TÉCNICO LISBOA

Generative Design for Energy Efficiency

Energy Analysis and Optimization

Catarina Rodrigues Ferreira da Rocha

Thesis to obtain the Master of Science Degree in

Architecture

Supervisors: Prof. António Paulo Teles de Menezes Correia Leitão

Examination Committee

Chairperson: Prof. Manuel de Arriaga Brito Correia Guedes
Supervisor: Prof. António Paulo Teles de Menezes Correia Leitão
Members of the Committee: Prof. Ana Paula Filipe Tomé

June 2017

Acknowledgments

First I would like to thank my advisor António Menezes Leitão, for all his support and never ending patience to endure all the setbacks and all the little and big details where I needed his help.

I would also like to thank my advisor's research group, ADA, for all the support, with a special thanks to Carmo Cardoso and Bruno Ferreira.

I also thank Prof. Maria da Glória Gomes, who helped overcome the challenges regarding energy simulations.

To my family, specially my parents, thank you so much for putting up with this never ending work to get my degree. Without your support, I would have never reached this point.

To all my beloved friends, because you are super awesome and always support me. Bukkatuna rules and you are the very best friends I could wish for. And of course, my sweet girls spread all over the world, thank you so much for everything. Thank you all, for your patience with this programming-loving soon-to-be architect. You are the reason why I managed to get here. Your company in the good and bad times, all the talks, the drinks, and the parties. All those times I reached for your help because the script was not working, or because my macbook does not handle some software. Thanks for all the supportive talks and for all the mocking ones. They got me here, and I hope they will never end.

A special thanks to Joana Cabral, Rui Morais, and Catarina Dente, for you are my very best friends and you have endured all my suffering by my side. You, most of all, brought me this far!

And finally, to the most supportive and loving person in my world, the most special thanks to Pedro Soares, because without you, this would not have been possible.

Abstract

Energy efficiency is one of the core topics in Architecture nowadays. The research here presented reinforces the importance of the use of building energy analysis throughout the entire design process, rather than just running the analysis when the model is finished. By using Generative Design, the architect can explore numerous different design options in a short amount of time, and can analyse and compare all these options using the available energy analysis tools.

In the present work, we use Rosetta as a model generating tool, Autodesk Revit as a model visualisation tool, and Autodesk Insight 360 as an energy analysis tool. A workflow is proposed, using Generative Design to produce multiple design variations from the initial concept of the case study presented. The results retrieved from Insight 360 are then used to improve that design.

Keywords

Generative Design; BIM tools; Energy Analysis; Energy Optimization.

Resumo

Eficiência energética é um dos temas centrais da Arquitectura nos dias de hoje. O trabalho aqui apresentado reforça a importância da análise energética ao longo de todo o processo de concepção de um edifício, em vez de apenas a realizar no final. Ao usar Desenho Generativo, o arquitecto consegue explorar inúmeras opções do modelo num curto espaço de tempo, e consegue analisar e comparar todas essas opções através das ferramentas de análise disponíveis.

Neste trabalho, usamos a ferramenta Rosetta para gerar modelações tridimensionais do edifício, o programa Autodesk Revit para visualizar esses modelos, e a plataforma Autodesk Insight 360 como ferramenta de análise energética. É proposta uma metodologia de trabalho que usa o Desenho Generativo para produzir múltiplas variações do edifício a partir do seu conceito inicial. Os resultados obtidos a partir do Autodesk Insight 360 são depois usados para melhorar o projecto.

Palavras Chave

Desenho Generativo; Ferramentas BIM; Análise Energética; Optimização Energética.

Contents

List of Figures

List of Tables

Acronyms

ASHRAE American Society of Heating and Air-Conditioning Engineers

BIM Building Information Model(ing)

CAD Computer-Aided Design

EUI Energy Use Intensity

GA Genetic Algorithm

GD Generative Design

IFC Industry Foundation Classes

Na maximum value for annual nominal needs of useful energy for the preparation of hot sanitary water

Nac annual nominal needs of useful energy for the preparation of hot sanitary water

Ni maximum value for annual nominal needs of useful energy for heating

Nic annual nominal needs of useful energy for heating

Nt maximum value for global annual nominal specific primary energy needs

Ntc global annual nominal specific primary energy needs

Nv maximum value for annual nominal needs of useful energy for cooling

Nvc annual nominal needs of useful energy for cooling

NZEB Nearly-zero Energy Buiding

RCCTE *Regulamento das Características de Comportamento Térmico dos Edifícios*

RECS *Regulamento de Desempenho Energético dos Edifícios de Comércio e Serviços*

REH *Regulamento de Desempenho Energético dos Edifícios de Habitação*

TPL Textual Programming Language

VPL Visual Programming Language

ZEB Zero Energy Building

1

Introduction

Contents

Architectural design consists of presenting a solution to a given problem. This problem, even with its various constraints, such as site, climate, construction cost, and regulations, has a plethora of possible solutions. Being so, designing implies selecting amongst several viable solutions, depending on the criteria applied, in order to improve the final solution. 'Design can be seen as an evolutionary process'. [1] As environmental sustainability becomes a core topic of our society, architects are lead to design buildings with better energy performance, in order to achieve a more sustainable architecture.

Climatic conditions have always been seen as an important factor to take into account, since buildings were shelters from rain, wind, and cold. As the architect Philip Johnson once said, 'All architecture is shelter'.

As time passed, people started to seek comfort, instead of only a shelter, and concerns with the right amount of lighting, ventilation, and warmth started to emerge. Providing a comfortable living space started to be the main objective, thus acting as a modifier of the outdoor climate. This means that the energy demand of a building becomes an important characteristic. One type of energy demand is sunlight, since it can illuminate and provide natural heating. The orientation and shape of the building can influence indoor thermal comfort as well, factors taken into account already by ancient civilizations. [2] This modification started has a passive one, relying on the building itself, its orientation, and materials used, and only later an active approach appeared, when technologies that allowed for indoor ventilation and thermal control emerged.

Industrial and economic revolutions made it possible to develop new technologies and to explore and discover new materials and construction methods. All of this contributed to a better comfort in built areas, both public and private buildings. Architects started taking into account particular thermal efficiency calculations bearing in mind factors like the building's orientation, materials, etc., with the intent of measuring whether the environment created would be comfortable or not.

With current technologies, most of this comfort is achieved through machines which implies spending a lot of energy. If there is a machine that does the work, why bother spending time and money in careful designing and expensive materials? This kind of thinking, alongside with a rapid technological evolution and increasing construction demand, highly contributed to environmental pollution and destruction, by simply relying on active indoor environment control, instead of taking advantage of a passive approach to reduce energy consumption. 'Along with amazing technological advances, the Industrial Revolution of the mid-19th century introduced new sources of air and water pollution. By the middle of the 20th century, the effects of these changes were beginning to be felt in countries around the world.' [3]

To fight this growing problem, regulations were created, ensuring a sustainable consumption of energy and resources. Regulations emerged from international conferences and charters, like the International Energy Charter, [4] and now each country has its own building regulations, based on international directives. These international directives are being constantly updated, in order to promote a better and more sustainable way of living, e.g., the directives regarding the European Union goals for 2050, regarding sustainability.[1]

The energy performance of a building is determined by its response as a *complete system* to the external environment and the internal environmental demands of the occupants. The system response, in turn, depends on the combination of individual attributes that have been assembled to produce the building. [5]

An energy efficient building is the 'one that uses the minimum necessary energy to be built and used'. [6] Ideally, a building should be auto-sufficient, i.e., it should have methods of reducing energy consumption, of reusing energy and resources, and even of producing its own energy. Being able to design energy efficient buildings will not only bring environmental benefits, but also economic ones, since it reduces consumption and thus the economic burden of a building, compensating for the initial investments. We can already see examples of this type of buildings, with the *Passivhaus*, a german word that means passive house. As described in the Passive House Institute website, a 'Passive House is a building standard that is truly energy efficient, comfortable and affordable at the same time'.[2]

To promote energy-efficient buildings, regulations were created, ensuring a sustainable consumption of energy and resources. Due to these regulations, it is mandatory to ensure certain consumption limits when designing a new building.

Being so, architects need to ensure that their designs fulfill these particular energy requirements before being built. To do so, there are a set of manual calculations, that according to the area's climate, material, and building morphology will give the architects and engineers an estimation. However, these calculations are very exhausting and error prone. Fortunately, in recent years, some computational tools to substitute these calculations were developed. These tool made the entire process process faster and are a source of encouragement for architects to explore better performing designs. An example of such tools is *Autodesk Insight 360*, a simulation engine used to perform energy analysis in *Autodesk Revit*, among many other alternatives. Some of these tools also allow for simulations throughout the different stages of the design process, as is the case of *Autodesk Insight 360* and others, like *DIVA*, a simulation engine that works with *Rhinoceros 3D* software. Hence, the user is able to perform a general analysis on a more conceptual stage of the work, and more detailed ones in the final stages. This possibility allows for a better notion of the foreseeable energy consumption of a building, even in initial stages, and thus a more sustainable design is promoted since early stages of the design process.

In parallel, there is also a growing body of knowledge on parametric modeling, which considerably helps architectural practice, allowing a faster method of reproducing and changing a three-dimensional model of a building. Parametric modeling consists in describing an object through different parameters and relationships between those parameters, which enables variations on the model itself. [1] A mathematical

[1] https://ec.europa.eu/clima/policies/strategies/2050˙en
[2] http://passivehouse.com/

approach is used to describe certain aspects of the general design intent, creating functions describing the object. The values applied to the different functions allow them to create several design options, all following the same principles and deriving from the same parameters. This approach can generate a wide variety of design options in a short amount of time.

By combining a parametric approach with an efficient evaluation system, architects can find better design options, having in mind the energetic behavior, even in initial stages of the design process. This work proposes a **parametric approach** to the creation, evaluation, comparison, and improvement of models, regarding their **energy consumption**. Therefore we will guide the design towards reducing the energy consumption, making this a performance-oriented design exploration.

1.1 Objectives

This work aims to research a more efficient method of evaluating and optimizing energy consumption in architecture.

On a first approach, we will understand how the simulation software works, its accuracy and the analysis of the results. This will help us to better grasp the workflow of the software in use for the second phase of the project, where a project will be analyzed and improved in terms of energy performance. Different simulation software will also be compared so that one can be selected for further studies.

Two buildings will be analyzed. The first one will be a family house, as it is an easier project to simulate, by being a single-family house with only one floor and with few rooms. Afterwards, a more complex building will be analyzed, this one projected by parametric generation and not by the traditional method. This will allow us to understand the advantages of Generative Design (GD), such as time consumption and detailing capacity, and how the combination of GD and energy analysis tools can be advantageous.

After having the energy simulations, an optimizing approach will be used, based on observation and comparison of different model options and their respective analysis. The optimization process will regard the energy efficiency of the building, creating new and improved versions, until it reaches the best possible solution within the tested spectrum of hypothesis.

1.2 Methodology

The work process will start with a literature review, exploring energy efficiency in architecture and the use of Building Information Model(ing) (BIM) tools. There is also an emerging use of GD, an approach used in this work and that is going to be introduced as a new workflow for architectural design. Finally, the connection of BIM tools with energy analysis software will be presented, introducing the main objective of this research: optimizing the energy analysis workflow in architecture.

To start the simulation phase, two case studies were chosen, one to make a quick simulation and the second to be generated with a parametric approach. These two will be tested and the energy simulation results analyzed. The main goal of the first simulation will be to better grasp the process of energy analysis in architecture with current technologies, by comparing the mathematical method to the results provided by the available energy simulation tools, and the second simulation will demonstrate the improvements in efficiency provided by the combination of GD and energy analysis tools. The workflow proposed is illustrated in Figure 1.1.

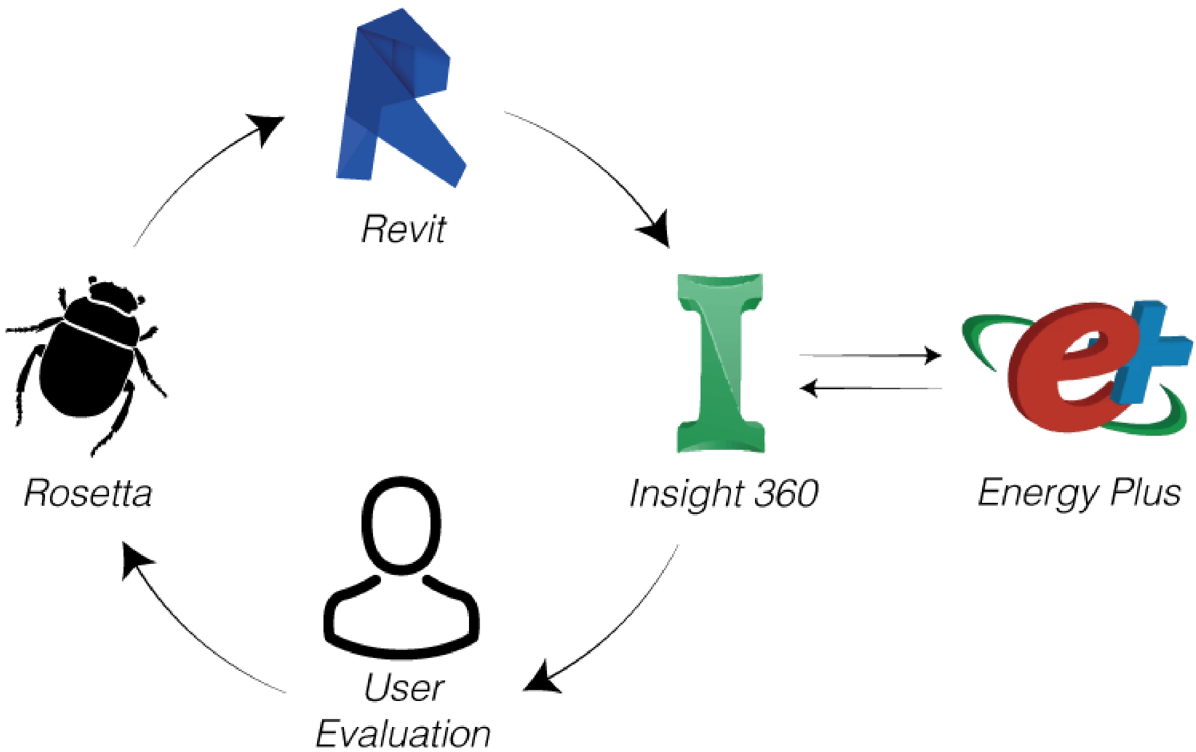


Figure 1.1: The workflow proposed in this research. A script is written, and with the use of *Rosetta*, the model is generated in *Autodesk Revit*. The model is then analyzed by *Autodesk Insight 360*, and the results read by the user, who will make changes in the script according to the results provided, and a new model generated. The loop ends when the user decides that the result achieved is adequate.

1.3 Structure

This thesis is organized as follows:

Chapter 1 makes an introduction to the main directives of this research. This chapter has three subchapters referring, respectively, to the objectives of this research, the methodology used, and the structure of the thesis.

Chapter 2 is a review about energy consumption in Architecture, introducing some important references to the existing methods of performing energy analysis. In this chapter, a Case Study is presented to support the comparison of different methods. The concept of GD is also introduced in this chapter. The contents are divided in the following subchapters: Subchapter 2.1 makes an introdution to the energy analysis method; Subchapter 2.2 presents Case Study I, that will be used as a base for the comparison of different energy analysis methods; Subchapter 2.3 explains the mathematical method for energy analysis, also comparing the results obtained from that method to the results produced by the energy simulation tools; finally, Subchapter 2.4 makes an introduction to the method used in the next chapter, introducing the concept of GD.

In Chapter 3, a new Case Study is used to demonstrate the advantages of GD in Architecture modeling and energy analysis. The contents of this chapter are divided as follows: Subchapter 3.1 introduces Case Study II, the one that will be used to explain the workflow proposed in this research; in Subchapter 3.2, the workflow proposed is presented and explained in detail, referring the results obtained and the challenges that emerged; in Subchapter 3.3, an automatic approach to this workflow is proposed, explaining all the factors that need to be taken into account to implement said automation.

Chapter 4 presents the conclusions drawn from this research in Subchapter 4.1, and proposing some areas of interest for future work in Subchapter 4.2.

2

State of the Art

Contents

In this chapter, we will address some important concepts about energy consumption in architecture. We will start by understanding what energy efficiency means in the built space, and also how to measure it. We will also explain the mathematical process behind the energy analysis tools, using as a reference the Portuguese regulations.

Knowing the basis of this process, we will look at different energy analysis tools, their main workflow and their pros and cons. Having this in mind, we will ascertain the reasons for using the chosen software for this research.

Finally, we will talk about Generative Design (GD), a growing subject within architecture, that has a lot to offer to this art and its own development. GD will be the base of the workflow proposed in this research, due to the advantages it brings to architectural design.

2.1 Energy Efficiency in Buildings

2.1.1 An introduction

In an era where protecting the environment and reducing Humanity's footprint are the main concerns, energy efficiency is a central topic in Architecture nowadays.

According to the International Energy Agency, 'energy efficiency is a way of managing and restraining the growth in energy consumption. Something is more energy efficient if it delivers more services for the same energy input, or the same services for less energy input.' [6] When applying this principle to buildings, one should use as less energy as possible to ensure a comfortable environment, ideally producing enough energy to sustain itself, through energy collecting systems, thus achieving what is referred to as a Zero Energy Building (ZEB).

> 'A zero energy building (ZEB) produces enough renewable energy to meet its own annual energy consumption requirements, thereby reducing the use of non-renewable energy in the building sector. ZEBs use all cost-effective measures to reduce energy usage through energy efficiency and include renewable energy systems that produce enough energy to meet remaining energy needs. There are a number of long-term advantages of moving toward ZEBs, including lower environmental impacts, lower operating and maintenance costs, better resiliency to power outages and natural disasters, and improved energy security.' [7]

We can also talk about Nearly-zero Energy Buidings (NZEBs)[1], as referred by the European Commission. Over the years, some directives were approved by them, having as main legislations the *Energy Performance of Buildings Directive* [8], in 2010, and the *Energy Efficiency Directive* [9], in 2012. There was also an

[1] https://ec.europa.eu/energy/en/topics/energy-efficiency/buildings/nearly-zero-energy-buildings

update of the *Energy Performance of Buildings Directive* [10], in 2016.

Renewable energies, alongside with a change in design and materials used, all work to a better energy performance of buildings. This not only improves the environment, but it also translates into a long term economic gain.

2.1.2 Energy rating of buildings

To ensure efficient buildings, there are regulations, as seen previously, at national and international level, regarding the construction of new buildings and renovation of existing ones. As stated in the International Energy Charter (based on a European charter), the main goal to consider is 'recognizing the global challenge posed by the trilemma between energy security, economic development, and environmental protection.' [4] Every country has its own regulations regarding construction, but they all follow international directives in order to achieve sustainability. The reference Portuguese regulation for habitational buildings is the *Regulamento de Desempenho Energético dos Edifícios de Habitação* (REH), which means *Habitational Buildings' Regulation for Energy Performance*.

In Portugal, there are energy performance classes defined to rank buildings, new and existing ones. These classes consider a set of parameters and values, which let us evaluate if a building is sustainable. There are 8 classes (A+, A, B, B-, C, D, E, and F), where A+ corresponds to a better energy performance, and F to a worse one. In new buildings, these classes only vary between A+ and B-, while existing buildings can have any class. Calculating a building's class depends on its typology, whether its function is an habitational one or a building for public use, like schools, hospitals, or others. The general calculation for habitational buildings consists in dividing the global annual nominal specific primary energy needs (Ntc) by the limit value of those same needs, the maximum value for global annual nominal specific primary energy needs (Nt). This division gives us the energy class ratio (R_{Nt}), [11] as we see in equation 2.1. The classes are then attributed according to table 2.1. We will understand how to calculate the referred values further along this work.

$$R_{Nt} = \frac{Ntc}{Nt} \tag{2.1}$$

Where:

R_{Nt} is the energy class ratio;

Ntc are the global annual nominal specific primary energy needs;

Nt is the maximum value for global annual nominal specific primary energy needs.

Energy Performance Class	R_{Nt} value
A+	$R_{Nt} \leq 0,25$
A	$0,26 \leq R_{Nt} \leq 0,50$
B	$0,51 \leq R_{Nt} \leq 0,75$
B-	$0,76 \leq R_{Nt} \leq 1,00$
C	$1,01 \leq R_{Nt} \leq 1,50$
D	$1,51 \leq R_{Nt} \leq 2,00$
E	$2,01 \leq R_{Nt} \leq 2,50$
F	$R_{Nt} \geq 2,51$

Table 2.1: R_{Nt} value ranges, to determine energy performance classes for habitational buildings. [11]

The calculations needed to ascertain the referred values will be explained further on, in the first section of Subchapter 2.3.

2.2 Case Study I

2.2.1 Building Specifications

The first building chosen was a family house in *Azenhas do Mar, Sintra*.

The proposed project consists in merging four small blocks to a bigger, central one. The difference in heights allows identifying a space hierarchy. Therefore, in the initial volume with a higher floor height, lies the core of the house: the living room. Another space considered as an essential one, as living space is concerned, is the office and library room. While the living room is the main social space, the office is the core of a more personal living experience, of silence and meditation, and that is why it stands out from the house, in a suspended volume facing south, opened to the outside. The remaining blocks feature the house's functional spaces. The western block contains the main bedroom and two other standard bedrooms, facing the villa and the ocean, every one of them equipped with its own sanitary facilities. In the northern block, we can find the kitchen, as well as the dining room, presenting itself as the connection between the kitchen and the living room. In that block we also have a pantry and an extra bedroom. The eastern block is the garage, with a storage and laundry room. For further details, the technical drawings are available in the Appendix A.

The concept derived from the idea of visual space hierarchy, creating a set of elements that live around the core of the house. The house started as a nucleus, a central regular space where the main living areas would be localized. Then a set of four blocks were added in each corner of the central one, allowing for the different functions of the house to live towards its core. Finally, there was an intention to distinguish the two main living areas, the living room and the office. While the living room is the living heart of the house, the main social space, the office intends to be the core of a more personal experience. To achieve this differences, maintaining their hierarchy comparatively to the rest of the house, the heights were changed. The living room is higher than the rest of the house, promoting an open concept, while the office closes within itself and stands out, with a lower height and risen from the ground.

The house is projected outwards, with large glazed areas towards the ocean and the village, but also looks within itself, with a small glass garden next to the living room.

In Figure 2.1 we can see the evolution of this concept. In Figure 2.2, there is a view of the house, as well as its plan.

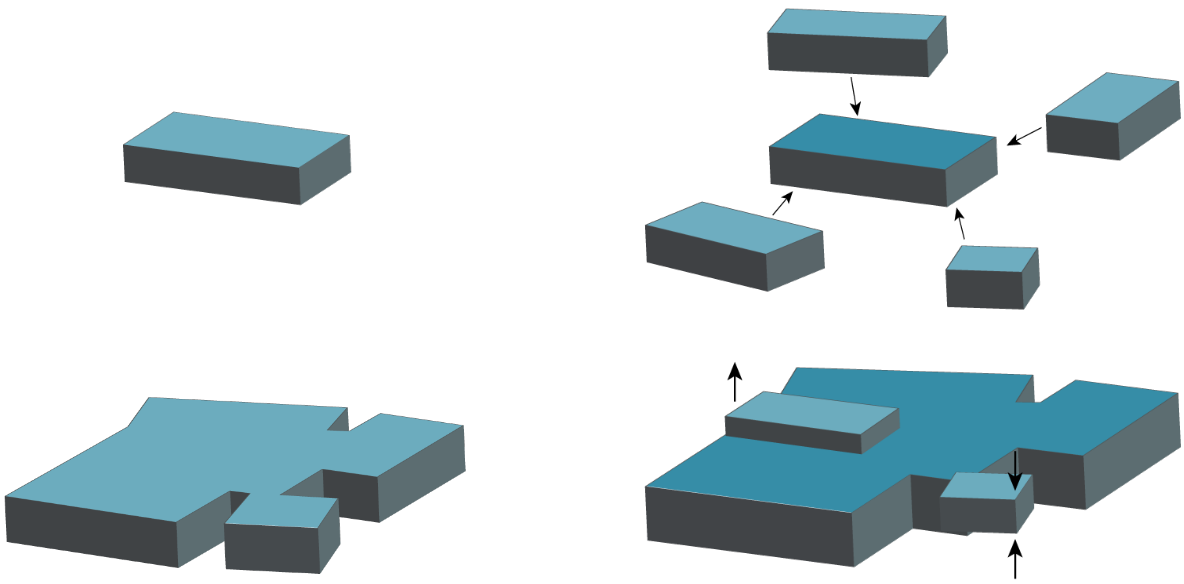
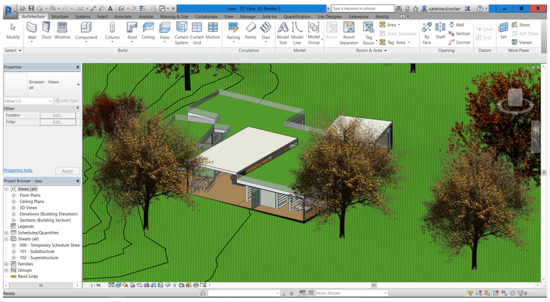


Figure 2.1: The evolution of the conceptual design for Case Study I - On the top left corner, the initial core volume; on the top right corner, the addition of four volumes surrounding the core of the house; on the bottom left corner, the volume resulting from the additions; on the bottom right corner, the manipulation of heights to create a space hierarchy.



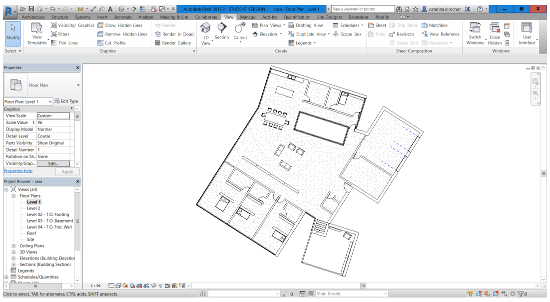


Figure 2.2: On the left, a three-dimensional views of the model in Autodesk Revit 2017. On the right, a plan of the house.

This building was chosen for being an accessible project, as drawing an analysis are concerned, once it is a family house with only one floor. It was also a good choice regarding function, being an habitational one, because that is the type of environment that we also want to test later on, in the next case study.

2.3 Energy Analysis Tools

In this section, the main subject will be how to perform an energy analysis to a building. There are several options available to architects, some of them more time consuming than others, but also with different degrees of complexity and detail. We remind that the main goal of this section is to introduce some of the options available to architects nowadays, summarizing how to adopt each method, and providing an overview of some of the existing tools.

Firstly, we will look to the mathematical method, i.e., the several steps that one has to go through to calculate the annual energy needs of a building. For this, we used a case study, presented before as Case Study I, a family house in *Azenhas do Mar, Sintra*. This method is explained by using a Portuguese regulation called REH.

Afterwards, we will take a look at different tools that make these calculations. There are various options, depending on the complexity of the evaluation intended. For an early stage of design, it helps to make a less detailed but faster evaluation, just to grasp the kind of energy consumption a building will have. When the building is already in its final stage, a more complex and detailed energy analysis should be made, to adjust constructive details.

