TECHNISCHE UNIVERSITÄT DRESDEN

Towards Parametric Design: Automation of Complex Geometrical Modelling

Master Thesis Report

submitted to the Faculty of Civil Engineering
of the Technische Universität Dresden
Institute of Construction Informatics

By

Amr Omara

Supervisor:	Prof. Dr.-Ing. habil. Karsten Menzel
Second examiner:	Prof. Dr.-Ing. Raimar J. Scherer
Tutor:	Prathap Valluru, M.Sc Dipl.-Ing. Johannes Frank Schüler
Employer's Liaison	Alexander Fritsch, B.Eng

Date of submission: 10.11.2019

TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Bauingenieurwesen Institut für Bauinformatik

Task sheet for the Master Thesis
(Aufgabenstellung für die Masterarbeit)

Name: Amr Omara Student No.: 4741437

Program: ACCESS

Title: **Towards Parametric Design: Automation of Complex Geometrical Modelling**

(Ein Beitrag zum Parametrischen Entwerfen
Automatisierung komplexer geometrischer Modellierungsprozesse)

Goal:

Knowledge-based engineering focuses on methods and tools for the support of product development. The application of parameterization for geometric modelling is well known and provides an important basis for knowledge-based design methods and tools. In order to make work routines both user-friendly and highly automatic, modern BIM-software offers the ability to control the internally used parameters externally.
Thus, parametric modelling of geometries can be extended to nD-Parametric Modelling, i.e. to additionally parametrize financial or scheduling data. An interface to a spreadsheet or text processor enables the bi-directional control of such parameters that drive the geometry within the BIM-model.

The main objective of this thesis is to develop algorithms to automate complex geometry models and compare them with alternative modeling approaches. To achieve this, the student is expected to work on an analysis of market requirements. Based on this the student is expected to develop a variable calculation model in the initial phase. In the second phase, the development of adaptive families for the geometrical models using BIM tools should be achieved. Furthermore, an integrated control of the geometrical model and the management models (cost, scheduling) should be achieved.

Finally, a comparative study should be carried out between different modeling methods with respect to the potential and feasibility for parametrization.

Scope of the work:

The following tasks shall be addressed in the Thesis:

1. Development of a variable calculation model using application tools according to market analysis requirements.

2. Creation of adaptive families for the geometrical model using BIM tools.

3. Interface development to enable geometry model automation using mathematical algorithms and integration in an existing design and procurement process.

4. Comparative study with alternative modeling methods in terms of planning costs and determination of user satisfaction with the proposed method.

It is strongly recommended to use the following tools for the task:

- Revit
- Dynamo
- Excel

Supervisors:	Prathap Valluru, M.Sc.
Employer's Liaison:	Alexander Fritsch, B.-ENG.
Responsible permanent academic staff:	Prof. Dr.-Ing. habil. Karsten Menzel
Second examiner:	Prof. Dr.-Ing. Raimar J. Scherer
Topic handed over to student:	GRP:RJS Digital unterschrieben von GRP:RJS DN: c=DE, o=Technische Universitaet Dresden, ou=CIB, cn=GRP:RJS Datum: 2020.04.03 16:07:04 +02'00'
Expected date for submission:	

Karsten Menzel Digital unterschrieben von Karsten Menzel Datum: 2020.04.03 15:18:42 +02'00'

Prof. Dr.-Ing. habil. Karsten Menzel

Head, Institut für Bauinformatik

DRESDEN concept Exzellenz aus Wissenschaft und Kultur

Declaration of originality

I confirm that the work contained in this report has been done by me under the guidance of my supervisor, and it has not been submitted to any other institute for any degree. For all the materials (data, theoretical analysis, figures, and text) used in this report from other sources, due credit have been given to the respective authors by citing them in the text and giving their details in the references.

Date, Place

Signature

Acknowledgements

First and foremost, I would like to show my sincerest gratitude to Prof. Dr.-Ing. habil. Karsten Menzel for offering me such an interesting and beneficial theises topic. Secondly, I would also like express my sincere gratitude to Prof. Dr.-Ing. Raimar J. Scherer. Special thanks and appreciation to my supervisors Msc. Prathap Valluru and Msc.Johannes Frank Schüler for thier great efforts in assisting and evaluating me throughout this work.

My kindeest words of sincere gratefulness are addressed to B.Eng.Alexander Fritsch without his continues support, motivation and immense knowledge in every step of the project, it would not have been accomplished. Special words of appreciation goes to the head of planning team Parkhäuser Dipl.-Ing.Heiko Frey in company Goldbeck for his kindness and support and to all of my colleges especially B.Eng.Imre Kiss.

Finally, but most importantly, I address my words of love to my family, my supportive Wife Basma and my little son Abdulrahman for their continuous love and encouragement. This work is dedicated to them.

Abstract

Parametric modeling and Building Information Modeling (BIM) are modeling concepts that have recently become very attractive. Creating complex models based on BIM method can be very time consuming, and inaccurate due to human mistakes. That is why implementing a parametric modeling concept into the BIM process would bring remarkable advantages. Parametric modeling is a modeling concept with the potential to alter the geometry of the model when the dimensions or other input value is varied. The implementation of the parametric modeling into the BIM workflow can have different types and forms. These types could be e.g. the traditional parametric modeling using the BIM systems[1] only or it could be procedural parametric modeling with combining the BIM systems with a graph-based system[2] or/and any other systems. This study tries to investigate the optimum workflow in terms of the performance and the usability of the user to model such a complex geometry like a Helical ramp. This workflow will be compared with another workflow in a comparative study with respect to the feasibility of parametrization.

[1] In this study Autodesk revit is chosin as a Main BIM system
[2] Dynamo is the graph-based system used in this study

Kurzfassung

Parametrische Modellierung und Building Information Modeling (BIM) sind Modellierungskonzepte, die in letzter Zeit sehr attraktiv geworden sind. Die Erstellung komplexer Modelle auf der Grundlage der BIM-Methode kann sehr zeitaufwendig und aufgrund menschlicher Fehler ungenau sein. Aus diesem Grund würde die Implementierung eines parametrischen Modellierungskonzepts in den BIM-Prozess bemerkenswerte Vorteile bringen. Die parametrische Modellierung ist ein Modellierungskonzept mit dem Potenzial, die Geometrie des Modells zu ändern, wenn die Abmessungen oder andere Eingabewerte verändert werden. Die Implementierung der parametrischen Modellierung in den BIM-Workflow kann verschiedene Arten und Formen haben. Diese Typen könnten z.B. die traditionelle parametrische Modellierung sein, bei der nur die BIM-Systeme verwendet werden, oder es könnte sich um eine prozedurale parametrische Modellierung handeln, bei der die BIM-Systeme mit einem graphenbasierten System oder/und beliebigen anderen Systemen kombiniert werden. In dieser Studie wird versucht, den optimalen Workflow in Bezug auf Performance und Benutzerfreundlichkeit bei der Modellierung einer so komplexen Geometrie wie einer Wendelrampe zu untersuchen. Dieser Workflow wird in einer Vergleichsstudie mit einem anderen Workflow im Hinblick auf die Durchführbarkeit der Parametrisierung verglichen.

Table of content

List of Figures

List of Tables

1. Introduction

1.1. Motivation and problem definition

Nowadays, as a result of the huge development of the construction industry, the demand for complex geometries in the construction field has raised (Sala , 2004). One of these complex geometries is the helical ramp, which is the main focus of this study. The need for helical ramps is increasing these days for msny reasons. Firstly it is the best solution to ease the traffic in the large park buildings, especially the park buildings that have high traffic like in the shopping mall or airports. According to (Alexander, 1988) the transportation ramps has to be moved out of the parking building when the number of parking places exceeds a certain number. Otherwise, the traffic will be overloaded and insufficient and can have very bad consequences for the parking structure during its service life. Besides, the poor traffic of the parking structure can lead to negative economic impacts on the associated facility, e.g. (airport or shopping center).

Nevertheless, the correct modeling of such complex geometries is challenging in terms of accuracy and time of modeling due to their complex nature. Therefore BIM can play a role in such cases, which BIM is an approach, that is distinguished by the generation and usage of consistent and reliable information about a building project. The quality of the information is the key feature of BIM and its process. The more reliable this information is, the better the quality of the BIM model (Autodesk). That is why in the case of a complex structure like the helical ramp, the advantages are not only the faster modeling process which saves time and cost for the projects but also avoids mistakes that can cost a lot to solve during the production, execution, and service phases. In the case of such a structure, the accuracy tolerance has to be very limited (Alexander, 1988) and some small errors can lead to huge difficulties during the construction.

State-of-the-art BIM software, such as Autodesk Revit, ArchiCAD or AecoSIM, provides a large number of default parametric dependencies, such as dependencies of the columns heights and floor types, and the position of foundations related to the floor level, etc. (Autodesk).. The use of this default parameterization enables easy modifications of BIM models. Furthermore, it becomes possible to modify the description of the model (or of the construction plans or schedules, respectively). BIM software can also be used to create individual families to create a specific model element with special characteristic that not available in the software libraries and to develop individual data structures. However, because of the complexity of creating such families and structure data especially in case of complex geometries using BIM softwares requires expert knowledge and is thus limited to BIM managers or to a few BIM enthusiasts (Ignatova, et al.). For that reason and to parameterize the generation of a complex geometry and to make it easy for the average user, we need more than just the normal use of this software, we need a workflow or a method that include some tools that are executed in a certain order. These tools could be just some pre-parameterised families or they could be combined with some other tools such as graph-based tools (visual programming methods), calculation tools etc.

thus the need for an easy, accurate, and fast BIM workflow for such geometry is very high. As aforementioned, this can have a lot of advantages not only by saving time during the design phase but also saves costs during the production, execution and even has a remarkable economical impact on the associated facility.

In this study the way of parametrize the generation of a complex geometry is studied(the Helical ramp). Two BIM workflows are used in this study with different approaches.

The first workflow used only some pre-parameterized families with huge number of parametric dependencies. These families are designed to get some required inputs from the user and be placed in a specific order and a specific way in order to achieve the correct modelling of the helical ramp. This workflow is implemented by the company Goldbeck for modelling the Helical ramp.

The second workflow is a workflow developed by the author of this study. In this workflow the author coupled some tools together with pre-parameterised families. These tools consist on the one hand of a graph-based system tool, which makes the placement of the elements faster and more precise. On the other hand, a calculation tool that reduces the calculation effort of BIM systems and makes the input interfaces more user-friendly.

1.2. Aim of the study

As we can see in the well-known graph Fig(1), the BIM can tremendously reduce the work during the documentation and the coordination phases compared to the traditional CAD-based design (Fridrich, et al., 2014). Is this valid also in the case of the complex geometries? Or the traditional BIM process is not enough alone?.

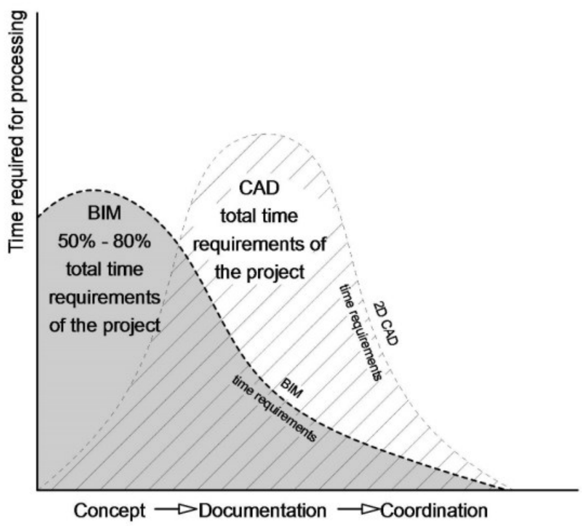


Figure 1: Comparison of time requirements of work in CAD vs. BIM (Fridrich, et al., 2014)

The current state of BIM software is too simplistic and only supported a limited complexity in geometry modeling (COENDERS, 2010). According to T. Michael (Michael, 2016) The lack of the availability of computational and automation approaches in the modeling process can lead to information inaccuracy or information loss. Contrarily BIM workflows that depend on parametric modeling can deliver building information that is more coordinated, more reliable, higher quality, and internally more consistent. However, BIM systems are limited in their ability to automate the generation of geometry, especially for complex geometries (JANSSEN, 2015). By implementing parametric modeling methods into the BIM process we can parameterize the generation of the BIM models and ensure the quality and the accuracy of the output. As mentioned above, such automated or semi-automated generation of BIM models can provide more reliable and higher quality information, and offers several other advantages, such as faster, less error-prone, and a greater variety of model shapes (Fridrich, et al., 2014). The BIM process with parametric modeling can be designed to generate models at different scales, starting from a single building element, a part of the building, or even the entire building.

Since there are many ways to design a BIM workflow, the main aim of this thesis is to investigate what can be nowadays attained through Modeling automation, both by literature research and by proposing a novel concept for the automation of the generation of complex geometries. The author introduces an automated workflow for the generation of BIM geometry and documentation. This study argues that the parametric modeling phase may take some additional time at the beginning (the hatched part in Figure 2), but it can save more time in the following phases.

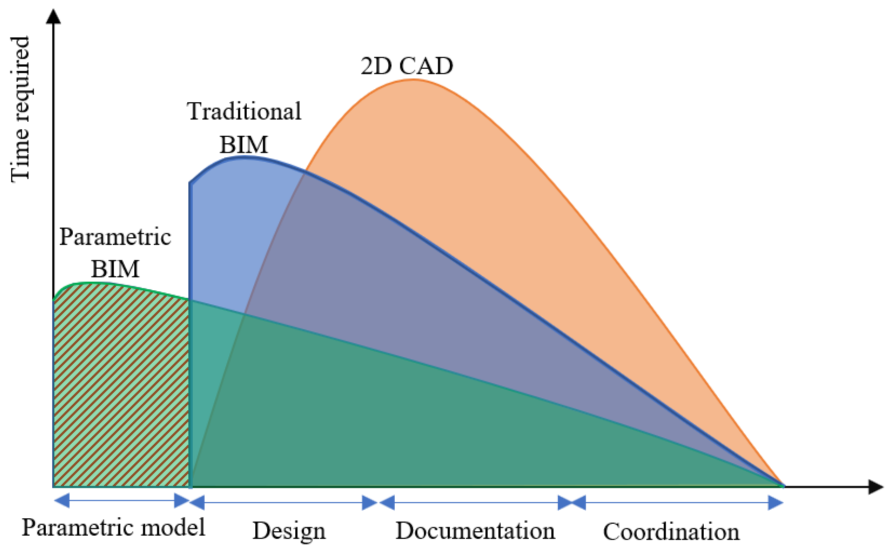


Figure 2: Comparison of time requirements of work in CAD vs. Traditional BIM VS. Parametric BIM

1.3. Thesis outline

Chapter 1 provides an overview of the study goals and motivation of developing a BIM parametric workflow for the generation of complex geometries especially the helical ramp. It also discusses the aim of the study.

Chapter 2 gives firstly a basic explanation of the BIM concept. Then the concept of parametric modeling and its different types, which can be found in the literature, are explained. Then the possibility of implementing the parametric modeling concept into the BIM workflow is discussed. The last part of this chapter discusses a case study of developing a BIM workflow for the generation of the helical ramp, which was developed by the company Goldbeck.

Chapter 3 discusses at the first part the geometrical principles of the helical ramp, the boundary conditions that need to be met, and the output required from the model. The second part of this chapter explains the proposed BIM workflow and the methodology used in developing it.

Chapter 4 presents in detail the implantation of the methodology discussed in chapter 3. This chapter explains also the concepts and the structures of the developed algorithms and interfaces and how they interact and combined to form the proposed workflow.

Chapter 5 compares mainly between the two BIM workflows introduced in this study in terms of performance and usability.

Chapter 6 presents conclusions about the main observations and results addressed in this study.

2. Literature review and case study

2.1. BIM

BIM is an abbreviation of the term "Building information modeling". According to the US National Building Information Model Standard Project Committee, "Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. BIM is a shared knowledge resource for information about a facility forming"

BIM as a concept started in the 1970s. the first time the term building information modeling was officially published including the abbreviation "BIM" (Nederveenab, et al., 1992). In 2002, Autodesk and other software vendors have started to get involved in the BIM field.

BIM is sometimes mistakenly thought of as just software. However, BIM must not only be seen as a software but as an embedded process. This process starts with the creation of a smart 3D model with all associated information, which enables documentation, coordination during the design, construction, and the building life cycle (Autodesk, 2020). It should be noted that BIM is a package of data that can contain all relevant information (Fridrich, et al., 2014). The BIM method is distinguished by its high-quality, reliable, and fully coordinated information (Abedin, 2016).

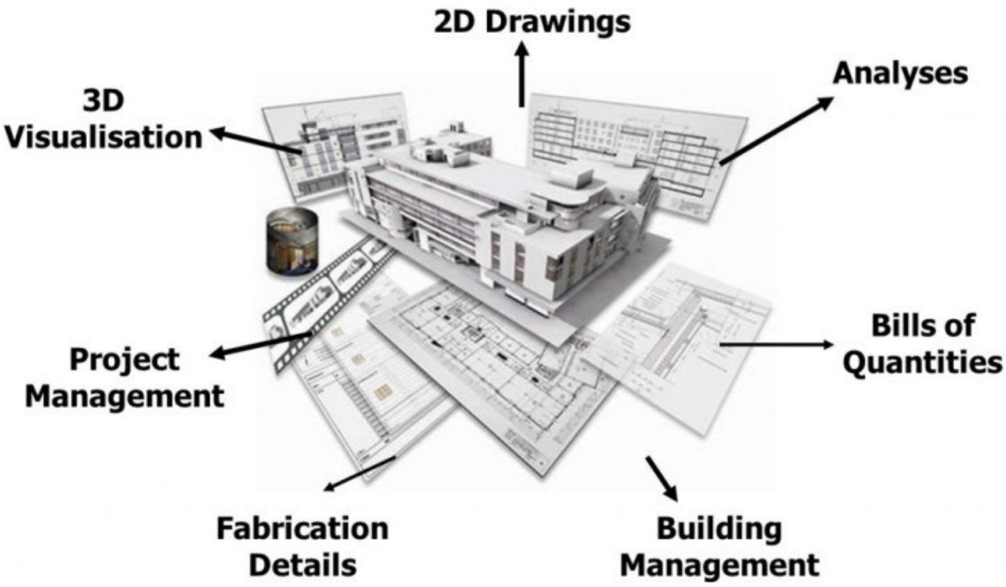


Figure 3: Application of BIM **(Maia, et al.)**

According to construction innovation, (Innovation, 2007) there are many benefits of BIM e.g. faster, more effective processes, better design, controlled whole life costs, and better production quality. BIM can also offer the design team a high degree of confidence and minimize conflicts (Building infromation modeling in the architecture-engineering construction project in Surabya, 2016).

2.2. Parametric modeling

Parametric modeling is a process that depending on the relations between the different attributes of the geometry to change its shape. These relations can be performed through a number of equations, constants, or variables. In which, one attribute has changed the effects of these changes on the other attributes are applied automatically with no need for any manual calculations.

This concept is similar to adding a series of interconnected equations in Excel or any other spreadsheet if one value is changed all other values are automatically changed. This can be also applied with the modeling of geometry by creating a chain of parameters (Geometrical or Mathematical parameters) that form the geometry. Whereas, by manipulating any of these parameters, all of the other parameters are adjusted and the shape of the geometry accordingly changed.

To illustrate this concept, consider a very simplified example of parametric modeling. An example of modeling a single beam where the profile height is related to its length. Whereas, when the length is changed the profile hight is automatically changed. The mathematical relations could be as following :

if L > 2 m then h = 300 mm,

else if L > 2.5 m then h = 320 mm,

else then h = 350 mm.

Figure 4: Dimension parameter of the beam Example.

This very simple example can be done inside the BIM system e.g. Autodesk Revit. By defining two parameters for the length and height of the beam and using the above-mentioned logic Fig(4). More complex dependencies could be performed in order to create more complex geometries.

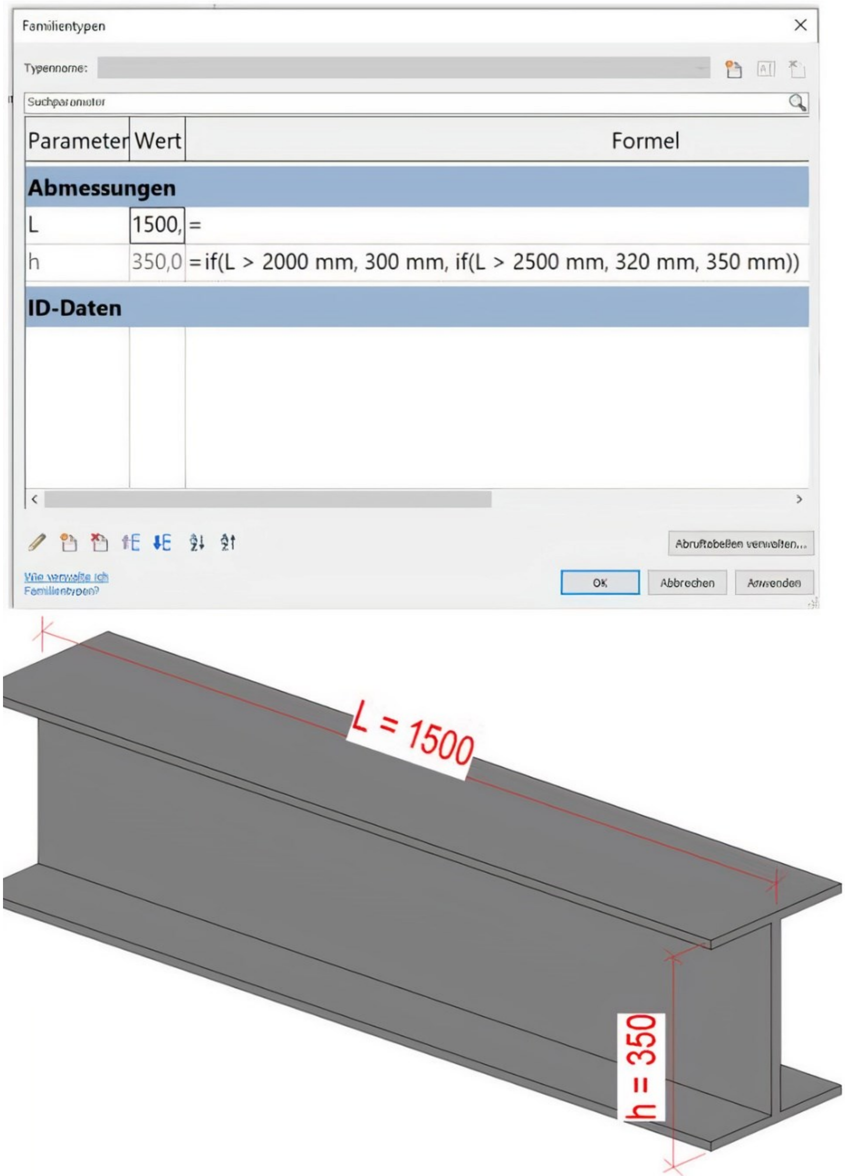


Figure 5: Defining parameters for a beam geometry in Autodesk Revit.

However, this kind of parametric modeling used only a single-operation iteration. This means the changes are done only in one element or several identical elements per iteration. To change several unidentical elements with different or same parameter values simultaneously, we need to develop a multi-operation iteration. These methods are described in detail below.

In the literature, there are different types of parametric modeling. Parametric modeling is classified based on the way they support iteration. This taxonomy allows us to clearly classify the parametric modeling methods and the systems that support these methods. According to (JANSSEN, et al., 2015) the parametric modeling methods are classified into four types:

- Object modeling
- Associative modeling

- Dataflow modeling
- Procedural modeling

The object modeling does not support any iteration. Associative modeling allows for only single-operation iteration, dataflow support implicit multi-operation iteration, and procedural modeling allows for explicit multi-operation iteration.

Most of the BIM systems available in the market support only either Associative modeling or object modeling. The Associative modeling allows only a single operation iteration, which means, the system applies only one operation simultaneously to some geometries and with the same value. For example, if we have a number of plates and the operation is "changing the thickness of these plates". The single operation iteration means, that the new input value (The thickness of the plate) will be the same for all plates. This can be done using Autodesk Revit by creating the parameters as been done in the above example Fig(5), but with creating the parameters as a Type parameter. So, when changing the parameter once this will be applied simultaneously to all plates with the same type in the project. Autodesk Revit also supports the Object modeling type but in this case, the parameters have to be created as an Instance parameter. With the Instance parameter, the changes will be applied only on the single element, which means there is no iteration.

The multi-operation iteration means to give multiple input values for the geometries. For example, if we have a number of plates, that we need to change their thicknesses with different thicknesses values. The new thicknesses values are nested in a list and the plates are also nested in another list. The algorithm iterate over both lists and gives each plate the new thickness value Fig(6). However, in the case of the implicit multi-operation iteration, the user has to ensure that the list of the plates and the list of the thicknesses are correctly ordered, in which each plate gets the corresponding value.

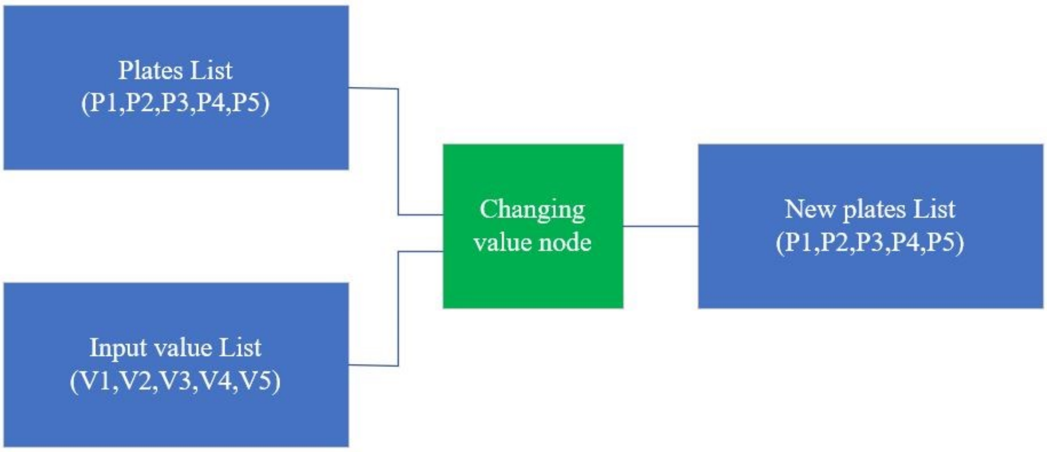


Figure 6: Implicit Multi-operation concept

On the other hand, in the case of the explicit multi-operation iteration, there is an additional matching algorithm Fig(7). This algorithm is responsible for reordering the lists, in which each plate gets the desired value.

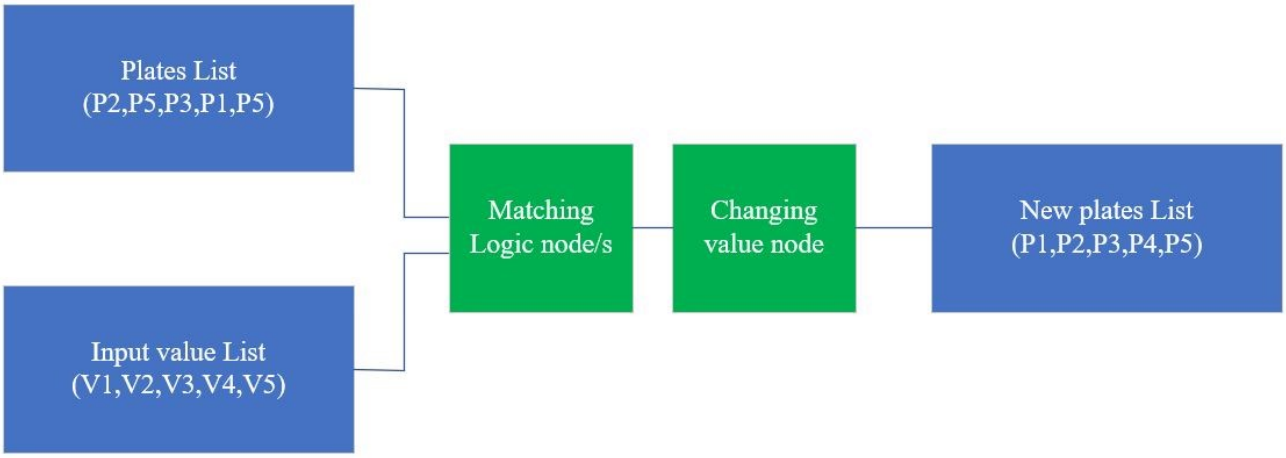


Figure 7: Explicit Multi-operation concept

By default, the BIM system can not support the multi-operation iterations. In order to achieve one of the multi-operation iteration systems, we need to enhance the BIM system e.g. by implementing a graph-based system to the workflow. Autodesk dynamo[1] supports the explicit multi-operation iterations, by adding some logic nodes to match the lists.

2.3. BIM workflow and parametric modeling

In order to create a powerful and efficient parametric BIM workflow in case of complex geometries, the Dataflow or procedural modeling approaches should be applied. This needs to enhance the BIM system to implement multi-operation iteration i.e implicit or explicit multi-operation iteration. According to (JANSSEN, 2015) there are two approaches, the embedded approach or the coupled approach.

The embedded approach: in this approach, the BIM system is extended by adding some more rules and conditions to achieve the parameterization of the modeling.

The coupled approach: in this approach, a graph-based system is coupled with the BIM system. This allows the graph-based system to be used to generate elements and to manage the data in the BIM systems. According to (JANSSEN, 2015) there are two types of this approach, tightly coupled approach and loosely coupled approach see Fig(8). In the case of the tightly approach, the graph-based systems are connected with the BIM system through

[1] The graph-based system used in this study

the Application Programming Interface (API), the graph-based systems generate or modify the geometry directly when the script is executed.

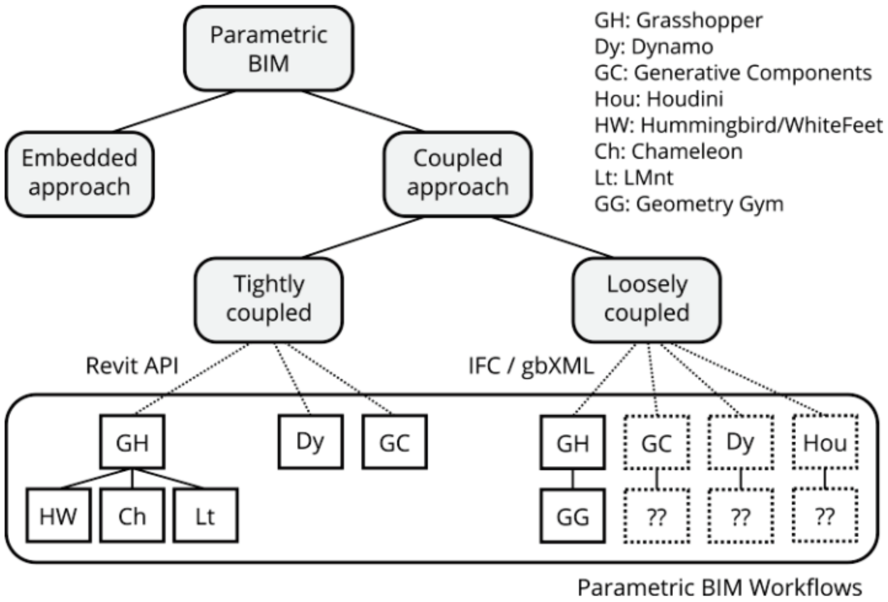


Figure 8: Approaches of implementing Parametric modeling into the BIM process (JANSSEN, 2015)

On the other hand, the loosely coupled approach used a completely different way, which depends on creating a new file to be imported into the original BIM system to create/modify the geometry. The generated file (cooked Model) is created using a graph-based system and then is used to create the model that to be imported in the BIM system (Exchange Model) it should be in a standard file format e.g. IFC or gbXML see Fig(9).

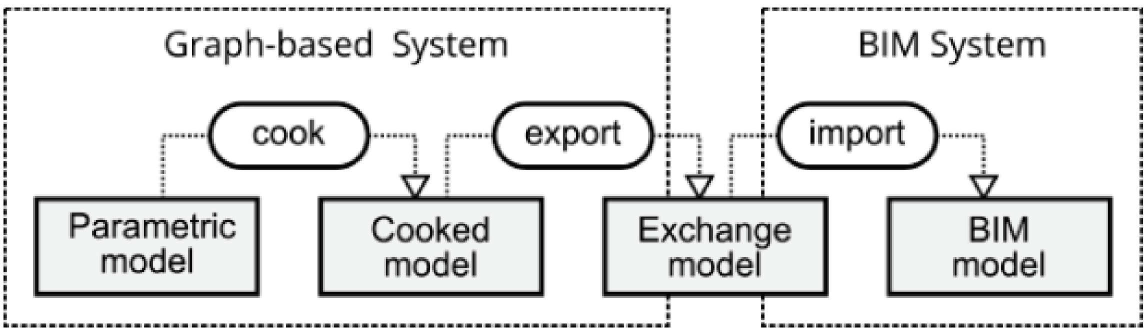


Figure 9: The workflow of the loosely coupled approach (JANSSEN, 2015).

The loosely coupled approach is recommended in the case of a Multi-disciplinary environment or with different BIM systems (Open BIM), which through using a standard file format the users are allowing to connect tools and systems supporting different forms of collaboration and exchange. One of the benefits of this approach, in the case of the Graph-based system and the BIM system, have not been compatible with each other, For example, using Autodesk-Dynamo as a Graph-based system to cook a file for Archicad as a BIM system. On the other hand, the tightly approach miss this Advantage, which the graph-based system and the BIM system have to be compatible to each other, However, the tightly approach is more dynamic and faster, regarding changing or repeating the process, which the parametric model is making the change directly on the BIM system without importing and exporting steps.

The two BIM workflows used in this study are using different approaches. The first workflow[1] used the embedded approach, by extended the Autodesk Revit system to be able to achieve the parameterization of the modeling. This Workflow is used now in Goldbeck to model the Helical ramps and it will be explained in detail in next section. The second Workflow[2], which is developed by the author of this study used a modified version of the tightly coupled approach, by coupled both Autodesk Dynamo as a graph-based system and Microsoft Excel as a computational system with Autodesk Revit as the main BIM system, to achieve explicit multi-operation iteration. This approach is discussed in depth in section 3.3.3. The Comparative study in Chapter 5 argues that using the proposed coupled approach is more effective in terms of performance and usability.

2.4. Market analysis and BIM

In this chapter we will make a market analysis for implemnting the neu BIM technologies in the construction industry in Germany especially the market of the building the garage parking buildings and the advantages of this implemntings.

2.4.1. Multi-storey car park market in germany (Parkhaus Market)

A multi-storey car park is usually a building with several storeys, with parking spaces for cars or motorbikes and, more rarely, for trucks or bicycles. Parking garages are usually built in larger cities to make better use of scarce inner-city space and to relieve the street space from parked cars so-called stationary traffic. One floor of a multi-storey car park is called a parking deck. A high garage is a garage that is accessible via ramps and is not at ground level, an underground garage is a garage below ground level.

