# Towards a Parametric Variation of Floor Plans: a Preliminary Approach for Vertical Residential Buildings

Wallace Roberto Silva Dornelas[1], Andressa Carmo Pena Martinez[1]

[1] Universidade Federal de Viçosa, Viçosa, Brazil
Wallace.dornelas@ufv.br; andressamartinez@ufv.br

**Abstract.** In the context of the housing demands that respond to several family profiles, allied with the potential of the algorithmic approaches to Architecture, this paper aims to describe an exploratory process of possible solutions toward a generative system of housing distribution in vertical multifamily buildings. As a method, this work presents a parametric design process of a multifamily building, simulating a variety of shape solutions for apartment buildings, in a Grasshopper definition. The work also discusses the data transmission between the parametric modeling using Grasshopper in the Rhinoceros interface and the connection of the final design to Graphisoft's Archicad BIM-based software. As a result, the parametric model allows several design solutions for several building shapes and contexts. For this study, to fully explore the design possibilities, we applied the method in the context of a Brazilian metropolitan city.

**Keywords:** Generative design, Parametric model, Parametric architecture, Housing, Vertical buildings

## 1 Introduction

The traditional practice in Architecture is, by nature, customized according to the user demand. However, it is still focused only on a single final product. On the other hand, despite the advances in computational technologies such as algorithmic and parametric design, the Civil Construction sector remains more inert to manufacturing changes at a large scale related to digital fabrication, especially in South America. In this context, there is still a lack of practical experiences that apply the real possibilities that computational design brings to the practice of architecture in our days.

Before the First Industrial Revolution, the manufacturing process was predominantly artisanal. The production processes ended up being customized since it was not possible to establish quality control to produce identical objects from the hands of the same craftsman. According to Celani & Frajndlich (2016), with the emergence of specialized machines, mass production came on the scene, which means a large-scale production of identical objects. However, with the advent of computers, other computational design and manufacturing processes emerged. As Celani & Frajndlich (2016) state, this allowed the use of non-specialized machines to produce also non-identical objects, such as industrial robots and other computer-controlled machines (or CNCs). The versatility of the machinery makes it possible to explore new and even unique results, directed to specific problems and specific needs (Celani & Frajndlich, 2016; Celani & Pupo, 2008; Kolarevic & Duarte, 2019).

Meanwhile, since the 1960s, we have been discussing the terms digital design, computational design, and algorithmic design, among others (Caetano et al., 2020). These terminologies are nothing more than attempts to address the same theme, precisely the use of computers as a tool for design processes. According to Caetano, Santos, and Leitão (2020), in the last two decades, computational design techniques have been applied in Architecture more focused on new approaches based on computing, such as simulations, optimizations, and new manufacturing methods. Also, according to them, the explorations of computing in architecture have gone deeper through design procedures automation, distributing project tasks and managing large amounts of information, dealing with changes in projects quickly and flexibly, and assisting in form-finding processes through automated feedback.

Another aspect of our questions wanders about the Open Building concept. Defined and deeply studied by John Habraken (1961) and Stephen Kendall (1999), Open Building is a design approach that tries to make possible for long-term usage of the building, for different family profiles as the years pass by, and for the same family as they change their needs due to the passing of time. To accomplish that, they think of a building as composed of several layers, which are grouped into two main levels: the Support (or Skeleton) and the Infill (Avalone & Fettermann, 2020; Habraken, 1961; Kendall, 2004). The Support is supposed to be the most "solid" parts of the building, such as structure, accesses, and common circulation areas, as the Infill can be the interiors of each housing unit, internal walls, and even plumbing systems, that can be designed more flexibly, as to allow for future interventions (Kendall, 2004). In Open Building, these different layers take into account their lifecycle separately, and not the building's as a whole, and that can be used to design each of the layers as separate parts of a complex design solution.

In this context, this paper shows a possible application of an algorithmic and parametric design toward a generative system for housing in a vertical residential building. In addition to the demand for a variety of housing units for different family profiles, the vertical building has some constraints such as water and sewage installations, and vertical structural elements, among others, that we need to overcome to achieve different solutions. That is where the Open Building comes in, as we can structure our building according to an open building process, where we define what are Supports and Infills, and allow each of them to be treated and solved separately. As a premise, we seek to combine the algorithmic and parametric design to achieve several possible outcomes, structured algorithmically, to meet diverse demands on a large scale, more than only flexible solutions for standard floor plans.

