Streamlining Workflows in construction with Parametric Design

Thomas Ruwet, Manuel Elias

Bureau Greisch, Liège, Belgium

Contact: truwet@greisch.com

Abstract

Parametric tools optimise design and production in engineering and architecture by automating tasks, improving accuracy, and streamlining manufacturing. This article explores the potential of these tools for BIM modeling, the use of digital terrain models, structural optimisation, and the systematisation of complex assemblies. Through concrete cases it demonstrates how these technologies accelerate 3D modeling, data exchange, quickly test various scenarios, and facilitate the transition between design and manufacturing. The transformation of these workflows opens new doors in terms of expertise and innovation.

Keywords: Parametric, design, production, automation, modeling, optimisation, BIM, interoperability.

1 Introduction

Parametric design has revolutionised engineering and architectural practices. Far more than just a modeling tool, it enables the optimisation of design and production processes by automating repetitive tasks, enhancing accuracy, and improving interoperability between software.

In an industry where BIM modeling requirements, structural performance, and manufacturing efficiency are increasingly demanding, these tools provide a strategic advantage by reducing lead times, minimising errors, and rapidly exploring multiple design solutions. But how much can they truly simplify our workflows? How many steps and stakeholders are still required between an engineer's initial sketch and the production of final technical drawings?

This article examines this transformation through real-world projects, focusing on two key phases: design and deliverable production.

.

2 Design

From project to context, from context to concept, from concept to structure, and from structure to optimisation, there are many steps where we can save effort to focus our time and energy on finding the best solutions.

2.1 Software

Rhino (Rhinoceros) is a classic modeling software. It is not tied to any specific profession, making it highly versatile. Its management of complex geometries is remarkable while remaining relatively easy to use. Grasshopper is a graphical algorithm editor that serves as Rhino's parametric interface, in which it is natively integrated. It allows users to describe step-by-step modeling procedures in Rhino. Updating an input affects the entire script that follows.

2.2 Applications

2.2.1 Digital Terrain Model Utilisation

The first sketches of a project are usually made long before receiving a work base as precise as topographic surveys. Therefore, we must utilise all available resources to compensate for this lack of context.

Digital Terrain Models (MNT) are one such resource. A LiDAR is deployed during an aerial survey, generating a surface model that is then processed through an algorithm to retain only the ground topography. The result is transcribed into a GeoTIFF file (image) that can be processed using software like Saga to generate a 3D model (mesh).

In this example, via Grasshopper, an initial estimate of excavation volumes was made.

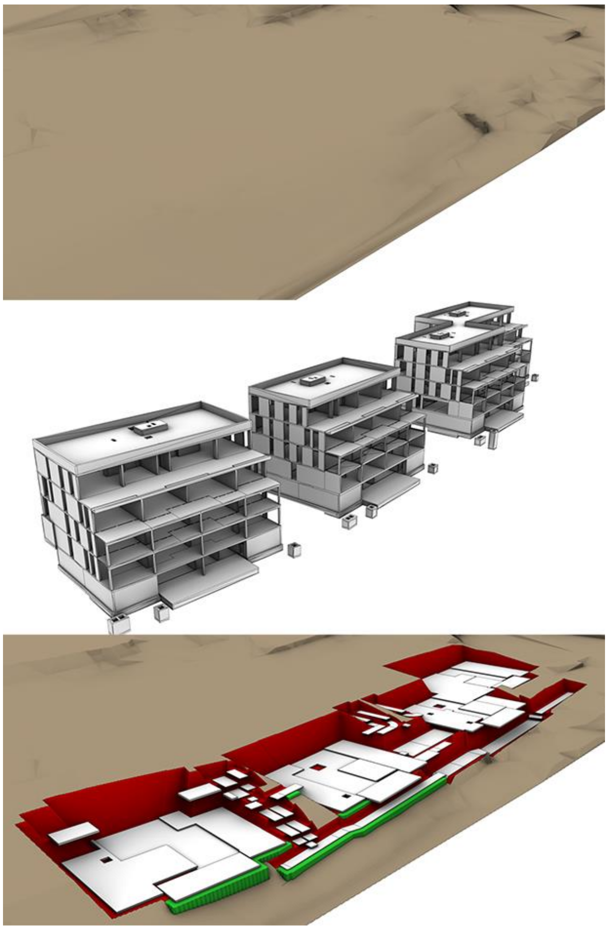


Figure 1. Excavation et backfill

Input: The terrain, projected foundations, excavation overhang width, and slope angle.