In principle, a difference must be made according to the type of development:

- Via ramps, where the drivers themselves drive their vehicles to the car park and pick them up there again. There are different types of ramps A space-saving variant is known as split-level system, full storey ramp or Helical ramp. The last variant is usually used in case of high traffic parks (Alexander, 1988)

[1] This workflow will be named in this study as the existing method
[2] This workflow will be named in this study as the propsed method

- Via lifts where the drivers hand in their vehicles and keys. This technology is not widely used in Germany (Hasse, 2015).

The first car parks were built in Germany in the 1920s. However, parking garages only became really popular in Germany after the Second World War, when mass mobilisation began here too. The five-storey and first public car park Hauptwache in Frankfurt am Main was built in 1956 as a consequence of the increasing traffic density in Frankfurt city centre (Kleinmanns, 2011).

In view of the currently growing number of cars and the scarcity of space, building a new multi story parking are becoming increasingly important, especially in the city centres. While at the end of the 1960s around 260 car parks throughout Germany offered space for 90,000 cars (Hupfer 2011). Market observers estimate that 1.1 million of a total of around 4.8 million parking spaces in Germany are on public roads and traffic routes. The management of these designated parking spaces (on-street market) is reserved for local authorities by law. The remaining 3.7 million parking spaces are allocated to the so-called off-street segment, i.e. parking on non-public roads and traffic routes. Of these, around 70% are subject to charges (e.g. multi-storey car park, underground car parks, unrestricted market square with parking machines) (Fokus).

2.4.2. BIM in multi-storey car park market

There are various types of building systems of the multi srtory garages and the helical ramps in germany. Most of the companies rielied on the concrete cast is situ way of building such building (Hasse, 2015). Some companies use the Hoesch Additiv system, which is suitable for use as floor decking system in multi storey buildings and car parks. However, only the system used by goldbeck is discussed in this section, which is relied on precast concrete plates with a steel skleton as explained in section 3.1.

Using this system has advantages not only in terms of fast production and construction, but it opens also the door for implementing parameterization concepts in terms of of the design and planning. By using such a system, the implementing of the parametric models are possible through making a pre-parametrized BIM families and using the power of the graph based BIM system to model them prescisly and fast in addition the ability to modifying the design and the modelling easily using the BIM parametric tools. However with the traditional cast in situ systems each individual Helical ramp has to be modelled from the beginning using the normal drafting methods.

The company Goldbeck produced approximately 100 Parkhause yearly with value from 350 Mio. Euro to 480 Mio. Euro. Where the value of the mulri-story garages built by the company in the year 2017/2018 was 311Mio euro in German-speaking countries and almost 50 Mio. Euro in other European countries. This number is increased to reach 480 Mio. Euro in the year of 2019/2020 with almost 450 Mio. Euro only in Germany.

On average, the company builds 3 to 5 out of every 100 parking garages with at least 2 helical ramps. With the traditional methods of cad drafting, the completion of a spiral ramp normally took about 2 weeks. This time (2 weeks) is required when an experienced engineer is responsible for the task. However an average user were not eligible to deal with such a complex task. For that reason it is always worthwhile for the company to implement new modelling technologies to accelerate this process and increase the accuracy of the modelling. These implementations are also worthwhile as they could save time and cost of the modelling process in addition to the cost of corrections in manufacturing and assembly due to the lack of modelling accuracy.

In this study we discuss a case study (mentioned in section 2.5) of of implemintig such a parametric BIM technology to automate the generation of the helical ramp of the parking garages. In addition to this, the author introduce a new concept of the parameteric generation of the helical ramp. In chapter 5 the planinig costs of both methods are discussed.

2.5. Case study with existing methodology

The modeling of the helical ramp represents always as a complex and challenging topic when it comes to large multi-story parks. Therefore, Goldbeck has developed in the last 10 years some modeling workflows/methods to model the helical ramp. Firstly, the company relied on 2D modeling with AutoCAD and a simple Excel calculation spreadsheet to calculate the plate and beam coordinates. Thereafter, the company decided to take advantage of the benefits of 3D models and BIM. Therefore, a new tool was developed with Revit by the developers of the company. In this section, we will have a closer look at this method in detail. This will be used subsequently for the comparison with the purposed method developed by the author.

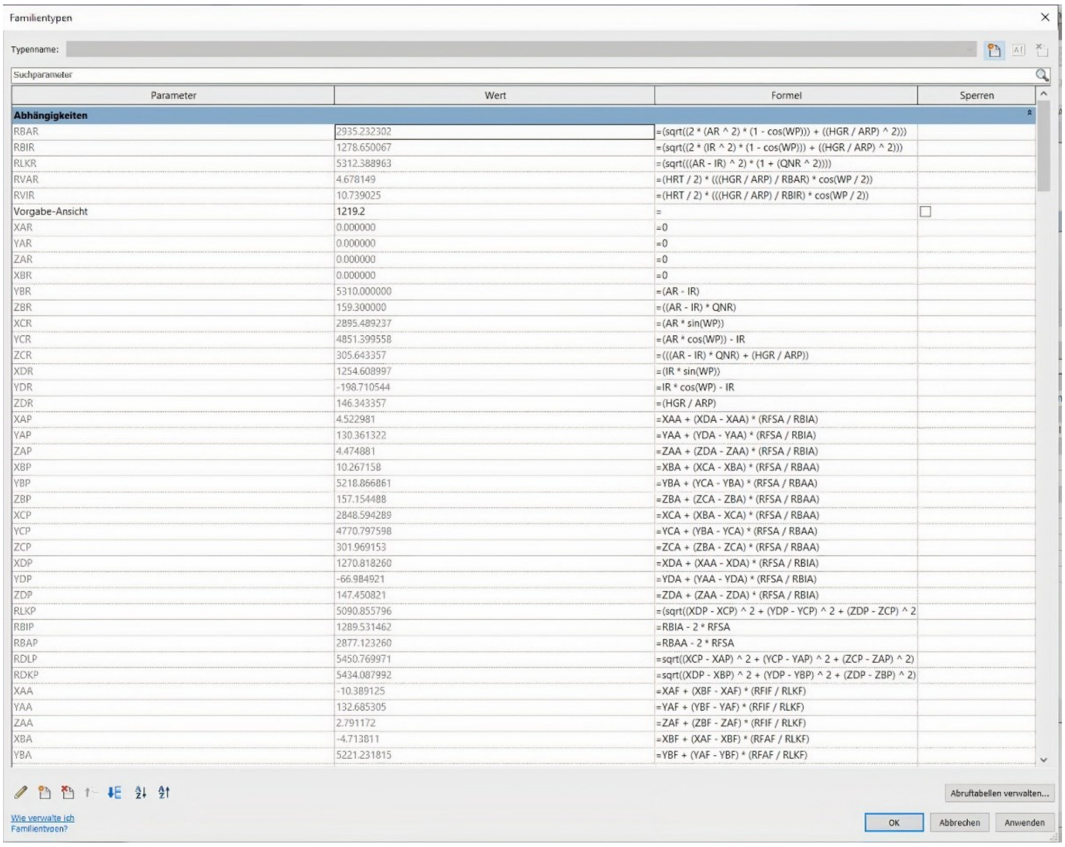
2.5.1. The concept of the method

As aforementioned the developer of this method decided to depend only on the BIM system without the assistance of any other systems. The parametric modeling concept of this method is to create a number of associative parametric modeling processes, in which the user performs them one after the other.

The associative modeling processes are single operation iteration processes, in which the user enters the input data of the elements one by one. Through these input changes, the geometries of the elements are automatically adjusted.

The developer created parametric constraint-based Revit families with a large number of formulas and constraints. Through these formulas, Revit runs all the trigonometric calculations in the background and then rounded up to the metric again (Revit, 2019). Thereby all the required elevations and dimensions of the ramp plates and beams are calculated Fig(10). Four different parameterized Revit families are developed (Installation surface family, Ramp plate family, Landing family, and beam family).

Because everything is done inside the BIM system, the user input interface has to be the normal BIM system user interface. This means the user has to search for the parameters that need to be changed.



Parameter	Wert	Formel	Sperren
RBAR	2935.232302	=(sqrt((2 * (AR ^ 2) * (1 - cos(WP))) + ((HGR / ARP) ^ 2)))	
RBIR	1278.650067	=(sqrt((2 * (IR ^ 2) * (1 - cos(WP))) + ((HGR / ARP) ^ 2)))	
RLKR	5312.388963	=(sqrt(((AR - IR) ^ 2) * (1 + (QNR ^ 2))))	
RVAR	4.678149	=(HRT / 2) * (((HGR / ARP) / RBAR) * cos(WP / 2))	
RVIR	10.739025	=(HRT / 2) * (((HGR / ARP) / RBIR) * cos(WP / 2))	
Vorgabe-Ansicht	1219.2	=	☐
XAR	0.000000	=0	
YAR	0.000000	=0	
ZAR	0.000000	=0	
XBR	0.000000	=0	
YBR	5310.000000	=(AR - IR)	
ZBR	159.300000	=((AR - IR) * QNR)	
XCR	2895.489237	=(AR * sin(WP))	
YCR	4851.399558	=(AR * cos(WP)) - IR	
ZCR	305.643357	=(((AR - IR) * QNR) + (HGR / ARP))	
XDR	1254.608997	=(IR * sin(WP))	
YDR	-198.710544	=IR * cos(WP) - IR	
ZDR	146.343357	=(HGR / ARP)	
XAP	4.522981	=XAA + (XDA - XAA) * (RFSA / RBIA)	
YAP	130.361322	=YAA + (YDA - YAA) * (RFSA / RBIA)	
ZAP	4.474881	=ZAA + (ZDA - ZAA) * (RFSA / RBIA)	
XBP	10.267158	=XBA + (XCA - XBA) * (RFSA / RBAA)	
YBP	5218.866861	=YBA + (YCA - YBA) * (RFSA / RBAA)	
ZBP	157.154488	=ZBA + (ZCA - ZBA) * (RFSA / RBAA)	
XCP	2848.594289	=XCA + (XBA - XCA) * (RFSA / RBAA)	
YCP	4770.797598	=YCA + (YBA - YCA) * (RFSA / RBAA)	
ZCP	301.969153	=ZCA + (ZBA - ZCA) * (RFSA / RBAA)	
XDP	1270.818260	=XDA + (XAA - XDA) * (RFSA / RBIA)	
YDP	-66.984921	=YDA + (YAA - YDA) * (RFSA / RBIA)	
ZDP	147.450821	=ZDA + (ZAA - ZDA) * (RFSA / RBIA)	
RLKP	5090.855796	=(sqrt((XDP - XCP) ^ 2 + (YDP - YCP) ^ 2 + (ZDP - ZCP) ^ 2	
RBIP	1289.531462	=RBIA - 2 * RFSA	
RBAP	2877.123260	=RBAA - 2 * RFSA	
RDLP	5450.769971	=sqrt((XCP - XAP) ^ 2 + (YCP - YAP) ^ 2 + (ZCP - ZAP) ^ 2)	
RDKP	5434.087992	=sqrt((XDP - XBP) ^ 2 + (YDP - YBP) ^ 2 + (ZDP - ZBP) ^ 2)	
XAA	-10.389125	=XAF + (XBF - XAF) * (RFIF / RLKF)	
YAA	132.685305	=YAF + (YBF - YAF) * (RFIF / RLKF)	
ZAA	2.791172	=ZAF + (ZBF - ZAF) * (RFIF / RLKF)	
XBA	-4.713811	=XBF + (XAF - XBF) * (RFAF / RLKF)	
YBA	5221.231815	=YBF + (YAF - YBF) * (RFAF / RLKF)	

Figure 10: Screenshot of the family parameters

2.5.2. The method’s Workflow

Firstly we take a look at the parameterized families in this workflow

<u>Installation surface family</u>: the main concept of this Revit family is creating a number of dummy plates arranged in a helical pattern Fig(11). These plates are used to place the ramp plates on them afterward.

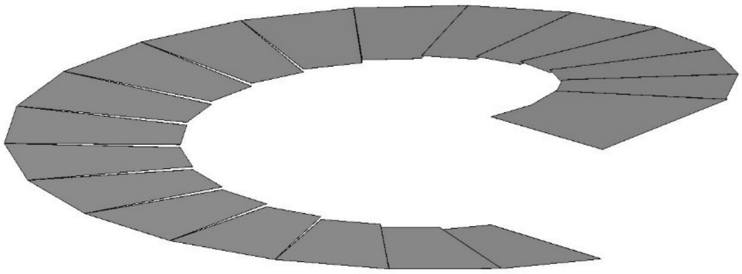


Figure 11: The installation surface family

Konstruktion		
Breite_Feld_außen	2782.3	=sqrt(Projektion_Feld_außen ^ 2 + Steigung_Feld_außen ^ 2)
Breite_Feld_innen	1447.6	=sqrt(Projektion_Feld_innen ^ 2 + Steigung_Feld_innen ^ 2)
Breite_Podest_außen	5514.4	=sqrt(Projektion_Podest_außen ^ 2 + Steigung_Podest_außen ^ 2)
Breite_Podest_innen	2855.5	=sqrt(Projektion_Podest_innen ^ 2 + Steigung_Podest_innen ^ 2)
Projektion_Feld_außen	2778.3	=2 * Außenradius * sin(Feldwinkel / 2)
Projektion_Feld_innen	1445.5	=2 * Innenradius * sin(Feldwinkel / 2)
Projektion_Podest_außen	5514.1	=2 * Außenradius_Podest * tan(Feldwinkel)
Projektion_Podest_innen	2855.3	=2 * Innenradius_Podest * tan(Feldwinkel)
Quersteigung_Feld	126.2	=(Außenradius_Feldmitte - Innenradius_Feldmitte) * Querneigung Rampe
Quersteigung_Podest	122.7	=(Außenradius_Podest - Innenradius_Podest) * Querneigung Rampe
Rotation_Feld	3.09°	=atan(Steigung_Feld_außen / Projektion_Feld_außen) * if(Steigung Rampe im Uhrzeigersinn, (-1), (1))
Rotation_Podest	0.57°	=atan(Steigung_Podest_außen / Projektion_Podest_außen) * if(Steigung Rampe im Uhrzeigersinn, (-1), (
Steigung_Feld_außen	149.8	=Steigung_Rampe / Anzahl Rampenfelder
Steigung_Feld_innen	77.9	=Steigung_Feld_außen * (Projektion_Feld_innen / Projektion_Feld_außen)
Steigung_Podest_außen	54.5	=2 * Außenradius * sin(Feldwinkel) * Neigung Parkhaus * cos(Drehung zu Giebel)
Steigung_Podest_innen	28.2	=Steigung_Podest_außen * (Projektion_Podest_innen / Projektion_Podest_außen)
Steigung_Rampe	2695.5	=Geschosshöhe + if(Steigung Rampe ab Tiefpunkt Podest, Steigung_Podest_außen, -Steigung_Podest_a
Versatz Podest	109.2	=Versatz_Rampe + (40 mm * Querneigung Rampe)
Versatz_Rampe	108.0	=Neigung Parkhaus * Abstand zu Tiefpunkt

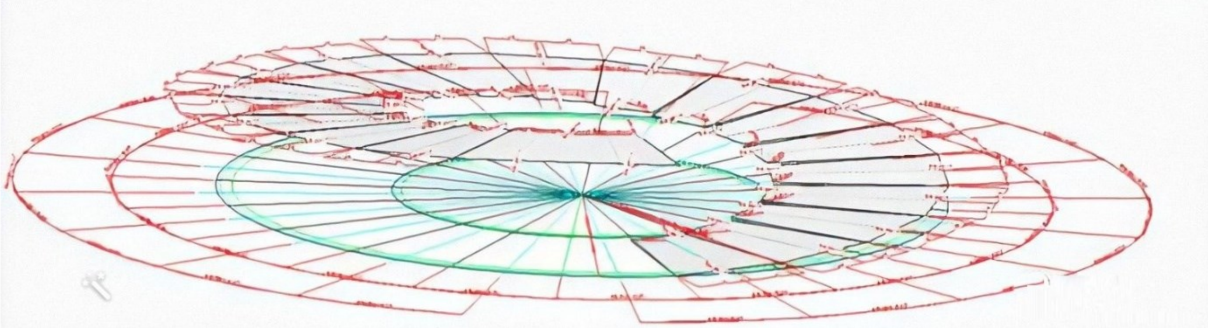


Figure 12: Screenshot of the constraints and formulas of the surface installation family

Through a large number of formulas and reference lines, the vertical position of each plate is calculated in a way that shapes the helical ramp. By changing the inputs e.g. (outer radius, inner radius, story height, etc.), the positions are adjusted according to the new inputs. In other words, the function of this Family is to adjusts the positions in the vertical direction and it will be used as a basis for the Ramp/Landing plate families.

Ramp/Landing plate family: these families are parameterized families that adjust all dimensions of the plate in the horizontal base e.g. (The outer dimensions of the plate, the sidewalk width, etc.) Fig(13).

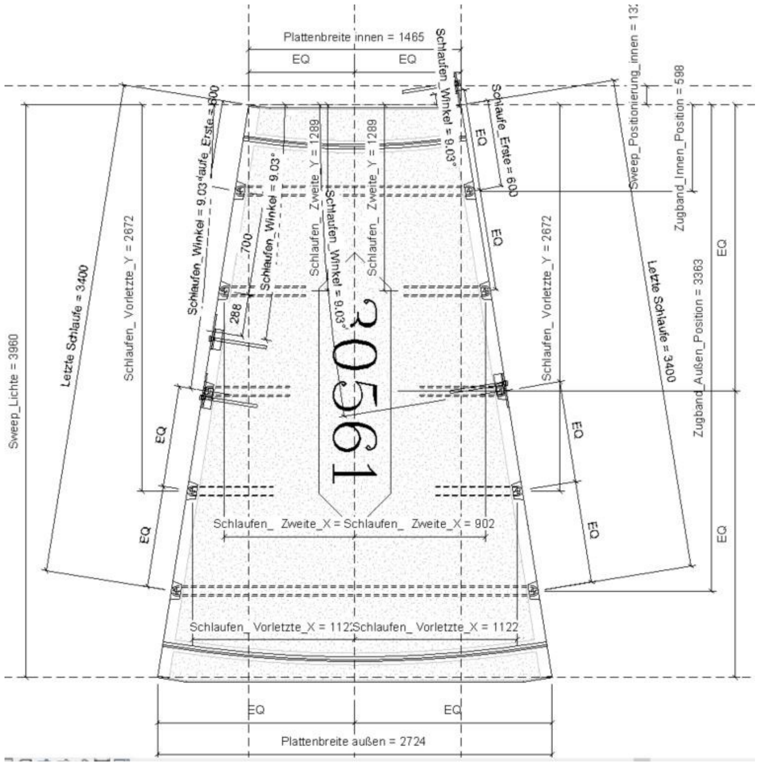


Figure 13: Parameterized Revit family for the Ramp plate (existing method)

The family was parameterized, whereas by changing the inputs e.g. (outer radius, inner radius, angle to the gable side, etc.), all the dimensions of the plate will be adjusted automatically according to the inputted values.

Beam family: it is also a fully parameterized family, which through the formulas and constraints, gets the correct vertically position at each side[1], length, and rotation at each side.

The workflow of this method is divided into 3 steps:

Step 1: modeling the installation surface family and change the input values of its parameters according to the required dimensions and position e.g. (outer radius, inner radius, angle to the gable side, etc.) Fig(14). The main function of the installation surface family is to adjust the coordinates in the vertical direction and will be used as a basis to place the plates on it.

[1] Due tot he hilical shape nature, the beams will have a different vertical position and rotaion at each side(inner side and outer side), in order to form the helix pattern.

Step 2: place the plate families one by one on the installation surface family Fig(14). As done in step 1, the user has to reinput values of some parameters but now for the plate family see Fig(14) e.g. (outer radius, inner radius, angle to the gable side, the plate number, etc.).

Step 3: the user has to model the beams on a horizontal level. Then he has to input their parameter values e.g. (outer radius, inner radius, angle to the gable side, the beam number, etc.).

using these inputs and through the formulas and the constrains that have been programmed inside the family, all the dimensions and elevations are calculated in a way that each plate/beam is positioned in the correct elevation and with the correct dimensions.

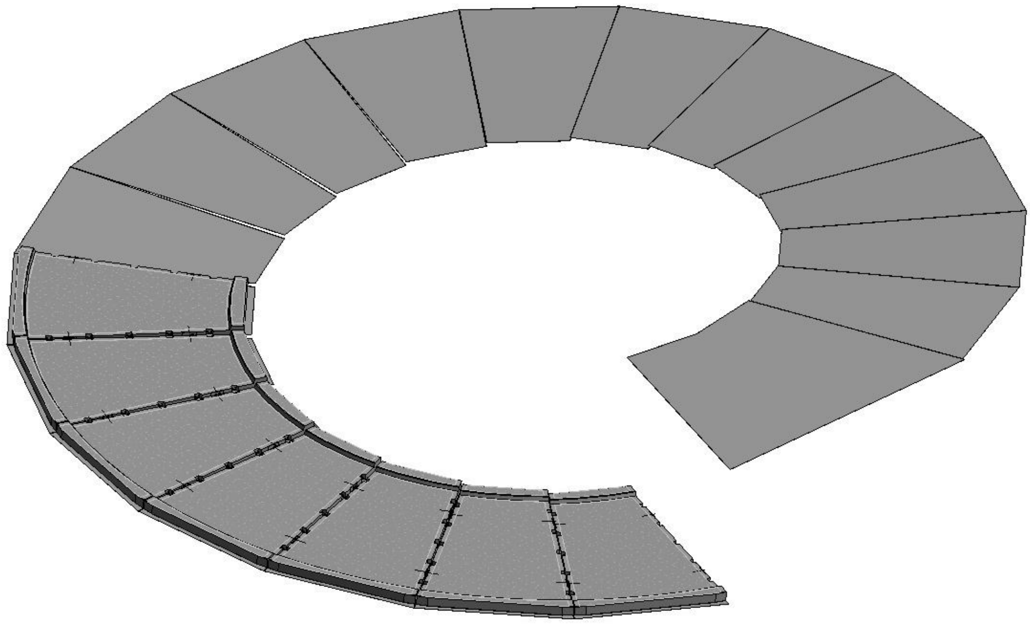


Figure 14: Some of the ramp plates are placed on the installation surface family

2.5.3. The method's input Interface

This modeling method does not have a clearly defined input interface. Since it uses only the BIM system, there is no chance to create a user-friendly input interface. The user has to search for the family parameters that need to be changed for the case he wants to model. For example, the inner/outer radius and other geometrical inputs Fig(15).

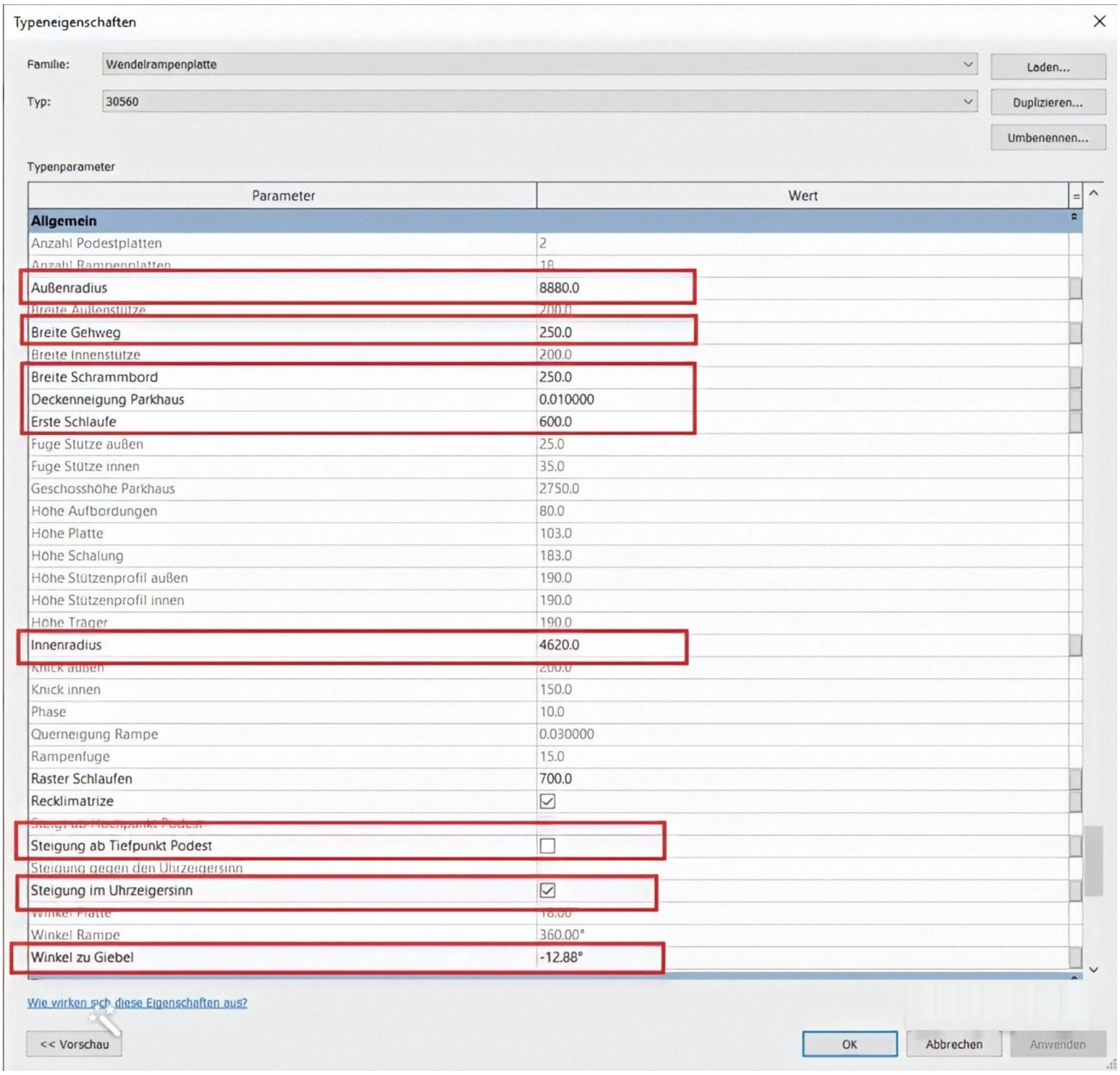


Figure 15: Screenshot from a Revit family parameter window

These inputs have to be manually given in each of the aforementioned family types (Installation surface family, Ramp plate family, Landing family, and beam family)

3. Research methodology

In this chapter, we will discuss the methodology of the research of developing an automation workflow for modeling the Helical Ramp. First, the geometrical aspects will be covered to identify the geometry that needs to be modeled. In this first part the structural system, the boundary conditions, and the required outcomes are discussed. And in the last part of this chapter the proposed BIM workflow, the conditions of the performance, and the Usability are explained.

3.1. Geometrical aspects of the helical ramp

A Helical Ramp for a multi-story car park can have different structural schemes. There are various approaches in the market to build such a structure using different building materials. Nevertheless, this study has only considered the design and the construction way, used by the company Goldbeck. This assumption is helpful to limit the complexity of the work. Such as the variability of the construction methods and the building materials e.g. some companies use cast in situ concrete, which has completely different boundary conditions and design processes. In this section, the Geometrical aspects of the parking garage as well as of the helical ramp used by Goldbeck are introduced.

3.1.1. Multi-storey Park system

To understand the system of the Helical Ramp, we have to take a look into the system of the garage building, which is served by the ramp. The Structural scheme of the Park consists of the steel skeleton and precast concrete Plates Fig(16). The connection between the steel girders and the concrete plates is carried out by some connectors Fig(20). These connectors sustain the horizontal forces, while the vertical forces are maintained by the weight of gravity from the concrete plate through the beams reaching to the columns.

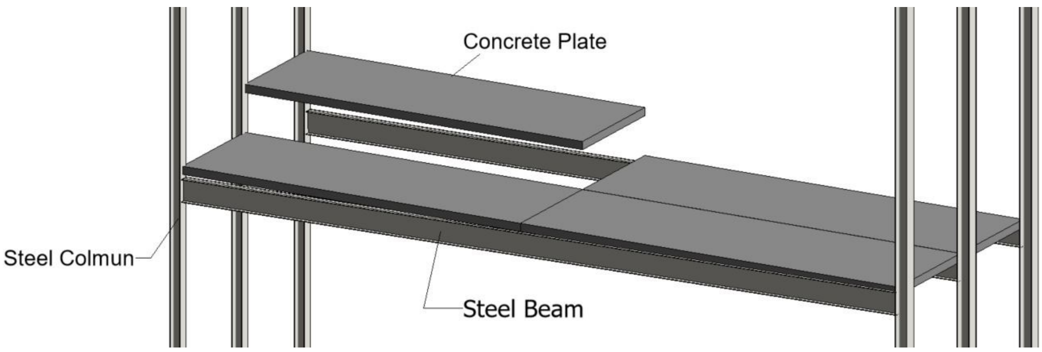


Figure 16: Structural scheme of the garage building

The basic of the Goldbeck parking garage system consists of 16 m wide units. This wide ensues of 6 m roadway as well as parking spaces of 5 m depth to the left and right. A 16 module is 2.5m or 2.7m wide and thus corresponds to a parking space width. There is a transverse slope perpendicular to the parking spaces for the water drainage Fig(17). The floor height is 2,75 m, while the minimum headway height is 2,1 m.

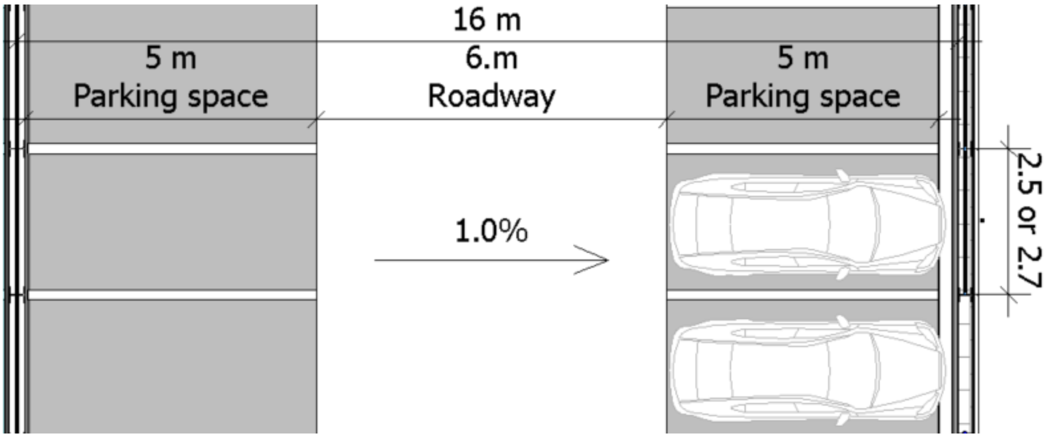


Figure 17: Plan view of the garage

3.1.2. Helical Ramp System

Like the Multi-story parking structure scheme, the Helical Ramp structure scheme is also consist of a steel skeleton and precast concrete Plates Fig (18) with connectors between the plates and the steel beams.

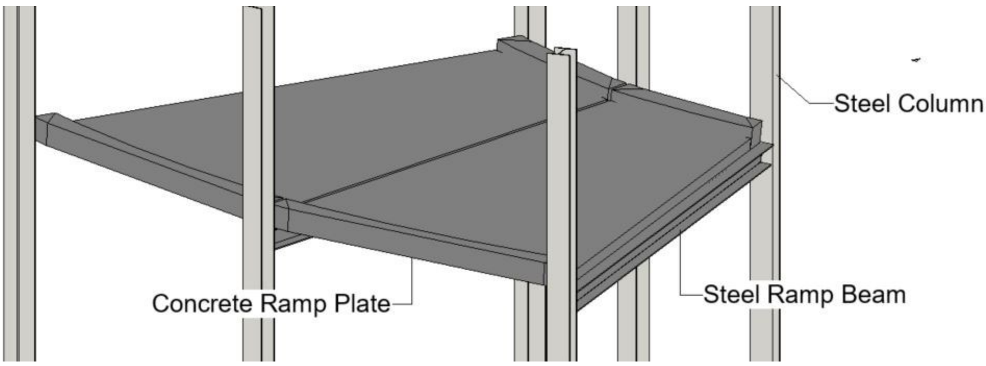


Figure 18: Structural scheme of the helical ramp

The beams and the plates are positioned in a specif way to ensure the smoothness of the roadway of the ramp. Each beam/plate is rotated in 2 directions (Longitudinal and transversal). it has two different longitudinal rotations at the inner and outer side and one transversal rotation Fig(19). All of these rotations in addition to the elevations of the beam's ends and plate's corners have to be correctly calculated and modeled to ensure the smoothness of the ramp.

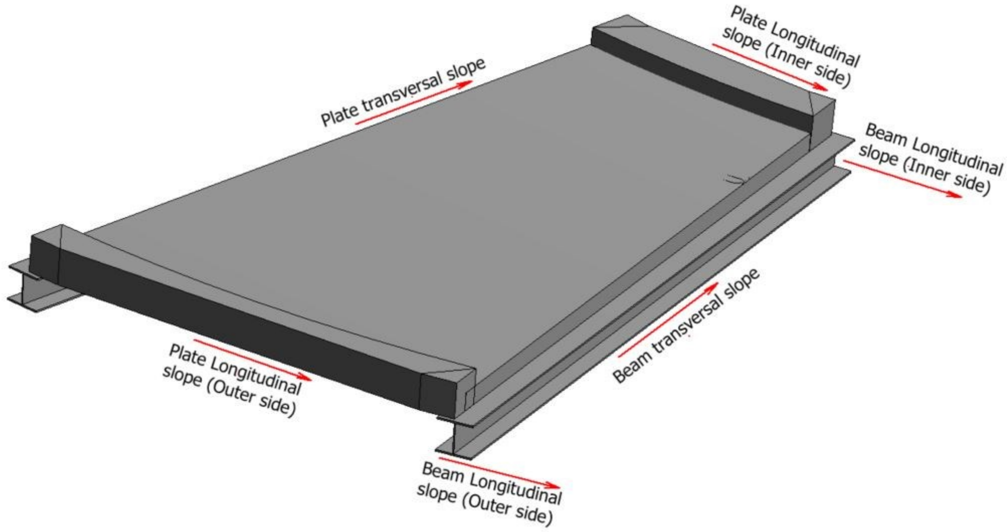


Figure 19: The slopes of the beams and plates

For a geometry as complex as the helical ramp, many boundary conditions must be met, in order to correctly model the structural elements and avoid the conflicts during the production and construction phase. These conditions will be demonstrated in this section, which will help us to imagine how complex the geometry is. These conditions are used as a basis for the mathematical calculation model and the parameterized adaptive Revit families, which are demonstrated in detail in chapter 4.

