble des Base points regroupés par couche.
Next base points	List of points	Liste qui contient les Base points pour la couche suivante.
5th position	List of points	Liste qui contient les 5 ^{ème} positions des droxels de la couche.
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
R	[Scale]	Paramètre géométrique du droxel.
B	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
F	[%]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.
Unstable substructures	List of groups of points	Liste qui contient les Base points des droxels des substructures instables regroupés par substructure.
Structure	Brep	Géométrie de la structure à droxeler.
Depth	[Scale]	Épaisseur de la structure à droxeler.

Tableau 6 : Outputs du cluster 02 - Layer

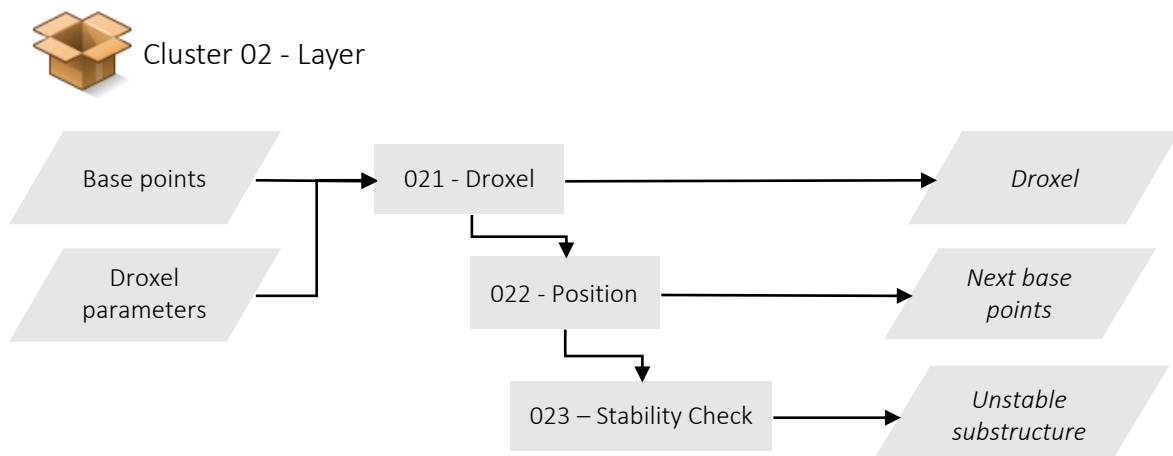
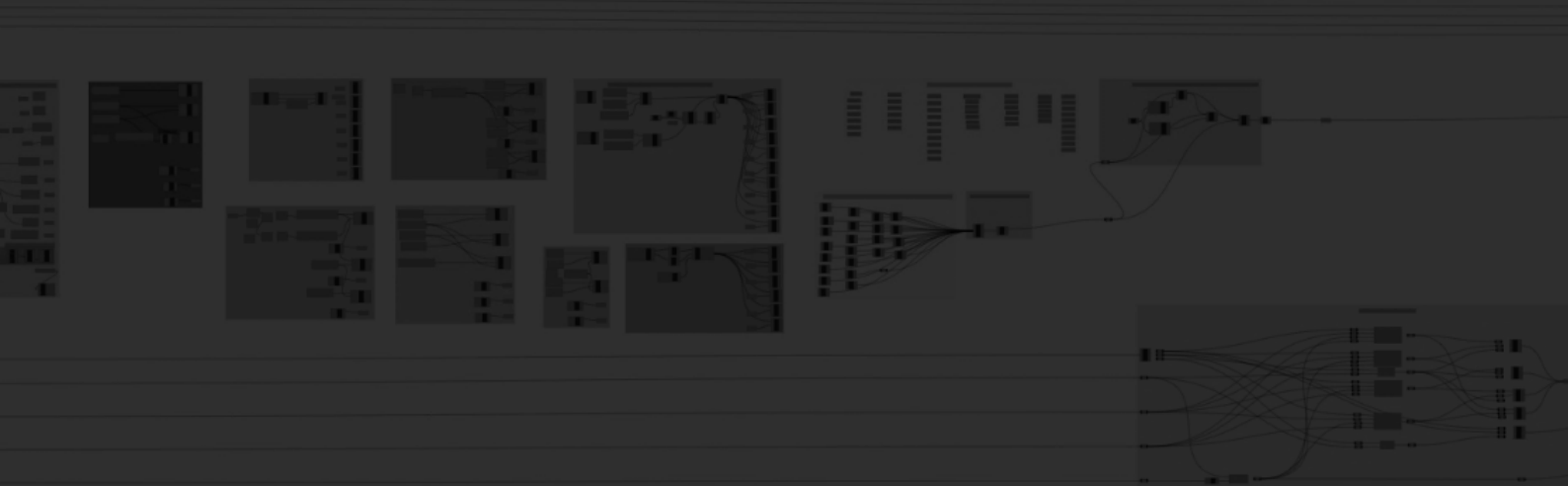


Figure 22 : Flow chart du cluster 02 - Layer



4.1.4. 021 – Droxel

Le cluster **021 – Droxel** construit un droxel à partir d'un Base point (*Figure 2*). Le cluster prend en inputs (*Tableau 7*) les huit paramètres qui définissent un droxel (*Figure 1*) et son Base point.

Inputs	[Unité] / type d'objet Gh	Description
Base point	List of points	Base points pour y construire les droxels de la couche.
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
R	[Scale]	Paramètre géométrique du droxel.
B	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
F	[%]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.

Tableau 7 : Inputs du cluster 021 – Droxel

Outputs	[Unité] / type d'objet Gh	Description
Droxel	List of Breps	Liste qui contient les droxels de la couche.
4 next positions	List of points	Positions des voisins supérieurs 1, 2, 3 et 4 de chaque droxel.
5th position	List of points	Positions du voisin supérieur 5 de chaque droxel.
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
R	[Scale]	Paramètre géométrique du droxel.
B	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
F	[%]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.

Tableau 8 : Outputs du cluster 021 - Droxel

Le processus « Construction » calcule tous les points formant le premier quart du bloc à partir des coordonnées du Base point. Les positions de ces points sont affichées à la *Figure 3*. Une fois tous les points trouvés, les contours des surfaces du volume sont créés. Ensuite, par un jeu de symétrie, le volume total est entièrement défini (*Figure 23*). Le cluster est également en charge de localiser les cinq positions où de nouveaux droxels peuvent être empilés, celles des voisins supérieurs (*Figure 6*). De cette manière on s'assure de ne pas avoir des droxels qui flottent, autrement dit sans voisin inférieur.

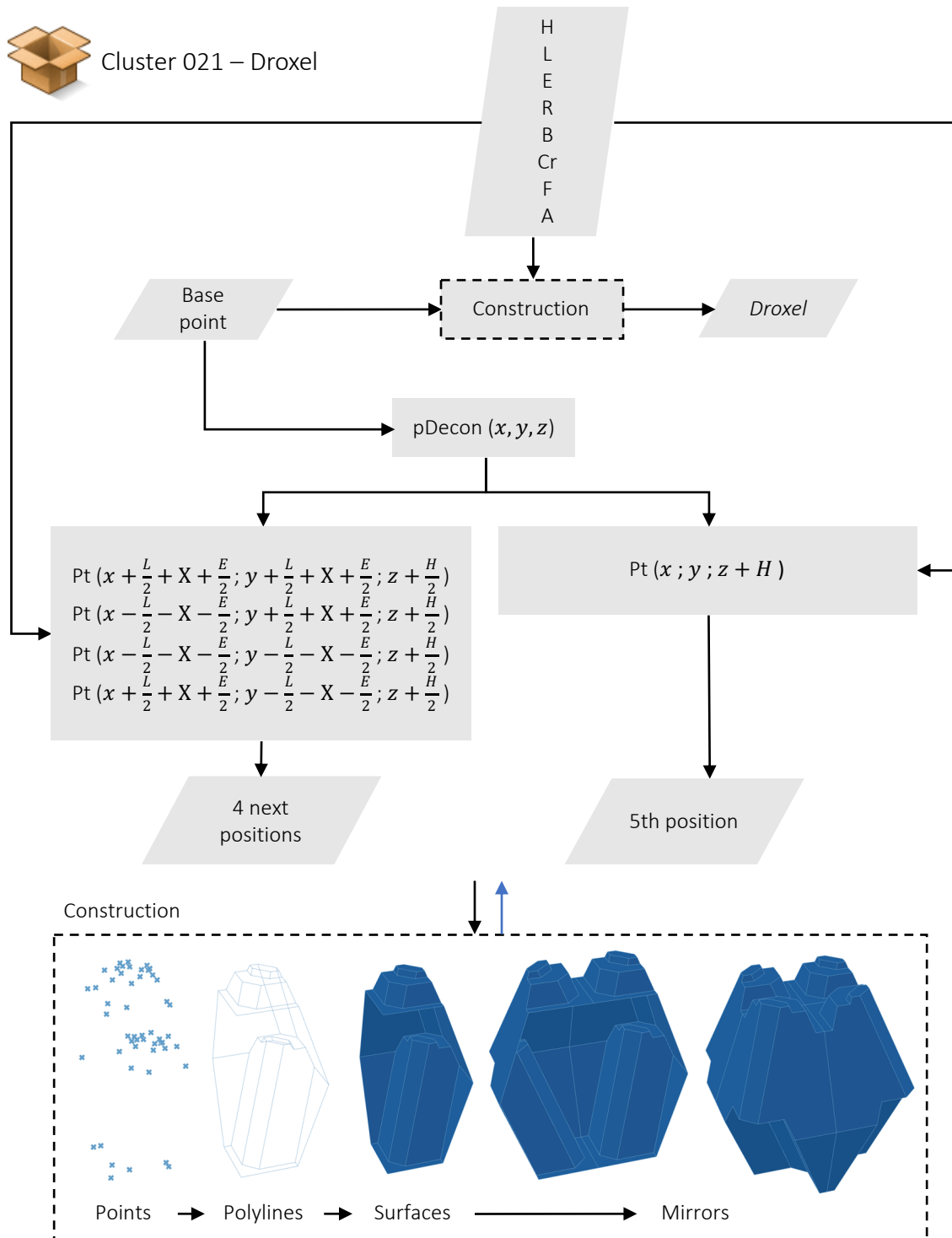
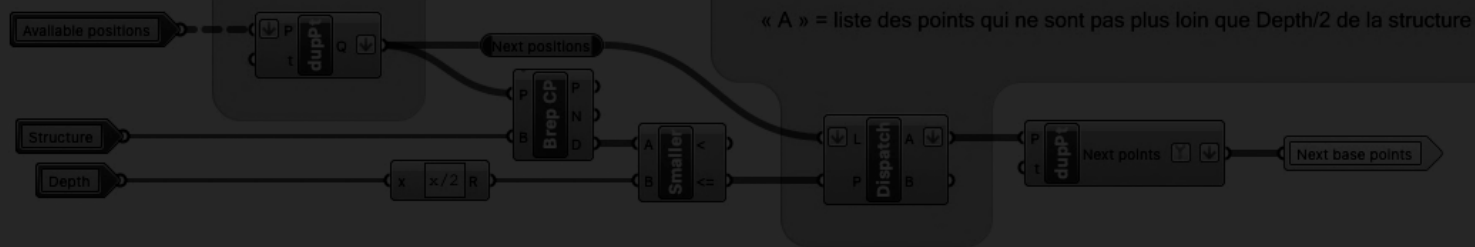


Figure 23 : Flow chart du cluster 021 – Droxel

« PolyLines » est un composant Grasshopper qui connecte une liste de points selon l'ordre de la liste.

« Mirror » est un composant Grasshopper qui effectue une symétrie selon un plan de symétrie.



4.1.5. 022 – Position

Le cluster **022 – Position** appartient au cluster **02 – Layer**. Il sert à ne sélectionner que les Base points qui ne sont pas plus éloignés de la demi *Depth* de la structure à droxéliser.

Inputs	[Unité] / type d'objet Gh	Description
Available positions	List of points	Liste de tous les Base points disponibles pour la couche suivante.
Structure	Brep	Géométrie de la structure à droxéliser.
Depth	[Scale]	Épaisseur de la structure à droxéliser.

Tableau 9 : Inputs du cluster 022 - Position

Output	[Unité] / type d'objet Gh	Description
Next base points	List of points	Liste de tous les Base points acceptés pour la couche suivante.

Tableau 10 : Outputs du cluster 022 - Position



Cluster 022 - Position

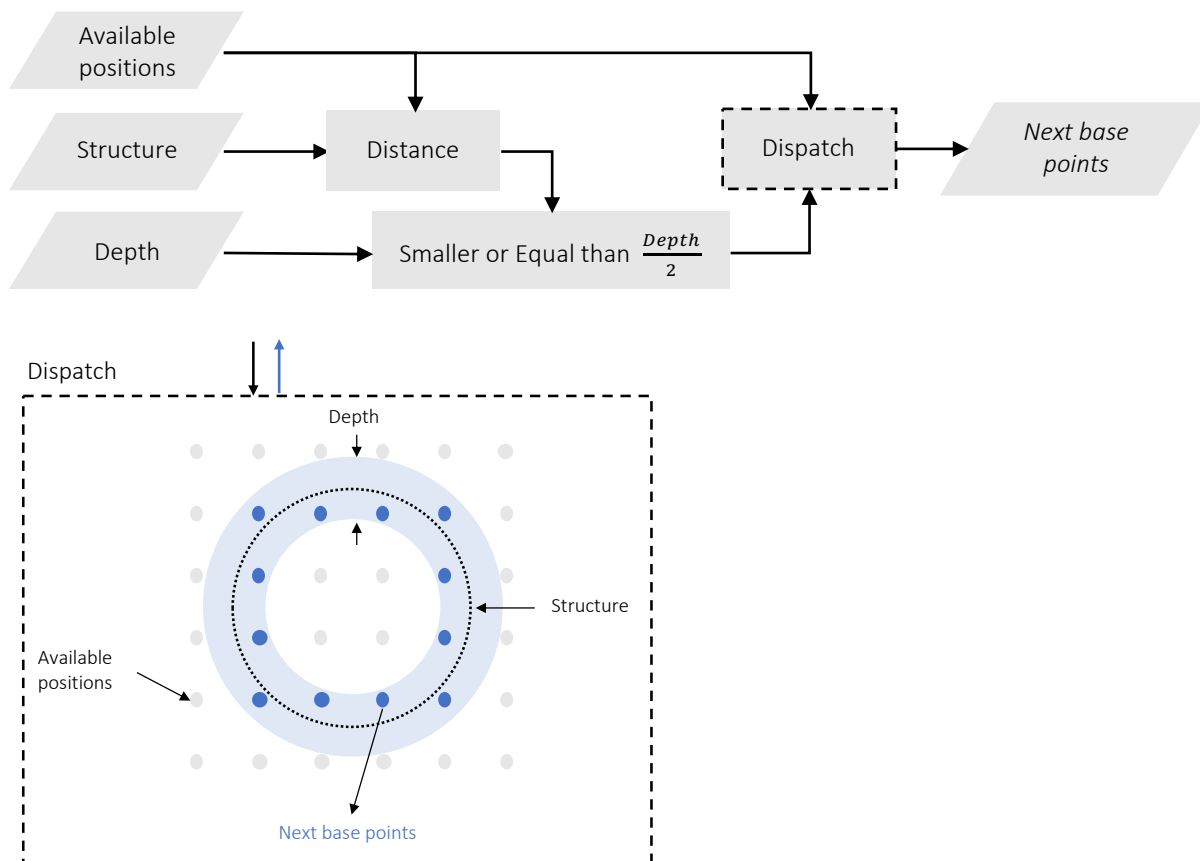


Figure 24 : Flow chart du cluster 022 – Position

4.1.6. 023 – Stability Check

Le cluster **023 – Stability Check** analyse l'impact de l'ajout de nouveaux droxels sur la stabilité globale. On ne travaille pas avec les droxels en tant que tels mais avec leurs Base points qui sont leur identité. Les nouveaux droxels sont en réalité l'output du cluster **022 – Position**. On suppose que toute la couche est posée en même temps. Chacun des nouveaux Base points est analysé en admettant que les autres Base points de sa couche sont déjà présents.

Inputs	[Unité] / type d'objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$= \frac{H}{2} * \tan(A)$
R	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.
Point_i	List of points	Liste contenant les Base points acceptés pour la couche suivante.
All previous points	List of points	Liste contenant tous les Base points des droxels déjà existants.
Unstable substructures	List of groups of points	Liste qui contient les groupes des Base points des substructures instables précédents regroupés.

Tableau 11 : Inputs du cluster 023 – Stability Check

Output	[Unité] / type d'objet Gh	Description
Unstable substructures	List of groups of points	Liste qui contient les groupes des Base points de toutes les substructures instables.

Tableau 12 : Outputs du cluster 023 – Stability Check

Tout d'abord il compile les dix clusters **0231 – Step Sub** consécutifs (Figure 25), c'est-à-dire le nombre de couches que l'on peut construire ou le même nombre de fois que le cluster **02 – Layer** est présent dans le canevas principal.

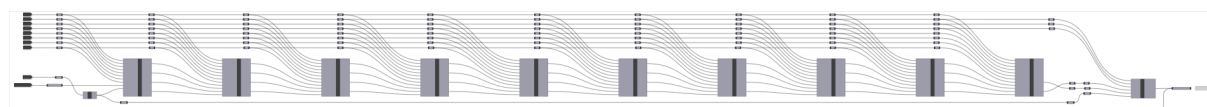


Figure 25 : Cluster 023 - Stability Check

Le fait qu'il soit en série permet de voyager dans les couches de droxels du haut vers le bas afin de recenser les Substructures.

Un ensemble de droxels porte le nom de Substructure en fonction des conditions d'appui de son ou ses droxels situés au plus bas de l'ensemble. Cet ensemble ne peut donc grandir que si les droxels situés au plus bas ne respectent pas la condition qui ferait de cet ensemble une Substructure.

Ensuite le cluster **0232 – Sub Edit** (Figure 26) fait en sorte de ne retenir que les Substructures qui ont été jugées instables. Ces Substructures viennent s’ajouter à la liste « Unstable substructures » qui regroupe celles détectées précédemment.

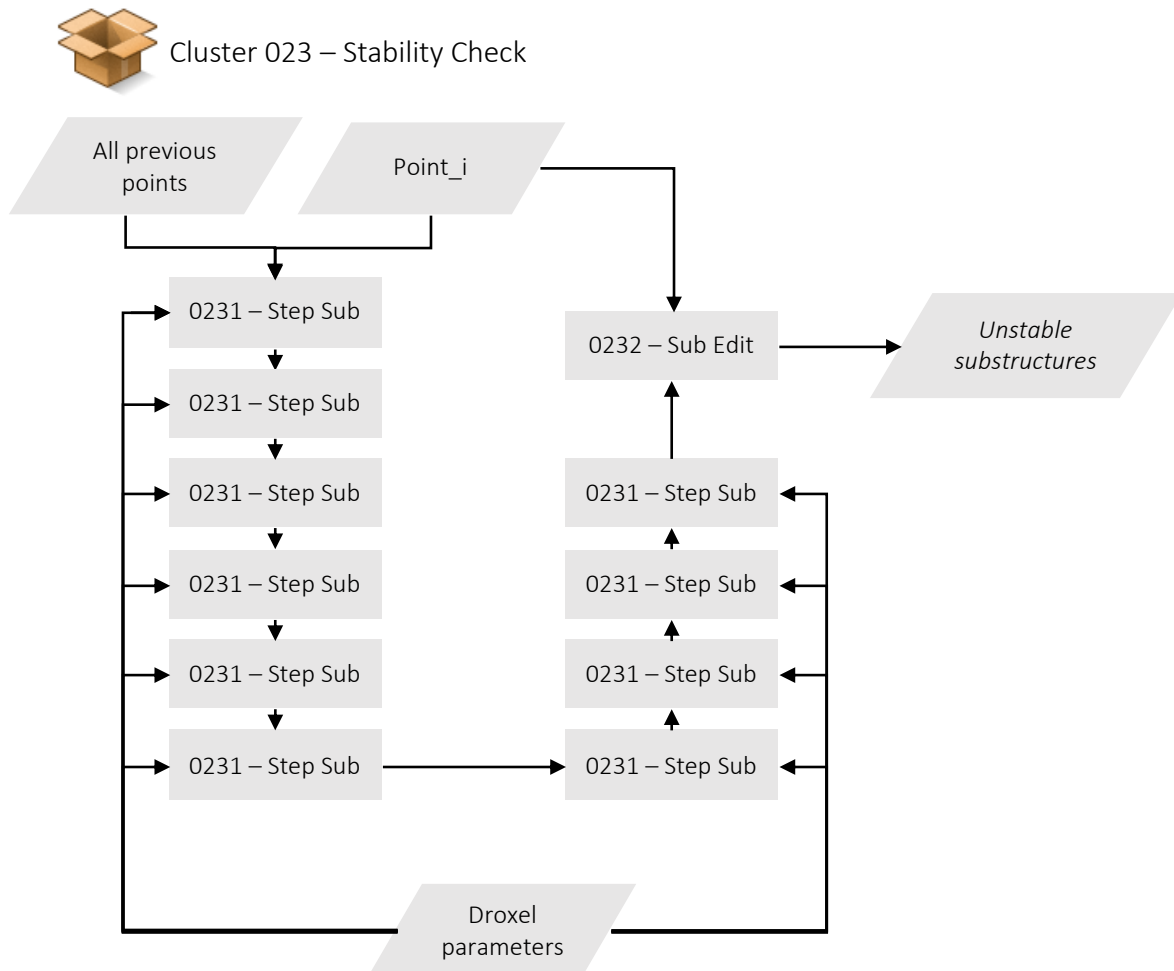


Figure 26 : Flow chart du cluster 023 – Stability Check

4.1.7. 0231 – Step Sub

Le cluster **0231 – Step Sub** regroupe les processus importants pour la création d'informations sur les Substructures. Chaque Base point de la nouvelle couche génère un ensemble composé d'autres Base points connectés parmi les couches inférieures. Un ensemble grandit au fur et à mesure qu'on descend dans la structure (la succession des clusters permet de descendre). Le flow chart de la *Figure 27* montre l'enchaînement des processus du cluster **0231 – Step Sub**.

En effet, les inputs *point_i* sont les Base points situés au plus bas du groupe formé par le cluster précédent (*Tableau 13*). Tout d'abord on ajoute au *point_i* ses voisins inférieurs. Ensuite pour chaque voisin inférieur on ajoute ses voisins supérieurs et également les voisins supérieurs de ces derniers et ainsi de suite jusqu'à la dernière couche (la nouvelle).

On identifie les Base points situés au plus bas de cet ensemble et on détermine à quel mode de pose ils appartiennent (*Figure 4*). Le mode de pose des droxels situés au plus bas d'un ensemble permet d'établir si cet ensemble est une sous-structure.

En parallèle on calcule et on teste si la projection verticale du centre de gravité du groupe de droxels est incluse dans la surface d'appui du groupe. Le résultat du test sera gardé dans le cas où l'ensemble s'avère être une Substructure.

Le résultat final qui ressort d'un cluster **0231 – Step Sub** pour un Base point qui entre en input, est un ensemble de Base points, lié à un statut : soit « *Hooked : not a substructure* » soit « *Center of gravity test : Stable* » soit « *Center of gravity test : Unstable* ».

Inputs	[Unité] / type d'objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$= \frac{H}{2} * \tan(A)$
R	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.
Tol	[Scale]	Distance de sécurité pour réduire la surface d'appui d'une Substructure.
All previous points	List of points	Liste contenant tous les Base points des droxels déjà existants.
point_i	List of groups of points	Liste des Base points situés au plus bas des potentielles ⁸ Substructures précédemment trouvées regroupés pour chacune des nouvelles positions analysées par le cluster 023 – Stability Check (Point_i).

⁸ Parler de « potentielle Substructure » signifie parler de sous-ensemble en général car il est possible qu'il soit une sous-structure ou non.

Statements	List of groups of texts	Statuts des potentielles Substructures précédemment trouvées regroupés pour chaque Base points de Point_i.
Sub	List of groups of points	Liste contenant les potentielles Substructures précédemment trouvées, regroupées pour chaque Base points de Point_i.

Tableau 13 : Inputs du cluster 0231 - Step Sub

Outputs	[Unité] / type d'objet Gh	Description
All points	List of points	Liste contenant tous les Base points des droxels déjà existants incluant également les Point_i.
Lowest	List of groups of points	Liste reprenant les Base points situé au plus bas des potentielles Substructures, regroupés pour chaque Base points de Point_i.
Statements	List of groups of texts	Statuts des potentielles Substructures regroupés pour chaque Base points de Point_i.
Sub	List of groups of points	Liste contenant les potentielles Substructures potentielles générées et regroupées pour chaque Base points de Point_i.