Output: Red for excavation areas, green for backfill areas. The volumes are exported into an Excel table sorted by lot and soil type (schist or clay).

This type of model is also very useful for 3D rendering to visualise or present the design of a structure in its immediate context.

2.2.2 Karamba

Karamba is a Grasshopper plugin that enables finite element analysis. Specifically developed to allow designers to quickly visualise structural forces, it integrates perfectly with Grasshopper's algorithmic logic.

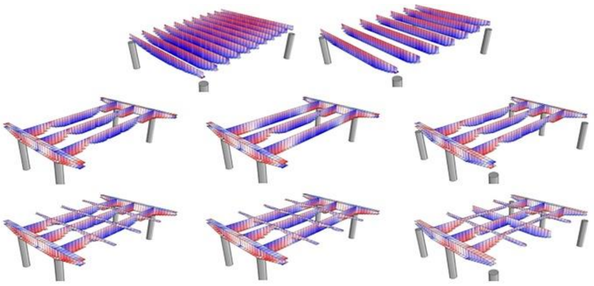


Figure 2. Structure analysis

Numerous iterations can be considered and analysed to select the most promising solutions.

Although Karamba is primarily used for preliminary design, comparisons with results from FinelG (calculation software developed by Greisch in collaboration with the University of Liège) have proven the plugin's reliability. However, the final structural dimensioning studies are still performed using FinelG.

Data exchange between these different structural analysis programs is now facilitated through a SAF (Structural Analysis Format) file. Based on Excel, this open format developed by the Nemetschek group and managed by SCIA aims to streamline communication among engineers.

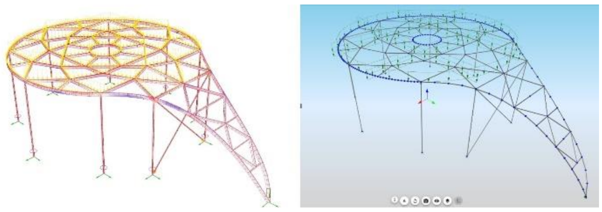


Figure 3. Data exchange between structural analysis programs

2.2.3 Genetic algorithm

A genetic algorithm is based on the theory of species evolution. It tests a set of parameters (a population) initially at random. Each subsequent population is generated from the results that best meet the optimisation criteria. This type of algorithm is used for complex problems involving numerous parameters and optimisation criteria.

In this project (the canopies of Legiapark in Liège), the architect envisioned a tree-like structure. The footing positions were fixed. The script tested various arrangements for the "branches" to achieve an organic yet structurally coherent layout.

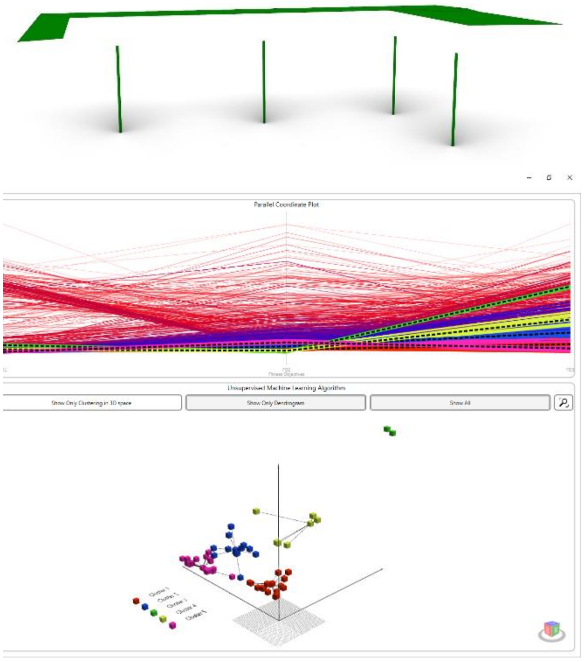


Figure 4. Genetic algorithm in Wallacei

The script is broken down into steps:

1. Defining geometry (wireframe model)
2. Constructing the model (mechanical properties of bars)
3. Structural analysis (Karamba)
4. Optimisation (Genetic algorithm via Wallacei plugin)

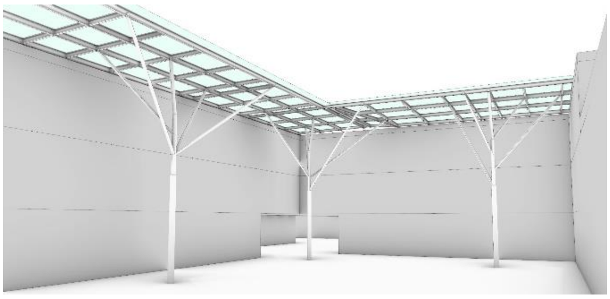


Figure 5. Canopies of Legiapark

2.2.4 Optimisation for machining

Parametric design showcases its full power when dealing with numerous similar but unique elements. Consider the example of the Riyadh Metro canopy in Saudi Arabia.