However, this paper presents only a conceptual model, in the early stages of the design, to be used after all the studies concerning feasibility have already been made and before the definition of spaces and layout distribution. Our proposed model combines a generative design system through a graphical programming language (in Grasshopper) and a BIM-based model (in Archicad) of a possible solution to a vertical residential building.

This study also does not involve the end-users or any platform for public interaction or design choice. However, the code allows solutions suitable for different inputs according to several family profiles. As a case study, we have applied and tested the algorithmic definition in the context of a metropolitan city in Brazil. The aim was to explore a variety of apartments according to different resident demands.

The contribution of this work relies on a take on a code to help in the design of apartment plans on vertical buildings, which is an approach with a few examples from literature. One of those is the customization approach from Veloso et al. (2018), but that is a study regarding shape grammars and an interface for users. We, at our actual stage, think of a code to help designers think of architecture on an Open Building approach, where the Support and Infill portions can be previewed and conceived without affecting the future flexibility necessary to Open Buildings.

## 2 Methods

The research methods follow four stages: 1) Analysis of some case studies of algorithmic-parametric processes to housing generative systems; 2) Development of a parametric model, using Grasshopper and Rhinoceros software; 3) the integration with BIM software Graphisoft's Archicad, to data extraction, documentation, and visualization of the virtual model. However, in this paper, we focus only on the third and fourth stages.

The Grasshopper definition, based on one example developed by Swedish Architect Jesper Wallgren (2020), and made available at his website (*finch3d.com*) allows: a) the design of the building shape or perimeter, such as L, I, U shapes, and several others, in a way it adapts to different site contexts and terrains sizes; b) Subdivision of each floor plan level in different apartment units, with size solutions ranging from 45m² to the total area of the floor; c) Internal subdivision of each apartment into rooms, according to the size of prefabricated and standard constructions components (drywall, steel-frame, and others); d) the subdivision of each apartment unit into “dry” and “wet” areas, to rationalize the plumbing system, according to the need for solutions of bathrooms, restrooms and kitchens; e) Definition of variations of porch and balconies to all the units; f) A preliminary structural launch, to preview the beams and columns positions, according to the building floor plan; g) Connection to a BIM-based software, to obtain data such as material, standard components, and documentation.

### 2.1 Parametric modeling

We defined some ground rules to develop the Grasshopper definition:

The units would follow a linear location in the floor plan, which references the perimeter of the building, determined by previous terrain analysis. For this purpose, the designer needs to provide only a curve-type geometry in Rhinoceros, containing the total length of the building and perpendicular to the division. As input, we can select the width of the units, and the code will distribute them in rectangular blocks, combined into different housing units.

To control the areas of each apartment, we used as a reference a subdivision algorithm created by the Swedish architect Jesper Wallgren, (available at *finch3d.com*) that generates adaptative plans. His grasshopper definition works based on the total area of a floor plan, subdivided into different parts according to a numerical factor. Then, we specified some target areas as the sizes we want to reach for each apartment. And then we have a fourth parameter, that randomizes and redistributes the list of targets so that we can verify different organizations for each housing unit. This improvement allows us to test different results for each plan and analyze the distribution according to the building shape or other architectural constraints. In other words, the code can take the divided pieces and merge them accordingly to the areas we provide as targets. In the following step, we can decide the number of levels or floors and adapt the functionalities to the particularities we face in each context.

Each apartment is divided into two parts: one related to wet areas, which concentrates the sanitary facilities and plumbing systems, and the opposite one is destined for areas of personal or social living, such as bedrooms, living rooms, and dining rooms. The sanitary facilities side concentrates the plumbing and is reserved for bathrooms, restrooms, kitchens, and laundries, close to the access and vertical circulation. The other side grants possibilities for balconies and allows more natural lighting and air circulation.