Tableau 14 : Outputs du cluster 0231 - Step Sub

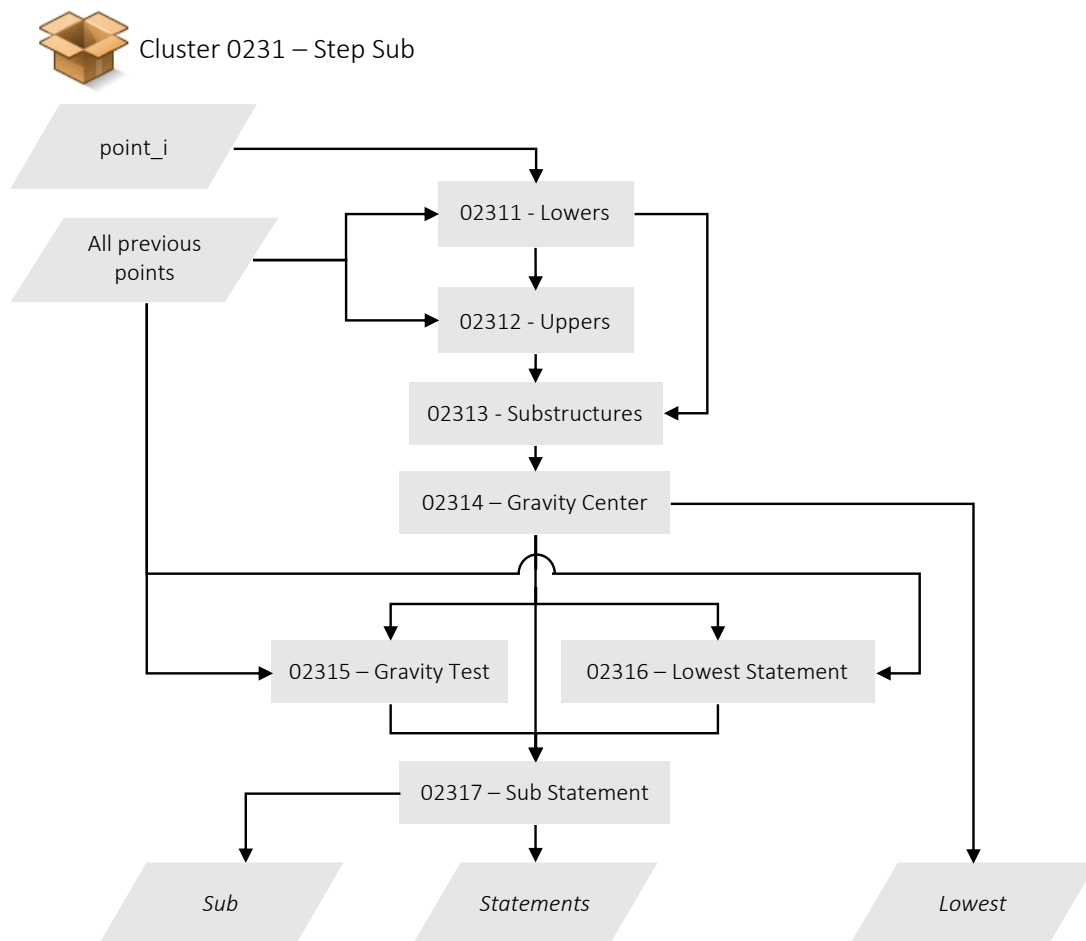
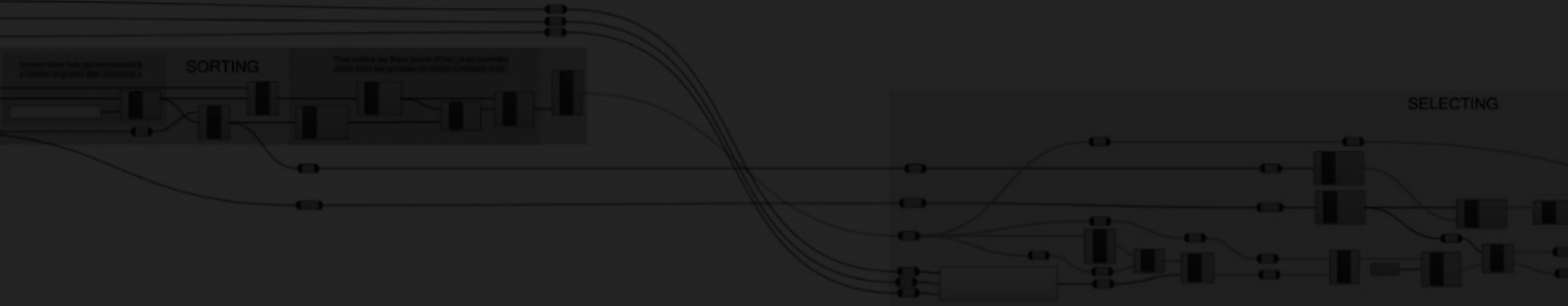


Figure 27 : Flow chart du cluster 0231 - Step Sub



4.1.8. 0232 – Sub Edit

Le cluster **0232 – Sub Edit** intervient une seule fois par couche (cluster **02 – Layer**) et après que tous les clusters **0231 – Step Sub** de la couche soient compilés. Le cluster est ensuite divisé en deux procédures, le « Sorting » et le « Selecting » (*Figure 28*).

Dans la procédure « Sorting » on traite deux listes qui sont synchronisées : une liste des groupes de droxels engendrés par chaque droxel de la nouvelle couche et une seconde reprenant leurs statuts. Cette dernière liste peut contenir les différents statuts suivants : *Hooked : not a substructure – Center of gravity test : Unstable – Center of gravity test : Stable*. On ne va sauvegarder que les groupes de droxels liés au statut « *Center of gravity test : Unstable* », qu'on peut à présent nommer des sous-structures instables.

Les sous-structures sont des ensembles qui regroupent des Base points. On arrange chacune de ces listes pour que les Base points de la nouvelle couche soient en première place dans les listes. Cet arrangement sert à faciliter le code dans le cluster **03 - Preview** pour mettre en évidence les droxels qui engendrent les sous-structures. La *Figure 29* est un exemple du fonctionnement du « Sorting ».

Ensuite les sous-structures instables passent par la procédure du « Selecting ». Ce processus sert à faire une sélection parmi les sous-structures instables. Certaines sous-structures jugées instables par l'algorithme ne le sont pas en réalité du à la méthode de droxelisation par couches. Expliquons cela avec ce qui suit.

Si nous émettons l'hypothèse que tous les droxels de la nouvelle couche sont posés en même temps, des résultats absurdes pourraient survenir. En effet, l'ajout d'un droxel peut générer une sous-structure que l'algorithme juge instable par ses conditions d'appui et d'équilibre.

Sans le « selecting » le cluster **0232 – Sub Edit** ne prend pas en compte le lien avec les autres sous-structures générées par les droxels de la même couche.

- Si le droxel est seul dans sa couche et qu'aucun autre bloc n'est mis en place en même temps que celui-ci, il n'y aura pas de contradiction.
- De la même façon, aucun autre problème ne surgit si la sous-structure jugée instable n'inclut aucun autre droxel de la nouvelle couche.
- Ou encore si tous les droxels de la nouvelle couche qui appartiennent à la sous-structure instable génèrent également une instabilité.

Cependant, une contradiction apparaît lorsqu'au moins un des droxels qui appartient à la nouvelle couche et à la sous-structure instable ne génère pas d'instabilité.

Cette situation contradictoire qu'il faut gérer est représentée à l'*Image 1*⁹. Avant de compiler le « selecting », le programme établit que ce système est instable. Alors que, comme on peut le voir, les droxels sont à l'équilibre. Les sous-structures 1 et 2 sont stables mais la sous-structure 3, elle, est instable à ce stade.

On constate que le droxel B est présent dans les sous-structures 2 et 3, mais aussi que la sous-structure qu'il génère est en équilibre. Or un seul état est possible pour le droxel B. Il faut donc considérer que la sous-structure 3 est stable parce que le droxel B ne crée pas d'instabilité (« Selecting » : *Figure 28*).

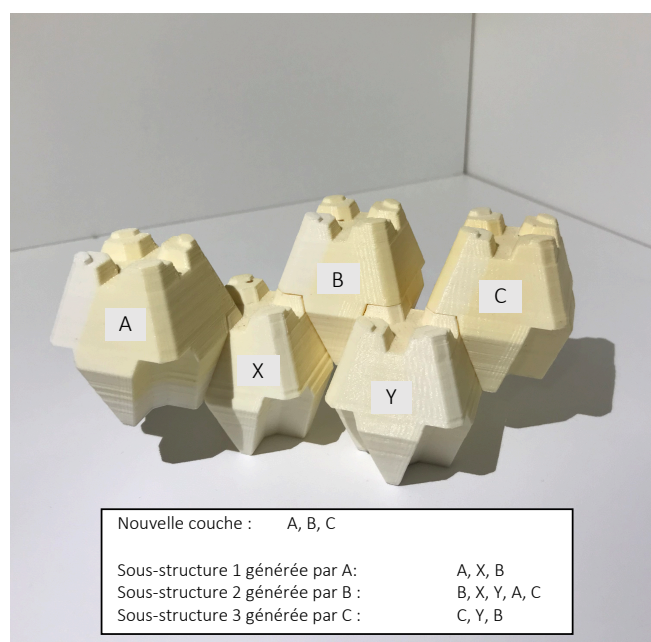


Image 1 : Exemple du processus "selecting"

Inputs	[Unité] / type d'objet Gh	Description
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$\frac{H}{2} * \tan(A)$
Sub	List of groups of points	Liste des ensembles de Base points (potentielles Substructures) générés par chaque Base point de la couche suivante (Point_i).
Statements	List of texts	Liste des statuts des ensembles de Base points générés par chaque Base de la liste Point_i.
Point_i	List of points	Liste des Base points pour la couche suivante.

Tableau 15 : Inputs du cluster 0232 - Sub Edit

Output	[Unité] / type d'objet Gh	Description
Unstable sub	List of groups of points	Liste qui contient tous les groupes de base points des Substructures instables générées par les droxels de la couche (chaque point de Point_i).

Tableau 16 : Outputs du cluster 0232 - Sub Edit

⁹ Les droxels X et Y ne sont pas encastrés. Ils sont donc « not hooked ». Il existe dès lors automatiquement des sous-structures.

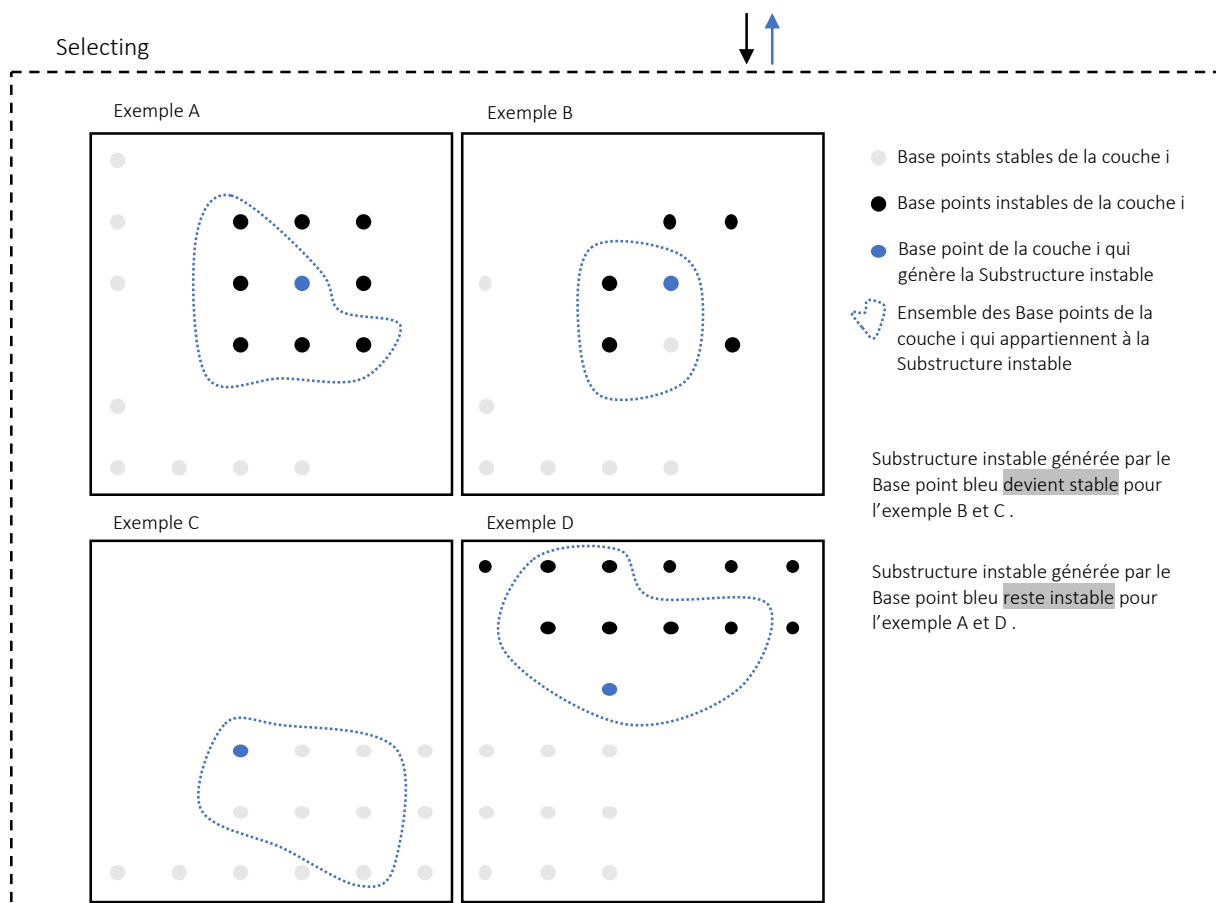
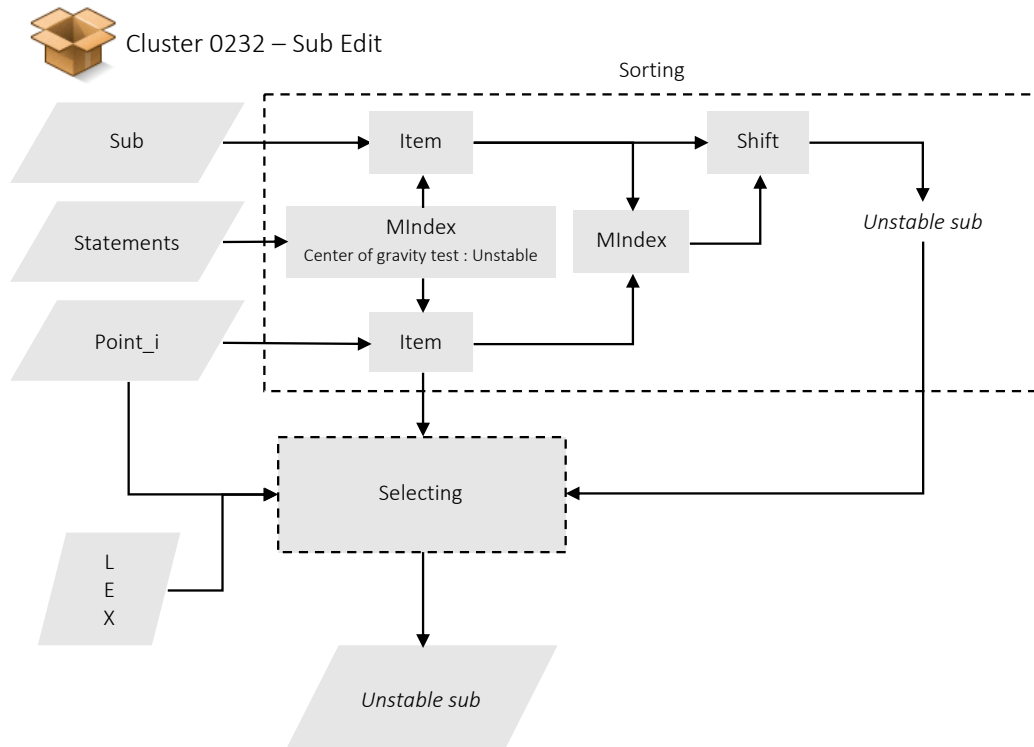


Figure 28 : Flow chart du cluster 0232 - Sub Edit

« MIndex » est un composant Grasshopper qui trouve l'indice d'un élément ciblé dans une liste. Dans le cas étudié, on recherche le texte « Center of gravity test : Unstable » dans la liste « Statements ».

« Item » récupère un élément spécifique dans une liste grâce à son indice.

« Shift » décale les membres d'une liste (« Sub ») d'un certain offset (indice du Base point de Point_i dans la sous-structure).

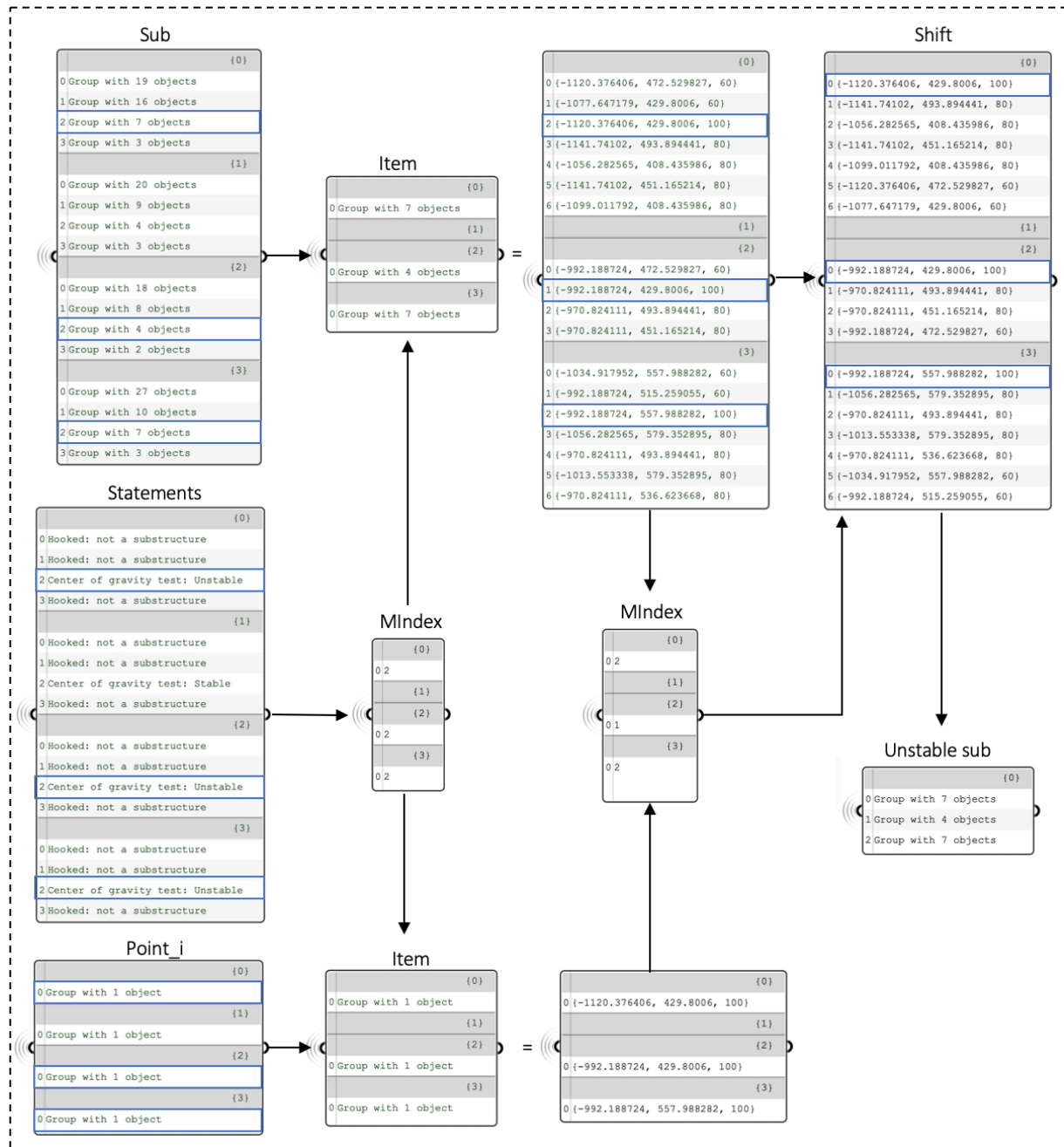
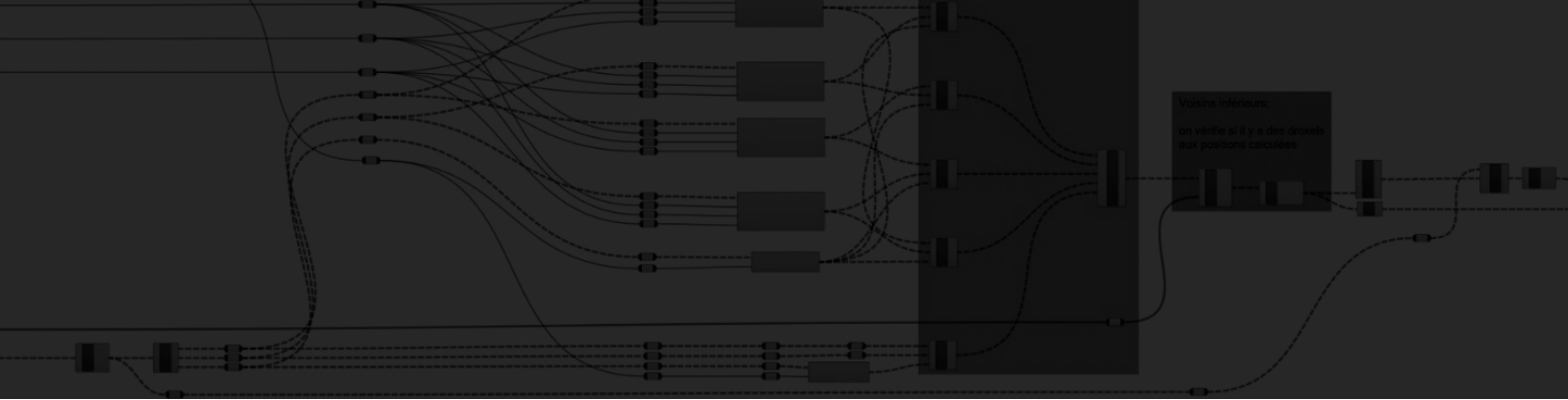


Figure 29 : Exemple du processus « Sorting »¹⁰

¹⁰ Il faut garder en tête que les trois listes en inputs sont structurées de la même manière.



4.1.9. 02311 – Lowers

Le cluster **02311 – Lowers** trouve les voisins inférieurs (*Figure 6*) d'un droxel.

On procède de la manière suivante : on calcule les positions sur lesquelles pourraient se trouver des droxels inférieurs et on compare ces positions avec les tous les Base points déjà existants de la structure.

On ajoute les Base points qui correspondent aux positions avec le composant Grasshopper « Intersection » (*Figure 30*) qui crée une liste avec les objets similaires et communs provenant de deux listes initiales.

On calcule le nombre de voisins inférieurs car cela est utile dans le cluster **02317 – Sub Statement** pour confirmer l'hypothèse que les droxels de la première couche sont encastrés. En effet si un droxel ne possède pas de voisin inférieur ($nlo = 0$) cela signifie qu'il appartient à la première couche.

Inputs	[Unité] / type d'objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$= \frac{H}{2} * \tan(A)$
All points	List of points	Liste contenant tous les Base points des droxels déjà existants.
point_i	List of points	Liste des Base point situés au plus bas des potentielles Substructures précédemment trouvées.

Tableau 17 : Inputs du cluster 02311 - Lowers

Outputs	[Unité] / type d'objet Gh	Description
lowers	List of groups of points	Liste comprenant tous les Base points des droxels qui sont inférieurs aux droxels des point_i regroupés par Point_i.
nlo	List of groups of numbers	Nombre de voisins inférieurs.

Tableau 18 : Outputs du cluster 02311 – Lowers



Cluster 02311 – Lower

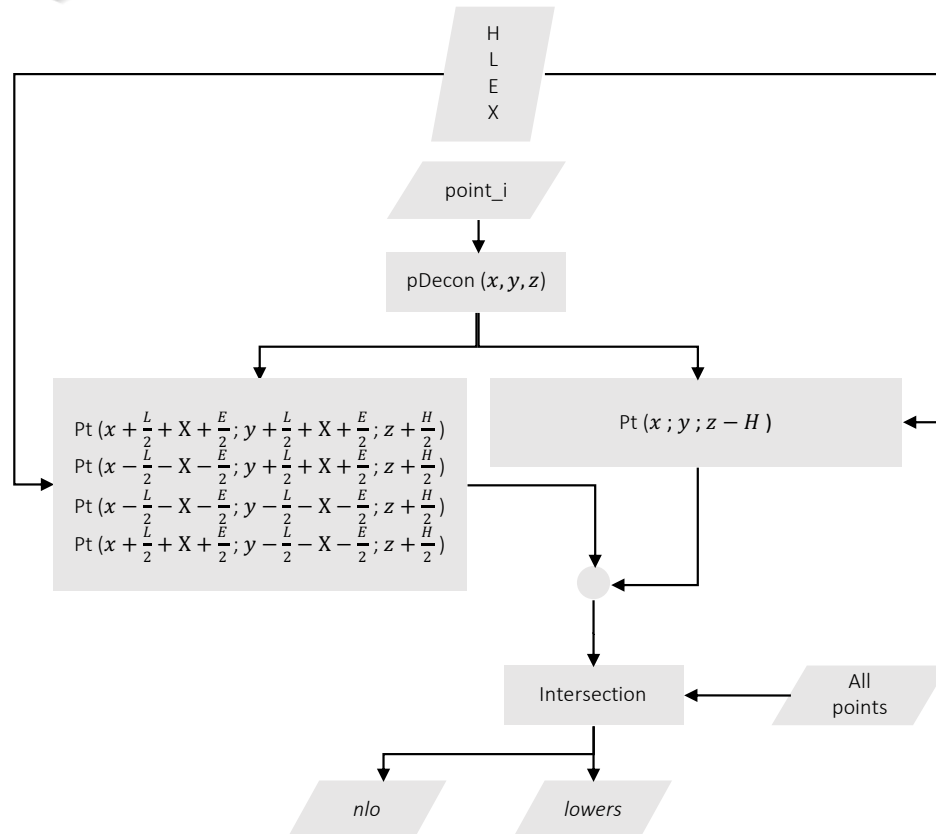


Figure 30 : Flow chart du cluster 02311 – Lower



4.1.10. 02312 – Uppers

Le cluster **02312 – Uppers** cherche les voisins supérieurs (Figure 6). Ce cluster existe pour alléger le canevas du cluster **0231 – Step Sub** et concentre dix autres clusters en séries. De même que dix clusters sont nécessaires pour voyager vers le bas dans la structure, dix autres le sont également pour remonter les couches de droxels (Figure 31).

A chaque fois que le cluster **0231 – Step Sub** est compilé on descend dans les couches et on s'éloigne de la nouvelle couche. Pour recenser une sous-structure, il faut ajouter les voisins supérieurs mais également leurs voisins supérieurs et ainsi de suite jusqu'à la dernière couche de droxels existante. S'il peut y avoir dix couches de droxels cela signifie qu'il est possible de devoir en remonter le même nombre. Ce n'est rien d'autre que le même procédé qui se répète (Figure 31).

Inputs	[Unité] / type d'objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$\frac{H}{2} * \tan(A)$
All points	List of points	Liste comprenant tous les Base points.
find upper	List of groups of points	Liste des Base points des voisins inférieurs regroupés par Point_i.

Tableau 19 : Inputs du cluster 02312 - Uppers

Outputs	[Unité] / type d'objet Gh	Description
All uppers	List of groups of points	Liste comprenant tous les Base points des voisins supérieurs consécutifs des point_i regroupés par Point_i.