2.3.1 Mathematical method for energy analysis

The mathematical method for energy analysis consists in calculating the heat gains and losses, considering all necessary factors, such as location, orientation, materials, among others. The sum of the results of all elements of the building gives us its total heat gains and losses, which allows us to conclude if the building needs mechanical heating or cooling methods to compensate for too many losses or gains.

For this calculations, we used the REH, applied in Portugal, as the Case Study I is located in this country.

The REH was created in 2013, meant to be an update to the previous regulation, the *Regulamento das Características de Comportamento Térmico dos Edifícios* (RCCTE), which means *regulation of thermal behavior characteristics of buildings*. We can see the reasons for this update in the legislation that approved the REH:

> 'The update of the existing national legislation involves changes at several levels, featuring, first of all, the structural and systematic changes, joining in one document the matter previously regulated in three different documents, thus proceeding in a significant reorganization which aims to promote the conceptual and terminological harmonization and the ease of interpretation by the recipients of these regulations. In second place, the clear separation of the scope of application of REH and RECS,[2], the first exclusively regarding habitational buildings, and the last regarding commercial and services buildings, which facilitates the technical treatment and administrative management of the processes, simultaneously recognizing the technical specificities of each type of building in the matters that are most relevant to their characterization and improvement of energy performance' [12] (translated from the original).

We will now look at some of the calculations needed to ascertain energy performance of a building, based on the regulations referred before.

To know a building's energy consumption, we need to calculate its Ntc (equation 2.2). Afterwards, we need to ensure that this value does not exceed the Nt (equation 2.3). To calculate Ntc, we need to first know the building's heating energy needs (for the winter months), cooling energy needs (for the summer months) and hot water systems energy needs. The first two are defined as, respectively, annual nominal needs of useful energy for heating (Nic) (equation 2.4) and annual nominal needs of useful energy for cooling (Nvc) (equation 2.5). Hot water systems energy needs is the amount of energy needed to heat sanitary hot water, defined as annual nominal needs of useful energy for the preparation of hot sanitary water (Nac) (expressed in the equation as Q_a/A_p). The calculation of the Ntc value also takes into account the necessary electrical energy needed for the ventilation systems (expressed in the equation as W_{vm}) and consumed energy produced from renewable energy sources (expressed in the equation as E_{ren}), as well as the efficiency of the equipments (expressed in the equation as η_k) and the energy conversion factors (expressed in the equation as F_{pu}).

To calculate the limit value Nt, we also need the equivalent limit values for the other parameters, i.e., maximum value for annual nominal needs of useful energy for heating (Ni), maximum value for annual nominal needs of useful energy for cooling (Nv) and maximum value for annual nominal needs of useful energy for the preparation of hot sanitary water (Na). Knowing these values, we can calculate Ntc (equation 2.2) and Nt (equation 2.3).

[2] *Regulamento de Desempenho Energético dos Edifícios de Comércio e Serviços* (RECS) which means *Commercial and Services Buildings' Regulation for Energy Performance*

$$Ntc = \sum_j(\sum_k \frac{f_{i,k} \cdot Nic}{\eta_k}) \cdot F_{pu,j} + \sum_j(\sum_k \frac{f_{v,k} \cdot \delta \cdot Nvc}{\eta_k})F_{pu,j} + \sum_j(\sum_k \frac{f_{a,k} \cdot Q_a/A_p}{\eta_k})F_{pu,j} + \\ + \sum_j \frac{W_{vm,j}}{A_p} \cdot F_{pu,j} - \sum_p \frac{E_{ren,p}}{A_p} \cdot F_{pu,p} \quad [kWh_{EP}/m^2 \cdot year] \tag{2.2}$$

$$Nt = \sum_j(\sum_k \frac{f_{i,k} \cdot Ni}{\eta_{ref,k}}) \cdot F_{pu,j} + \sum_j(\sum_k \frac{f_{v,k} \cdot Nv}{\eta_{ref,k}})F_{pu,j} + \\ + \sum_j(\sum_k \frac{f_{a,k} \cdot Q_a/A_p}{\eta_{ref,k}})F_{pu,j} \quad [kWh_{EP}/m^2 \cdot year] \tag{2.3}$$

Where:

f_i are the heating energy needs supplied by the system in use;

Nic are the annual nominal needs of useful energy for heating;

η_k is the efficiency of the equipments;

F_{pu} is the energy conversion factor;

f_v are the cooling energy needs supplied by the system in use;

δ is a factor that depends on the heating gains usage, and can take the values of 1 or 0 according to that same use;

Nvc are the annual nominal needs of useful energy for cooling;

f_a are the energy needs for the production of hot water supplied by the system in use;

Q_a are the useful energy needs for the production of hot water;

A_p is the useful pavement area;

W_{vm} is the electrical energy needs for the ventilation systems;

E_{ren} is the consumed energy produced from renewable energy sources;

Ni is the maximum value for the annual nominal needs of useful energy for heating;

$\eta_{ref,k}$ is the reference value for the efficiency of the equipments;

Nv is the maximum value for the annual nominal needs of useful energy for cooling;

k ranges over the values relative to the systems in use;

j ranges over the values relative to all sources of energy, including renewable energy;

p ranges over the values relative to sources of renewable energy.

To calculate Ntc, we need the values for Nic (equation 2.4) and Nvc (equation 2.5).

$$Nic = \frac{Q_{tr,i} + Q_{ve,i} - Q_{gu,i}}{Ap} \quad [kWh/m^2 \cdot year] \tag{2.4}$$

$$Nvc = \frac{(1 - \eta_v) \cdot Q_{g,v}}{Ap} \quad [kWh/m^2 \cdot year] \tag{2.5}$$

Where:

$Q_{tr,i}$ is the heat transmission by conduction through the building's envelope;

$Q_{ve,i}$ are the heat losses due to the air renovation;

$Q_{gu,i}$ are the useful heat gains from lighting, equipments, occupants and glazed areas, during the heating season;

η_v is the heat gains usage factor in the cooling season;

$Q_{g,v}$ are the total heat gains of the building, during the cooling season;

Ap is the useful pavement area.

For these calculations to be made, there are several steps to take in account. First of all, a location must be ascertained. In this case, the family house is located in *Azenhas do Mar*, a village by the sea, near the capital. The village is part of the parish of *Colares*, near *Sintra*. This is important because locations affect their specific average temperature values, as well as the degree-days.[3]

Two of those factors are the climatic winter zone (I) and the climatic summer zone (V). The climatic winter zones are defined by the number of degree-days, in the base of 20ºC, and this corresponds to the heating season.[4] The climatic summer zones are defined by the average outside temperature during the conventional cooling season.[5] We will need this later on, to know the maximum admitted values for the surface coefficient of heat transfer and for solar maximum allowable factors in glazed areas. For the climatic winter zone (I), we use the degree-days to attribute the zone, between I1, I2, and I3, being the first with a lower value of degree-days and the last with the highest. For the climatic summer zone (V), we use the average outside temperature to attribute the zone, between V1, V2, and V3, being the first with the lower temperature and the last with the highest. The reference values can be seen in Table 2.2 and Table 2.3.

	Degree-days	
	Minimum	Maximum
I1	940	1500
I2	1510	2100
I3	2150	3000

Table 2.2: The definition of the three winter zones (I1, I2, and I3), according to the values observed in the tables available in RCCTE (values in $^\circ C \cdot days$).

	Average Outside Temperature	
	Minimum	Maximum
V1	26	31
V2	32	33
V3	34	37

Table 2.3: The definition of the three summer zones (V1, V2, and V3), according to the values observed in the tables available in RCCTE (values in $^\circ C$).

[3] Degree-days are a measure of how long, in days, does the outside air temperature, in degrees Celsius, is higher (in case of cooling degree-days) or lower (in case of heating degree-days) than a certain base temperature.

[4] The heating season is the conventionally defined group of months (January, February, March, April, May, October, November, December) in which the temperatures are lower than the comfortable average, and we need to use heating methods.

[5] The cooling season is the conventionally defined group of months (June, July, August and September) in which the temperatures are higher than the comfortable average, and we need to use cooling methods.

After these initial factors, others have to be taken into account. The display of the constructive elements, as well as their material composition, are key factors in energy performance. So after ascertaining the initial location related values, we start analyzing all the constructive elements of the building, like walls, floors, ceilings, openings, etc. Walls separating the exterior from the interior are usually the first ones to be analyzed, as well as glazed areas. These elements, alongside with the slab that separates the building from the ground and the roof are the elements that separate the inside space of the building from the outside. So, they are the boundaries of the house. It is also important to ascertain the difference between heated and non-heated spaces. Heated spaces are usually rooms and living rooms. They are the ones where the inhabitants usually spend most of their time, and need to be thermally comfortable. Non-heated spaces are the transitional ones, or the ones with secondary functions, like for example a garage or a laundry room. The walls that separate heated from non-heated spaces are also very important to take into account, as they are also boundaries, but in this case separating the spaces that have specific thermal values for comfort, from the ones that do not have those values.

For each constructive element, we need to ascertain certain key values, like dimensions, thickness, thermal characteristics of the different layers and the order of those same layers.

All these calculations take time and expertise to be performed, even in a simple project like the one being used at this stage. To help reach an approximate result, even without profound knowledge in this area, there are spreadsheets available, created by a portuguese institute called "ITeCons".[6] Their documents are available on their website,[7] and they were used to perform the necessary calculations. Using the spreadsheet, the calculations took an average of 5 hours to complete.

With these tools, we can ascertain the average consumption of this case study, even if in a simplified version of it. As referred before, this type of tools require a level of expertise that the average architect does not possess. As written in the REH, 'the operation of the various experts and entities involved is clarified and detailed, [...] within a context of rigor and demand, bound to a quality control' [12] (translated from the original). So, even though the concepts and general calculations are taught to architecture students, the calculations are never as accurate as if you had experts in the matter performing them, and that is why an expert is required in the final evaluation of the building.

One other factor to take into account is the fact that the spreadsheet available expects numerical and textual descriptions of the building, instead of a three-dimensional model of the building itself. Being so, we could be describing a wide variety of possibilities using the same values. Besides that, the values may not be exact, once approximations are automatically applied in early stages of dimensional descriptions, which may contribute to some differences later on. The advantage of having a three-dimensional model is the exact nature of the values being used.

[6] www.itecons.uc.pt
[7] www.itecons.uc.pt/p3e

Lastly, we are using a Portuguese regulation to perform the calculations. The regulation used is based on international directives, which means that it should be similar to other regulations. However, the small differences that exist might be enough to produce some differences.

Being so, we consider the result to be just an approximation to what the final value would be if it were done by an expert in the area, who would know in detail all the necessary considerations to take into account in order to prevent possible errors in the calculations. The final value will be presented further on, when compared to the alternative approach, this last one using the available energy simulation tools.

2.3.2 Modeling and analysis software

There is a wide variety of energy analysis tools available that comply with the existing regulations, some more complex than others, and requiring different levels of expertise. In this chapter, we will explain the basic workflow and main characteristics of each of the tools considered as relevant for this research. The criteria used for this selection was based on the capability of whole building simulation (instead of analyzing just one component, like lighting, air quality or other) and the portability of the input and output files. Since it is a building analysis, most tools need a three-dimensional model of the building. The ones that need only numeric information were excluded, as they were more restrictive in terms of the form of the building, as already seen before. Furthermore, buildings generated by GD usually have non-Euclidean forms, thus being difficult to describe correctly with only numerical values. We will present some tools that would be a great asset from the architectural point of view, i.e., from a conceptual to a more detailed phase of design, allowing the architect to design and redesign the building according to the energy analysis results in the different stages of the project.

For this work, some tools will be referred and succinctly explained, with the purpose of providing a rapid overview of the available options nowadays. We will start by presenting some important software in this field of studies, but that could not be used for the purpose of this work due to their workflow. Afterwards, we will look at some viable alternatives to the method proposed, justifying the choice of software that is being used in this research.

2.3.2.A Modeling software

Before looking at simulation tools, we will introduce some modeling and visualization tools, all of them with the possibility of adding simulation engines that work with these tools.

Revit[8]

Revit is *Autodesk*'s reference Building Information Model(ing) (BIM) tool. As a BIM software, it is able to apply semantics to the modeled objects, integrating in just one file all the necessary information

[8] www.autodesk.com/products/revit-family/overview

about the building. This way, *Autodesk Revit* supports multidisciplinary model design that allows for collaborative work inside design and construction teams.

Autodesk Revit works with building components, like walls and slabs. Unlike in Computer-Aided Design (CAD) tools, such as *AutoCAD* or *Rhinoceros 3D*, the user creates specific architectural elements, instead of abstract geometric ones. These components belong to *families*, where all the construction details are specified, including the thermal characteristics of the elements. For example, if I want to build a wall made of concrete, insulation and plaster, I just create a new wall family that has specific depth dimensions for each of these components, define what is the order of those same elements from the outside inwards, and it is even possible for the user to create its own material, with textures (for renders) and thermal conductivity (for thermal analysis), among many other characteristics.

Rhinoceros 3D[9]

Rhinoceros 3D is a three-dimensional modeling software that uses NURBS[10] curves, which allows the user to create free-form surfaces and solids.

SkecthUp[11]

SketchUp is a three-dimensional modeling software, a CAD tool. Similarly to other CAD tools, it allows for geometric modeling, but does not recognize the constructive essence of the elements drawn.

2.3.2.B Analysis software

We will now take a quick look at some options that were put aside due to the fact that they are a separate analysis software and mostly need to import files or to redesign models in the new software. Even having this constraint, these programs are amongst the most used ones by professionals in this field of studies, thus worth referring.

Energy Plus[12]

"EnergyPlus has its roots in both the BLAST and DOE–2 programs. BLAST (Building Loads Analysis and System Thermodynamics) and DOE–2 were both developed and released in the late 1970s and early 1980s as energy and load simulation tools. Their intended audience is a design engineer or architect that wishes to size appropriate HVAC equipment, develop retrofit studies for life cycling cost analyses, optimize energy performance, etc. Born out of concerns driven by the energy crisis of the early 1970s and recognition that building energy consumption is a major component of the American energy usage

[9] www.rhino3d.com

[10] NURBS are Non-Uniform Rational B-Splines, a mathematical representations of 3D geometry (*in* https://www.rhino3d.com/nurbs)

[11] www.sketchup.com

[12] www.energyplus.net

statistics, the two programs attempted to solve the same problem from two slightly different perspectives. Both programs had their merits and shortcomings, their supporters and detractors, and solid user bases both nationally and internationally.
Like its parent programs, EnergyPlus is an energy analysis and thermal load simulation program. Based on a user's description of a building from the perspective of the building's physical make-up, associated mechanical systems, etc., EnergyPlus will calculate the heating and cooling loads necessary to maintain thermal control setpoints, conditions throughout an secondary HVAC system and coil loads, and the energy consumption of primary plant equipment as well as many other simulation details that are necessary to verify that the simulation is performing as the actual building would. Many of the simulation characteristics have been inherited from the legacy programs of BLAST and DOE–2." [13]

Energy Plus is a simulation engine that is presented as a very plain program, but requires a lot of expertise in the field of energy analysis to be used. It reads the inputs and produces outputs in text files, similar to spreadsheets.

As it needs text-like inputs, to use this software we would need to create Industry Foundation Classes (IFC)[13] files within Energy Plus, with all the information about the building being modeled. Every time we changed something in the project, we would need to recalculate the necessary numbers to recreate the input needed for Energy Plus to run the simulation. Furthermore, and even though it is considered an acceptable practice, studies show that the export of a BIM model into an IFC or gbXML[14] file format and a posterior import into a BIM software may produce some information losses. [14] [15]

Even though it is considered a building certification software, it could not be directly used in the type of approach proposed by this research, since we intend to use a visualization software as an intermediate step of the process. We have the possibility of using Design Builder, which is an modeling interface, that we will refer to next, but will not use, due to its characteristics. Like Design Builder, we have other options of user-friendly tools that connect to Energy Plus to run the simulations, thus making them ideal for architects to work with. We will present some of those next.
Even having this possibility, we propose as future work a direct connection to this software, eliminating the need to export files or the knowledge necessary to work directly with his tool.

Design Builder[15]

Design Builder is an energy simulation interface software that allows a link with BIM projects, working

[13] IFC files are a file format for BIM projects, as they include all the constructive information of the building, to be used in BIM software.
[14] gbXML is an industry supported schema for sharing building information between disparate building design software tools - *in* www.gbxml.org
[15] www.designbuilder.co.uk

both in early stages of design and with detailed solutions. It connects to Energy Plus to run the energy simulation, and imports and exports gbXML files, which allows for BIM models to be used in Design Builder and then in the original software again. We can see it as the BIM-like interface with Energy Plus.

TAS[16]

TAS is a building modeling and simulation software that produces whole building energy analysis that uses American Society of Heating and Air-Conditioning Engineers (ASHRAE)[17] energy efficiency standards or building directives from the United Kingdom. It includes a three-dimensional modeler that supports three-dimensional CAD imports. It also supports importing information files such as gbXML or IFC formats, that input a range of numerical information about the building, and that are produced by BIM software.

FineGREEN[18]

FineGREEN is a software that works in a three-dimensional modeling environment with BIM principles and uses Energy Plus to perform energy analysis simulations. It imports IFC files, to receive three-dimensional information on the model to be analyzed, thus reproducing the model in its own modeling environment.

N++[19]

N++ is a software that combines a CAD-like environment with Energy Plus to model a building and simulate its energetic consumption. It does not import files from other tools, which makes it harder to work with during the design process. It is only possible to import CAD files with the plans of the different floors to serve as base to create the zoning needed for Energy Plus to run the simulation. Each CAD file has one plan, and in N++ you can define which floor it represents, the height of the floor, the height of the windows and doors, etc.

eQUEST[20]

eQUEST means Quick Energy Simulation Tool, and provides an easy to use software that makes energy simulations based on DOE-2. This software has a CAD-like environment and imports two-dimensional CAD files. It then allows the user to work in a three-dimensional view, providing all the information needed. It works with detailed models or simple masses, being able to simulate energy consumption during the different stages of design.

[16] www.edsl.net

[17] ASHRAE is an American society that works to ensure environmental sustainability, through correct energy efficiency in built spaces - *in* www.ashrae.org/about-ashrae

[18] www.4msa.com/FineGreenENG.html

[19] www.expertapp.com/npp.php

[20] www.doe2.com/equest

CYPETHERM Suite[21]

CYPETHERM is a software that uses a BIM environment to simulate energy analysis. It supports importing IFC files into it's modeling work space, and then uses ASHRAE directives to produce energy simulations and results. It proposes a consolidation approach, where the model is exported to CYPETHERM software, the several specialties are evaluated and the problems highlighted through the analysis simulation, and then the IFC file is again exported to the BIM software to be consolidated.

IDA Indoor Climate and Energy[22]

IDA Indoor Climate and Energy is a software that works in a BIM environment, allowing to import IFC files, as well as CAD files. It has a version handling system, that keeps track of the changes made on the model every time it is uploaded to the IDA software. This system allows for faster simulations, as it only reruns the needed simulations based on changes made, and it also provides comparisons between versions. It's energy analysis accuracy is validated by ASHRAE and the International Energy Agency, among others.

Seifara Architecture[23]

Seifara Architecture is a performance analysis tool that works with an existing modeling software. They have a plug-in available for *Autodesk Revit* and *SketchUp*. It produces real-time analysis, working from the beginning of the design process.

Even though this could be considered a good option to use, in what energy analysis is concerned, this is a very expensive and exclusive software. Having in mind the purpose of this research, the software was not considered due to the difficult access, even for trial.

In Table 2.4 we show a summary of the characteristics of the analysis tools referred.

Software	Import and Export	Imports 3D	Plug-in	Accessible?
Design Builder	✓	✓	-	✓(trial)
TAS	✓	✓	-	×
FineGREEN	✓	✓	-	✓(trial)
N++	✓	×	-	✓
eQUEST	✓	×	-	✓
Cypetherm Suite	✓	✓	-	×
IDA ICE	✓	✓	-	✓(educational)
Seifara	-	-	✓	×

Table 2.4: The previously referred analysis tools, and respective characteristics, regarding the workflow (import and export method or plug-in), their ability to import three-dimensional descriptions of the models, and their accessibility.

[21] www.cypetherm-suite.en.cype.com
[22] www.equa.se/en/ida-ice
[23] www.sefaira.com/sefaira-architecture

As we can see, tools like *Design Builder*, *TAS*, *FineGREEN*, *CYPETHERM Suite*, and *IDA Indoor Climate and Energy* work with importation and exportation methods, and that translates in a need to repeat the import and export process every time a change is made in the model. This makes the process very lengthly and so we chose against this methodology. Tools like *N++* and *eQUEST* do not import three-dimensional files, only two-dimensional ones, and further specifications have to be made directly in these software, which makes the previous process of import and export even slower. Finally, the case of the last tool referred, *Seifara Architecture*, that works as a plug-in, i.e., works directly in the modeling tool. Even though this is the type of approach we are aiming for, since it has no import and export process, this tool is of very difficult access, and so we could not use it.

We will now look at the options considered as viable solutions for this research, since they have a different workflow, which will be explained next.

Insight 360[24]

Autodesk Revit has a connection to an online simulation platform called *Autodesk Insight 360*. This platform uses Energy Plus (a reference tool in energy analysis) to run the simulations, and works directly from *Autodesk Revit*. When using this energy analysis tool, the user needs to generate the model in *Autodesk Revit* and then define the options for the energy simulation. The simulation is performed by *Autodesk*'s servers and the results shown in *Insight 360*'s website.

DIVA[25]

Rhinoceros 3D has a plug-in for daylighting and energy simulation called DIVA. This plug-in reads the three-dimensional model created in Rhinoceros and asks the user to apply the materials to the model. It then runs the simulation for daylighting or thermal loads, according to what the user wants. DIVA also uses Energy Plus to run the thermal loads simulations, and uses Radiance for daylighting simulations.

OpenStudio[26]

SketchUp has a plug-in called *OpenStudio*, that allows for the energy simulation of the building designed in *SketchUp*. *OpenStudio* is a 'collection of software tools to support whole building energy modeling using Energy Plus and advanced daylight analysis using Radiance.'[27]

[24] www.insight360.autodesk.com
[25] www.diva4rhino.com
[26] www.openstudio.net
[27] www.openstudio.net

2.3.2.C Process and results

For the analysis process, we used *Autodesk Revit* as a design and simulation tool. The goal is, with the same software that we will later on use, understand the changes in workflow that derives from changing from a mathematical process, described earlier, to an automatic one, using a software that allows for both design and energy analysis.

The main difference is that we no longer need to perform all the calculations, as we would need to do with the use of Energy Plus to get the right values as inputs, since the software will do them for us. We will now describe the process done to perform an energy analysis in the same case study used before (Case Study I - family house in *Azenhas do Mar*).

First of all, we need to create the three-dimensional model in *Autodesk Revit*. When the project was designed, a CAD software was used, *Autodesk AutoCAD*. When changing to a BIM tool, we were only able to import a site plan, with contour lines to model the terrain and the perimeter of the construction site. From this guidelines, we modeled the whole building in *Autodesk Revit*, from where we extracted the technical drawings that can be consulted in appendix A.

As this building has only one floor, it was easy to generate a three-dimensional model by hand, using the *Autodesk Revit* software. This part of the process took about 24 hours of working time, spread over the course of four days. In Figures 2.3 and 2.4 we can see the model generated in *Autodesk Revit*.

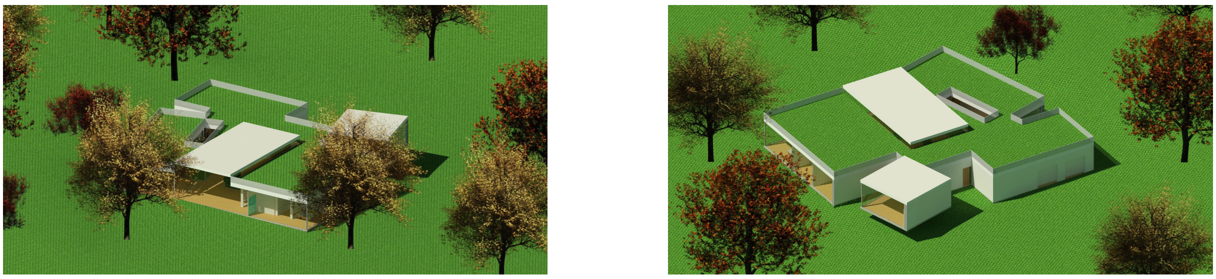


Figure 2.3: Three-dimensional views of the model in Autodesk Revit 2017.

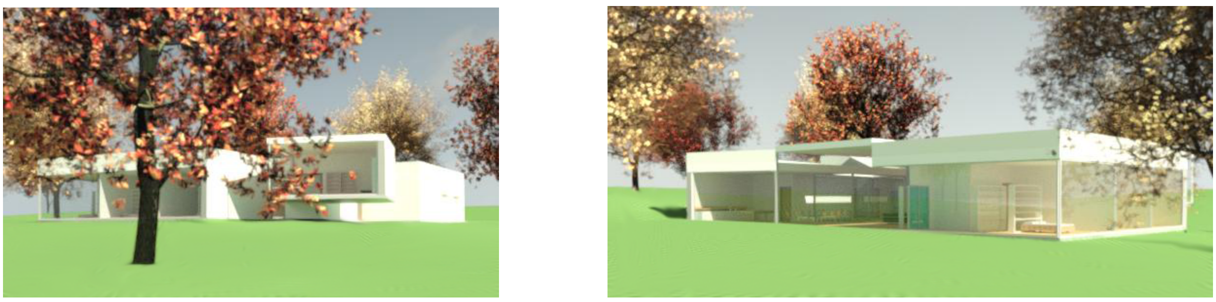


Figure 2.4: Three-dimensional views of the model rendered with Autodesk Revit 2017.

We will now look into the modeling process, namely the attribution of materials and characteristics to the objects designed. As already referred, BIM tools such as *Autodesk Revit* use families to assign different materials to the elements of the building. In figure 2.5, we can see an example of a family used in this project, with the general characteristics and the structure defined by me for this type of element.

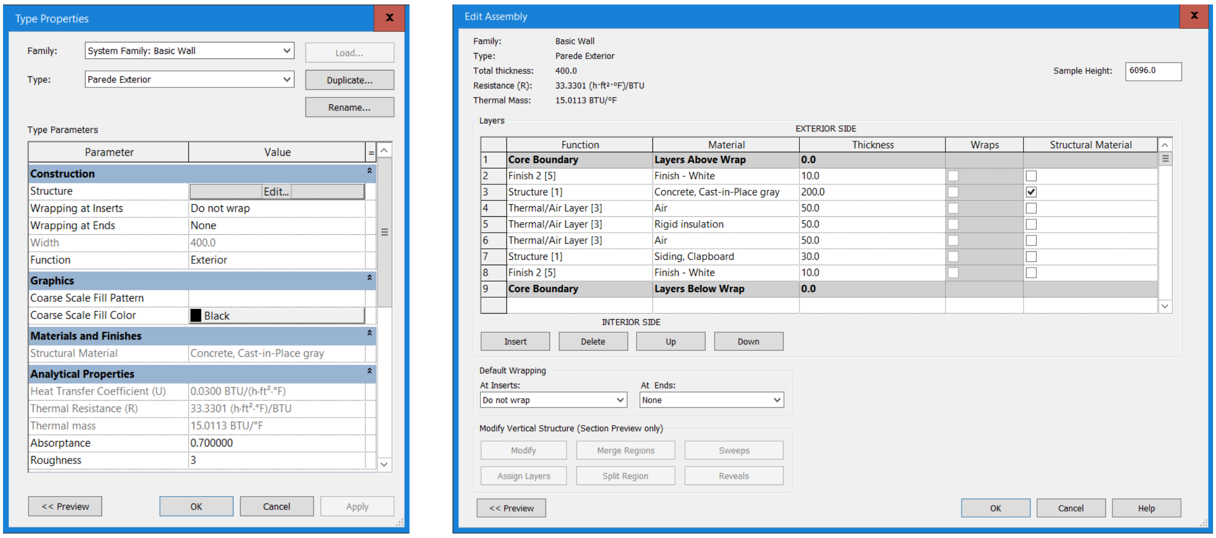


Figure 2.5: On the left, the general characteristics of the exterior walls used in Case Study I. On the right, the assembly structure of the different layers of materials of that same wall.