3.1.3. Geometrical boundary conditions

As will be discussed in detail in the following sections, the main concept of the proposed workflow developed in this study is to create a mathematical algorithm that calculates the coordinates of the structural elements using geometric inputs see Table(2) and boundary conditions. The calculated coordinates are then exported to parameterized adaptive families so that the structural elements with the correct dimensions and positions will be modeled. The geometric boundary conditions guarantee a seamless production and execution phase without clashes, as well as effective utilization of the structure during the service phase (Rynkovskaya, 2019). The geometrical boundary conditions can be summarized in the following points

- <u>The clear height of the floor</u>
  According to the civil engineering codes (2015) (2016) for parking, the clear height of the storey is one of the most important aspects, that should be taken into account while modeling the helical automobile ramp.
- <u>The smoothness of the entrance between the car park and the ramp</u>

The elevations of the landings of the ramp should be adjusted to ensure smooth access between the car park and the ramp. This maintains the operational efficiency of the Structure.

- The connection between the structural elements.
  The connection between the beams and the plates should be modeled neatly and with minimal errors to ensure the smooth assembly of the structure. the joints between the plates must also be considered, which must be preserved along the length of the plate Fig(20).

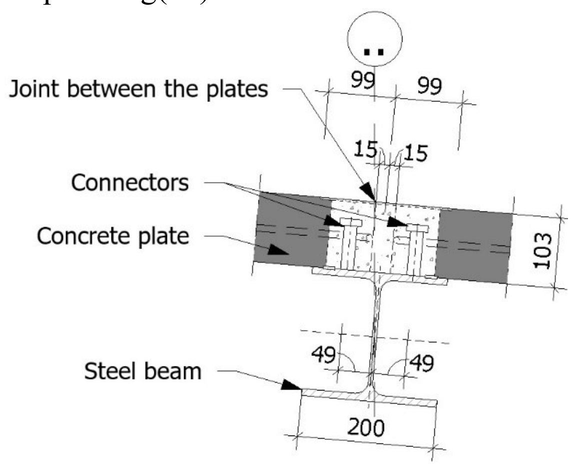


Figure 20: the connection between the beam and the plate

3.1.4. The required Outputs from the geometrical model.

According to the standard of Goldbeck, the following outputs are essential for the production and construction phases. These required outputs will be used as a reference in creating the Revit families (section 3.4.3).

- **Accurate 3D Model**

In the case of Helical Ramp there are some benefits of delivering an accurate 3D Model as following:

- An accurate 3D model is essential for the collaboration, which ensuring all design stakeholders have insight into the model. It
- It helps to get a preconstruction visualization.
- It could be used to make a clash detection before the construction phase.
- It helps to get an accurate cost calculation.

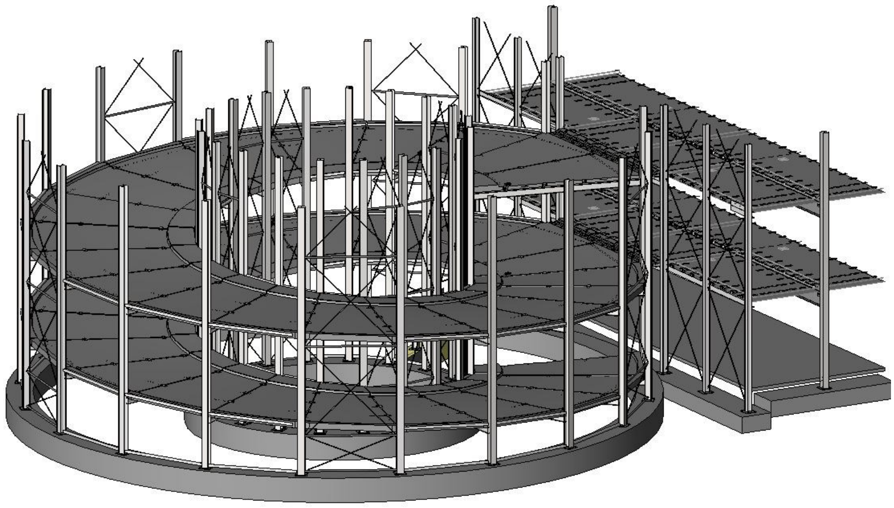


Figure 21: Example of a finished 3D Model of Helical Ramp

- Floor plans

Floor plans with all required dimensions and details for the construction phase.

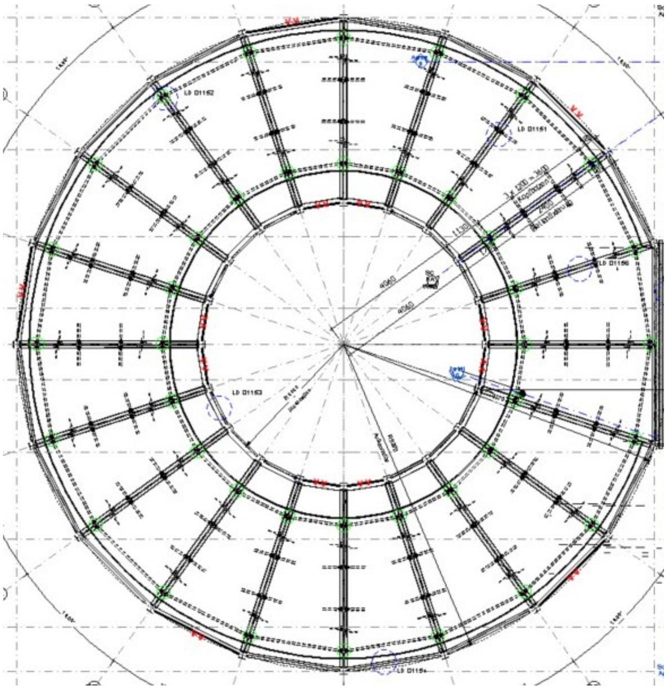


Figure 22: Example of a finished floor plan of the Helical ramp

- **Shop drawing for the plates**

Complete shop drawing for the ramp plates and landing's plates are required for the production phase. The shop drawing must have the following outputs:

  - The elevation of each corner point of the plate
  - All dimensions of the plates including all the required details.

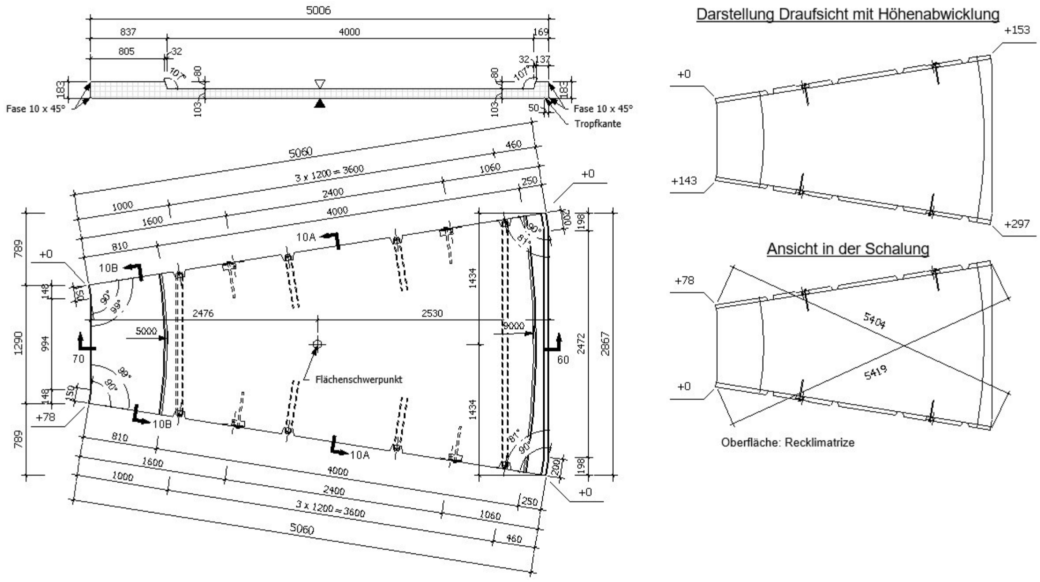


Figure 23: Example of a finished shop drawing of a ramp plate

- **Schedule of the beams**

A schedule for all beams is required to be automatically generated. This Schedule must have the following outputs for the steel production of the beams:

  - The elevation of each beam-column connection bolts for the inner and outer sides
  - The inclination angles of the beams at the inner and outer sides

<Wendelrampenträger rechts>				
A	B	C	D	E
Achse	Höhenkote Achse Lochbild a	Drehung Lochbild außen	Höhenkote Achse Lochbild in	Drehung Lochbild innen
Ebene 00 bis Ebene 01				
81	-91 mm	2.95°	-245 mm	-6.57°
80	+58 mm	2.95°	-96 mm	-6.57°
79	+208 mm	2.95°	+54 mm	-6.57°
78	+358 mm	2.95°	+203 mm	-6.57°
77	+507 mm	2.95°	+353 mm	-6.57°
76	+657 mm	2.95°	+502 mm	-6.57°
75	+806 mm	2.95°	+652 mm	-6.57°
74	+956 mm	2.95°	+802 mm	-6.57°
73	+1105 mm	2.95°	+951 mm	-6.57°
72	+1255 mm	2.95°	+1101 mm	-6.57°
71	+1404 mm	2.95°	+1250 mm	-6.57°
70	+1554 mm	2.95°	+1400 mm	-6.57°
69	+1704 mm	2.95°	+1549 mm	-6.57°
68	+1853 mm	2.95°	+1699 mm	-6.57°
67	+2003 mm	2.95°	+1849 mm	-6.57°
66	+2152 mm	2.95°	+1998 mm	-6.57°
65	+2302 mm	2.95°	+2148 mm	-6.57°
64	+2451 mm	2.95°	+2297 mm	-6.57°
63	+2599 mm	1.77°	+2443 mm	-3.93°
62	+2633 mm	0.57°	+2484 mm	-1.27°

Figure 24: Example of the required beam schedule

3.2. The methodology of gathering new ideas and concepts

To go more into detail and get more new ideas and concepts, the author decided to collect some ideas and concepts before starting to design the workflow proposed in this study. Since collecting concepts and ideas from the literature is essential, obtaining ideas and concepts from the people working in this field would also be very beneficial to consolidate the methodology used in this study. In this chapter, the method used in this study to assemble the new concepts and ideas is discussed.

A workshop was held by the author with a group of experienced engineers from the company Goldbeck. A creativity technique was applied to get the foremost creative ideas from the members.

3.2.1. The applied creativity technique (6-3-5 technique)

The 6-3-5 technique is a unique form of brainstorming through graphic media, specifically, it is classed among the intuitive and advanced methodologies, as it consists of cyclically advancing the inspiration of other members. The basis of such a technique is the conviction that the success of an idea generation process is determined by the degree of input and integration with the proposals of the other members.

The 6-3-5 method (Bernd, 1969) was introduced as an alternative to brainstorming. The name of this method is reflecting the structure, in which a team of 6 members writes 3 ideas every 5 minutes. After each five-minute round, the concepts are passed on to the neighboring member. The team can then be inspired by the ideas of the others. Assuming all participants properly finish the workshop, a 30-minute workshop should generate 108 ideas. The results of the workshop would then be used for further concept design and assessment (Wodehouse, et al., 2011).

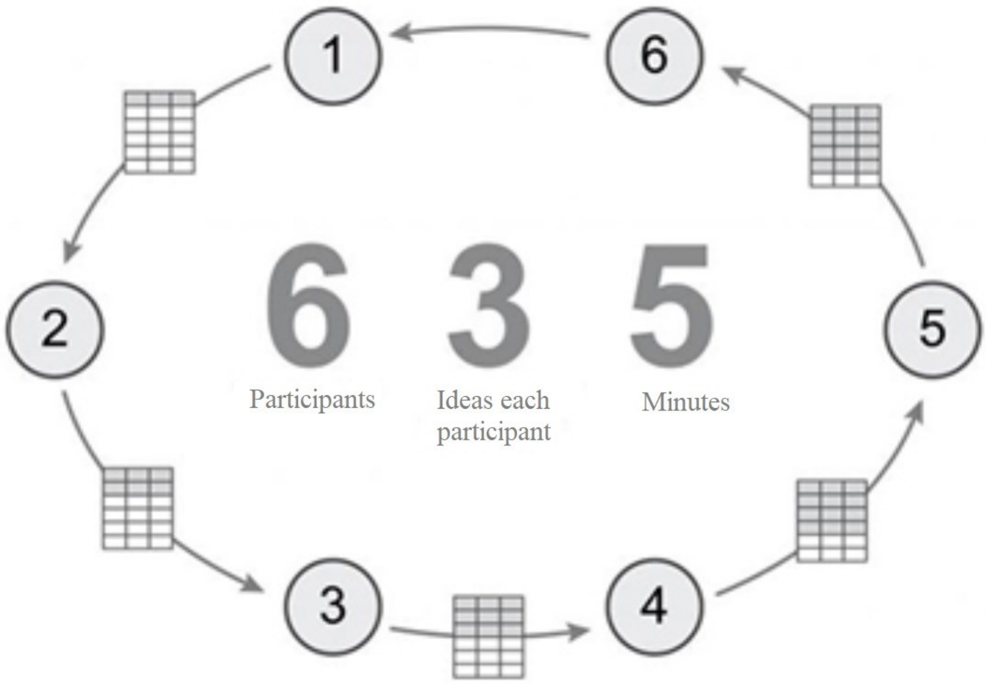


Figure 25: Sketch shows the process of the 6-3-5 method.

3.2.2. The workshop

In this section, the implementation of the method 6-3-5 is discussed. The Author has organized a workshop. Six experienced engineers in topic modeling helical ramp are invited to join this workshop. Table 1 shows the agenda of this workshop

Table 1: The Agenda of the workshop

	Part of the workshop	**Time[Min]**
1	Introduction to the topic	5
2	Explanation of the method 6-3-5	5

3	6-3-5 Rounds	30
4	Discussion round	20

The author served as the moderator of the workshop. Due to the difficulties of conducting this workshop in a room, it was mandatory to hold it online. The challenge was to adapt this technique so that it could be held online. The author created an online table with 3 columns and 6 rows Fig(26). Each column represents an idea and each row represents a participant. Each cell has a card, pressing on the card opens a window with a question at the top, in addition to six empty spots, one for each participant see Fig(27).

Figure 26: online table for the method 6-3-5.

Figure 27: The window opened by pressing the card.

In the first round, each participant opens the three cards in the row with his/her name and writes three ideas in a five-minute time box. After the first five minutes (of the first round) the moderator moves the cards in each row one row down and starts the second round. In the second round, each participant finds the ideas that the other participant wrote in the last round and therefore has to write 3 ideas that develop the ideas of the other participant again. This sequence is repeated 5 times until each participant gets his own cards again and sees how his very first 3 ideas are developed by the other participants. After that, a discussion round was done, which each user had 5 min to discuss his ideas for the group.

3.2.3. The results and assessments

The total results of this workshop were 3 ideas times 6 participants times 6 rounds, equivalent to 108 ideas in 30 minutes. It is also noteworthy that there is a certain redundancy in the ideas, but eventually, the amount of new ideas is still considerable. The author collected these ideas and assessed them to get some ideas that can be applied in this study.

The evaluation of the workshop results is based on categorizing all ideas into five categories as follows:

- Interesting ideas that could be applied in this study
- Interesting ideas, but not suitable for the purpose of the study (Future studies)
- Ideas that are not suitable for the objective of the study
- Repeated ideas

The ideas derived from this workshop were useful in either gaining a deeper understanding of the helical ramp system and implementing some concepts in the design of the helical ramp modeling workflow. These concepts will be discussed in the following chapters

3.3. The workflow methodology

In designing the proposed BIM tool, the author tried to combine what can be learned from BIM and also from the computer science field to obtain a tool with good performance for the user and with good output quality for the BIM process. This chapter demonstrates the methodological aspects of the proposed workflow. At first, the general aspects regarding performance and usability are discussed from a computer science perspective. Then the proposed BIM methodology is explained in Section 3.2.3.

3.3.1. Tool Performance.

The performance of the tool is one of the most important aspects, that has been taken into consideration. The performance of the software tool is particularly important in engineering and scientific studies, where complex and time-consuming calculations are performed, and the processing time is expensive and limited (Mahmudova, 2019). To make the term performance clear. The performance is a capability of the software tool to be less dependent on the resources of the device: processing time, and transmission capability of communication channels of capacity occupied in internal and external memory. According to (Mahmudova, 2019) there are some recommendation to achieve high performance in the software area

- Using an additional program to increase software performance
- Using software capabilities to increase its Performance

- Increasing the Programmer's Performance to increase software performance
- Parallelize the processes
- Exclusion of any task, that is, manages on without it.

3.3.2. Tool usability

Usability is one of the important factors in designing a new software tool (Bevan, 1995). There are many different preceptive to define and evaluate usability. According to ISO/IEC 9126-4 (2001) standard, the difference between usability and the Performance in use is a matter of context of use. Specifically, when usability is defined, the focus is on improving the user interface (Jain, et al., 2012). According to the ISO Society for Computer Science (Nielson, 2010), there are some factors that can be used to define the usability of the software tool. These factors were used in this study as boundary conditions in the development of the tool in this work and will be used to evaluate the workflow in chapter 5 and in the comparison with the other workflow that describes in section 2.4.

- Learnability

This attribute describes to what extent the user can learn the application and use its functions. By addressing the following question we can assess the learnability of the tool (Jackson, et al., 2011), What does it require for the user to learn the basic/advanced functions? special training, tutorials, or only instructions.

- Efficiency

This attribute gives an idea of the extent to which the application can offer all the functional variability required by the market. Table (2) shows the functional variabilities in the case of Helical Ramp and its necessity.

Table 2: The functional variabilities in case of the helical ramp

	Function	Description	Necessity
1	clockwise/ counterclockwise	This function allows the user to control whether the Ramp rise clockwise or counterclockwise	High
2	Position of the Ramp	This function is a general function that has some other sub-functions related to the position of the helical ramp to the main building of the park, e.g. the ramp located at the right/left side of the main building, the ramp at the gable/long side, etc. All these functions are described in detail in section 3.4.2.	High

3	Automatic generation of more stories	This feature offers the user the ability to automatically generate more stories without having to manually copy/model them.	Medium
4	Ramp Slope	Adjustment of the ramp slope according to the demand	High
5	Ramp dimensions	Flexibility in the choice of ramp dimensions ▪ Inner radius ▪ Outer radius ▪ Height of the story	High
6	Number of Fields	Flexibility in the choice of Number of Fields	Medium
7	Wide of Entrance	Flexibility in the choice of car entrance width	High
8	Controlling the error	This option offers the user the possibility to manually control the error of the calculation model, which gives the application more flexible.	Low

- Memorability

It is defined as the characteristic of the algorithm that allows the developer to recognize the elements and functionality of the algorithm after a certain time not adjusting it (Nielsen, 1993). The clarity of the algorithm in terms of its structure and functionality plays the main role in this context.

- Satisfaction

It is defined as the level to which the application is user-friendly, attractive, and trustworthy for users (Seffah, et al., 2006).

User-friendly: It is the degree to which the tool's interface is straightforward and understandable to the user.
Attractive: It is the ability of the system to be visually attractive to the user.
Trustworthy: It is the confidence that the application offers to its users. In this attribute, there are two main points to be discussed. Firstly, the stability of the program that the user expects. Secondly, the transparency of the results, the more transparent the internal processes are to the user, the more confidence the user has in the application.

3.3.3. The proposed workflow Methodology.

The author attempted to design the new tool by developing a workflow that takes into account the above-mentioned computer science aspects (the performance and usability aspects) as well as the BIM aspects mentioned in Section 2.3.

As mentioned in section 2.3, there are various approaches could be followed, either by depending only on the BIM system or by getting assistance from other systems e.g. graph-based systems. On the other hand from the computer science perspective and as mentioned in the previous sections, the main idea of increasing the software performance is to make it less dependent on the resources of the device. This will lead to a decrease in the processing time and increases the transmission capability of communication channels of capacity occupied in internal and external memory. According to (Mahmudova, 2019) there are some recommendations to achieve high performance, e.g. by using an additional program, this can be done by inserting Microsoft Excel into the process. Another recommendation is to parallelize the processes, this could be done by splitting the processes to the software in the workflow, in other words, do not let just one software do all the processes.

By considering the above-mentioned aspects, the possible basic workflow could be categorized into three main workflows. Firstly, workflow 1 depends only on the BIM system i.e. all the parametric processes done within Autodesk Revit. This workflow allows us to achieve only associative modeling with single operation iteration. The second workflow is to implement a graph-based system with the BIM system i.e. using Autodesk Revit and Autodesk Dynamo. This supports Dataflow modeling with explicit Multi-operation iteration. The last workflow is to implement more than an assistant system to the process side by side with the BIM system i.e. using Autodesk Revit, Autodesk Dynamo, and Excel. In this workflow, Excel is used to increase the performance by moving all the complex mathematical calculations to it.

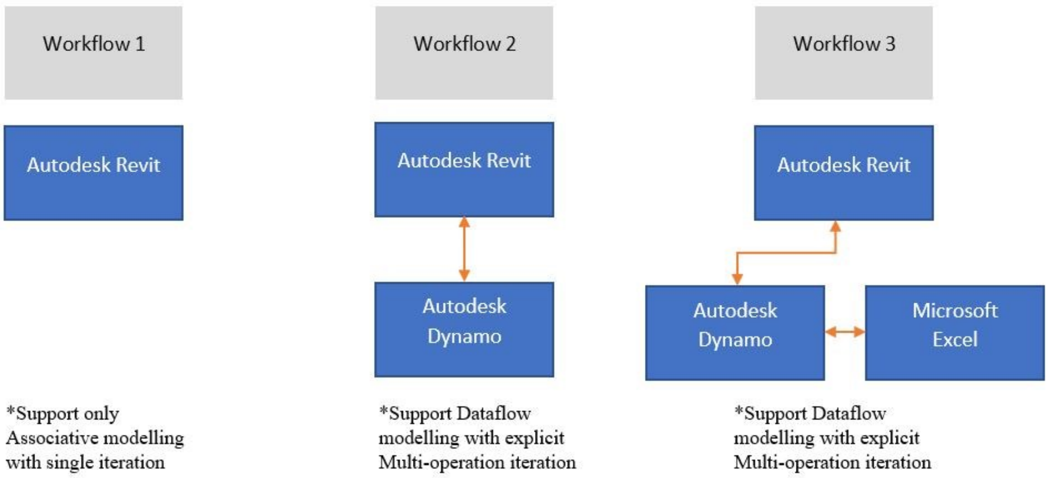


Figure 28: 3 different basic workflows of BIM parametric modeling.

In order to choose one of these basic workflows to design the detailed workflow for the Helical ramp, the author decided to test all of the three basic workflows in a simple benchmark task. The task was created to show the advantages and disadvantages of the mentioned BIM workflows concerning parametric modeling. The benchmark task is to generate a number of floor plates with the shape of an octagon. The plate dimensions are associated with the floor level. The dimensions are decreased till the flip point, then the plate dimensions are increased again, in which the tower takes the shape of sandglass Fig(29).

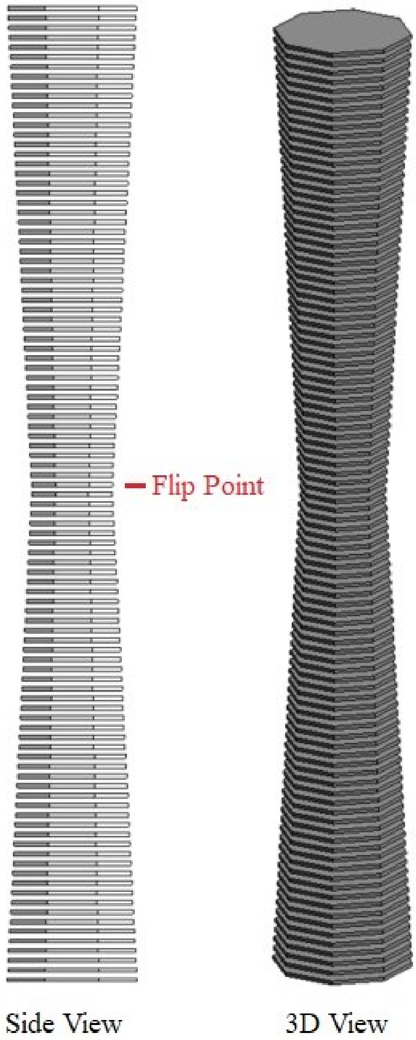


Figure 29: A side view and 3D view of the benchmark task

The task should be parameterized so that the user can change the following parameters and the tower's shape is automatically adjusted:

1. Number of floors
2. Total height of the tower
3. Length of the first plate

4. Thickness of the plates
5. The position of the flipping point

The above-mentioned workflows are used to carry out this task.

- **Workflow 1**

In this workflow, only the BIM system is used. Because the BIM system supports only associative modeling with single operation iteration, several single operation steps have to be performed one after the other. An Adaptive family is created with all required parameters and logic. There are two types of parameters that can be done in Autodesk Revit families.

Type parameter: This allows us to change the parameter value that applies to all elements of the family type. This type of parameter is appropriate for the values that are not changed between the plates.

Instance parameter: Enables to change the parameter value for each instance individually.

The parameters are categorized into three categories:

- The input parameters
    - Number of floors (Type parameter)
    - Total height of the tower (Type parameter)
    - Length of the first plate (Type parameter)
    - Thickness of the plates (Type parameter)
    - The position of the flipping point (Type parameter)
    - The plate’s number (Instance parameter)
- The calculated parameters
    - The level Hight (Type parameter)
    - The length of the plate (Instance parameter)
    - Octagon corner (Instance parameter)
    - The level of the four corners of the plate (A1, A2, A3, and A4) (Instance parameters)

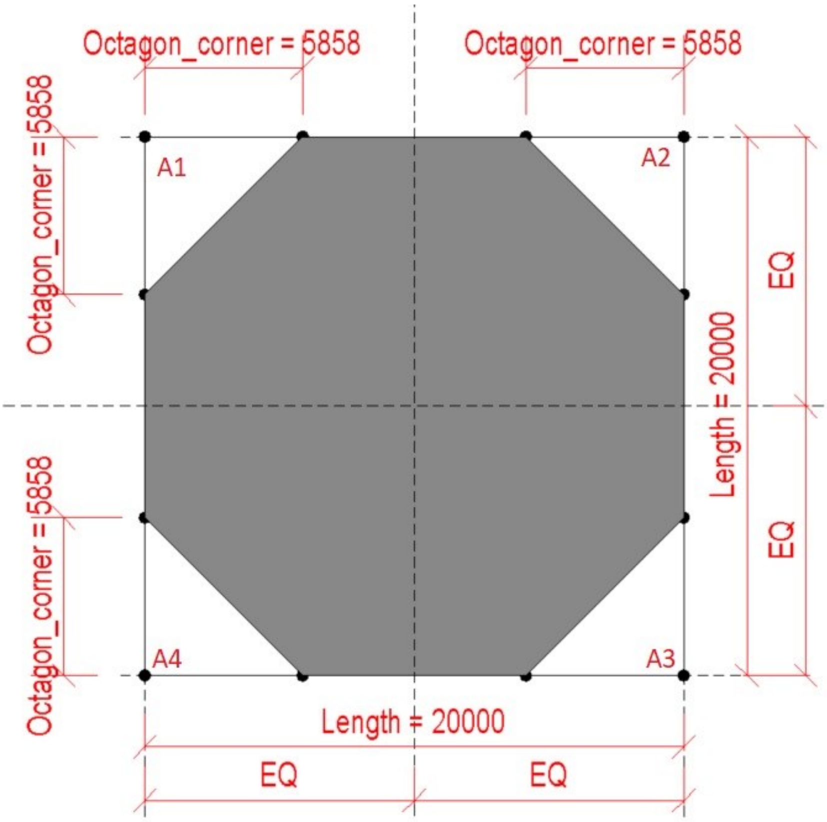


Figure 30: Plan view for the family parameters of the Octagon plate.

The steps of this workflow are as shown in Fig(31).

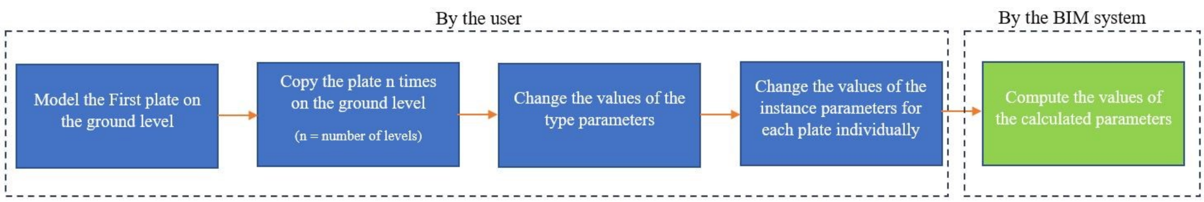


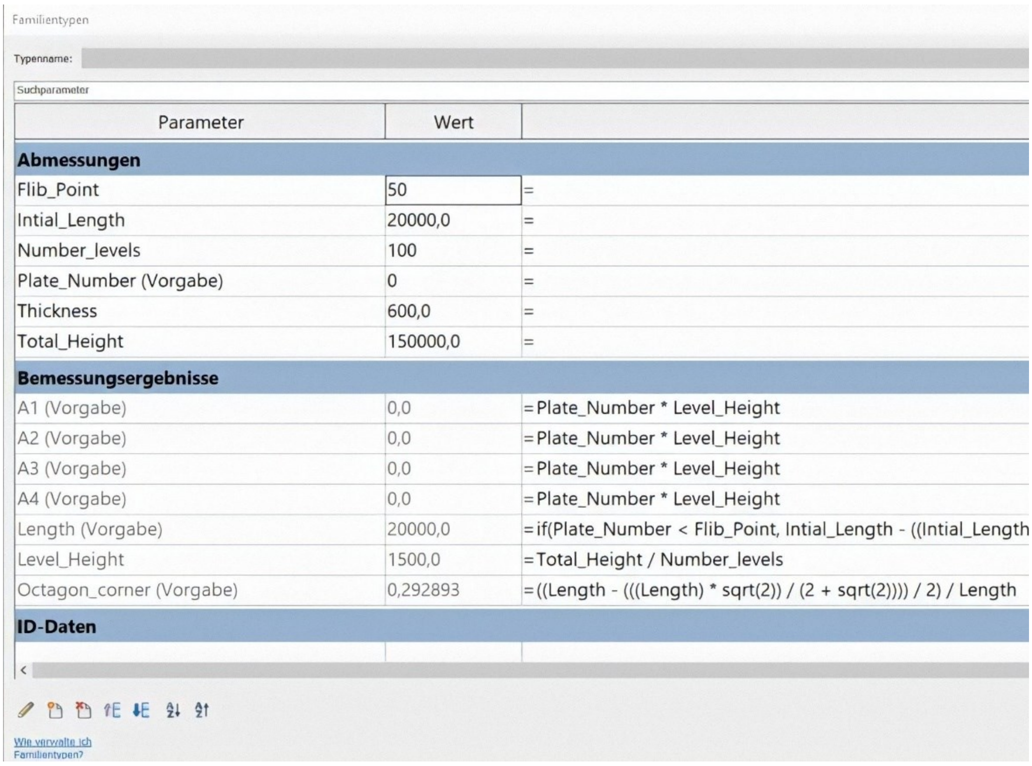
Figure 31: Steps of Workflow 1

Firstly, the user has to model the first plate at the ground level. Then he copies the family as many times as he wants at the same level. Then he inputs the desired values of the above-mentioned input parameters. Through the logic formulas, the calculated parameters will be automatically computed and consequently, each plate will take the right position and the correct dimension, in which the tower takes the sandglass shape.

The logic of the formulas of the calculated parameters are as follow:

- *Level Height = Total_Height / Number_levels* *[1]*

- *Length = if(Plate_Number < Flib_Point, Intial_Length - ((Intial_Length / Number_levels) * Plate_Number * 0,75), (Intial_Length - ((Intial_Length / Number_levels) * Flib_Point * 0,75)) + ((Intial_Length / Number_levels) * (Plate_Number - Flib_Point - 1) * 0,75))* [2]
- *Corner Octagon = ((Length) * sqrt (2)) / (2 + sqrt (2))* [3]
- *A1 = A2 =A3 = A4 = Plate_Number * Level_Height* [4]


Familientypen

Typenname:

Suchparameter

Parameter	Wert	
Abmessungen		
Flib_Point	50	=
Intial_Length	20000,0	=
Number_levels	100	=
Plate_Number (Vorgabe)	0	=
Thickness	600,0	=
Total_Height	150000,0	=
Bemessungsergebnisse		
A1 (Vorgabe)	0,0	=Plate_Number * Level_Height
A2 (Vorgabe)	0,0	=Plate_Number * Level_Height
A3 (Vorgabe)	0,0	=Plate_Number * Level_Height
A4 (Vorgabe)	0,0	=Plate_Number * Level_Height
Length (Vorgabe)	20000,0	=if(Plate_Number < Flib_Point, Intial_Length - ((Intial_Length
Level_Height	1500,0	=Total_Height / Number_levels
Octagon_corner (Vorgabe)	0,292893	=((Length - (((Length) * sqrt(2)) / (2 + sqrt(2)))) / 2) / Length
ID-Daten		

Wie verwalte ich Familientypen?

Figure 32: The parameters of the Autodesk Revit Family of the benchmark task.

- **Workflow 2**

In this workflow, a graph-based system (Autodesk Dynamo) is used with the BIM system(Autodesk Revit). A data flow modeling with An explicit multi-operation iteration is performed. The adaptive family that prescribed in workflow 1 is used. Additionally, an algorithm using Autodesk dynamo is written.

The main concept of the dynamo script is to make a multi-operation simultaneously. The process of the dynamo script is divided into four main steps. Firstly, collecting the values of the parameters by the user through a user-friendly interface Fig(34). Then to model all the plates at the ground level. After that changing the values of the parameters into all plates (the type and instance parameters) according to the inputted values.