Tableau 20 : Outputs du cluster 02312 – Uppers

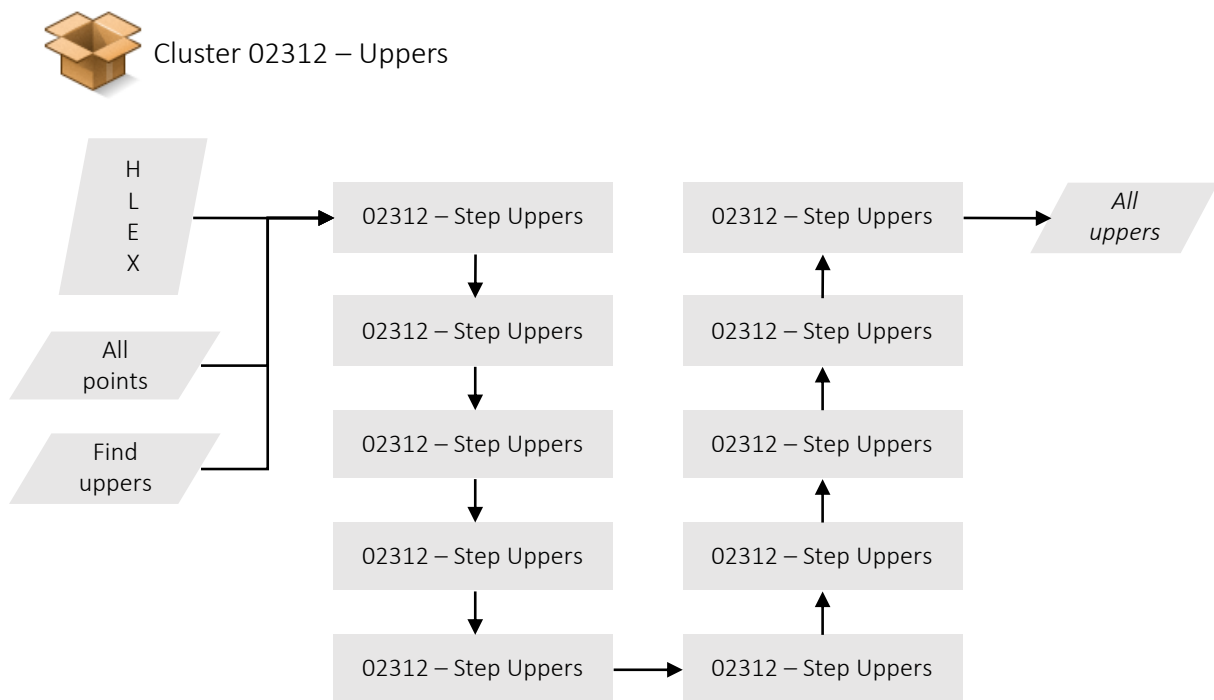
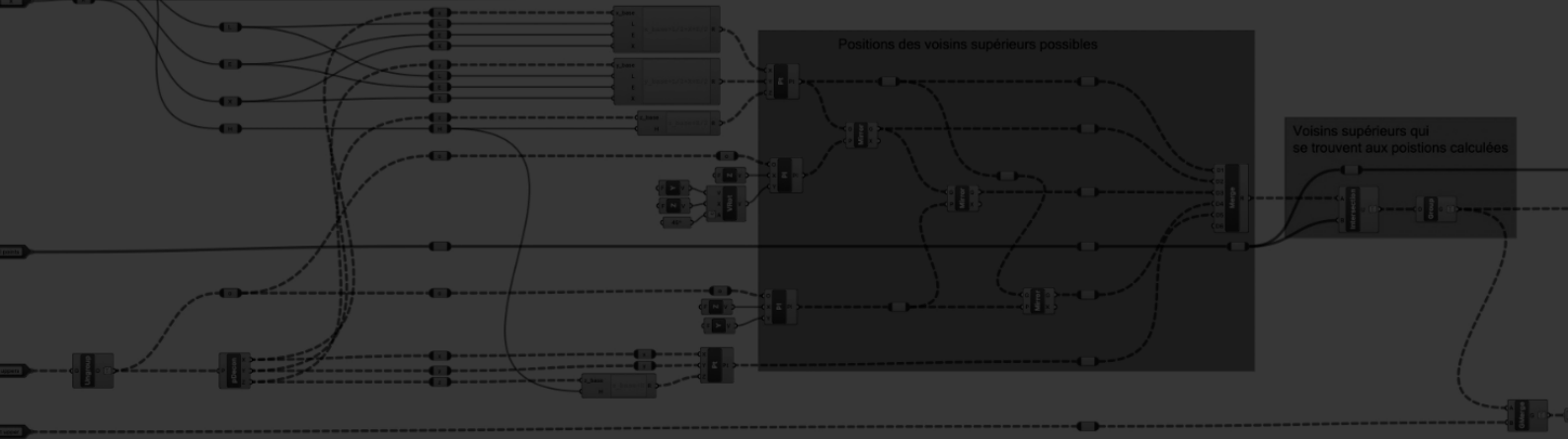


Figure 31 : Flow chart du cluster 02312 – Uppers



4.1.11. 023121 – Step Uppers

Le cluster **023121 – Step Uppers** détecte les voisins supérieurs d’un droxel. Ce cluster est répété 10 fois dans le cluster **02312 – Uppers** afin de pouvoir remonter dans la structure au moins jusqu’à la couche analysée.

Le cluster commence par détecter les voisins supérieurs des droxels trouvés par le cluster **02311 – Lowers** et ainsi de suite pour tous les voisins supérieurs. On additionne successivement tous les Base points qui forment petit à petit toute la Substructure.

On remarque dans le flow chart (*Figure 32*) que la manière pour trouver les voisins n’est pas tout à fait identique à celle utilisée précédemment. Lorsqu’on trouve le voisin 1, on utilise des symétries autour du Base point pour obtenir les voisins 2, 3 et 4 (*Figure 6*) au lieu de trouver chacun d’eux séparément.

Inputs	[Unité] / type d’objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$= \frac{H}{2} * \tan(A)$
All points	List of points	Liste comprenant tous les Base points.
find upper	List of of groups of points	Liste des Base points dont il faut trouver les voisins supérieurs.
Count upper	List of groups of points	Liste de tous les voisins supérieurs précédents.

Tableau 21 : Inputs du cluster 023121 - Step Uppers

Outputs	[Unité] / type d’objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$= \frac{H}{2} * \tan(A)$
All points	List of points	Liste comprenant tous les Base points.
upper	List of of groups of points	Liste des Base points des voisins supérieurs détectés.
Count upper	List of groups of points	Liste de tous les voisins supérieurs.

Tableau 22 : Outputs du cluster 023121 - Step Uppers



Cluster 023121 – Step Uppers

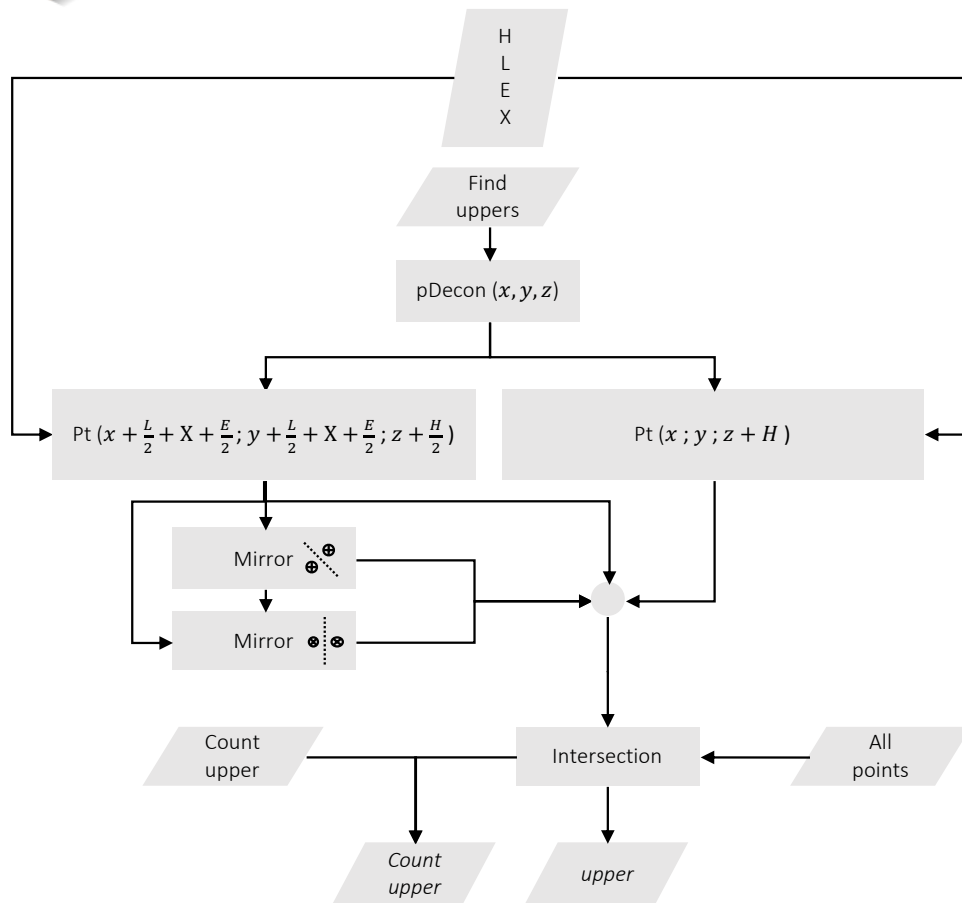
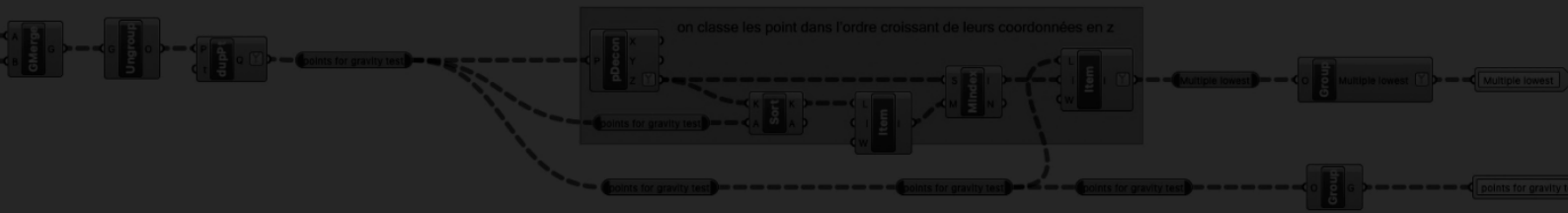


Figure 32 : Flow chart du cluster 023121 - Step Uppers



4.1.12. 02313 – Substructure

Le cluster **02313 – Substructure** additionne les voisins inférieurs et tous les voisins supérieurs successifs pour former les ensembles de droxels susceptibles d’être des Substructures. Une fois l’ensemble formé, il est possible de le classer pour trouver les Base points situés au plus bas de l’ensemble (*Figure 33*).

Inputs	[Unité] / type d’objet Gh	Description
uppers	List of groups of points	Liste comprenant tous les Base points des voisins supérieurs successifs.
lowers	List of groups of points	Liste comprenant les Base points des voisins inférieurs.

Tableau 23 : Inputs du cluster 02313 – Substructure

Outputs	[Unité] / type d’objet Gh	Description
Multiple lowest	List of groups of points	Liste comprenant les Base points des droxels situés au plus bas dans les Substructures potentielles.
points for gravity test	List of groups of points	Liste comprenant les Base points des droxels des Substructures potentielles regroupés par Point_ <i>i</i> .

Tableau 24 : Outputs du cluster 02313 – Substructure

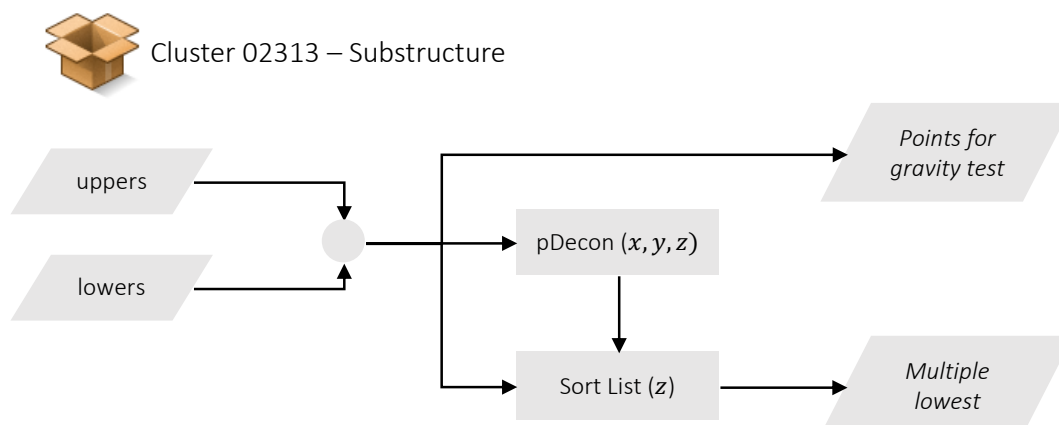
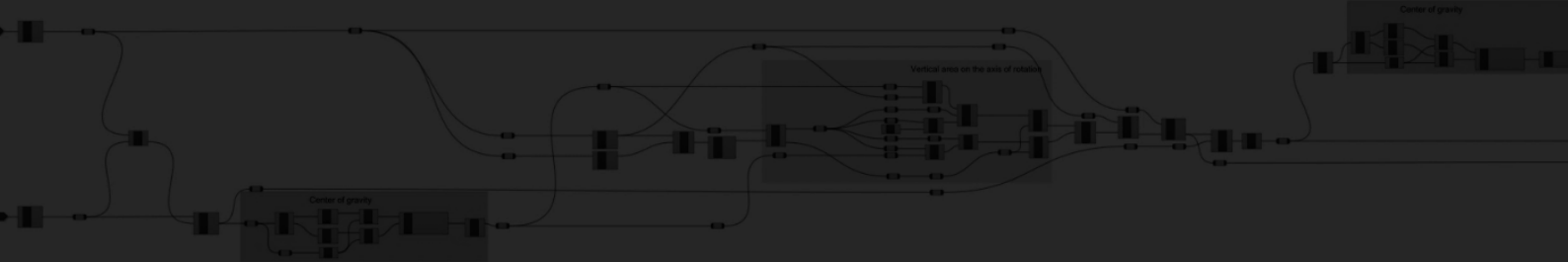


Figure 33 : Flow chart du cluster 02313 – Substructure

« Sort List » est un composant Grasshopper qui trie une liste cible dans un certain ordre. On peut lier une liste optionnelle qui sera synchronisée avec la liste cible. Ici on classe l’ensemble (uppers, lowers) dans l’ordre croissant de leurs coordonnées en z.



4.1.13. 02314 – Gravity Center

Le cluster **02314 – Gravity Center** s’occupe de déterminer les droxels parmi les plus bas qui vont effectivement être dans la Substructure potentielle assemblée par le cluster **02313 – Substructure** et ensuite il calcule le centre de gravité de l’ensemble.

Une Substructure instable bascule autour d’un axe de rotation qui se trouve sous les droxels situés au plus bas. Donc s’il existe plusieurs de ces droxels, seulement ceux positionnés sur l’axe basculeront avec la Substructure. Ce phénomène est illustré dans l’*Image 2*. Les droxels situés au plus bas sont 0.A, 0.B et 0.C. Aucun d’entre eux n’est accroché. Cependant on constate que le droxel 0.A ne bascule pas avec la sous-structure car il ne se trouve pas sur l’axe de basculement. Dès lors celui-ci ne sera pas pris en compte pour le calcul du centre de gravité de la Substructure ni pour le reste de l’analyse.

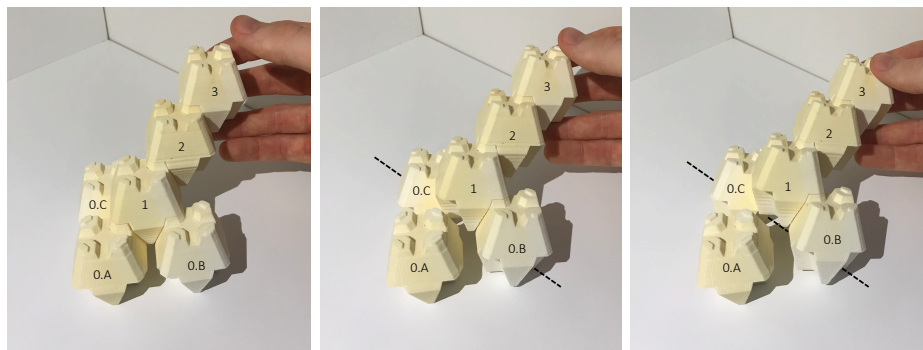


Image 2 : Exemple du cluster 02314 - Gravity Center

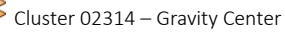
La position du centre de gravité est rapidement calculée. Dans cet algorithme tous les droxels sont de la même taille. Dans ce code Grasshopper, la masse n’étant pas paramétrable, chaque droxel possède toujours la même masse. Donc pour obtenir ses coordonnées en x et y, il suffit de pondérer la somme des coordonnées respectivement en x et en y de tous les Base points composant la sous-structure par leur nombre.

Inputs	[Unité] / type d’objet Gh	Description
Multiple lowest	List of groups of points	Liste comprenant les Base points des droxels situés au plus bas dans les Substructures potentielles.
points for gravity test	List of groups of points	Liste comprenant les Base points des droxels des Substructures potentielles.

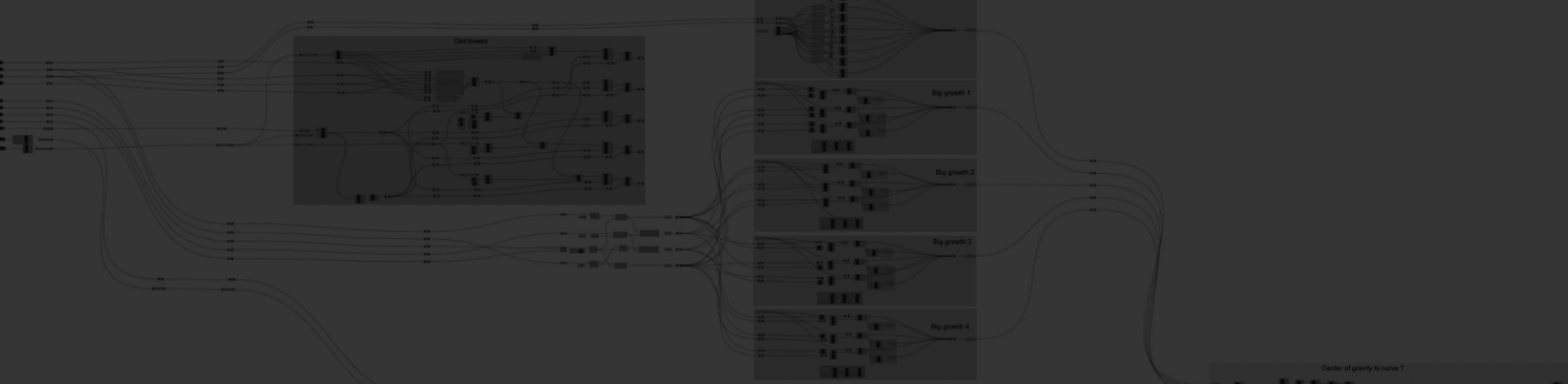
Tableau 25 : Inputs du cluster 02314 - Gravity Center

Outputs	[Unité] / type d’objet Gh	Description
Center of gravity	List of groups of point	Liste des centres de gravité de chaque potentielle Substructure.
points for gravity test	List of groups of points	Liste comprenant les Base points des droxels des Substructures potentielles.
Closest lowest	List of groups of points	Liste comprenant les Base points des droxels les plus bas et pertinents pour l’analyse dans les Substructures potentielles.

Tableau 26 : Outputs du cluster 02314 - Gravity Center



43



4.1.14. 02315 – Gravity Test

Le cluster **02315 – Gravity Test** construit la zone d’appui de chaque Substructure et teste si la projection verticale de leur centre de gravité se trouve à l’intérieur.

Il faut collectionner tous les droxels voisins sur lesquels ce groupe de droxels s’appuie et construire la surface de contact entre les deux. On travaille dans le repère cartésien général. On peut donc déterminer les positions exactes des surfaces de contact. Deux droxels empilés peuvent avoir plusieurs possibilités comme surface de contact. Soit une des quatre grandes excroissances ou soit la croix supérieure du droxel inférieur (*Figure 35*). On utilise ensuite le composant « Convex Hull » qui calcule le plus petit polygone convexe qui contient toutes les surfaces.

Si l’utilisateur paramètre le facteur « Tolerance » dans le canevas principal, la zone obtenue grâce au « Convex Hull » est réduite d’un offset égal au paramètre « Tolerance » multiplié par la longueur « L ». Il ne reste plus qu’à regarder si la projection verticale du centre de gravité est incluse dans cette surface à l’aide du composant « InCurve ». Le dernier objet du cluster, qui va fournir l’output, est une condition « IF ». On définit le statut du groupe de droxels dans la liste nommée « Gravity_2 » par stable ou instable selon le résultat du « InCurve ».

Inputs	[Unité] / type d’objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$= \frac{H}{2} * \tan(A)$
R	[Scale]	Paramètre géométrique du droxel.
Cr	[Scale]	Paramètre géométrique du droxel.
A	[°]	Paramètre géométrique du droxel.
Tol	[Scale]	Distance de sécurité pour réduire la surface d’appui d’une Substructure.
All points	List of points	Liste comprenant tous les Base points.
center of gravity	List of groups of point	Liste des centre de gravité de chaque potentielle Substructure.
points for gravity test	List of groups of points	Liste des Base points des potentielles Substructures.

Tableau 27 : Inputs du cluster 02315 - Gravity Test

Outputs	[Unité] / type d’objet Gh	Description
Gravity_2	List of groups of text	Résultats du test du centre de gravité pour chaque ensemble de droxels regroupés par Point_i.

Tableau 28 : Outputs du cluster 02315 - Gravity Test



Cluster 02315 – Gravity Test

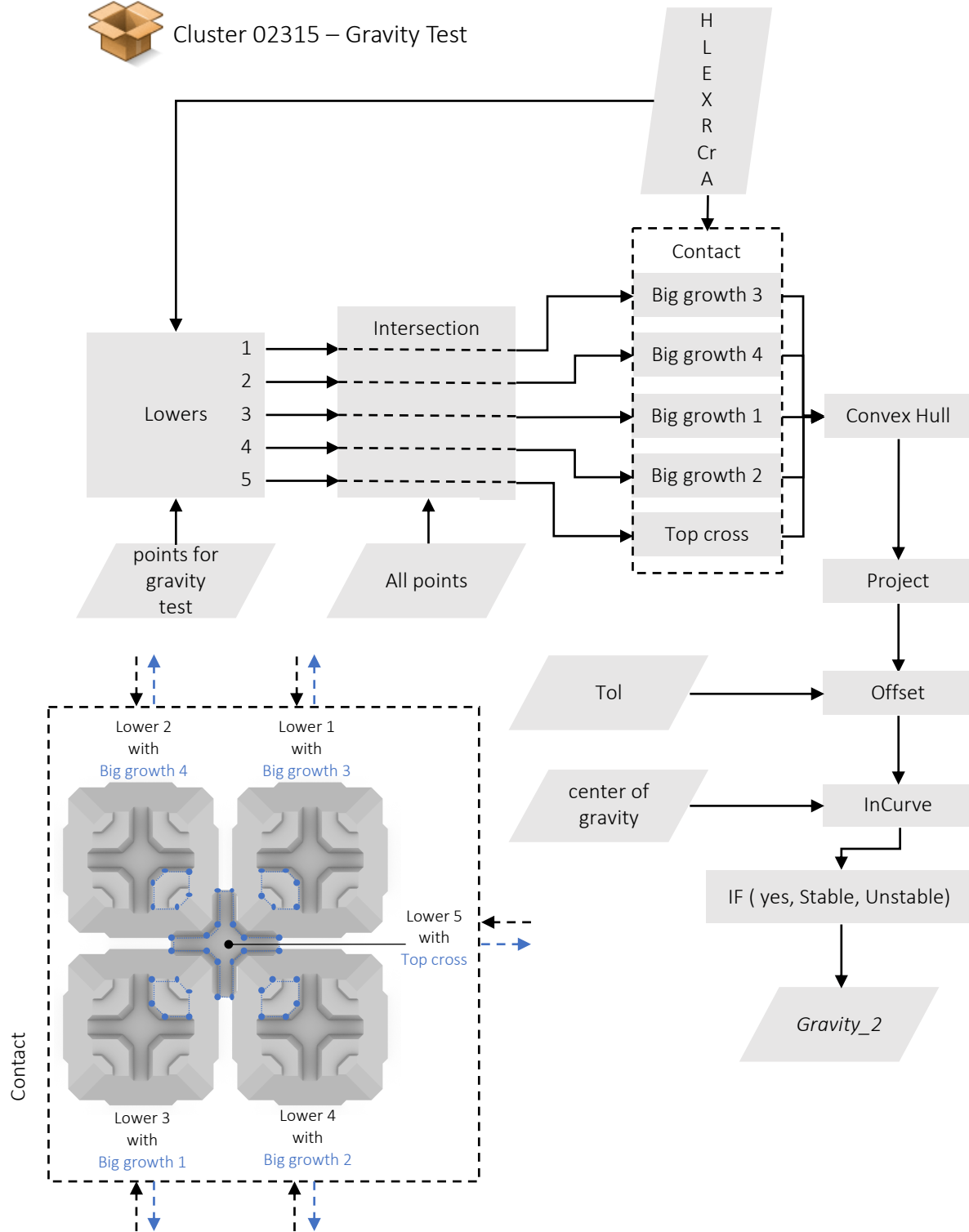


Figure 35 : Flow chart du cluster 02315 - Gravity Test

4.1.15. 02316 – Lowest Statement

Le cluster **02316 – Lowest Statement** évalue si un ensemble de droxels est une Substructure en trouvant le mode de ses droxels situés au plus bas. Un droxel de [mode 1, 3.a et 3.b](#) est « not hooked ». Les [modes 3.a et 3.b](#) ont chacun quatre possibilités différentes. Neuf situations différentes sont obtenues pour un droxel « not hooked » (*Figure 4*). Le cluster se résume à une condition « If(..) » (*Figure 36*). Il faut calculer les positions potentielles pour les voisins inférieurs (*Figure 6*) des droxels situés au plus bas et les confronter aux Base points qui existent réellement. En fonction de l'existence de tel ou tel voisin on peut déterminer si le droxel appartient à une des neuf variantes des modes « not hooked » (*Figure 36*). Si les droxels sont dans une de ces neuf variantes, leur groupe de droxels est qualifié de « Substructure » et de « Not a substructure » dans le cas contraire.

Inputs	[Unité] / type d'objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel.
L	[Scale]	Paramètre géométrique du droxel.
E	[Scale]	Paramètre géométrique du droxel.
X	[Scale]	$= \frac{H}{2} * \tan(A)$
All points	List of points	Liste comprenant tous les Base points.
Lowest	List of groups of points	Liste des Base points situés au plus bas des potentielles Substructures.

Tableau 29 : Inputs du cluster 02316 - Lowest Statement

Outputs	[Unité] / type d'objet Gh	Description
Statement of the lowest	List of groups of texts	Statuts des droxels situés au plus bas des ensembles de droxels (potentielles Substructures) regroupés par Point_i.