There are several default families to choose from already in *Autodesk Revit*, and one can easily edit those families, as well as create new ones. As seen in Figure 2.5, the structure of the wall, i.e., the several layers of materials that compose the wall, can be edited, their order changed, different materials attributed and new layers added. When choosing the material, we can also create new ones, or edit the existing materials *Autodesk Revit* provides. In Figure 2.6 we see some of the settings that a material can have.

After having the building modeled in *Autodesk Revit*, with all the different materials assigned, we go to the energy analysis phase. First, we need to define the settings intended for the energy analysis. It can be a simple one, or a more complex analysis, depending on the time that we want to spend on the analysis and the level of complexity that the project has. As we can see in Figure 2.7, we need to choose the location, type of building, if it uses conceptual masses (volumes that represent a building, but have no characteristics), building elements, or both, among others.

For this house, we select *Azenhas do Mar*, near *Sintra*, as our location. This will automatically choose the right climatic file, that has the needed information on the climate in that region. We choose a *Single Family* building type, using building elements. By choosing a *Single Family* building type, we are restricting the analysis to a building that has a typical energy consumption for a house of a single family, which would be probably a use of about 5 people, using the house mainly in the morning and at

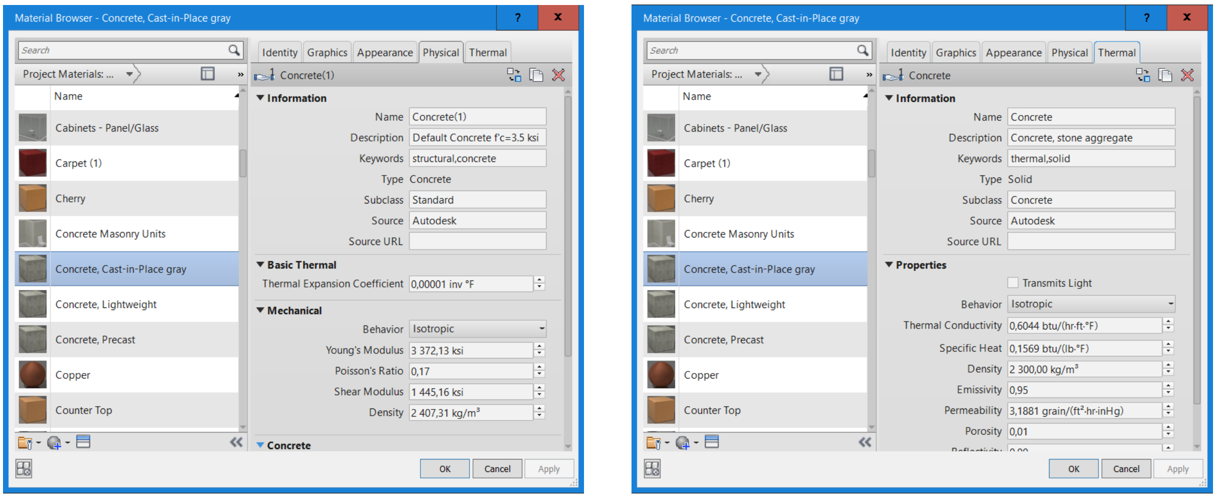


Figure 2.6: Some of the options a material can have in Revit. In this case, for material "Concrete Cast-in-Place gray".

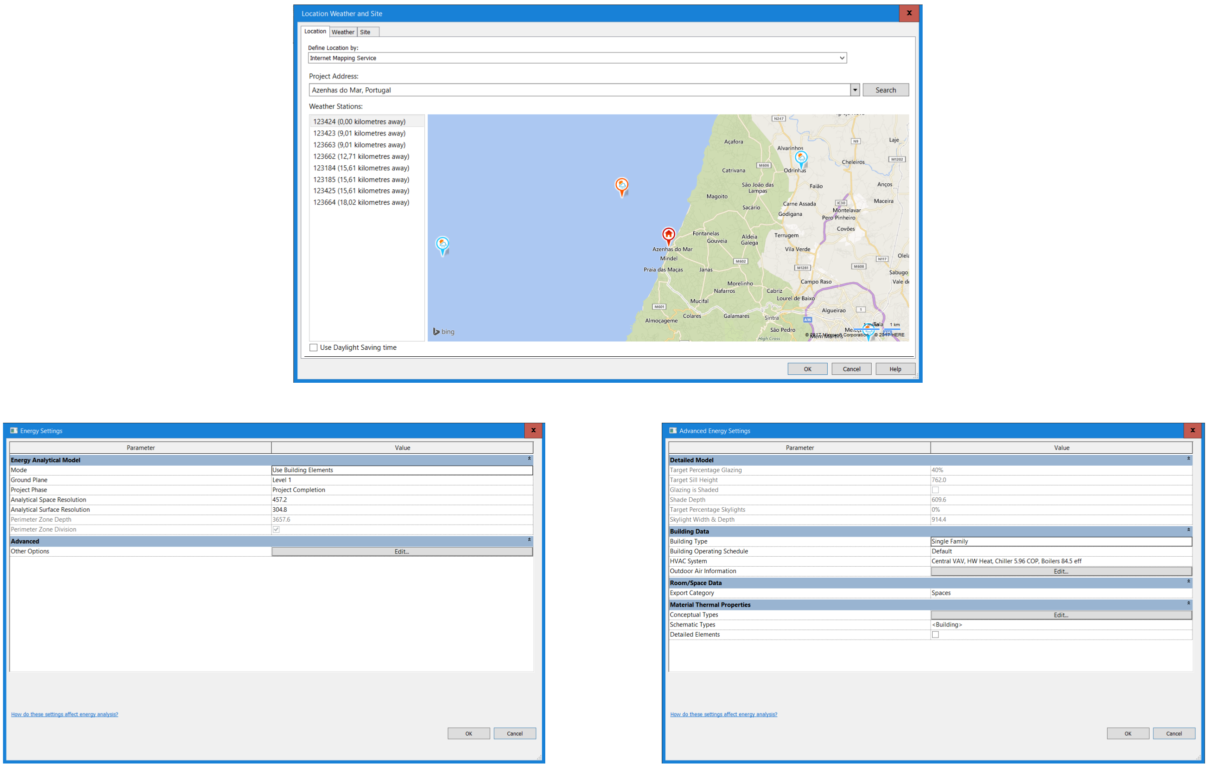


Figure 2.7: The Energy Settings windows in Autodesk Revit 2017, where we can choose location, typology of the building, among others.

the end of the day. This is a different type of schedule and a different number of users than, for example, a kindergarten, although this last one could have a spatial description very close to the one given in this case study. So, defining the type of use is of the utmost importance, to assure a correct approximation in number of users and schedule of energy use.

After changing these settings, we create an energy analytical model, and afterwards send the file to *Autodesk Insight 360*. We immediately receive an email confirmation from *Autodesk Insight 360*, announcing that the model was received and is being analyzed. Around 10 minutes later, another email was sent by them, confirming the success of the analysis and providing us with all the information produced by the analysis.

When looking at the results given by *Autodesk Insight 360*, we can see the calculations for the Energy Use Intensity (EUI) values, like mean, maximum, and minimum values, given in kWh/m^2·year. We can also ascertain annual costs, by providing electricity and gas tax rates.

As referred before, there are international directives that then result in national regulations. In Portugal, we currently use the REH for habitational buildings. *Autodesk Insight 360* uses American regulations, namely the ASHRAE directives. This means that even though both are considered valid, they may produce different results.

Before looking at the results, we need to understand the difference between the types of energy calculated, and thus the type of results that we get using different methods. When analyzing energy performance, we calculate three types of energy consumption. Using the Portuguese regulation REH and its terminologies, we can divide energy into three types: useful energy, final energy, and primary energy. The useful energy is calculated regarding the actual energy losses and gains, when calculating the values of Nic, Nvc and Nac. The final energy regards the amount of energy that is used from the energy network, and this already implies that, depending on the efficiency of the equipments, the amount of actual energy needed from the network varies. So the final energy is calculated when we divide the useful energy by the usage factor of the equipments, i.e., by their efficiency. That translates in calculating the values of Nic/η, Nvc/η and Nac/η. Finally, the primary energy is the value that considers the type of energy being used, whether it is electrical energy, solid fuels, liquid fuels, gas fuels or thermal energy produced from renewable sources. This translates in multiplying the final energy by the conversion factors F_{pu} associated with each type of energy. Being so, we use the conversion factor of the type of energy used for heating with the Nic value, the conversion factor of the type of energy used for cooling with the Nvc value, and the conversion factor of the type of energy used for heating the water in the sanitary systems with the Nac value.

It is important for us to grasp these differences before looking at the values, as the result given by using the REH (the Ntc value) corresponds to a different type of energy given by Autodesk Insight 360 (the EUI value). While the first gives the results as primary energy, the last calculates them in useful energy, as we can see in equation 2.6.[28]

[28] http://www.hpbmagazine.org/attachments/article/12033/10Su-Building-Energy-Use-Intensity.pdf

$$EUI = \frac{Annual\ Building\ Energy\ Use}{Building\ Area} \quad [kWh/m^2 \cdot year] \tag{2.6}$$

Where:

EUI is the Energy Use Intensity of the building;

$Annual\ Building\ Energy\ Use$ is the total annual amount of useful energy spent by the building;

$Building\ Area$ is the total useful area of the building.

This means that we need to compare useful energy values, instead of primary energy values. So, we will not use the final Ntc value obtained from the application of the REH, but instead calculate the sum of energy gains and losses through the Nic, Nvc and Nac values.

Having this is mind, we will now look at the results produced by both approaches, in order to compare their performance and ascertain the viability of using Autodesk Insight 360 in the next stage of our research.

Approach	Time consumed	Useful Energy result
REH	$300min$	$135,43kWh/m^2 \cdot year$
Insight 360	$10min$	$117kWh/m^2 \cdot year$

Table 2.5: The final results from the energy analysis performed in Case Study I, using two different approaches. In the middle column, the time consumed in energy analysis and calculations, and in the right column, the results produced by both approaches, in useful energy.

As we can see in Table 2.5, the value of useful energy resulting from the calculations performed based on the REH is approximately $135kWh/m^2 \cdot year$. The EUI result given by Autodesk Insight 360 is $117kWh/m^2 \cdot year$. Both these values, even though they are similiar, still have a difference of about $18kWh/m^2 \cdot year$. We can ascribe this difference to the following factors, already explained previously:

- Different regulations were used, even though they are all based in international directives and have similar goals;
- One of the approaches uses a three-dimensional model with all the characteristics of each constructive element, while the other uses mathematical and textual descriptions of those same elements, this last using approximate values;
- Both were performed by a non-expert in energy analysis. Using Revit and Insight 360, this factor has almost no impact, as there is no need for knowledge in the area to be able to perform the analysis. On the other hand, using a spreadsheet that assumes knowledge in energy analysis, the results may not be so close to reality.

One other factor we can compare is the time spent on the task. Besides not needing basic knowledge in energy analysis concepts, it also takes 30 times less time to perform the simulation using Insight 360 than to do the calculations by hand, even using the previously done spreadsheet.

We can conclude that, even though there are differences in the results, they are not substantial in an initial stage of the design, moreover seeing that the fastest approach of the two needs no knowledge on the topic, whereas the other option assumes some knowledge on the matter. Even though it is of the utmost importance that architects have basic knowledge on sustainability in architecture, and how to design efficient buildings, this knowledge will rarely be at an expert level. Tools that aid the architect achieving better energy performances, at initial stages of design, promote an increasing action towards this goal. An analysis only takes 10 minutes, instead of 5 hours, and the results are considered reliable as reference values. They do not replace official analysis made by experts, but they give the architect the guidelines to understand the energy impact of a specific design option.

After understanding the basic workflow on the energy analysis tool being used, we will introduce a concept that allows us to, not only produce more complex shapes, but also the generation of those same shapes in a matter of seconds. Moreover, it allows us to rapidly introduce changes to the design.

2.4 Generative Design

'Generative design is not about designing a building.
It's about designing the system that designs a building.'
by Lars Hesselgren, Director of KPF Research [16]

2.4.1 Introduction

Generative Design (GD) is a process that produces various design solutions from a set of rules and constraints defined by algorithms. In Figure 2.8, Figure 2.9, and Figure 2.10 we can see several buildings designed with a GD approach.

An algorithm is defined as a 'procedure for computing a function', [17] i.e., a mathematical description of an action that we want to perform. We can consider them as a set of steps to follow in order to achieve a specific goal.

Algorithmic systems are the basis of GD, expressing, by a set of rules written in a specific language, the goals of the design, and the steps to achieve them.

> 'Thinking in terms of algorithms is a mapping process of design objectives onto step-by-step descriptions. Such a process helps designers decompose context, understand relationships and devise methods to judge the utility of the outcome'. [18]

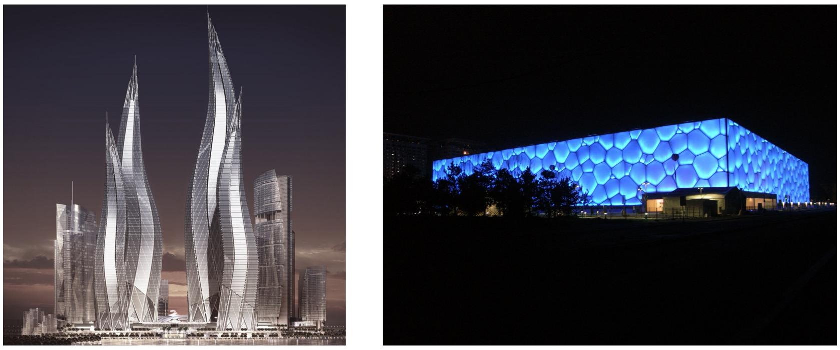


Figure 2.8: On the left, the Dubai Towers, in Dubai, by tvsdesign architects (image available in: *www.tvsdesign.com/projects/the-dubai-towers/*). On the right, the Water Cube, in Beijing, by PTW architects (image available in: *www.ptw.com.au/wp-content/uploads/2012/11/OC˙Night195.jpg*).

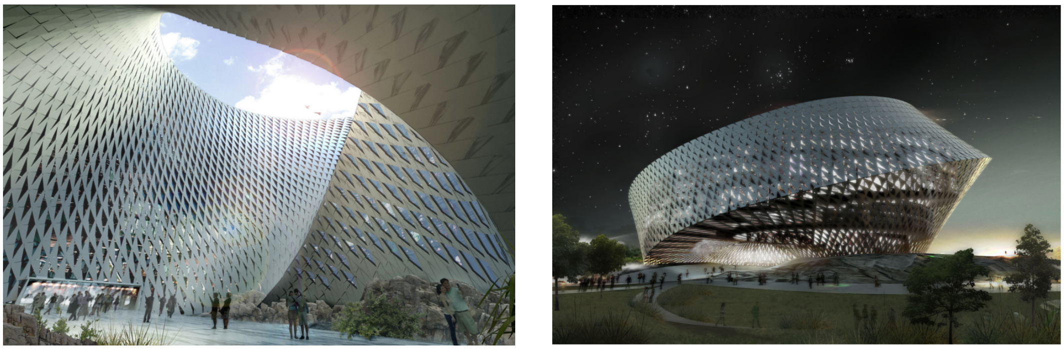


Figure 2.9: The Astana National Library, in Kazakhstan, by BIG architects (images available in: *www.big.dk/#projects-anl*).

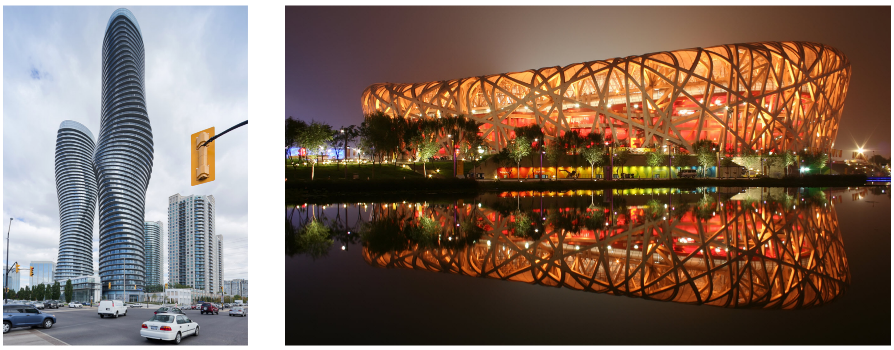


Figure 2.10: On the left, the Absolute Towers, in Canada, by MAD architects (image available in: *www.i-mad.com/work/absolute-towers/?cid=5*). On the right, the Bird's Nest Stadium, in Beijing, by Herzog & de Meuron architects (image available in: *https://beijingbirdsnest.wordpress.com/birds-nest-facts/*).

In this case, we are talking about computers understanding and performing actions defined by humans, and so the language used to communicate is necessarily a programming language, i.e., a methodical communication system to transmit a thought process to a machine. [19]

Applying algorithmic systems to design requires 'clear design intents expressed on discrete steps and units'. [20] This defined components can then communicate through associations, constraints, and rules. Associations are links that allow for changes in the results by altering the values of the variables. Constraints are the terms that must be fulfilled. A rule, when satisfied, produces an action. [20]

In GD, a computational approach is used, creating functions that describe the design intent. The different values applied to the parameters of these functions are what allow them to create several design options, all with the same principles. [1] GD is nowadays seen as a way to combine the creative mind of the architect with the effectiveness of modern technologies, thus leading to a more efficient work and to greater opportunities and different options of design. [20] It is to note that the introduction of these approaches are not a substitute to creativity in architecture, instead, they support it. [21]

This algorithmic approach dates from an era that came many centuries before the invention of digital computers. If an algorithm can be seen as a set of rules, we can simply look at the work of Alberti, so well known among architects, *De re aedificatoria* to find several algorithms. His work provided a set of rules to achieve a classical design process, and it is still seen as a fundamental historical reference. We can see some examples of his algorithms in Figure 2.11.

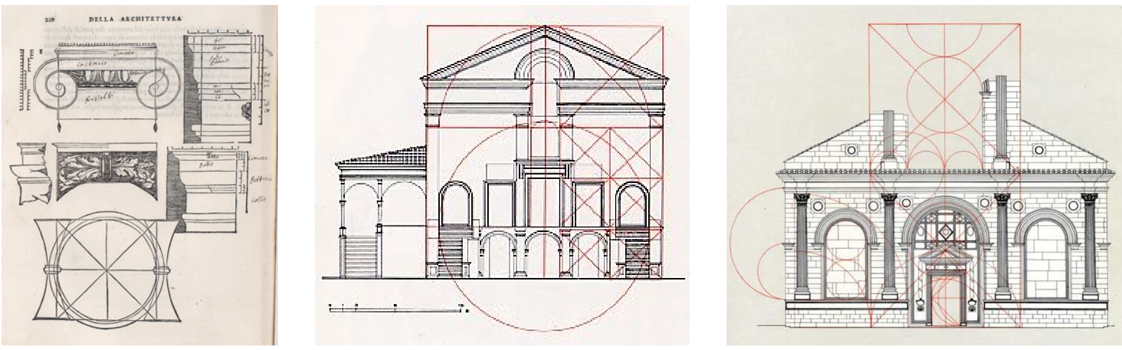


Figure 2.11: Images from Alberti's book *De re aedificatoria* (images available in: *https://s-media-cache-ak0.pinimg.com*).

Nowadays, with the existence of computers, this notion of implementing algorithms in architecture was simplified, [19] and we use computers to input the rules and let the machine run them and produce the intended results. Computers also allow us to achieve more complex shapes and geometries, through mathematical abstraction.

GD started more connected to CAD tools. These tools are mainly used for representation purposes, as they only work with geometry, such as lines and solids, without giving them attributes, like the simple constructive difference between a wall and a slab. In terms of geometry, these two elements are just two parallelepipeds that happen to have different orientations in their spatial arrangement, i.e., walls are mainly vertical and slabs are mainly horizontal. GD allows architects to have a phase of design exploration in early stages of design, but sometimes the designs they obtain are very hard, if not impossible to build. When reaching this stage, it is important to have a tool that understands the constructive characteristics of each type of element, and so BIM tools emerge.

With BIM tools, the paradigm shifts considerably. These tools consider the attributes of each constructive element, as well as their natural interdependence between each other. For example, a simple box in CAD tools can be either a slab or a wall when changing to BIM tools, depending on their orientation. Also, in BIM, one cannot create a window unless it is hosted in a wall, or design a staircase unless there are two levels to host each end of it, which changes the design methodology. [14]

The GD application to BIM tools is what interests us in this research, for it provides both the benefits of an algorithmic approach as well as the advantages of using a tool that works with all the constructive attributes of each building element, including the thermal behavior of the materials each element uses.

2.4.2 GD as a tool for energy analysis

By associating a GD approach to an efficient evaluation system, architects can explore more design options which take into account the building's energy behavior, even in early stages of design. 'The early stages of building design include a number of decisions which have a strong influence on the performance of the building throughout the rest of the process. It is therefore important that designers are aware of the consequences of these design decisions.' [22]

Using GD to improve the energy consumption of a building can be a great way to reduce the time consumed doing this task. As seen before, GD consists in writing a set of rules designed to constrain the general concept of a building, allowing the user to then make some changes in the values of specific dimensions or relations, producing countless different design options.

When we want to run an energy analysis simulation, we provide the analysis tool with a three-dimensional model, or a set of parameters that describe that same model, depending on the tool being used. Once we have the model described, the tool performs the energy analysis.

When using a traditional modeling approach, if we wanted to change some aspects of our building to try to improve energy consumption, we would have to manually introduce those changes, either on the three-dimensional model or on the parameters that describe it. Either way, we would do this probably just a few times, comparing the results and choosing one option among two or three. Considering that each analysis simulation takes time to run and produce results, and that changing the model to test new options also takes time, we would spend several hours of work on each building variation, and this demotivates architects from exploring their designs into obtaining better performing ones.

Assuming a good knowledge of the modeling tools and good programming skills, the time spent on manually producing a building model can be close to the time spent on creating a set of algorithms to generate it.

The big difference only comes afterwards, when it becomes necessary to change the model to generate variations of it. After having the model in the analysis software, the time that the simulation itself takes depends on the complexity of the model and on the settings defined in the software for the detailing of the analysis. When we run the first simulation, we evaluate the results and change the model according to the total energy consumption value given by the simulation. These changes can be easily made manually if the model is not too complex but, usually, architects need to perform this kind of analysis in big and complex models. Changing a large and complex three-dimensional model can take a lot of time, and this time increases considerably according to the number of changes that the architect wants to make.

On the other hand, GD can make the process of changing the model much faster. The architect just needs to adjust parameter values according to the changes needed, and run the program again to rapidly generate a new model.

2.4.3 Programming languages in GD

With the growing importance of technologies in our society, it is already starting to become essential to have programming courses in architecture degrees, providing future architects with the necessary knowledge to make the most of the tools which are becoming available.

In generative design there is a very popular approach to programming based on the use of Visual Programming Languages (VPLs). These programming languages are considered to be more accessible to users with little or no knowledge in programming. The main example, when it comes to the applications to nowadays architecture practice, is *Grasshopper*, a 'graphical algorithm editor tightly integrated with Rhino's 3-D modeling tools. (...) Grasshopper requires no knowledge of programming or scripting, but still allows designers to build form generators from the simple to the awe-inspiring'.[29] Two other similar tools are *Dynamo* for *Autodesk Revit* and *Generative Components*, for *Bentley Systems MicroStation*. *Dynamo* works in a similar way as *Grasshopper*, being a plug-in for *Autodesk Revit*, while *Generative Components* is a software that combines the BIM environment with VPLs, allowing for parametric design and direct manipulation of the model simultaneously.

There are several studies that show how a VPL is much more accessible to students that have never programmed before, such as the typical architecture student. VPLs work in what is called a *flow-based language*, where 'multiple components are linked together to form a transformation chain, from input data (e.g. sliders) to generated geometry'. [23]

On the other hand, Textual Programming Languages (TPLs) work in a *block-based* approach. Whereas in *flow-based languages* data travels along a network of functions and transformations, in *block-based languages* data exists as variables that are defined in blocks of written code. [23] We can see the positive

[29] www.grasshopper3d.com

effects that programming skills have in architecture students in studies such as [24], [25], [26], [27] and [28]. As written by Gallas et al., [25] 'Architecture students are not intended to be experts on programming'.

According to Giordano et al., [29] 'visual programming environments have an advantage in avoiding any syntactic problems and by allowing the students to concentrate on problem solving activities and to experiment with their creativity by looking at different ways of solving the problem at hand.' In [26], a VPL is used, [27] uses a TPL, and [28] compares both approaches, arguing that, when it comes to scaling the project and the time consumed in changing the code, VPLs do not go as far a TPLs.

Even though they may seem easier at the beginning, VPLs can turn into very complex systems, almost impossible to understand, as we can see in the examples shown in Figure 2.12. A comparative study made by Leitão et al. [19] shows how TPLs can be more beneficial to generative design. In this study, the same design problems were solved using both approaches, comparing the time spent in each task and the complexity of the algorithm created. One of the tasks was to perform changes in the initial design, since, 'in GD, programs must be easily adaptable to changing requirements.' [19]

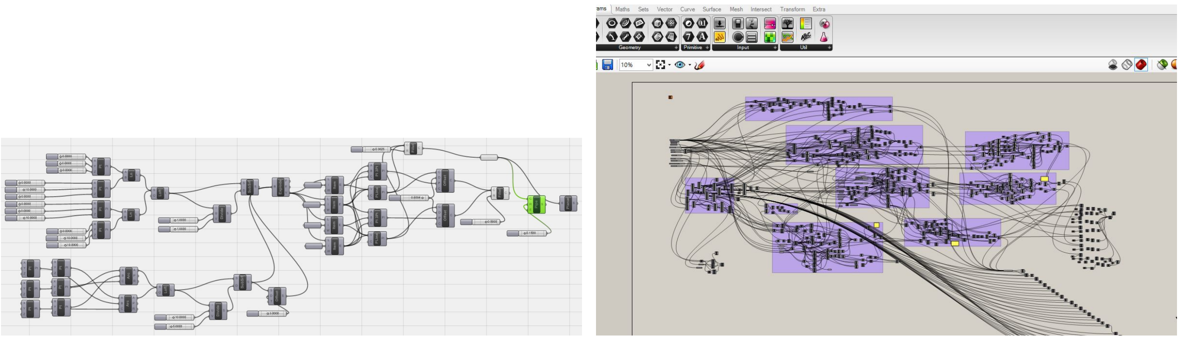


Figure 2.12: Examples of Grasshopper code: a simple example on the left, available in *wiki.theprovingground.org/richard-armitstead*, and a complex one on the right, available in *fab.cba.mit.edu/classes/863.13/people/wildebeest/projects/week1/index.html*.