The steps of this workflow are as follow:

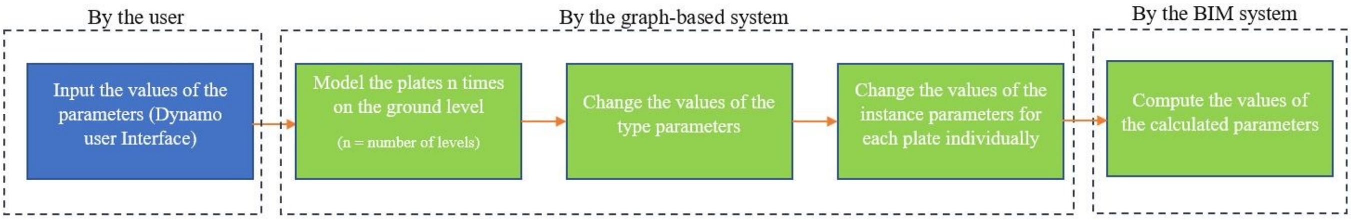


Figure 33: Steps of Workflow 2

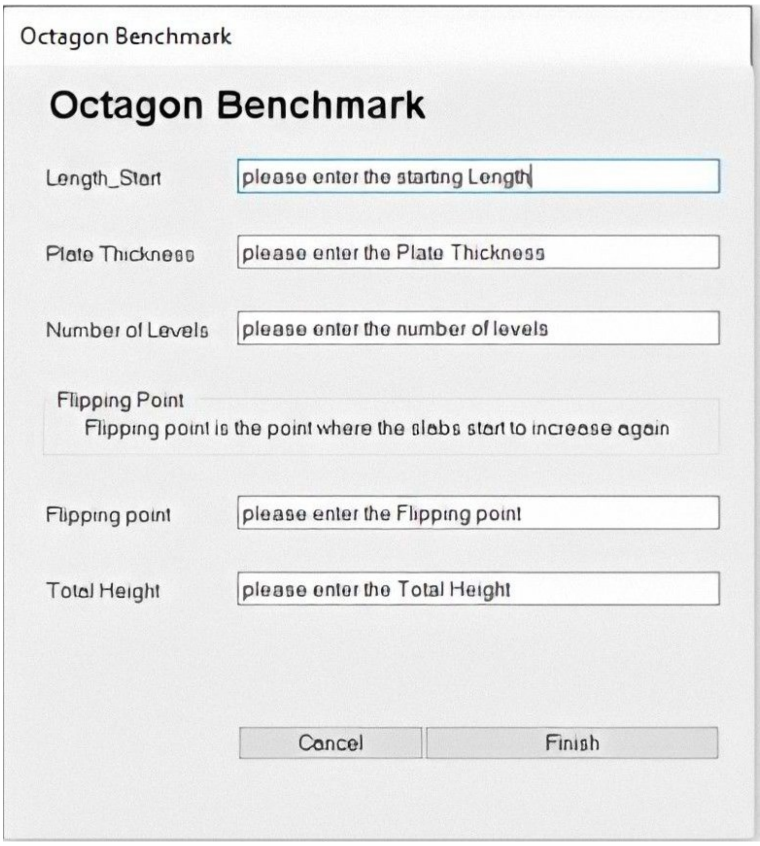


Figure 34: User Interface to input the values of the parameters

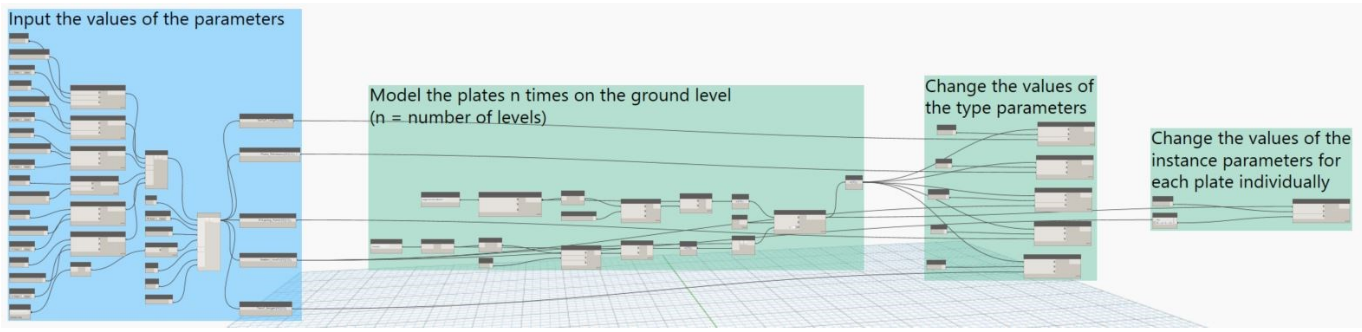


Figure 35: The dynamo script of the workflow 2 showing the four steps of the process.

After inputting the values of the parameters and press on the Finish button, the following 3 steps of the process are done automatically without any help from the user. That is why this kind of modeling process is called multi-operation modeling. Then the values of the calculated parameters are computed inside the BIM system through the formulas and conditions that programmed into the adaptive families.

- **Workflow 3**

In this workflow, a graph-based system (Autodesk Dynamo) and an external calculation system (Microsoft Excel) are used along with the BIM system (Autodesk Revit).

A calculation tool is programmed using Excel VBA, which does all the mathematical calculations that are needed and moved out from the Revit family. Correspondingly a modified version of the adaptive family that is used in Workflow 1 and 2 is used, in which all the mathematical formulas are removed. A new dynamo script is written, which it exports the calculated values from the calculation model and write them into the adaptive family.

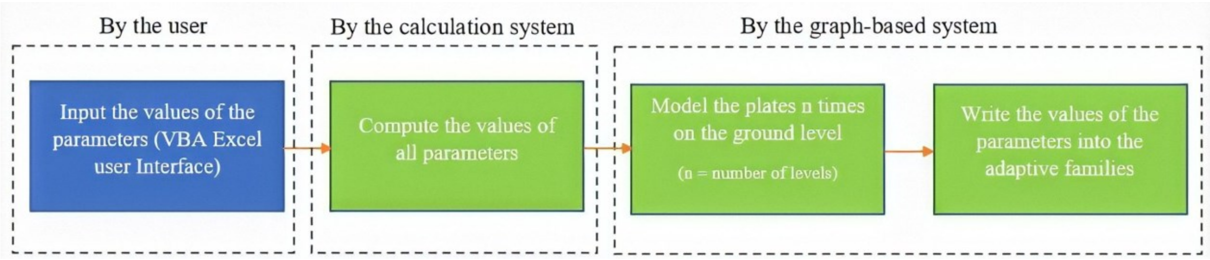


Figure 36: Steps of workflow 3

A user interface is also programmed by an Excel VBA to collect the values of the entered parameters and to execute the calculation model. First, there is an initial interface, which is divided into three parts (input, calculation, and output) Fig(37). The user shall press the "Data input" button. Then a window opens where the user can easily enter all the Input parameters Fig(38).

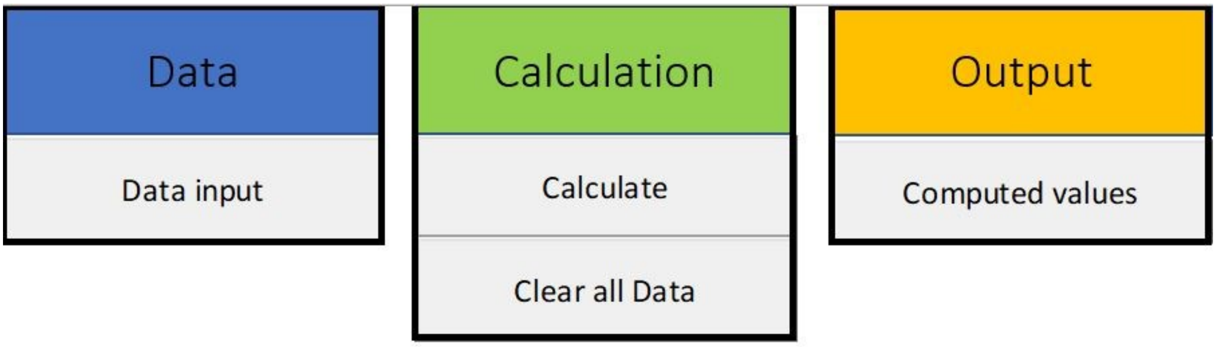


Figure 37: Excel VBA interface of workflow 3

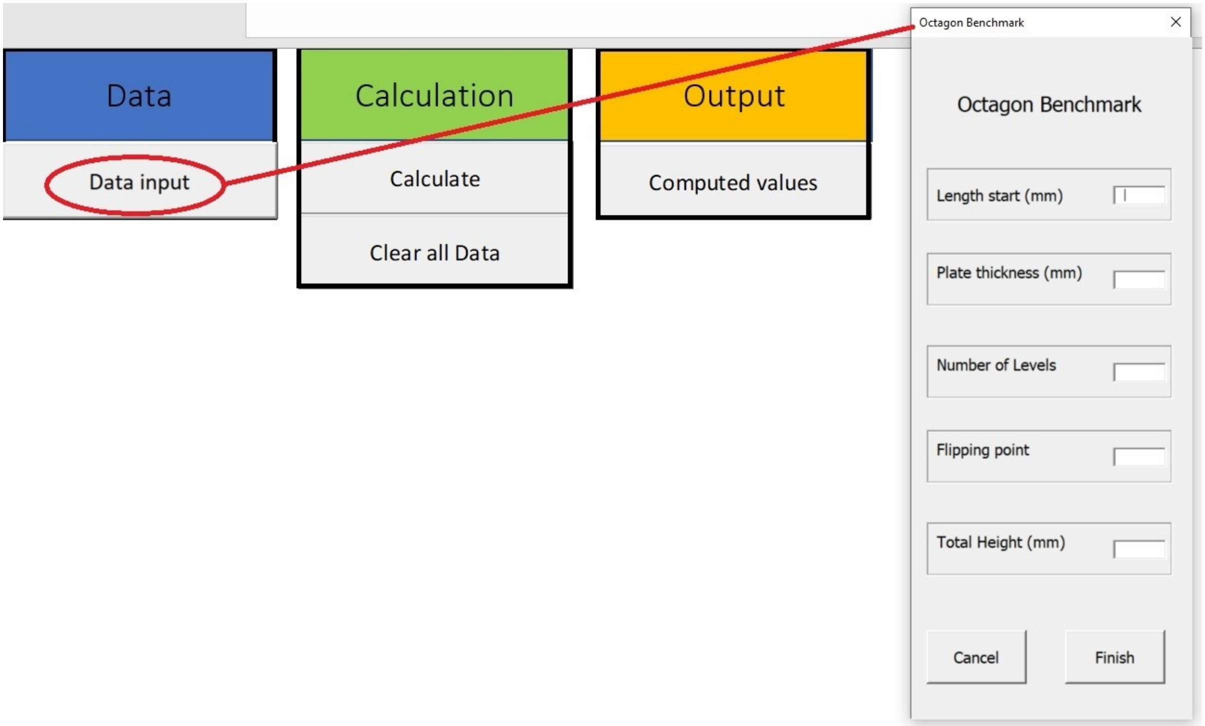


Figure 38: Data input window of the user interface

Next, the user can start the calculation model by pressing the "Calculate" button, where all mathematical calculations are performed in the background and all parameter values are calculated and ready to be exported to the adaptive families. The values can be checked by pressing “Computed values”. Afterward, the dynamo script is used, in which the plates are modeled and the calculated values are automatically written to them. The dynamo script has three main parts (Collect the values from the calculation model, model the plates, and write the values into the adaptive families) Fig(39).

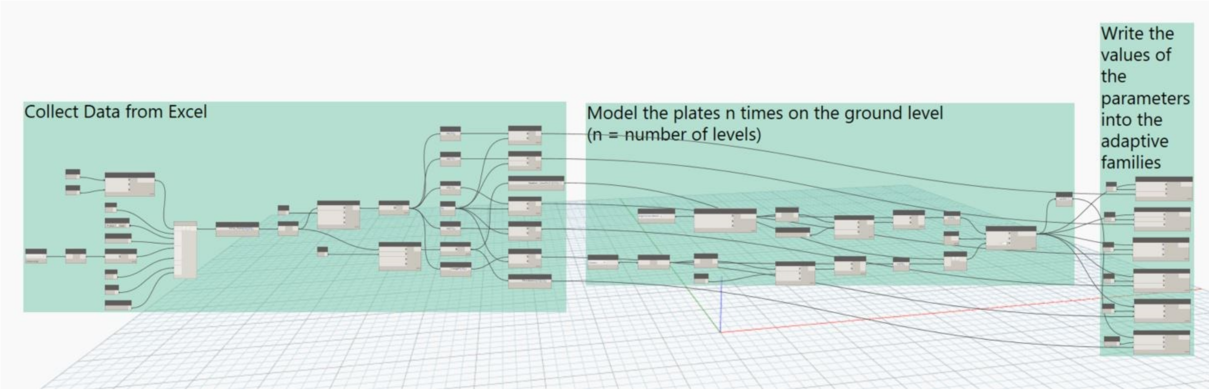


Figure 39: Dynamo Skript of workflow 3

The first part of the script is to collect the data from the calculation model. When the dynamo script is executed, a window opens and asks to select the calculation model file Fig(40). Then the script models the plates automatically and writes the calculated values into them.

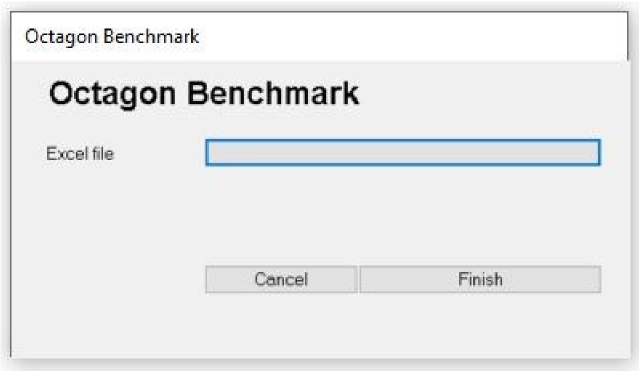


Figure 40: Window of reading the calculation model.

- **Comparison between the workflows**

After testing the three workflows and in order to choose one of these basic workflows to design the detailed workflow for the Helical ramp task, a comparison between them will be done in terms of the performance and the usability.

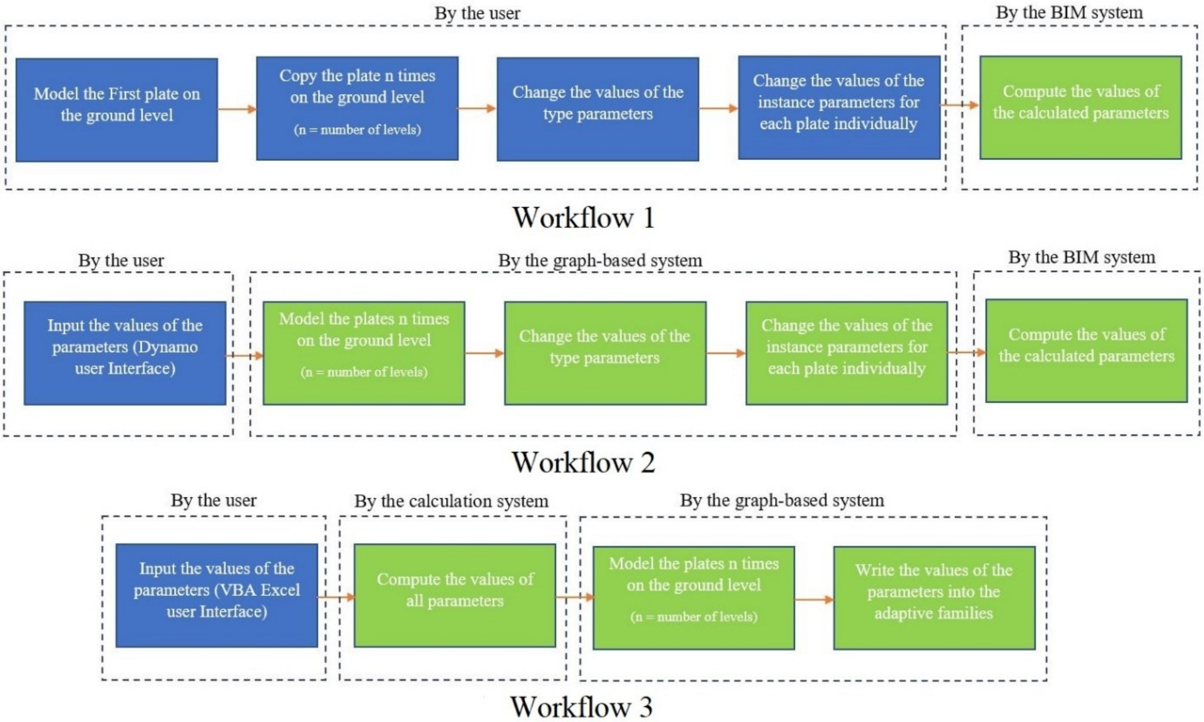


Figure 41: the steps of the three basic workflows used in the comparison

First, the comparison in terms of performance. As mentioned in section 3.3.1 the main two aspects that we can use to compare between the workflows are the processing time and the occupied internal and external memory.

- **Processing time**: the processing time is generally divided into two parts. Firstly the processing time needed to model and copying the model elements (plate adaptive families) and the processing time of computing the values of the calculated parameters. Workflow 1 has the largest processing time because the processing time needed for modeling and copying the elements manually is larger than doing this automatically through the graph-based system as done in workflow 2 and 3. On the other hand, it has been noticed that using an external computational system (as Workflow 3) leads to a decrease in the processing time of the computing of the values of the parameters than performing this calculation inside the BIM system (as Workflow 1 and 2). That is why Workflow 3 has the smallest processing time due to using a graph-based system and computational system Fig(42).

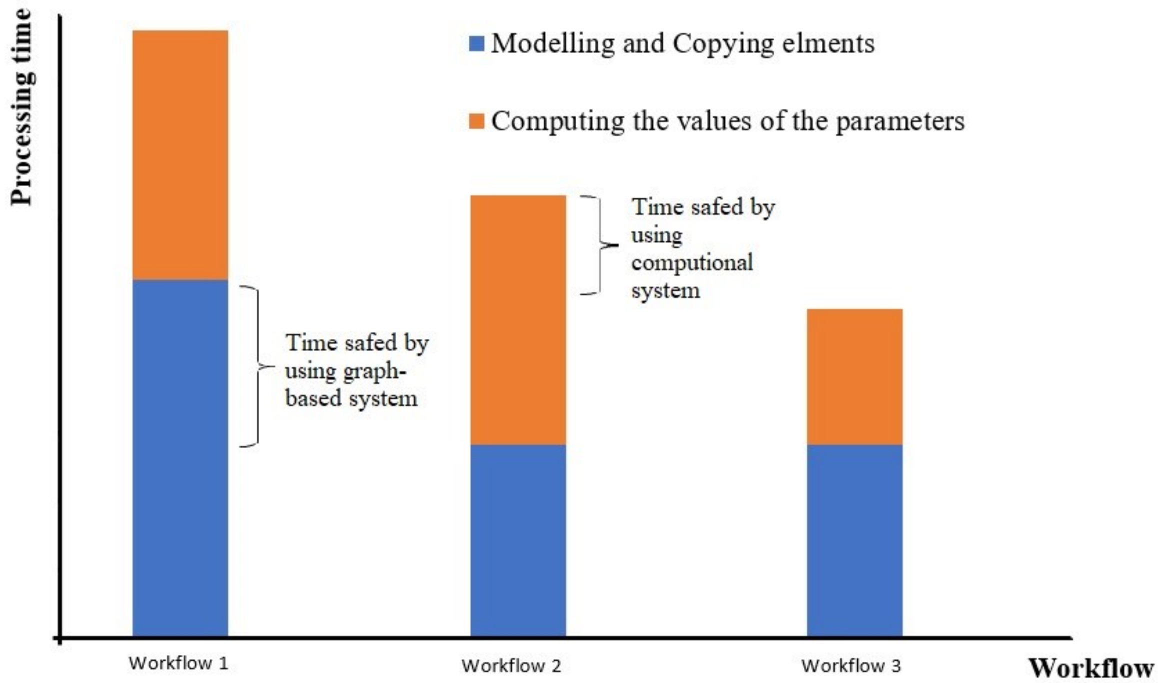


Figure 42: Graph shows the processing time of the different workflows

- **Occupied internal and external memory:** it has been noticed that there are no noticeable differences between the different workflows in terms of the occupied memory. However, it has been noticed that the parallelization of the processes in workflow 2 and 3 has decreased slightly the occupied internal memory than workflow 1.

Secondly, the comparison in terms of usability. As mentioned in section 3.3.2 the usability aspects that can be used to compare the workflows are as following :

- **Learnability**: it is obvious that workflows 2 and 3 are easier to follow than workflow 1 due to the fewer steps to be performed by the user. in workflow 1 the user has to perform a number of steps in a certain order, which may require a tutorial, while in workflow 2 and 3 the user only has to perform only one step, see Fig(41).
- **Efficiency:** with such a simplified task, there is no noticeable difference in functional variability between workflows, since all workflows offer the same degree of variability.
- **Satisfaction:** as mentioned in section 3.3.2, the satisfaction aspect is divided into 3 main points (user-friendly, attractive, and trustworthy for users) (Seffah, et al., 2006).
  - **User-friendly:** Workflow 2 and 3 used additional systems in addition to the BIM system, allowing us to program a user-friendly interface where the user enters the input data. On the other hand, Workflow 1 used the BIM system only, and the BIM system by nature does not have a straightforward interface, which is not user-friendly.

- **Trustworthy:** The clearer the calculation formula and the easier it is to follow, the higher the user's confidence in the tool. Since not every user can follow the formula in the programmed adaptive families, moving the calculation to a spreadsheet (as workflow 3) gives the user more confidence in the tool. where every user can follow the formulas and check if the program delivers a suitable result.

From the comparison, it is obvious, that using additional systems along with the BIM system (as Workflow 2 and 3) offers higher performance, more flexibility, and a better usability level. That is why the Author has decided to follow the concept of Workflow 3 in designing the detailed workflow for the Helical ramp task (complex task).

4. Methodology implementation

This chapter covers the implementation of the methodology discussed in Chapter 3 to design a detailed parametric BIM workflow to a parametric generation of complex geometry (the Helical ramp). The Benchmark task in chapter 3 is used as a guideline. However, the Helical ramp task is more complex. The programmed tools (the calculation model, Autodesk Revit families, and dynamo scripts) are discussed in detail.

4.1. The main concept and workflow

This section discusses the proposed workflow. The workflow follows a modified version of the coupled approach, with an explicit multi-operation iteration. The main idea is to use a graph-based system and a calculation system along with the BIM system to increase the performance and usability of the workflow. Autodesk Revit is used this task as a BIM system, Autodesk Dynamo as the graph-based system, and Microsoft Excel as the calculation system. Two dynamo scripts were written and one VBA Excel calculation model. The Workflow is divided into five main steps.

Firstly, starting with some steps in Autodesk Revit (BIM system). Then the first dynamo script will be executed to do some modeling and modifying tasks in addition to reading data from Revit and exporting them to the calculation model. Afterward, staring the steps in the VBA excel model, which is responsible for two main tasks. First, to collect some more data from the user through a user-friendly data input interface, secondly, is the mathematical computational model, which uses the collected data from Revit and from excel to calculate all the needed parameter values. Then the second dynamo script is run to export the calculated values by the calculation model to Revit again in addition to place some Revit families that are needed in making the shop drawings.

Fig(43) shows an interaction diagram for the detailed workflow with all the steps of modeling the Helical ramp. The boxes are marked with a frame shows how these steps will be done e.g. by the user, by the graph-based system, or by the calculation system. the boxes have two colors blue and green. The blue boxes are the steps that have been done by the user and the green boxes are the steps that have been done automatically either by the graph-based system or by the calculation system.

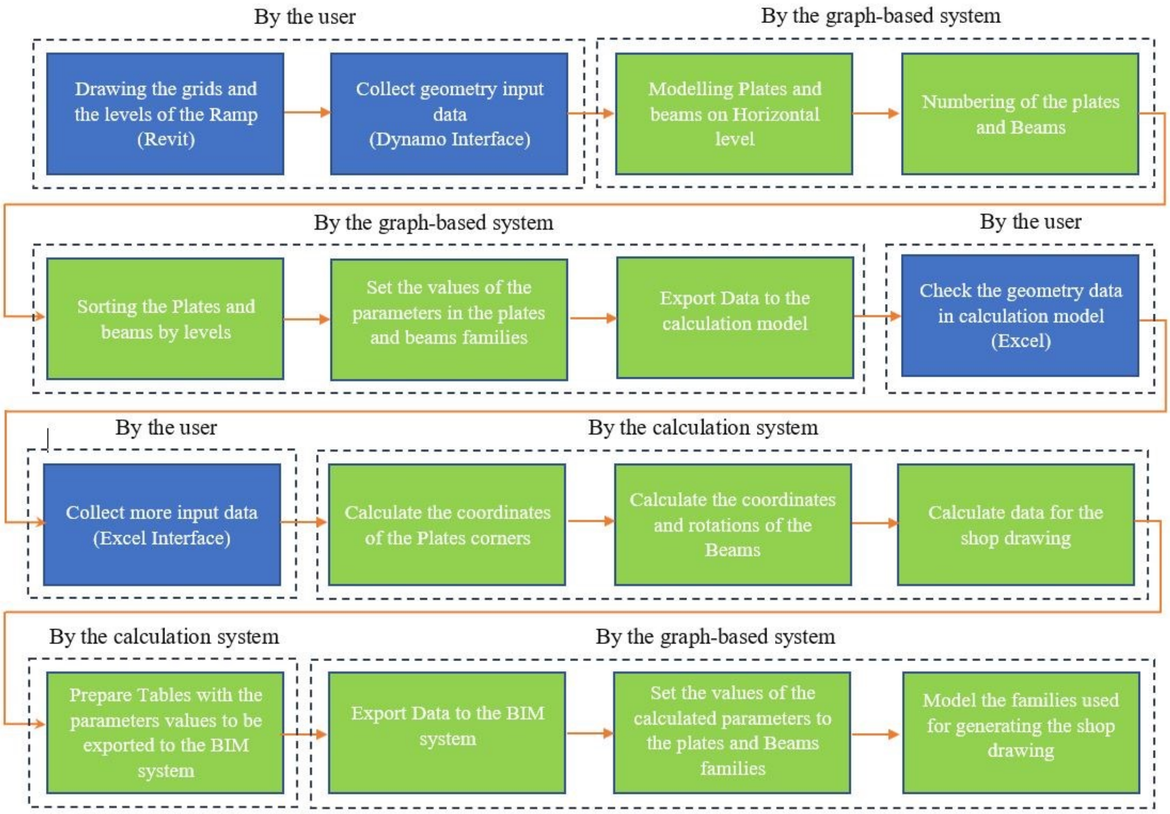


Figure 43: The Proposed workflow of modeling the Helical Ramp

Firstly the user has to draw the grids and the levels manually in the BIM system (Autodesk Revit), which will be used as a basis to read the geometry data Fig(45). After drawing the grids and levels and make the dimensions, the first dynamo script can be executed to collect the geometry data mentioned above Fig(44).

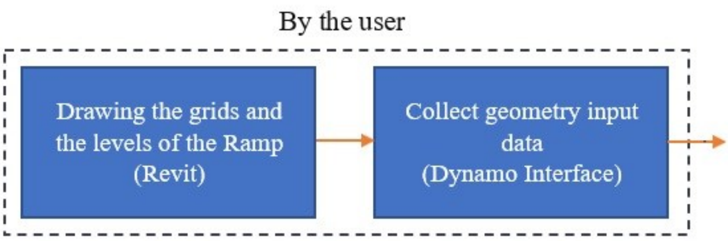


Figure 44: First part of the workflow

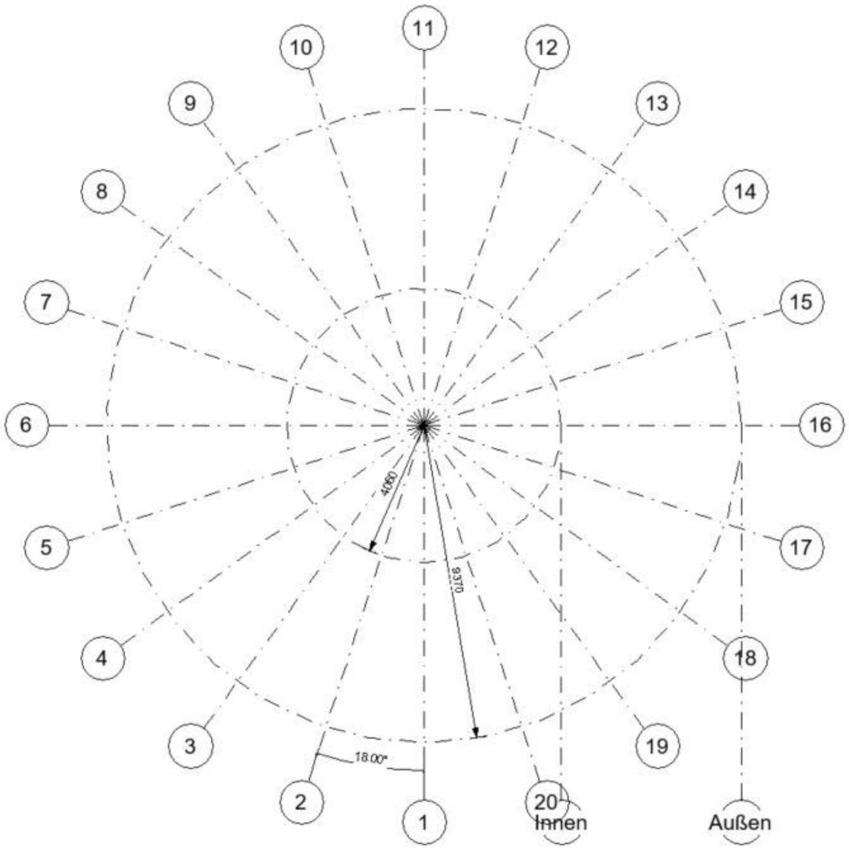


Figure 45: Grids of the Helical ramp as the first step in the workflow

Through the dynamo script and using the inputted data from the previous steps, the following five modeling and modifying steps will be done automatically Fig(46), these steps and the dynamo script will be discussed in detail in section 4.4. The script ends with exporting the data to the Excel calculation model.

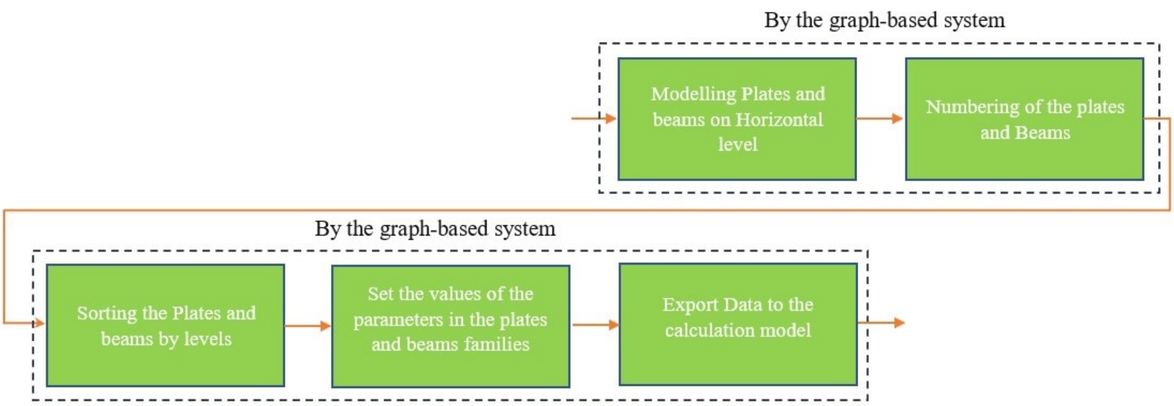


Figure 46: Second part of the Workflow

Then the user has to do some steps in Excel. Firstly, check the exported value from Revit. Secondly, enter some more input data and run the calculation model Fig(47).

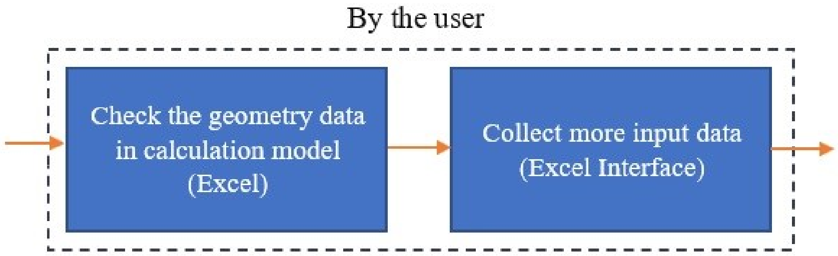


Figure 47: Third part of the workflow

Using the collected data from Revit and excel, perform the calculation model some mathematical calculations to prepare the parameter values to be exported again to Revit. The calculation model performs four main steps Fig(48). These steps will be discussed in detail in section 4.3.

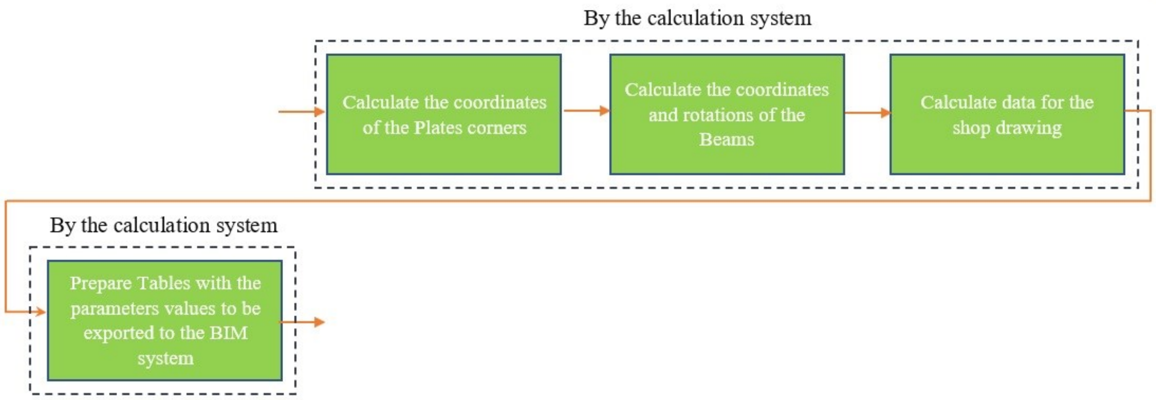


Figure 48: Fourth part of the workflow

The last part of the workflow is done using a graph-based system (Autodesk dynamo). The second dynamo script is responsible for exporting the already computed values by the calculation model in the previous part of the workflow to Revit and finalize the 3D geometry and prepare the shop drawings Fig(49).