The wall’s surfaces follow an industrial standard drywall structure, 1.5 meters multiplier apart. This modulation of 1.5 meters can fit minimal bathrooms, and laundries, increasing to minimal bedrooms of 3.0 meters by 3.0 meters, and so on, up to larger dimensions for any room. With this starting point logic, the code can solve the first pre-dimensional steps and subdivide the floor area into multiples of 1.50 meters by 1.50 meters.

As a final step, it is possible to define a preliminary structural system, stating the parameters for the distance and dimensions of columns and beams. However, the code does not allow designing the layouts and furniture of the apartments, which could also be a design decision tool to personalize each housing unit.

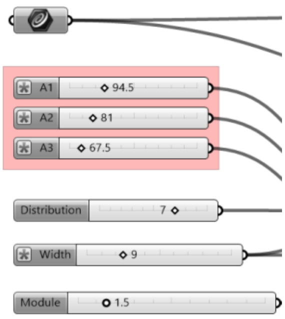


**Figure 1.** The image illustrates the main parameters as inputs to the algorithm. The geometry component is an input for the guide curve. “A1”, “A2” and “A3” represent the target areas, defined for each apartment. “Distribution” is the randomizer of the list of target areas after quantification. “Width” means the depth of the units. And “module” is the size of the prefab constructive components. Source: the authors.

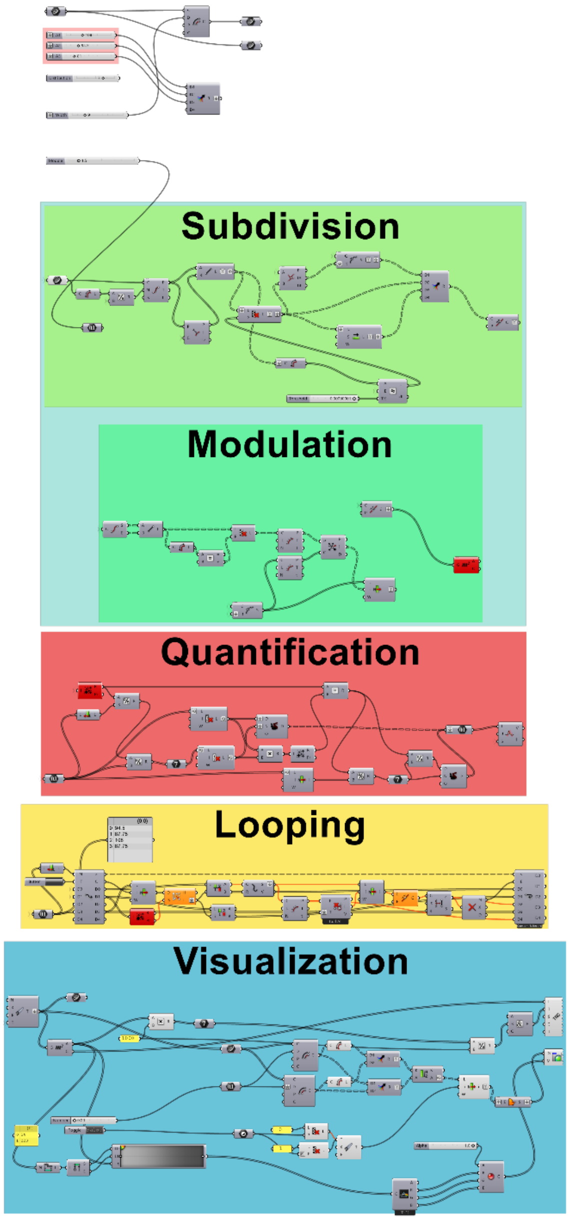


**Figure 2.** This Grasshopper definition allows the subdivision of each floor plan. The stages defined as subdivision and modulation organize the geometry according to the settings of Width and Module parameters. Quantification and looping derive from Jesper Wallgren's (2020) code. The first sums the target area and relate the number of them it will create in the "looping" stage. "Visualization" is used to define the visual tridimensional appearance of the resulting model. Source: the authors.