Tableau 30 : Outputs du cluster 02316 - Lowest Statement

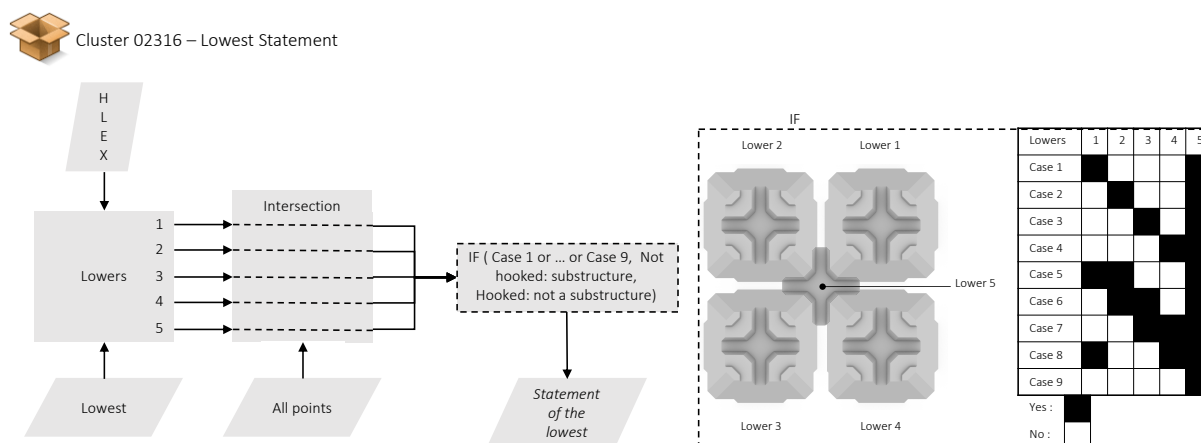


Figure 36 : Flow chart du cluster 02316 - Lowest Statement

4.1.16. 02317 – Sub Statement

Le cluster **02317 – Sub Statement** est le dernier processus du cluster **0231 – Step Sub**. Il assemble les informations au sujet des collections de droxels analysés. Il est codé sous forme d’une condition « IF ». Si tous les droxels situés au plus bas d’un ensemble sont qualifiés de « not hooked », on renvoie le résultat du test du centre de gravité, « Gravity_2 ». Sinon, dans le cas où il y a encore des droxels inférieurs, l’ensemble est qualifié de « not a substructure ».

Inputs	[Unité] / type d’objet Gh	Description
Statement of the lowest	List of groups of texts	Statuts des droxels situés au plus bas des ensembles de droxels.
Gravity_2	List of groups of text	Résultats du test du centre de gravité.
points for gravity test	List of groups of points	Liste comprenant les Base points des droxels des Substructures potentielles.
nlo	List of groups of numbers	Nombre de voisins inférieurs.

Tableau 31 : Inputs du cluster 02317 - Sub Statement

Outputs	[Unité] / type d’objet Gh	Description
Statements	List of groups of text	Statut des Substructures potentielles pour chaque Point_i.
Points for sub	List of groups of points	Base points des droxels pour chaque collection.

Tableau 32 : Outputs du cluster 02317 - Sub Statement

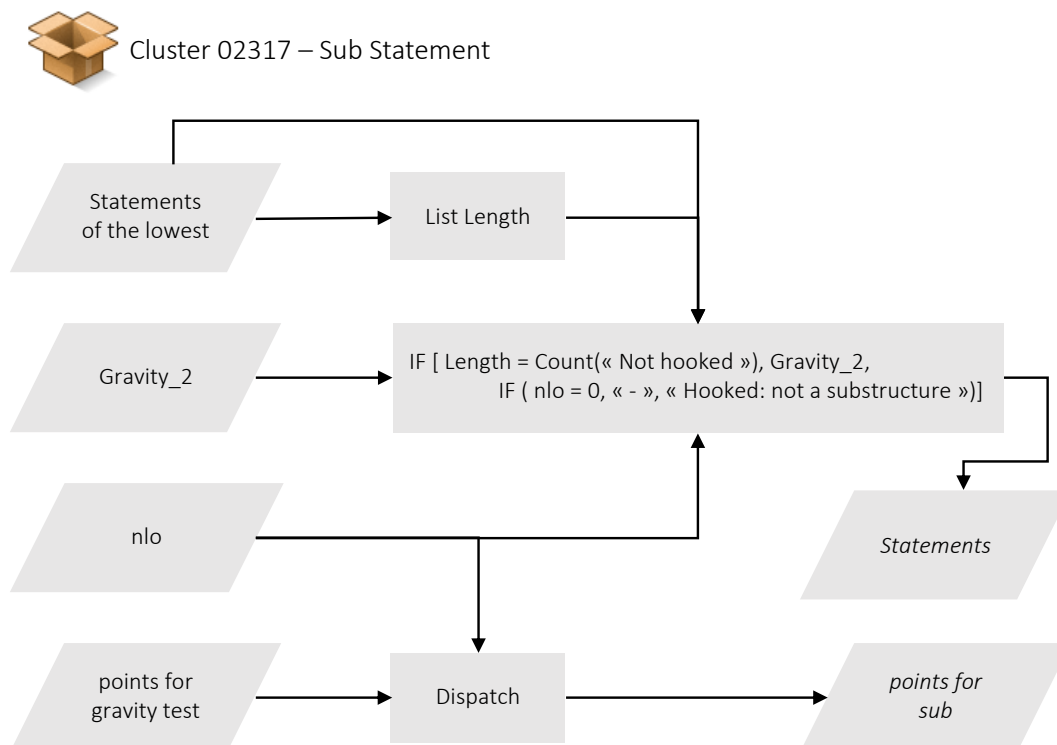


Figure 37 : Flow chart du cluster 02317 - Sub Statement

5. Application du code Grasshopper avec des exemples

Plusieurs hypothèses sont établies lors de la programmation en Grasshopper. On suppose que tous les droxels d'une même couche sont mis en place au même moment. C'est à dire que lorsqu'on analyse l'impact de l'addition d'un nouveau droxel sur la stabilité globale de la structure, on suppose que tous les autres droxels communs à la couche sont déjà présents.

Afin de comparer quelques exemples de structures droxelisées à la réalité, on dispose de droxels imprimés en 3D à échelle réduite et d'un plateau d'essai sur lequel on posera les droxels. Les droxels de la première couche sont suffisamment enfoncés dans le plateau pour qu'on les considère encastres (du moins pour l'exemple A et B). Les blocs ont été mesurés précisément à l'aide d'un pied à coulisse (*Image 3*). Les paramètres géométriques des droxels ont donc pu être entrés dans le programme Grasshopper (*Tableau 33*).

Inputs		[Unité]
H	40	[mm]
L	27	[mm]
E	3.5	[mm]
R	4.2	[mm]
B	1.2	[mm]
Cr	4	[mm]
F	40	[%]
A	17	[°]
Tol	2.55	[%]



Image 3 : Pied à coulisse

Tableau 33 : Paramètres géométriques des droxels imprimés¹¹

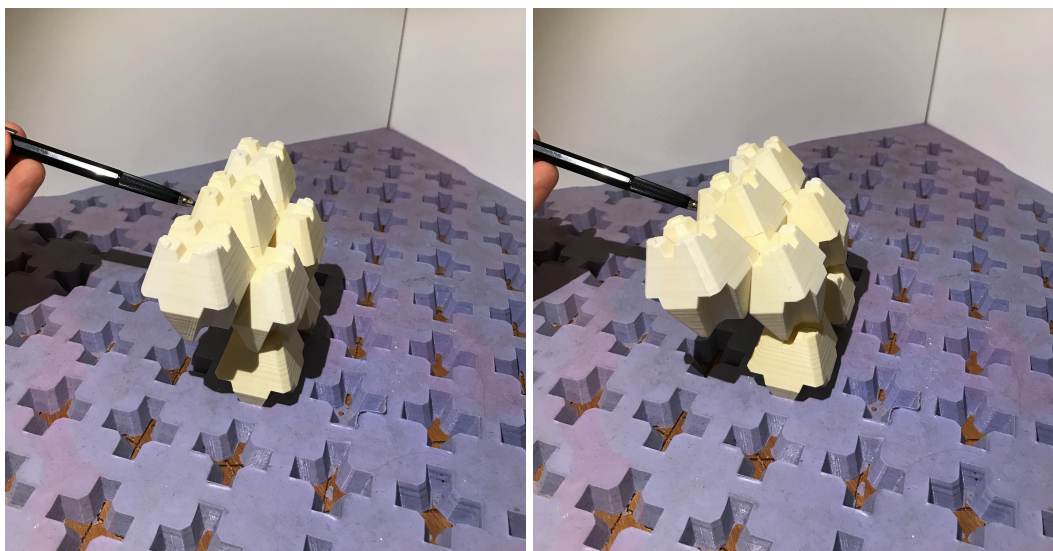
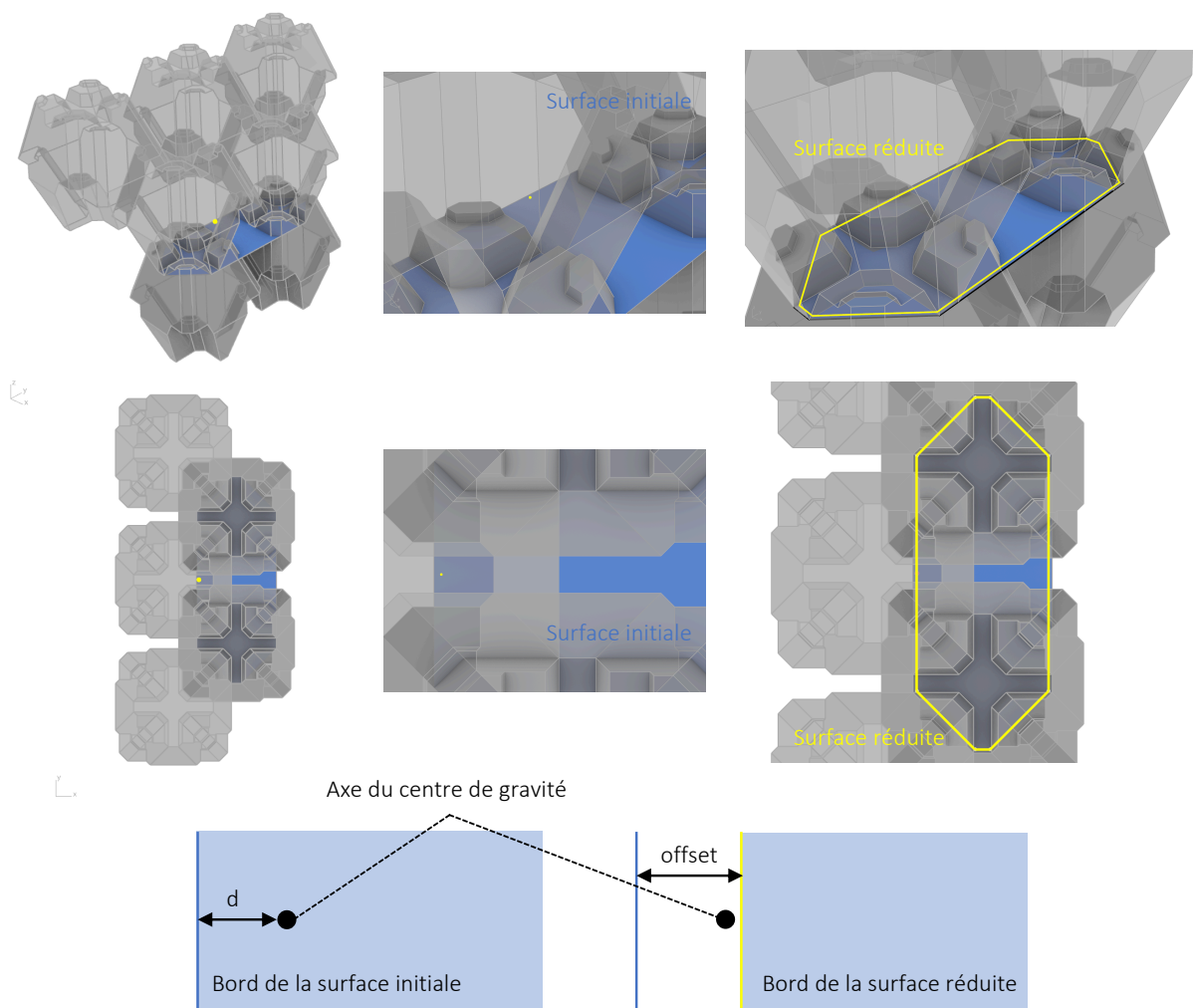
Le paramètre *Depth* varie selon les exemples. Nous fixons le paramètre *Tol* à 2,55 %, ce qui signifie que les surfaces d'appui des sous-structures sont réduites d'un offset égal à 0,6885 mm.

$$offset = L * Tol = 27 \text{ mm} * 2,55 \% = 0,6885 \text{ mm} \quad (Eq. 1)$$

Ce facteur est évalué sur base de comparaison entre le programme et la réalité pour des structures simples. On travaille avec une structure simple qui ne bascule pas en théorie mais qui en réalité bascule ou est sur le point de basculer (*Image 4*). On mesure la distance entre l'axe du centre de gravité de la structure et l'extrémité la plus proche de sa surface d'appui.

Ensuite l'offset est calculé pour qu'il soit à peine plus grand que la distance *d* de façon à ce que l'axe tombe à l'extérieur de la zone (*Figure 38*). On peut donc obtenir une nouvelle surface d'appui réduite par rapport à la surface initiale et ainsi choisir un facteur *Tol* adéquat avec l'équation de l'offset (*Eq. 1*).

¹¹ Ces inputs sont utilisés pour tous les exemples avec les droxels imprimés en 3D.



5.1. Exemple – A

Le premier exemple A est une sorte de petit abri en forme d'ogive (*Figure 39*). Lorsqu'on entre la structure dans le programme, aucune instabilité n'est détectée lors de la droxelisation proposée par l'algorithme (*Figure 40*) avec les inputs du *Tableau 34*.

Inputs		[Unité]
H	40	[mm]
L	27	[mm]
E	3.5	[mm]
R	4.2	[mm]
B	1.2	[mm]
Cr	4	[mm]
F	40	[%]
A	17	[°]
Depth	34	[mm]
Tol	2.55	[%]

Tableau 34 : Inputs de l'exemple A

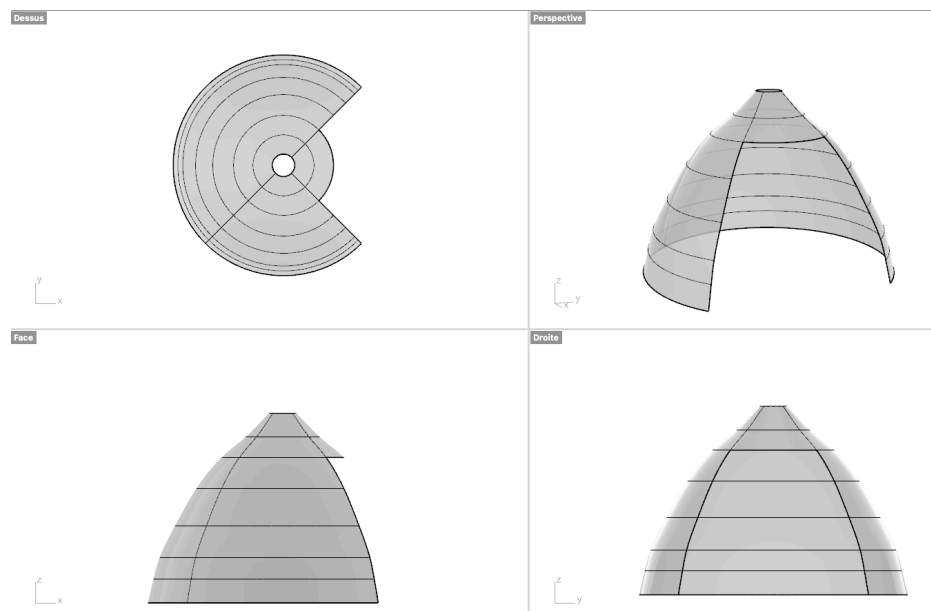


Figure 39 : Structure de l'exemple A

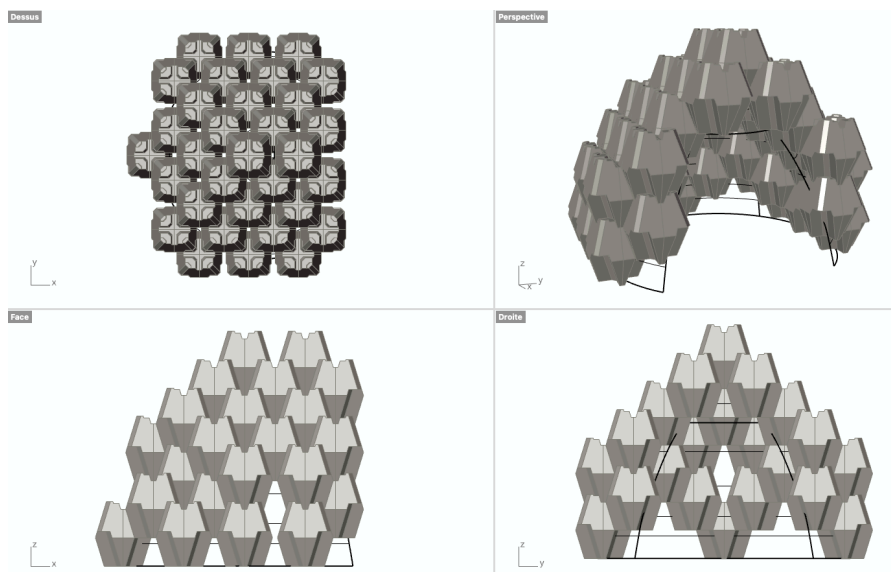


Figure 40 : Droixelisation de l'exemple A

On compare cette structure de droxels à la réalité en se servant des droxels imprimés en 3D à échelle réduite. On peut suivre couche par couche la droxelisation déterminée par l'algorithme étant donné que le programme n'a pas trouvé de sous-structure instable (*Figure 41*).

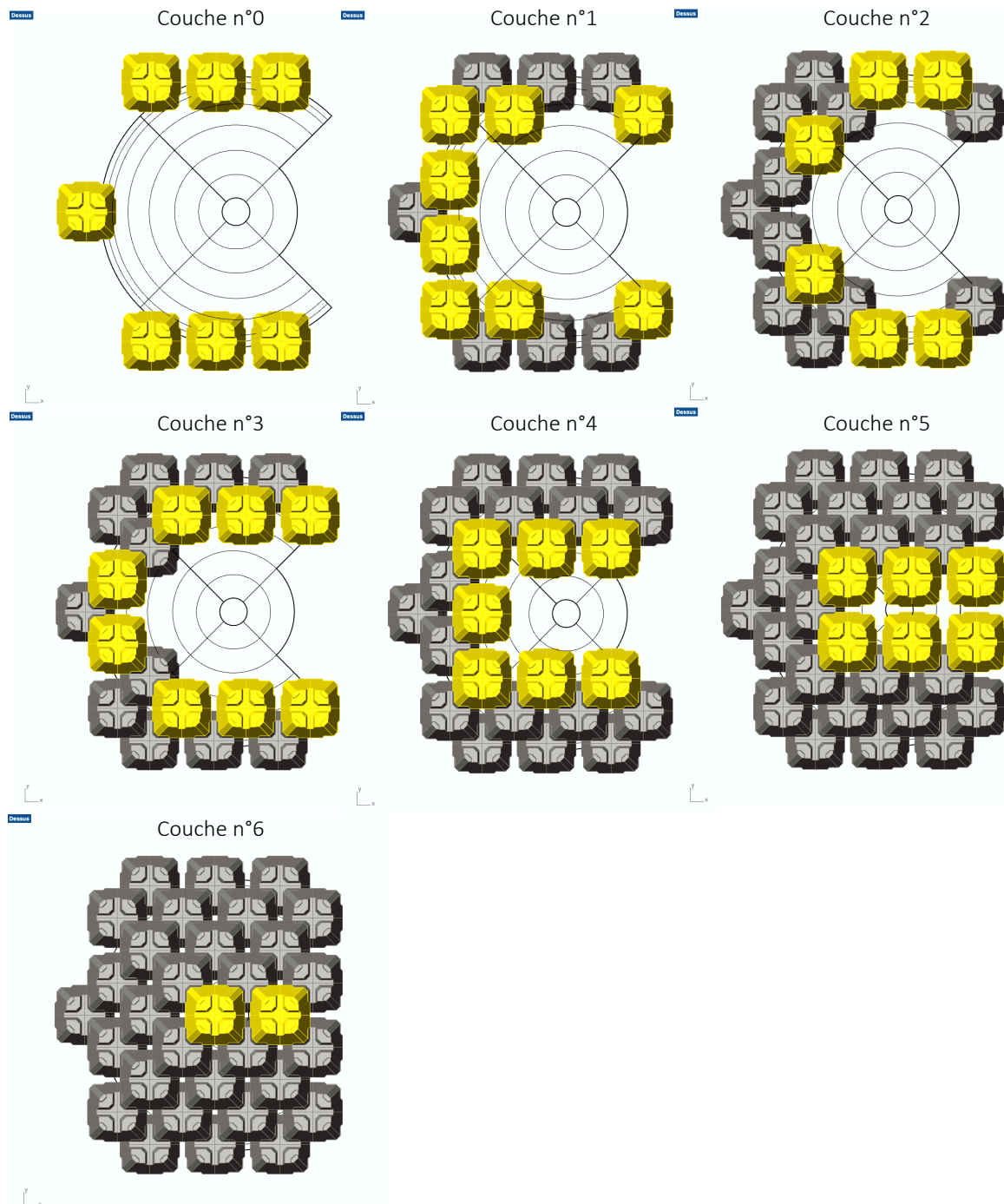


Figure 41 : Modélisation 3D de la séquence de pose de l'exemple A

On reproduit les séquences de pose du programme sur le plateau d'essai avec les droxels imprimés en 3D (*Image 5*).

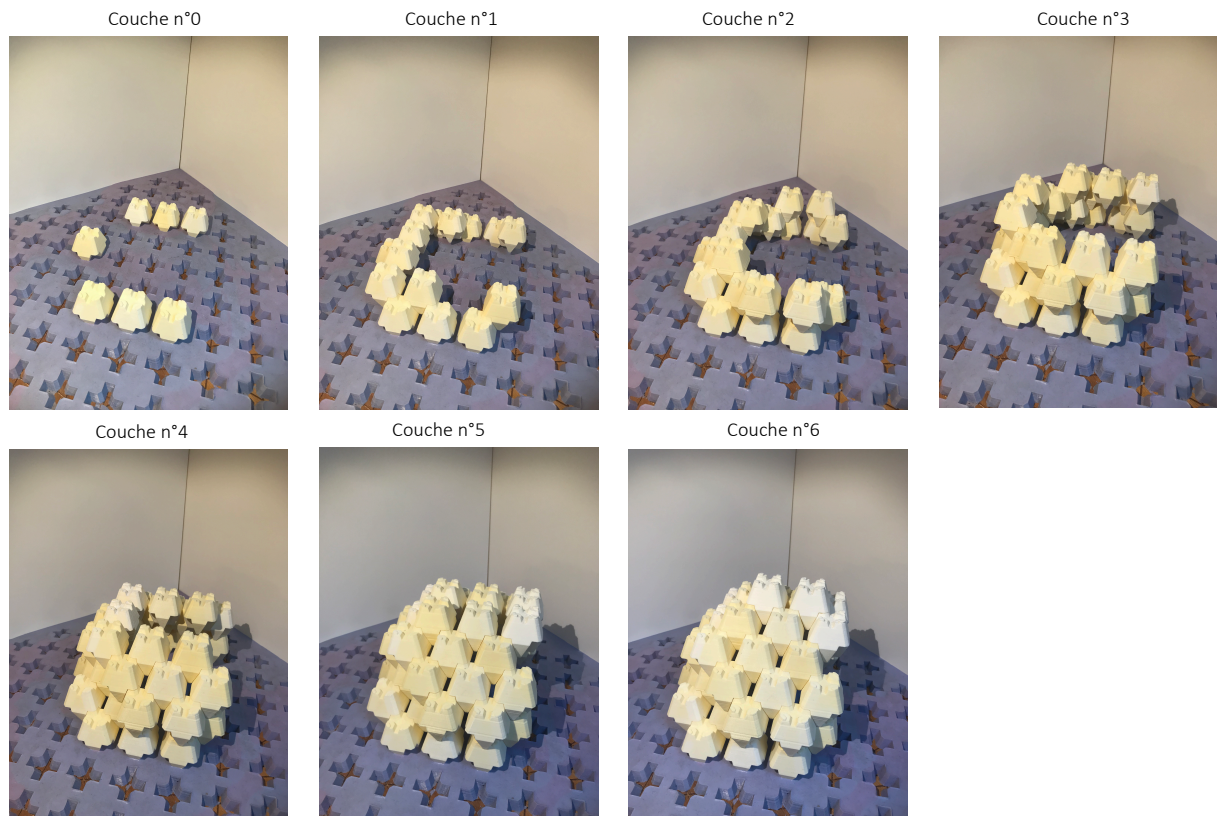


Image 5 : Séquence de pose de l'exemple A

Pour cet exemple, il apparaît que la structure est stable à tout moment lors de la mise en place des droxels et qu'aucune charge externe, mis à part le poids des droxels, n'est appliquée sur le système (*Image 6*).



Image 6 : Vue de l'exemple A droxelisé avec les blocs imprimés en 3D

5.2. Exemple – B

L'exemple B est une structure style préau en porte-à-faux modélisée à la *Figure 42*. Les paramètres d'entrée sont identiques à ceux de l'exemple A car les mêmes droxels sont utilisés. Cependant on décide de diminuer la *Depth* à 38 mm (*Tableau 35*). Le programme Grasshopper droxélise la structure couche par couche et analyse la stabilité globale à chaque étape. L'algorithme détecte deux premières sous-structures instables à l'ajout de la 5^{ème} couche (*Figure 45*). On va effectuer la même séquence de pose (*Figure 44*) avec les droxels imprimés en 3D pour savoir si cette même sous-structure se détache du système dans la réalité (*Image 7*). Nous nous attendons de surcroît à ne pas pouvoir placer tous les droxels de la droxelisation proposée à la *Figure 43* avant que la structure ne se déséquilibre.