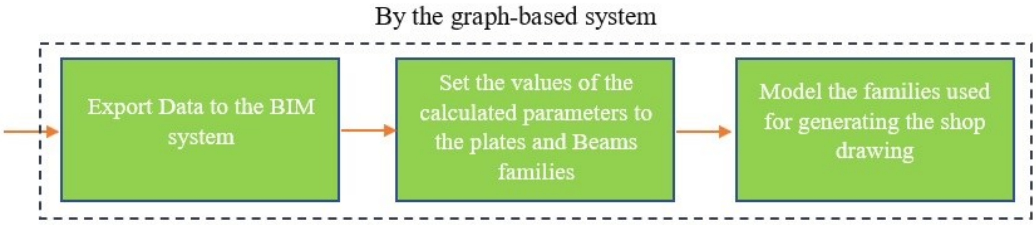


Figure 49: Fifth part of the workflow

Table 3: the required inputted values

	Input data	Unit	Type of input
1	Inner radius	mm	By selecting a dimension (Dynamo interface)
2	Outer radius	mm	By selecting a dimension (Dynamo interface)
3	Angel between grids	degree	By selecting a dimension (Dynamo interface)
4	Number of landings	---	By choosing from a dropdown menu (Dynamo interface)
5	Rotation of ramp to gable side	degree	By writing in a cell (Dynamo interface)
6	Starting grid of the ramp	---	By selecting a grid (Dynamo interface)
7	Number of levels	---	By choosing from a dropdown menu (Dynamo interface)
8	Elevations of levels	mm	By selecting the levels (Dynamo interface)
9	The direction of the ramp	CW/CCW	By choosing from a dropdown menu (Dynamo interface)
10	Columns height	mm	By writing in a cell (Dynamo interface)
11	Ramp’s grids number	Ascend/Descend	By graphically selecting (VBA Excel interface)
12	Type of ramp	Gable/Long	By graphically selecting (VBA Excel interface)
13	Position of the ramp	Right/left	By graphically selecting (VBA Excel interface)
14	The transversal slope of the ramp	%	By writing in a cell (VBA Excel interface)
15	The slope of the attached parking building	%	By writing in a cell (VBA Excel interface)
16	Position of the low point	Down/up	By graphically selecting (VBA Excel interface)
17	An additional offset of the ramp	mm	By writing in a cell (VBA Excel interface)

4.2. Concept of the Revit families

This section demonstrates the Revit families that are programmed in this study and used in the workflow. Four families for the 3D geometry have been created see Fig(50). In addition to three families for the 2D shop drawings. Table(4) shows those families and the used templates for each one.

Table 4: Families name and the used family template

	Family name	**Family template**
1	Ramp Plate	Generic Adaptive family
2	Ramp Landing 1	Generic Adaptive family
3	Ramp Landing 2	Generic Adaptive family
4	Ramp Beam	Generic Model family
5	2D Ramp plate	Generic Detail Item
6	2D Ramp Landing 1	Generic Detail Item
7	2D Ramp Landing 2	Generic Detail Item

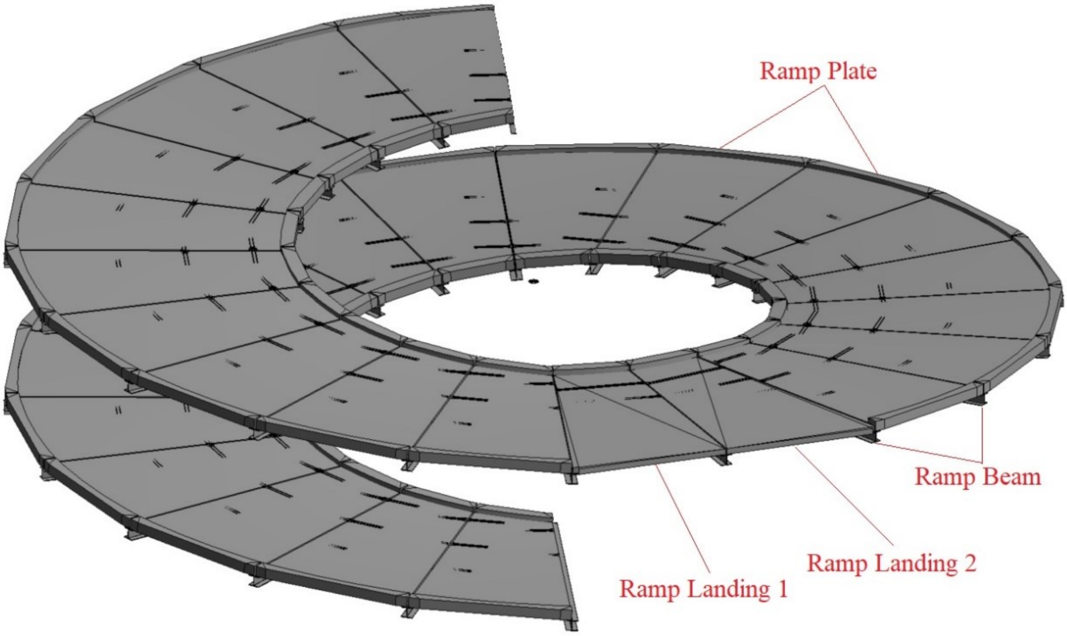


Figure 50: 3D geometry of the helical ramp shows the different types of families

- **Ramp plate family**

This family one of the main components of the helical ramp, which is repeated to form the shape of the helical ramp. The ramp plate has an approximately trapezium-shaped form. The plate has two thicknesses, one for the plate and the other for the sidewalk and the curb Fig(51). The family must also include small parts (Steel Connectors and Transport anchors), which are important for the production stage Fig(51). This family needs to be parameterized in such a way that with any change of input parameters, the dimensions of the plate are adopted automatically. Not only the dimensions in the horizontal plane need to be adjustable but also in the z-direction and also the elevation of the plate corners. That is why the adaptive family template has been chosen because it is the only template that allows the user to model points and parametrized their vertical offset.

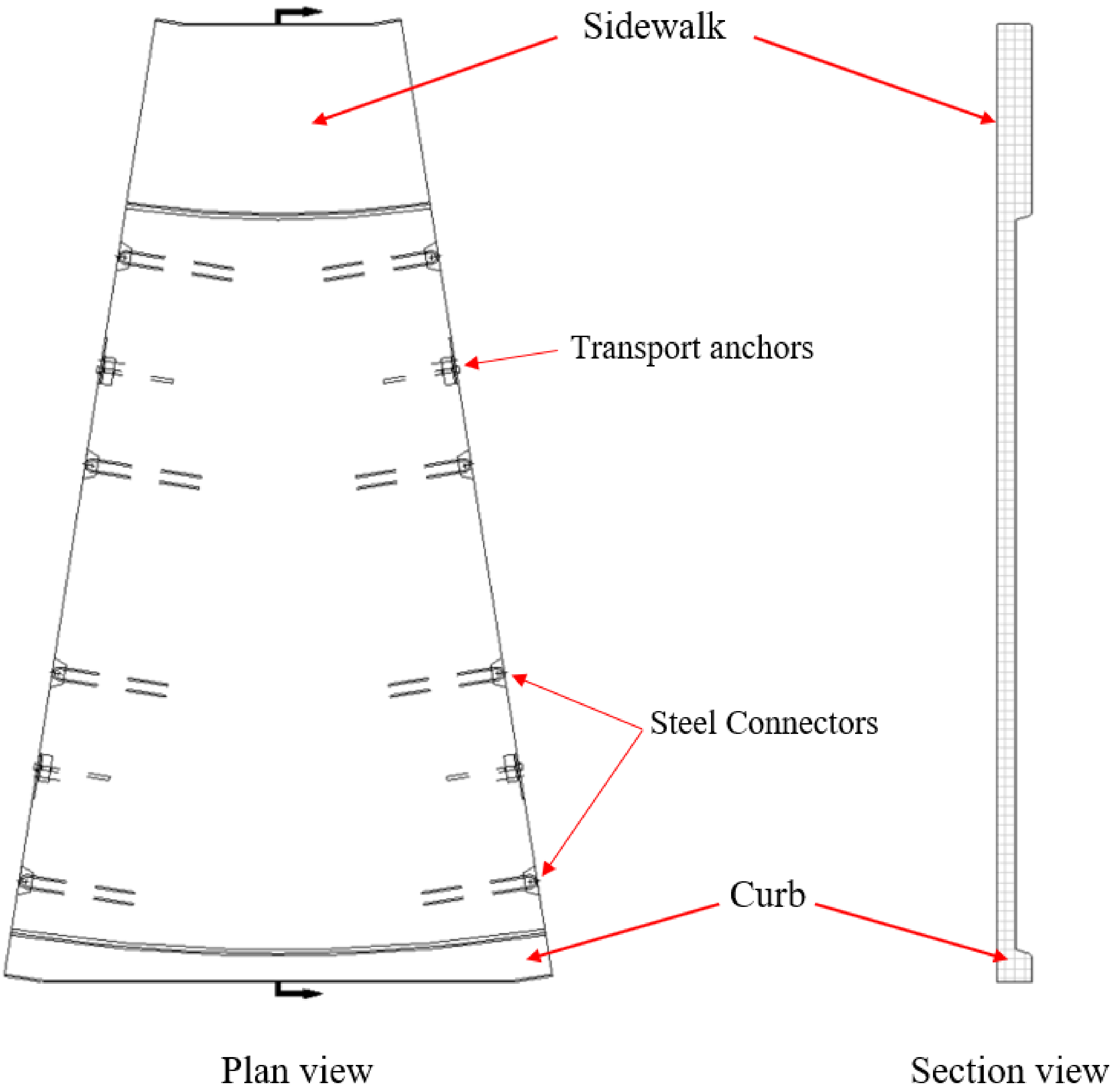


Figure 51: Plan view and section view of the ramp plate

The concept of the parameterization in this Family is to define an origin point with two origin reference planes that represent the center of the spiral ramp. Each point of the plate corners is modeled as an adaptive point and is locked with reference planes in the horizontal and vertical direction, whereby the point moves with these reference planes. These reference planes are controlled through parametric dimensions by the origin reference planes Fig(52-a). These parametric dimensions are calculated inside the family through formulas with respect to some geometrical parameters e.g Inner radius, Outer radius, etc. See Table (5). Then the adaptive points of the corners are connected with reference lines Fig(52-b). Afterward, the reference lines can be used as a basis to place all needed reference points. Then through these reference points, the geometry of the plate is generated Fig(52-c). The position of the reference points on the reference lines is controlled through a built-in parameter, which is parameterized in such a way to get the required outer shape of the plate and the sidewalks.

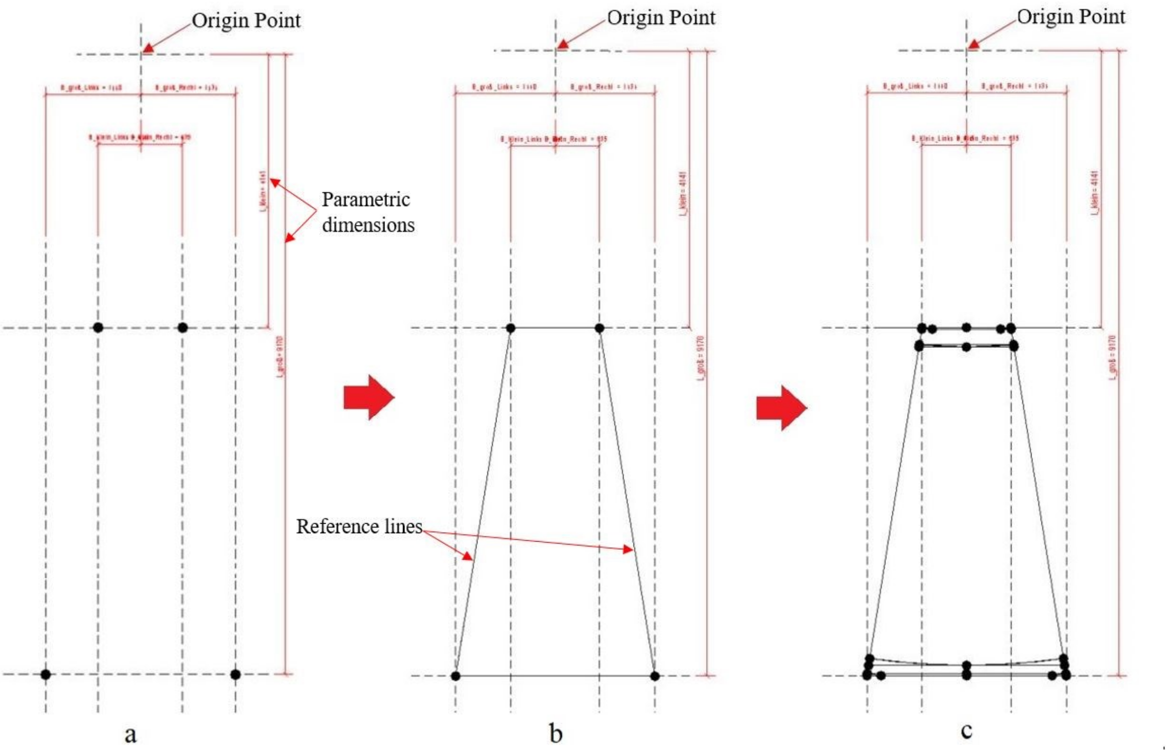


Figure 52: steps of creating the ramp plate family.

The parameters in this family can be divided into 4 categories as follow :

1. **Input parameters** are input parameters that needed to be used to calculate other values Fig(53) e.g. (Inner radius, Outer radius, Plate thickness, Number of fields, etc.). For the description of each parameter see Table(5).

Parameter	Wert	Formel
Abmessungen		
Anzahl Felder	20	=
Fuge_Außen	25,0	=25 mm
Fuge_Innen	35,0	=35 mm
Seitliches_Fuge	15,0	=15 mm
Dicke-Platte	103,0	=103 mm
Dicke_Aufbordung	183,0	=183 mm
Aufbordung_Außen_Breite	250,0	=
Aufbordung_Innen_Breite	250,0	=
Stütze_Höhe_Außen	190,0	=
Stütze_Höhe_Innen	190,0	=
R_Außen	9370,0	=
R_Innen	4060,0	=

Figure 53: Input parameters of the Ramp plate family

2. **Calculated parameters**: are parameters that use the input parameters or other calculated parameters in mathematical formulas to calculate some values to adjust the geometry Fig(54). For the description of each parameter see Table(5).

Parameter	Wert	Formel
Berechnungsmodell		
Winkel	18,00°	=360° / (Anzahl Felder)
L_Außen	9169,8	=(cos(Winkel / 2) * (R_Außen - Abstand_Außen + M1)) + (tan(Winkel / 2) * Seitliches_Fuge)
L_Innen	4140,8	=(cos(Winkel / 2) * (R_Innen + Abstand_Innen)) + (tan(Winkel / 2) * Seitliches_Fuge)
B_Innen_Links (Vorgabe)	646,0	=(tan((Winkel / 2) - Seitliche_Fuge_Winkel_Innen) * L_Innen) - X_Versatz_Innnen
B_Außen_Links (Vorgabe)	1440,3	=(tan((Winkel / 2) - Seitliche_Fuge_Winkel_Außen) * L_Außen) - X_Versatz_Außen
B_Außen_Recht (Vorgabe)	1433,9	=(tan((Winkel / 2) - Seitliche_Fuge_Winkel_Außen) * L_Außen) + X_Versatz_Außen
B_Innen_Recht (Vorgabe)	635,2	=(tan((Winkel / 2) - Seitliche_Fuge_Winkel_Innen) * L_Innen) + X_Versatz_Innnen
L_seite	15,2	=Seitliches_Fuge / (cos(Winkel / 2))
Abstand_Innen	130,0	=(Fuge_Innen) + (Stütze_Höhe_Innen / 2)
Abstand_Außen	120,0	=(Fuge_Außen) + (Stütze_Höhe_Außen / 2)
Seitliche_Fuge_Winkel_Außen	0,09°	=atan(Seitliches_Fuge / (R_Außen - Abstand_Außen))
Seitliche_Fuge_Winkel_Innen	0,21°	=atan(Seitliches_Fuge / (R_Innen + Abstand_Innen))
L_Mitte (Vorgabe)	4560,0	=(((L_Außen - L_Innen) / cos(Winkel / 2)) - M1) - (Aufbordung_Innen_Breite + Aufbordung_Außen_Breite)
Abstand_Schlaufe (Vorgabe)	1200,0	=if(L_Mitte > 3900 mm, 1200 mm, if(and(L_Mitte < 3900 mm, L_Mitte > 3700 mm), 1100 mm, if(and(L_Mitte < 3700 mm, L_Mitte > 3100 mm), 1000 mm, 900 mm)))
Erste_Schlaufe (Vorgabe)	730,0	=((L_Mitte / 2) + Aufbordung_Innen_Breite) - (Abstand_Schlaufe * 1,5)
Erste_Schlaufe_Ursprung (Vorgabe)	4861,8	=(cos(Winkel / 2) * Erste_Schlaufe) + L_Innen
Erste_schlaufe_H_L (Vorgabe)	760,2	=(sin(Winkel / 2) * Erste_Schlaufe) + B_Innen_Links
Erste_schlaufe_H_R (Vorgabe)	749,4	=(sin(Winkel / 2) * Erste_Schlaufe) + B_Innen_Recht
Abweichung_innen (Vorgabe)	-0,6	=((B_Außen_Recht + B_Außen_Links) * tan(Abweichung_Winkel) / 2)
Abweichung_Außen (Vorgabe)	-0,3	=((B_Innen_Links + B_Innen_Recht) * tan(Abweichung_Winkel) / 2)
Abweichung_Winkel (Vorgabe)	-0,03°	=(atan((X_Versatz_Innnen - X_Versatz_Außen) / (L_Außen - L_Innen)))
Y_Versatz_Links_Außen (Vorgabe)	9169,5	=L_Außen + Abweichung_Außen
Y_Versatz_Links_Innen (Vorgabe)	4140,2	=L_Innen + Abweichung_innen
Y_Versatz_Rechts_Außen (Vorgabe)	9170,1	=L_Außen - Abweichung_Außen
Y_Versatz_Rechts_Innen (Vorgabe)	4141,4	=L_Innen - Abweichung_innen

Figure 54: Calculated parameters of the Ramp plate family

3. **External calculated parameters:** are the parameters that are calculated in the calculation model, which they need a complex level of mathematical calculations. Therefore they are outsourced to be done outside the family to increase the performance of the family. Afterward, the results of the calculations will be imported directly to these parameters. For the description of each parameter see Table(5).

Parameter	Wert	Formel
Abmessungen		
Berechnungsmodell		
Bemessungsergebnisse		
Sonstige		
A1 (Vorgabe)	0,0	=
A2 (Vorgabe)	0,0	=
A3 (Vorgabe)	0,0	=
A4 (Vorgabe)	0,0	=
Erhöhung_A1 (Vorgabe)	0,000000	=
Erhöhung_A2 (Vorgabe)	0,000000	=
Erhöhung_A3 (Vorgabe)	0,000000	=
Erhöhung_A4 (Vorgabe)	0,000000	=
Höhe_S1 (Vorgabe)	0,000000	=
Höhe_S2 (Vorgabe)	0,000000	=
Höhe_S3 (Vorgabe)	0,000000	=
Höhe_S4 (Vorgabe)	0,000000	=
X_Versatz_Außen (Vorgabe)	0,0	=
X_Versatz_Innnen (Vorgabe)	0,0	=
Position (Vorgabe)	0	=
Ebene_Filter (Vorgabe)	0	=

Figure 55: External calculated parameters of the Ramp plate family

4. **Reporting parameters** are parameter types whose values are determined by a certain dimension in the family model. These parameters are used to extract the final dimensions of the plate after all other parameters have been modified Fig(56). For the description of each parameter see Table(5).

Parameter	Wert	Formel
Abmessungen		
Berechnungsmodell		
Bemessungsergebnisse		
Bericht_H_Innen (Bericht)	1281,3	=
Bericht_H_Außen (Bericht)	2864,3	=
Bericht_V_Links (Bericht)	5060,0	=
Bericht_V_Rechts (Bericht)	5060,0	=
Bericht_Diagonal_1 (Bericht)	5408,8	=
Bericht_Diagonal_2 (Bericht)	5412,2	=

Figure 56: Reporting parameters of the Ramp plate family

Table 5: All the parameters used in ramp plate family

	Parameter name	**Parameter type**	**Description**
1	Number_fields	Input Parameter	The number of fields of the Helical ramp.
2	R_Outer/R_Inner	Input Parameter	Parameters of the dimensions of the Helical ramp. One parameter for the inner radius and one for the outer radius
3	Thickness_Plate	Input Parameter	
4	Thickness_Sidewalk	Input Parameter	
5	Joint_Outer/ Joint_Inner	Input Parameter	The widths of the joints between the plate and the columns at the outer and the inner side.
6	Joint_Side	Input Parameter	The width of the joints between the plates
7	Outer/Inner column Height	Input Parameter	The column section height at the outer and the inner side.
8	Outer/Inner Sidewalk Width	Input Parameter	The width of the sidewalk at the outer and the inner side.
9	Angle	Calculated parameter	The angle between the fields of the helical ramp, which is calculated by 360° /Number of fields
10	L_Outer / L_Inner	Calculated parameter	The vertical distance between the center point of the helical ramp and the outer/inner edge of the plate.
11	B_Inner_Left/ B_Outer_Left/ B_Inner_Right/ B_Outer_Right	Calculated parameter	The Horizontal distance between the center point of the helical ramp and the left/right edge of the plate at the outer and the inner side.

12	L_side	Calculated parameter	The width of the side joint in a horizontal level
13	Spacing_Outer/ Spacing_Inner	Calculated parameter	The distance between the plate edges and the centerline of the columns at the outer and the inner side.
14	Joint_Side_Angle_ Outer/ Joint_Side_Angle_ Innen	Calculated parameter	The angle that needs to be subtracted from the main angle to form the side joint between the plates at the outer and the inner side.
15	L_Middle	Calculated parameter	The middle length of the plate without the length of the sidewalk and the curb
16	Spacing_Hooks	Calculated parameter	The spacing between the hooks of the steel connectors
17	First_Hook	Calculated parameter	The distance between the first hook and the edge of the plate
18	First_Hook_Origin	Calculated parameter	The vertical distance between the first hook and the center point of the helical ramp
19	First_Hook_H_Left/ First_Hook_H_Right	Calculated parameter	The Horizontal distance between the first hook and the center point of the helical ramp at the left/right side
20	Y_Offset_Left_Outer/ Y_Offset_Left_Inner/ Y_Offset_Right_Outer/ Y_Offset_Right_Inner	Calculated parameter	The required offset of the left/right plate edge in the y-direction to center the plate with the beam at the outer and inner side.
21	A1/A2/A3/A4	External calculated parameter	The Elevation of the four corners of the plate. Each plate has a unique elevation for each point to form the spiral shape of the ramp. These values are calculated external by the calculation model and then are exported into this parameter.
22	Increment_A1/ Increment_A2/ Increment_A3/ Increment_A4	External calculated parameter	The increment required to be made in the corners of the plate's formwork during manufacture to maintain the twisted shape of the plate. These values are also calculated in the calculation model.
23	Altitude_S1/ Altitude_S2/ Altitude_S3/ Altitude_S4	External calculated parameter	The altitude of each plate corner with respect to the other corners, which needed in the manufacture of the plate to maintain the twisted shape. These values are also calculated in the calculation model.

24	X_Offset_Outer/ X_Offset_Inner	External calculated parameter	The required offset of the plate's edge in the X direction to center the plate with the beam at the outer and the inner side. These values are also calculated in the calculation model.
25	Position	External calculated parameter	The numbering of the plate. Each plate is given a specific number to identify each plate in the import process afterward and to assign the correct values to each plate. These values are calculated in the first dynamo script see Fig(43).
26	Level_Filter	External calculated parameter	All the plates that belong to the same floor are given a specific number. This value along with the value of the position parameter is required to identify each plate and to assign the correct values to each plate. These values are calculated in the first dynamo script see Fig(43).
27	Report_H_Inner/ Report_H_Outer	Reporting parameters	The horizontal dimension between the right and left edges of the plate at the inner side / the outer side.
28	Report_V_Left/ Report_V_Right	Reporting parameters	The inclined Vertical dimension between the plate corners at the inner side and the outer side.
29	Report_Diagonal_1/ Report_Diagonal_2	Reporting parameters	The inclined diagonal dimensions between the plate corners.

- **Ramp landing 1/ Ramp landing 2**

The concept of these families is the same as the above-described ramp plate family but with a different geometry Fig(57). The main difference is the outer side of the plate, which in the landing families there is no curb.

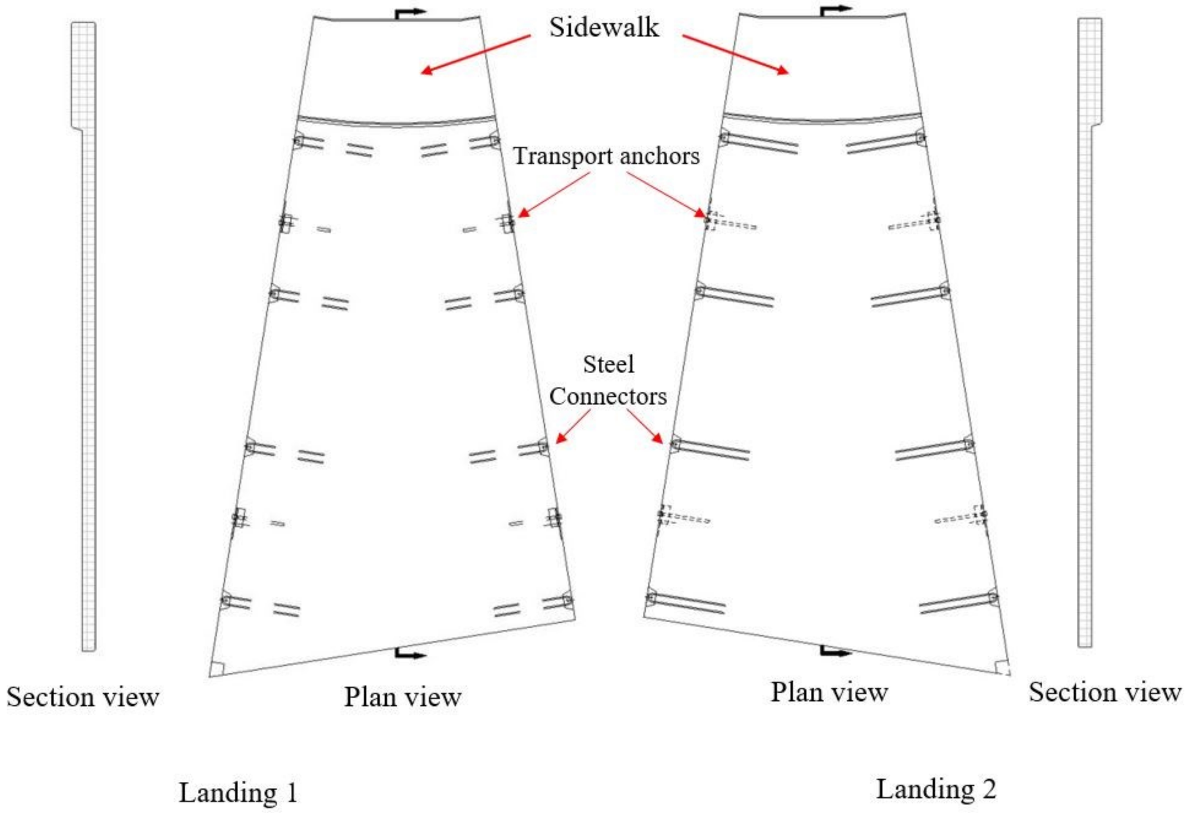


Figure 57: Plan views and section views of the Landings families.

As already mentioned, the concept of these families is the same as the already described family of ramp plates with the same parameter groups. therefore this will not be mentioned here again.

- **Ramp beam**

The Beam family is simpler than the plate families. As shown in Fig(50), each beam supports two plates. This family must be parameterized considering three main points. First, the length of the beam is automatically adopted whenever the input parameters are changed. Secondly, the elevation of the beam must be adjustable at both edges. Lastly, the rotation of the beam profile on both sides must be adjustable. As shown in Fig(58) each beam side must be able to have a separate rotation, in which the beam has a twisted shape.

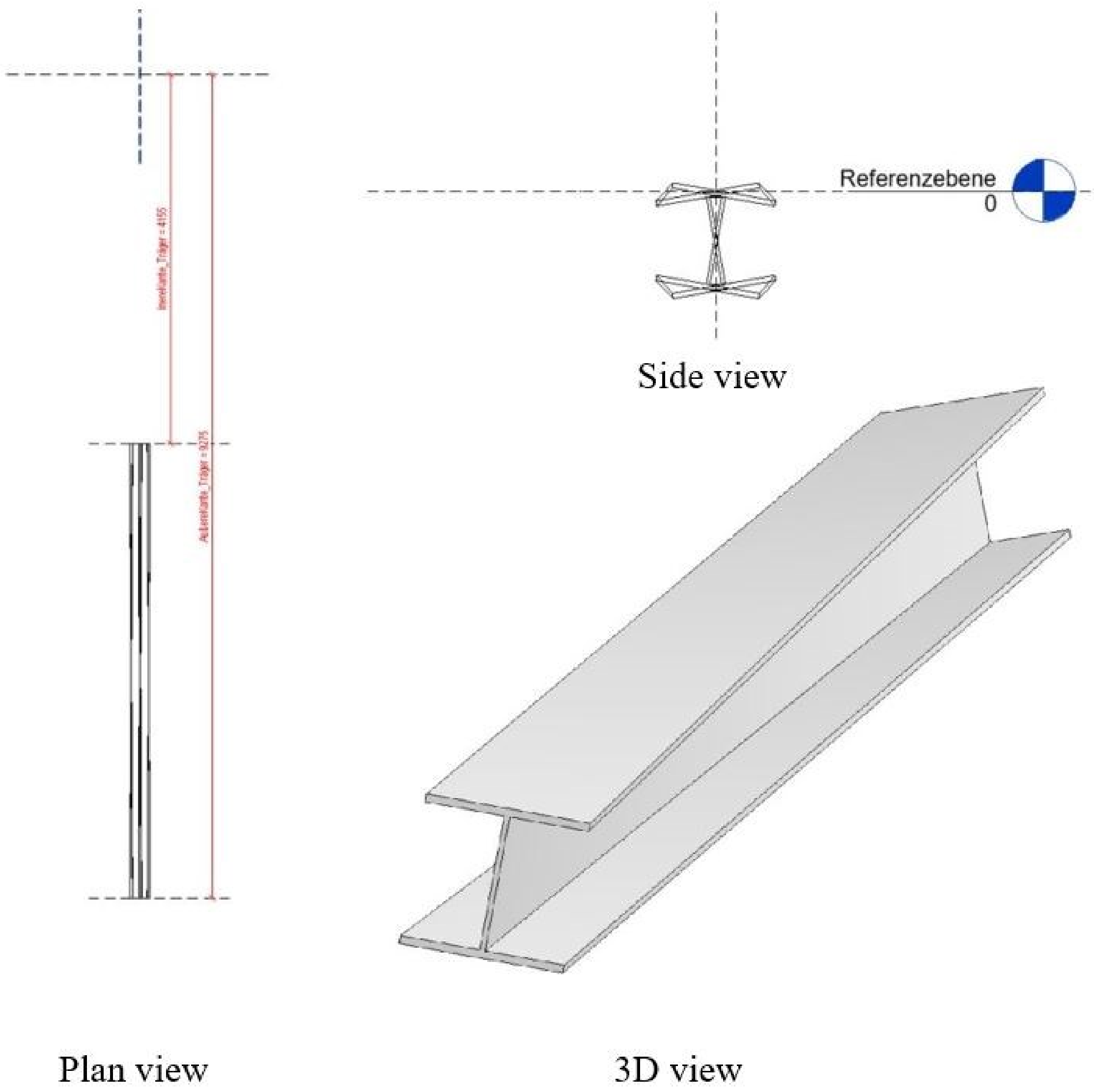


Figure 58: The Beam family from different views.

The concept of the parameterization in this Family is to define an origin point with two origin reference planes that represent the center of the spiral ramp. Then, two horizontal reference planes are modeled for the start and end of the beam see Fig(58). At each one of these reference planes, a profile is added and locked with it. A sweep between these two profiles is generated to form the geometry of the beam. A rotation parameter and z-direction offset parameter are assigned to each profile separately Fig(59).

Profil 1	
Profil	HEA200-neu : HEA200-neu
Horizontaler Profilversatz	0,0
Vertikaler Profilversatz	0,0
Winkel	0,00°
Profil ist gespiegelt	☐
Profil 2	
Profil	HEA200-neu : HEA200-neu
Horizontaler Profilversatz	0,0
Vertikaler Profilversatz	0,0
Winkel	0,00°
Profil ist gespiegelt	☐

Figure 59: Assigning rotation and vertical offset parameters to the beam profiles.

The parameters in this family can be divided into 3 categories as follow :

1. **Input parameters** are input parameters that needed to be used to calculate other values Fig(60) e.g. (Inner radius, Outer radius, Beam height, etc.). For the description of each parameter see Table(6).

Parameter	Wert
Abmessungen	
R_Außen	9370,0
R_Innen	4060,0
Stütze_Höhe_Außen	190,0
Stütze_Höhe_Innen	190,0
Träger_Höhe (Vorgabe)	190,0

Figure 60: Input parameters of the Ramp plate family

2. **Calculated parameters**: are parameters that use the input parameters or other calculated parameters in mathematical formulas to calculate some values to adjust the geometry Fig(61). For the description of each parameter see Table(6).

Parameter	Wert	
Abmessungen		
Berechnungsmodell		
AußereKante_Träger (Vorgabe)	9275,0	=R_Außen - (Stütze_Höhe_Außen / 2) - Letzte_Träger_L
InereKante_Träger (Vorgabe)	4155,0	=R_Innen + (Stütze_Höhe_Innen / 2)
Z-Versatz-AußenKreis (Vorgabe)	-95,0	=Z_Versatz_Außen_Überflansch - (Träger_Höhe / 2)
Z-Versatz-InneKreis (Vorgabe)	-95,0	=Z_Versatz_Innen_Überflansch - (Träger_Höhe / 2)

Figure 61: Calculated parameters of the Ramp plate family

1. **External calculated Parameters:** are the parameters that are calculated in the calculation model, which they need a complex level of mathematical calculations.

Therefore they are outsourced to be done outside the family to increase the performance of the family. Afterward, the results of the calculations will be imported directly to these parameters. For the description of each parameter see Table(6).

Parameter	Wert
Abmessungen	
Berechnungsmodell	
Sonstige	
Querschnittsdrehung-Außenkreis (Vorgabe)	0,00°
Querschnittsdrehung-Innenkreis (Vorgabe)	0,00°
Z_Versatz_Außen_Überflansch (Vorgabe)	0,0
Z_Versatz_Innen_Überflansch (Vorgabe)	0,0
Ebene_Filter (Vorgabe)	0
Position (Vorgabe)	0
Letzte_Träger_L (Vorgabe)	0,0

Figure 62: External calculated parameters of the Ramp plate family

Table 6: All the parameters used in ramp plate family

	Parameter name	**Parameter type**	**Description**
1	R_Outer/R_Inner	Input Parameter	Parameters of the dimensions of the Helical ramp. One parameter for the inner radius and one for the outer radius
2	Outer_column_Height/ Inner_column_Height	Input Parameter	The column section height at the outer and the inner side.
3	Beam_Height	Input Parameter	Parameter of the height of the beam profile.
4	Outeredge_beam	Calculated parameter	The distance between the origin point and the outer edge of the beam.
5	Inneredge_Girder	Calculated parameter	The distance between the origin point and the inner edge of the beam.
6	Z_offset_Outer/ Z_offset_Inner/	Calculated parameter	The required offset in the z-direction from the reference level, which is always equal the half-length of the beam-height plus the value calculated by the calculation model see Fig(58).
7	Cross_section_rotation_ Outer/ Cross_section_rotation_ Inner	External calculated parameter	The rotation of the beam cross-section, in which the positive value is rotation in clockwise and the negative value is counterclockwise. One parameter for the inner side and one for the outer side. These values are calculated external by the calculation model and then are exported into this parameter.