The challenge of this project was to rationalise the assemblies of the mesh. These are hexagonal blocks from which six branches extend at constantly varying angles.

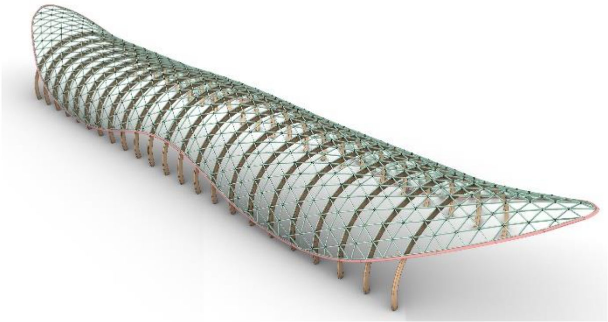


Figure 6. Riyadh Metro canopy, Saudi Arabia

An initial draft was proposed, but it made bar cutting impossible for the local steelworker.

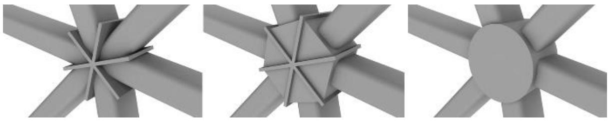


Figure 7. Inital assembly draft

An optimisation was therefore proposed to concentrate all the assembly complexity within the node while respecting the ideal heights and diameters of the bars.

The result is a solid hexagonal piece with machined pockets of varying dimensions. These dimensions are recorded in a table and annotated in a typical plan view.

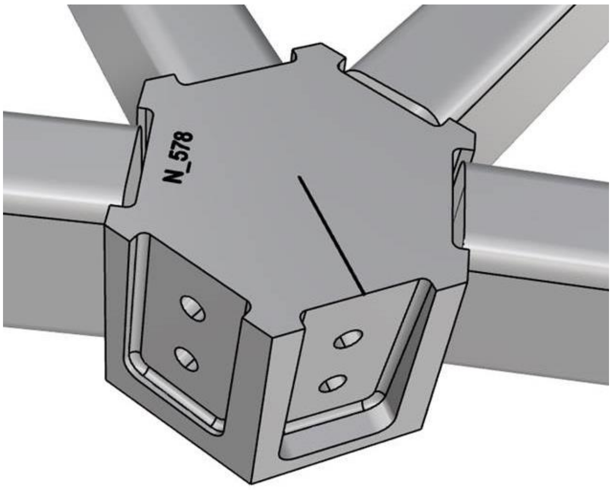


Figure 8. Final design

2.2.5 Abidjan

The next example is the Abidjan Tower. For this project, the inputs were the outer envelope designed by the architect and the floor plans. From there, the structure developed from the exterior inward: walkway, edge beam, columns, floors, core, etc.

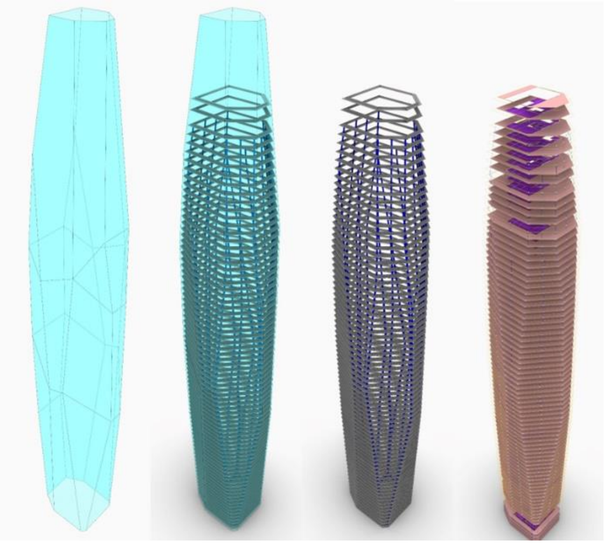


Figure 9. The structure follows the envelope

A wireframe model was created for use in our structural analysis software. Simultaneously, each element of this wireframe model was enriched with attributes defining all necessary information for its modeling in Revit. This brings us to our next topic: document production.