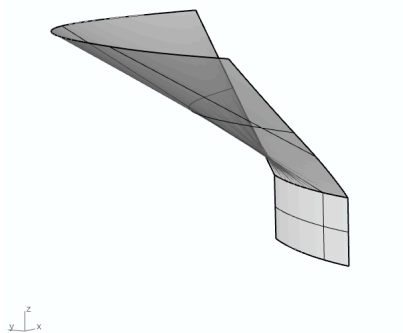


Figure 42 : Structure de l'exemple B

Inputs		[Unité]
H	40	[mm]
L	27	[mm]
E	3.5	[mm]
R	4.2	[mm]
B	1.2	[mm]
Cr	4	[mm]
F	40	[%]
A	17	[°]
Depth	28	[mm]
Tol	2.55	[%]

Tableau 35 : Inputs de l'exemple B

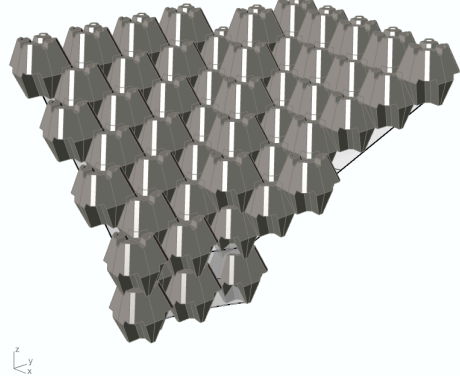


Figure 43 : Droxelisation de l'exemple B

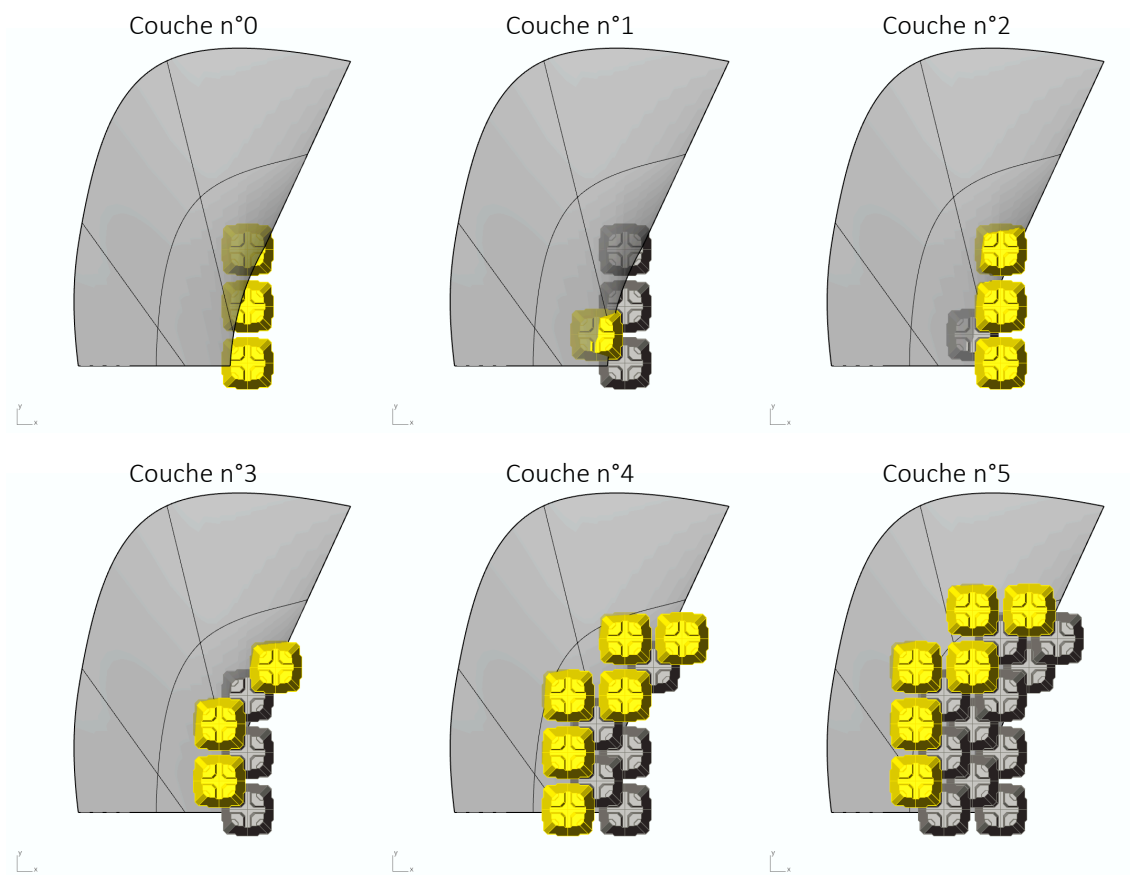


Figure 44 : Modélisation 3D de la séquence de pose de l'exemple B

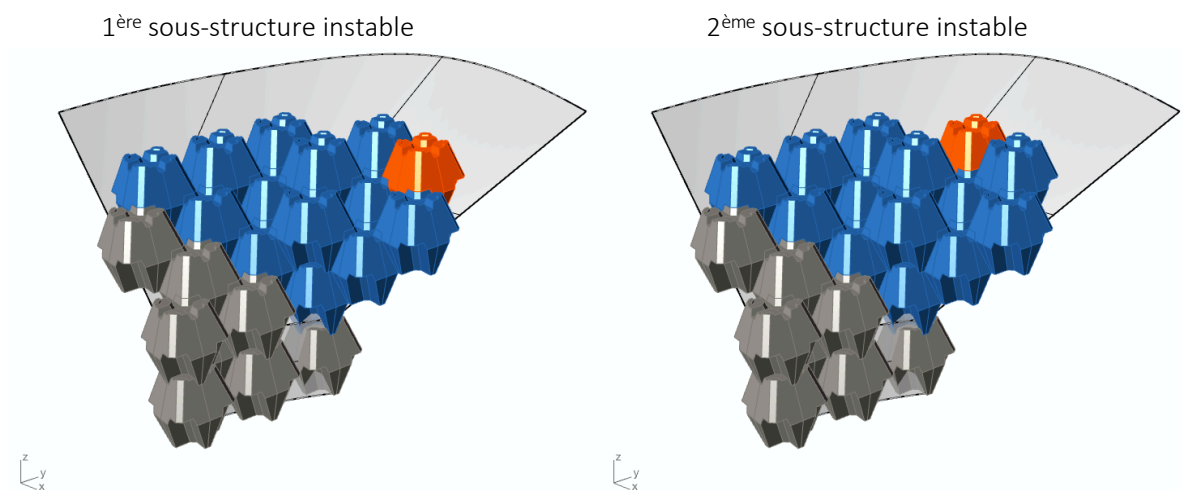


Figure 45 : Sous-structures instables de l'exemple B détectées par l'algorithme

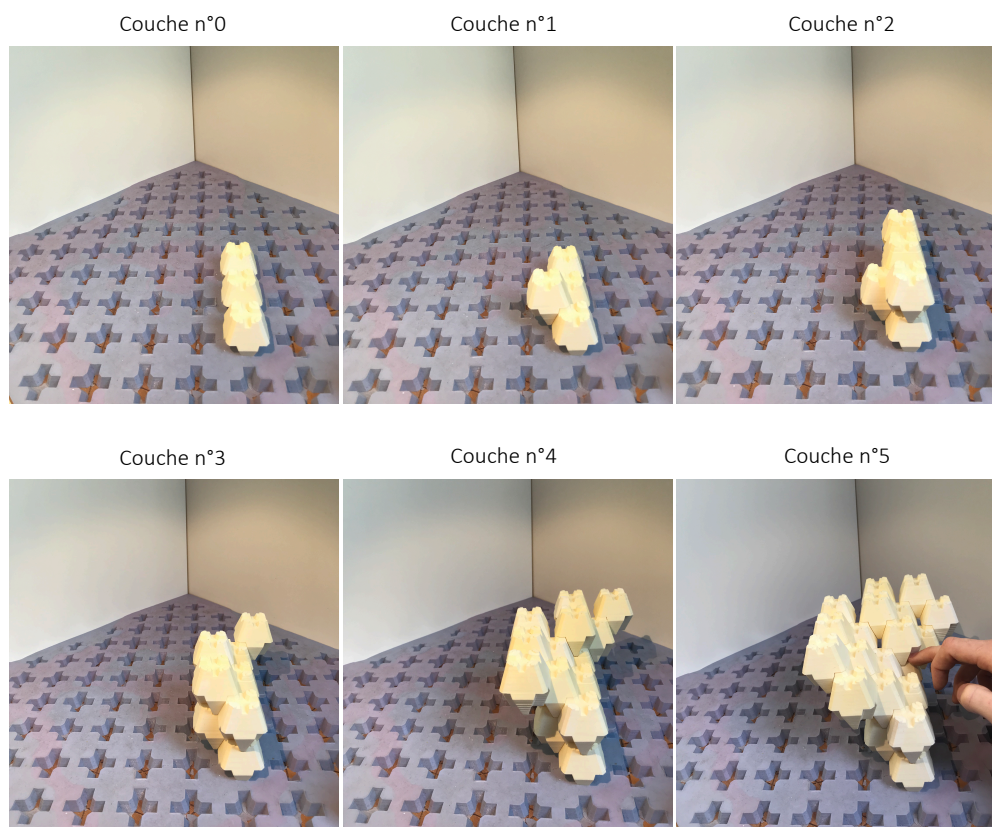


Image 7 : Séquence de pose de l'exemple B

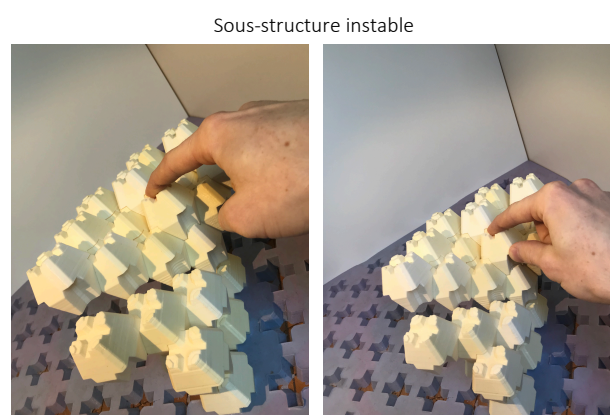


Image 8 : Instabilité de l'exemple B

Sur le plateau d'essai on constate un basculement du même ensemble de droxels (Image 8) lors de la pose des deux droxels en orange mis en évidence par l'algorithme à la Figure 45. Cette proposition de droxelisation n'assure pas la stabilité structurelle à chaque étape.

Les résultats principaux des différents processus sont représentés dans les figures reprises dans les pages 56 et 57 selon l'ordre hiérarchique des clusters du code Grasshopper.

Les Base points de la couche n°0 sont calculés à l'aide du cluster 011 – Grid (Figure 46).

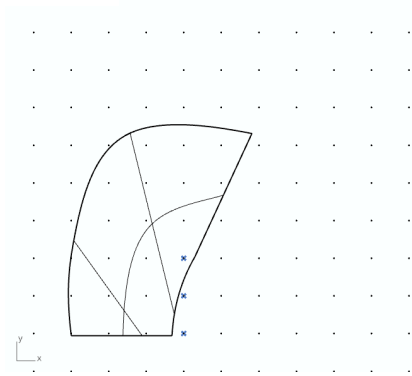


Figure 46 : Base points pour la couche n°0 de l'exemple B

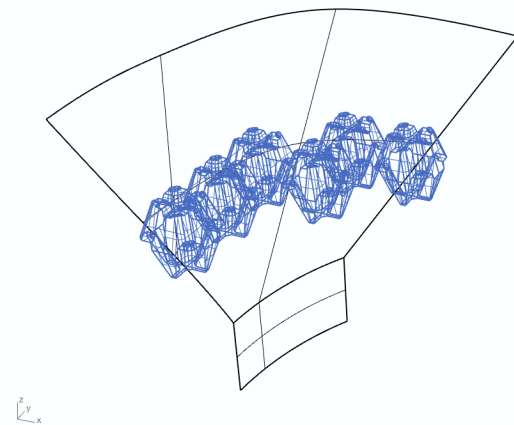


Figure 47 : Droxels de la couche n°4 de l'exemple B

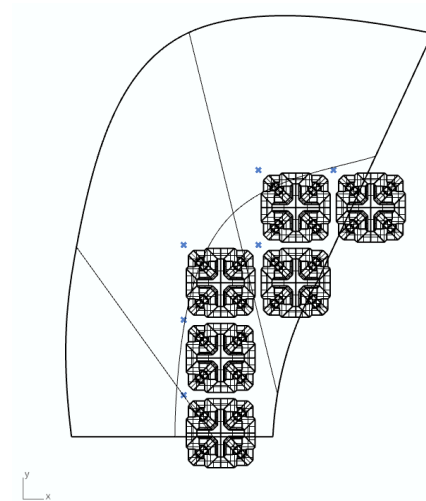
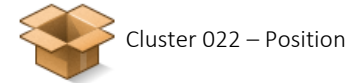


Figure 48 : Base points de la couche n°5 de l'exemple B

Pour la suite de l'exemple on se focalise sur le 5^{ème} Base point de la couche n°5 car il est à l'origine d'une instabilité (Figure 49).

Une fois dans le cluster **0231 – Stability Check**, le programme compile les dix cluster 0231 – Step Sub successifs. La figure (Figure 50) illustre les trois premiers d'entre eux. Il construit petit à petit la sous-structure générée par le Base point. Lorsque la sous-structure n'est pas attachée, son statut est défini par le cluster **02315 – Gravity Test** pour savoir si celle-ci est en équilibre ou non. L'axe de son centre de gravité est à l'extérieur de sa zone d'appui donc son statut devient « Unstable » (Figure 51).

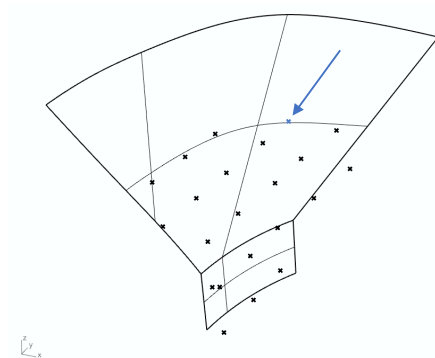


Figure 49 : Base point n°5 de l'exemple B

6. Droxels de changement de pente

Lorsqu'on empile des droxels de mêmes dimensions, ils adoptent une pente unique. Ce qui est restreignant pour la droxelisation de quelle que structure en trois dimensions que ce soit. En effet, si la pente de la structure est différente de la pente que forment les droxels choisis ou si la pente varie dans l'espace comme, par exemple dans le cas d'un dôme ou d'une arche, les droxels n'épouseront pas au mieux la géométrie de la construction souhaitée avec ce genre de blocs. On illustre le concept des pentes uniques que prennent des droxels de [mode 2.a](#) dans la *Figure 52* où on constate que les droxels gris forment une pente α par rapport à l'horizontale qui est plus raide que la pente β des droxels bleus.

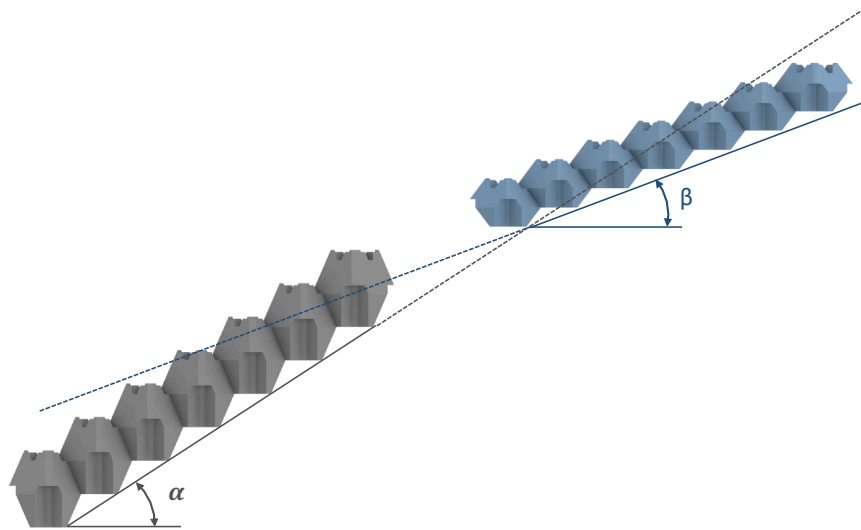


Figure 52 : Pente des droxels de mode 2.a

En respectant une condition, expliquée plus loin dans le texte, il est possible d'empiler des droxels de géométries différentes. Cette condition concerne les paramètres géométriques des droxels supérieurs car ceux-ci déterminent la position du Base point et donc du bloc. Il est aussi nécessaire de construire une couche de transition entre les anciens et les nouveaux droxels (*Figure 53*).

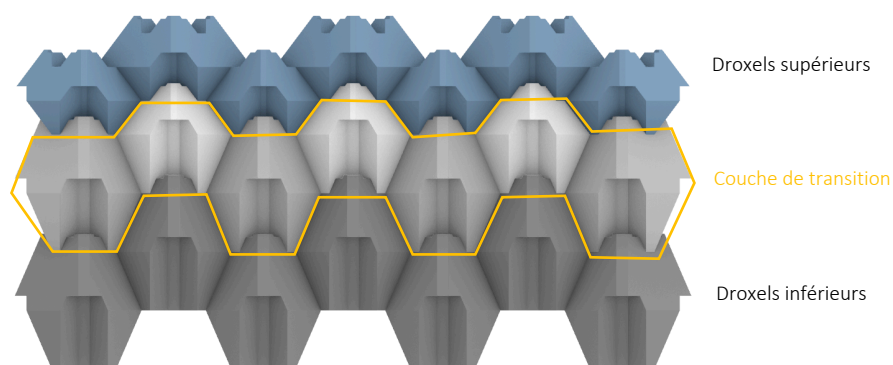


Figure 53 : Couche de transition

Il est indispensable que les Base points des nouveaux droxels respectent le même espacement dans le plan horizontal que ceux des droxels déjà existants. Tous les droxels d'une colonne dans la *Figure 53* ont des Base points de mêmes coordonnées en x et y. Les Base points de chaque type de droxel forment un

quadrillage dans un plan horizontal, en ce compris ceux des droxels supérieurs et inférieurs. Tous ces quadrillages sont identiques. A l'aide de la vue « du dessus » (Figure 54) du mur de la Figure 53, on va expliquer la condition de changement de pente mentionnée plus haut.

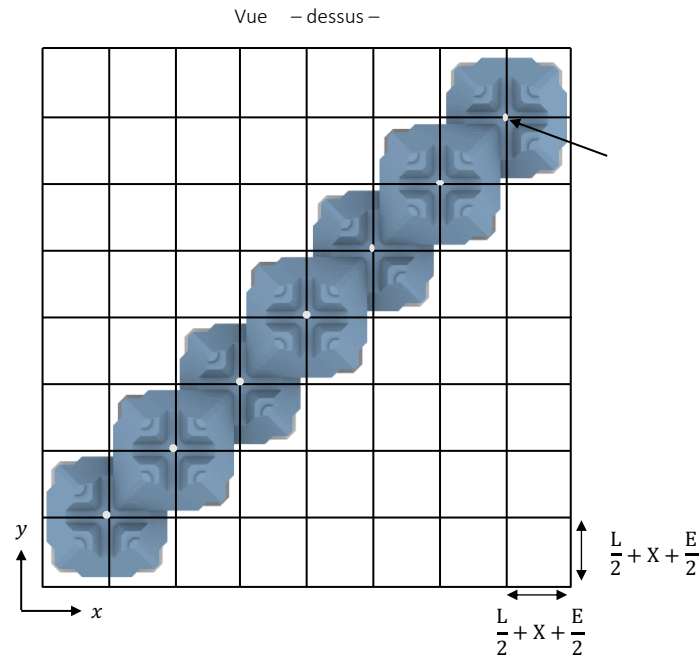


Figure 54 : Condition de changement de pente

On utilisera l'indice *inf* pour les paramètres des droxels inférieurs et *sup* pour les supérieurs. Les cellules qui composent la grille des droxels inférieurs sont de côté $\frac{L_{inf}}{2} + X_{inf} + \frac{E_{inf}}{2}$ et $\frac{L_{sup}}{2} + X_{sup} + \frac{E_{sup}}{2}$.

Dès lors,

$$\frac{L_{inf}}{2} + X_{inf} + \frac{E_{inf}}{2} = \frac{L_{sup}}{2} + X_{sup} + \frac{E_{sup}}{2} \quad (1)$$

car les grilles doivent être identiques.

Ensuite on peut écrire que,

$$L_{inf} + H_{inf} * \tan(A_{inf}) + E_{inf} = L_{sup} + H_{sup} * \tan(A_{sup}) + E_{sup} \quad (2)$$

car $X = H * \frac{\tan(A)}{2}$.

Si les paramètres inférieurs sont posés, la relation (2) devient une équation à quatre inconnues. Pour la résoudre il faut choisir trois paramètres parmi H , L , E et A des droxels supérieurs et le quatrième est défini par l'équation. Il n'y a aucune contrainte concernant R , B , Cr et F des nouveaux droxels.

Une idée pour restreindre ce choix est d'imposer la pente que prennent les nouveaux droxels. La pente pour des droxels empilés en porte-à-faux de [mode 2.a](#) est définie à la figure suivante (Figure 55). Elle est fonction de H , L , E , et A . Ce qui signifie que nous avons deux équations avec quatre inconnues et dès lors plus que deux choix pour ces quatre paramètres.

La couche de transition est composée de droxels dits de « pré-transition » et de « transition ». En effet, si au-delà d'une certaine couche nous voulons changer le type de droxel dans le but d'affiner la droxelisation, ces nouveaux droxels doivent aussi bénéficier des modes de pose en lien avec les couches précédentes (Figure 56). Le droxel de transition permet de placer des droxels supérieurs en porte-à-faux. On construit des excroissances pour qu'elles s'adaptent aux encoches des nouveaux blocs. On modifie également la croix supérieure pour y placer un nouveau droxel. Mais il faut également que la croix inférieure soit identique à celles des anciens droxels. Le droxel de pré-transition est obligatoire pour que les nouveaux droxels puissent être posés à n'importe quel endroit dans la nouvelle couche (Figure 56). Grâce à cette couche de transition, les droxels inférieurs ne sont pas en contact avec les droxels supérieurs, ce qui permet

positionner les droxels avec la nouvelle géométrie à la position voulue. Cette couche transitoire est justifiée car une fois construite, tous les nouveaux blocs des couches suivantes peuvent de nouveau avoir un gabarit identique.

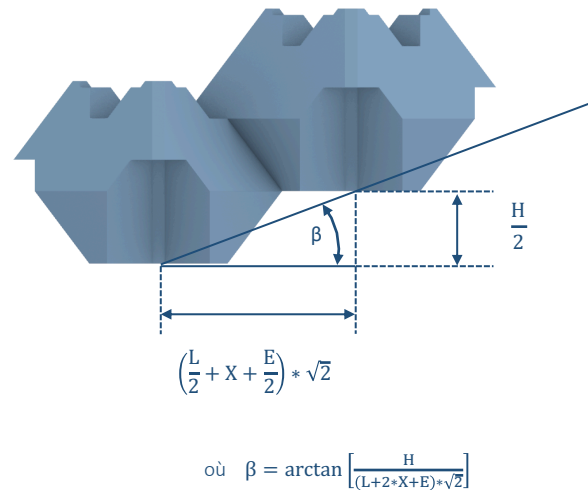


Figure 55 : Pente des droxels de mode 2.a

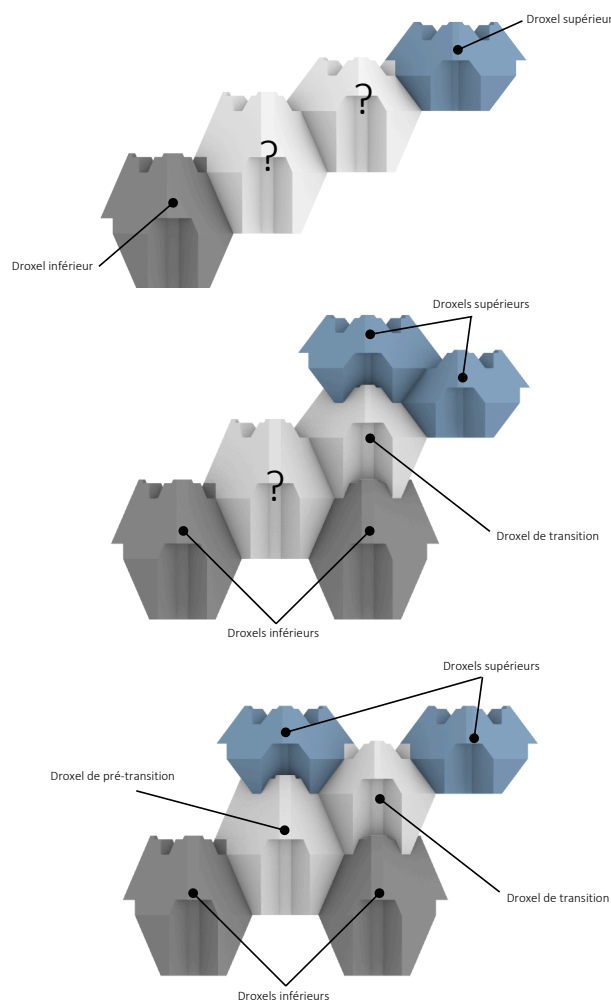


Figure 56 : Droxel de "pré-transition" et de "transition"

6.1. Code Grasshopper pour la couche de transition

Il est important de préciser que le code Grasshopper pour constituer la couche de transition est indépendant de l'algorithme précédent. Le changement de pente ne fait pas partie du code de la droxelisation. Pour améliorer la droxelisation, il serait judicieux d'intégrer une version du programme ci-dessous.

Le code Grasshopper (*Figure 58*) permet de réaliser l'exemple de la *Figure 57*. Il est composé du cluster **021 – Droxel** pour construire les droxels « normaux » (le supérieur et l'inférieur), mais aussi d'un cluster **Pre-transition Droxel** pour construire le droxel de pré-transition et d'un cluster **Transition Droxel** pour le droxel de transition. Dès qu'un Base point est introduit pour développer le premier droxel (inférieur) les Base points suivants sont fournis par les outputs des clusters.