	Z_Offset_ upperflange _Outer/ Z_Offset_ upperflange _Inner	External calculated parameter	The elevation of the upper flange at the edge of the beam. Each beam has unique values of the elevations at the outer and inner edge, where they can build the spiral shape of the ramp.
8	Position	External calculated parameter	The numbering of the beam. Each beam is given a specific number to help identify each beam during the subsequent import process and to assign the correct values to each plate. These values are calculated in the first dynamo script, see Fig(43).
9	Level_Filter	External calculated parameter	All the beams that belong to the same floor are given a specific number. This value along with the value of the position parameter is required to identify each beam and to assign the correct values to each plate. These values are calculated in the first dynamo script see Fig(43).

- **2D Ramp plate / 2D Ramp Landing 1 / 2D Ramp Landing 2**

  To obtain high-performance families as well as comprehensive workshop drawings, the idea is to model the 3D plate without all the geometrical details and create a 2D family with all the geometrical details. This family reads the final dimensions from the 3D plate family and adapts itself accordingly.

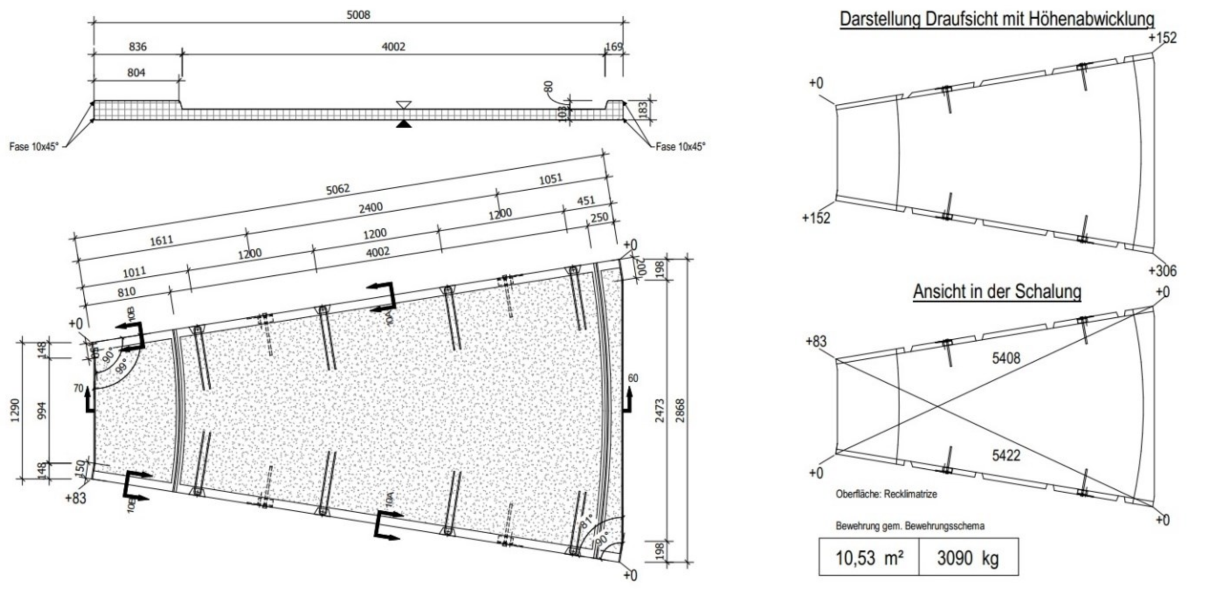


Figure 63: The shop drawing done by the 2D family.

All report parameters in the 3D families are exported to the 2D family to obtain the correct final shop drawing to be used in manufacturing. This is done for the ramp plate and the landing plate families.

4.3. The structure of the graph-based algorithm via Dynamo

Before we go into the details of the written Dynamo scripts used in this study, it is necessary to explain the basics of the Dynamo program.

- Basics of dynamo

Dynamo is a graphical programming environment. where the user can visually making a script by defining custom pieces of logic and connect them in such a way to define sequences of actions that compose the required algorithm Fig(64). These algorithms can be used for a wide range of applications starting from processing data to generating geometry.

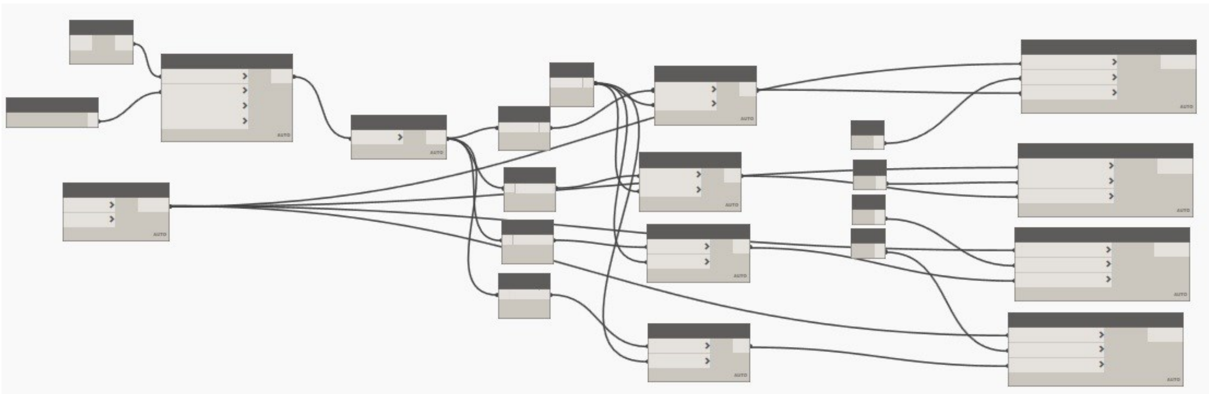


Figure 64: Example of the shape of a dynamo script.

These pieces of logic are called nodes. Each node runs a specific operation, this operation may be a simple one like storing a number or it might be complex like generating a specific geometry. Each node mostly has four main parts as shown in Fig(65). The first part shows the name of the node and mostly this describes the function of the node. The second part is the input required for the operation of the node. These inputs come from other nodes by connecting the two nodes with wires. The third part is the node output, which is the result of the node operation and could be used as an input for another node. The fourth part is showing the output list of the node operation.

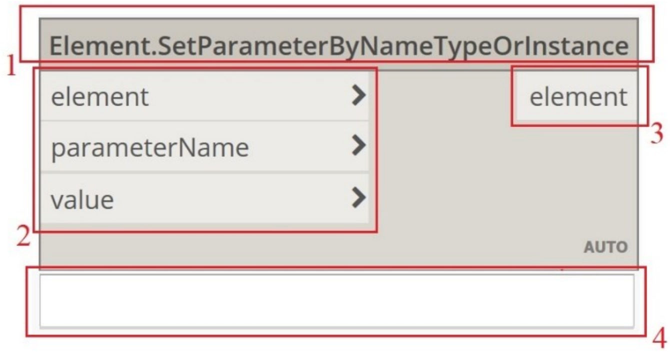


Figure 65: Typical node in dynamo.

There is a wide range of libraries and packages in dynamo to search in. the dynamo libraries are on the left side of the dynamo interface with a search cell to search in Fig(66). There, one can find the basic nodes that come with the default installation of Dynamo, the added node by the package manager, and the custom nodes created by the user. To find a suitable node for the needed purpose, the searching criteria in dynamo libraries have to be clear. One can search by the keyword, but this will return a large number of results. One can also search by the library hierarchy (library.category.nodeName or category.nodeName), which will return with appropriate results for the purpose.

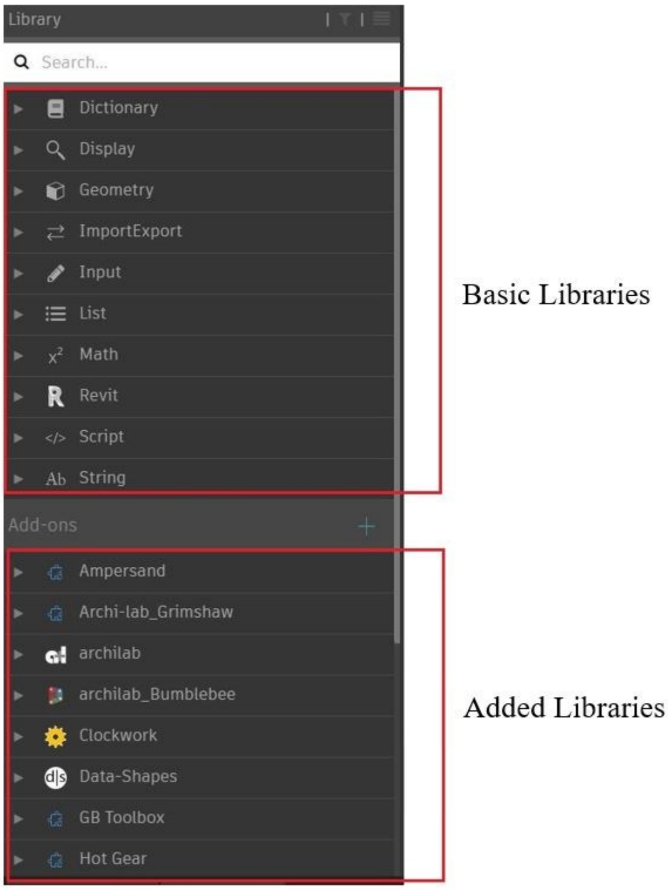


Figure 66: Libraries tap in dynamo.

As mentioned above in addition to the basic nodes and the added nodes, Dynamo offers the ability to program custom nodes, which has many advantages e.g.

1- Cleaning the script by adding more than one node in only one node
2- Fast adapting the changes if we have multiple copies of the custom node in the script.

- Basics of Python in dynamo

In addition to the custom node, Dynamo offers a very powerful and flexible feature, by allowing to create custom node by writing a python script node. Python is a programming language that is so popular because of its syntax style. It is well-readable and easy to learn. Python is supporting packages and modules and can be implemented into existing applications. One of the most important advantages of graphical programming is that the user can create a program without learning a syntax programming language. However, it has also some limits or missing functions that exist in the textual programming e.g. looping, conditional statements(if/then), and advanced mathematical operations. In this study, the python nodes are used to extend the functionality of Dynamo.

The Python node is similar to the other nodes in Dynamo, which are scripted in the graphical programming environment. It can be found in the library section Fig(66) under the category "Core".

To illustrate how Dynamo and Python work together, a short Python code is explained line by line to give an insight into how a node can be created with Python code.

The function of the custom created python node is to remove some walls by identifying their lengths. In this Revit project, there are some walls with different lengths Fig(68). The script will delete all the walls with lengths longer than 4 meters and save their id in a list. This all is done using a custom python node Fig(67).

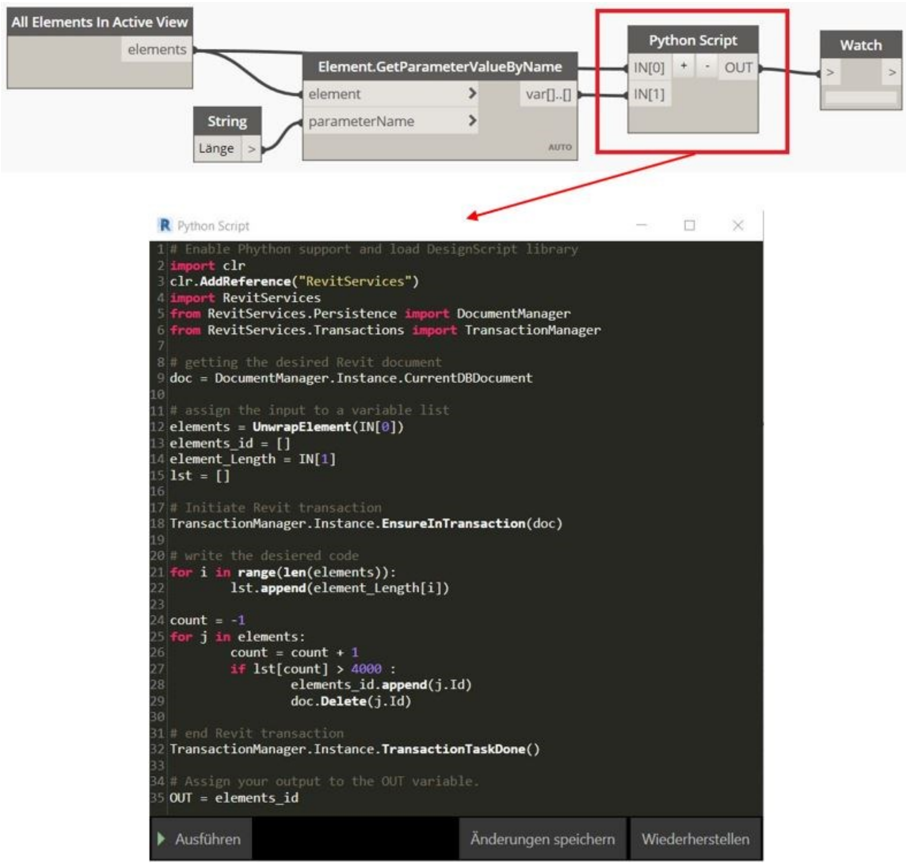


Figure 67: dynamo script with python custom node.

- Line 2 to line 6 :
  From line 2 to line 6 the required libraries are imported that are required in this script. Line 2 imports **CLR**, which stands for "Common Language Runtime. Microsoft has set up a mediator to convert the standard code of the CLI specification to MS machine code. This requires several steps in the .NET Framework.

Line 3 and 4 imports **clr.AddReference("RevitServices").** This is a method defined in the CLR to load in the Revit service libraries.
Line 5 imports the **DocumentManager**, which is an existing database referred to as the .rvt document where we save our models.
Line 6 imports the **TransactionManager.** Any change to a document can only be done inside an open transaction for the document. The changes become part of the document only when the active transaction is committed and these changes can be undone by the transaction's destructor.

- Line 9 assigns the active document to a variable called "doc". The active document is retrieved via the document manager.

- Line 12 to 15 :
From line 12 to line 15 the arrays are defined and assigned to the input values.

Line 12 assigns the wall elements to a list called "elements". Besides, these input elements have to be unwrapped. UnwrapElement in Python allows the element to be accessed directly in the Revit API, by passing through the Python interpreter. Wrapped elements are located in the namespace Revit.Elements. All wrapped elements extend the abstract class Revit.Elements.Element. This class provides a public property InternalElement, which contains a reference to the underlying RevitAPI element of type Autodesk.Revit.DB.Element.
Line 13 declares an empty list for the id of the element to save the data in it.
Line 14 assigns the input values of the lengths of the walls to a list called "element_length"
Line 15 declares a help empty list.

- Line 18 opens a transaction for the active document "doc" using the transaction manager

- Line 21 to line 29:
The main code of the node is written in this part of the script.
Line 21 starts a For loop. A for loop is to execute an operation with a specific number of times, which is equal to the number of elements in this case.
Line 22 appends the values of the lengths of the walls to the predefined empty list "lst".
Line 24 and line 26 creating a counter variable "count".
Line 25 starts another For loop.
Line 27 is an if condition to find the walls longer than 4 meters.
Line 28 save the id values of the walls to be deleted into the predefined list.
Line 29 executes the delete operation using the Revit API delete method. Revit API offers a great number of operations, that can be found in (Revit, 2020).

Line 31 closes the transaction that has been opened before we executed the delete operation. If a transaction has been started and is not yet completed, the standard destructor will automatically close it. It is not recommended to rely on the default behavior.

Line 34 defines the required output of the node by assigning it to the “OUT” variable, which is the list of the id values of the deleted walls.

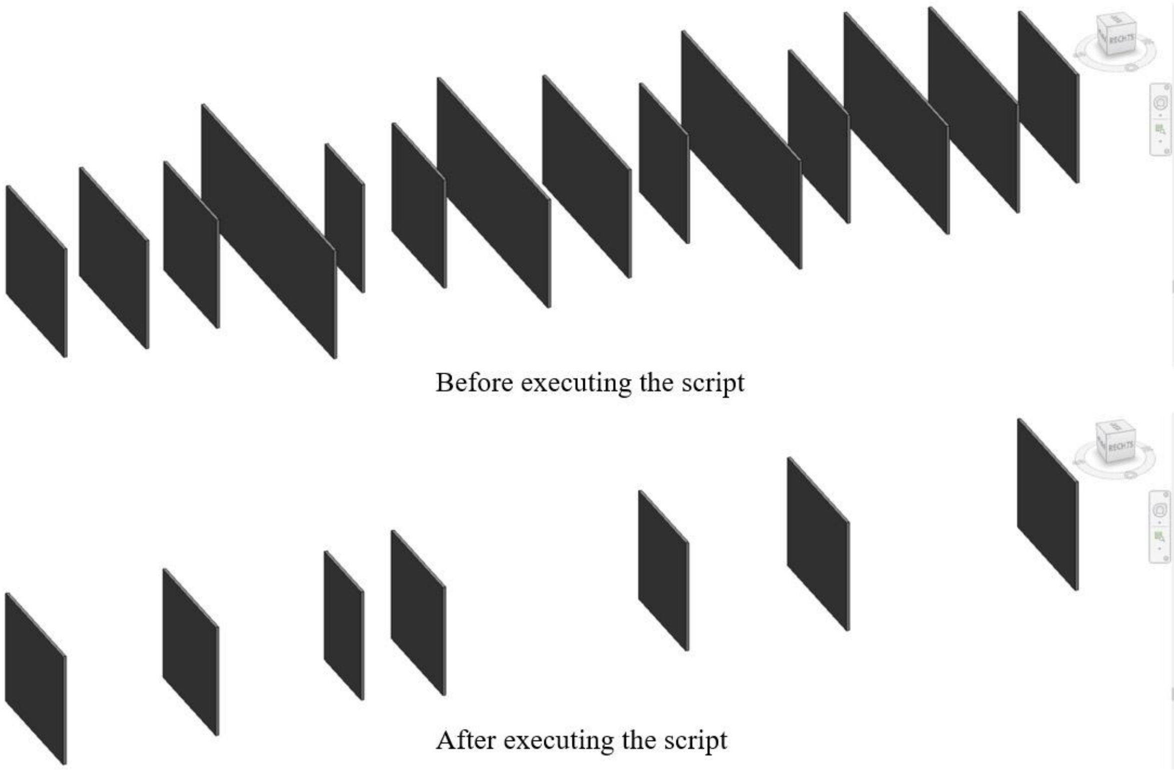


Figure 68: 3d view shows the views before and after executing the explained script.

- Script 1

The main function of the first script is to model the element at a horizontal level, numbering them, sorting them by level, and set the values of the parameters collected in from the dynamo interface. Afterward, export all the data to the calculation model Fig(69). In this section, the main nodes used for each function will be described[1].

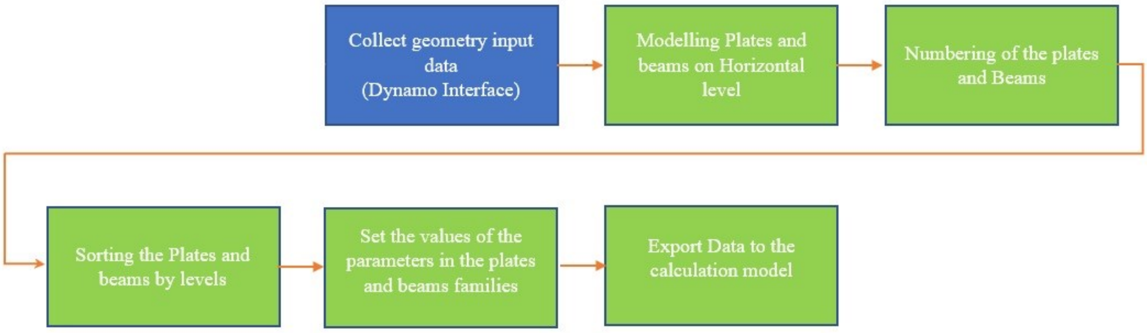


Figure 69: Tasks of script 1.

1. Collect geometry input data: in this task, the "data shape" package has been used to design the interface. This package offers a variant option to input the data in the interface e.g. (by selecting the model element, by entering a text value, by creating a dropdown list, etc.) Fig(70).

   The main nodes to build this interface are as shown in Fig(71). The data collector nodes and the input form node, however, the input form node used in this script is a modified version of the one from the data shape package. The python script is been modified in order to get the desired interface design on the interface. The "list create" node is only required when more than one input in the interface is required.

[1]This section descripe only the main nodes needed, for the complete code check the dynamo script.

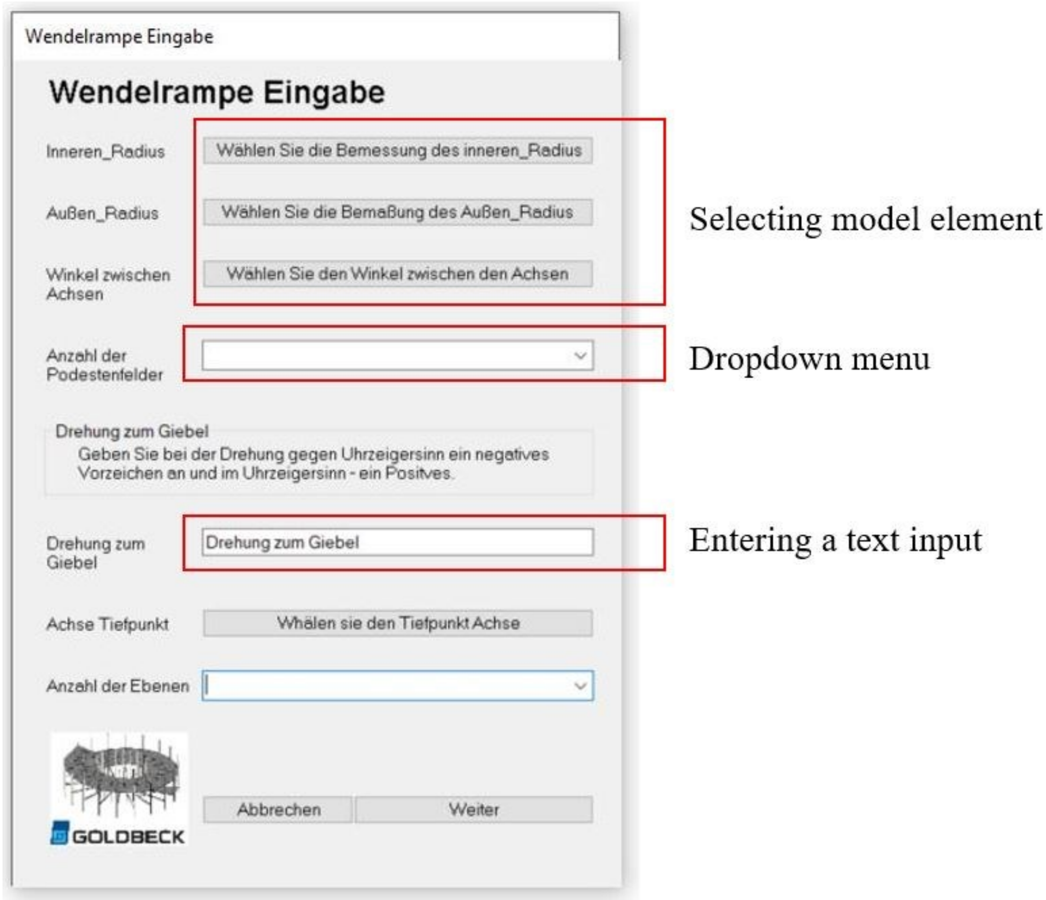


Figure 70: Interface with dynamo using the data shape package.

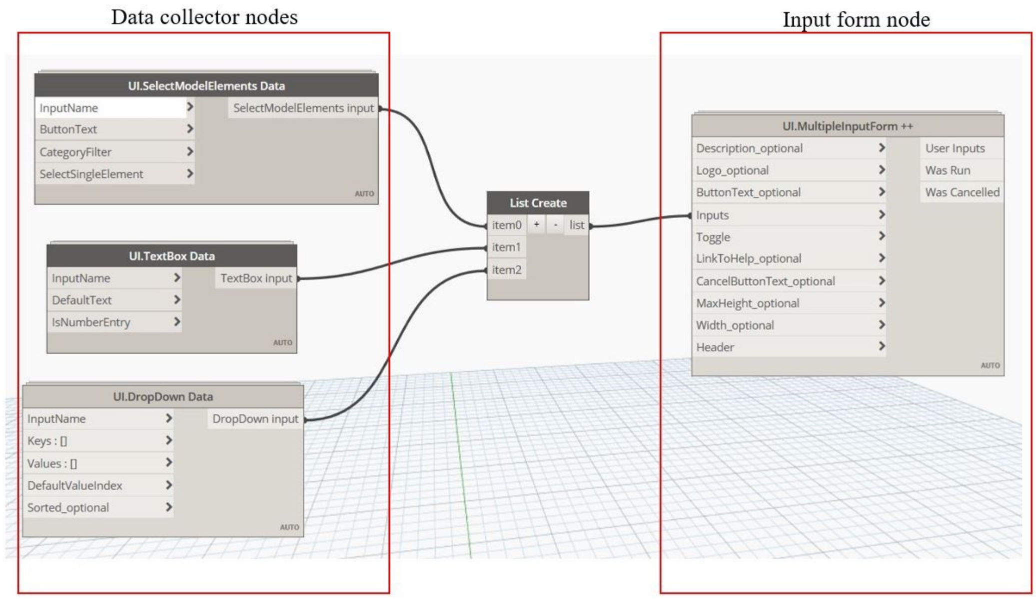


Figure 71: The main nodes of building an interface in dynamo.

2. Modeling plates and beams on a horizontal level: in this task there three main nodes are used. Firstly node to identify the place of the first plate/beam. Secondly, the node to place the first plate/beam at all levels. Finally to copy and rotate the first plate/beam to place the rest of the plates/beams[1] Fig(72).

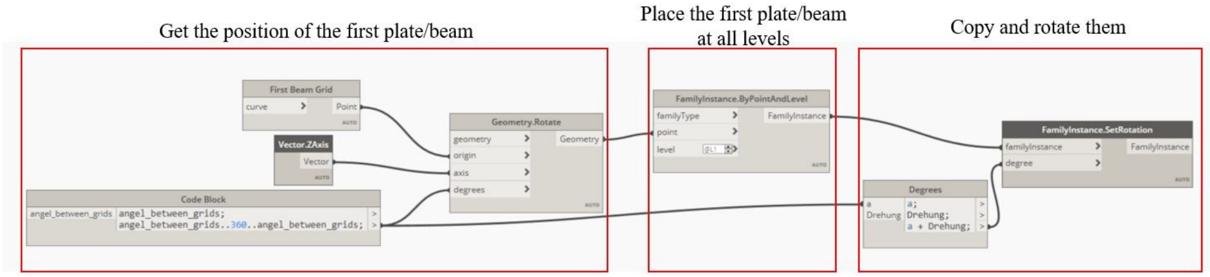


Figure 72: the main nodes of the plates/beams of the helical ramp.

3. Numbering the plates/beams: The numbering of the plates/Beam is a very important step. Each plate is given a specific number to identify each plate in the import process afterward and to assign the correct values to each plate. These values are written in a parameter called "Position" see table 4. The dynamo script of numbering is quite complex, which is written to make it flexible for the user to start the ramp at any place.[2]
4. Sorting the plates by level: All the plates/beams that belong to the same floor are given a specific number. This value along with the value of the position parameter is required to identify each plate/beam and to assign the correct values to each plate. By using the code block we can generate a list from 1 to the number of floors as shown in Fig(73) using the syntax (First number..Last number..Increment rate). Then assign this to the elements by setting these values to the parameter Level_Filter see Table 5.

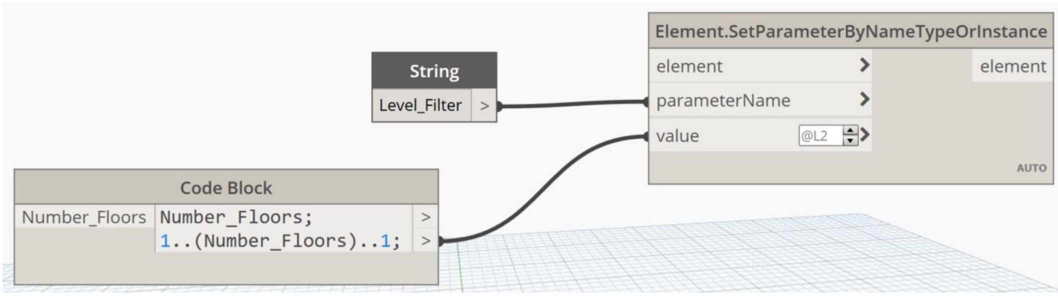


Figure 73: sorting the plates/beams by level.

5. Set the values of the parameters into the Plate/Beam families: using the same node "setparamterByName" is the best way to set a value of a parameter using dynamo either type or instance parameter. All parameters collected by the dynamo interface (see Table 3) have been set directly into the families in this step, e.g. inner radius, outer radius, etc. Fig(74).

[1] This method is done for both the beams and plates
[2] For the whole code of this part please check the dynamo script

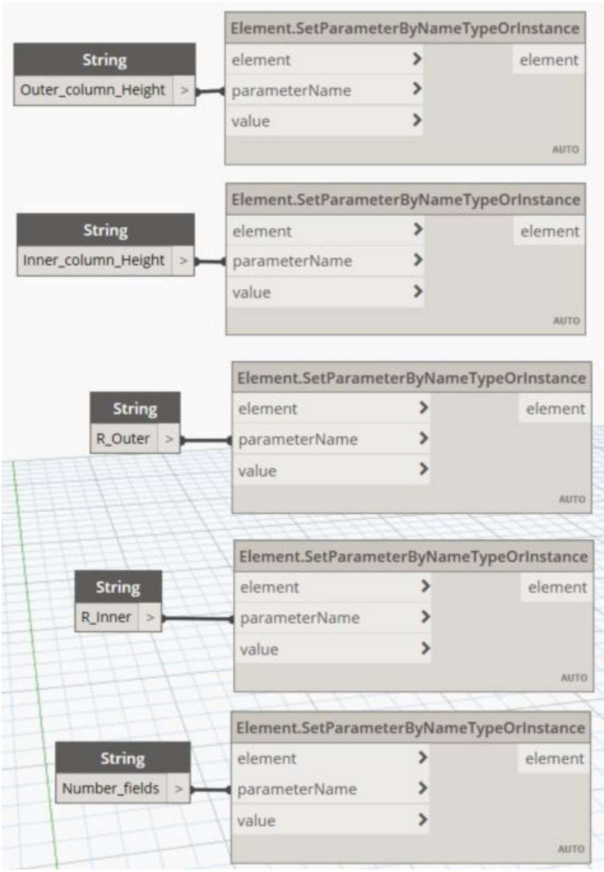


Figure 74: Example of setting the values of the plate/beam parameter values in dynamo.

6. <u>Data Export to the calculation model:</u> the main node used for exporting the data to excel is from the basic dynamo package "ImportExport" and the node name is "ExportExcel". As shown in Fig(75), we choose the sheet name and the position of the first cell in Excel, then the data to be exported, which we assemble in a list.

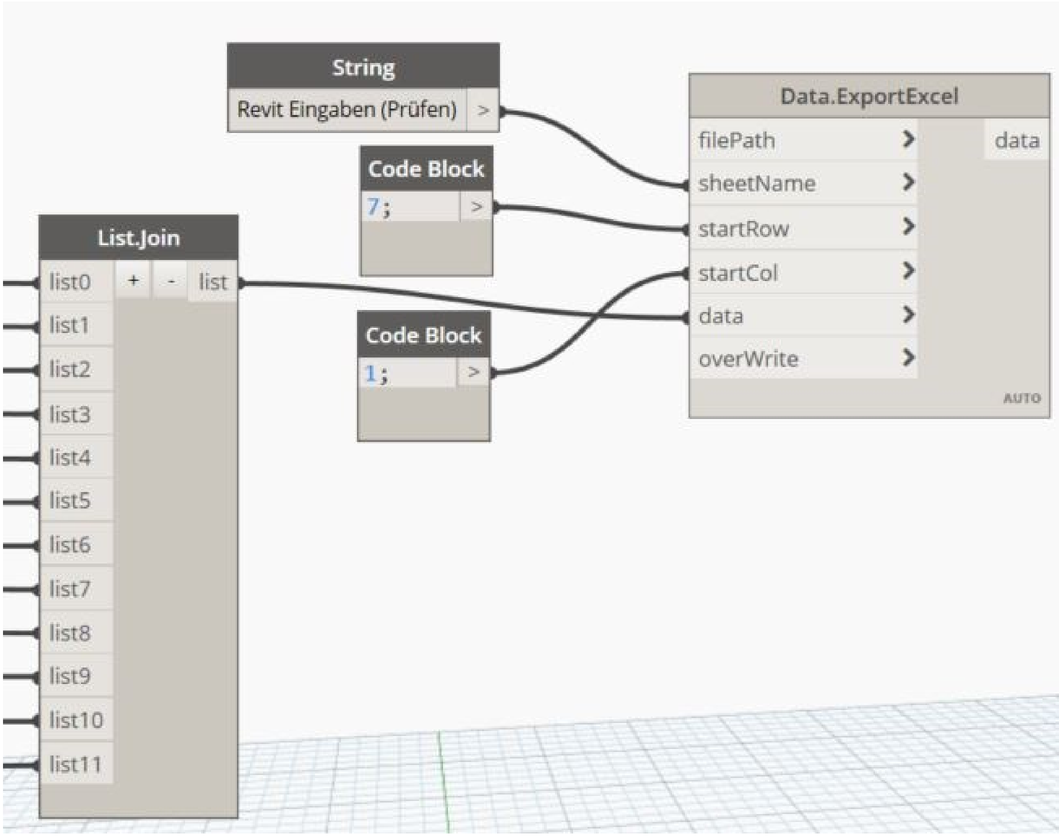


Figure 75: Example of exporting data to excel using dynamo.

- Script 2

The main functions of the second script are to export the data from the calculation model to the BIM system, set the values of the parameters calculated by the calculation model, and model the 2d families used for the shop drawings Fig(76).

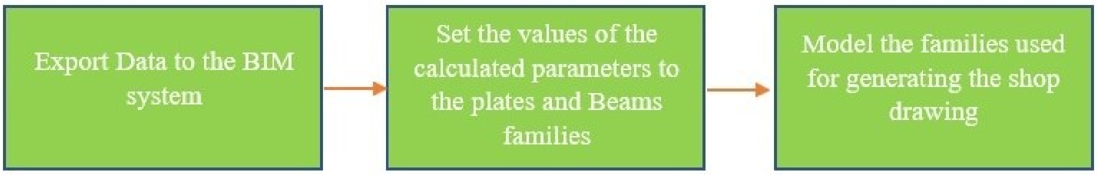


Figure 76: Tasks of script 2.

1. <u>Export data to the BIM system:</u> to read the data from the calculation model we need to use again a node from the “ImportExport” basic dynamo package called “ImporExcel”. The input of this node is the file path of the excel file and the sheet name Fig(77). The output of this node needs to be filtered and sorted in a such way to be able to write it in the plate/beam families.

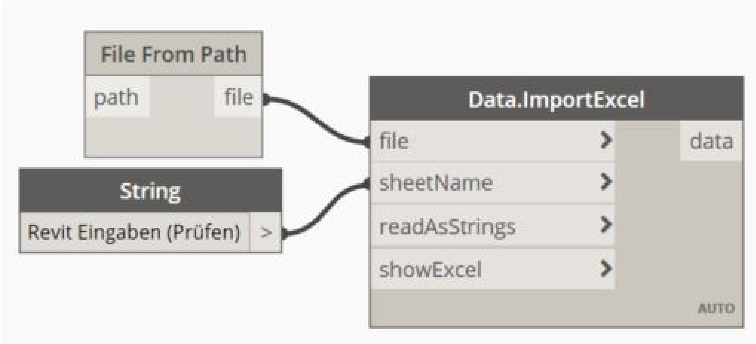


Figure 77: Example of importing data from excel using dynamo.

The selection of the file path is done through an interface by using nodes from the "Data shape" package Fig(78). The output of this node is the input of the previous step in Fig(74).

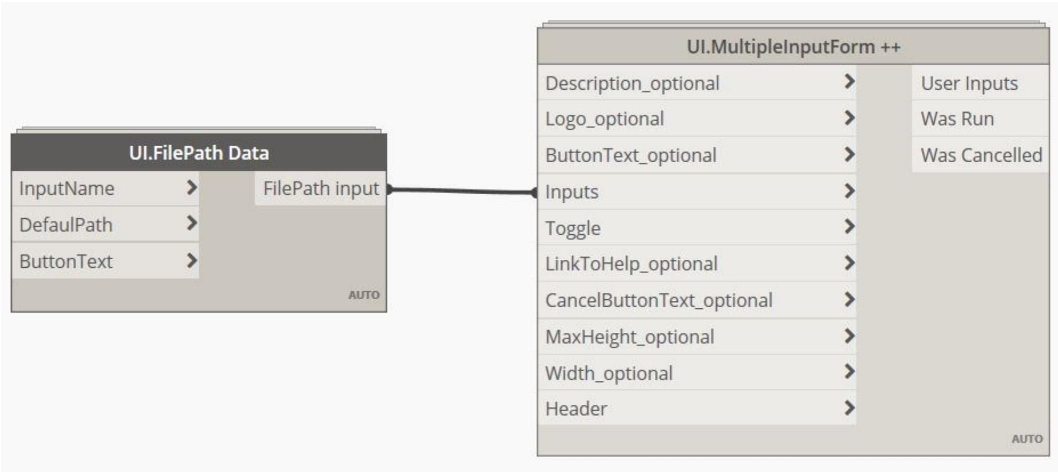


Figure 78: The main node of the interface of selecting the file path.

In this step, the user has only browse and select the required excel file using the interface Fig(79).

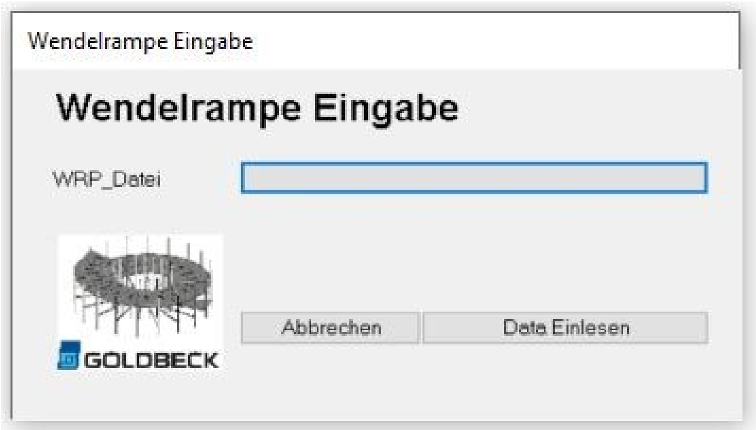


Figure 79: Interface of selecting the excel file.

2. <u>Set the values of the calculated parameters to the plates/beams</u>: after collecting and sorting the values from excel in step 1. We use the node "setparamterByName" exactly like step number 5 in script 1. This node is the best way to set a value of a parameter with Dynamo either as a type or instance parameter.

4.4. Concept of the mathematical calculation algorithm

This section discusses the calculation model used in the aforementioned workflow. This calculation model is made using Microsoft Excel. The feature of the standard usage of Excel is used along with Excel VBA. VBA stands for Visual Basic for Applications. Excel VBA is Microsoft's programming language for Excel and other Microsoft programs. Where writing the operations to be done through the form of textual code.

As discussed in section 4.1, the calculation model is implemented to enhance the performance of the workflow and accelerate the processing time of the workflow. The calculated-parameters that require extensive calculations are calculated in this calculation model instead of being calculated in the BIM system. As shown in Table 5 and Table 6, there are some parameters in the plate and beam families that are calculated in the calculation model. Then the final values are exported to be written to these parameters.

- **Structure of the calculation model**

  By opening the Excel file of the calculation model, we get a desktop interface with some buttons Fig(80). This interface is divided into three sections: Input, calculation, and output section. Each section has some buttons that redirect the user to a certain page of the calculation model or execute a specific operation.

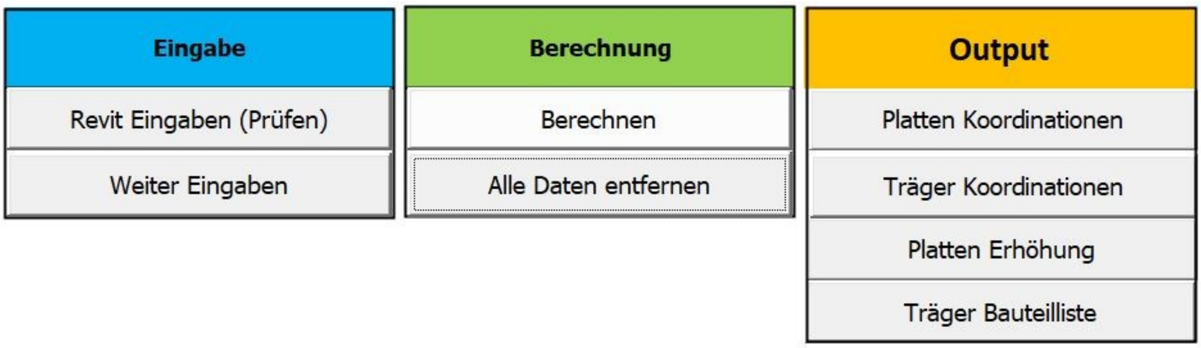


Figure 80: The desktop interface of the calculation model.

- **Input section**

The first button of this section (Revit input check) directs to a page with all the data are exported from Revit Fig(81). This button is programmed by creating a subroutine using the VBA syntax to open this page when this button is pressed. Another button is programmed on this page to take the user again to the desktop page as showing in Fig(81).

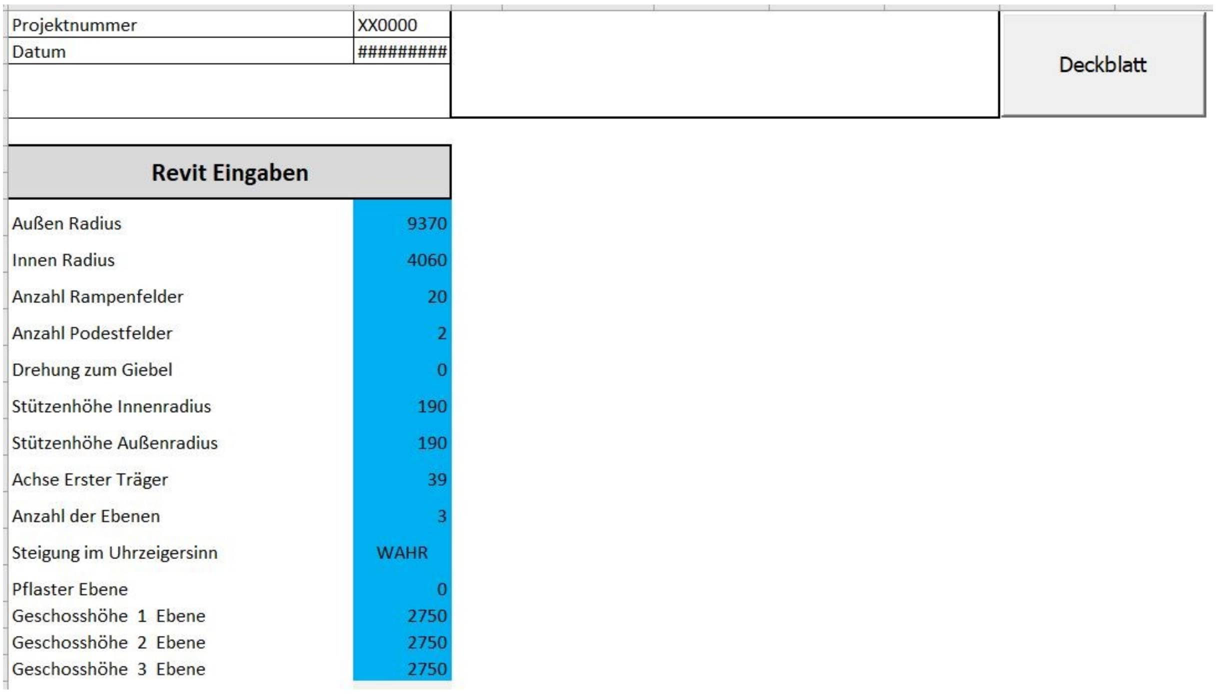


Figure 81: Page of the data exported from Revit in the calculation model.

The second button is to input some more inputs, which are required for the calculation of the parameters. The inputs required in this step are listed in table 7.

Table 7: Inputs of the Calculation model.

	Input data	**Unit**	**Type of input**	Figure
1	Ramp's grids number	Ascend/Descend	By graphically selecting (VBA Excel interface)	79
2	Type of ramp	Gable/Long	By graphically selecting (VBA Excel interface)	80
3	Position of the ramp	Right/left	By graphically selecting (VBA Excel interface)	81
4	The transversal slope of the ramp	%	By writing in a cell (VBA Excel interface)	81/82
5	The slope of the attached parking building	%	By writing in a cell (VBA Excel interface)	81/82
6	Position of the low point	Down/up	By graphically selecting (VBA Excel interface)	81/82
7	An additional offset of the ramp	mm	By writing in a cell (VBA Excel interface)	

The author decided to use graphical methods for entering these parameters to make it more user-friendly. The user is given a series of windows, each of which graphically asks the user to select or make a specific entry. Then the next window appears until all inputs are entered. This is done using the power of VBA to visualize this. The following figures show these steps, which show the input method of the parameters in table(7).

The first window shows two animated photos with two buttons asking the user whether the sequence of the ramp number in descending or ascending order Fig(82). By pressing on one of them will appears the second window.

The second window shows two photos with two buttons asking the user whether the ramp is a gable side ramp or long side ramp Fig(83). In this window, there is a back button to go back to the previous window. Depending on what is selected, a different window will appear.

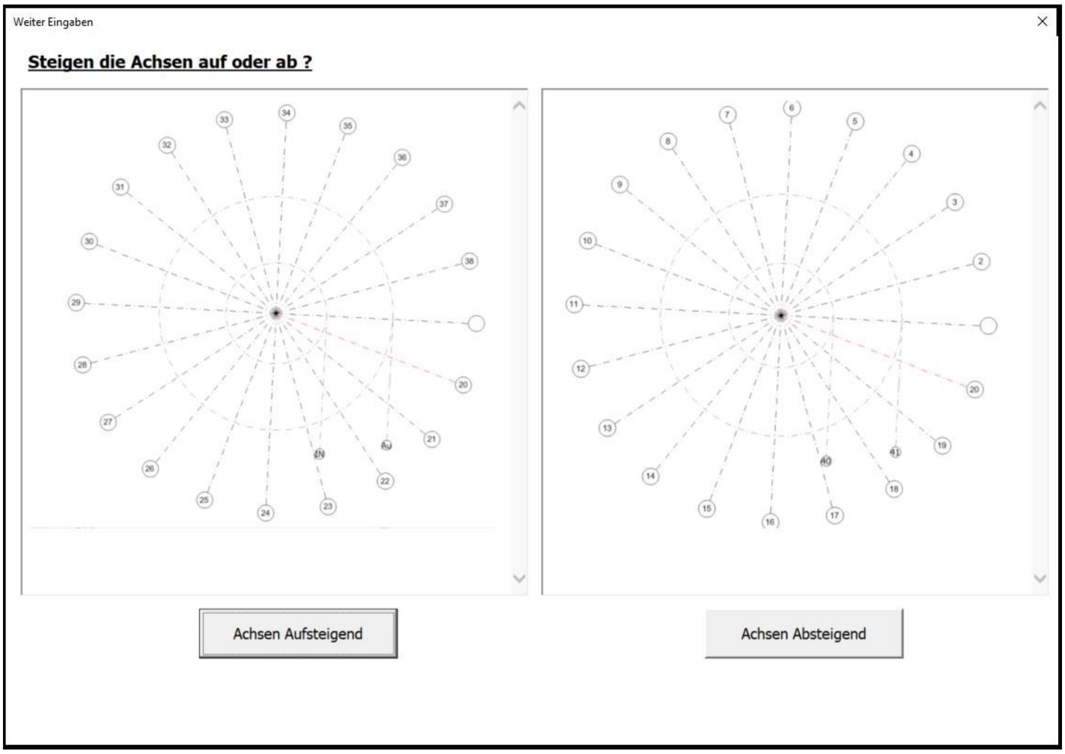


Figure 82: Window asks for the sequence of the grid numbers.

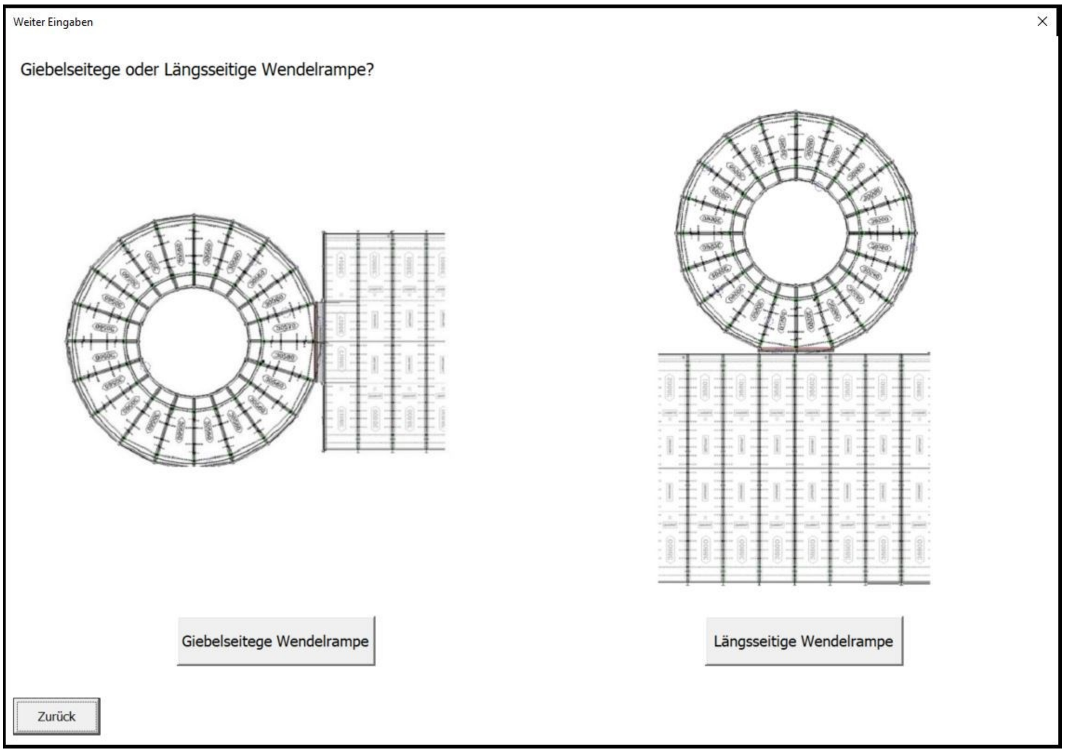


Figure 83: Window asks for the type of the helical ramp.

The third window with a gable side ramp appears in Fig(84). The window asks if the gable side ramp is selected in the previous step. In this window, the user has a simplified 2d drawing of the helical ramp with the parking building. First, the user must choose whether the ramp is located to the right or left of the parking building by graphically selecting the right or left ramp. Depending on this selection, the 2D drawing underneath is altered to match the selection. Then the user has to select the location of the low point of the parking building. Depending on this the dimensions will be changed to adopt with the selection. After that the transversal slope of the ramp, the slope of the parking building, and the dimension of the location of the ramp.

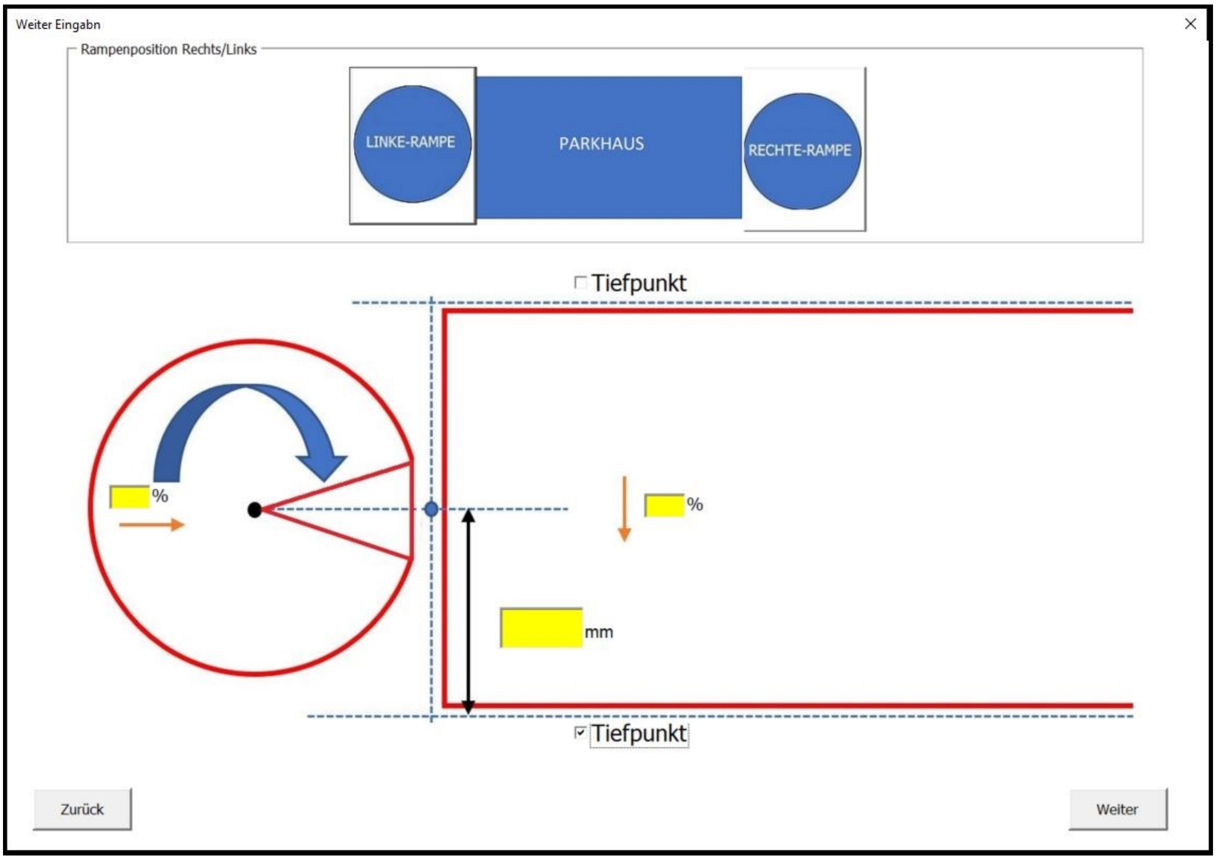


Figure 84:Gable side ramp window.

If the case of long side-ramp is selected, the third window with a long side ramp appears Fig(85). The same inputs as discussed in the case of the gable side ramp have to be entered.

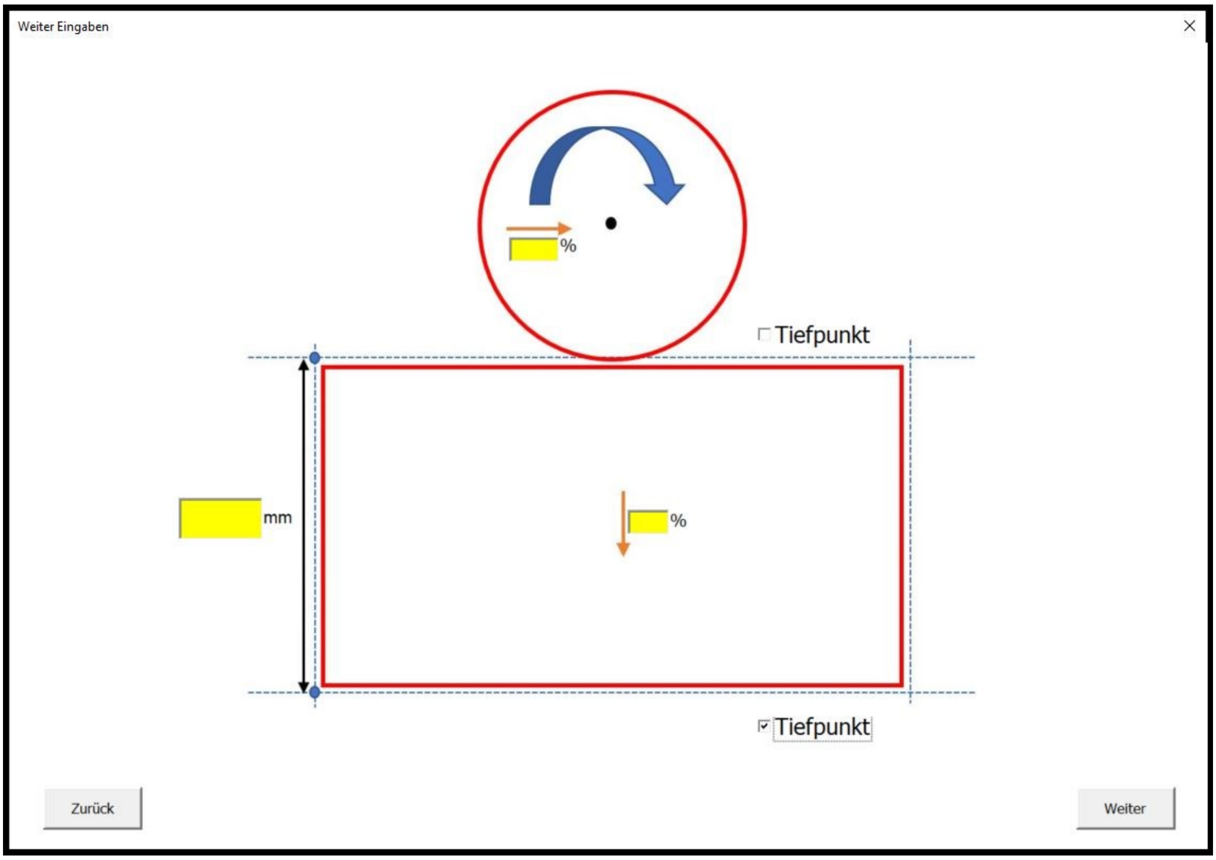


Figure 85: Long side ramp window.

- **Calculation section**

The first button is the calculation button, which is executing the calculation model. By pressing on this button number of subroutines will be carried out[1]. First, copy the calculation sheet based on the number of levels, in which each level has a separate calculation sheet. The calculation sheet is a normal-based Excel sheet, in which some parameters are calculated Table(8).

Table 8: The parameters calculated in the calculation sheet.

	Parameter name	Description
1	E1_Pd1_1/E1_Pd1_2/ E1_Pd1_3/ E1_Pd1_4	The coordinates of the 4 point corners of the first landing plate. These are calculated based on the opposite points of the parking-building slab to ensure a smooth vehicle passage between the ramp and the building.

[1] For the complete code, check the VBA excel file.

2	E1_Pd2_1/E1_Pd2_2/ E1_Pd2_3/ E1_Pd2_4	The coordinates of the 4 point corners of the second landing plate. These are calculated based on the opposite points of the parking-building slab to ensure a smooth vehicle passage between the ramp and the building.
3	E1_P1_1/E1_P1_2/ E1_P1_3/ E1_P1_4	The coordinates of the 4 point corners of the first ramp plate, which calculated used a complex formula based on the helix_Pitch, the helix angle, the ramp dimensions.
4	E1_PL_2/E1_PL_3	The coordinates of the inner corner points of the last ramp plate, calculated on the basis of the coordinates of the adjacent landing plate to ensure a smooth connection between the landing and ramp plate
5	Helix_Pitch	The pitch of a helix is the height of a complete helix cycle, measured parallel to the axis of the helix see Fig(86).
6	Helix_Angle_Inner/ Helix_Angle_Outer	The angle between any helix and an axial line on its circular right cylinder. Because the inner side and the outer side have different radius they have also different helix angles
7	Step	It is the key parameter in the calculation model. It represents the difference between the points with the same position in the adjacent plates, e.g. the difference between point 1 in the first plate and point 1 in the second plate or the difference between point 2 in the 4th plate and point 2 in the 5th plate. This value is constant for all points in all ramp plates.
8	Transversal_ Slope _Landing	The transversal landing slope is different from the transversal slope of the plates. Therefore it must be calculated.
9	Pd1_Helix_Angle_Inner/ Pd1_Helix_Angle_Outer/ Pd2_Helix_Angle_Inner/ Pd2_Helix_Angle_Outer/	The helix angle of the landings at the outer and inner side.

10	Pd_T_1_Angle_Inner/ Pd_T_2_Angle_Inner/ Pd_T_3_Angle_Inner/ Pd_T_1_Angle_Outer/ Pd_T_2_Angle_Outer/ Pd_T_3_Angle_Outer/	The helix angle of the three beams of the landings at the inner and outer sides.
11	Ramp_Case	There are 8 cases of the ramp. These cases are depending on the position of the ramp, the direction of the ramp, and the position of the low point see Table(9).

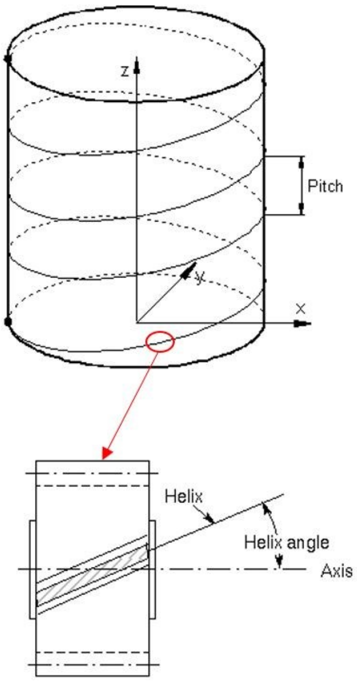
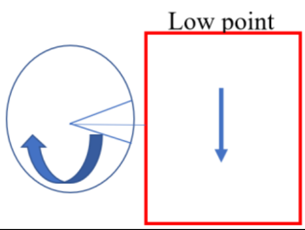
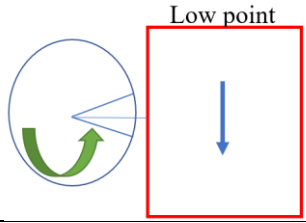
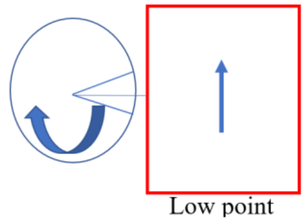
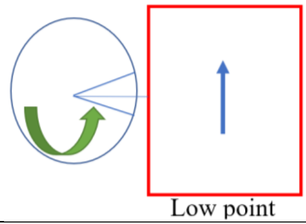
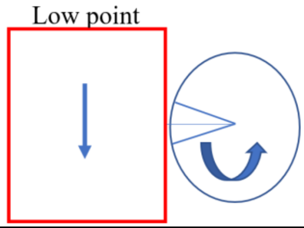
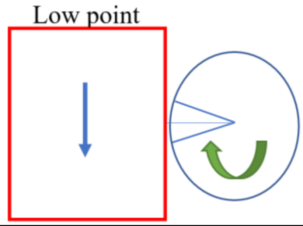
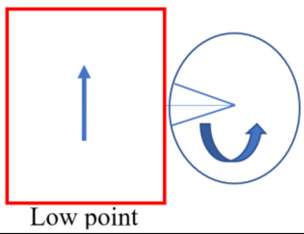
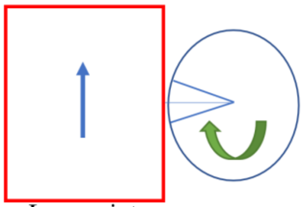


Figure 86: sketch shows the Helix pitch and Helix angle.

Table 9: different cases of the helical ramp.

	Low point	Clockwise/Counterclockwise	Left/Right	Sketch
Case 1	Top	Clockwise	Left	

Case 2	Top	Counterclockwise	Left	
Case 3	Down	Clockwise	Left	
Case 4	Down	Counterclockwise	Left	
Case 5	Top	Clockwise	Right	
Case 6	Top	Counterclockwise	Right	
Case 7	Down	Clockwise	Right	

Case 8	Down	Counterclockwise	Right	

second, a different subroutine will read the calculated parameters at each level and save them in arrays. Then using the calculated parameters e.g (step, helix pitch, etc..) will calculate the coordinates and the increments of all plates and the coordinates see Fig(88) and Fig(89) and the coordinates and rotations of all beams see Fig(89). This will be calculated depending on the ramp case and writes them in tables. Fig(87) shows a part of the code as an example.

```
'schreiben die Träger Infos arrays in jede Excel Spalte ( Träger-Koordinationen sheet)
'----------------------------------------------------------------------
count = -1
For j = 0 To N_E
        For i = 0 To N_F
                count = count + 1
                Worksheets("Träger_Bauteilliste").Range("D" & (2 + count)).Value = H_Außen(count) - (Worksheets("Revit Eingaben (Prüfen)").Range("M1").Value / 2)
                Worksheets("Träger_Bauteilliste").Range("F" & (2 + count)).Value = H_Innen(count) - (Worksheets("Revit Eingaben (Prüfen)").Range("M1").Value / 2)

                ' Drehung Lochbild

                ' Case 1
                If Worksheets("Weiter Eingaben").Range("B9").Value = 1 And Worksheets("Revit Eingaben (Prüfen)").Range("B17").Value = True Then
                Worksheets("Träger_Bauteilliste").Range("E" & (2 + count)).Value = Q_Außen(count) * -1
                Worksheets("Träger_Bauteilliste").Range("G" & (2 + count)).Value = Q_Innen(count)
                End If

                ' Case 2
                If Worksheets("Weiter Eingaben").Range("B9").Value = 0 And Worksheets("Revit Eingaben (Prüfen)").Range("B17").Value = True Then
                Worksheets("Träger_Bauteilliste").Range("E" & (2 + count)).Value = Q_Außen(count) * -1
                Worksheets("Träger_Bauteilliste").Range("G" & (2 + count)).Value = Q_Innen(count)

                        If i = N_F Then ' Letzte träger ist umgekehrt
                        Worksheets("Träger_Bauteilliste").Range("E" & (2 + count)).Value = Q_Außen(N_F + ((N_F + 1) * j))
                        Worksheets("Träger_Bauteilliste").Range("G" & (2 + count)).Value = Q_Innen(N_F + ((N_F + 1) * j)) * -1
                        End If

                End If

                ' Case 3
                If Worksheets("Weiter Eingaben").Range("B9").Value = 1 And Worksheets("Revit Eingaben (Prüfen)").Range("B17").Value = False Then
                Worksheets("Träger_Bauteilliste").Range("E" & (2 + count)).Value = Q_Außen(count)
                Worksheets("Träger_Bauteilliste").Range("G" & (2 + count)).Value = Q_Innen(count) * -1

                        If i = N_F Then ' Letzte träger ist umgekehrt
                        Worksheets("Träger_Bauteilliste").Range("E" & (2 + count)).Value = Q_Außen(N_F + ((N_F + 1) * j)) * -1
                        Worksheets("Träger_Bauteilliste").Range("G" & (2 + count)).Value = Q_Innen(N_F + ((N_F + 1) * j))
                        End If

                End If


                ' Case 4
                If Worksheets("Weiter Eingaben").Range("B9").Value = 0 And Worksheets("Revit Eingaben (Prüfen)").Range("B17").Value = False Then
                Worksheets("Träger_Bauteilliste").Range("E" & (2 + count)).Value = Q_Außen(count)
                Worksheets("Träger_Bauteilliste").Range("G" & (2 + count)).Value = Q_Innen(count) * -1
                End If


        Next i
Next j

End Sub
```

Figure 87: Part of the VBA code as an example.

	Ebene	Platten	Z-Versatz			
			P1	P2	P3	P4
1	E1	P1	-135,55489	-287,35489	-135,5031	18,743746
2	E1	P2	20,483278	-131,31673	20,535067	174,78191
3	E1	P3	176,52144	24,721437	176,57323	330,82007
4	E1	P4	332,5596	180,7596	332,61139	486,85825
5	E1	P5	488,59778	336,79776	488,64957	642,89642
6	E1	P6	644,63593	492,83594	644,68774	798,93457
7	E1	P7	800,67407	648,87408	800,72589	954,97272
8	E1	P8	956,71222	804,91223	956,76404	1111,0109
9	E1	P9	1112,7504	960,95038	1112,8022	1267,0491
10	E1	P10	1268,7886	1116,9885	1268,8405	1423,0873
11	E1	P11	1424,8268	1273,0267	1424,8787	1579,1255
12	E1	P12	1580,865	1429,0649	1580,9169	1735,1637
13	E1	P13	1736,9032	1585,1031	1736,9551	1891,2019
14	E1	P14	1892,9414	1741,1414	1892,9933	2047,2401
15	E1	P15	2048,9795	1897,1796	2049,0315	2203,2783
16	E1	P16	2205,0176	2053,2178	2205,0696	2359,3164
17	E1	P17	2361,0557	2209,2559	2361,1077	2515,3545
18	E1	P18	2517,0938	2365,2939	2517,1458	2671,3926
19	E1	P19	2675,9548	2520,7012	2491,7461	2647
20	E1	P20	2647	2491,7461	2462,7913	2618,0452
21	E2	P1	-135,55489	-287,35489	-135,5031	18,743746
22	E2	P2	20,483278	-131,31673	20,535067	174,78191
23	E2	P3	176,52144	24,721437	176,57323	330,82007
24	E2	P4	332,5596	180,7596	332,61139	486,85825
25	E2	P5	488,59778	336,79776	488,64957	642,89642
26	E2	P6	644,63593	492,83594	644,68774	798,93457
27	E2	P7	800,67407	648,87408	800,72589	954,97272
28	E2	P8	956,71222	804,91223	956,76404	1111,0109

Deckblatt

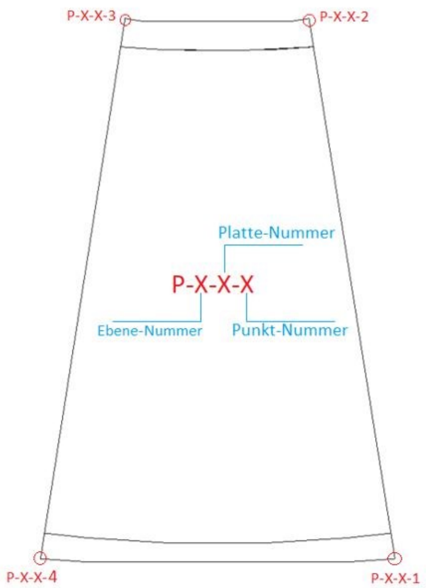


Figure 88: Table of the plates coordinates at each level with a legend of the point name.

Projektnumm	XX0000					
Datum	15.04.2020					
Ebene		Erhöhung	Darstellung Draufsicht mt Höhenabwicklung			
			1	2	3	4
E1	Rampen Platte	83	152	0	152	306
E1	Podest 1	10	184	29	0	155
E1	Podest 2	23	184	29	0	155
E2	Rampen Platte	83	152	0	152	306
E2	Podest 1	10	184	29	0	155
E2	Podest 2	23	184	29	0	155
E3	Rampen Platte	83	152	0	152	306
E3	Podest 1	10	184	29	0	155
E3	Podest 2	23	184	29	0	155

Deckblatt

Figure 89: Table of the increments of the plates.

			Z-Versatz		Querschnittsdrehung		
1	Ebene	Träger	Höhenkote Innenkreis	Höhenkote Außenkreis	Querschnittsdrehung Innenkreis	Querschnittsdrehung Außenkreis	Deckblatt
1	E1	T1	-290,1825867	-137,410141	6,758821487	3,072894096	
2	E1	T2	-134,1444397	18,62802505	6,758821487	3,072894096	
3	E1	T3	21,89372826	174,6661835	6,758821487	3,072894096	
4	E1	T4	177,9318848	330,7043457	6,758821487	3,072894096	
5	E1	T5	333,9700623	486,7425232	6,758821487	3,072894096	
6	E1	T6	490,0082397	642,7807007	6,758821487	3,072894096	
7	E1	T7	646,0463867	798,8188477	6,758821487	3,072894096	
8	E1	T8	802,0845337	954,8569946	6,758821487	3,072894096	
9	E1	T9	958,1226807	1110,895142	6,758821487	3,072894096	
10	E1	T10	1114,160767	1266,93335	6,758821487	3,072894096	
11	E1	T11	1270,198975	1422,971558	6,758821487	3,072894096	
12	E1	T12	1426,237183	1579,009766	6,758821487	3,072894096	
13	E1	T13	1582,275391	1735,047974	6,758821487	3,072894096	
14	E1	T14	1738,313599	1891,086182	6,758821487	3,072894096	
15	E1	T15	1894,351807	2047,124268	6,758821487	3,072894096	
16	E1	T16	2050,390137	2203,162354	6,758821487	3,072894096	
17	E1	T17	2206,428223	2359,200439	6,758821487	3,072894096	
18	E1	T18	2362,466309	2515,238525	6,758821487	3,072894096	
19	E1	T19	2514,741455	2672,981689	2,732133627	1,24697876	
20	E1	T20	2492,473877	2647,050293	1,294553876	0,575937688	
21	E2	T1	-290,1825867	-137,410141	2,732133627	1,249977708	
22	E2	T2	-134,1444397	18,62802505	6,758821487	3,072894096	

Figure 90 Table of the beams coordinates and rotations at each level.

The second button is the Erase button. Pressing this button deletes all steps executed by the calculation button. This is used when the user wants to start from the beginning, e.g. when he has to change something in the input.

- **Output section**

  This section shows the output of the calculation model. Each button shows a table which will be exported to Revit to be written in Revit families. It is not required in the workflow to check these data before exported to Revit. However, the author decided to allow the user to access these data before exported to enhance the transparency and the confidence of the results. Each button direct to one of the tables in Fig(88 to 90).

5. Discussion and analysis of results

As was done in the benchmark task, we will use the aspects mentioned in Chapter 3 as a reference for evaluating and comparing the workflow proposed in this study with the existing workflow described in the case study in Chapter 1. The existing workflow can be equated with the first workflow in the benchmark task where all model elements are modeled manually and all mathematical calculations are performed in the BIM system only.

5.1. Comparison in terms of the performance

As mentioned in Section 3.3.1, the two main aspects of performance that we can use to compare workflows are the processing time and the occupied internal and external memory.

- The processing time

The processing time, in this case, could be divided into some parts e.g Plates modeling, beams modeling, Entry of input parameter values, Computation of calculated parameters, etc.

In order to make a comparison between the two workflows, a statistic study has been performed by modeling a number of helical ramps with different dimensions using both workflows. This study computes the average time needed for each part of the workflow and the total time needed. Table (10) shows the average time needed for each part of each workflow.

Table 10: Comparison between the workflows in terms of the processing time.

	Part of Workflow		Average time [min]	
			existing workflow	Proposed workflow
1	Drawings the grids		3	3
2	Inserting the Families into the project		2	1
3	Modeling the plates*	Place the plates	40	1
		The entry of input parameter values of the plates	5	2
		Computation of calculated parameters of the plates	21	1
4	Modeling the beams*	Place the beams	5	1
		The entry of input parameter values of the beams	10	2
		Computation of calculated parameters of	3	1

	the beams		
5	Modeling the above stories**	15	0
6	Modification on the Entries**	42	1
7	Generation of the shop drawings***	50	4
8	System Crash	10	0
	Total	**205**	**18**

* The modeling of the plates and beams for the proposed workflow is done together. However, in the existing workflow is done separately.
** The time needed for these parts of the existing workflow is highly dependent on the number of floors. However, in the proposed workflow this doesn't play a big role.
*** The time calculated in this study is only for one plate.

As we can see from Table 10, the time needed to model the helical ramp with the proposed workflow is significantly reduced compared to the existing workflow.

In the following, each part of the comparison in table 9 will be discussed.

- <u>Drawings the grids</u>
  In this part, there is no change in the average time, which is done manually by the user in both workflows.
- <u>Inserting the Families into the project</u>
  The time required for this part is reduced by almost half. The reason for this is probably that the complexity of the families and therefore their size is less in the proposed workflow (Revit, 2019). For example, the plate family in the existing workflow is 3048 KB, however, the plate family in the proposed workflow is 1784 KB.
- <u>Modeling the plates/Beams</u>
  - **Place the plates/Beams**
    The average time of this part of the workflow is significantly reduced by the proposed workflow. The reason for this significant reduction in time is the use of the power of the graph-based system to place the model elements instead of placing them manually. This has been learned from the benchmark task see Fig(42).
  - **The entry of input parameter values**
    The time needed to enter the input parameter values is also decreased due to the straightforward interface in the proposed workflow. Whereas in the existing workflow, the user has to look for the parameters that need to be changed.
  - **Computation of calculated parameters**
    Since an external calculation system was used in the proposed workflow, the time needed to compute the calculated parameters in the proposed workflow is considerably reduced compared to the existing workflow. Where the existing

workflow performs all calculation operations in the BIM system, which increases the time.

- Modeling the above stories.
  Modeling the above stories using the existing workflow is achieved by copying the model elements and pasting them into the above stories. This step costs significant time due to the large size and complexity of the families. In contrast, in the proposed workflow all stories are modeled from the beginning with the graph-based system.
  It is also worth mentioning that in case of height differences between the stories, the copy/paste method does not work and the time needed to model the entire helical ramp is doubled. where each floor must be modeled separately. On contrary to the proposed workflow, the differences in heights between stories will not play any role in extending the time, where each story is modeled separately from the beginning.
- Modification on the Entries
  Each modification of the entries after completion of the modeling process costs a lot of time in the existing workflow, where all parameters have to be recalculated. On the other hand, the proposed workflow solves this problem by simply re-entering the parameters into the calculation model and re-reading them by the graph-based system, which takes much less time compared to the existing workflow.
- Generation of the shop drawings
  Due to the use of the smart 2D family mentioned in section 4.2 for the creation of the shop drawing, this also reduces the time in the proposed workflow. On the other hand, the existing workflow used the traditional way to generate the shop drawing.
- System Crash
  It was noticed that when using the existing workflow, the system crashes several times due to the large size and complexity of the families, resulting in a loss of time. However, this is not the case with the proposed workflow, where the processes are parallelized between different systems, not just one system as per the existing workflow.
- Total required time
  Due to the above-mentioned aspects, the overall time reduction of the entire process by the proposed workflow could be reduced by about 90% compared to the existing workflow.

As mentioned above, 4 main implementations have the most significant impact on reducing time in the proposed workflow, these are:

1- Implementing a graph-based system
2- Implementing a calculation system
3- Developing a user-friendly input interface
4- Developing smart 2D families for the shop drawings

Using the calculated average times listed in Table 10, the relative weight of the benefit of each implementation in terms of time-saving is summarized in Fig(91).

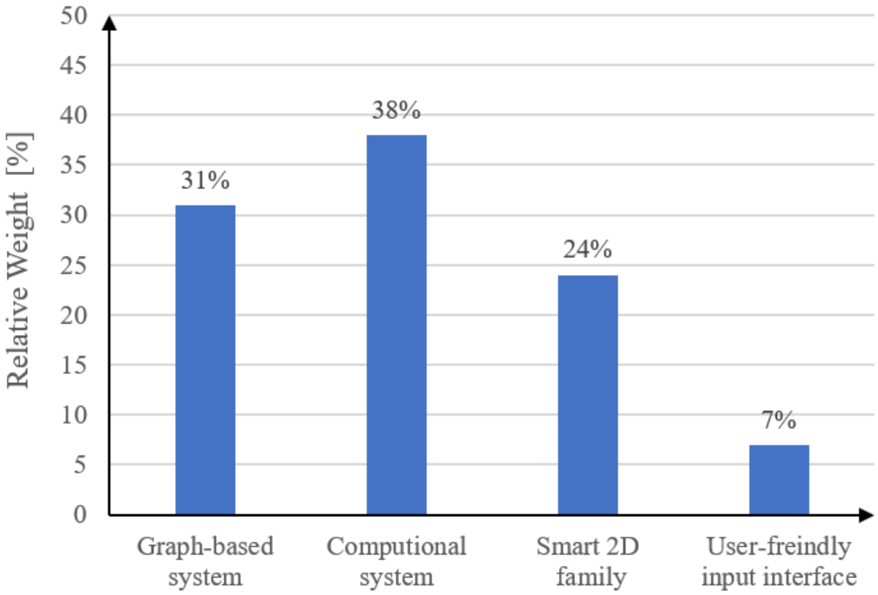


Figure 91: The relative weight of the benefit of each implementation in terms of time-saving

- Occupied internal and external memory.
  In terms of the internal occupied memory, it has been noticed due to the parallelization of the processes between different systems the internal occupied memory has been slightly decreased in the proposed workflow than the existing workflow.
  On the other hand, the external occupied memory is decreased in the proposed workflow compared with the existing workflow. Table(11) shows the occupied external memory for each family for each workflow.

Table 11: External Occupied memory for each workflow.

	Element	External occupied memory [KB]	
		Existing Workflow	Proposed Workflow
1	Ramp plate family	3048	1784
2	Landing family (Right)	3544	1512
3	Landing family (Left)	2740	1488
4	Ramp Beam Family	492	448
5	Installation surface family	860	---
	Total	**10684**	**5232**

5.2. Comparison in terms of usability

As aforementioned, the usability of the workflow is one of the most important aspects of the comparison and evaluation of the workflow. In this section, the aspects mentioned in 3.2.2 could be used as references to compare and evaluate the workflows.

Learnability: to assess the learnability of a workflow, a key question has to be answered (Jackson, et al., 2011) namely "What does it require for the user to learn the basic/advanced functions of the workflow?".Based on the experience with working with the current method(existing Workflow), the users have always needed a 1-day special training to learn how it works. Moreover, the dependence only on the BIM system interface can be seen as very complex with several different but interconnected modeling steps, resulting in a steep learning curve for inexperienced users.

However, the engineers will need only a tutorial video to be able to work with the proposed workflow. Where it has user-friendly interfaces and a few not interconnected steps.

Efficiency: this attribute can be asses based on two main aspects. First to which extend the workflow can offer functional variability. Second, the quality of the output.

Regarding the functional variability and according to table(2), both workflows mainly support all the high-necessity variabilities e.g. dimension of the ramp, the position of the ramp, etc.. however the proposed workflow offer more variability possibilities that do not available in the existing workflow e.g. variability in the number of fields, Automatic generation of more stories and Manual controlling of the error. These additional variabilities give the proposed workflow preference over the existing workflow.

Concerning the quality of the output, and since the accuracy tolerance has to be very limited (Alexander, 1988) and some small errors can lead to huge difficulties during the construction, it is very crucial to ensure that the proposed workflow also produces satisfactory results. Therefore, the proposed workflow was used to model a number of existing helical ramps to compare the results with the results of the ramps already built to ensure that the accuracy tolerance of the workflow is within satisfactory limits. In this comparison, results from 7 existing projects have been compared to the results from the workflow developed in this study. The comparison is based on the dimensions of the plates and the positions of the beams Table(12-14). Fig(92) shows the dimensions of the ramp's plate.

Table 12: The accuracy differences for the dimensions and increments of the ramp plate.

Ramp Plate									
		Projet 1	Project 2	Project 3	Project 4	Project 5	Project 6	Project 7	
		Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Average Delta [mm]
Dimensions	H_Inner	1	0	1	1	0	1	1	1
	H_Outer	9	1	1	1	3	0	0	2
	V_Left	2	3	3	3	3	3	3	3
	V_Right	1	2	2	2	2	1	1	1
	Diagonal_1	3	2	3	2	2	2	2	2
	Diagonal_2	2	4	4	4	4	3	3	3
Increments	P2	0	0	1	1	0	0	0	0
	P1	1	1	1	1	1	1	1	1
	P2	0	0	0	0	0	0	0	0
	P3	1	0	3	0	0	0	0	1
	P4	1	1	3	1	1	1	1	1

Table 13: The accuracy differences for the dimensions and increments of the Landing plates.

Landing 1/2									
		Projet 1	Project 2	Project 3	Project 4	Project 5	Project 6	Project 7	
		Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Delta [mm]	Average Delta [mm]
Dimensions	H_Inner	1	1	1	1	0	0	0	1
	H_Outer	2	1	1	1	0	0	1	1
	V_Left	5	5	5	5	4	4	5	5
	V_Right	0	0	0	0	0	0	0	0
	Diagonal_1	3	5	3	8	2	2	3	4
	Diagonal_2	2	0	2	3	2	2	2	2
Increments	P2	0	1	1	1	1	1	1	1
	P1	0	0	0	0	0	0	0	0
	P2	0	1	0	0	0	0	0	0
	P3	1	0	1	1	1	1	1	1
	P4	0	0	0	0	1	1	1	0

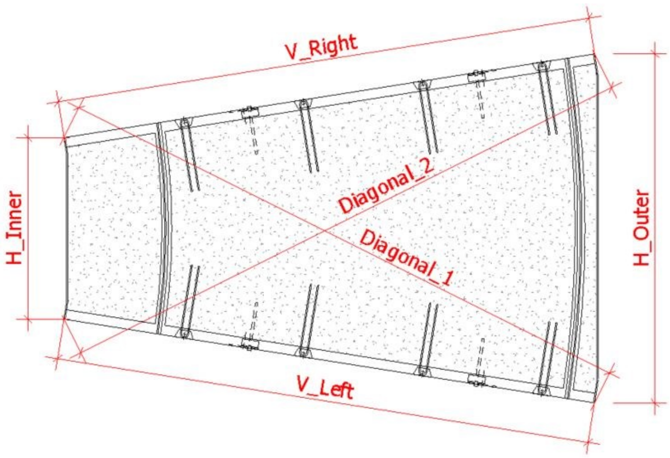


Figure 92: Dimensions of the ramp's plate.

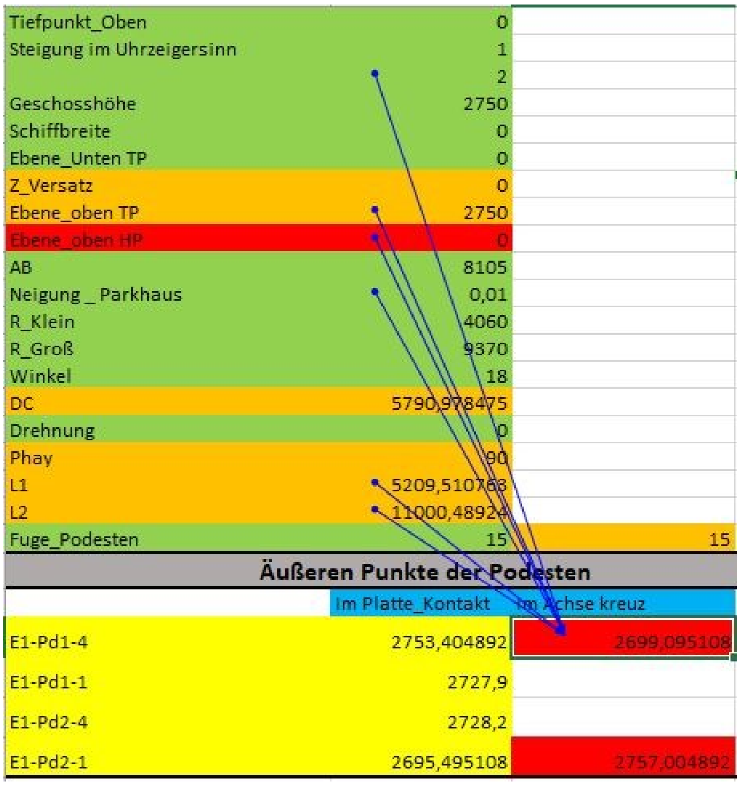
Table 14: The accuracy differences for the beam position.

Beam Position																						Average Delta[mm]
Project 1	Delta Innen[mm]	2	2	2	3	2	2	2	2	2	3	2	2	2	2	2	1	2	2	1	2	2
	Delta Outer[mm]	1	0	0	1	1	1	1	0	0	1	1	1	0	0	0	0	1	0	1	8	1
Project 2	Delta Innen[mm]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	0	2
	Delta Outer[mm]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	3	5	1
Project 3	Delta Innen[mm]	3	2	2	3	2	3	2	2	2	2	1	2	2	2	2	1	2	1	2	2	2
	Delta Outer[mm]	1	1	1	0	1	0	1	1	0	1	0	1	0	0	1	0	1	0	1	9	1
Project 4	Delta Innen[mm]	2	3	2	3	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	1	2
	Delta Outer[mm]	1	1	1	0	1	1	1	1	0	1	0	1	0	0	1	0	1	0	1	9	1
Project 5	Delta Innen[mm]	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	6	4	2
	Delta Outer[mm]	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	3	3
Project 6	Delta Innen[mm]	2	2	2	2	2	3	3	2	2	2	2	2	2	2	2	1	1	1	1	1	2
	Delta Outer[mm]	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	8	1
Project 7	Delta Innen[mm]	2	2	2	3	2	2	2	2	2	3	2	2	2	2	2	1	2	2	1	2	2
	Delta Outer[mm]	1	1	1	0	1	0	1	1	0	2	1	2	2	2	2	1	2	1	2	2	1

As shown in Table(12-14), the average differences (delta) between the values of the existing helical ramp and the values of the workflow proposed in this study are within 2 mm, which is within the satisfactory limit.

Satisfaction: it is the extent to which workflow is user-friendly, attractive, and trustworthy for the user

- **User-friendly**: comparing both workflows, the proposed workflow has straightforward and understandable steps. That because of the designed input interfaces using Dynamo and VBAsee Fig(70) and Fig(82-85). Where all inputs are gathered in one place without the need to search for what/where to input.
  On the other hand, the existing workflow depends only on the interface of the BIM system. By their nature, BIM systems have a very complex user interface, and adding advanced procedural parametric modeling functionality results in an overly complex user interface (JANSSEN, 2015). Where the user must be aware of what needs to be entered at each step and look for it in the BIM system interface.
- **Attractive**: as mentioned in section 3.3.2 the attractivity is to the ability of the system to be visually attractive to the user. The author designed the user-interface using images and graphical objects to make the system more attractive and straightforward to the user.
- **Trustworthy**: as aforementioned this attribute represents the confidence of the user in the workflow. There are two main points here to be evaluated namely, the stability and the transparency of the internal processes.
  The stability: the BIM system by its nature large complex data sets. Running all calculations within the BIM systems as the existing workflow can greatly reduce latency and robustness (JANSSEN, 2015). This is already evident within the existing workflow, where we often experienced system crashes, see table(10). This reduces the stability of the workflow and consequently the user's confidence in the workflow.
  The transparency of internal processes: It is not easy to achieve transparency of the internal calculation in such a complex parametric modeling task, where the calculations are extremely complex. However, the proposed workflow provides partial transparency of the internal calculations. This is achieved by performing all calculations in Excel. Excel offers the possibility to track the calculation easily by following the arrows, see Fig(93). On the other hand, it is in fact extremely difficult to trace the computation in the BIM system, especially with such a complex parametric modeling task with several hundred constraints and mathematical formulas.



Tiefpunkt_Oben	0	
Steigung im Uhrzeigersinn	1	
	2	
Geschosshöhe	2750	
Schiffbreite	0	
Ebene_Unten TP	0	
Z_Versatz	0	
Ebene_oben TP	2750	
Ebene_oben HP	0	
AB	8105	
Neigung _ Parkhaus	0,01	
R_Klein	4060	
R_Groß	9370	
Winkel	18	
DC	5790,978475	
Drehnung	0	
Phay	90	
L1	5209,510763	
L2	11000,48924	
Fuge_Podesten	15	15
Äußeren Punkte der Podesten		
	Im Platte_Kontakt	Im Achse kreuz
E1-Pd1-4	2753,404892	2699,095108
E1-Pd1-1	2727,9	
E1-Pd2-4	2728,2	
E1-Pd2-1	2695,495108	2757,004892

Figure 93: Example of how to track the calculation in Excel.

5.3. Comparison in terms of planning costs

As mentioned in a pervious chapter, that the company Goldbeck produces approximately 100 parking garage yearly with value from 350 Mio. Euro to 480 Mio. Euro. On average, the company builds 3 to 5 out of every 100 parking garages with at least 2 helical ramps. For that reason it is always worthwhile for the company to implement new modelling technologies to accelerate this process and increase the accuracy of the modelling. These implementations are also worthwhile as they could save time and cost of the modelling process in addition to the cost of corrections in manufacturing and assembly due to the lack of modelling accuracy.

In this section we will discuss the costs that could be saved by implemintig the proposed methodology developed in this study with comparing to the existing methodology discussed in section 2.5.

The comparison will be performed in terms of four parameters :

- Costs of modelling time
- Quality of the required user
- Costs of learning
- Costs of modyfing
- Cost of data structures

5.3.1. Costs of modelling time.

Using the traditional methods of cad drafting, the completion of a spiral ramp normally took about two weeks. These two weeks are required when an experienced engineer is responsible for the task. However an average user were not eligible to deal with such a complex task. However by implementing a BIM workflow descrieped in section 2.5 (the existing methodology) the time required to fully complete the modelling of a helical ramp using this method is reduced to less than a week.

On the other hand, the author tried to overcome these problem by introducing the new concept developed in this study. This study argues that the proposed workflow could reduces the time needed for fully completion of a helical ramp in less than one work day. Thereby the costs for modelling and design for each spiral ramp are saved to a decent extent.

5.3.2. Quality of the required user.

By using the existing workflow, due to the complexity of the workflow and the lack of the user-friendly interface , only an experienced engineer can be in charge of this task. However, the proposed workflow does not require previous experience in modelling such complex geometries, which can be easily performed by an average user. This is the benefit of the high level of usability of the workflow discussed in the previous section. Consequently, the costs for an experienced engineer could be saved and employed for other tasks.

5.3.3. Costs of learning.

One of the most important factors is the cost of learning the workflow, which depends on the learning ability of the workflow, discussed in the previous section. The higher the level of learnability, the lower the cost of learning. For example, the existing workflow used by goldbeck requires special training to teach the engineer how to use the method. On contrary, due to the high level of performance and user-friendliness of the workflow proposed in this study, only a short instruction is needed to enable the user to use the method efficiently, reducing the costs of learning significantly.

5.3.4. Costs of modyfing

Modifications to the model may be necessary for many reasons, e.g. changes in design, errors in the production, etc.. Therefore, if the modelling method is not flexible enough to adapt the model to the new changes, this will costs repeating the modelling from scratch. because of using only the BIM system in the existing method and due to the complexity of the BIM families used in modelling, each tiny change needed after completion of modelling costs time equivalent to the time needed to repeat modelling from scratch. Vice versa, the proposed method is designed in such a way that any change can be easily managed by re-entering the changes into the calculation model and re-exporting the values into the BIM system, which is performed easily, see table (10).

5.3.5. Cost of data structures

This workflow also took advantage of the power of the Building Information modelling methodology, which, by storing all the information of the structural elements within the BIM elements, makes it very easy to create a data structure of the elements. This structure could be used for the scheduling of the project, for the production or even for the execution and assembly of these structural elements. On the other hand, the existing workflow does not have this capability, thus the data structure of the elements requires additional costs.

6. Conclusion

The Parametric modelling and Building Information Modelling (BIM) are introduces nowadays as very promising modelling concepts. However, there are several ways to develop a parametric BIM workflow. For this reason, this thesis aims to investigate which properties should be included in a parametric workflow in order to obtain a powerful workflow in terms of performance, cost and user-friendliness. This investigation was carried out by literature research and by proposing a novel concept for automating the generation of complex geometries. This research focuses on a specific type of complex geometry, namely the helical ramp for multi-storey car parks. However, the concepts discussed and developed in this study could also be implemented in any other type of complex geometry.

Through a literature research, the author has discussed the various types of existing parameteric modelling, where there are two main types of the parameteric modelling, either by using the BIM system alone(embedded approach) or by combining the BIM system with a graph-based system(Coupled approach). However, the new concept introduced in this theises is a modified version of the coupled approach. This concept introduces an additional item to be coupled with the BIM system namely a computational system along with the graph-based system will be coupled with the BIM system. Autodesk Dynamo is used as the graphics-based system, Microsoft Excel as the computational system and Autodesk Revit as the main BIM system, whereby through this integration the workflow achieves an explicit multi-operational iteration.

The concept of parametric BIM workflow proposed in this study combined the knowledge from the field of computer science with that of BIM. Many aspects from the field of computer science were taken into account, such as processing time, occupied capacity of internal and external memory, memorability, efficiency, learnability, etc. On the other hand, there are various approaches from the field of BIM in the literature that can be followed. These approaches indicate which type of systems could be involved in the workflow, e.g. BIM system alone or BIM system with a graph-based system, etc. To make solid judgments, the author formulates a simple benchmark task in which the different workflow concepts are tested. From this test, it was found that the workflow where the BIM system was used in combination with other systems (graph-based system and computer system) achieved better results in terms of performance and usability. All of these aspects are then used as a refrenceses in desining the detailed workflow for the generation of the the parking garage helical ramp.

The author has designed a detailed workflow for the parametric generation of the geometry of the helical ramp and automated generation of its shop drawings, where all the afoermentiod aspects are implemented This workflow is a modified version of the coupled approach workflow, where the author added a computational system to the workflow to reduce the computational effort done by the BIM system and to be able to obtain a more user-friendly interface. The tools developed in this workflow include parameterized families in the BIM system, a calculation algorithm using VBA-Microsoft, a graph-based algorithms.

The families are smart families that automatically alter their dimensions and elevations according to any change in the input parameters.

The calculation algorithm is a highly complex mathematical calculation model that calculates all the necessary parameters of the structural elements. Outsourcing these calculations from the BIM system significantly reduces the calculation effort done by the BIM system.

The graph-based algorithms have two main functions. First, the automatic placement of the structural elements, where reduces the time of the modelling and increases the Precision. Second, it serves as a connector between the BIM system and the calculation system.

All of these tools are organized in a way to compose the workflow.

On the other hand, a case study was discussed in which a different workflow for the parametric generation of the helical ramp of the car park was discussed. This workflow exclusively uses the BIM system without the assistance of any other systems. The modelling concept behind this workflow is generating a series of associative parametric modelling processes inside the BIM system, where the user carries them out in a specific sequence. This approach does not support explicit multi-operational iteration, where only a single iteration could be achieved.

The developer of this workflow created highly complex parametric constraint BIM families with hundreds of parametric dependencies. Because of these dependencies, the BIM system performs all calculations in the background. Thereby the geometry of the ramp is created. However, the due to the huge calculation effort need to be performed inside the BIM system, the performance of this workflow is very poor.

Finally, a comparison is carried out between the workflow mentioned in the case study with the workflow proposed in this study with regard performance and the usability. This comparison has shown that either the performance aspects or the usability aspects are significantly improved by implementing the following implementation into the workflow:

- **Implementation of a graph-based system**
  The reason behind this is that the graph-based system could provide multi-operation iteration processes that would allow us to automate and parallelize many steps, e.g. (placing elements, modifying geometries, etc.).
- **Implementation of a calculation system**
  By their very nature, BIM systems have poor mathematical computational capabilities. However, the parameterisation of a complex geometry requires a high level of mathematical calculation complexity that the BIM system cannot sustain. Therefore, the outsourcing of the computing workload in this case saves the performance loss considerably.
- **Development of a user-friendly input interface.**
  The development of a custom user interface that suits the purpose of the workflow increases the usability of the workflow. In top of that, working with a simple input

interface, depending on the graphical way of entry, enhances learnability, efficiency, attractiveness and trustworthiness

- **Development of intelligent 2D families for workshop drawings**
  Developing smart 2d families to be used in generating the shop drawing allow us to work with a light families. And obtain high-performance families as well as comprehensive workshop drawings

7. References

Hasse, Jürgen. 2015. Zur Kulturgeschichte und Heterotopologie des Parkhauses. Jul 2015.

Jain, Piyush, Dubey, Sanjay Kumar and Rana , Ajay. 2012. Analysis And Performance Evaluation Of Software System Usability. *International Journal of Computer Applications.* 17 April 2012, pp. (0975 – 8887).

2015. *Parking Structures: Recommended practice for design and construction.* USA : The Precast/Prestressed Concrete Institute, 2015.

2016. *Recomendations for the codes "SP 113.13330.2016. Parking structures".* Moscow : s.n., 2016.

Abedin, Mohammad Saiful . 2016. *Anwenden von grafischen Programmiermethoden für die parametrische Brücken Modellierung und Generierung von Modellvarianten.* 2016.

Alexander, Sven. 1988. Spiral Ramps for Parking Structures —The Prefabricated Solution. *The New Parking Facility at Fornebu Airport, Oslo, Norway.* 1988.

Autodesk. 2020. Live Webinar AEC Collection Essentials: Bridge Design Workflow. [Online] Autodesk, 25 August 2020. https://www.autodesk.com/solutions/bim.

—. Parametric Building Modeling: BIM's Foundation.

Bernd, Rohrbach. 1969. Kreativ nach Regeln – Methode 635, eine neue Technik zum Lösen von Problemen. *Absatzwirtschaft 12: 73–75.* 1969.

Bevan, Nigel. 1995. Measuring usability as quality of use. *Software Quality Journal.* June 1995, pp. 115-130.

Building infromation modeling in the architecture-engineering construction project in Surabya. Chandra, Herry Pintardi , Nugraha, Paulus and Putra, Evan Sutanto . 2016. 2016, Procedia Engineering.

COENDERS, Jeroen L. 2010. Parametric and associative design as a strategy for conceptual design and delivery to BIM. *Symposium of the International Association for Shell and Spatial Structures.* 2010.

Fokus, quantum . Szenarien für die Entwicklung des Parkraummarktes in Deutschland.

Fridrich, Jan and Karel, Kubečka . 2014. BIM – The Process of Modern Civil Engineering in Higher Education. *Procedia - Social and Behavioral Sciences.* August 2014.

Ignatova, E., et al. PARAMETRIC GEOMETRIC MODELING IN CONSTRUCTION.

Innovation, The Cooperative Research Centre for Construction. 2007. *Adopting BIM for facilities management Solutions for managing the Sydney Opera House.* 2007.

Jackson, Mike, Crouch, Steve and Baxter, Rob. 2011. Software Evaluation: Criteria-based Assessmen. *SOFTWARE EVALUATION: CRITERIA-BASED.* NOVEMBER 2011.

JANSSEN, PATRICK and STOUFFS2, RUDI. 2015. TYPES OF PARAMETRIC MODELLING. *National University of Singapore, Singapore.* 2015.

JANSSEN, PATRICK. 2015. PARAMETRIC BIM WORKFLOWS. *Proceedings of the 20th International Conference of the Association for Computer-Aided Architectural Design Research.* 2015.

Kleinmanns, Joachim . 2011. Parkhäuser: Architekturgeschichte einer ungeliebten Notwendigkeit. 2011.

Mahmudova, Shafagat. 2019. Analysis of Software Performance Enhancement and Development of Algorithm. *International Journal of Innovative Science and Research Technology.* 1 Januaray 2019.

Maia, Lino , Mêda, Pedro and Freitas, João. BIM methodology, a new approach - case study of structural.

Michael, Thomas. 2016. *Current application of graphical programming in the design phase of a BIM project: Development opportunities and future scenarios with 'Dynamo'.* 2016.

Nederveenab, G.A.van and Tolman, F.P. 1992. Modelling multiple views on buildings. *BIM published.* 1992.

Nielsen, Jakob. 1993. Usability engineering. *London: Academic Press.* 1993.

Nielson, Jacksob. 2010. *Introduction to usability.* 2010.

Revit, Autodesk. 2019. Autodesk Revit 2019 Help Manual. [Online] Autodesk , 2019. https://help.autodesk.com/view/RVT/2019/ENU/.

—. 2020. Revit Api Documents. [Online] Autodesk Revit, 2020. https://www.revitapidocs.com/.

Rynkovskaya, Marina. 2019. Studying the shape of a helical ramp. *Proceedings of the IASS Annual Symposium 2019 – Structural Membranes.* 7-10 October 2019.

Sala , N. 2004. Complexity in architecture: a small scale analysis. *Design and Nature II, M. W. Collins & C. A. Brebbia.* 2004.

Seffah, Ahmed, et al. 2006. Usability measurement and metrics: A consolidated model. *Software Qual Journal.* 2006, pp. 159-178.

Wodehouse, Andrew and William , Jon. 2011. Augmenting the 6-3-5 method with design information. *Research in Engineering Design* . January 2011.