3 Production

By production, we mean the creation of a 3D model and/or 2D drawings. Creating a purely geometric 3D model has never been an end in itself. This is even more true since BIM became a standard in our sector. Thus, transitioning a model from its conceptual phase to its BIM version has become imperative.

3.1 Software

Revit has become a major player in BIM. The construction sector is now inseparable from it, and many constructions project specifications mention "Revit model." Revit’s strength in modeling lies in its use of families. These are dynamic, parametric blocks that can be adapted into multiple forms. However, while Revit is parametric in its own way, it lacks the ability to generate complex designs like Grasshopper. Rhino Inside Revit combines the strengths of both software. This Revit plugin allows the transfer of geometries between the two programs. In our case, it is used at the end of the Grasshopper script to control the placement and parameterisation of pre-created Revit families.

3.2 Applications

3.2.1 Abidjan

Continuing from where we left off: a wireframe and surface model in which each element contains the necessary information for its creation and parameterisation in Revit.

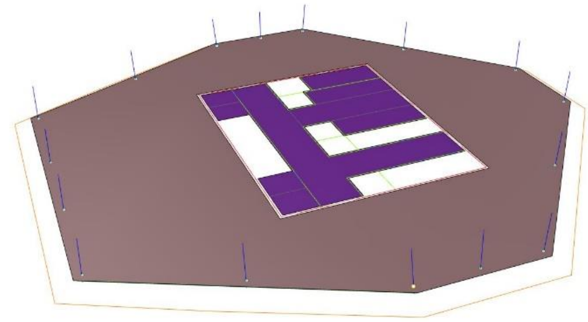


Figure 10. Wireframe model & data

Why this intermediate step? Simply to enable a handoff between the design and production teams.

A modeller using this skeleton model frozen at a given moment in time doesn't need to bother with the whole process that preceded it and can concentrate on the task in hand: creating a Revit model.

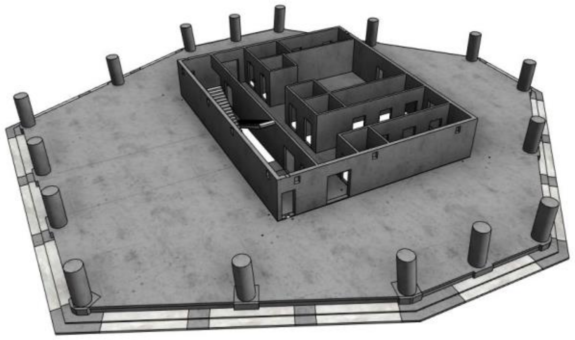
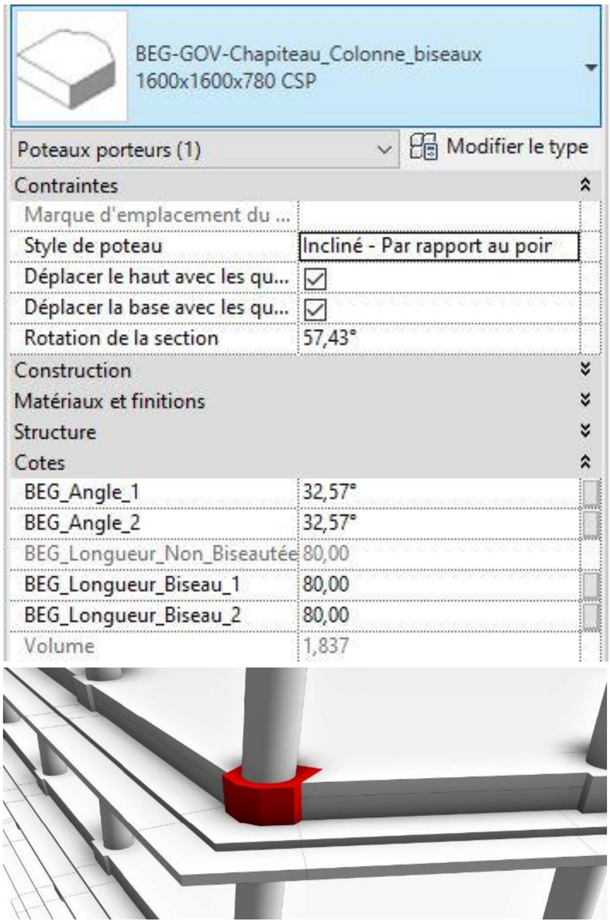


Figure 11. The Revit model is taking shape

All these parameters are integrated into Revit families corresponding to each type of object: beam, walkway, capital, column, etc.