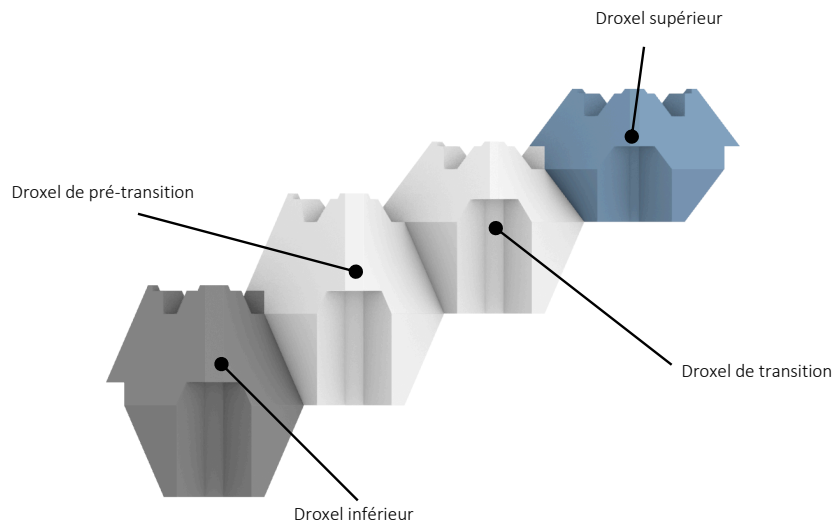


Figure 57 : Visualisation du programme de la couche de transition

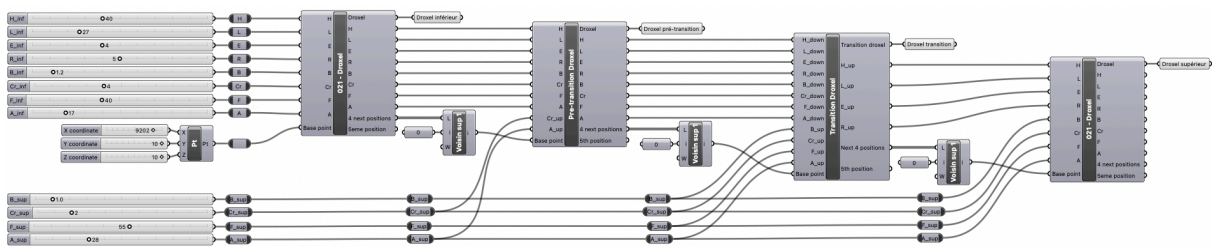


Figure 58 : Programme Grasshopper de la couche transitoire

On choisit librement les paramètres géométriques du droxel inférieur (*Figure 57*). Concernant les paramètres du droxel supérieur, on impose de façon arbitraire que $L_{sup}=L_{inf}$, $E_{sup}=E_{inf}$ et $R_{sup}=R_{inf}$. Il reste donc à définir B_{sup} , Cr_{sup} , F_{sup} et pour finir soit A_{sup} soit H_{sup} . La relation (2) nous impose de choisir trois des paramètres parmi H , L , E et A . On décide de conditionner H (*Figure 59*).

Le paramètre H est calculé dans le cluster *Transition DroxeL* grâce à la relation (2),

$$L_{inf} + H_{inf} * \tan(A_{inf}) + E_{inf} = L_{sup} + H_{sup} * \tan(A_{sup}) + E_{sup} \quad (2)$$

$$L_{inf} + H_{inf} * \tan(A_{inf}) + E_{inf} = L_{sup} + H_{sup} * \tan(A_{sup}) + E_{sup} \quad (3)$$

$$H_{sup} = \frac{H_{inf} * \tan(A_{inf})}{\tan(A_{sup})} \quad (4)$$

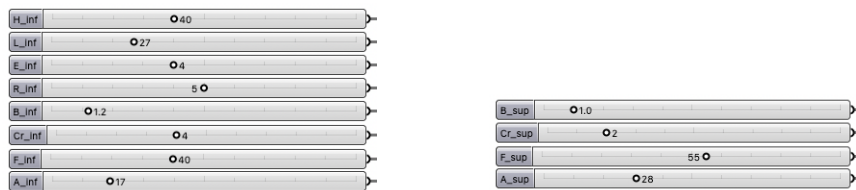


Figure 59 : Paramètres du droxel inférieur et supérieur

6.1.1. Cluster Pre-transition Droxel

Le cluster **Pre-transition Droxel** fonctionne de la même manière que le cluster **021 – Droxel**. Il calcule les coordonnées des points de contrôle des surfaces délimitant un droxel. Le droxel de pré-transition est similaire au droxel inférieur à l'exception de la croix supérieure qui doit s'adapter au droxel supérieur. Les coordonnées en x et en y des points de la croix supérieure doivent être recalculés avec les paramètres du droxel supérieur (Figure 60).

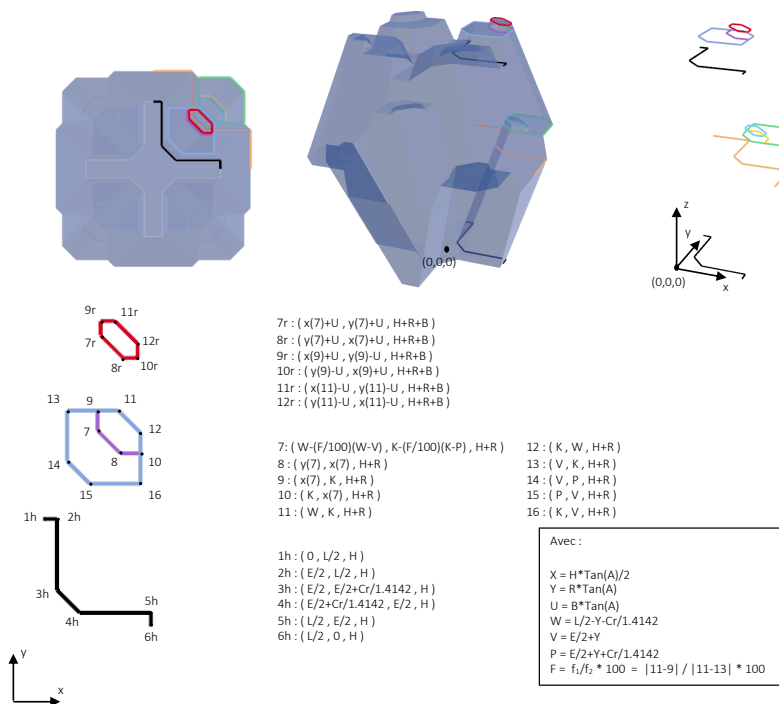


Figure 60 : Points de contrôle pour le droxel de pré-transition

Étant donné que $L_{sup}=L_{inf}$, $E_{sup}=E_{inf}$ et $R_{sup}=R_{inf}$ dans notre exemple, seuls les points 2h, 3h, 4h, 5h, 13, 14, 15 et 16 sont déterminés par les paramètres des droxels supérieurs (Figure 61).

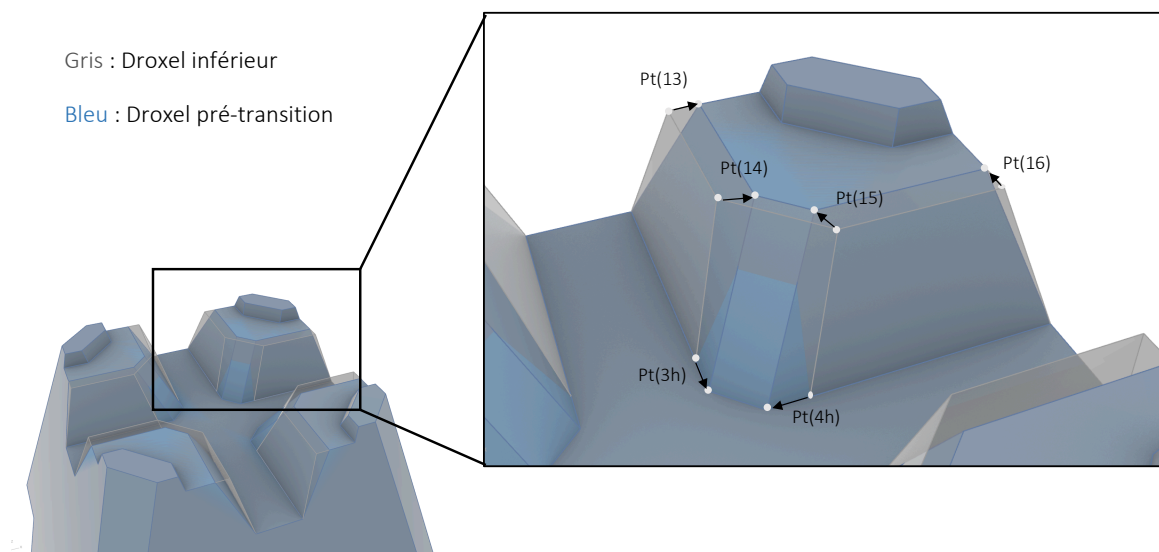


Figure 61 : Points de contrôle à modifier pour le droxel de pré-transition

Inputs	[Unité] / type d'objet Gh	Description
H	[Scale]	Paramètre géométrique du droxel inférieur.
L	[Scale]	Paramètre géométrique du droxel inférieur.
E	[Scale]	Paramètre géométrique du droxel inférieur.
R	[Scale]	Paramètre géométrique du droxel inférieur.
B	[Scale]	Paramètre géométrique du droxel inférieur.
Cr	[Scale]	Paramètre géométrique du droxel inférieur.
F	[%]	Paramètre géométrique du droxel inférieur.
A	[°]	Paramètre géométrique du droxel inférieur.
Cr_up	[Scale]	Paramètre géométrique du droxel supérieur.
A_up	[°]	Paramètre géométrique du droxel supérieur.
Base point	Point	Base point pour y développer le droxel de pré-transition.

Tableau 36 : Inputs du cluster Pre-transition Droxel

Outputs	[Unité] / type d'objet Gh	Description
Droxel	Brep	Droxels de pré-transition.
H	[Scale]	Paramètre géométrique du droxel inférieur.
L	[Scale]	Paramètre géométrique du droxel inférieur.
E	[Scale]	Paramètre géométrique du droxel inférieur.
R	[Scale]	Paramètre géométrique du droxel inférieur.
B	[Scale]	Paramètre géométrique du droxel inférieur.
Cr	[Scale]	Paramètre géométrique du droxel inférieur.
F	[%]	Paramètre géométrique du droxel inférieur.
A	[°]	Paramètre géométrique du droxel inférieur.
4 next positions	List of points	Positions des voisins supérieurs 1, 2, 3 et 4 du droxel de pré-transition.
5th position	List of points	Positions du voisin supérieur 5 du droxel de pré-transition.

Tableau 37 : Outputs du cluster Pre-transition Droxel

6.1.2. Cluster Transition Droxel

Le cluster **Transition Droxel** construit le droxel de transition par assemblage (Figure 64) avec la moitié supérieure du nouveau droxel (bleu) et la moitié inférieure de l'ancien droxel (gris). Le processus de construction est similaire au cluster **021 – Droxel**. Le cluster développe les deux moitiés avec les paramètres des droxels inférieurs et supérieurs (Figure 63) et ensuite assemble les deux à l'aide du composant Grasshopper « Brep Join » (Figure 62).



Figure 62 : Composant Grasshopper « Brep Join »

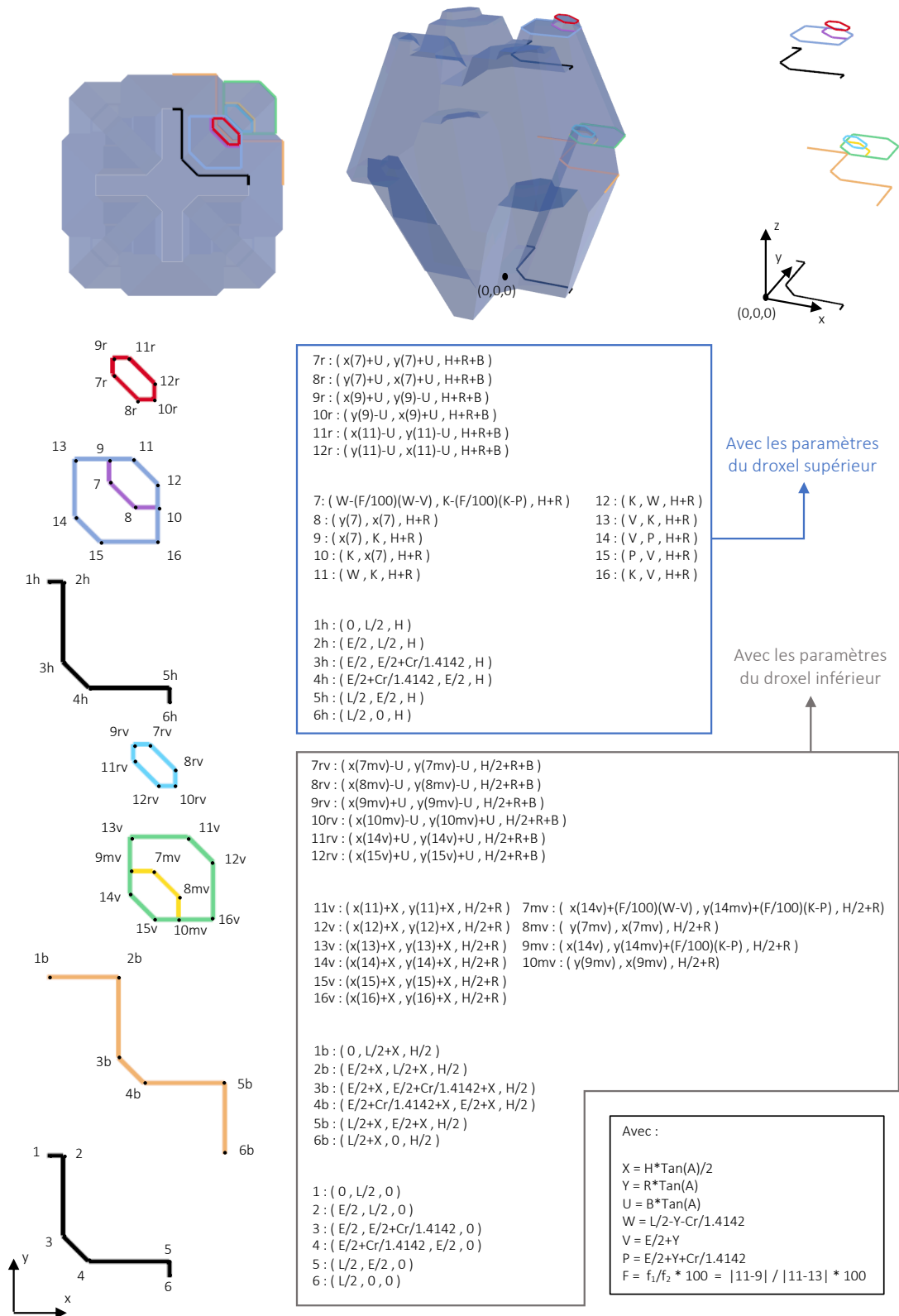


Figure 63 : Points de contrôle pour le droxel de transition

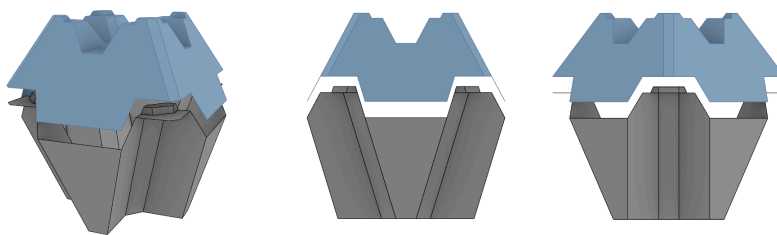


Figure 64 : Assemblage du droxel de transition

Inputs	[Unité] / type d'objet Gh	Description
H_down	[Scale]	Paramètre géométrique du droxel inférieur.
L_down	[Scale]	Paramètre géométrique du droxel inférieur.
E_down	[Scale]	Paramètre géométrique du droxel inférieur.
R_down	[Scale]	Paramètre géométrique du droxel inférieur.
B_down	[Scale]	Paramètre géométrique du droxel inférieur.
Cr_down	[Scale]	Paramètre géométrique du droxel inférieur.
F_down	[%]	Paramètre géométrique du droxel inférieur.
A_down	[°]	Paramètre géométrique du droxel inférieur.
B_up	[Scale]	Paramètre géométrique du droxel supérieur.
Cr_up	[Scale]	Paramètre géométrique du droxel supérieur.
F_up	[%]	Paramètre géométrique du droxel supérieur.
A_up	[°]	Paramètre géométrique du droxel supérieur.
Base point	Point	Base point pour y développer le droxel de transition.

Tableau 38 : Inputs du cluster Transition Drexel

Outputs	[Unité] / type d'objet Gh	Description
Transition Drexel	Brep	Droxels de transition.
H_up	[Scale]	Paramètre géométrique du droxel inférieur.
L_up	[Scale]	Paramètre géométrique du droxel inférieur.
E_up	[Scale]	Paramètre géométrique du droxel inférieur.
R_up	[Scale]	Paramètre géométrique du droxel inférieur.
4 next positions	List of points	Positions des voisins supérieurs 1, 2 ,3 et 4 du droxel de transition.
5th position	List of points	Positions du voisin supérieur 5 du droxel de transition.

Tableau 39 : Outputs du cluster Transition Drexel

7. Pistes d'amélioration du code Grasshopper

Droxel par droxel

La principale amélioration qu'il faudrait pouvoir apporter au programme serait de droxéliser, non seulement couche par couche, mais droxel par droxel. De cette façon, en fonction de l'analyse de la stabilité globale de l'impact de chaque nouveau droxel, on pourrait déterminer la bonne position de celui-ci. Si on arrive à se défaire de l'hypothèse que tous les droxels d'une couche sont mis en place au même moment, on pourrait proposer un algorithme de séquences de pose de chaque droxel.

Une solution à ce problème serait de coder un script (C# ou Python) pour itérer un processus d'ajout de droxel individuel par couche. Car il est en effet difficile d'itérer des processus en Grasshopper avec des objets géométriques.

Une autre idée serait de créer au préalable une liste de droxels pour ensuite les placer les uns après les autres. Dès que la position d'un nouveau droxel serait trouvée, on analyserait son influence sur la stabilité globale avec le processus des sous-structures avant de passer au droxel suivant. Si la position ne génère aucune instabilité, le nouveau droxel serait accepté. Sinon une solution serait étudiée pour conserver le droxel. Comme par exemple ajouter des contrepoids.

Coefficient de frottement

Une théorie importante sur l'influence de l'angle de frottement des droxels devrait être incluse dans le programme afin de se rapprocher au plus près de la réalité.

Dans ses recherches, le Professeur Pierre Latteur a travaillé sur un aspect de cette théorie dans laquelle il est prouvé qu'il est possible de calculer la distance maximale d'un porte-à-faux qu'un droxel de [mode 2.a](#) peut supporter en fonction du coefficient de frottement des blocs et du facteur d'angle A du droxel.

Dans ce mémoire, l'angle de frottement n'est pas pris en compte. Ce qui signifie que le programme ne détecte pas d'instabilité pour certains systèmes qui sont pourtant instables.

Illustrons l'importance du coefficient de frottement avec les cas repris dans l'*Image 9*.

On applique un peu de magnésie sur les droxels imprimés en 3D pour démontrer, à leur échelle, l'influence du coefficient de frottement. On constate qu'un nombre plus important de droxels peuvent être empilés dans le cas B que dans le cas A car le coefficient de frottement est plus élevé.

Si on compare le cas C au cas D, on constate que le droxel au sommet de l'arche dans le cas C se soulève contrairement à celui du cas D. Les deux moitiés de l'arche s'appuient sur le droxel qui les relie au sommet et dans la situation C, il apparaît que les forces du système (forces de frottement et le poids) ne sont pas en équilibre.

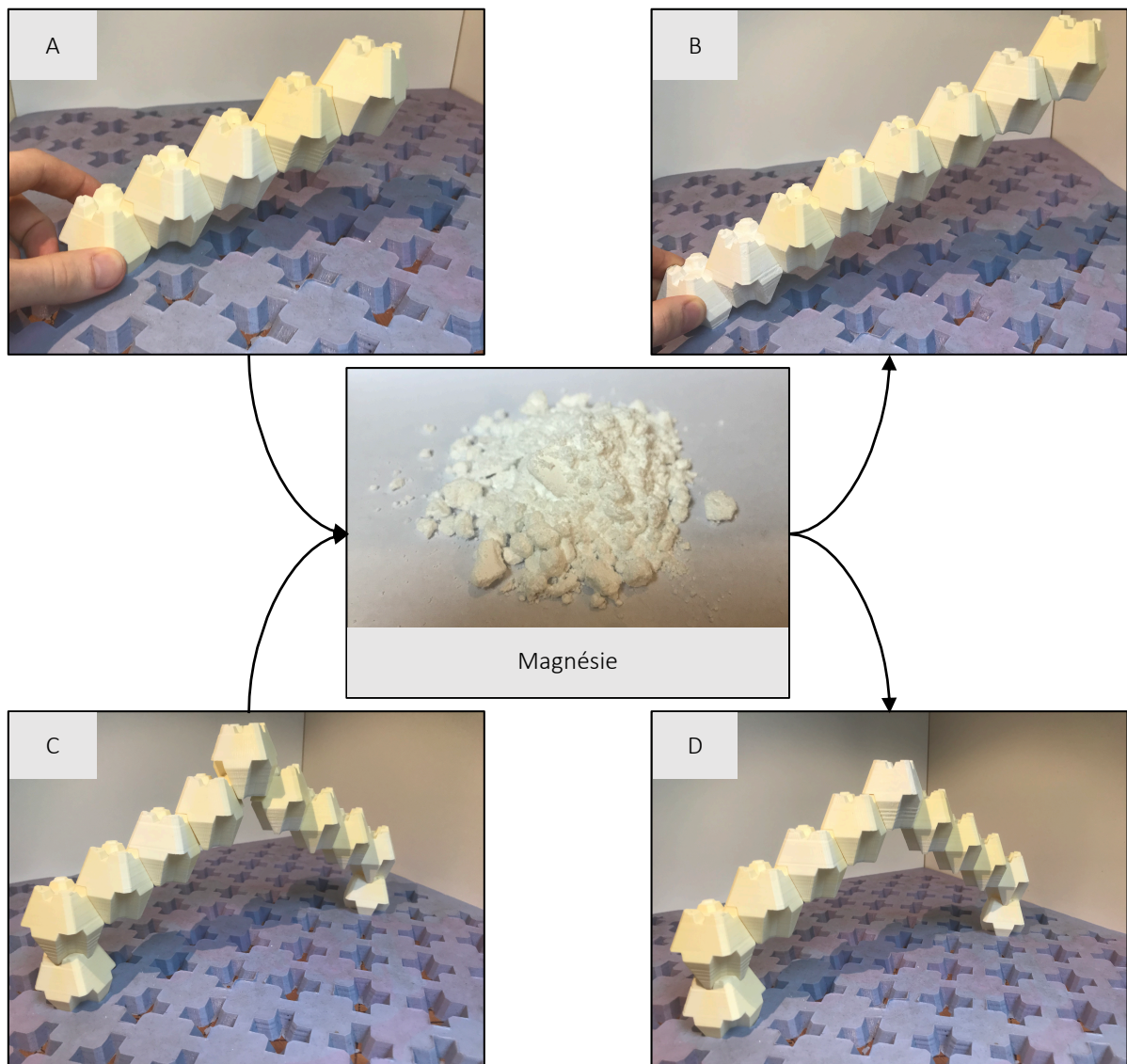


Image 9 : Exemple de coefficients de frottement différents

Pour des structures comme dans les cas A et B, le code Grasshopper ne détectera pas d'instabilité car il ne trouvera pas de sous-structure.

Dans ce mémoire on a émis l'hypothèse que les droxels de [mode 2.a](#) sont accrochés¹². Ces droxels ne peuvent donc pas engendrer de sous-structure.

Une idée pour améliorer l'algorithme serait de considérer qu'un droxel de [mode 2.a](#) fait partie de la catégorie « not hooked ». De cette manière, les sous-ensembles ayant comme élément le plus bas un droxel de [mode 2.a](#) pourraient être qualifiés de sous-structures (*Figure 65*). Une sous-structure de ce type (*Figure 65*) est en porte-à-faux et s'accroche au droxel du bas. C'est alors qu'une zone d'appui virtuelle serait construite pour l'analyse effectuée dans le cluster **02315 – Gravity Center**.

La zone d'appui virtuelle serait limitée par la distance maximale (qui est fonction du coefficient de frottement des droxels) que peut supporter le droxel du bas (*Figure 65*).

Le programme serait dès lors capable de tester si la projection verticale de la résultante de la sous-structure de ce type est stable ou non.

¹² Signifie qu'ils ne peuvent pas se renverser.

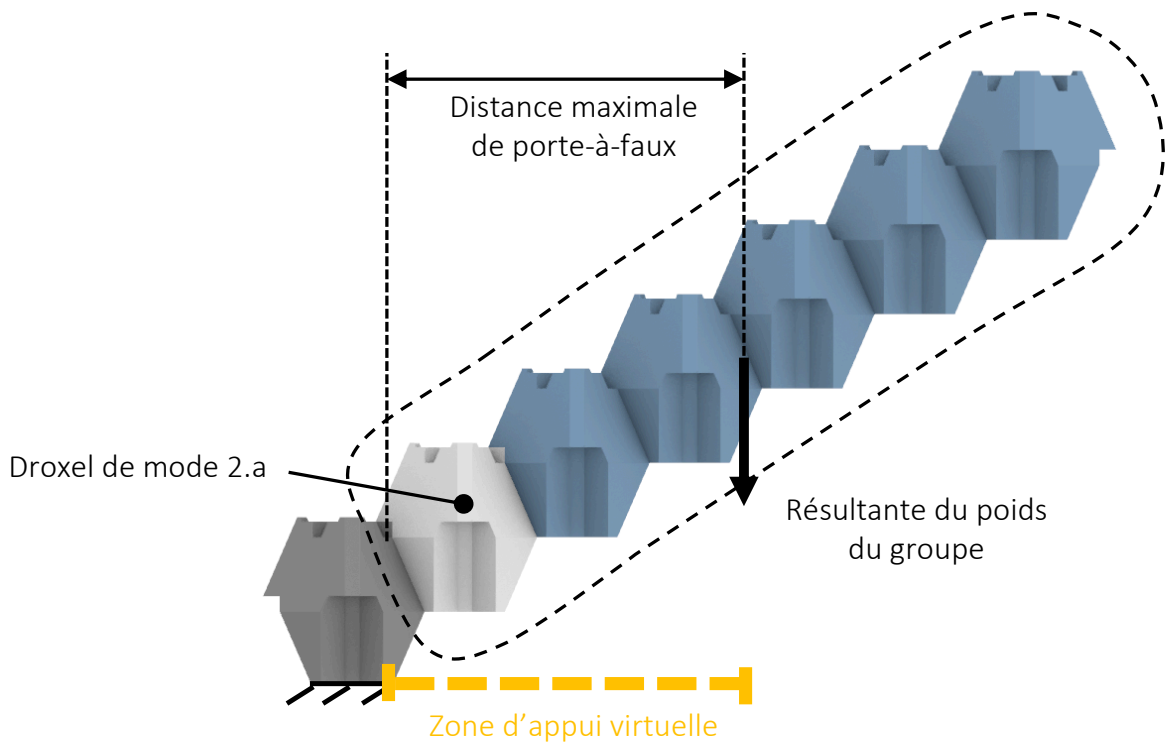


Figure 65 : Piste d'amélioration (1)

Qualifier un droxel de [mode 2.a](#) de « not hooked », en lui construisant une zone d'appui virtuelle comme expliqué ci-dessus, permettrait de résoudre un autre cas. Un cas assez particulier expliqué ci-dessous.