After reading the studies referred before, and also after experimenting both approaches, we conclude that TPLs have a better workflow than VPLs, once we seldom work with simple projects and want to exponentiate the advantages of a GD method, i.e., of parametrically defining a building and being able to rapidly change and increase its complexity. This task is proven to be easier to perform when working with a TPL, and that is the reason why we use one in this research, called *Racket*, which is a 'modern member of the LISP family of languages, designed for pedagogical and practical purposes.' [30]

3

Algorithmic Approach to Energy Efficiency and Optimization

Contents

As seen so far, Generative Design (GD) can help improve efficiency, reducing the time consumed to create new models by being able to test various design options in a matter of minutes.

We will now understand how to implement this kind of approach to a more complex case study, using the model created to perform energy analysis and consequently improve the model generated previously.

3.1 Case Study II

3.1.1 The building

The building chosen as a case study for this part of the research was *Beirut Terraces*, an habitational building in Lebanon, designed by Herzog & de Meuron. This architecture studio located in Basel, Switzerland, is a result of a partnership between two architects, Jacques Herzog and Pierre de Meuron. Their office was established in 1978, and since then they have a growing team of international members, now counting with 40 associates and 380 collaborators, working across Europe, Asia, and the Americas.[1]

The concept designed by Herzog & de Meuron starts as a square-based volume in the marina area of Beirut. The idea is to create a sense of randomness, with varying combinations and ambiances, to create different spaces, both interior and exterior. To do so, the architects opted to design five different modules for the floor slabs, and by arranging them throughout the 25 residential floors, create the different floors.

This modular concept creates a sense of randomness in the 25 residential floors of the building. As we go through the floors, their interior display also changes, consequence of the shapes of the bottom and top slab of each floor. The lower floors have more and smaller apartments, while the top floors have fewer and bigger ones. Floors can have about 6 apartments maximum, having some duplexes and two penthouse apartments on the top floor.

According to the architects, the design was based on five principles: 'layers and terraces, inside and outside, vegetation, views and privacy, light and identity.'[2] By creating this random volume, the architects produce several different spaces and connections between them, with a building opened to the outside, and simultaneously creating privacy.

[1] www.herzogdemeuron.com
[2] www.dezeen.com/2010/07/27/beirut-terraces-by-herzog-de-meuron

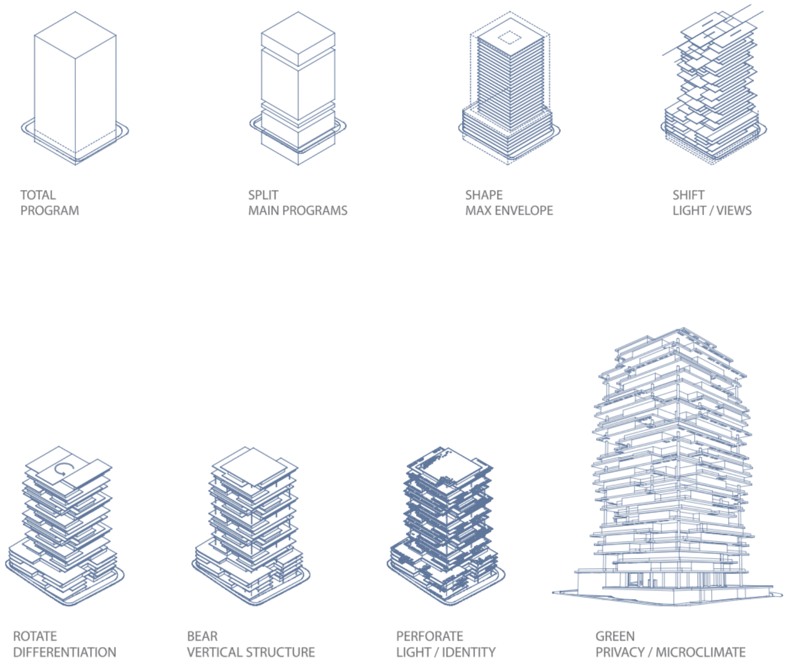


Figure 3.1: Images explaining the conceptual evolution of the buildings design. [31]

In this case, we can see that the architects probably designed this building from the outside inward, creating the volume through several conceptual steps that lead to the general form the building has now. This evolution of the application of different conceptual rules is explained in Figure 3.1. The technical drawings of some of the floors of the building can also be found on Appendix A.

The main goal of this concept is to differentiate all the apartments available, creating 130 'living experiences', as described in the project's website.[3] In this platform, they allow clients to choose from the different available apartments, instead of having a model-apartment, once they are all different from each other in terms of the arrangement of spaces, living areas, terrace areas and views of the city. In Figure 3.2 we can see the building, with its different floors and directions. In Figure 3.3, the detailing of the differentiation between slabs is evidenced.

We chose this case study as it presents a major challenge in what energy consumption is regarded. We can rapidly and instinctively identify that the massive use of glass will make this building an energy consumption problem, making it almost impossible to obtain a good energy performance. Being so, we want to use the workflow here proposed to show how, even with a building with this characteristics, we can still improve energy consumption at different stages of the design. In this work, we will only be evaluating one of those stages, as we only work with the shape of the building. Further tests should be made, changing different aspects of the building, like materials, display of apartments, and so on, in

[3]www.beirutterraces.com

Figure 3.2: Images of the building, available in *www.beirutterraces.com/luxury-apartments-pictures-beirut*

Figure 3.3: Images showing details of the building, available in *www.beirutterraces.com/luxury-apartments-pictures-beirut*

order to improve its performance even more.

3.1.2 Modeling strategy

The generation of the model in *Autodesk Revit* was made using *Rosetta*. *Rosetta* is a programming environment that connects front-end programming languages to backend Computer-Aided Design (CAD)/Building

Information Model(ing) (BIM) applications. [32] Through *Rosetta*, we were able to parametrize the model used as a case study and then perform the energy analysis using the same software as a backend for both stages of the process. This was possible since the chosen software itself provides the tools needed to perform the simulation. *Autodesk Revit* and *Autodesk Insight 360* were used in this research as model generation and energy consumption simulation tools. The energy consumption simulation program used by *Autodesk Insight 360* is Energy Plus, which in turn is based on the DOE-2 program. Both are considered to be reference programs when working on buildings energy analysis.

The building chosen as a case study for this approach was decomposed in its essential constructive elements to generate a simple model representative of its concept.

The set of algorithms created has in mind the natural evolution of the architectural design, starting by evaluating the expected energy consumption of a building in its initial stage of design, and then performing more analysis along the way, promoting a sustainable design process. This way, architects can define their general concept and evaluate its different variations, choosing the one that achieves a better energy consumption, and then maintain this method throughout the whole design process.

We started by defining the basic elements that compose the essential form of the building. In this case, the building has 25 residential floors, and is divided into the following constructive elements: slabs, columns, core area, and walls.

Slabs:

Each slab has a shape that follows one of five different types. All of them are based on a square, but the edges have some breaks, forming polygons with 8, 10 or 12 edges, depending on the type of the slab. The dimensional relations were ascertained, and we can see the plans and the three-dimensional models of each type in Figure 3.4.

One other aspect of the slabs is the relations between them. According to the existing building, we defined some rules about the sequence that the slabs can have. By doing this, we create three possible options of ordering the slabs: (1) we can arrange them randomly, where the type of slab of a given floor does not depend on the types of slabs in the floors immediately above or below; (2) we can manually define the order of the slabs, forcing the type of slab of each floor, either by defining a specific order or by defining the type of slab for each floor manually; (3) we can define a rule that chooses from a set of possible types of slabs according to the floor immediately above or below, allowing the program to generate different options, always complying with the same rule.

In this case, we first considered the third option, and defined the rule based on what was observed in the actual building. This produced the following relations:

- Type 1: can be built immediately above or below types 2, 4 or 5;

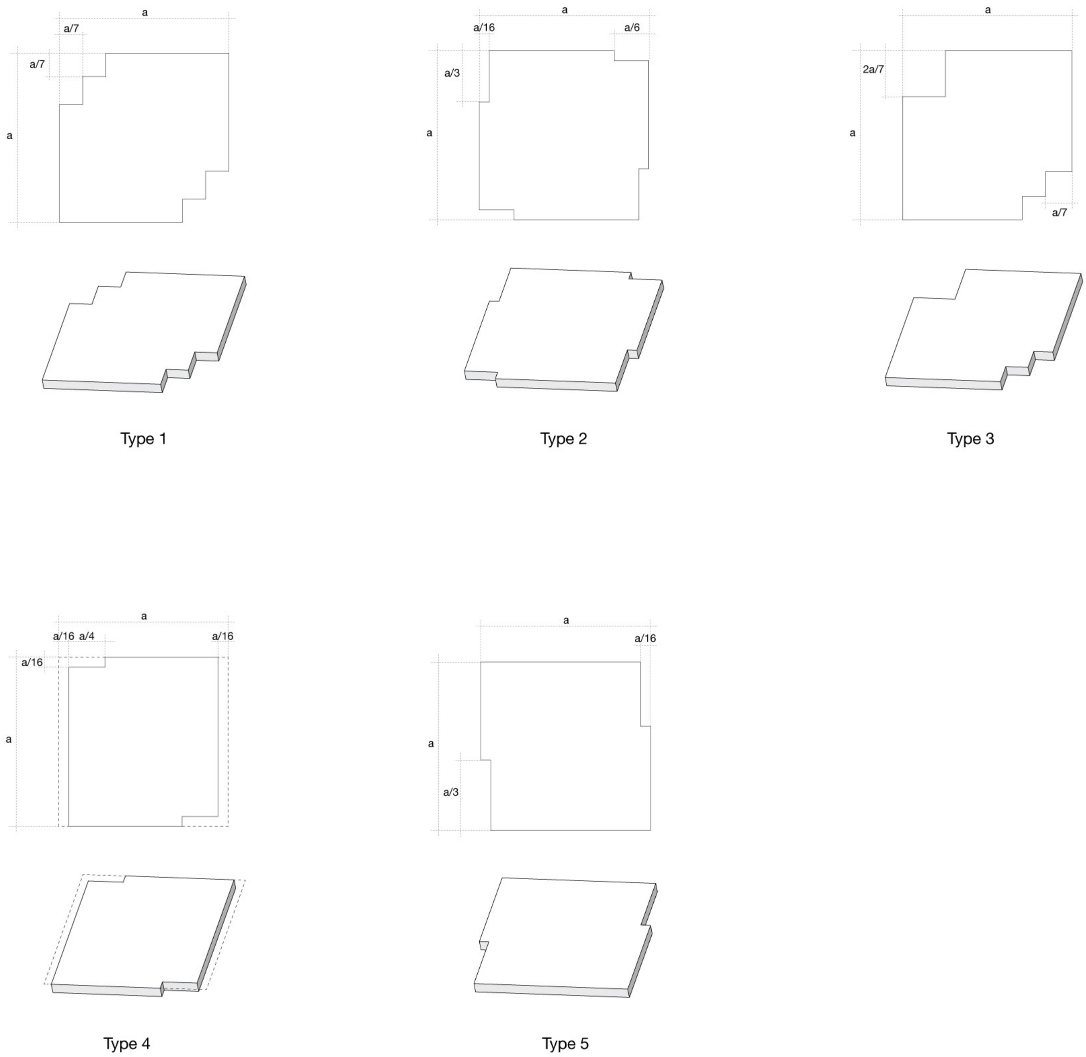


Figure 3.4: The five different types of slabs existing in the building.

- Type 2: can be built immediately above or below types 1, 2, 3 or 4;
- Type 3: can be built immediately above or below types 2, 4 or 5;
- Type 4: can be built immediately above or below types 1, 2, 3 or 5;
- Type 5: can be built immediately above or below types 1, 3 or 4.

Afterwards, we decided to give the program more room for changes in the model, allowing for a wider variety of options to choose from. Thus, we went with the first option, where the program randomly chooses slabs for each floor.

Finally, we wanted to take advantage of GD to ensure that the randomness factor would be present, just like it was intended in the original design by Herzog & de Meuron. As we only had to write the script, instead of drawing each slab by hand or relying on a copy-paste process, we could easily make all the slabs different from each other. For the building to resemble its original form, we analyzed the five types of slabs in the plans provided by the architects,[4] and created a set of dimensional limits and constraints for each corner of the slab, thus producing numerous hypothesis from a single script.

This would be really hard to achieve if we were not using GD to generate the model. Each change in dimensions, limits and constraints would lead to hours of manual work in the model, always having room for a few mistakes. Using GD, we only need to update a few values for the building to change completely, taking less than 20 seconds to generate the model all over again, with no mistakes. Generating the slabs with this final option, giving more dimensional freedom, we only need to set the limits within which the slabs can variate, give the number of levels required and set the maximum side of the square in which the building can be built. This way, we can increase the randomness of the model without the extra working hours needed if using the traditional approach.

We will now look at the mathematical descriptions of some of the variations introduced, to better understand the method used. For the functions that create the slabs, we only need a total of four parameters to generate the whole set of slabs:

- p - the center point of the building's implantation area;
- a - the side of the implantation area square;
- n - the number of floors that we want in the building;
- h - the height of each floor.

Starting with the center point $p = (x, y, z)$, we created the four corners of the square where the slab is inserted, with the following points:

- $p1 = (x - (a/2), y + (a/2), z)$
- $p2 = (x + (a/2), y + (a/2), z)$
- $p3 = (x + (a/2), y - (a/2), z)$
- $p4 = (x - (a/2), y - (a/2), z)$

When coding, we work in a clockwise direction, so the first point, $p1$, is the top left corner of the slab. This point will be at the exact corner of the slab, if it were a perfect square with its center on point p, as we can see by the descriptions above and in Figure 3.5.

[4] www.beirutterraces.com

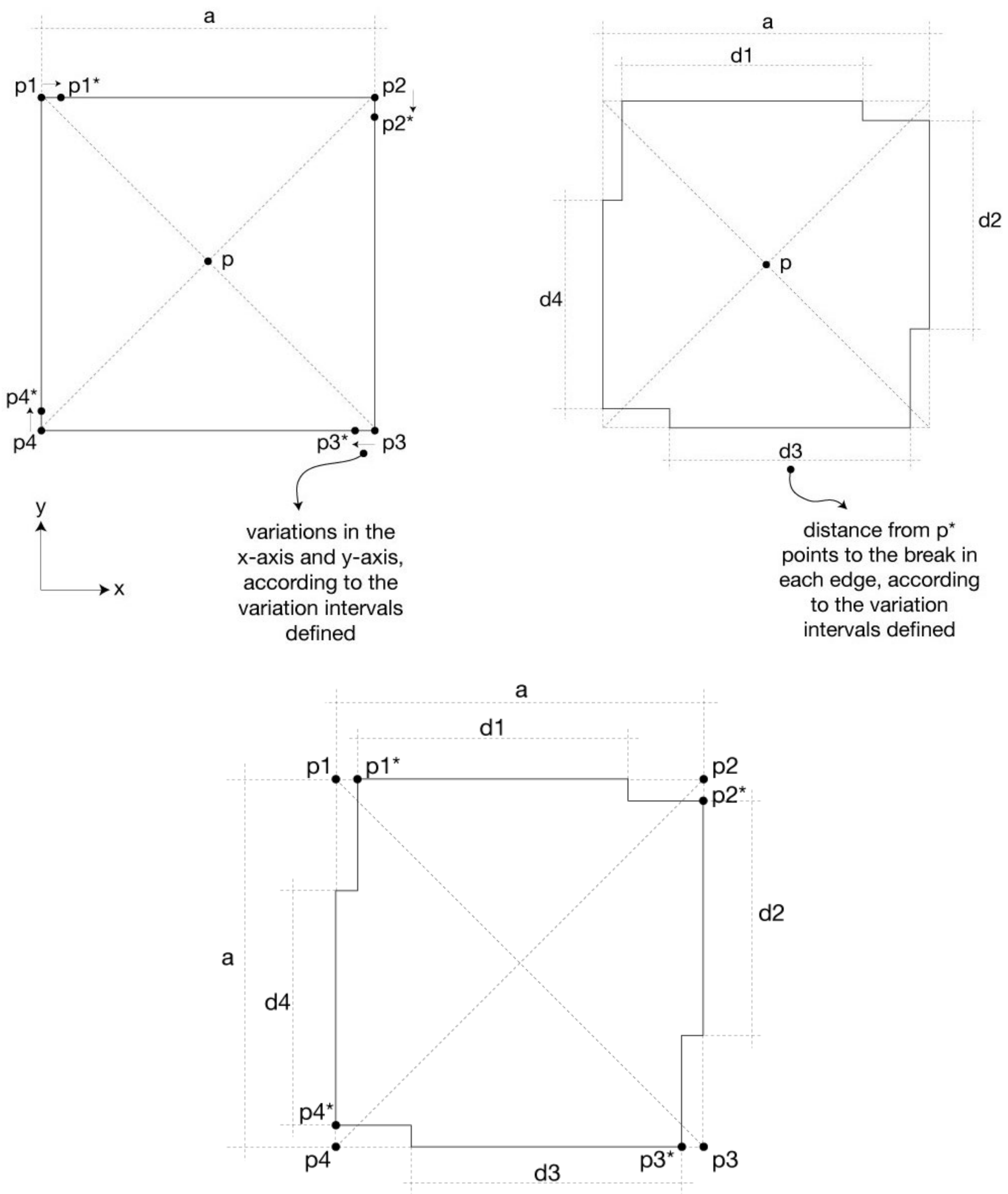


Figure 3.5: In this image we can see the initial four corners ($p1$, $p2$, $p3$, $p4$), the corners resulting from the variations ($p1^*$, $p2^*$, $p3^*$, $p4^*$) and the four distances created in each edge ($d1$, $d2$, $d3$, $d4$), thus generating the default slab, seen in the third image.

Then we randomly offset these points using axonometric coordinates, i.e., using the **x-axis** and the **y-axis**. This will prevent the slab from being obliged to reach the maximum side of the square, just like we can see in the original building. This creates four new points:

- $p1^* = (x - (a/2) + \Delta, y + (a/2) + \Delta, z)$
- $p2^* = (x + (a/2) + \Delta, y + (a/2) + \Delta, z)$
- $p3^* = (x + (a/2) + \Delta, y - (a/2) + \Delta, z)$
- $p4^* = (x - (a/2) + \Delta, y - (a/2) + \Delta, z)$

Where Δ is a variation interval defined by two limit numbers, where all the possible numbers within that range can be used to add to the **x** and **y** coordinates of each point. The architect provides the limits of the variation interval, and the computer randomly chooses a number within that interval.

After setting the new four corners, we define the distances between the corners and the breaks in the edges that follow each corner.

- $d1 = \Delta 1$
- $d2 = \Delta 2$
- $d3 = \Delta 3$
- $d4 = \Delta 4$

We can see this illustrated in Figure 3.5. In this case, we gave the same minimum and maximum variation values, but we can change this anytime, for any other values.

Now to produce all the slabs in the building, we take two steps. We first create a list of all center points of all the slabs, then just apply slabs to all the center points created. Creating a list of points relies on simply adding the height of each floor (h) to the center point p, in a recursive manner, i.e., adding h to the **z** coordinate of the last p point produced. This means that the first addition will be made to the original p point, the second will be made to the first point generated in this step, the third will be made to the second point generated in this step, and so on, doing this the number of times necessary to reach the number of floors (n) given.

Afterwards, we place a slab in each point previously produced. We can see this illustrated in Figure 3.6. Note that the first and last slabs are squares, since they are the first floor and the roof, and so have different shapes from the rest.

Columns:

Another defining element of the building is the column arrangement, that has a specific pattern,

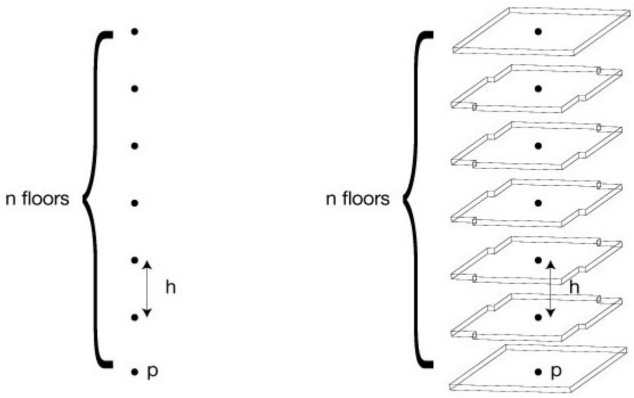


Figure 3.6: In this image we can see the several points created for the different floors, and the slabs applied to those same points.

continuous from the first to the top floor.

For the columns, they were distributed as a pattern instead of giving a specific number of columns. As we can see in Figure 3.7, if we change the total number of columns in the exterior rows, it will generate the rest of the columns, always complying with the given pattern.

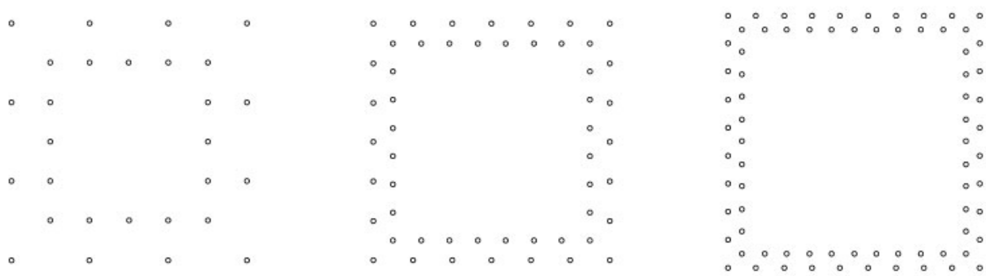


Figure 3.7: From left to right, columns simulation producing 12, 24 and 36 exterior columns.

Note that only even numbers were used. As it is the total number of columns in the exterior rows, and as it is a square, there is always an even number of rows, and therefore an odd number of columns would not work. Also note that the side of the square did not change its dimension, meaning that the spaces between columns will be necessarily smaller, as it is calculated from the relation between the edge of the square and the total number of columns in the exterior rows. The ideal relation for the building proposed is the first one, although we can clearly produce several different options changing the number of columns and/or the dimensions of the square.

The script used to produce this result was based on the following calculations:

- $nc1 = \lfloor (nc/4) + 1 \rfloor$

- $nc2 = nc1 + 1$
- $dc = a/(nc1 - 1)$
- $p0 = (x - (a/2), y + (a/2), z)$
- $p0^* = (x - ((a/2) - (dc/2)), y + ((a/2) - (dc/2)), z)$

Where:

nc is the total number of columns in the exterior row;

$nc1$ is the number of columns in each edge of the exterior row;[5]

$nc2$ is the number of columns in each edge of the interior row;

dc is the distance between columns;

a is the side of the implantation area square;

$p0$ is the insertion point of the first column in the exterior row;

$p0^*$ is the insertion point of the first column in the interior row.

To generate the columns, we used the side of the square (a) and the number of columns in each edge of the exterior row ($nc1$) to calculate the distance between columns (dc).

As there are two rows of columns, the exterior row and the interior row, we defined two starting points, point $p0$ for the exterior row, and point $p0^*$ for the interior row. Then we applied a list of points that creates a square of positions where to place the columns, based on the initial point used and the number of columns needed. To adapt the interior row to the size available, we changed the a value, reducing it (a^*). The reduction value was based on the number of columns in the exterior row, and was calculated by subtracting the distance between columns: $a^* = a - dc$. By generating a new side of the square (a^*), a new distance between columns is created (dc^*), using the number of columns for the interior row, and thus resulting of two rows of columns proportional to each other and complying with the same rules. The result produced is as shown in Figure 3.7.

We defined the height of the columns as the total height of the building, so that they are continuous. We also specified a given column-family to be used in the project, once *Revit* allows for families of objects to be imported and placed in different projects. For this building, we chose one of the families that *Revit* provides by default in its libraries. This function is directly defined in the script, and we choose the file where the column-family is designed and set the parameters available in the family, which in this case are the diameter of the column and its height. This was only possible for the columns due to family characteristics in *Revit*. Slabs and walls are not considered families that can be saved in separate files. The architect can define the layers intended, but cannot design a sample wall. For elements like columns, we can import a family file, that allows us to choose from round columns, square columns,

[5] Note that this needs to be a natural number, and so the result of the fraction is approximated to the lower closest natural number.

H-shaped columns, etc., or even create our own column and use it in the project, defining the section of the column and the dimensional parameters that can be changed, like height and diameter, for example. In this case, we used a circular column, the diameter was set to 0.2 meters and the height was set by the total height of the building.

Core Area:

Another important element is the core of the building, where we can find the staircases, the elevators, and the entrance foyers of each apartment. This area was considered as a unique area during an initial stage of the design, since it will have a very similar behavior in terms of energy consumption. This happens because these type of spaces are considered as non-heated or unconditioned spaces, due to being common areas outside of the apartments. Non-heated spaces are the ones that are not considered a concern in terms of temperature control, like garages, or common passage areas, as is the case of the elevators and staircases area in our case study. These are not areas in which we live in, but areas of secondary use. Being so, they are not climatized, and so are considered as non-heated spaces.

The core area was simply defined as a square made of four walls, repeated in all the levels of the building.

Walls:

Finally, we defined the glazed walls that separate the interior areas from the exterior. All the areas in contact with the exterior are glazed areas, and this allows us not to have to detail the interior areas of the apartments, because the whole building performance will not be affected by this.

Regarding energy analysis, when inside an apartment we need to look at its limits. The interior walls that divide the various rooms are not designed to be isolators, only space separators. Thus, in an initial stage, and having in mind that we want glazed areas surrounding all the floors of the building, the impact on global energy analysis of the interior walls will be close to none. In a more advanced stage of the design, we should detail them, making it possible to compute the energy consumption of each apartment. As we are now focused on whole building energy consumption, this is not the main concern, and therefore the interior walls need not be considered and we only defined the glazed walls that separate the interior areas from the exterior.

The main difficulty when designing the exterior walls was their relation with the slabs. The walls have to be inside of the slab of the corresponding floor, but they also have to regard this same rule for the slab immediately above. To solve this problem without compromising the random characteristics of the slabs and walls, we added a reduction factor to the script, so that the walls would not have the opportunity to reach the limits of the slabs. We will now show the steps taken to design the walls.

To design the exterior walls, we used the following method: (1) we defined the four corners of the polygon, as well as the four distances to the break in each edge, exactly in the same way as we did for the slabs; (2) the walls of a single floor are generated; (3) all the walls of the building are generated; (4) a reduction factor was added, to solve the problem referred before, making sure that the walls do not surpass the area of the slabs. We can see the result of this process in Figure 3.8.

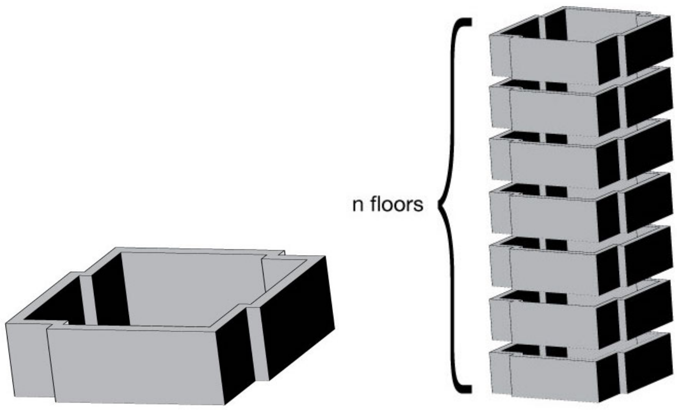


Figure 3.8: On the left, all the walls of a single floor; on the right, the walls of all floors.

In this case, we also had to guarantee the materials characteristics of these walls. The different layers composing the elements were not specified, only generic attributions were made. For the roof, slabs and the interior walls (core area), we changed the default generic *Revit* family to be composed of a single layer of concrete. We will now briefly explain the process needed to change the material of the elements.

When generating an element through the script, the element will be created with the default generic family for that element. For example, for the slabs, they are generated as a *floor* family and a *generic* type, as we can see in Figure 3.9.

When adding to the script the material to be applied to an element, its structure will change. In Figure 3.10 we see the floor family already being composed by concrete, and the thermal characteristics that are intrinsic to each of *Revit*'s default materials.

So we defined, directly in *Revit*, that the generic families that were going to be used by the script (walls, floors, and roof), would be using *Concrete, Cast In Situ* in one single layer, since we are still in an initial stage and just want the basics of the building. Further on, we should change this default family to have the different layers of materials intended for each constructive element.

After setting all the elements to use concrete, we now need to differentiate the exterior and interior walls, once the first ones are made of glass and the second ones made of concrete. Even though there is a command in Revit that quickly allows us to select all walls in the project with the same wall-family, the interior and exterior walls are created with the same family, and so this command is of no use in this

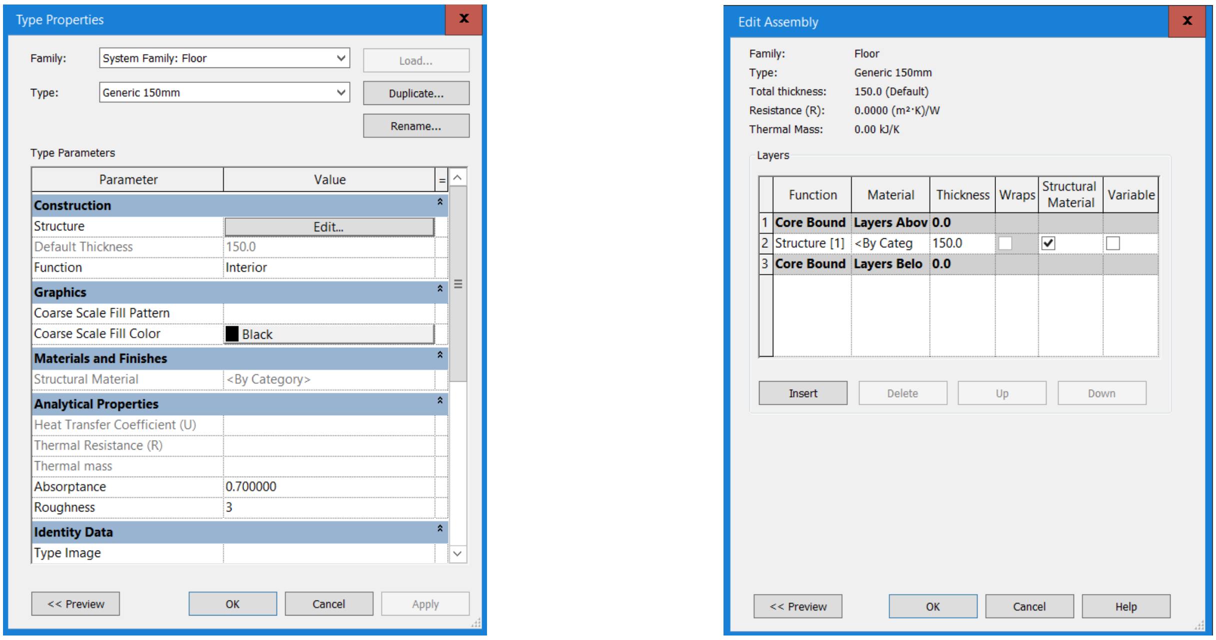


Figure 3.9: Choosing the *Edit type* option, we can view the characteristics of the family selected (window on the left). In this case, we have a generic floor. By then choosing the *Edit* option in the structure menu, we can see and edit the assembly of the different layers of each element (window on the right). In this case, the default families existent in *Revit* do not have any specific material.

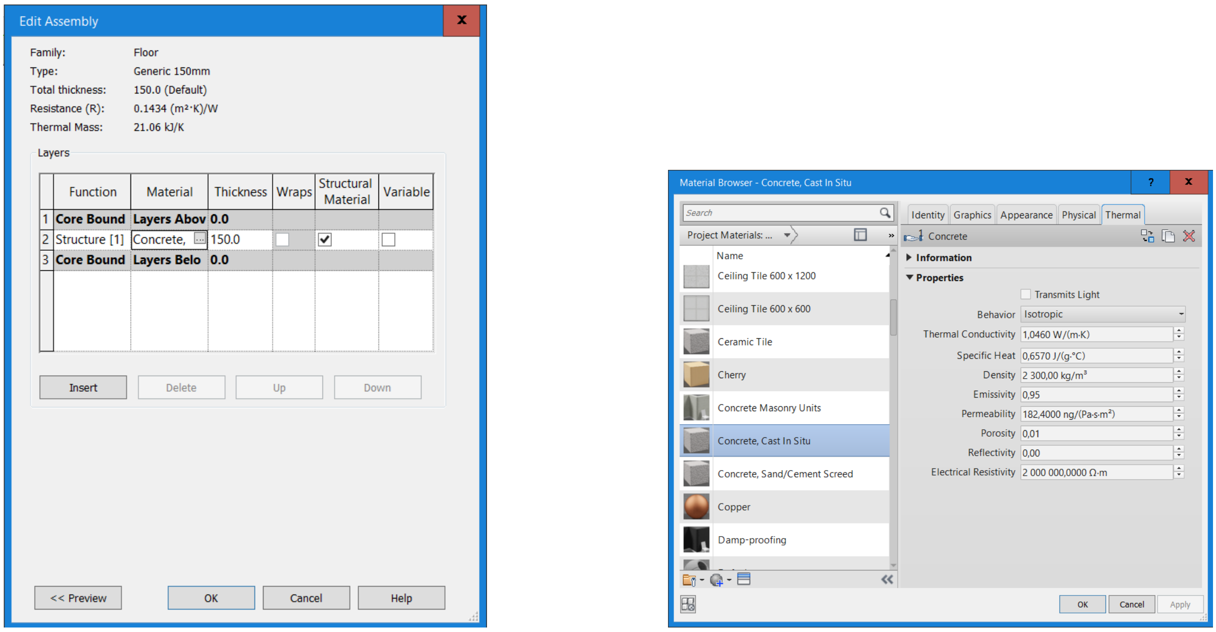


Figure 3.10: On the left, the assembly edition window of the different layers of the floor element, now with concrete as material. On the right, the thermal characteristics of the material.

case. As we did not want to select 300 walls by hand (12 exterior walls per floor times 25 floors), we attributed a different material to the exterior walls in the script, by creating a function that generates a new wall-family in *Revit* that also has only one layer in its structure, but this one made of glass, as we can see in Figure 3.11. As in the case of concrete, glass also has specific thermal characteristics, as we can see in Figure 3.12.

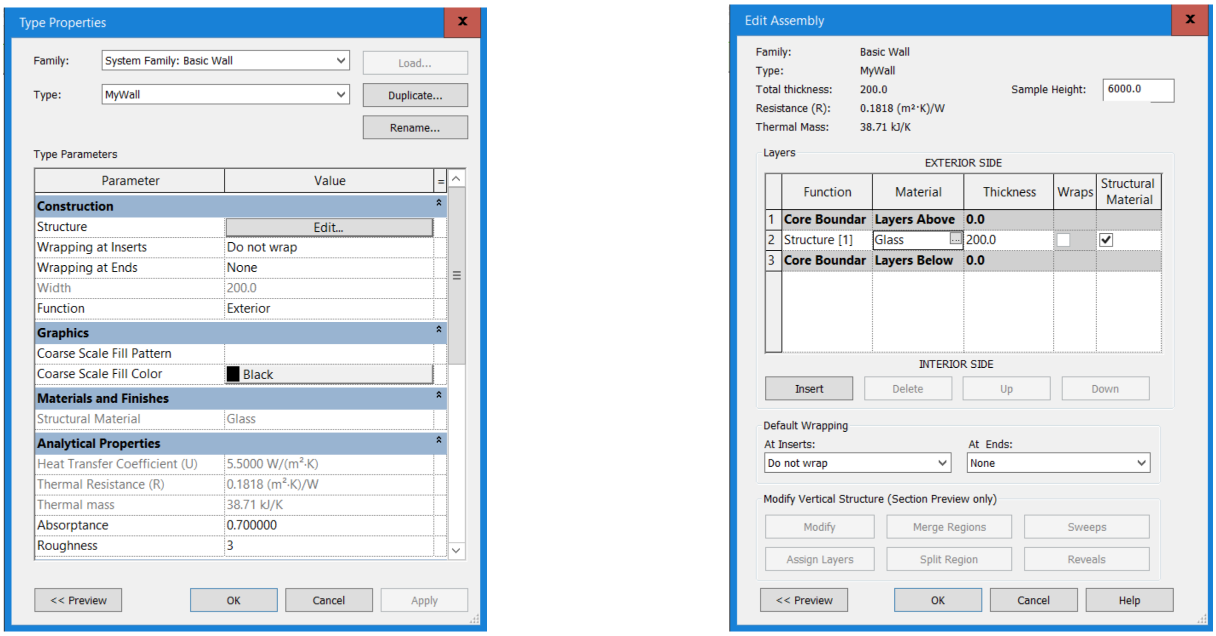


Figure 3.11: On the left, the general characteristics of the glass wall created by the script. On the right, the assembly edition window of the different layers of the glass wall, with glass as material.

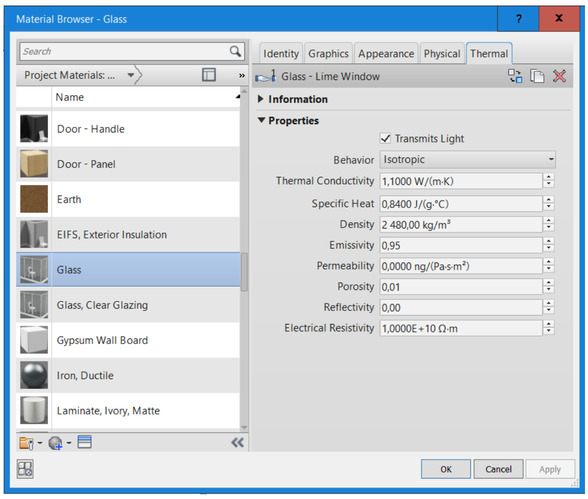


Figure 3.12: The thermal characteristics tab of the material *Glass*.

Building:

This whole process allowed us to parametrically define every component of a simplified version of the building, making it very easy for us to change some aspects of the generated model by only modifying a few values in the script and waiting less than 20 seconds for the computer to generate the whole building with the new values.

The function that generates the final building is a very straightforward one, that simply brings together all the previously defined components, adjusting their dimensions as needed and allowing for some variation factors, explained afterwards. This function receives as parameters the insertion point of the building that corresponds to the center of the square in which the building is built, the side of that square, the number of floors, the number of columns in the exterior row, and the height of each floor.

We can see in Figure 3.13 the model resulting from the script.

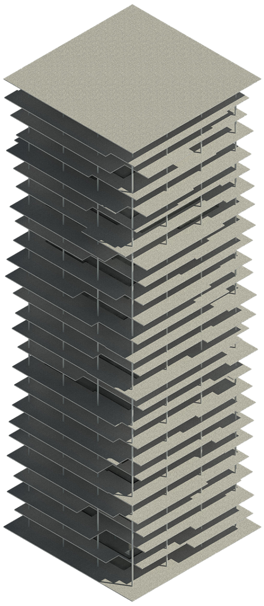


Figure 3.13: The complete model, generated by the script, visualized in Revit.

3.2 Energy Analysis Process

In this section, we will understand how to apply the energy analysis process to the proposed workflow. We will also compare the used approach with a very similar one that uses different tools, comparing both alternatives and justifying the choice made.

3.2.1 Analysis software

For this research, and as referred before, we chose *Autodesk Revit*, a BIM tool that has a direct connection to *Autodesk Insight 360*, an online platform that produces energy analysis on buildings modeled in *Revit*.

Autodesk Revit 2017 is currently one of the *backends* [32] that *Rosetta* supports, allowing us to work with GD through a Textual Programming Language (TPL) and thus reducing drastically the time consumed testing several design options, as well as widening considerably the range of design possibilities.

We could also use a different approach, by using *Rhinoceros*. In this case, we would have the possibility of working either with a Visual Programming Language (VPL) or a TPL, once *Grasshopper* is a VPL editor and connects directly with *Rhinoceros*, and also because *Rhinoceros* is a working *Rosetta backend*. [32]

The problem with this approach would be the energy simulation and analysis. If we perform the simulation on the *Rhinoceros* environment, we can use *DIVA*, an engine that performs daylighting and thermal loads simulation. The problem with this approach is that the thermal load simulation was not working during the time of this research, making this option unfeasible.

Another approach possible, when using *Rhinoceros*, is perform the analysis directly through *Grasshopper*. There are two plug-ins for *Grasshopper* that allow for energy analysis. One is called *Ladybug*, and it allows to perform certain energy analysis related tasks, like importing weather data, draw and customize diagrams, run and view different types of analysis, like radiation or shadow studies.[6] The other is called *Honeybee*, and this one works as a connection to different simulation engines, like Energy Plus, Radiance, Daysim, and OpenStudio, in order to produce different types of simulations, like energy, daylighting, comfort and lighting simulations.[7] Even though these tools seem ideal for the work in hands, they are restricted to the use of a VPL, once they work through *Grasshopper*. Once they use the same tools to perform the energy analysis as *Autodesk Insight 360*, and this last one allows for the use of a TPL, through *Rosetta*, our choice fell on using *Autodesk Revit* to model the building and *Autodesk Insight 360* to perform the energy analysis.

3.2.1.A Process and results

The energy analysis process can take some time, depending on the level of complexity of the model. By doing several analysis in an initial stage, while the project still has a lower complexity level and is still made of generic shapes, such as the exterior volume, we make it possible to run several simulations and choose between a plethora of different options. As the model grows in complexity and detail, its evaluation time will also increase. This increment in time can mean running just two to three simulations, instead of ten or twenty. Using this workflow, we can reach the final design stage with a better notion of the energy consumption of the building being designed, while being able to generate several design options along the way and choose the ones that we want, according to their energy performance and

[6] www.food4rhino.com/app/ladybug-tools
[7] www.food4rhino.com/app/ladybug-tools

the architects aesthetical criteria.

Using the advantages of GD, we also reduce considerably the time spent doing this task. We can generate a model and run the intended simulation. We then read the results and, considering previous experiences in energy simulations, change the parameters given, generate a new model, perform a new simulation, and so on. Using *Rosetta*, the process of generating a new model takes an average of 3 minutes to make the necessary changes to the script (depending on the complexity of the changes; the longest change took almost 10 minutes, while the shortest took just 3 seconds) and about 16 seconds for the computer to generate the new model. As we will see further on, the process here proposed would take too much time to complete using a manual approach, i.e., producing and changing the model directly in *Revit* and running the different energy analysis simulations.

We used this workflow in several analysis stages. We started by creating the original script, that produced the *"Default"* model, with dimensional constrains that tried to mimic the ones found on the original building. By analyzing the different types of slabs, as we have seen before, we could understand the range of dimensions that each parameter could have. We started with the corner points of the slabs, and calculated how much each corner deviated from the original one (see Chapter 3.1.2) in the five different slabs. We took the maximum and minimum value of each one as limits for the range of possible deviations that each corresponding corner could have. For example, the first corner, $p1^*$ (see Figure 3.5), in the **x-axis**, has a minimum variation of an eighth of the side of the square ($a/8$) in the positive direction, and a maximum variation of a sixteenth of the side of the square ($-a/16$) in the negative direction.

After having a reference script, which took approximately a day's work to complete, we wrote a set of ten possible changes, producing a set of ten different models, where we randomly changed some values and left others the same. We can see in Table 3.1, Table 3.2, Table 3.3, Table 3.4, Table 3.5 and Table 3.6, the *"Default"* values, as well as the values randomly chosen for the first set of ten models (from *"1.01"* to *"1.10"*). The limit values for the deviation of the four corners are the same for the slabs and for the walls, and the minimum and maximum range for the breaks between corners are equal in all edges and for both slabs and walls. The parameters of the function that produce the whole building (see Chapter 3.1.2) are kept as in the original script, and are defined as follows:

- p - center point of the building: (0,0,0)
- a - side of the square: 25 meters
- n - number of floors: 25 floors
- $n-ext$ - number of columns in the exterior row: 12 columns
- h - height of each floor: 3 meters

Model	P1			
	X-axis - Minimum	X-axis - Maximum	Y-axis - Minimum	Y-axis - Maximum
Default	$-a/16$	$a/8$	$-a/7$	0
1.01	0	$a/8$	$-a/7$	$a/16$
1.02	$-a/16$	0	0	$a/16$
1.03	0	$a/16$	$-a/7$	0
1.04	$-a/16$	$a/16$	$-a/16$	$a/16$
1.05	$-a/8$	$a/8$	$-a/8$	$a/8$
1.06	$-a/8$	0	$-a/7$	0
1.07	$-a/12$	$a/12$	$-a/7$	0
1.08	0	$a/12$	0	$a/16$
1.09	$-a/12$	0	$-a/16$	$a/16$
1.10	$-a/8$	$a/12$	$-a/16$	$a/16$

Table 3.1: The limit values for deviations in corner $p1$, for the *"Default"* model and for the first set of models (from *"1.01"* to *"1.10"*)

Model	P2			
	X-axis - Minimum	X-axis - Maximum	Y-axis - Minimum	Y-axis - Maximum
Default	$-a/16$	0	$-a/16$	0
1.01	$-a/16$	$a/16$	$-a/16$	$a/16$
1.02	0	$a/16$	0	$a/16$
1.03	$-a/8$	$a/8$	$-a/8$	$a/8$
1.04	$-a/8$	0	$-a/8$	0
1.05	0	$a/8$	0	$a/8$
1.06	$-a/8$	$a/8$	$-a/16$	$a/16$
1.07	$-a/8$	0	0	$a/16$
1.08	0	$a/8$	$-a/16$	0
1.09	$-a/8$	$a/8$	0	$a/16$
1.10	0	$a/8$	$-a/16$	$a/16$

Table 3.2: The limit values for deviations in corner $p2$, for the *"Default"* model and for the first set of models (from *"1.01"* to *"1.10"*)

Model	P3			
	X-axis - Minimum	X-axis - Maximum	Y-axis - Minimum	Y-axis - Maximum
Default	$-a/16$	0	$a/7$	0
1.01	$-a/16$	$a/16$	$-a/7$	0
1.02	0	$a/16$	0	$a/7$
1.03	$-a/12$	$a/12$	$-a/7$	$a/7$
1.04	$-a/12$	0	$-a/8$	$a/8$
1.05	0	$a/12$	$-a/8$	0
1.06	$-a/16$	$a/16$	0	$a/8$
1.07	0	$a/16$	$-a/8$	$a/8$
1.08	$-a/16$	0	0	$a/8$
1.09	0	$a/16$	$-a/8$	0
1.10	$-a/16$	0	$-a/8$	0

Table 3.3: The limit values for deviations in corner $p3$, for the *"Default"* model and for the first set of models (from *"1.01"* to *"1.10"*)

Model	P4			
	X-axis - Minimum	X-axis - Maximum	Y-axis - Minimum	Y-axis - Maximum
Default	0	$a/16$	0	$a/16$
1.01	$-a/16$	$a/16$	$-a/16$	$a/16$
1.02	$-a/16$	0	$-a/16$	0
1.03	$-a/12$	$a/12$	$-a/12$	$a/12$
1.04	$-a/12$	0	$-a/12$	0
1.05	0	$a/12$	0	$a/12$
1.06	0	$a/12$	0	$a/16$
1.07	0	$a/16$	0	$a/12$
1.08	$-a/12$	0	$-a/16$	0
1.09	$-a/16$	0	$-a/12$	0
1.10	$-a/16$	$a/12$	$-a/12$	$a/16$

Table 3.4: The limit values for deviations in corner $p4$, for the *"Default"* model and for the first set of models (from *"1.01"* to *"1.10"*)

Model	D1		D2		D3		D4	
	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum
Default	$a/10$	$a/2$	$a/10$	$a/2$	$a/10$	$a/2$	$a/10$	$a/2$
1.01 to 1.05	$a/10$	$a/2$	$a/10$	$a/2$	$a/10$	$a/2$	$a/10$	$a/2$
1.06 to 1.10	$a/7$	$a/4$	$a/7$	$a/4$	$a/7$	$a/4$	$a/7$	$a/4$

Table 3.5: The limit values for breaks between corners of each edge.

Columns RF	Walls RF	Core Area RF
0.8	0.7	$a/8$

Table 3.6: Reduction factors (RF) for the set of columns, the exterior walls and the core area walls. Applicable to all models tested in this set.

After having the models generated in *Autodesk Revit* using *Rosetta*, we ran the energy simulations of each one of the models. In Table 3.7, we can see the duration of each task. We used a simple manual timer to calculate the duration of each task, and based the duration of the energy analysis on the emails received from *Autodesk Insight 360*. For each analysis that we run through *Revit*, we receive an email from *Autodesk Insight 360* informing that the model was received in their servers, and then another email when the analysis is complete, giving us the time reference needed. In Figure 3.14 we can see the model that resulted from the original script, rendered in *Autodesk Revit*, as well as the set of ten models that resulted from the changes applied. Note that there are less than 10 models, and that three of them display *"error"* on the column corresponding to the time spent generating the model in *Autodesk Revit*. This happens because some of the values, randomly chosen, were not possible to reproduce in *Autodesk Revit*, since the values translated in shapes that *Autodesk Revit* does not recognize as possible, for being too small. If we were using a different backend, these values might not have been translated in errors, once each software has its own restrictions.

Model	Change original script	Generate model	Energy Analysis
Default	-	00 : 17	02 : 00
1.01	09 : 51	00 : 15	18 : 00
1.02	05 : 33	*error*	-
1.03	06 : 33	00 : 15	17 : 00
1.04	05 : 03	00 : 16	14 : 00
1.05	04 : 31	00 : 17	11 : 00
1.06	04 : 25	00 : 13	09 : 00
1.07	03 : 21	00 : 18	09 : 00
1.08	03 : 51	00 : 18	08 : 00
1.09	04 : 05	*error*	-
1.10	07 : 31	*error*	-

Table 3.7: Durations of each task, rounded to the nearest second (minutes:seconds).

After running the simulations, we compared the results, having in mind the Energy Use Intensity (EUI)[8] mean value, as well as maximum and minimum values. In Table 3.8 we can see these three values for each one of the models tested.

[8]See Chapter 2.3.2.C.

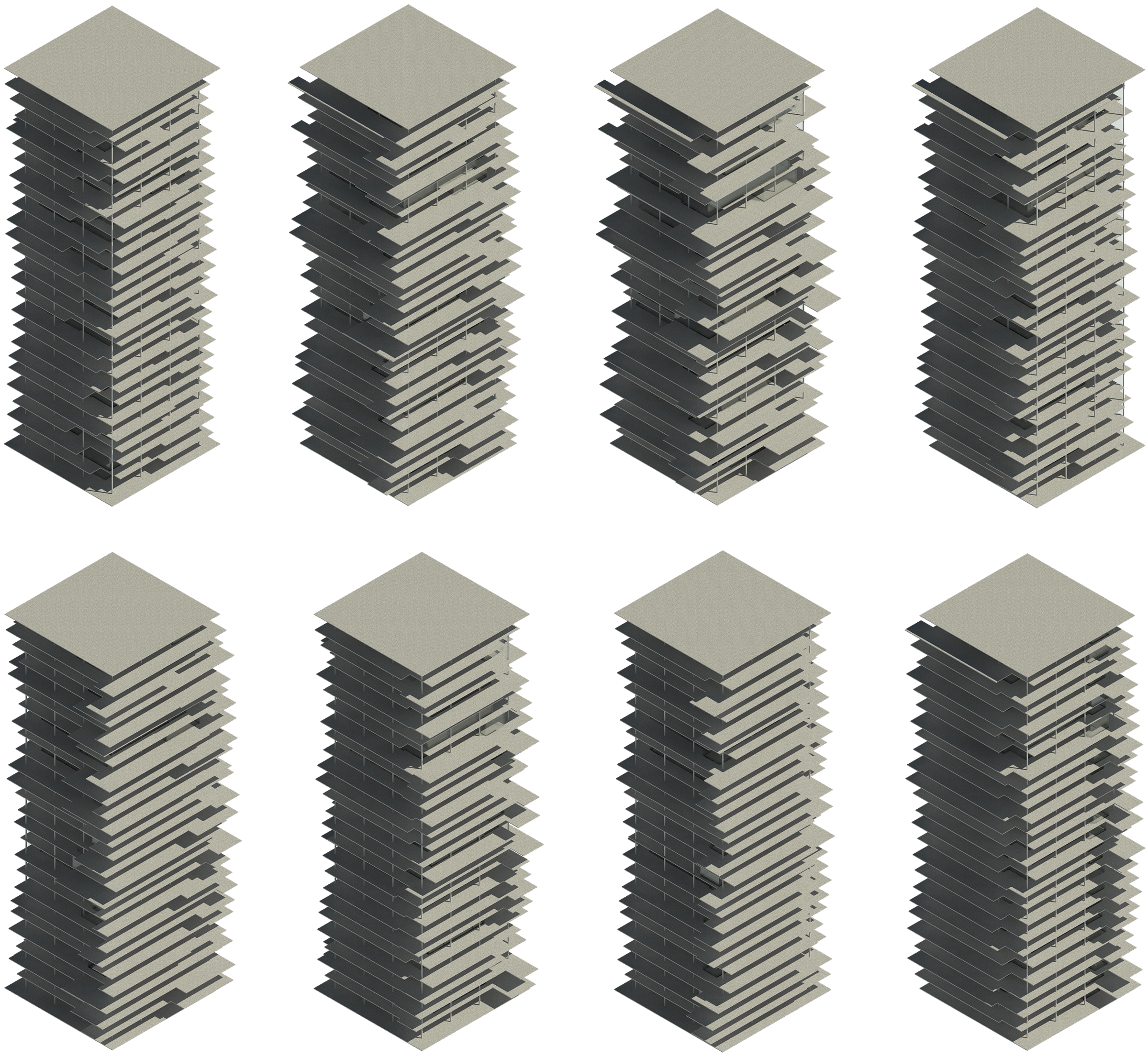


Figure 3.14: Top row, from left to right: *"Default"*, *"1.01"*, *"1.03"*, and *"1.04"*. Bottom row, from left to right: *"1.05"*, *"1.06"*, *"1.07"*, and *"1.08"*.

Model	EUI minimum value	EUI mean value	EUI maximum value
Default	153	332	879
1.01	150	329	870
1.03	150	332	893
1.04	149	327	865
1.05	147	325	849
1.06	143	314	820
1.07	146	318	832
1.08	149	327	861

Table 3.8: EUI minimum, mean and maximum values for each model tested (in $kWh/m^2/year$).

We had as main reference the mean EUI value, leaving us with model *"1.06"* as best performing one, with a mean EUI value of $314kWh/m^2/year$. Before proceeding to the next stage of the project, we have to manually guarantee that the model is in good conditions, i.e., that the relation between walls and slabs is adequate (as referred before, the walls can not surpass the available area of the slabs immediately above and below them). We quickly went through all the floors, making sure there were no walls outside of the slabs. We noticed a corner where some walls were outside the slab area, and so took that as reference for the next stage on the process.

For the second stage of this workflow, we took the values used to produce model *"1.06"* and changed them, according to the type of ranges used for that model. Now the numbers are not as random as before, but thought of according to the model we chose as a starting point for this second stage. We kept some values the same, once they were working well. We decided to change only the values in corner $p2$, once that was the corner where a wall had to be changed. We also changed the limit values for the breaks in the edges. In Table 3.9 and Table 3.10 we can see the values that were changed for this stage (models *"2.01"* to *"2.10"*), keeping the values used in model *"1.06"* as a comparison point.

Model	**P2**			
	X-axis - Minimum	**X-axis - Maximum**	**Y-axis - Minimum**	**Y-axis - Maximum**
1.06	$-a/8$	$a/8$	$-a/16$	$a/16$
2.01	0	$a/8$	$-a/16$	$a/16$
2.02	$-a/16$	$a/8$	$-a/16$	$a/16$
2.03	$-a/8$	$a/8$	0	$a/16$
2.04	0	$a/8$	0	$a/16$
2.05	$-a/16$	$a/8$	0	$a/16$
2.06	0	$a/8$	$-a/16$	$a/16$
2.07	$-a/16$	$a/8$	$-a/16$	$a/16$
2.08	$-a/8$	$a/8$	0	$a/16$
2.09	0	$a/8$	0	$a/16$
2.10	$-a/16$	$a/8$	0	$a/16$

Table 3.9: The limit values for deviations in corner $p2$, for the *"1.06"* model and for the second set of models (from *"2.01"* to *"2.10"*)

Model	**D1**		**D2**		**D3**		**D4**	
	Minimum	**Maximum**	**Minimum**	**Maximum**	**Minimum**	**Maximum**	**Minimum**	**Maximum**
1.06	$a/7$	$a/4$	$a/7$	$a/4$	$a/7$	$a/4$	$a/7$	$a/4$
2.01 to 2.05	$a/7$	$a/4$	$a/7$	$a/4$	$a/7$	$a/4$	$a/7$	$a/4$
2.06 to 2.10	$a/10$	$a/2$	$a/10$	$a/2$	$a/10$	$a/2$	$a/10$	$a/2$

Table 3.10: The limit values for breaks between corners of each edge.

We can see the duration of the different tasks referred before, in the first stage, but now for the second set of models, in Table 3.11. Once again, we looked at the EUI values, as shown in Table 3.12.

Model	Change original script	Generate model	Energy Analysis
2.01	$01:01$	$00:15$	$02:00$
2.02	$00:43$	$00:18$	$03:00$
2.03	$00:44$	$00:19$	$03:00$
2.04	$00:49$	$00:14$	$03:00$
2.05	$01:01$	$00:16$	$04:00$
2.06	$01:04$	$00:17$	$08:00$
2.07	$01:13$	$00:17$	$11:00$
2.08	$01:08$	$00:15$	$10:00$
2.09	$01:05$	$00:17$	$09:00$
2.10	$01:06$	$00:18$	$11:00$

Table 3.11: Durations of each task, rounded to the nearest second (minutes:seconds).

Model	EUI minimum value	EUI mean value	EUI maximum value
1.06	143	314	820
2.01	146	317	820
2.02	145	316	823
2.03	143	312	812
2.04	145	316	817
2.05	144	315	820
2.06	147	318	829
2.07	137	306	814
2.08	146	318	843
2.09	147	318	826
2.10	137	305	813

Table 3.12: EUI minimum, mean and maximum values for each model tested (in $kWh/m^2/year$).

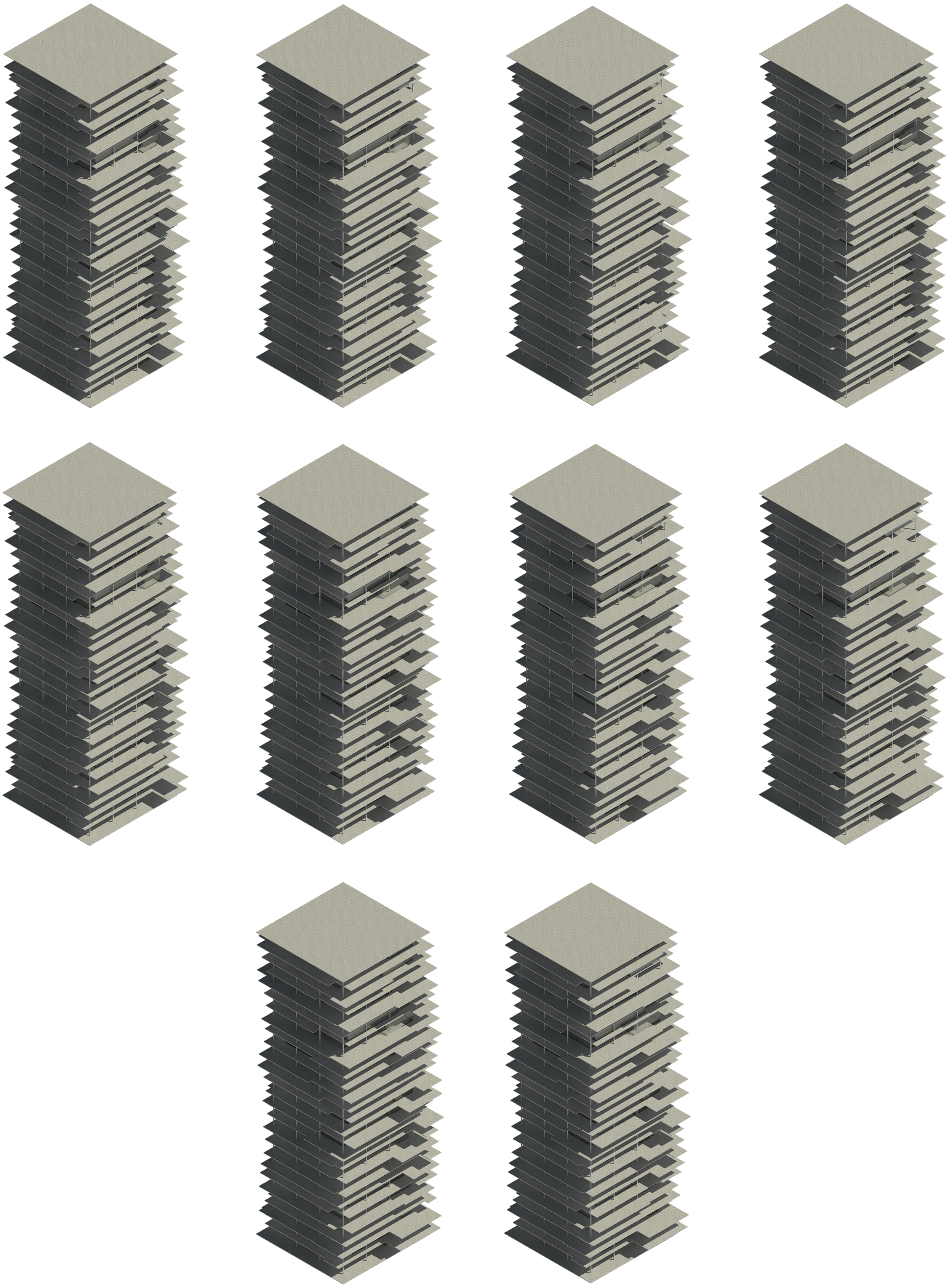


Figure 3.15: Top row, from left to right: *"2.01"*, *"2.02"*, *"2.03"*, and *"2.04"*. Middle row, from left to right: *"2.05"*, *"2.06"*, *"2.07"*, and *"2.08"*. Bottom row, from left to right: *"2.09"* and *"2.10"*.

Looking at the values in Table 3.12, we see that model *"2.10"* is the best performing one. Once again, we checked for any incoherences in the model. In this model, we also found a wall that was out of place, and so we opted to check the second best performing one, which is model *2.07*. We checked for incoherences in this model as well, for if this second one has no incoherences, then it has better values than *"2.10"*. Model *"2.07"* is better constructed than model *"2.10"*, so we will choose that one to continue on to the next stage.

As we still found some corners that could be adjusted, we changed a few values on corners $p1$ and $p2$. This time, as we are approaching a better result, we reduced the spectrum of tests performed to five models. So, we wrote the changes needed to generate the third set of models, from *"3.01"* to *"3.05"*. In Table 3.13 and Table 3.14 we can see the values changed from model *"2.07"* to the third set of models. Note that we only changed values in the **x-axis** for corner $p1$ and values in the **y-axis** for corner $p2$. In Table 3.15 we can see the durations of the tasks performed. In Figure 3.16 we see the models generated in this third set.

Model	P1		P2	
	X-axis - Minimum	X-axis - Maximum	Y-axis - Minimum	Y-axis - Maximum
2.07	$-a/8$	0	$-a/16$	$a/16$
3.01	$-a/12$	0	0	$a/18$
3.02	$-a/16$	0	$-a/16$	$a/18$
3.03	$-a/12$	$a/16$	$-a/12$	$a/18$
3.04	$-a/16$	$a/16$	0	$a/20$
3.05	$-a/8$	0	$-a/16$	$a/16$

Table 3.13: The limit values for deviations in corners $p1$ (x-axis) and $p2$ (y-axis), for the *"2.07"* model and for the third set of models (from *"3.01"* to *"3.05"*). Applied to walls.

Model	P1		P2	
	X-axis - Minimum	X-axis - Maximum	Y-axis - Minimum	Y-axis - Maximum
2.07	$-a/8$	0	$-a/16$	$a/16$
3.01 to 3.04	$-a/8$	0	$-a/16$	$a/16$
3.05	$-a/6$	0	$-a/16$	$a/8$

Table 3.14: The limit values for deviations in corners $p1$ (x-axis) and $p2$ (y-axis), for the *"2.07"* model and for the third set of models (from *"3.01"* to *"3.05"*). Applied to slabs.

Model	Change original script	Generate model	Energy Analysis
3.01	$00:38$	$00:14$	$05:00$
3.02	$00:47$	$00:17$	$03:00$
3.03	$00:41$	$00:15$	$02:00$
3.04	$00:49$	$00:15$	$02:00$
3.05	$00:19$	$00:16$	$03:00$

Table 3.15: Durations of each task, rounded to the nearest second (minutes:seconds).

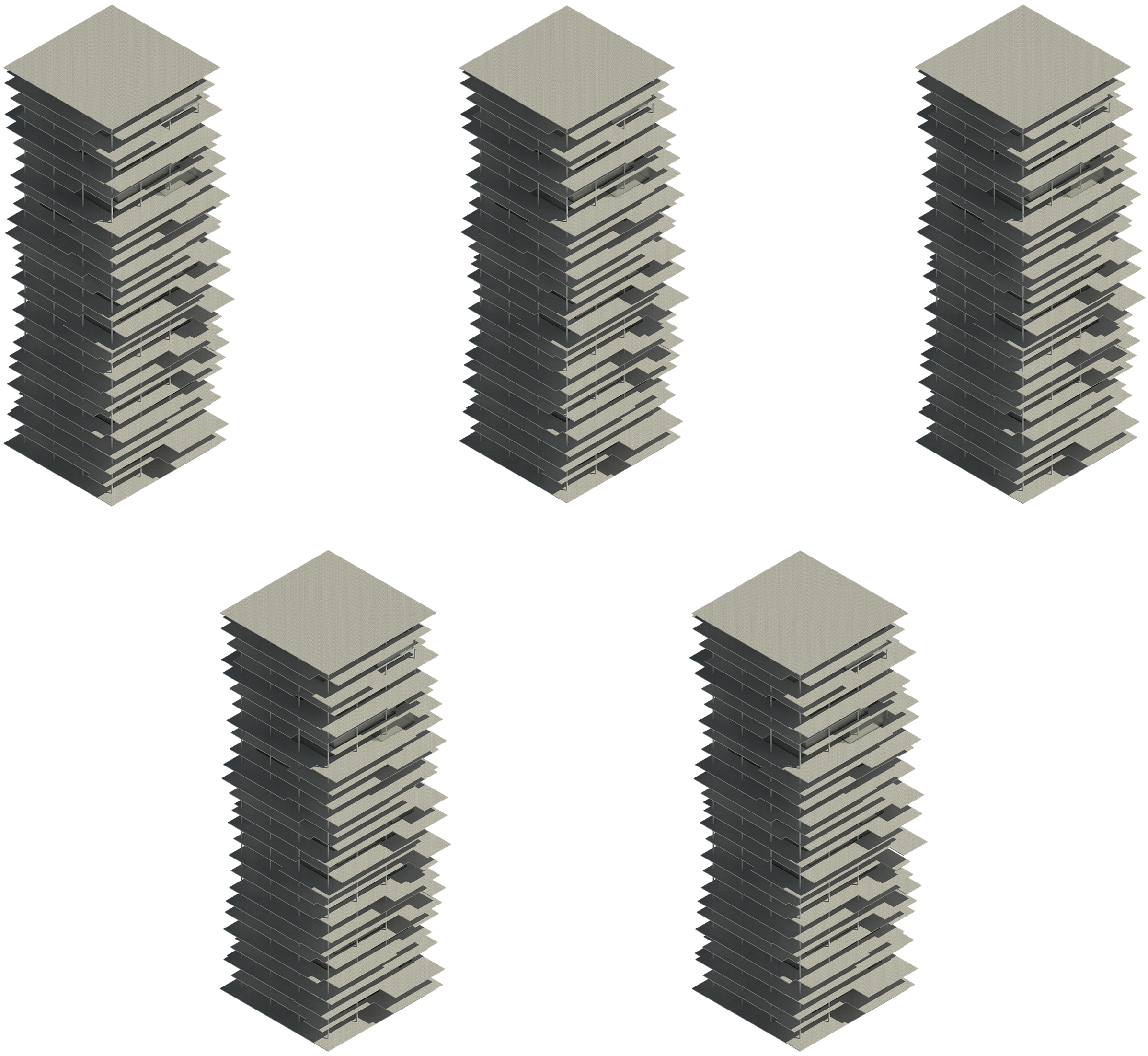


Figure 3.16: Top row, from left to right: *"3.01"*, *"3.02"*, and *"3.03"*. Bottom row, from left to right: *"3.04"* and *"3.05"*.

After running the analysis on the five models generated, we reached a point where the building is no longer improving, once the best performing model from this set has the same results as the model that we used as a starting point for this same set (see Table 3.16). This happens because we started to reduce the variation intervals, and the models tend to become more and more stable, and so this translates in results that no longer improve the energy consumption of the building.

Model	EUI minimum value	EUI mean value	EUI maximum value
2.07	137	306	814
3.01	137	307	814
3.02	138	307	810
3.03	146	318	835
3.04	137	308	817
3.05	137	306	814

Table 3.16: EUI minimum, mean and maximum values for each model tested (in $kWh/m^2/year$).

Even thought this last set of models has not improved energy use, it improved the consistency of the model, and so we will consider model *"3.05"* as our final one.

In these three sets, we essentially changed the consequence that the shape of the slabs and exterior walls have in what energy performance is regarded, i.e., the shadows. This is what changing the shape of these elements contributes in reducing energy consumption.

For the fourth set, we opted by a rotation of the building, thus changing its orientation towards the sun. As the plot available for the building is a square, corresponding to the size of the building, we only performed three tests, rotating in the three available options, to ascertain if a rotation of the building can contribute to a better energy performance. Model *"4.01"* has a rotation of 90 degrees counter-clockwise, model *"4.02"* has a rotation of 180 degrees, and model *"4.03"* has a rotation of 90 degrees clockwise. In Table 3.17 we see the durations of each task, and in Table 3.18 we can see the results obtained. In Figure 3.17 we can see the models produced in this set.

Model	Change original script	Generate model	Energy Analysis
4.01	00 : 13	00 : 19	03 : 00
4.02	00 : 08	00 : 14	03 : 00
4.03	00 : 09	00 : 16	03 : 00

Table 3.17: Durations of each task, rounded to the nearest second (minutes:seconds).

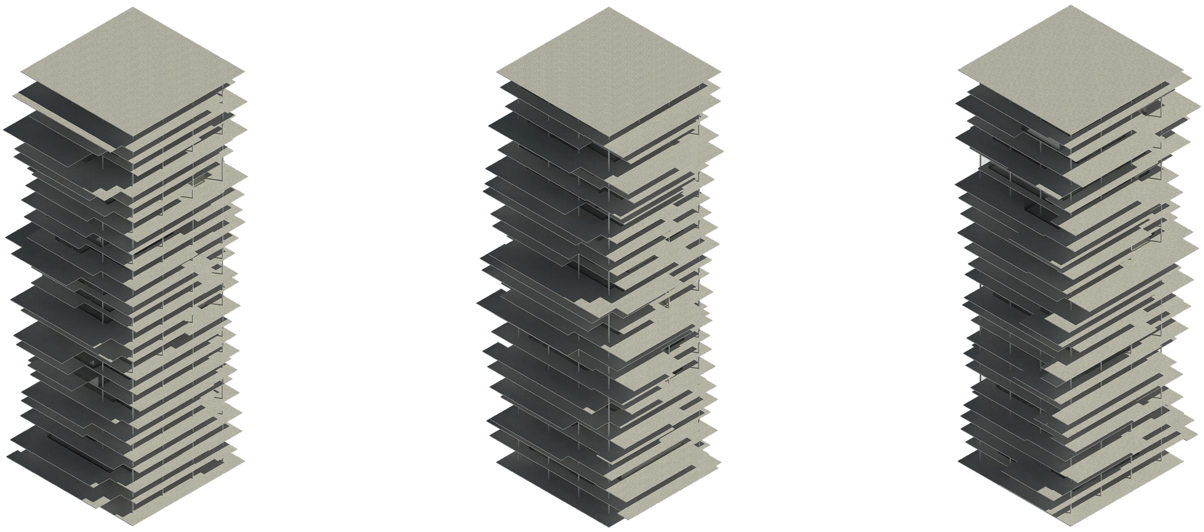


Figure 3.17: From left to right: *"4.01"*, *"4.02"*, and *"4.03"*.

Model	EUI minimum value	EUI mean value	EUI maximum value
3.05	137	306	814
4.01	154	323	829
4.02	146	321	851
4.03	156	329	841

Table 3.18: EUI minimum, mean and maximum values for each model tested (in $kWh/m^2/year$).

As we can see from the results obtained, there were no improvements in the fourth set of models, since the shadows previously improved, now have a different orientation, and so no longer contribute in improving energy consumption. Being so, we will continue on to the next testing stage with the model that performed better until now, which is model *"3.05"*.

In the fifth set of models, we changed the size of the core area. This area is related to the total size of the square in which the building is inserted. The original model has a core area size of one-eighth of the side of the square. Once again, we only performed three tests. The first model (*"5.01"*) has one-seventh of the side of the square, the second model (*"5.02"*) has one-ninth of the side of the square, and the third model (*"5.03"*) has one-tenth of the side of the square. In Table 3.19 we see the durations of each task, and in Table 3.20 we can see the results obtained. In Figure 3.18 we can see the models produced in this set.

Model	Change original script	Generate model	Energy Analysis
5.01	$00:13$	$00:16$	$04:00$
5.02	$00:06$	$00:15$	$12:00$
5.03	$00:07$	$00:16$	$08:00$

Table 3.19: Durations of each task, rounded to the nearest second (minutes:seconds).

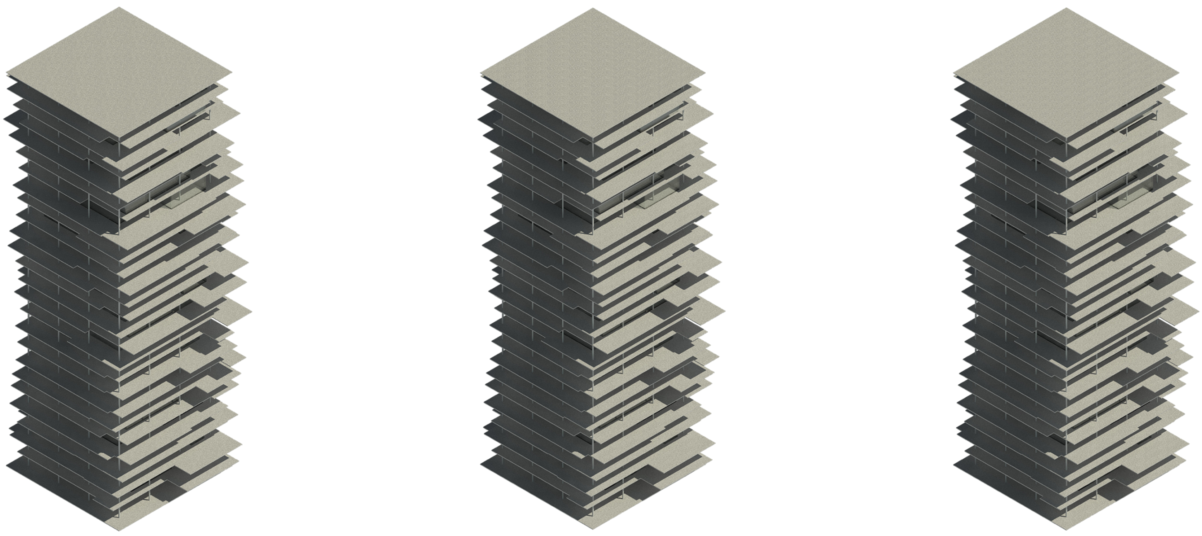


Figure 3.18: From left to right: *"5.01"*, *"5.02"*, and *"5.03"*.

The model chosen to continue on to the next stage is model *"5.01"*, as it was the best performing one. In the sixth and final set of tests performed, we changed the height of the floors. The original model has a height of 3 meters for each floor. In this set of models we tested five different options: 3,20 meters (model *"6.01"*), 3,40 meters (model *"6.02"*), 3,60 meters (model *"6.03"*), 3,80 meters (model *"6.04"*) and 4 meters (model *"6.05"*). In Table 3.21 we see the durations of each task, and in Table 3.22 we can see

Model	EUI minimum value	EUI mean value	EUI maximum value
3.05	137	306	814
5.01	135	303	813
5.02	138	307	811
5.03	139	309	809

Table 3.20: EUI minimum, mean and maximum values for each model tested (in $kWh/m^2/year$).

the results obtained. In Figure 3.19 we can see the models produced in this set.

Model	Change original script	Generate model	Energy Analysis
6.01	$00:04$	$00:19$	$06:00$
6.02	$00:03$	$00:19$	$11:00$
6.03	$00:04$	$00:17$	$13:00$
6.04	$00:03$	$00:17$	$12:00$
6.05	$00:03$	$00:20$	$09:00$

Table 3.21: Durations of each task, rounded to the nearest second (minutes:seconds).

Model	EUI minimum value	EUI mean value	EUI maximum value
5.01	135	303	813
6.01	136	311	846
6.02	137	316	876
6.03	138	323	908
6.04	139	330	939
6.05	140	337	972

Table 3.22: EUI minimum, mean and maximum values for each model tested (in $kWh/m^2/year$).

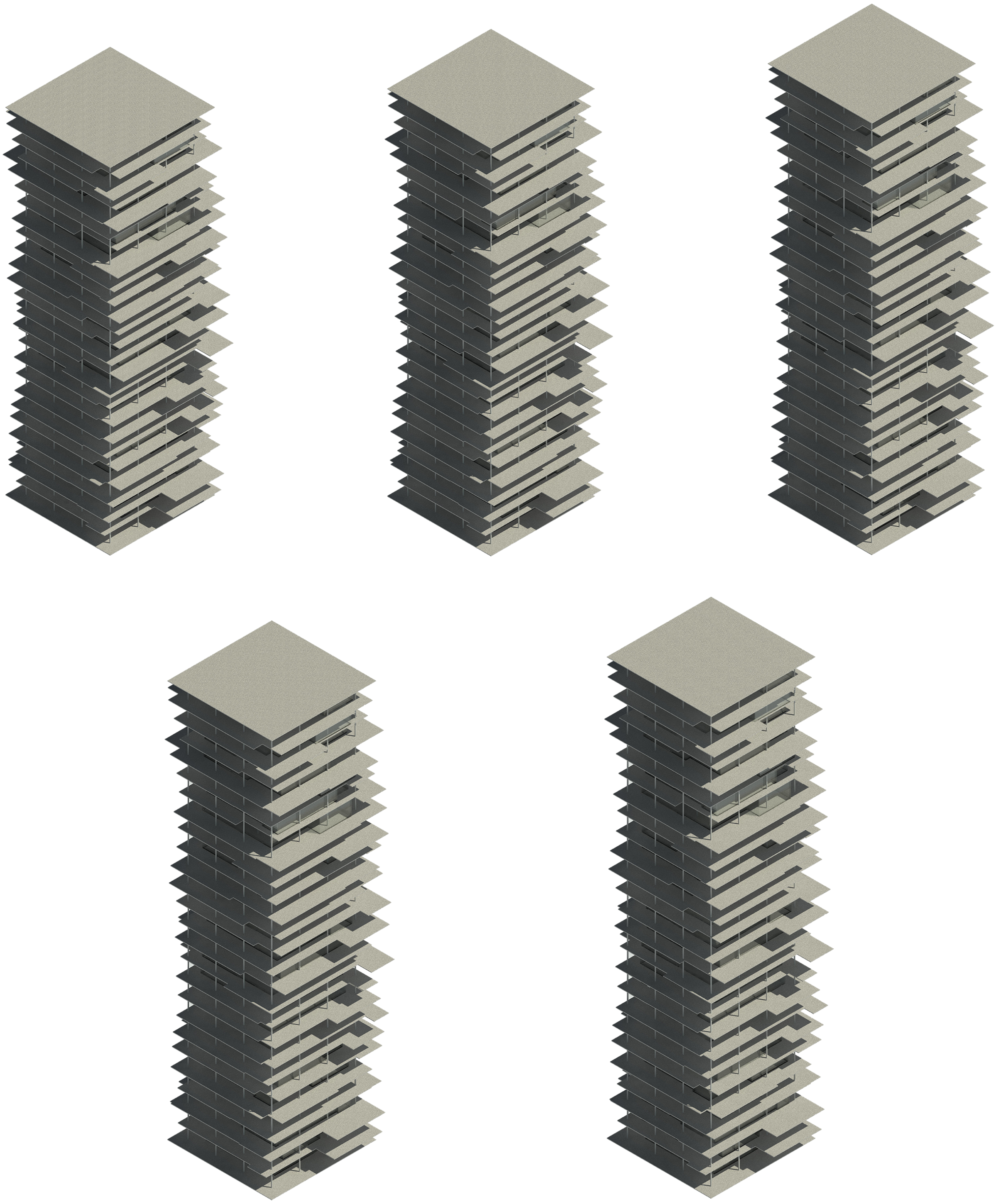


Figure 3.19: Top row, from left to right: *"6.01"*, *"6.02"*, and *"6.03"*. Bottom row, from left to right: *"6.04"* and *"6.05"*.

As we can see, the results produced by the sixth set did not improve the building's EUI value, and so the final choice will be model *"5.01"*. In Figure 3.20 we can see a graphic comparing all the EUI mean values of the several models tested. The models chosen in each set of tests to continue on to the next one are represented in green. If we wanted to continue producing tests, we would use model *"5.01"* and keep experimenting possible changes to improve the building's energy use.

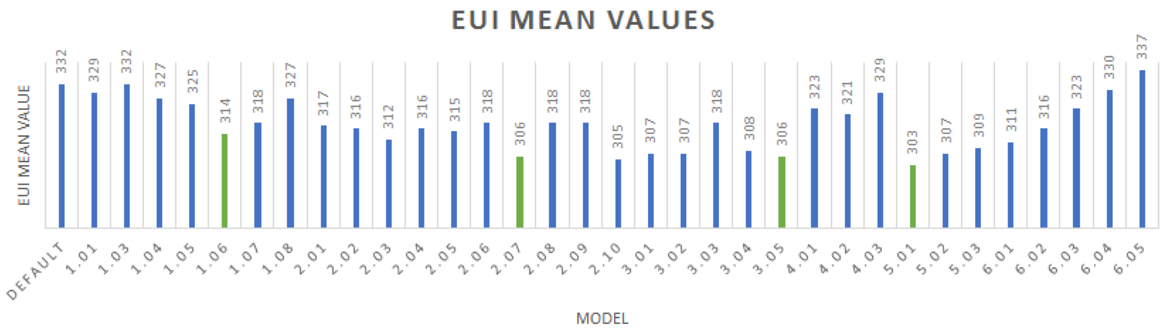


Figure 3.20: EUI results for all the models tested. In green, the models chosen to continue on to the next stage of tests.

Note that all these tests were rapidly done. All the changes were quick to implement, the models generated in seconds, and analysis made in a very short amount of time. We display the results of 34 analysis in 34 different models. All this process took, in total, a little more than 7 hours.

For comparison purposes, we represented the time consumed in these tasks in the graphic seen in Figure 3.21, comparing with the time that would be spent if we did the task manually, modeling everything directly in *Autodesk Revit*, instead of using GD and *Rosetta*. The graphic is just an approximation, based on user experience during the time of this research.

As seen in the image caption, the dark blue color represents the necessary time to produce the initial model in *Autodesk Revit*. As seen previously, using a GD approach has a lot of benefits in complex models, but it takes almost the same time to produce the original script has it would take to model the building directly in *Autodesk Revit*. In blue, we see an approximation of the time needed to perform the energy analysis. We noted that this duration varies a lot, once it depends on the availability of *Autodesk*'s servers to run the whole process. In this graphic we represent this as an average duration. In green is the time needed to apply changes to the original script, in order to produce a new model, and in beige is the time that *Rosetta* takes to produce the three-dimensional model in *Autodesk Revit* based on the script provided. In light beige is the average time needed to apply the same type of changes in the three-dimensional model, if we do it directly in *Autodesk Revit*. When we talk about the same type of changes, it means that we change all the slabs and exterior walls in the model, for example, as we did in the first three sets of tests. We see the big difference here, and this shows why GD can be so

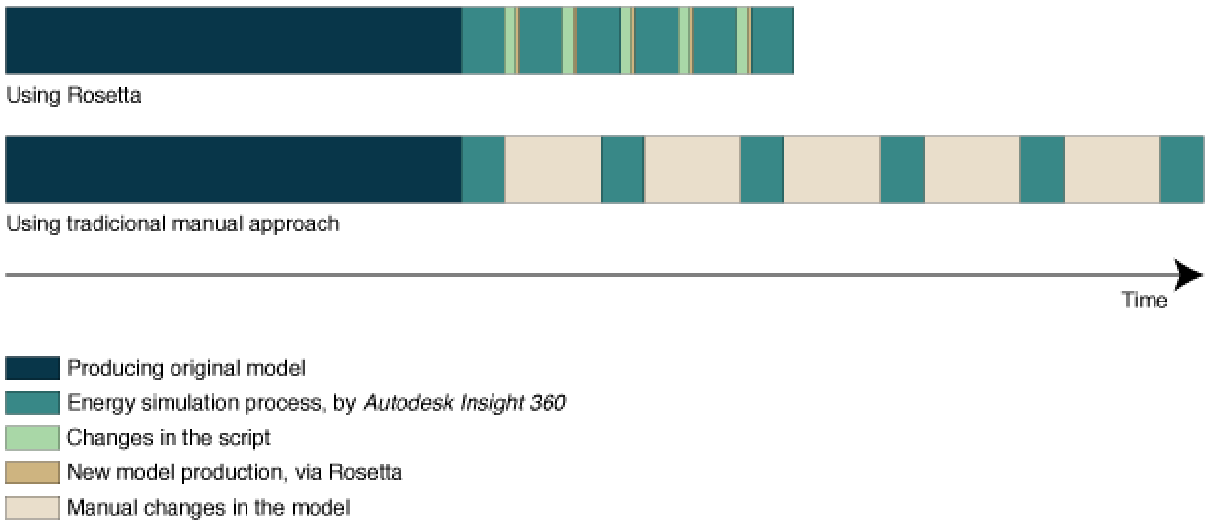


Figure 3.21: An estimation, based on user experience, of the time consumed for each task.

helpful in these tasks. The time that we would need to change, by hand, each one of the slabs and walls, while trying to keep the building with its sense of randomness, would be much greater than doing it by changing a few values in a script and running it again. We are talking about a task that can take about one hour, reduced to just one or two minutes, depending on the complexity of the changes in the model and the amount of values changed in the script.

Besides being clearly faster to change values in a script rather then trying to implement the same type of changes by hand, we have an additional advantage. While modeling directly in *Autodesk Revit*, we change the model, run the simulation, read the results, change the model again, and so on. Using GD, we can create several copies of the same script, each one with the different changes that we want to implement, and test more options at once. As already referred, *Autodesk Insight 360* is an online platform that uses *Autodesk*'s servers to process all the analysis, which allows us to run several analysis simultaneously. Doing this in a manual approach would be rather difficult, once keeping track of the changes made in a three-dimensional model is much more complicated than doing the same in a script. So, taking advantage of this simultaneous analysis would be much more complicated in a manual approach. By using GD, we produced six sets of models by applying different values for each one, which resulted in the possibility of analyzing several models at the same time. This allowed us to test a wider variety of options, thus leading to a more efficient result. With each set, we took a very short amount of time producing the different models, and then ran the simulations all at once. Once they were ready, we just needed to compare the results and choose the best performing one. That would mean that that specific script had better values than the others, and so we created the following set of models having in mind the values that worked better so far. While creating the following set, we decided which attribute we wanted to change and test, as explained before. This allowed us to progressively work towards a better energy performance, by testing various options for each parameter and keeping the values that worked better.

Even though this proposed approach is clearly more advantageous for this case study, we want to emphasize that this type of methodology is only suitable for complex designs. Taking the same type of graphic, we compare the timings that would be needed to generate a simple building with both approaches (GD and manual), as well as the timings needed to perform the same tasks in a complex building (see Figure 3.22 and 3.23).

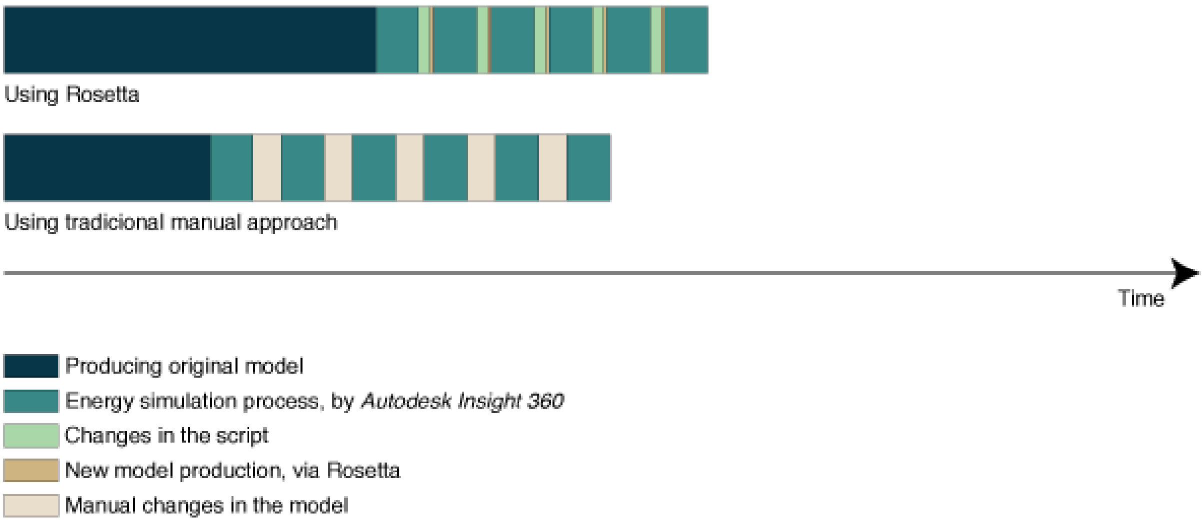


Figure 3.22: An estimation, based on user experience, of the time consumed for each task, assuming a simple building (normalized to the time consumed by using *Rosetta*).

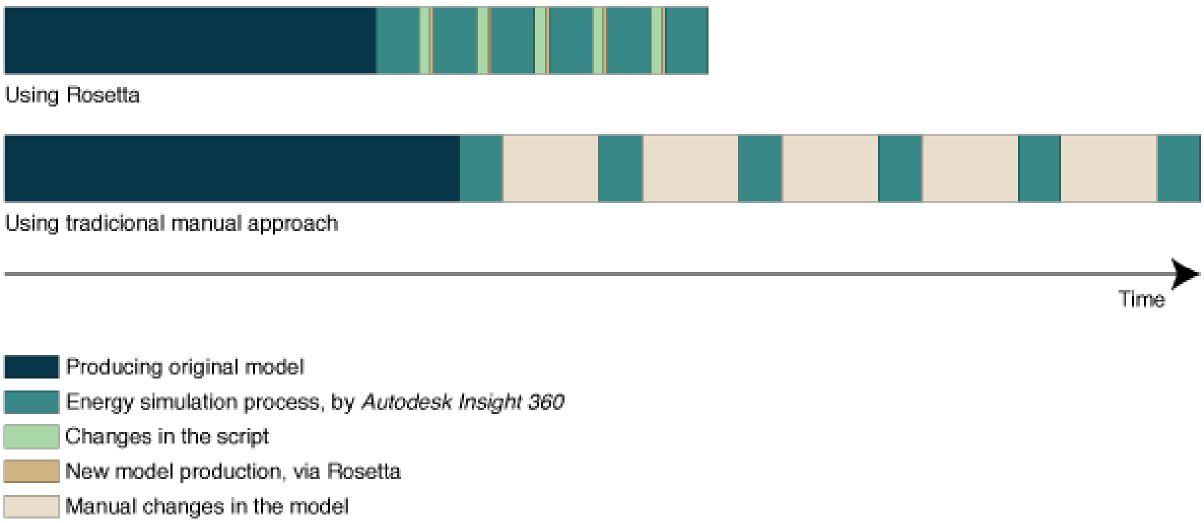


Figure 3.23: An estimation, based on user experience, of the time consumed for each task, assuming a complex building (normalized to the time consumed by using *Rosetta*).

As we can see, when designing a simple building, like Case Study I (see Chapter 2.2), a GD approach would not be helpful in reducing the time spent to perform these tasks. On the contrary, it takes much more time, since it is a small project with many different spaces, and there is no rule that can describe all of them, even with changing variables. In the long term, it might be helpful, once the changes in the model are rapidly implemented in the script and a new model generated in seconds. If we perform a great number of analysis and consequent changes, a GD approach would eventually perform better in this case. But in the short term, with a small number of analysis, a manual approach is more efficient for a simple building.

On the other hand, when designing a complex building, like the examples given previously (see Chapter 2.4), using GD can mean a considerable decrease in time spent generating the three-dimensional model. In this type of buildings, even though the shapes are all different from each other, they all comply with a rule that governs their general shape. This means that we just need to define that rule, and then apply the correct variables to produce the result intended.

3.3 An Automatic Approach to the Energy Analysis Process

In this work, we took the principle of genetic evolution and applied it to our model, creating the variations by hand. We started by doing this to better grasp the evolution of the model on a first stage, instead of the automatic approach. This process was already made faster by the use of GD, as it took less than 20 seconds for each model to be generated, instead of doing all the changes in the model by hand. To automate the process even further, we can use a Genetic Algorithm (GA), by defining the EUI value as the objective to be achieved (in this case, the smallest EUI value possible). We will now succinctly explain how GAs work.

A GA is an evolutive procedure based on the Darwinian notion of 'the survival of the fittest'. [33] It's a process of continuous search, evaluation and new proposal, in a cyclic way, in order to achieve more adequate results with each cycle.

In order to explain this algorithm's behavior, we will first define its terminology. An *individual* is a solution to the problem at hand. A *population* is a set of individuals. Each population created is called a *generation*. The set of parameters of interest of an individual is an *allele*. The *fitness* of an individual is its value, on that point, of the *objective function*. We better understand some of these concepts using a practical example, like the one on Figure **??**.

This type of algorithm searches random individuals in a universe of possibilities presented by the user, which is defined as the *design space*. The *design space* includes all the individuals that respond to the specific characteristics defined (see Figure **??**, image on the right). This initial population is evaluated according to an objective function, and the results are used to create a new generation, with

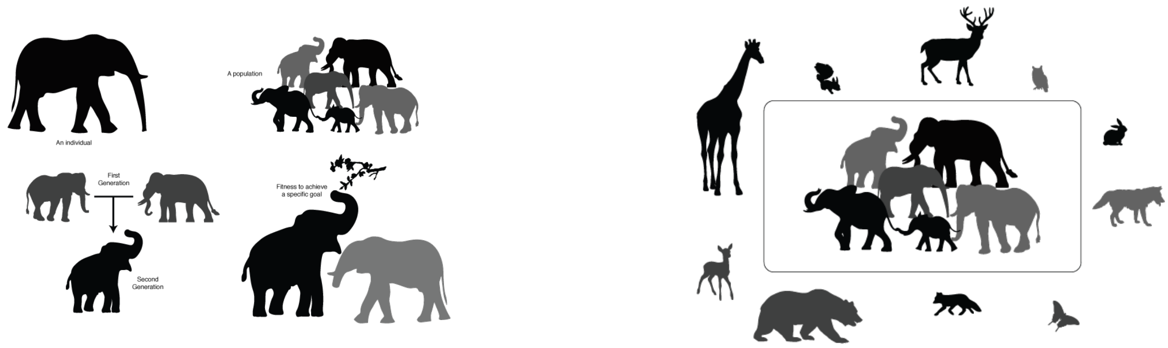


Figure 3.24: On the left, we can see some of the concepts regarding the explanation given previously about algorithms terminology. On the right, the *design space* is composed by the individuals that respond to the specific characteristics, while others are outside that space, even though they still exist.

better performance and fitness than the previous. The new population is, in turn, evaluated, and the cycle repeats itself until it reaches the limit imposed by the user.

In a simplified way, a genetic algorithm uses three types of operations to create new solutions: *reproduction*, *crossing* and *mutation*. Reproduction is based on the fitness of an individual, because the better its fitness, the more likely it will be chosen for reproduction. Crossing chooses two random chromosomes and switches them, creating a new individual. Mutation will change an allele of an individual, also randomly, finding new possibilities and making a *general* optimization, and thus avoiding an improvement of only one type of variation, this last called a *local* optimization.

By allying *Rosetta* with a GA, we are also able to mechanize this process, using an optimization approach and making the program create several different options and run a simulation for each one, giving as a result a set of analyses for the architect to choose from, thus being able to continue improving his project based on a better design.

A workflow that explores the advantages of GD with GAs has already been tested in the past, e.g. [34], which proposes a similar workflow, but using different tools. H. Chalabee uses *Rhinoceros 3D* as a visualization tool, using *Grasshopper* to generate the model. He then uses *DIVA* to perform the energy simulations, and a tool called *Galapagos*, that is presented as a 'generic platform for the application of Evolutionary Algorithms to be used on a wide variety of problems by non-programmers.'[9]

What we propose is to continue the research here presented, in order to achieve the same goal, but using a TPL. As seen before, we consider TPLs as better tools for complex designs that VPLs. By using *Grasshopper*, H. Chalabee is restricting his work to evaluation and improving simple designs. Using that approach with Case Study II would prove to be much more challenging, and require a lot of extra effort by the simple fact that VPLs do not work well with this level of design complexity.

[9] www.grasshopper3d.com/group/galapagos

4

Conclusion

Contents

4.1 Conclusions

Energy analysis tools are increasingly important in Architecture and Generative Design (GD) allowing us to automate their use, reducing time and effort and greatly expanding the design options available to the architect.

Using energy analysis tools in early stages of the design process helps improve sustainable architecture, as energy consumption concerns start at the beginning of the design and not only at the end, where the energy analysis is nowadays typically done.

This research proposes the use of a workflow that helps architects design sustainable buildings, by bringing together GD with Building Information Model(ing) (BIM) tools and energy simulation engines.

Using this workflow, architects use GD to create a parametric model in a BIM tool. The model is generated several times, with different parameters, and each one is analyzed in terms of energy performance, comparing the final results and allowing the architect to choose the best solution among the ones tested.

Besides saving precious time, this also opens a new range of options for architects, as it makes it possible to test many more design solutions than before, thus helping reach a better result than if just two or three were simulated, and reducing considerably the time spent doing the task.

4.2 Future Work

In this work, we took the principle of genetic evolution and applied it to our model, creating the variations by hand. This process was already made faster by the use of GD, as it took less than 20 seconds for each model to be generated, instead of doing all the changes in the model by hand. To automate the process even further, we can use a Genetic Algorithm (GA), an evolutive procedure based on the Darwinian notion of 'the survival of the fittest', [33] by defining the Energy Use Intensity (EUI) value as the objective to be achieved (in this case, the smallest EUI value possible).

As future work we are planning to develop an automatic optimization tool, improving the presented workflow and taking it further. We can already see some examples of optimization being applied, like Asl, [35] who proposes an optimization tool that uses *Autodesk Revit* and *Autodesk Green Building Studio* (recently updated to *Autodesk Insight 360*), generating alternative options in BIM and automatically performing energy simulations, giving as a result the optimal solution found between the ones tested. This work uses file exportation to send the model from *Autodesk Revit* to *Autodesk Green Building Studio*, and then again back to *Autodesk Revit*. It receives the parameters that are allowed to change and generates *Green Building Studio* files to be analyzed. This work was used to improve window dimensions in a two story house.

The goal is to take this further, since we want to change several parameters in different elements and in different stages of the design. For example, in the second case study presented in our research, we improved whole building energy consumption but, in the future, we plan to reach a stage where we also optimize the interior divisions, creating a better distribution of apartments.

For this to be possible, we identified some obstacles that need to be resolved.

One of the first obstacles encountered was guarantying the consistency of the model, which in this case study meant that the walls could not surpass the slab area available directly above and below. This is easily done in a Computer-Aided Design (CAD) software, with an intersection of both slab areas. However, this feature is not available in the BIM software used, since the slabs are in different levels, and so the intersection is not possible. We propose that a solution to this obstacle is encountered, so that this type of situation does not pose as a problem in future works.

One of the challenges already referred is the fact that, currently, we can only specify one material in the script, and thus produce a family-type with only one layer of that material. The possibility of adding materials directly in the script was fundamental in the case study tested, but this feature still needs improvement. In the future, we should be able to create several families directly in the script, so that the architect can create his own families and use them in his design, thus generating the model already with all the family-types applied to the different elements.

One other challenge to be overcome is the automation of the energy analysis process. During the testing phase, *Autodesk* updated their energy analysis tool, previously called *Green Building Studio*, to *Insight 360*. At a first glance, this meant a setback in the process, since we had to run the simulations already made all over again, to assure that all the simulations were performed in the same conditions. *Autodesk Insight 360* appears to be more user-friendly, yet is still incomplete, which provoked the need for some changes in our simulation process. Two of the settings needed for the simulation are the location and the type of use of the building. We had to change the location of our building, since the weather information on Beirut was not yet available at the time of this research. We also had to change the usage of the building, from a multi-family type to a hotel. The multi-family type was still disabled, and we had to choose the type that most resembled that function from the ones already available. Even though this problem appears to be temporary, it depends on *Autodesk*'s efficiency in solving the situation. Skipping this step and running the simulation directly with the same analysis tool *Autodesk Insight 360* uses (*Energy Plus*) resolves this type of situations.

This direct connection from *Rosetta* to *Energy Plus* is currently being explored, thus eliminating the intermediate step currently necessary of reproducing the model and only afterwards running the simulation. If *Energy Plus* were available as a backend for *Rosetta*, we could run several simulations in an even shorter amount of time, and in the end generating the best performing model in a visualization tool, like *Autodesk Revit* (workflow illustrated in Figure 4.1). This also allows for a script that sends the necessary information the the backend, depending on the type of input that this last requires.

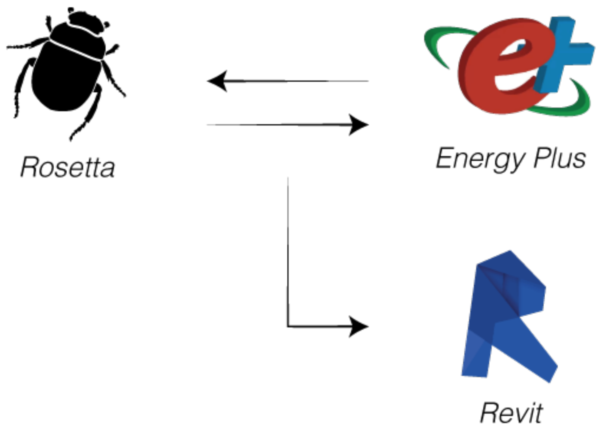


Figure 4.1: The workflow proposed for future work. A script is written, and with the use of *Rosetta*, the necessary information is sent to *Energy Plus*. The model is analyzed and the results sent back to *Rosetta*, allowing for automatic changes in the script according to the values given by *Energy Plus*. The loop ends when the result achieved reaches the specified goal, or after a number of analysis defined by the user. The final result is sent to *Autodesk Revit* for visualization.

By connecting *Rosetta* to *Energy Plus*, we also eliminate the human-error factor in the process of producing several building options from the initial script. The script itself will be able to change values according to the limits and parameters that the architect allows, running simulations and receiving results automatically.

An interesting concept to explore in this improved workflow will be Multi-Objective Optimization, allowing for the GA to improve several aspects of the building at once, instead of improving one at the time, as it was done in this research. By implementing this feature, the optimization process will produce better results and consume even less time to do it.

Bibliography

[1] A. Alfaris, “Emergence Through Conflict: The Multi-Disciplinary Design System (MDDS) (Ph.D. thesis),” 2009.

[2] N. D. Kaushika, K. S. Reddy, and K. Kaushik, “Energy Efficient Architecture,” in *Sustainable Energy and the Environment: A Clean Technology Approach*, 2016, pp. 153–172.

[3] H. Staff, “Water and Air Pollution,” 2009. [Online]. Available: http://www.history.com/topics/water-and-air-pollution

[4] *International Energy Charter*, 2015.

[5] D. A. Coley and S. Schukat, “Low-energy design: Combining computer-based optimisation and human judgement,” *Building and Environment*, vol. 37, no. 12, pp. 1241–1247, 2002.

[6] “International Energy Agency (IEA).” [Online]. Available: https://www.iea.org/topics/energyefficiency/

[7] (DOE) US Department of Energy, “A Common Definition for Zero Energy Buildings,” no. September, p. 22, 2015. [Online]. Available: http://energy.gov/sites/prod/files/2015/09/f26/bto_common_definition_zero_energy_buildings_093015.pdf%5Cnhttp://energy.gov/eere/buildings/downloads/common-definition-zero-energy-buildings

[8] “Energy Performance of Buildings Directive,” 2010.

[9] “Energy Efficiency Directive,” 2012.

[10] “Directive of the European Parliament and of the Council - Amending Directive 2010/31/EU on the Energy Performance of Buildings,” 2016.

[11] “Despacho n.º 15793-J/2013,” pp. 55–57, 2013.

[12] “Decreto-Lei n.º 118/2013,” pp. 4988–5005, 2013.

[13] EnergyPlus, "Getting Started with EnergyPlus," pp. 1–56, 2015. [Online]. Available: https://energyplus.net/sites/all/modules/custom/nrel_custom/pdfs/pdfs_v8.4.0/GettingStarted.pdf

[14] S. Feist, G. Barreto, B. Ferreira, and A. Leitão, "Portable Generative Design for Building Information Modelling," in *Living Systems and Micro-Utopias: Towards Continuous Designing, CAADRIA 21*, 2016, pp. 147–156.

[15] A. Golabchi and V. Kamat, "Evaluation of Industry Foundation Classes for Practical Building Information Modeling Interoperability," *International Association for Automation and Robotics in Construction*, 2013.

[16] A. W. Stocking, "Generative Design Is Changing the Face of Architecture," 2009. [Online]. Available: http://www.cadalyst.com/cad/building-design/generative-design-is-changing-face-architecture-12948

[17] H. J. Rogers, *Theory of Recursive Functions and Effective Computability*, 1987.

[18] M. El-Khaldi, "Mapping Boundaries of Generative Systems for Design Synthesis (M.Sc. thesis)," 2007.

[19] A. Leitão, L. Santos, and J. Lopes, "Programming Languages For Generative Design: A Comparative Study," *International Journal of Architectural Computing*, vol. 10, no. 1, pp. 139–162, 2012.

[20] R. Fernandes, "Generative Design : a new stage in the design process (M.Sc. thesis)," 2013.

[21] P. Szalapaj, *Contemporary Architecture and the Digital Design Process*, 2014.

[22] S. Petersen and S. Svendsen, "Method and simulation program informed decisions in the early stages of building design," *Energy and Buildings*, vol. 42, no. 7, pp. 1113–1119, 2010.

[23] G. Wurzer and B. Pak, "Lawnmower: Designing a web-based visual programming environment that generates code to help students learn textual programming," in *Digital Physicality — Physical Digitality, eCAADe 30*, vol. 1, 2012, pp. 655–664.

[24] A. Agkathidis, "Generative Design Methods," in *Real Time: Extending the Reach of Computation, eCAADe 33*, vol. 2, 2015, pp. 47–55.

[25] M.-A. Gallas, K. Jacquot, S. Jancart, and F. Delvaux, "Parametric Modeling: An Advanced Design Process for Architectural Education," in *Real Time: Extending the Reach of Computation, eCAADe 33*, vol. 2, 2015, pp. 149–157.

[26] M. P. Maldonado, “Digital Recipes: A diagrammatic approach to digital design methodologies in undergraduate architecture studios,” in *Fusion: Data integration at its best, eCAAde 32*, 2014, pp. 333 – 342.

[27] F. A. Cabecinhas, “A High-Level Pedagogical 3D Modeling Language and Framework (M.Sc. thesis),” 2010.

[28] A. Leitão and L. Santos, “Programming Languages for Generative Design: Visual or Textual?” in *Respecting Fragile Places, eCAADe 29*, 2011, pp. 549–557.

[29] D. Giordano and F. Maiorana, “Teaching algorithms: Visual language vs flowchart vs textual language,” *IEEE Global Engineering Education Conference, EDUCON*, pp. 499–504, 2015.

[30] A. Leitão and S. Proença, “On the Expressive Power of Programming Languages for Generative Design,” in *Fusion: Data integration at its best, eCAAde 32*, vol. 1, 2014, pp. 257–266.

[31] Herzog & de Meuron, “Beirut Terraces Brochure.”

[32] J. Lopes and A. Leitão, “Portable generative design for CAD applications,” *Integration Through Computation, ACADIA 31*, pp. 196–203, 2011.

[33] L. G. Caldas and L. K. Norford, “A design optimization tool based on a genetic algorithm,” *Automation in Construction*, vol. 11, no. 2, pp. 173–184, 2002.

[34] H. Chalabee, “Performance-based architectural design: Optimisation of building opening generation using generative algorithms (M.Sc. thesis),” 2013.

[35] M. R. Asl and S. Zarrinmehr, “Towards Bim-based Parametric Building Energy Performance Optimization,” in *Adaptive Architecture, Acadia*, 2013, pp. 101–108.

A

Case Studies I and II Technical Drawings

A.1 Case Study I - Family House in *Azenhas do Mar*

Figure A.1: Family House in *Azenhas do Mar* - Floor Plan.

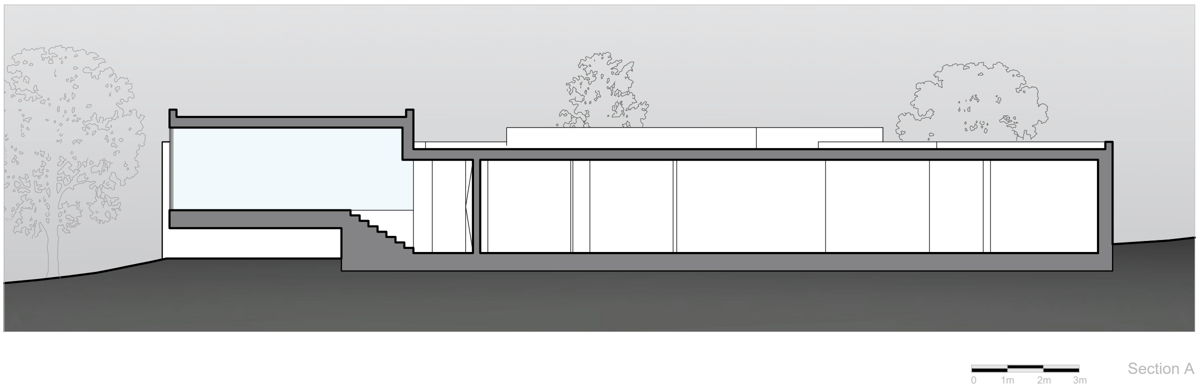


Figure A.2: Family House in *Azenhas do Mar* - Section A.

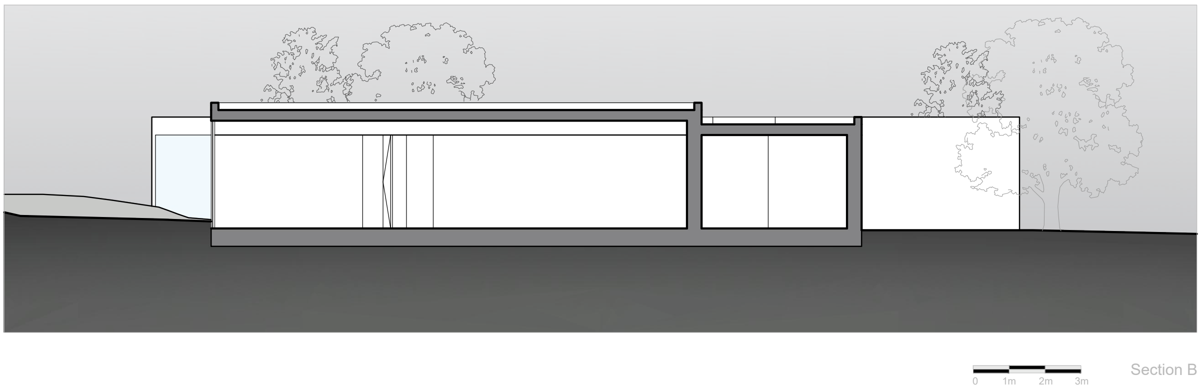


Figure A.3: Family House in *Azenhas do Mar* - Section B.

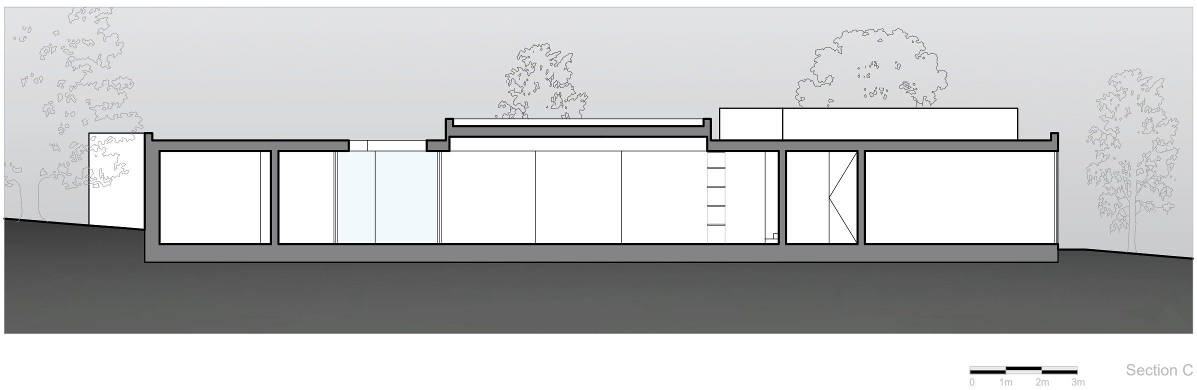


Figure A.4: Family House in *Azenhas do Mar* - Section C.

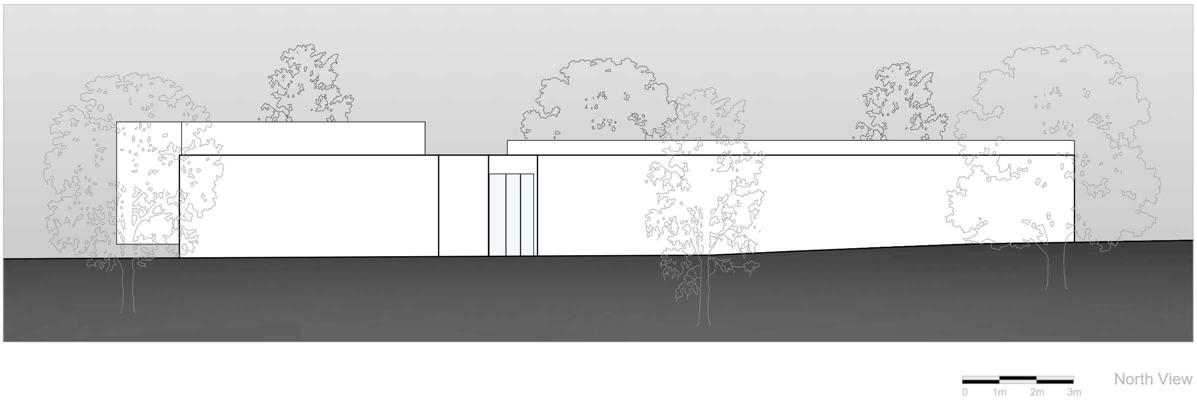


Figure A.5: Family House in *Azenhas do Mar* - North View.

Figure A.6: Family House in *Azenhas do Mar* - South View.

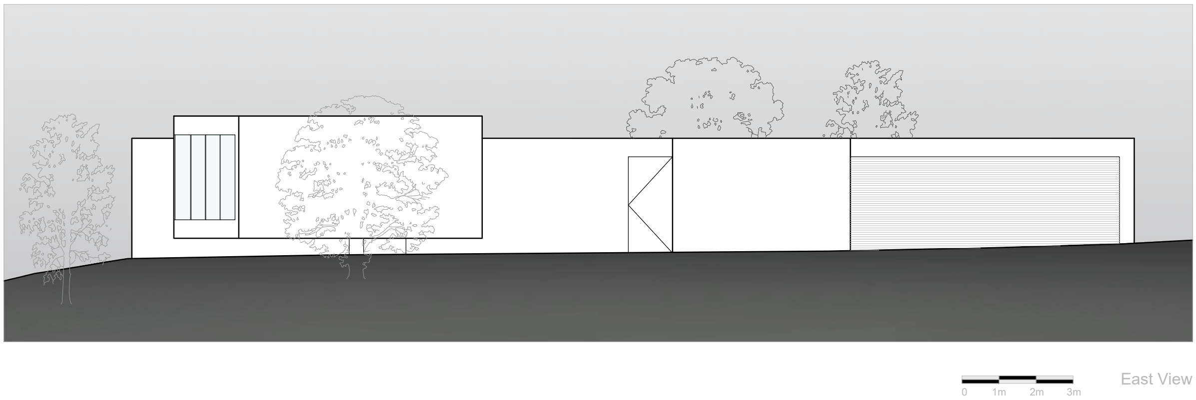


Figure A.7: Family House in *Azenhas do Mar* - East View.

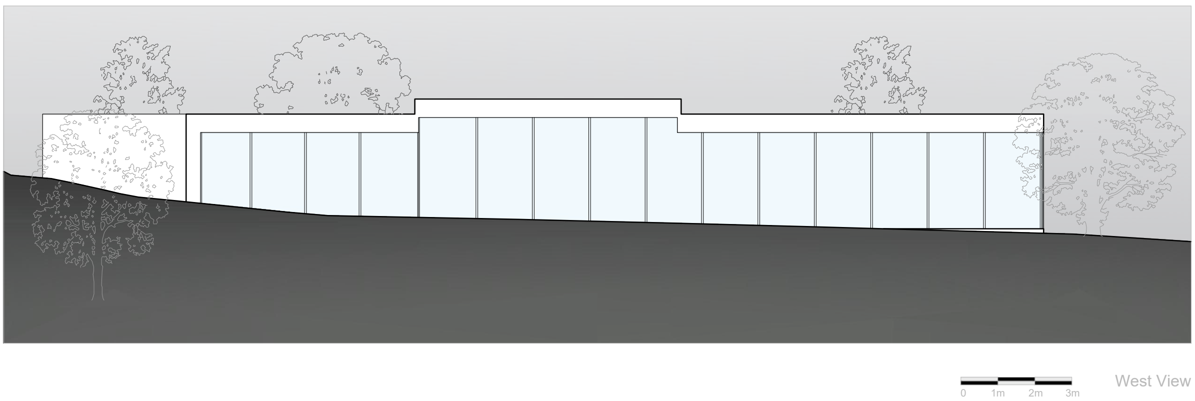


Figure A.8: Family House in *Azenhas do Mar* - West View.

A.2 Case Study II - Beirut Terraces

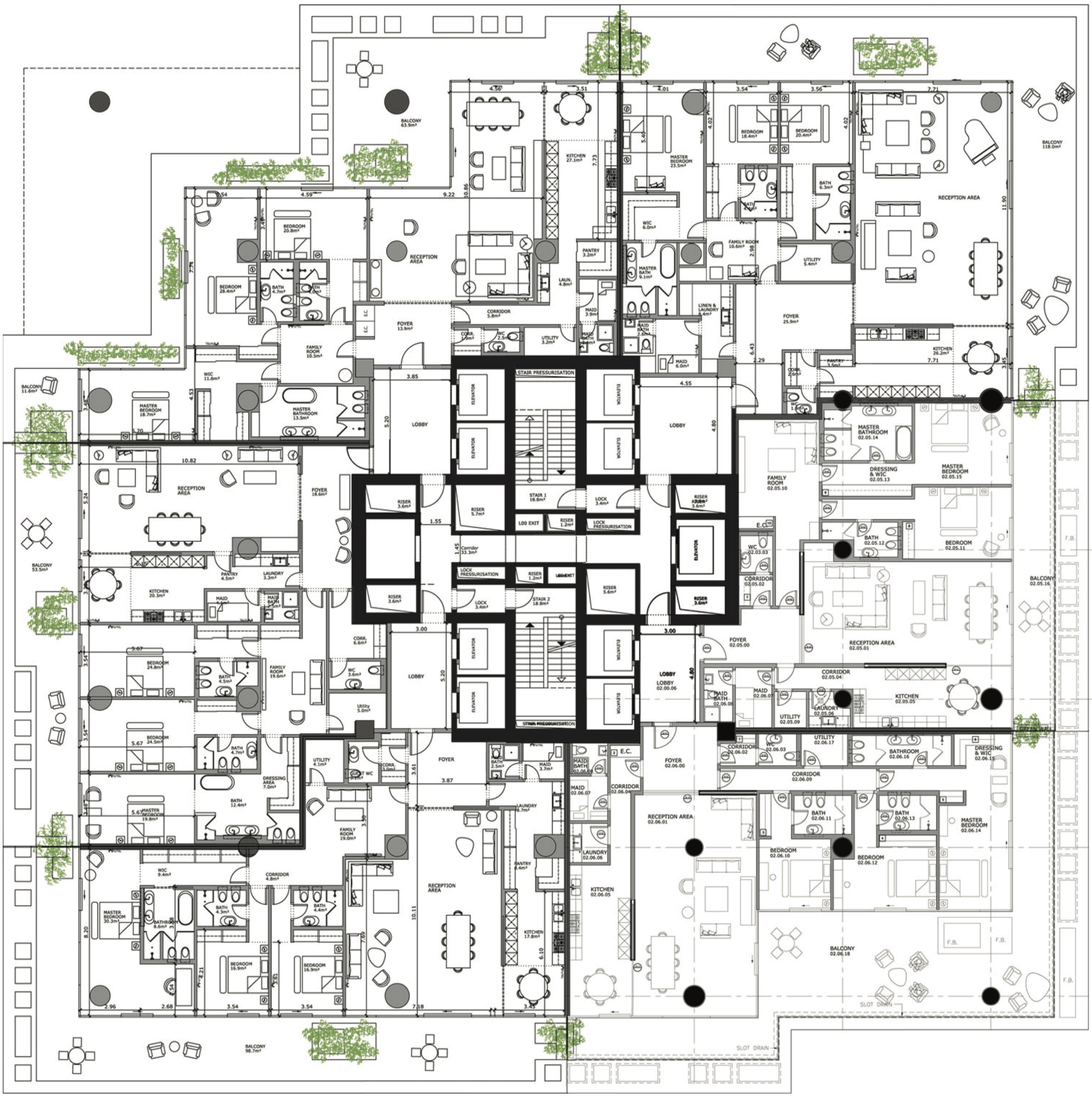


Figure A.9: Beirut Terraces - Level 2 plan - in *www.beirutterraces.com*

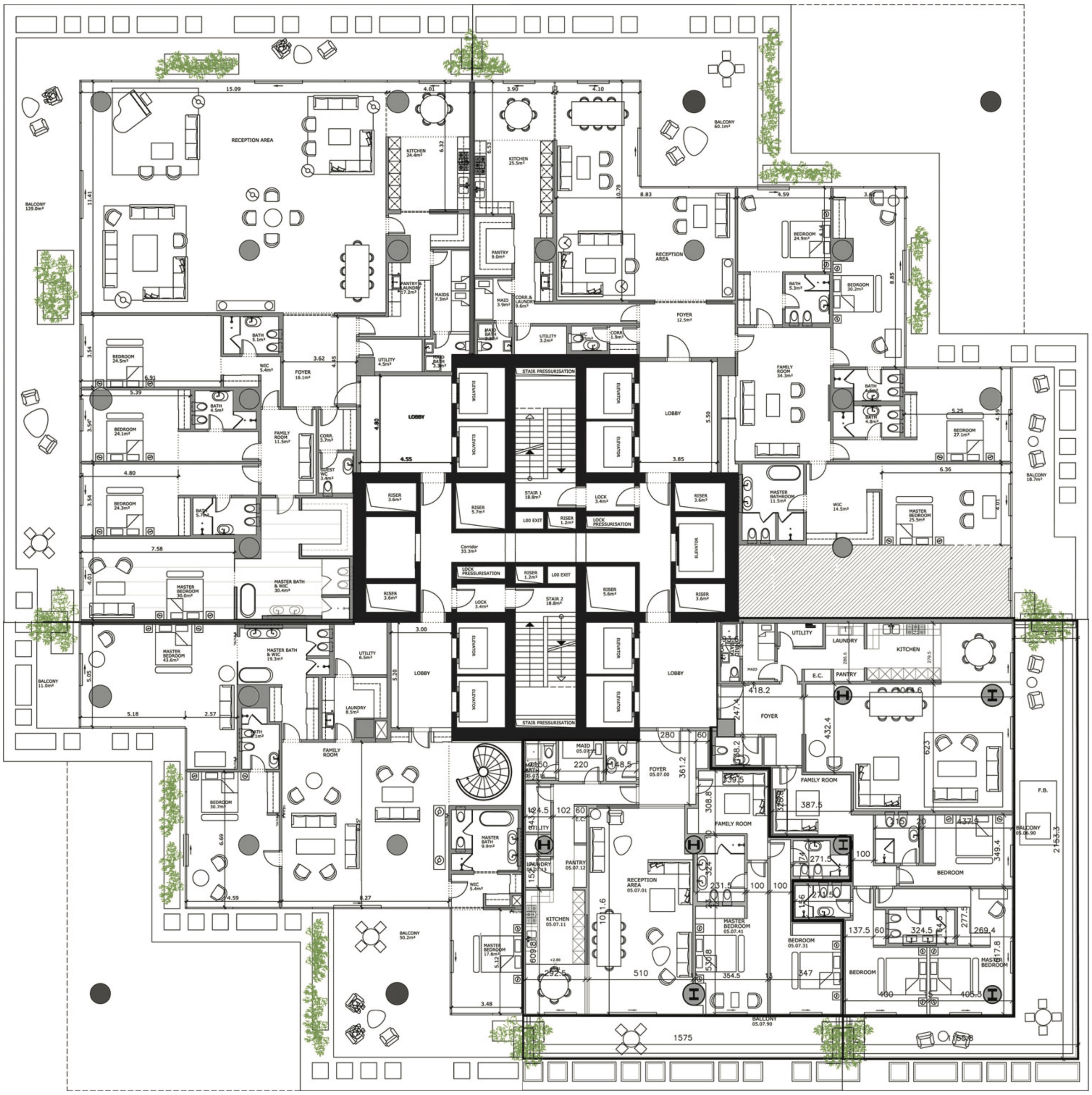


Figure A.10: Beirut Terraces - Level 5 plan - in *www.beirutterraces.com*

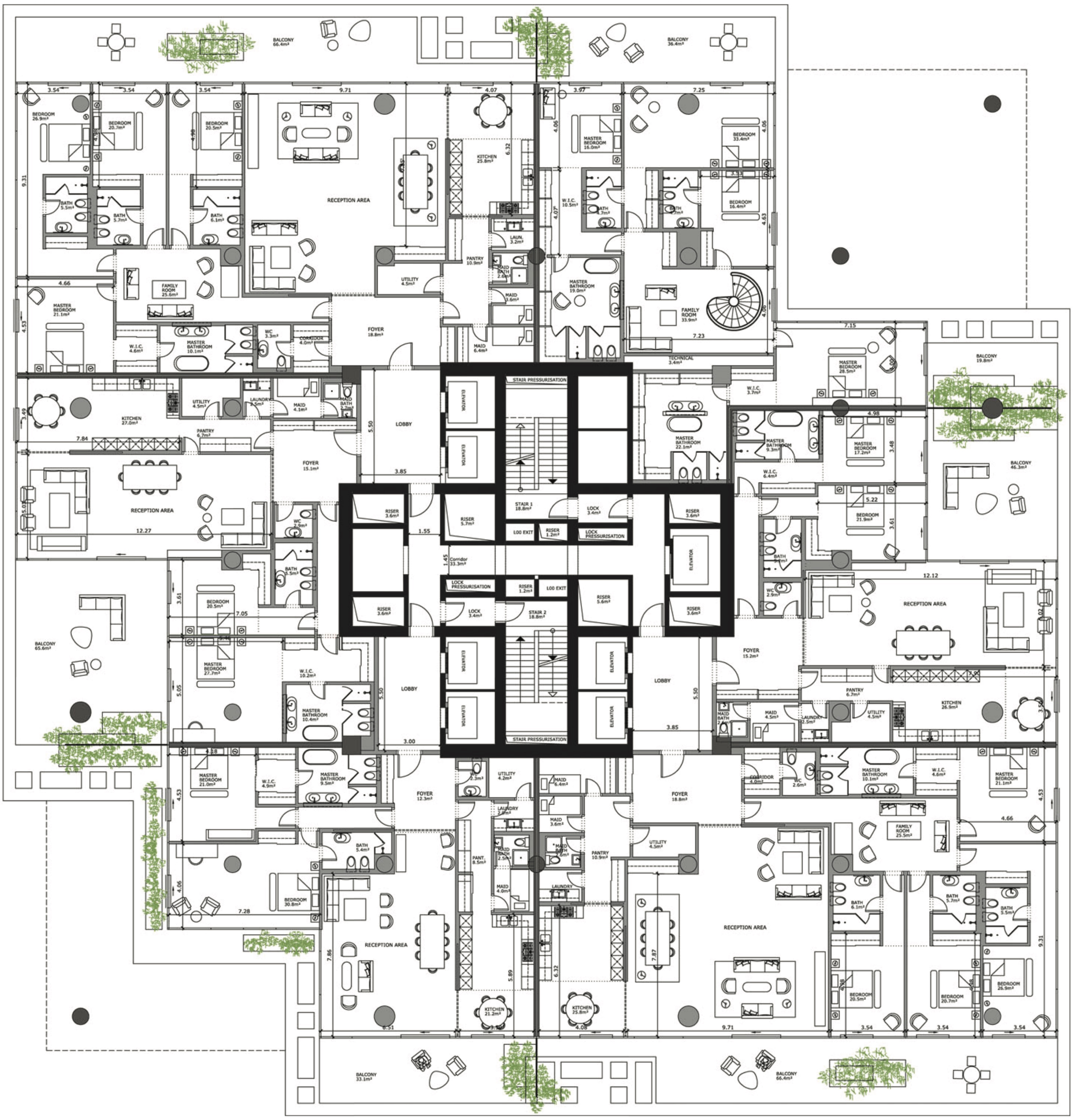


Figure A.11: Beirut Terraces - Level 11 plan - in *www.beirutterraces.com*

Figure A.12: Beirut Terraces - Level 12 plan - in *www.beirutterraces.com*

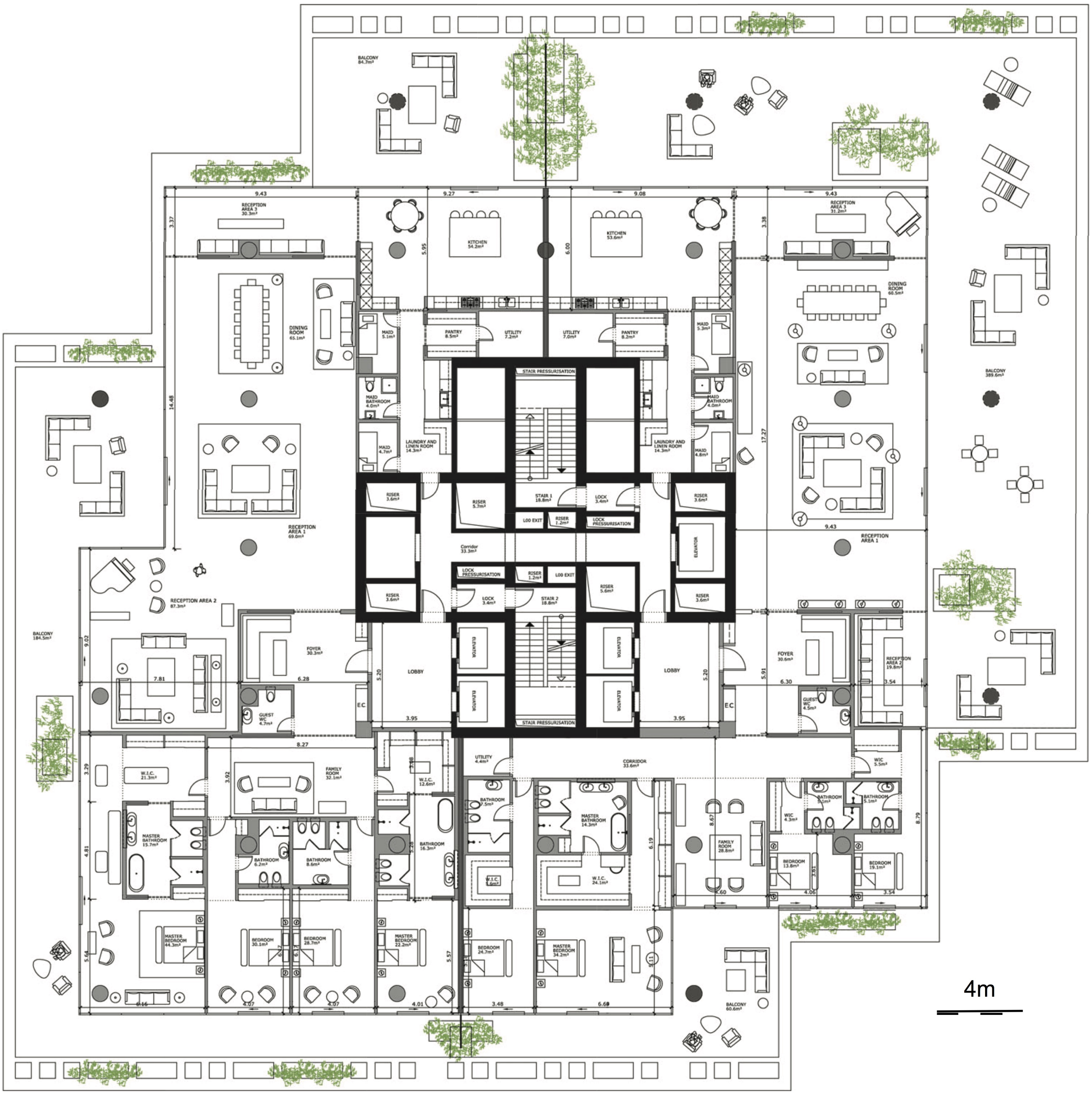


Figure A.13: Beirut Terraces - Level 25 plan - in *www.beirutterraces.com*