BEG-GOV-Chapiteau_Colonne_biseaux
1600x1600x780 CSP

Poteaux porteurs (1) — Modifier le type

Contraintes	
Marque d'emplacement du ...	
Style de poteau	Incliné - Par rapport au poir
Déplacer le haut avec les qu...	☑
Déplacer la base avec les qu...	☑
Rotation de la section	57,43°
Construction	
Matériaux et finitions	
Structure	
Cotes	
BEG_Angle_1	32,57°
BEG_Angle_2	32,57°
BEG_Longueur_Non_Biseautée	80,00
BEG_Longueur_Biseau_1	80,00
BEG_Longueur_Biseau_2	80,00
Volume	1,837

Figure 12. The object gets its shape from the precise parameters inherited from the global script

The Revit model is taking shape, and with it the possibility of quickly producing quantity take-offs. These measurements will enable us to quickly quantify the impact of certain construction choices and make any necessary adjustments.

This led to an analysis of the distribution of the prefabricated elements making up the perimeter walkways. These elements have standardised lengths and must be aligned with the glazing frames of the tower's double skin. While respecting these constraints, we were able to test three scenarios in which we varied the overlap lengths of the bars between the precast and the cast-in-place elements (65, 100 and 200cm).

The generation of these 3 scenarios enabled us to decide on the most economical solution and clearly illustrates the potential of a BIM model orchestrated by grasshopper.

3.2.2 Vilvoorde

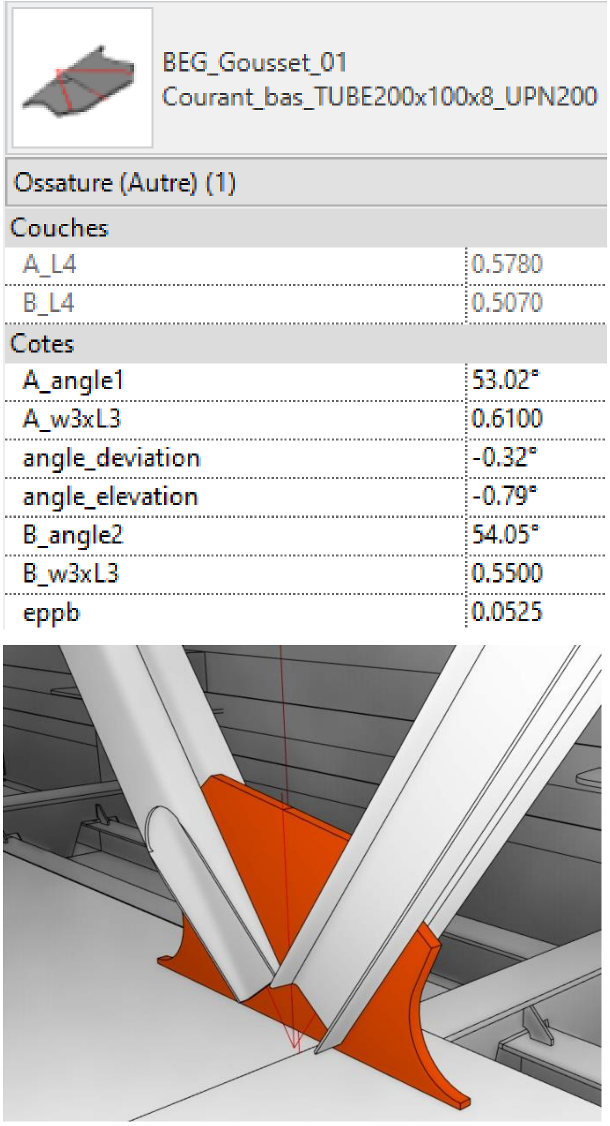
The renovation of the Vilvoorde viaduct is another example of a complex structure that would have been incredibly time-consuming to model in a "traditional" way using Revit.

Figure 13. Projected view of the inside of the bridge reinforced

The distribution of elements and their characteristics are transcribed into a single Excel table based on the plans of the existing structure and the material plans from the project phase. This table assigns each element (beam, reinforcement, gussets, trough, sheet metal, etc.) a longitudinal position and a type.

At the same time, numerous Revit families are created. Just like in the case of Abidjan, these families include all the necessary parameters to meet the various geometric constraints of the project. This complex parameterisation takes the form of numerous trigonometric formulas linking the different dimensions of the object.

These families are positioned and configured directly from the Grasshopper script, which compiles the geometry of the longitudinal profile (a combination of clothoids, curves, and straight lines) along with the Excel table of elements.



BEG_Gousset_01
Courant_bas_TUBE200x100x8_UPN200

Ossature (Autre) (1)

Couches	
A_L4	0.5780
B_L4	0.5070
Cotes	
A_angle1	53.02°
A_w3xL3	0.6100
angle_deviation	-0.32°
angle_elevation	-0.79°
B_angle2	54.05°
B_w3xL3	0.5500
eppb	0.0525

Figure 14. The complexity of setting the parameters for these gussets is integrated into the Revit family rather than into the grasshopper script.

Identification numbers, reinforcement categories, and the localisation parameters of each element are easily assigned from Grasshopper. At this stage, there is nothing preventing us from truly considering the model as real BIM model.

Rhino Inside Revit is also capable of generating views and layout and arranging the former onto the latter. This automation allows us to quickly create the multitude of views necessary to represent this structure, which is over 800 meters long: cross-sections, developed elevation, specific element details, etc.

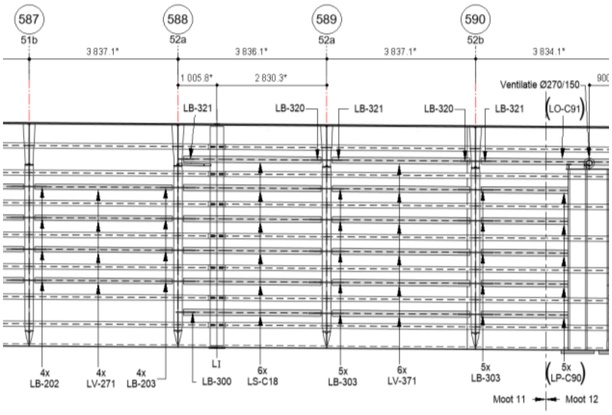


Figure 15. developed elevation of the sides of the bridge

Finally, some elements whose geometry follows a consistent logic but are unique in their dimensions (such as gussets) have been automatically dimensioned.

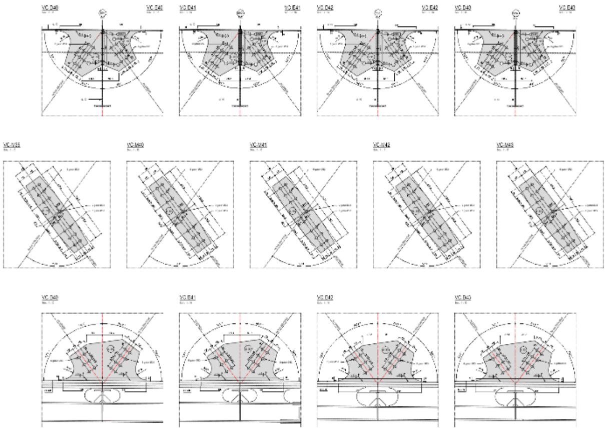


Figure 16. Automatically annotated gussets

4. Strengths and weaknesses

It is essential to clearly define the boundaries between each software: Grasshopper for geometric construction and Revit for production. Managing families and their specific functions is a crucial issue, as it helps direct efforts optimally, allowing for the full potential of each software to be harnessed without creating overly complex systems that make teamwork difficult, or even impossible.