La façon dont un droxel de [mode 2.a](#) est accroché (Figure 4) ne l'empêche pas de basculer dans le sens illustré dans l'Image 10. C'est une situation rare mais en tenir compte améliorerait le code. La zone d'appui virtuelle serait infinie dans la direction – BB – (Figure 5) et opposée à l'accroche du porte-à-faux (Figure 66).

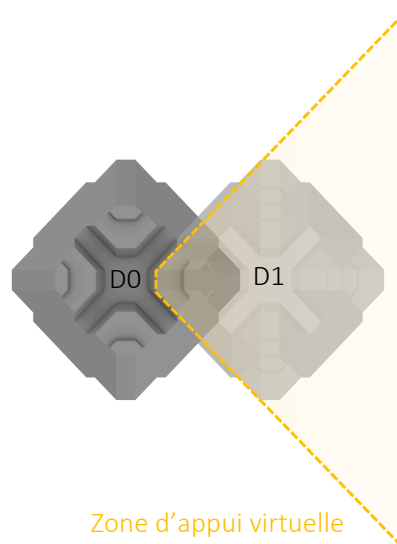


Figure 66 : Piste d'amélioration (2)

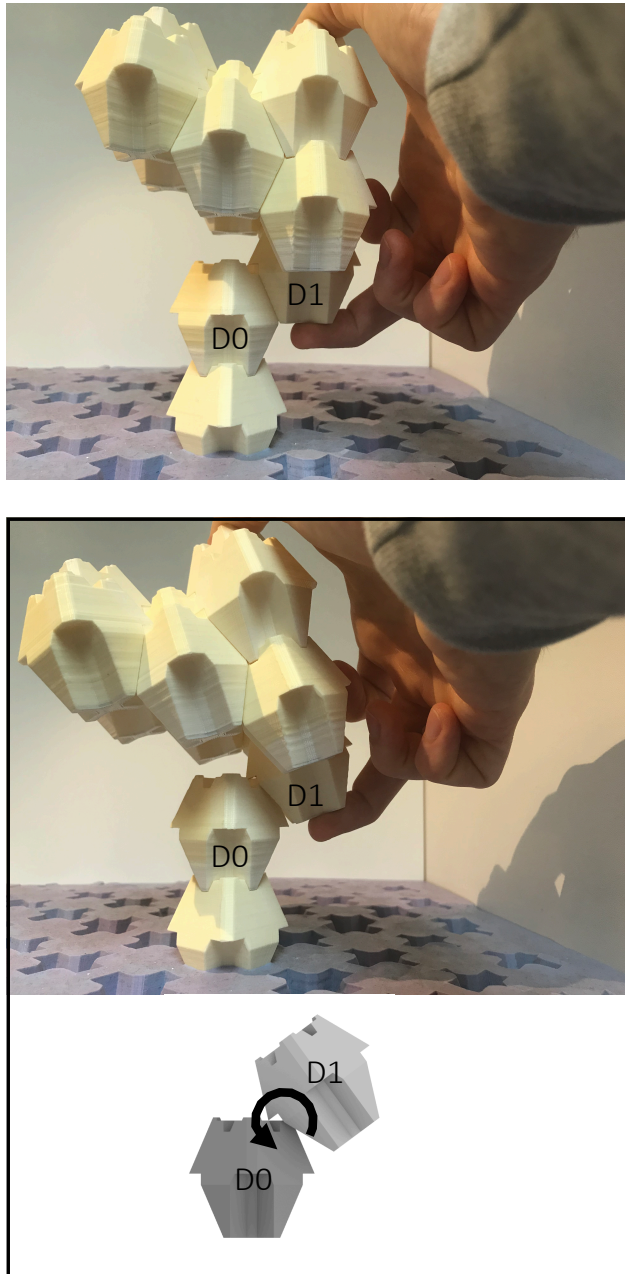


Image 10 : Exemple d'instabilité du mode 2.a

Au final, il serait judicieux de combiner cette zone avec celle définie par la théorie des angles de frottement.

8. Apporter une solution de droxelisation stable

Plusieurs concepts existent pour que l'ajout d'un droxel ne génère pas d'instabilité.

Un des objectifs pour équilibrer une structure est de déplacer le centre de gravité des sous-structures pour que leur projection verticale soit dans la zone d'appui des sous-structures.

- On pourrait placer d'autres droxels temporaires qui empêcheraient le basculement d'une sous-structure en faisant contrepoids. Ces droxels temporaires composeraient une liste indépendante. Et une fois la droxelisation terminée, un autre algorithme serait utilisé pour enlever un maximum de droxels temporaires.
- On pourrait également définir un paramètre de masse pour chaque droxel car en augmentant la masse de certains droxels d'une sous-structure son centre de gravité pourrait être déplacé.

8.1. Galapagos

Une autre idée est d'utiliser le plugin Galapagos de Grasshopper.

Ce plugin est un solveur qui fait varier les inputs d'un processus Grasshopper pour faire tendre son output vers une solution souhaitée.

Avec le programme actuel on pourrait imaginer faire varier le paramètre *Depth* d'une couche à l'autre pour éliminer un maximum de sous-structures instables. On rappelle que le paramètre *Depth* est l'« épaisseur » de la structure.

Le solveur Galapagos a été testé sur le code avec un exemple. On explique son fonctionnement à la section suivante.

8.1.1. Exemple C avec l'interface Galapagos

On teste le solveur Galapagos avec l'exemple C (Figure 67).

Dans un premier temps on compile normalement l'algorithme avec les inputs du Tableau 40. On définit une seule fois le paramètre *Depth*. Ce qui signifie que la distance maximale à laquelle les droxels peuvent se trouver de la structure est la même pour toutes les couches.

Le programme propose une droxelisation (Figure 68) et est susceptible de trouver des sous-structures instables. On constate effectivement que c'est le cas et huit sous-structures en déséquilibre sont détectées. On affiche la visualisation 3D des deux premières sous-structures à la Figure 69 et à la Figure 70. Les six autres sont illustrées dans les annexes. Les droxels de couleur orange sont les droxels responsables du déséquilibre de la sous-structure en bleu dans leur figure respective (Figure 69 et Figure 70).

Inputs		[Unité]
H	40	[mm]
L	27	[mm]
E	3.5	[mm]
R	4.2	[mm]
B	1.2	[mm]
Cr	4	[mm]
F	40	[%]
A	17	[°]
Depth	40	[mm]
Tol	0	[%]

Tableau 40 : Inputs de l'exemple C

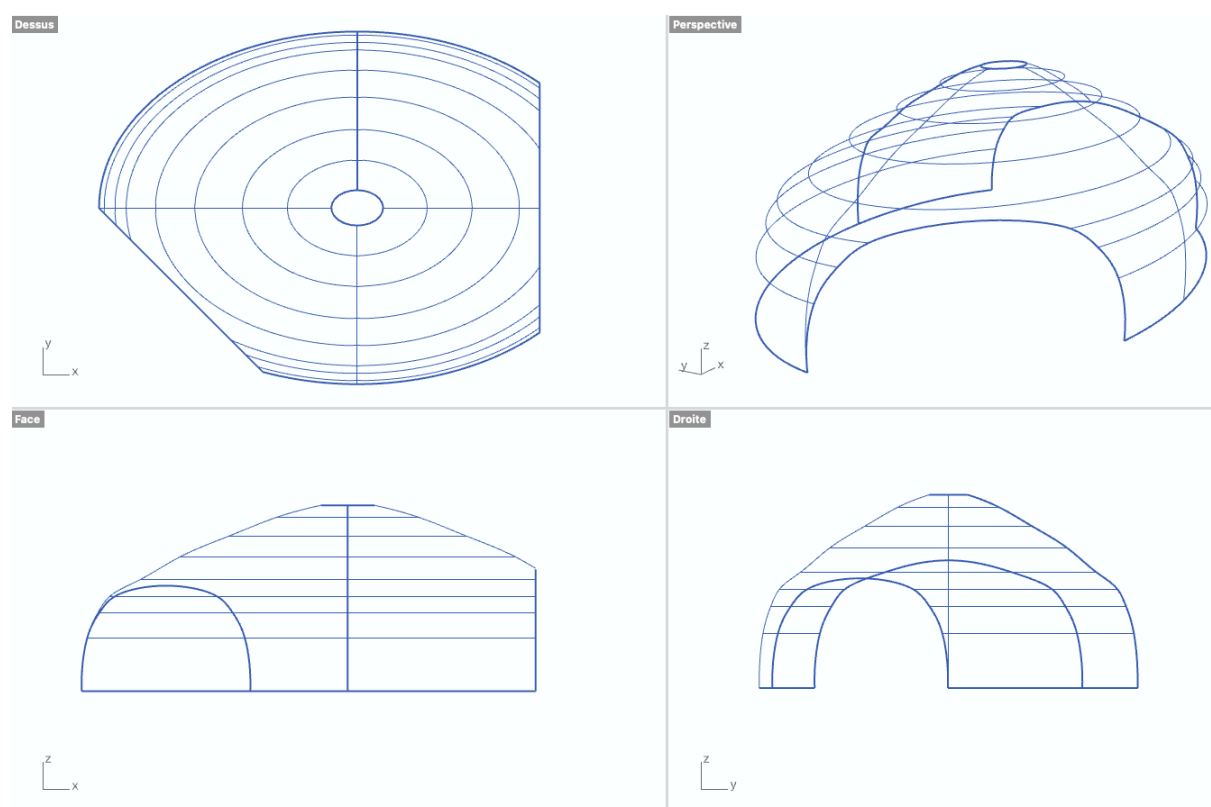


Figure 67 : Structure à droxéliser de l'exemple C

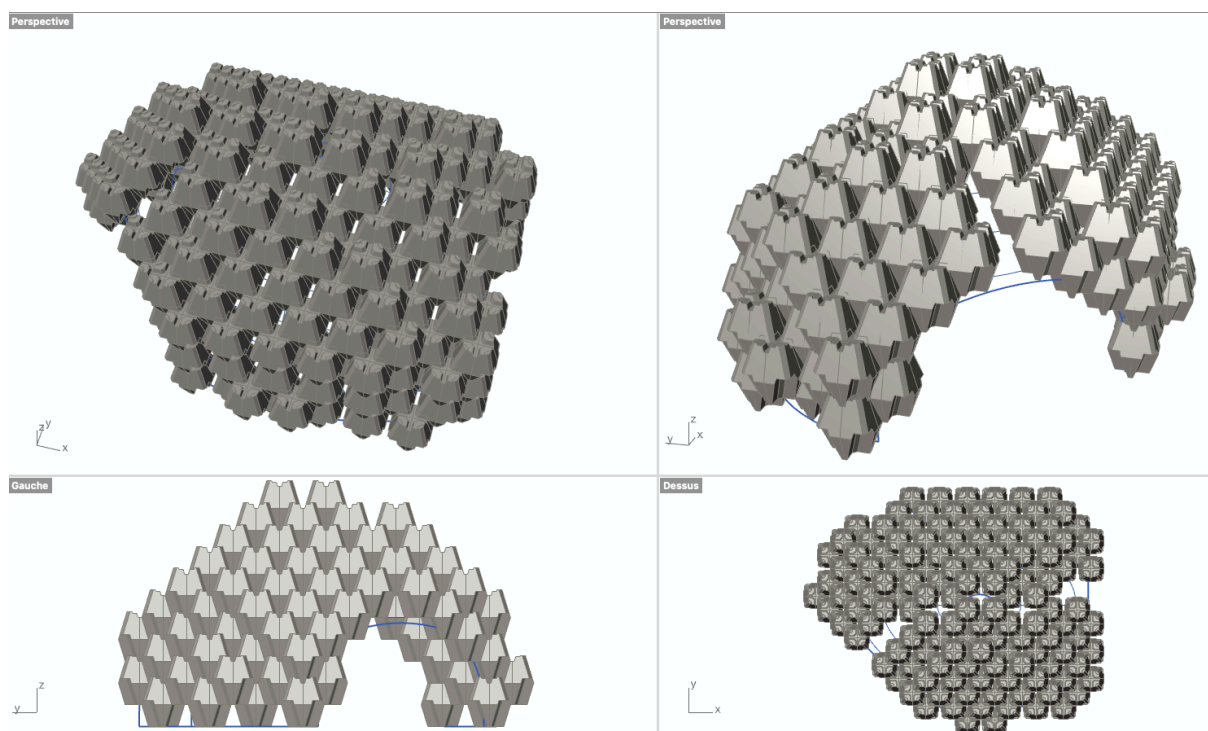


Figure 68 : Structure droxelisée de l'exemple C (1)

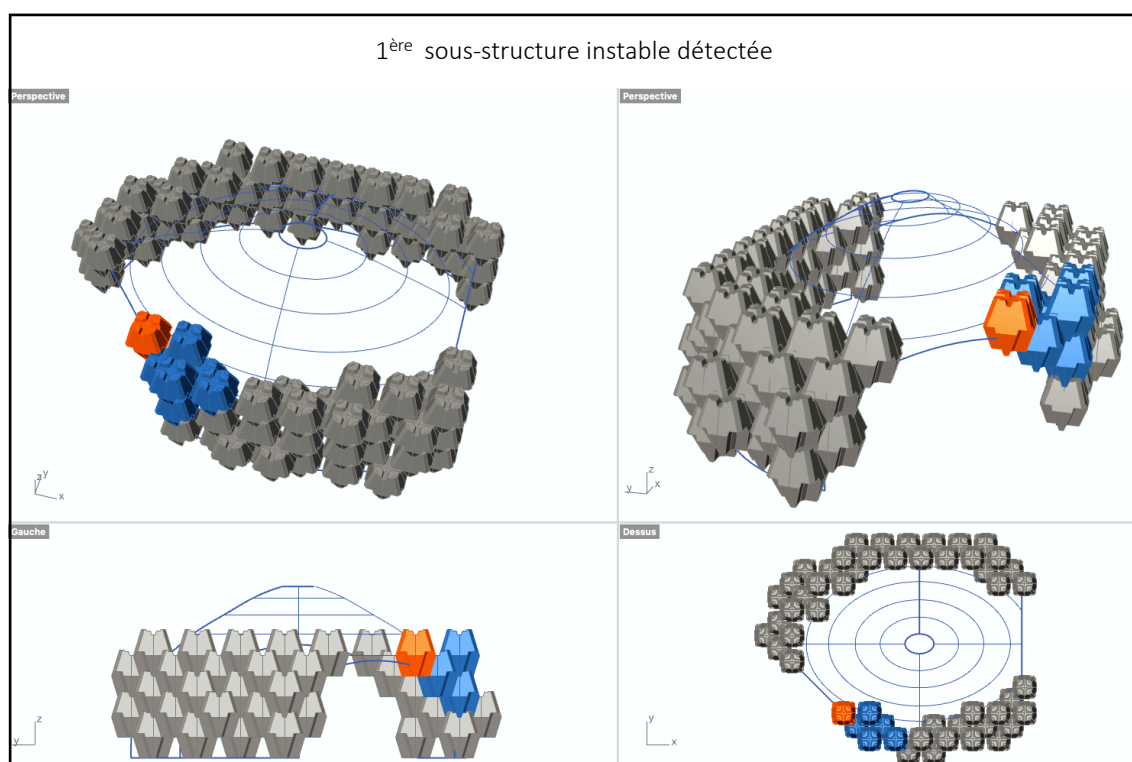


Figure 69 : 1^{ère} sous-structures instables détectées

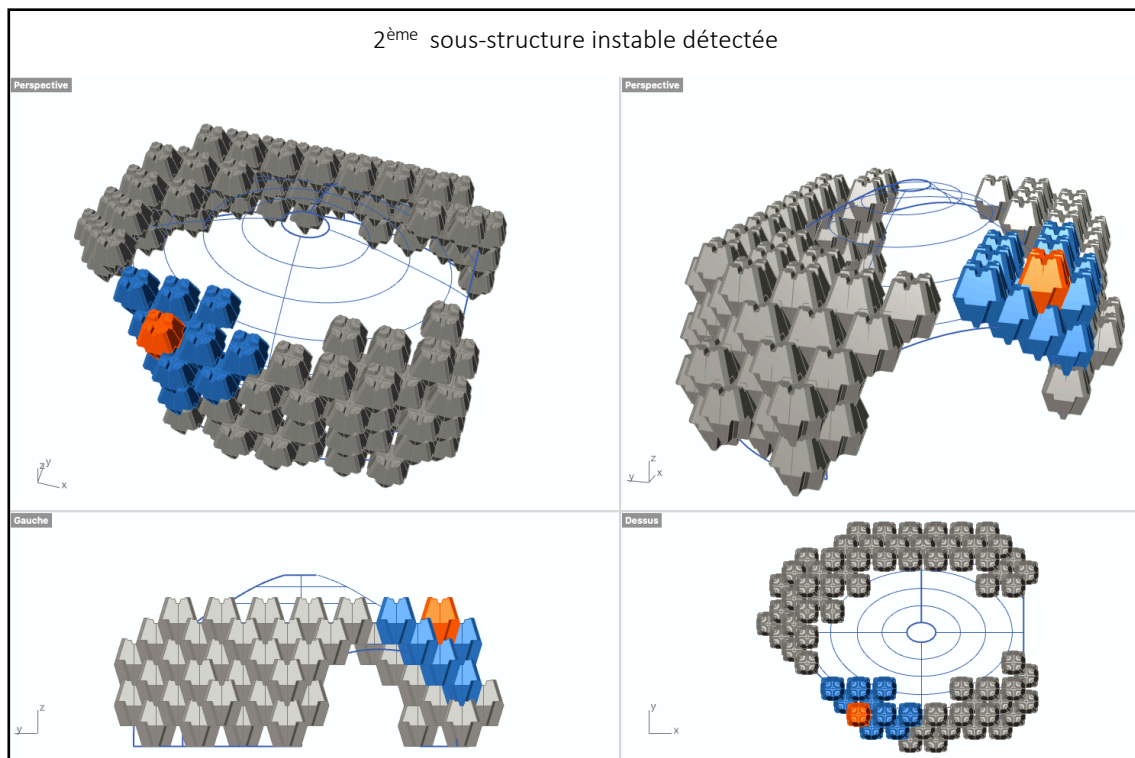


Figure 70 : 2ème sous-structures instables détectées

Il est dès lors utile de faire intervenir le solveur. Au lieu d'imposer le même facteur *Depth* pour toute la structure, on va le laisser « libre ». On crée un curseur pour chaque input *Depth* de chaque couche et on le relie à l'output *Genome* du solveur. Dans le cas présent on définit une valeur à minimiser, c'est-à-dire, le nombre de sous-structures instables, et on relie cette valeur à l'input *Fitness* du solveur (Figure 71).

Les curseurs *Depth* sont définis dans un intervalle :

- Limité par le bas avec une valeur supérieure à zéro pour éviter la solution triviale où tous les curseurs sont nuls.
Sinon cela signifierait que la structure n'a pas d'épaisseur et donc pas de droxel.
- Limité par le haut pour éviter que le solveur ne prenne des valeurs trop élevées.
Sinon cela signifierait remplir complètement l'espace de droxels.

« Galapagos utilise une relation entrée-sortie typique qui est commune à la plupart des solveurs d'algorithmes génétiques multi-objectifs. Une «fonction de fitness» est définie comme l'objectif à résoudre, et les relations de variables paramétriques sont définies comme des outputs. Galapagos définit les relations de sortie comme des génomes. Ces sorties, utilisant Galapagos, sont limitées à la fonction paramétrique du curseur dans l'environnement Grasshopper. La fonction de fitness est une variable unique, où les sorties paramétriques (génomes) peuvent être multi variables. Dans Galapagos, les fonctions de fitness peuvent être définies avec des critères de recherche pour obtenir une valeur numérique maximale, minimale ou spécifiée. » [7]

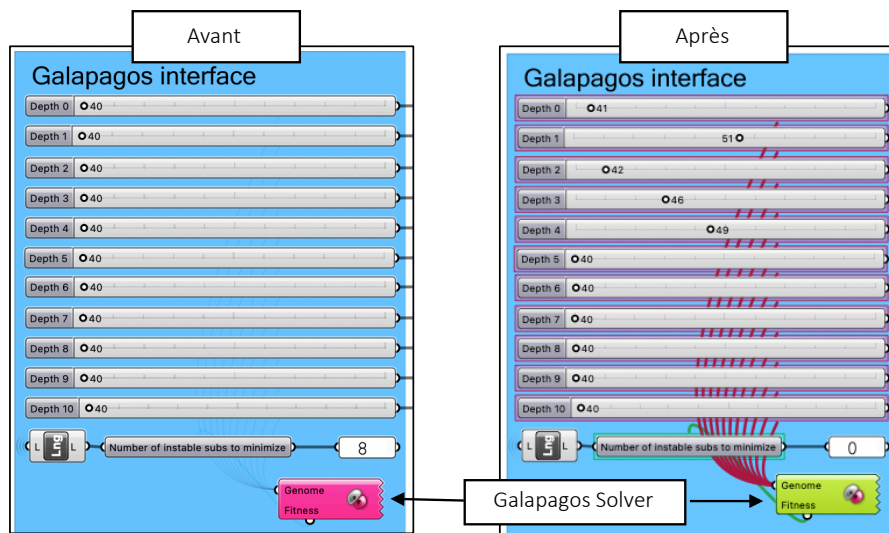


Figure 71 : Interface Galapagos

Le solveur va faire varier les curseurs jusqu'à obtenir une taille nulle de la liste des sous-structures instables (Figure 71). L'impact sur la droxelisation est visible à la Figure 72.

Si on augmente la valeur *Depth* pour une couche il y aura un plus grand nombre de droxels placés dans celle-ci et de meilleures chances de stabiliser les sous-structures.

La solution que propose le solveur n'est pas unique mais est satisfaisante. On peut diminuer l'intervalle des *Depth* pour réduire les libertés de Galapagos et, de cette manière, minimiser le nombre de droxels.

Toujours pour cet exemple, on a paramétré l'intervalle des curseurs *Depth* entre 40 mm et 60 mm. Il apparaît que le nombre de droxels a augmenté là où apparaissaient précédemment les sous-structures instables, ce qui les a éliminées (Figure 72). En effet, la position du centre de gravité et la surface d'appui des sous-structures sont influencées par la disposition et le nombre de droxels.

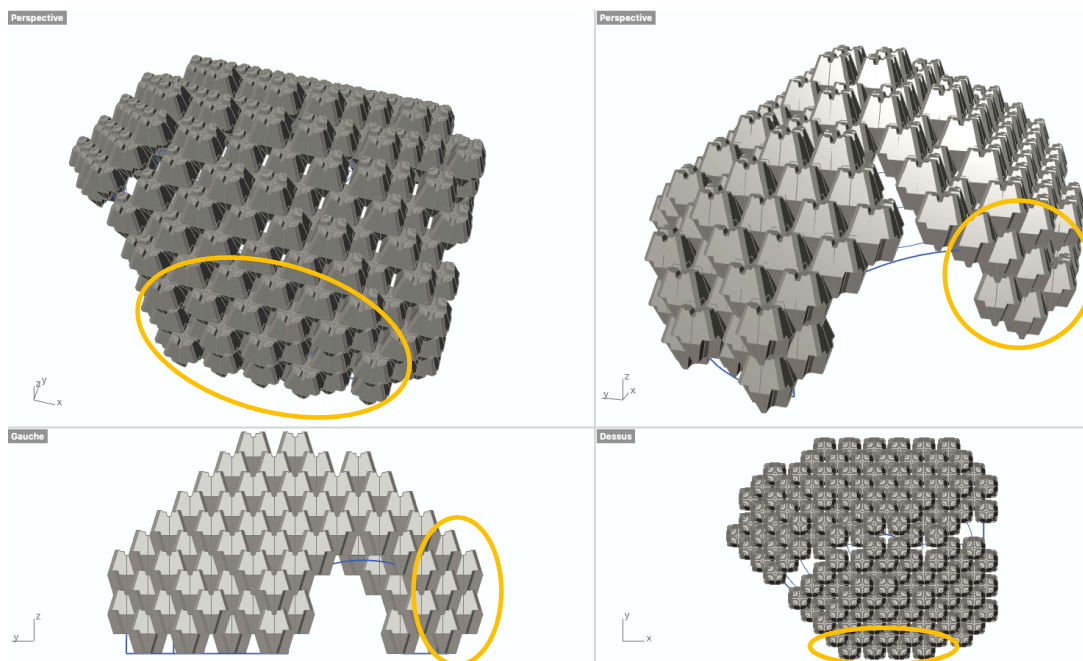


Figure 72 : Structure droxelisée de l'exemple C (2)

9. Conclusion

Le but de ce mémoire était de trouver un moyen d'automatiser le processus de droxelisation de structures sur base d'une approche paramétrique utilisant le logiciel Rhinoceros – Grasshopper et en y incluant une étude de stabilité globale afin de conceptualiser et de mettre en place les droxels de manière efficace.

Au cours de cette thèse j'ai développé un code Grasshopper pour droxéliser des architectures de tous types. Le programme développé automatise le processus de transformation de structures en droxels. Les modèles 3D Rhino entrés dans le programme sont traduits en droxels et son algorithme analyse l'influence de chaque droxel sur la stabilité globale du modèle lors de sa mise en place. C'est un code interactif dans lequel l'utilisateur est libre de choisir les dimensions des droxels et la largeur de l'espace que ceux-ci occupent autour de la structure initiale. Les structures empilables de type droxels se divisent en sous-structures. L'algorithme vérifie la stabilité globale en utilisant le principe des sous-structures. Il détecte les sous-structures instables lorsque la projection verticale de leur centre de gravité ne se trouve pas dans leur zone d'appui.

Le code ainsi conçu est visuel et peut facilement être amélioré ou modifié pour affiner la droxelisation et proposer des solutions stables au niveau global mais aussi local de manière algorithmique. Une seule méthode pour assurer la stabilité globale a été testée dans ce mémoire. Elle consiste à faire varier le facteur d'épaisseur des structures pour chaque couche aléatoirement à l'aide du solveur Galapagos (plugin de Grasshopper) pour tendre vers un nombre nul d'instabilités.

La question sur laquelle j'aimerais revenir aujourd'hui est la suivante :

« *Qu'aurais-je pu mieux faire ?* »

Pour répondre à cette question, il faut prendre un peu de recul. Car lorsqu'on se plonge dans la programmation il est difficile de lever la tête pour analyser la direction que l'on suit. Nos premiers choix peuvent nous enfermer dans un scénario à voie unique. J'évoque cela car j'aurais bien aimé prendre un plus grand temps de réflexion afin trouver un moyen d'optimiser mon code, notamment à l'aide de différents solveurs ou de scripts (Python, C#, etc.). Peut-être que Grasshopper a ses limites dans la programmation et que « *son fonctionnement est plus proche de la modélisation paramétrique que de la programmation algorithmique* » [1].