Once the base structural system could be defined through the algorithm, being the preliminary positioning of beams, columns, and slabs the main concern, in this case, we can say that this part can define the Support portion of the building, according to Habraken's (1961) and Kendall's (1999) definitions. In their studies, the Support is more than just the concrete portion of the building. It also includes the surroundings, the neighborhood, and the relationship of the residents with these elements (Avalone & Fettermann, 2020; Kendall, 2004). As to the Infill portion, which is not in the scope of this work, we think of it as being possibly solved by standardized solutions available in the construction market, such as drywall, wood, aluminum, and steel, and standard insulation and plumbing systems, as long as they can be easily assembled and disassembled as needed. All of these solutions already appear in other Open Buildings around the world (Kendall, 2017)

### 2.2 Data transmission to a BIM-Based software (Archicad)

As a means to keep the design process being fed by information concerning material costs and properties, we proposed the use of data transmission between the parametric modeling in Grasshopper to Graphisoft's Archicad, through the plugin Archicad Live Connection, as suggested by Veloso et al. (2018), in their study on the customized layout, due to its friendly interface and easy and fast translation of Grasshopper's code into Archicad's 3D model. As a first approach, to enable a lower-cost fabrication, we chose standard components, such as drywall and steel structures, already available in the construction industry. Even these discrete construction components are not customized, as digital fabrication processes such as 3d printing, the code allows several architectural solutions and combinations. Through this workflow, we can change the whole subdivision on each floor and update it in real time. The BIM model also updates all the information, representation, and statistics. This data transmission allows an interconnected system for sketching, evaluating, analyzing, reconfiguring, adjusting, and verifying every step in this workflow.

## 3 Results

We placed the building in a high-rise neighborhood, in a Brazilian metropolitan city, as a generative process requires diverse family profiles and specific solutions for the demands. We chose a lot where the legal restraints would allow a building of twenty floors, disregarding the parking and ground floors.

Following the dimensioning and the solar orientation studies, we defined the guide-curve as input into the algorithm, generating the first design possibilities. These possibilities will be directed towards the design of the Support, regarding

Habraken's (1961) Support Theory for Open Building, being considered as Support, in this case, the structural frame and the slabs for each housing unit. The Infill portion will then be able to be analyzed individually for each unit separately, in another stage of the design.

As the strategy is to attend to different family profiles, we tested several numerical values as inputs for the target areas. We decided the lower levels would have more units with smaller sizes, while the upper floors would have fewer units with larger sizes.

For the internal layouts, we defined a strategy to keep the bedrooms on the east side, while the wet areas, which concentrates on the water systems, on the other. All the rooms have natural light and air circulation as a premise. As the last step, we designed a system to create a series of balconies around the whole building, which follows the internal plan variety, on the façade. Following an alternated pattern of 1.5m or 3m advancement, the balconies draw a zigzag around each level, and the boundary angles change on the limits between apartments.

As a result, the algorithm allows us to explore different possible design solutions for the Support portion, within different building shapes. The architect can define the floor boundaries, which means the guide curve, after analyzing the site and the total area. The designer can also explore several possibilities before choosing the best ones according to some constraints or goals.

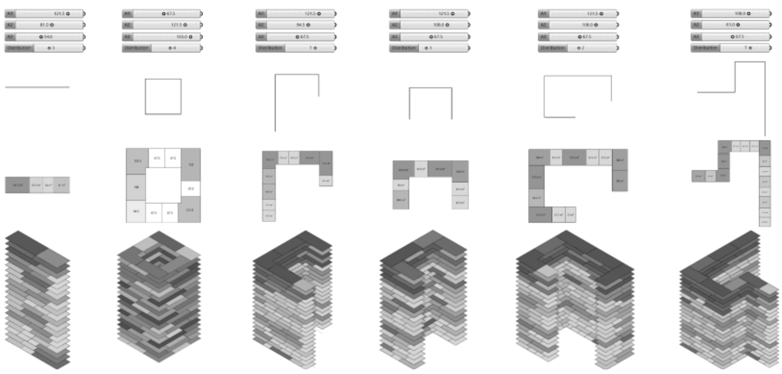

**Figure 3.** Some design possibilities. At the top, there are the target areas and distribution parameters. Right below it, there is the guide curve as the perimeter of each building. Then, an example of a floor subdivision in apartments and their vertical distribution is stacked in the shape of the building. Each different guide-curve results in different shaped buildings, so that the Support for said building can be studied and defined. After this stage, each unit can have its Infill portion designed individually. Source: the authors.

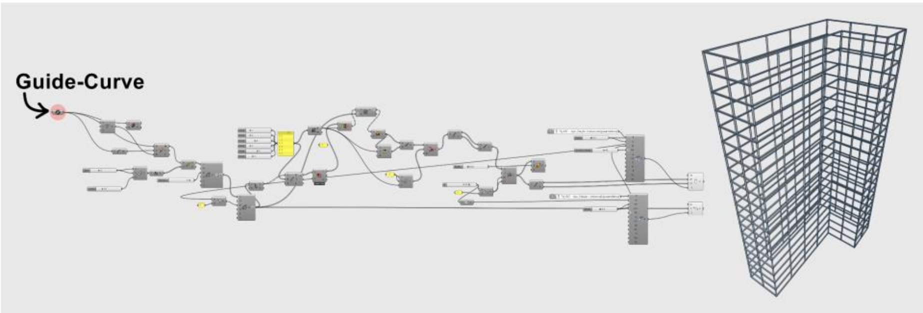


**Figure 4.** The schema of the structural elements, through the connection to Archicad. In this example, we show the shape of the guide curve for the selected urban context, in an L shape. The pillars and beams in this example are H and I profiles, chosen just as a generic representation of the possibilities. Source: the authors.

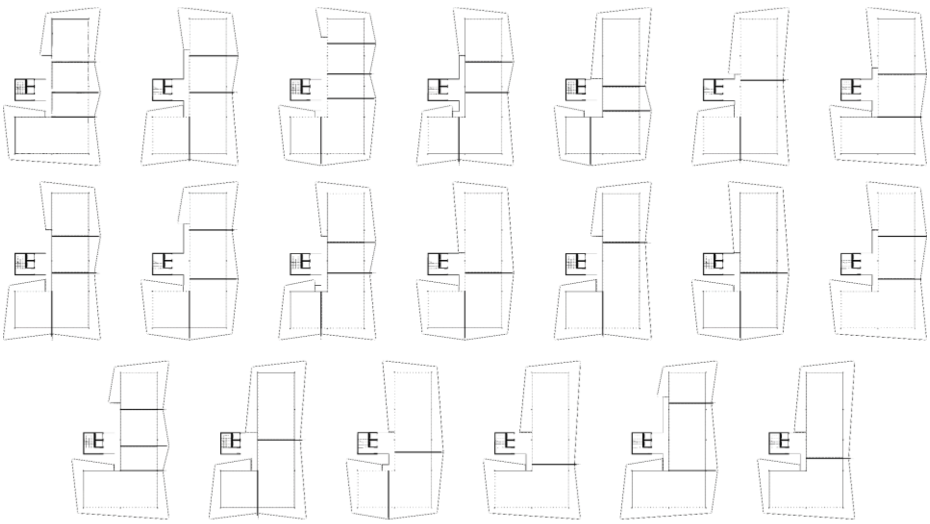

**Figure 5.** All twenty floors of the final building, generated in Archicad, were built with slabs and walls, and the balconies, were strategically designed to represent the variability of the plan and apartment sizes. Source: the authors.

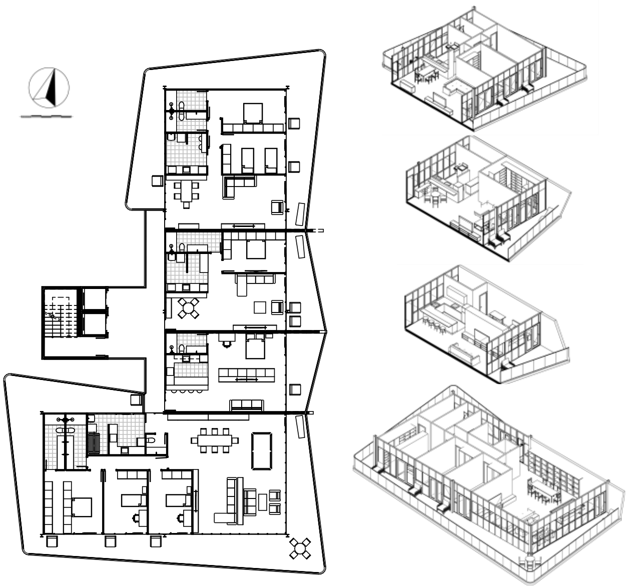

**Figure 6.** On the left, the first floor of the building, with a layout study, each room with a possible solution according to the unit size, thought to attend to the different family profiles. To the right, perspectives of the apartments for a better understanding of the created space. Source: the authors.

**Figure 7.** A final perspective of the building and the variability of the terraces, as they follow the floor plans. Source: the authors.

## 4 Discussion

Far from constituting an answer to the topic, this article aimed to shed light on potential applications, in the conceptual design stage, through an algorithmic process, towards a system that generates guides to possible solutions in architecture. Not as a tool to find a perfect design, but as a tool to help architects to study and analyze more options than what could be achieved manually. In that sense, we believe computational design can improve the decision-making process by helping architects think more about design rules than the design itself.

Placed between the feasibility study and the design of the internal layout, we believe that our code can help architects to visualize several different solutions for vertical buildings, instead of the usual typological plans, that don't work for every family profile, nor every people's needs. And, as an "in-between" stages tool, ours can have its results being used as input for other tools, that generates house and apartment plans, based on an initial geometry of the outline of the unit. Some of these tools are already available as plugins for Grasshopper and can be found at *www.food4rhino.com*. Testing how these tools can be connected to our own is something that we might want to explore in future works.

As a work in progress, there are still some limitations. The first improvement would focus on adding a function to exclude the results that have some problems regarding access, circulation, or openings. Another interesting addition would be a system to draw a preliminary approach to the plumbing system, connecting the "wet" areas and optimizing them.

For future works, we also aim to design a more user-friendly platform to help the decision makings. This platform could also allow architects to manipulate and test the parameters and give visual feedback about the tried solutions.

**Acknowledgments:** We thank the Graduate Program in Architecture and Urbanism (PPG.au), the Department of Architecture and Urbanism at the Federal University of Viçosa (DAU/UFV), the Coordination of Superior Level Staff Improvement (CAPES), and the Foundation for Support to Research from the Estate of Minas Gerais (FAPEMIG), for the support, funding and overall aiding in this work.

## References

Avalone, M. C., & Fettermann, D. de C. (2020). Demandas de customização em edifícios residenciais no Brasil: uma análise de agrupamentos empírico-qualitativa com base no sistema Skeleton/Infill. *Ambiente Construído*, *20*(2), 489–503. https://doi.org/10.1590/s1678-86212020000200411

Caetano, I., Santos, L., & Leitão, A. (2020). Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, *9*(2), 287–300. https://doi.org/10.1016/j.foar.2019.12.008

Celani, G., & Frajndlich, R. U. de C. (2016). From prototypical to prototyping: mass-customization versus 20th century utopias in architecture and urban design. *PARC Pesquisa Em Arquitetura e Construção*, *7*(3), 160. https://doi.org/10.20396/parc.v7i3.8647348

Celani, G., & Pupo, R. T. (2008). Prototipagem rápida e fabricação digital para arquitetura e construção: Definições e estado da arte no Brasil. *Cadernos de Pos-Graduação Em Arquitetura e Urbanismo*, 31–41.

Kendall, S. (2004). An Open Building Strategy for Achieving Dwelling Unit Autonomy in Multi-Unit Housing. *Housing and Society*, *31*(1), 89–102. https://doi.org/10.1080/08882746.2004.11430500

Kendall, S. (2017). FOUR DECADES OF OPEN BUILDING Realising Individual Agency in. *Architectural Design*, *87*(Issue 5), 54–61.

Kolarevic, B., & Duarte, J. P. (2019). *Mass Customization and Design Democratization*. Routledge.

Veloso, P., Celani, G., & Scheeren, R. (2018). From the generation of layouts to the production of construction documents: An application in the customization of apartment plans. *Automation in Construction*, *96*(September), 224–235. https://doi.org/10.1016/j.autcon.2018.09.013

Wallgren J. Finch 3D website - Templates for download. Retrieved October 14, 2020, from https://finch3d.com/