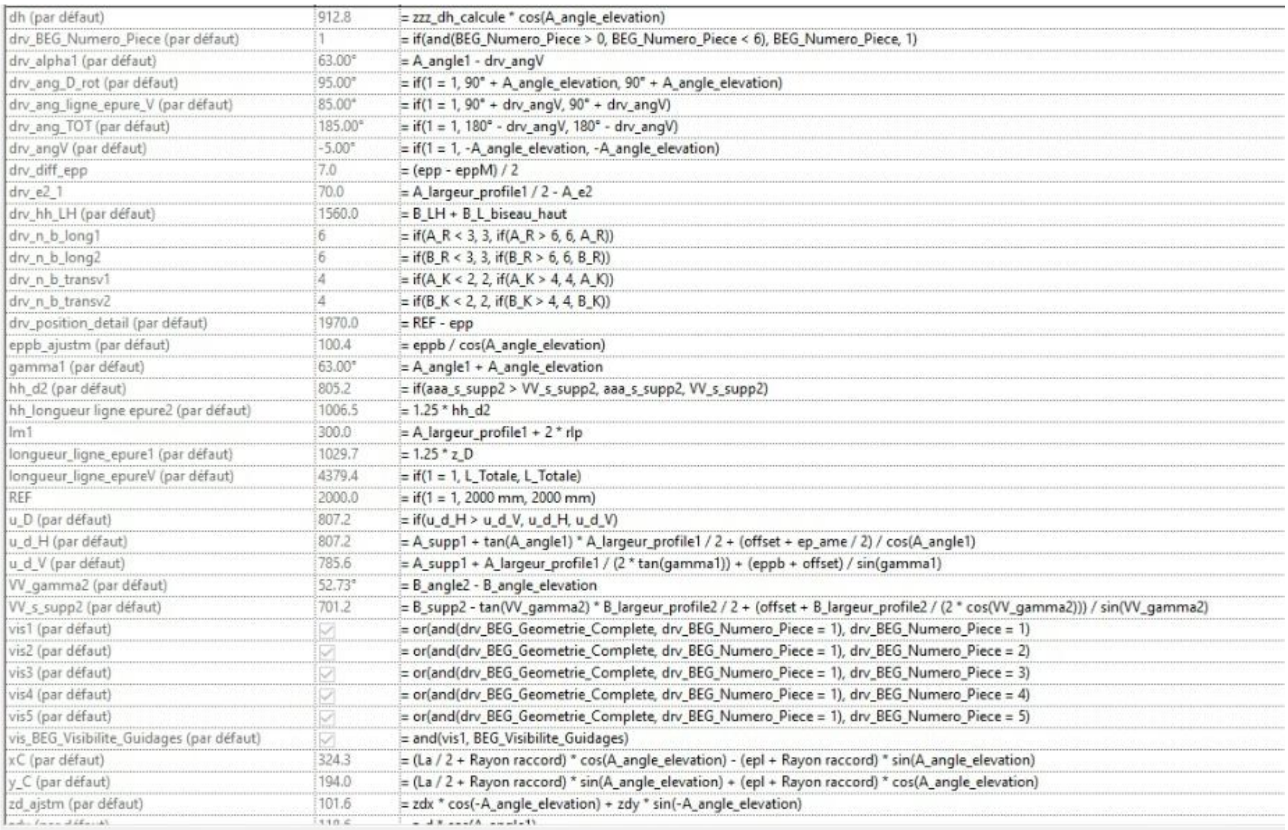
4.1 Example

Grasshopper is unparalleled when it comes to concatenating multiple geometric constraints. The multitude of functions available to find intersections between two objects is just one example illustrating the ease with which geometric logic can be established.

A Revit family, on the other hand, is built by linking geometries to reference planes, which are themselves positioned relative to each other based on parameters.

The physical and visual construction of Grasshopper is thus replaced in Revit by pure analytical geometry, relying on trigonometric equations that lead to excessively complex and unwieldy formulations.

During the Vilvoorde project, several such families were created by maximizing parameterization within Revit so that the input of values through Grasshopper was kept to an absolute minimum. In hindsight, a slightly more in-depth analysis within Grasshopper would have allowed these Revit elements to remain much simpler and easier to edit.



dh (par défaut)	912.8	= zzz_dh_calcule * cos(A_angle_elevation)
drv_BEG_Numero_Piece (par défaut)	1	= if(and(BEG_Numero_Piece > 0, BEG_Numero_Piece < 6), BEG_Numero_Piece, 1)
drv_alpha1 (par défaut)	63.00°	= A_angle1 - drv_angV
drv_ang_D_rot (par défaut)	95.00°	= if(1 = 1, 90° + A_angle_elevation, 90° + A_angle_elevation)
drv_ang_ligne_epure_V (par défaut)	85.00°	= if(1 = 1, 90° + drv_angV, 90° + drv_angV)
drv_ang_TOT (par défaut)	185.00°	= if(1 = 1, 180° - drv_angV, 180° - drv_angV)
drv_angV (par défaut)	-5.00°	= if(1 = 1, -A_angle_elevation, -A_angle_elevation)
drv_diff_epp	7.0	= (epp - eppM) / 2
drv_e2_1	70.0	= A_largeur_profile1 / 2 - A_e2
drv_hh_LH (par défaut)	1560.0	= B_LH + B_L_biseau_haut
drv_n_b_long1	6	= if(A_R < 3, 3, if(A_R > 6, 6, A_R))
drv_n_b_long2	6	= if(B_R < 3, 3, if(B_R > 6, 6, B_R))
drv_n_b_transv1	4	= if(A_K < 2, 2, if(A_K > 4, 4, A_K))
drv_n_b_transv2	4	= if(B_K < 2, 2, if(B_K > 4, 4, B_K))
drv_position_detail (par défaut)	1970.0	= REF - epp
eppb_ajustm (par défaut)	100.4	= eppb / cos(A_angle_elevation)
gamma1 (par défaut)	63.00°	= A_angle1 + A_angle_elevation
hh_d2 (par défaut)	805.2	= if(aaa_s_supp2 > VV_s_supp2, aaa_s_supp2, VV_s_supp2)
hh_longueur ligne epure2 (par défaut)	1006.5	= 1.25 * hh_d2
lm1	300.0	= A_largeur_profile1 + 2 * rlp
longueur_ligne_epure1 (par défaut)	1029.7	= 1.25 * z_D
longueur_ligne_epureV (par défaut)	4379.4	= if(1 = 1, L_Totale, L_Totale)
REF	2000.0	= if(1 = 1, 2000 mm, 2000 mm)
u_D (par défaut)	807.2	= if(u_d_H > u_d_V, u_d_H, u_d_V)
u_d_H (par défaut)	807.2	= A_supp1 + tan(A_angle1) * A_largeur_profile1 / 2 + (offset + ep_ame / 2) / cos(A_angle1)
u_d_V (par défaut)	785.6	= A_supp1 + A_largeur_profile1 / (2 * tan(gamma1)) + (eppb + offset) / sin(gamma1)
VV_gamma2 (par défaut)	52.73°	= B_angle2 - B_angle_elevation
VV_s_supp2 (par défaut)	701.2	= B_supp2 - tan(VV_gamma2) * B_largeur_profile2 / 2 + (offset + B_largeur_profile2 / (2 * cos(VV_gamma2))) / sin(VV_gamma2)
vis1 (par défaut)	☑	= or(and(drv_BEG_Geometrie_Complete, drv_BEG_Numero_Piece = 1), drv_BEG_Numero_Piece = 1)
vis2 (par défaut)	☑	= or(and(drv_BEG_Geometrie_Complete, drv_BEG_Numero_Piece = 1), drv_BEG_Numero_Piece = 2)
vis3 (par défaut)	☑	= or(and(drv_BEG_Geometrie_Complete, drv_BEG_Numero_Piece = 1), drv_BEG_Numero_Piece = 3)
vis4 (par défaut)	☑	= or(and(drv_BEG_Geometrie_Complete, drv_BEG_Numero_Piece = 1), drv_BEG_Numero_Piece = 4)
vis5 (par défaut)	☑	= or(and(drv_BEG_Geometrie_Complete, drv_BEG_Numero_Piece = 1), drv_BEG_Numero_Piece = 5)
vis_BEG_Visibilite_Guidages (par défaut)	☑	= and(vis1, BEG_Visibilite_Guidages)
xC (par défaut)	324.3	= (La / 2 + Rayon raccord) * cos(A_angle_elevation) - (epl + Rayon raccord) * sin(A_angle_elevation)
y_C (par défaut)	194.0	= (La / 2 + Rayon raccord) * sin(A_angle_elevation) + (epl + Rayon raccord) * cos(A_angle_elevation)
zd_ajstm (par défaut)	101.6	= zdx * cos(-A_angle_elevation) + zdy * sin(-A_angle_elevation)

Figure 17. A portion of the many parameters of a complex Revit family

5. Conclusion

The rise of parametric tools has profoundly transformed our design and production methods in the engineering and architecture sectors. With software like Grasshopper, Karamba, and Rhino Inside Revit, it is now possible to optimise processes by automating certain tasks, streamlining the transfer of data between different stages of a project, and reducing the time spent on manual adjustments.

The examples discussed—from the canopies of Legiapark to the Abidjan tower, and the Vilvoorde viaduct—demonstrate that these tools go beyond modeling. They actively contribute to streamlining construction choices, optimising structures, and efficiently integrating BIM. By enabling quick exploration of multiple scenarios and facilitating collaboration between engineers, architects, and modelers, these technologies become true performance enhancers.

The realm of possibilities is constantly expanding. Fields like energy and environmental management, in particular, can benefit greatly from parametric design, which enables more adaptive, efficient, and innovative solutions.

However, their full adoption depends on advanced mastery and the continuous evolution of practices. The challenge is no longer just to produce accurate models, but to strategically leverage these models to improve the quality, sustainability, and efficiency of projects. The future of parametric design lies in balancing automation with human expertise, flexibility with rigor, to meet the ever more complex challenges of contemporary engineering.