Malgré cela, j'ai pu produire un code fonctionnel qui correspond à mes attentes de départ. L'objectif global de la conception de programmes concernant les droxels est bien d'automatiser et de garantir un processus de conception stable à tout moment de la mise en œuvre. Ces programmes sont des outils de modélisation mais également de simulation pour éviter de réelles erreurs structurelles de conception.

10. Table des figures

Figure 1 : Géométrie du droxel	5
Figure 2 : Description du droxel	5
Figure 3 : Géométrie du droxel dans un système de coordonnées cartésiennes.....	6
Figure 4 : Modes de pose d'un droxel	7
Figure 5 : Directions principales d'un droxel (vue du dessus)	8
Figure 6 : Voisins d'un droxel.....	8
Figure 7 : Couche de droxels	9
Figure 8 : Exemple de projection verticale de résultante en dehors de la zone d'appui du mode 1	10
Figure 9 : Mouvement d'instabilité d'un droxel de mode 1 dans les directions -AA- et -BB-	11
Figure 10 : Exemple de projection verticale de résultante en dehors de la zone d'appui du mode 3.a et 3.b	11
Figure 11 : Mouvement de l'instabilité du mode 3.a	12
Figure 12 : Application du processus de détection (1)	13
Figure 13 : Application du processus de détection (2)	14
Figure 14 : Application du processus de détection (3)	15
Figure 15 : Canevas principal.....	16
Figure 16 : Composant Grasshopper pour supprimer les doublons	17
Figure 17 : Légende d'un flow chart.....	18
Figure 18 : Corrélation des clusters.....	19
Figure 19 : Flow chart du cluster 01 – Base	20
Figure 20 : Flow chart du cluster 011 – Grid	22
Figure 21 : Cluster 02 – Layer	23
Figure 22 : Flow chart du cluster 02 - Layer.....	24
Figure 23 : Flow chart du cluster 021 – Droxel	26
Figure 24 : Flow chart du cluster 022 – Position.....	27
Figure 25 : Cluster 023 - Stability Check.....	28
Figure 26 : Flow chart du cluster 023 – Stability Check	29
Figure 27 : Flow chart du cluster 0231 - Step Sub	31
Figure 28 : Flow chart du cluster 0232 - Sub Edit	34
Figure 29 : Exemple du processus « Sorting »	35
Figure 30 : Flow chart du cluster 02311 – Lowers	37
Figure 31 : Flow chart du cluster 02312 – Uppers.....	38
Figure 32 : Flow chart du cluster 023121 - Step Uppers	40
Figure 33 : Flow chart du cluster 02313 – Substructure.....	41
Figure 34 : Flow chart du cluster 02314 - Gravity Center	43
Figure 35 : Flow chart du cluster 02315 - Gravity Test.....	45
Figure 36 : Flow chart du cluster 02316 - Lowest Statement.....	46
Figure 37 : Flow chart du cluster 02317 - Sub Statement	47
Figure 38 : Surface d'appui réduite	49
Figure 39 : Structure de l'exemple A.....	50
Figure 40 : Droxelisation de l'exemple A.....	50
Figure 41 : Modélisation 3D de la séquence de pose de l'exemple A.....	51
Figure 42 : Structure de l'exemple B	53
Figure 43 : Droxelisation de l'exemple B	53
Figure 44 : Modélisation 3D de la séquence de pose de l'exemple B	54
Figure 45 : Sous-structures instables de l'exemple B détectées par l'algorithme	54
Figure 46 : Base points pour la couche n°0 de l'exemple B	56

Figure 47 : Droxels de la couche n°4 de l'exemple B	56
Figure 48 : Base points de la couche n°5 de l'exemple B	56
Figure 49 : Base point n°5 de l'exemple B.....	56
Figure 50 : Sous-structure de l'exemple B	57
Figure 51 : Surface d'appui de l'exemple B	57
Figure 52 : Pente des droxels de mode 2.a	58
Figure 53 : Couche de transition	58
Figure 54 : Condition de changement de pente	59
Figure 55 : Pente des droxels de mode 2.a	60
Figure 56 : Droxel de "pré-transition" et de "transition"	60
Figure 57 : Visualisation du programme de la couche de transition	61
Figure 58 : Programme Grasshopper de la couche transitoire	61
Figure 59 : Paramètres du droxel inférieur et supérieur	62
Figure 60 : Points de contrôle pour le droxel de pré-transition	62
Figure 61 : Points de contrôle à modifier pour le droxel de pré-transition	62
Figure 62 : Composant Grasshopper « Brep Join ».....	63
Figure 63 : Points de contrôle pour le droxel de transition	64
Figure 64 : Assemblage du droxel de transition.....	65
Figure 65 : Piste d'amélioration (1)	68
Figure 66 : Piste d'amélioration (2)	68
Figure 67 : Structure à droxéliser de l'exemple C	71
Figure 68 : Structure droxelisée de l'exemple C (1)	72
Figure 69 : 1ère sous-structures instables détectées	72
Figure 70 : 2ème sous-structures instables détectées	73
Figure 71 : Interface Galapagos.....	74
Figure 72 : Structure droxelisée de l'exemple C (2)	74

11. Table des images

Image 1 : Exemple du processus "selecting"	33
Image 2 : Exemple du cluster 02314 - Gravity Center	42
Image 3 : Pied à coulisse.....	48
Image 4 : Structure simple pour déterminer Tol	49
Image 5 : Séquence de pose de l'exemple A.....	52
Image 6 : Vue de l'exemple A droxelisé avec les blocs imprimés en 3D	52
Image 7 : Séquence de pose de l'exemple B.....	55
Image 8 : Instabilité de l'exemple B	55
Image 9 : Exemple de coefficients de frottement différents.....	67
Image 10 : Exemple d'instabilité du mode 2.a	69

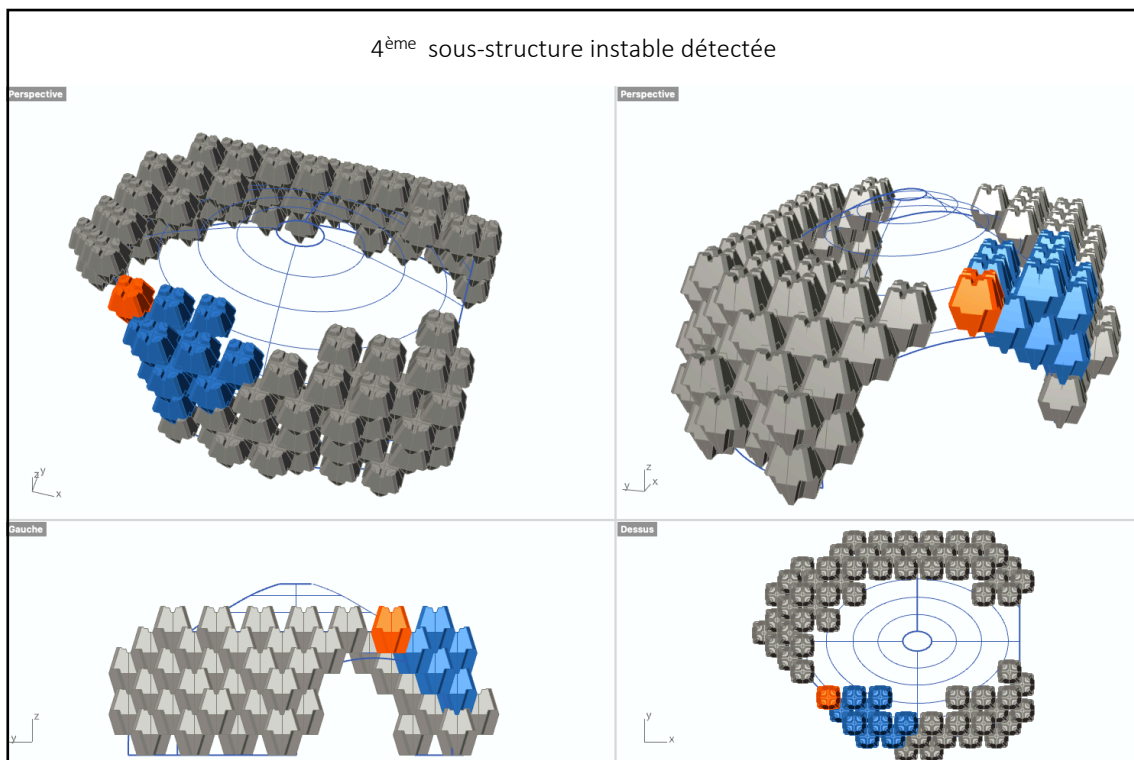
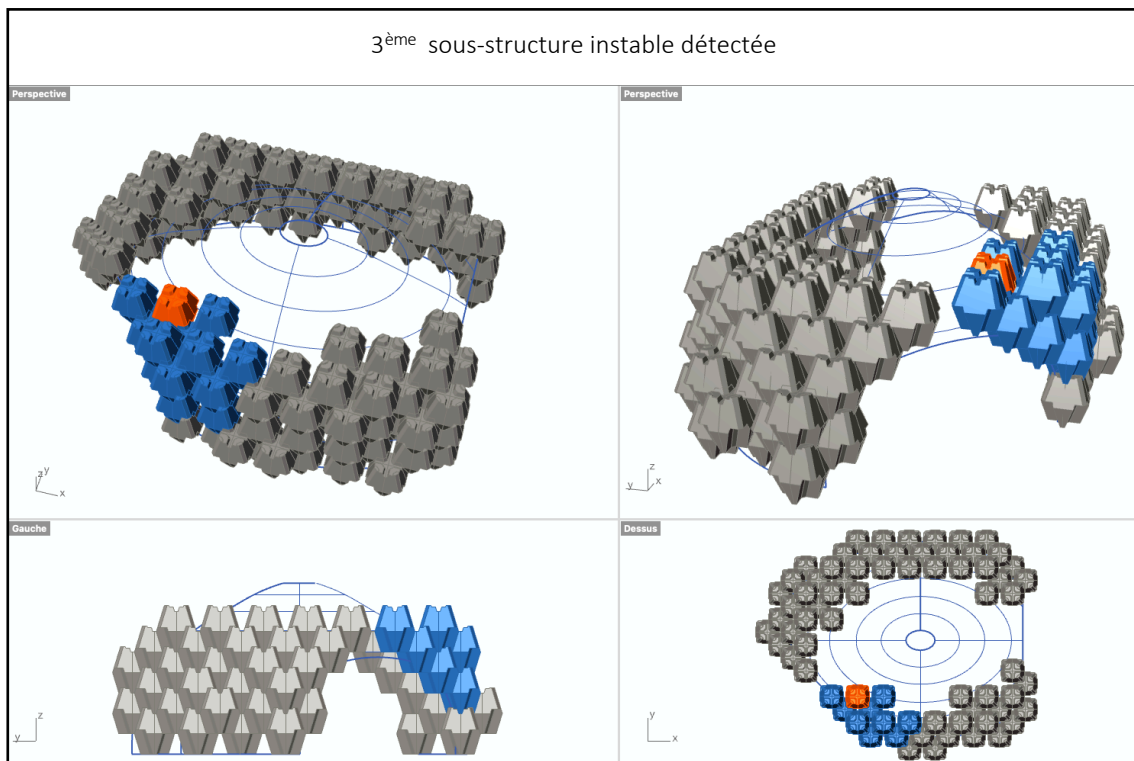
12. Table des tableaux

Tableau 1 : Inputs du cluster 01 - Base	20
Tableau 2 : Outputs du cluster 01 - Base	20
Tableau 3 : Inputs du cluster 011 – Grid	21
Tableau 4 : Outputs du cluster 011 – Grid	21
Tableau 5 : Inputs du cluster 02 - Layer	24
Tableau 6 : Outputs du cluster 02 - Layer	24
Tableau 7 : Inputs du cluster 021 – Droxel	25
Tableau 8 : Outputs du cluster 021 - Droxel	25
Tableau 9 : Inputs du cluster 022 - Position.....	27
Tableau 10 : Outputs du cluster 022 - Position.....	27
Tableau 11 : Inputs du cluster 023 – Stability Check	28
Tableau 12 : Outputs du cluster 023 – Stability Check	28
Tableau 13 : Inputs du cluster 0231 - Step Sub	31
Tableau 14 : Outputs du cluster 0231 - Step Sub	31
Tableau 15 : Inputs du cluster 0232 - Sub Edit	33
Tableau 16 : Outputs du cluster 0232 - Sub Edit	33
Tableau 17 : Inputs du cluster 02311 - Loweres	36
Tableau 18 : Outputs du cluster 02311 – Loweres	36
Tableau 19 : Inputs du cluster 02312 - Uppers.....	38
Tableau 20 : Outputs du cluster 02312 – Uppers	38
Tableau 21 : Inputs du cluster 023121 - Step Uppers.....	39
Tableau 22 : Outputs du cluster 023121 - Step Uppers	39
Tableau 23 : Inputs du cluster 02313 – Substructure.....	41
Tableau 24 : Outputs du cluster 02313 – Substructure.....	41
Tableau 25 : Inputs du cluster 02314 - Gravity Center	42
Tableau 26 : Outputs du cluster 02314 - Gravity Center	42
Tableau 27 : Inputs du cluster 02315 - Gravity Test	44
Tableau 28 : Outputs du cluster 02315 - Gravity Test	44
Tableau 29 : Inputs du cluster 02316 - Lowest Statement.....	46
Tableau 30 : Outputs du cluster 02316 - Lowest Statement.....	46
Tableau 31 : Inputs du cluster 02317 - Sub Statement	47
Tableau 32 : Outputs du cluster 02317 - Sub Statement	47
Tableau 33 : Paramètres géométriques des droxels imprimés	48
Tableau 34 : Inputs de l'exemple A	50
Tableau 35 : Inputs de l'exemple B	53
Tableau 36 : Inputs du cluster Pre-transition Droxel	63
Tableau 37 : Outputs du cluster Pre-transition Droxel	63
Tableau 38 : Inputs du cluster Transition Droxel	65
Tableau 39 : Outputs du cluster Transition Droxel	65
Tableau 40 : Inputs de l'exemple C	71

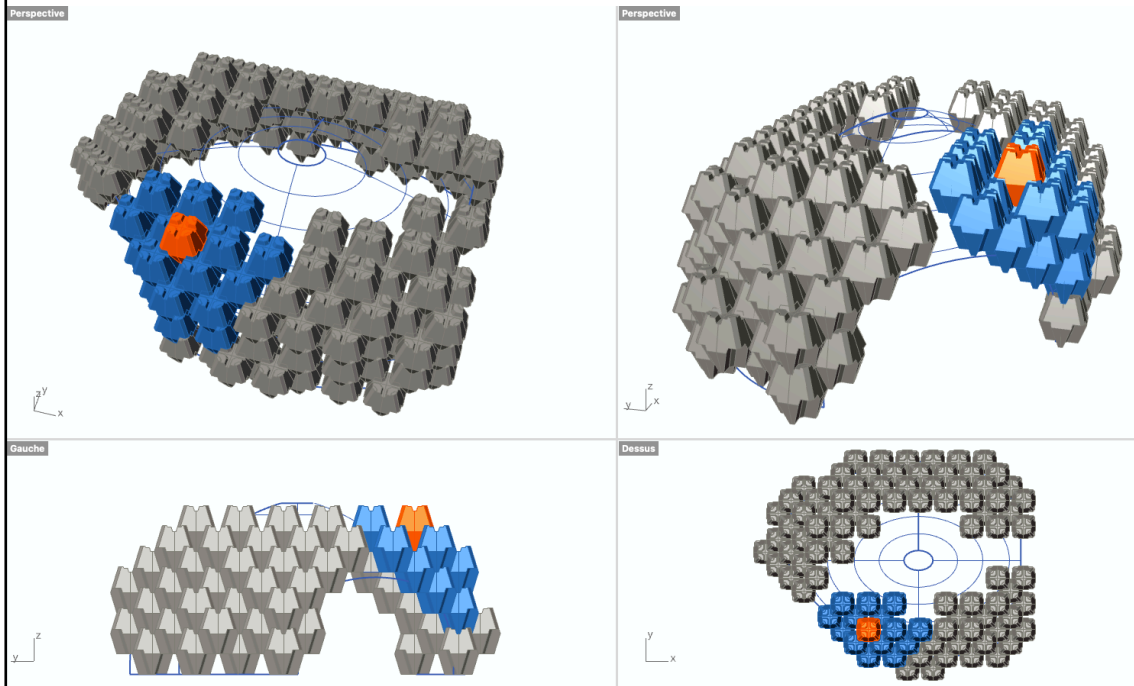
13. Références bibliographiques

- [1] AURELIE, D. B. (2012, 1 FEVRIER). *GRASSHOPPER ET LA PROGRAMMATION SUR RHINOCEROS 4*. DNARCHI. [HTTP://DNARCHI.FR/PEDAGOGIES/GRASSHOPPER-ET-LA-PROGRAMMATION-SUR-RHINOCEROS-4-UNE-INTRODUCTION/](http://dnarchi.fr/pedagogies/grasshopper-et-la-programmation-sur-rhinoceros-4-une-introduction/)
- [2] FORM2FAB. (S. D.). *OPTEAM DEBORAH SESSION #1*. [HTTPS://WWW.FORM2FAB.COM/RHINO3D/](https://www.form2fab.com/rhino3d/)
- [3] GOESSENS, S., MUELLER, C., & LATTEUR, P. (2018). *FEASIBILITY STUDY FOR DRONE-BASED MASONRY CONSTRUCTION OF REAL-SCALE STRUCTURES*. *AUTOMATION IN CONSTRUCTION*, 94, 458-480. [HTTPS://DOI.ORG/10.1016/J.AUTCON.2018.06.015](https://doi.org/10.1016/j.autcon.2018.06.015)
- [4] *GRASSHOPPER-TOI-PEDIA*. (S.D.). WIKI.BK.TUDELFT. CONSULTÉ LE 2020, À L'ADRESSE [HTTP://WIKI.BK.TUDELFT.NL/TOI-PEDIA/GRASSHOPPER](http://wiki.bk.tuelft.nl/toi-pedia/grasshopper)
- [5] ROBERT MCNEEL & ASSOCIATES. (S. D.). *RHINOCEROS 3D*. [WWW.RHINO3D.COM](http://www.rhino3d.com). [HTTPS://WWW.RHINO3D.COM/](https://www.rhino3d.com/)
- [6] *GRASSHOPPER DATA TREE EDITING - TOI-PEDIA*. (S. D.). WIKI.BK.TUDELFT. [HTTP://WIKI.BK.TUDELFT.NL/TOI-PEDIA/GRASSHOPPER_DATA_TREE_EDITING](http://wiki.bk.tuelft.nl/toi-pedia/grasshopper_data_tree_editing)
- [7] *GENETIC ALGORITHM SOLVER : GALAPAGOS*. (S. D.). CLIMATE FACADE. CONSULTÉ LE 2020, À L'ADRESSE [HTTP://CLIMATEFACADE.COM/GENETIC-ALGORITHM-SOLVER-GALAPAGOS](http://climatefacade.com/genetic-algorithm-solver-galapagos)

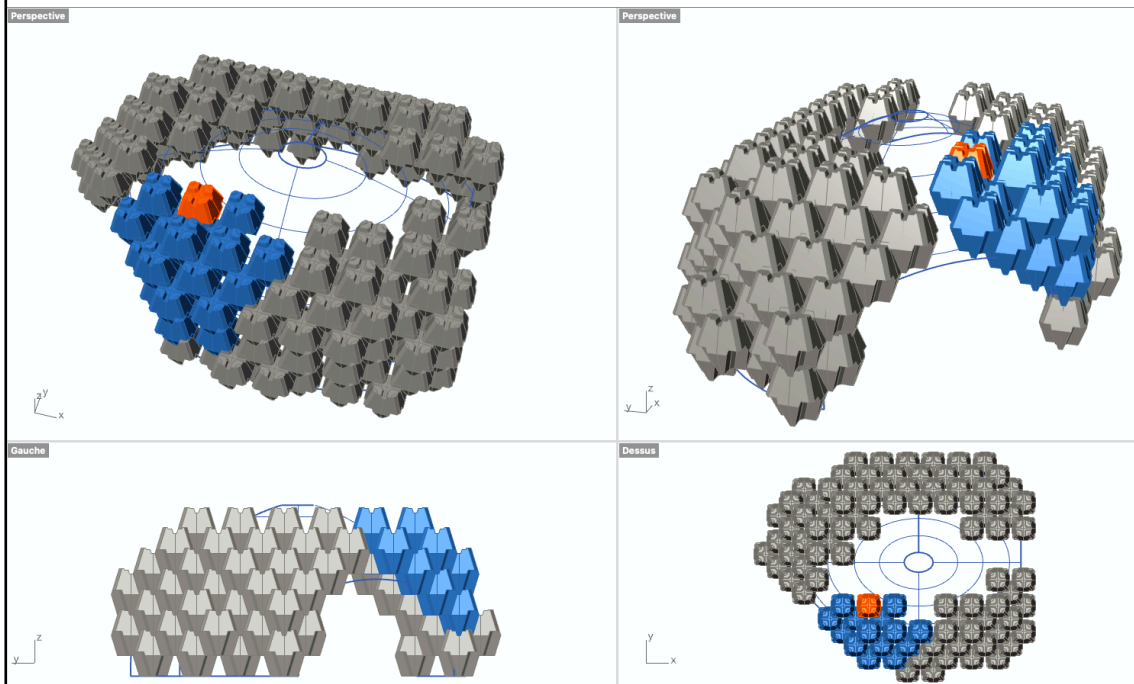
14. Annexes



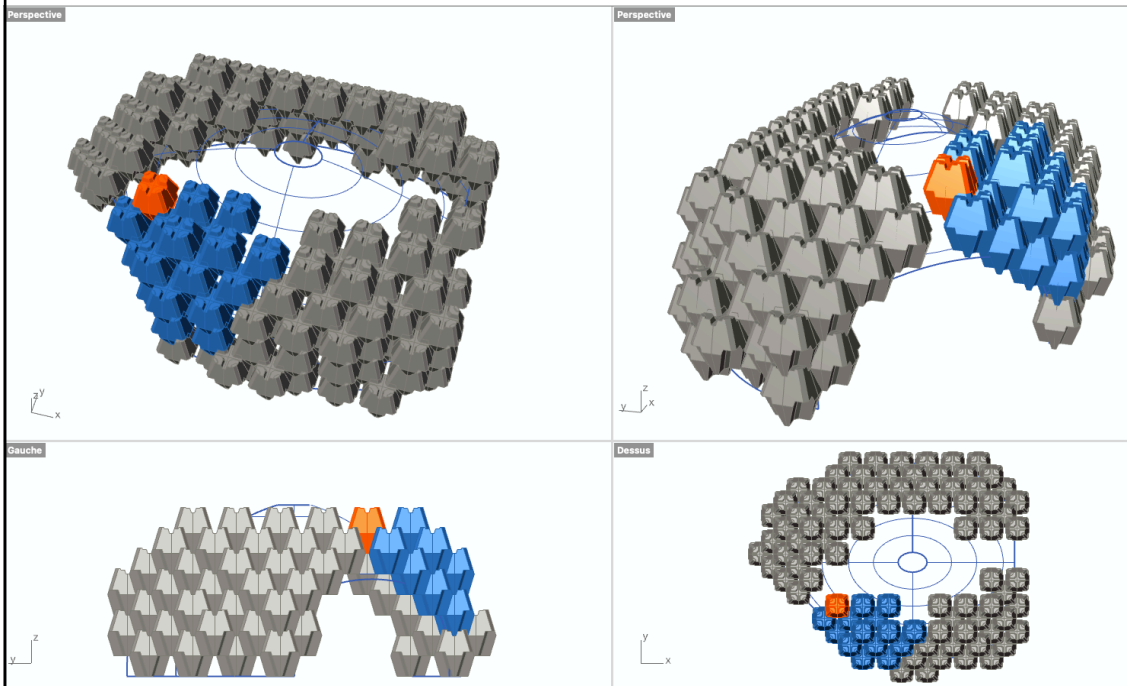
5^{ème} sous-structure instable détectée



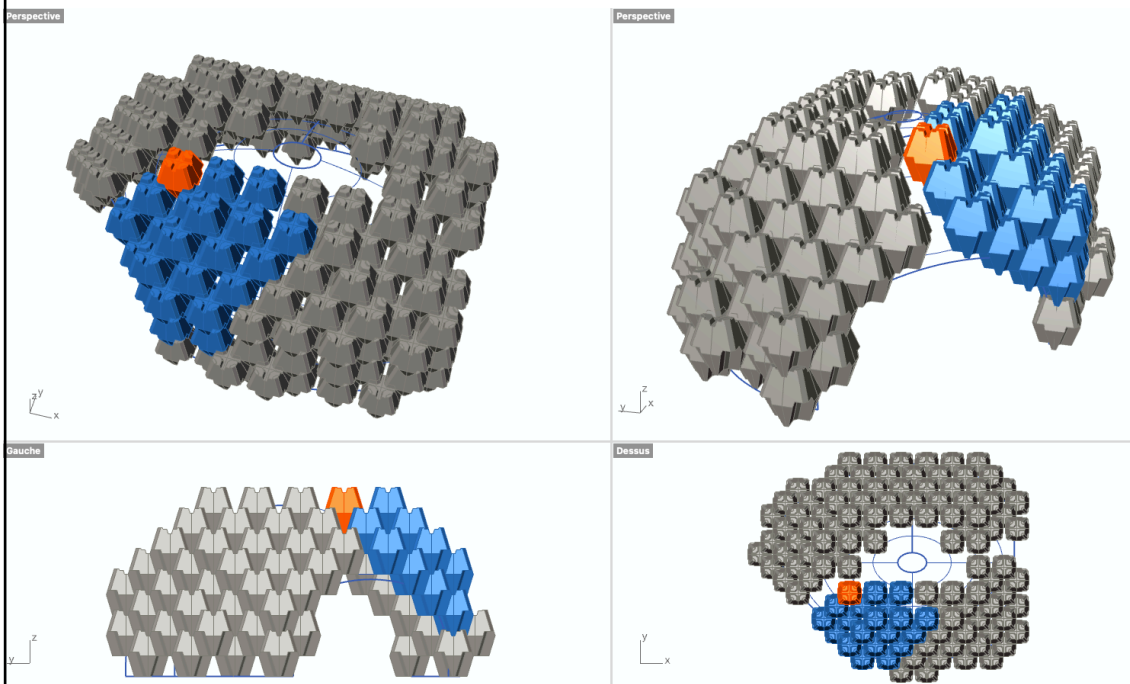
6^{ème} sous-structure instable détectée



7^{ème} sous-structure instable détectée



8^{ème} sous-structure instable détectée



UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl