C.C.A. Hendriks

A Conceptual Integrated Parametric Design Tool for Excavation Sites

Master thesis

SET

a

Smart Engineering Tool

TU Delft

Witteveen + Bos

A Conceptual Integrated Parametric Design Tool for Excavation Sites

Master thesis

By

C.C.A. Hendriks

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on September 7th, 2018 at 15:30 PM.

Student number:	4076826
Start Thesis:	11th December 2017
Hydraulic Engineering:	Hydraulic Structures

Thesis committee:	
Prof.dr.ir. J.G. Rots,	TU Delft
Dr.ir. G.A. van Nederveen,	TU Delft
Dr.ing. M.Z. Voorendt,	TU Delft
MSc. J.W.C. Hendriks	Witteveen+Bos

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Preface

I have set myself the goal for this master thesis, to choose a subject that would challenge me and where I could put my creativity into. What is a bigger challenge than understanding programming at a fundamental level by creating a tool that will increase effectiveness in the field and will lay a foundation for something that is still in its infancy in civil engineering? Programming has always fascinated me, but I never had the opportunity to learn this properly. In the job interview at Witteveen+Bos with Hans van Daelen and Leonie Koops, the idea of this thesis was born. I'm grateful for the opportunity they provided me, to perform my master thesis at Witteveen+Bos.

This master thesis is accompanied by the design tool SET that was developed from scratch. The lines of code that are written for the final version of SET consists of three times as many lines compared to this report. SET and the results of this report together, present a tip of the iceberg regarding the changes that digital system will bring to civil engineering in the upcoming years. I am looking forward to what the future will bring and you?

Cas Camiel Andreas Hendriks,
Den Haag, 2018

Acknowledgments

I would like to thank my graduation committee, Prof.dr.ir. J.G. Rots, Dr.ir. G.A. van Nederveen, Dr.ing. M.Z. Voorendt and MSc. J.W.C. Hendriks (Witteveen+Bos) for having an open mind towards the development of software which is a different approach of a civil engineering master thesis. This thesis would not have been possible without their guidance. I want to express my gratitude for their support, guidance, advice and insights during my thesis.

Subsequently I would like to give thanks to my daily advisor Jeroen Hendriks who made time available for consultation on a weekly basis, guiding me through the process by asking the right questions to make up my mind and keeping me from wandering off into the endless possibilities of programming. I would like to point out the efforts of Mark Voorendt, he has put additional effort in coaching me in the process of writing this thesis.

Additionally, I would like to show my sincere thanks to Koen de Jong from Witteveen+Bos, for his help with integrating Plaxis into SET.

I would like to give thanks to Pal van Ogtrop, Jan Ruigrok and Stefanos Gkekas, for their advice on geotechnical issues.

One of my personal goals was to learn how to program. For their advice, experience, knowledge and feedback on programming, I would like to thank the following persons from Witteveen+Bos: Maarten Veerman, Marc Taken, Pieter-Bas de Visser, Arjan Luttikholt, Javier Martínez Avilaj, Kaichen Tian.

Lastly, I would like to show my appreciation to the open source community. Without their legacy this thesis would not have been possible. Many open source, tools, frameworks, modules, scripts and programming languages made this thesis possible.

Abstract

Building Information Modelling (BIM) technologies are changing the conventional structural design process. BIM level-1 and level-2 are providing guidelines and frameworks for standardization with the standard Industry Foundation Classes (IFC) and collaborative design respectively. However, the predicted full potential of BIM, BIM level-3, which is a smart design system, is still in its infancy stage. Many studies focus on single technologies that BIM envelopes and do not grasp the benefit of incorporating all technologies into one model. Additionally, the industry is lagging in the adaptation of BIM technologies, since, as of today, only one company in the world is BIM level-2 certified. Witteveen+Bos recognises the potential of smart systems and is developing many innovative software solutions. In this thesis, as the objective stated, a conceptual partly BIM level-3 excavation site design tool has been developed. The developed tool, integrates parametric design with Plaxis, GIS and RockWorks. The developed tool helps stakeholders to visually explore many design solutions in a short amount of time, based on the costs, MKI, structural- and environmental requirements and the characteristics of the surroundings. The name SET (Smart Engineering Tool) was given to this tool.

Three essential boundary conditions are distinguished for the design of an excavation pit. Soil Layers, groundwater levels and the surrounding buildings. These boundary conditions impose loads and requirements on the excavation pit. The way the groundwater can be kept out determines mostly the costs. For this reason, SET works-out designs with a natural impermeable clay layer, an artificial underwater concrete floor or an artificial impermeable gel layer. Additionally, designs with different retaining walls and grout anchors or struts are worked-out. The design of an excavation pit must meet structural and environmental requirements. A vertical, horizontal and moment equilibrium must hold in the excavation pit. In urban areas, the most important concern is avoiding inadmissible damage or hindrance to adjacent structures, because the surrounding buildings are owned by third parties and deformations, causing damage to these buildings, happen quick. For this reason, the design of the excavation pit must meet the settlement requirements of the surrounding buildings. For each design, “Economisch Meest Voordelige Inschrijving” (EMVI) (Economically Most Attractive Tender) scores are used to quantify each design on durability, the impact on the surroundings, hindrance and risk. EMVI in combination with the cost estimates is used to find the design with the best cost to quality ratio. Rijkswaterstaat has determined a “Milieu Kosten Indicator” (MKI) (Environmental Cost Indicator) value for each building material. In this thesis, this value is used to quantify the environmental impact of a design

An analysis was made on the capabilities that a design tool should have. A literature study was performed on the BIM technologies. From this analysis six BIM technologies are included in the development of the tool. These are: parametric design, algorithmic design, collaborative design, central repository, interoperability and standardization. Additionally, for the development of the design tool, different programming languages and development platforms were considered. A combination of Python and JavaScript together with the Web-based Graphics Library (WebGL) was found to be most suitable. A Python back-end processes the data and runs a design algorithm, while the web-based front-end facilitates collocative design and 3D visualisations. The sever, created with the Python module Django, is used for the communication between Python and JavaScript. The JavaScript module Three.js is used to utilize the WebGL capabilities.

The development of the design tool, called SET (Smart Engineering Tool), is based on the method of rapid prototyping. This includes prototyping, test and review, refine and iterate. The prototyping process is split into four stages: (1) Interoperability and Standardization, (2) Design algorithm, (3) Collaboration, Interaction and 3D Visuals and (4) Integration. In each of these stages, an essential part of SET was developed. SET is interoperable with Plaxis 2D, a geotechnical finite element program and uses the input of GIS and RockWorks. SET’s internal design algorithm determines what excavation site designs should be calculated by Plaxis. This is based on user settings, and the boundary conditions, surrounding and environmental requirements. The boundary conditions are extracted from GIS files and from RockWorks models. The surrounding buildings are extracted from GIS, and RockWorks supplies the soil stratification. The design tool interprets these boundary conditions and standardizes the data. All data is transferred to a parallel computer, where finite element calculations are performed in Plaxis. Plaxis calculates the vertical, horizontal and moment equilibria and the deformations of the soil. SET analyses the successful Plaxis results and performs a unity check on bending moments in the retaining walls together with an analysis on the settlements requirements of the surrounding buildings. SET is tested and reviewed, by a geotechnical expert and an engineer, with two test cases. The results of these tests cases were analysed and validated. The functionalities and the user-friendliness of the tool have been scrutinized. Based on these tests, SET was reviewed, and refinements and iterations of the design tool emerged.

During the development of SET, Witteveen+Bos was working on a tender. The tendered project is a design of a bus lane next to the existing train tracks and is called “Hoogwaardig Openbaar Vervoer in het Gooi” (high quality public transport), in short “HOV in ‘t Gooi”. The currently level intersection of the bus lane with the ‘Oosterengweg’ was used to test SET with two test cases. A tunnel for the Oosterengweg underneath the train track and bus lane, was the first case and a tunnel for the train tracks and bus lane underneath the Oosterengweg, was the second case. The first case, was a similar solution as the final tender design of Witteveen+Bos. The second case however, was an “out of the box” design and which was initially deemed unfeasible by Witteveen+Bos. The testcases took around one hour to prepare in SET. For both test cases, the corresponding Plaxis calculations took around a day of computation time to finish. As a result, SET found many feasible designs for both test cases, in two days. The design of the train and bus tunnel, that was deemed unfeasible, was found to be feasible and almost a third of the cost compared to the Oosterengweg tunnel. However, it must be noted that the costs to redirect the train tracks and the maximum obstruction time of the train was not included in these designs.

It can be concluded that SET is partly BIM level-3. The foundations of the SET lie in BIM level-1 and -2. The design tool supports 2D and 3D visuals and parametric design is used in an object-oriented manner for the design of the excavation pit. The tool does not use the all-enveloping BIM standard IFC. However, the design tool does use standardization. Based on requirements from the surroundings and the environment, SET helps to explore a large number of designs in a short period of time and visualizes different designs in 3D and in an interactive graph. The data is managed from a server, which acts as a central repository. Stakeholders can access this server from a web browser and are able to analyse their preferred designs within the extended design space.

SET shows 6 major benefits. (1) During the conceptual design phase, the feasibility of designs in a large design space can be analysed without being labour intensive and within a short amount of time. (2) The decisions concerning the final design can be postponed to the very end. Thus, designs that would conventionally be deemed unfeasible in the conceptual stage can emerge as feasible in later stages. (3) The possibility of a collaborative platform stimulates shared and parallel decision making. Each stakeholder can incorporate various requirements, limitations and responsibilities into the model, which utilizes the knowledge and expertise of each stakeholder. (4) The design tool can be used from the very beginning in meetings and discussions with clients to convey information in real time, by the means of 3D visuals. Clients can be granted remote access to SET and evaluate design solutions from the comfort of their office, without sharing valuable scripts and sensitive or private data. (5) Calculations can be processed automatically on a parallel server without the supervision of an engineer. (6) SET integrates multiple design tools like GIS, RockWorks and Plaxis. Overall, SET reduces the required labour to create a design and increases the productivity of an engineer. This reduces the costs of the design process and allows engineers to create “out of the box designs”.

Table of contents

List of abbreviations and terms

AJAX	Asynchronous JavaScript And Xml
ASCII	American Standard Code for Information Interchange
BIM	Building Information Modelling
CAD	Computer Aided Design
CPT	Cone Penetration Tests
CPU	Central Processing Unit
CSV	Comma Separated Values
EMVI	Economisch Meest Voordelige Inschrijving (Economical Most Attractive Tender)
FEM	Finite Element Method
GIS	Geographic Information System
GPU	Graphical Processing Unit
HOV in ‘t Gooi	Hoogwaardig Openbaar Vervoer in het Gooi (high quality public transport)
IFC	Industry Foundation Classes
JS	JavaScript
LOD	Level Of Detail
MKI	Milieu kosten indicator (Environmental cost indicator)
OpenGL	Open Graphics Library
SET	Smart Engineering Tool
SLS	Serviceability Limit State
UCF	Underwater Concrete Floor
ULS	Ultimate Limit State
URL	Uniform Resource Locator
WebGL	Web Graphics Library

1 Introduction

1.1 Motivation

In conventional structural design the processes are usually inflexible, labour intensive and expensive. Many design considerations within the full so-called design space go unexplored due to manpower constraints, time constraints (Horst, 2014, p. 65) and increasing environmental requirements. Hence, this limits the discovery of possible out of the box and superior design solutions. Witteveen+Bos thinks that ‘smart’ design systems can contribute to superior design solutions and a reduction of the amount of labour and costs. These ‘smart’ systems are thought to change the conventional design processes in the near future. Witteveen+Bos wants to develop a ‘smart’ system that helps in the exploration of an extended design space of an excavation pit, while considering the requirements given by the environment and the surroundings.

1.2 Background

The development of 'smart' systems and, in particular, Building Information Modelling (BIM) technologies are changing conventional structural design processes. According to a large review on BIM-Enabled Structural Design by Chi et al. (2014, p. 136), "BIM technologies are beginning to play an essential role in structural design practices and already show considerable benefits". According to Azhar (2011, p. 243), "BIM improves collaboration within project teams which will lead to improved profitability, reduced costs, better time management, and improved customer–client relationships". Nevertheless, BIM is in its early development stages and the full potential of BIM has to be proven in the future. Different levels of maturity of BIM are distinguished and are shown in Figure 1-1. BIM level-1 and level-2 systems are object oriented, focus on collaboration and promote the use of standards. BIM level-3 requires the integration of 'smart' functionalities. Allen (2016) predicts a level beyond BIM level-3 which integrates machine learning and artificial intelligence in design tools. As of today, one company in the world is certified in BIM level-2 (Ooms, 2017) and according to Eadie, Browne, Odeyinka, McKeown, and McNiff (2013, p. 145) "BIM is mainly used in the conceptual phase stages of a project with progressively less use in the later stages". Additionally, BIM is mainly used in architecture (Azhar, 2011). For now, according to Chi et al. (2014, p. 136) "BIM is most suitable to be used in design and pre-construction stages and it has potential to be improved in the expected future."

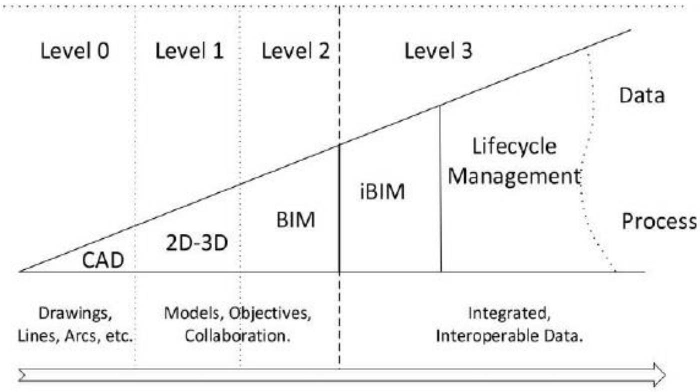


Figure 1-1 BIM Maturity Diagram (Bew & Richards, 2008)

The development of BIM started with the invention of the reliable transistor in 1947, which heralded an exponential development in computer technology. In engineering the computer has become an essential part of the design process. In the 1970s computer-aided design, in short CAD, became an alternative to hand-drawn blueprints. Relations between objects in a CAD model were parameterized by a set of rules and variables and so, parametric design was born (Phillips, 2010). Parametric design allowed small adjustments to automatically propagate through the design (Aish & Hanna, 2017, p. 150) and, thereby turned the computer into an "active" design contributor. In architecture, for example, designers gradually started to use algorithms to model complex structures. Mathematical functions and algorithms, which are assigned to the parametrised connections within an object, change objects according to input and adjustments. This so-called algorithmic modelling provided the engineer with a tool to explore vast numbers of designs. In architecture, the use of algorithmic parametric design is mainly focussed on the aesthetics of a building and algorithms are set up to create organic lines and natural shapes (Vermeij, 2006). Figure 1-2 shows an example of free form architectural design. In these architectural designs form is preferred over functionality. This is in contrast to civil engineering, where functionality generally takes precedence over appearance. Algorithmic parametric design, which is part of BIM level-3, might be usable to optimize a civil engineering structure. Small optimizations within the design of civil engineering objects could have large benefits.

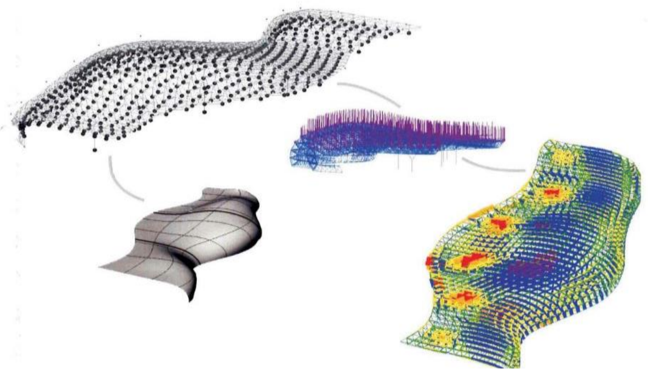


Figure 1-2 An example of a parametric designed structure (Vermeij, 2006)

The process of conventional design, which BIM aims to improve, is linear and iterative. According to Mileham, Currie, Miles, and Bradford (2007, p. 117), in the conventional conceptual phase, “most of the unavoidable costs are locked into the product and up to 50% of the avoidable costs are also generated,” they claim that “the accuracy of a cost estimate improves with the volume of information available” (Figure 1-5). In the conceptual design stage, little of the end-product is known. Knowledge is gained through research, measurements and iterative design processes, from which additional requirements emerge. At every iteration, the design is adjusted to the requirements. In the long run, this iterative process gradually turns a preliminary design into a final design. During the design process, every stakeholder has his own role and level of involvement (Figure 1-3). The design of a structure requires a fine-tuned collaboration between the stakeholders because all decisions and adjustments mutually influence other stakeholders. These changes must be communicated and incorporated to the work of co-workers, which is a labour-intensive process and is prone to mistakes and information loss (Figure 1-4).

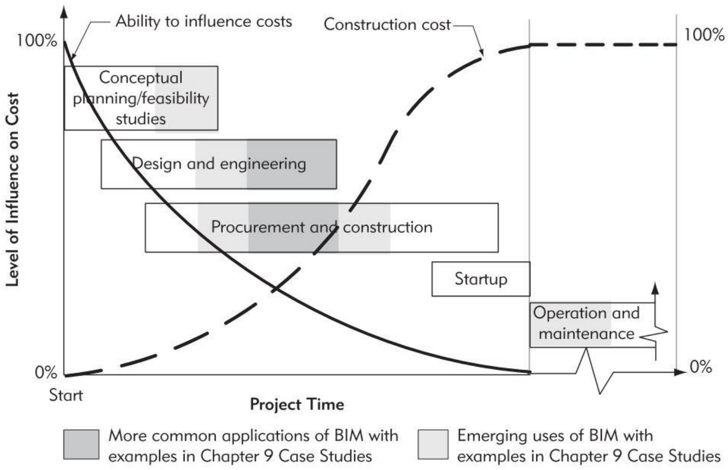


Figure 1-5 Cost influence in every design stage (Eastman, Teicholz, Sacks, & Liston, 2011)

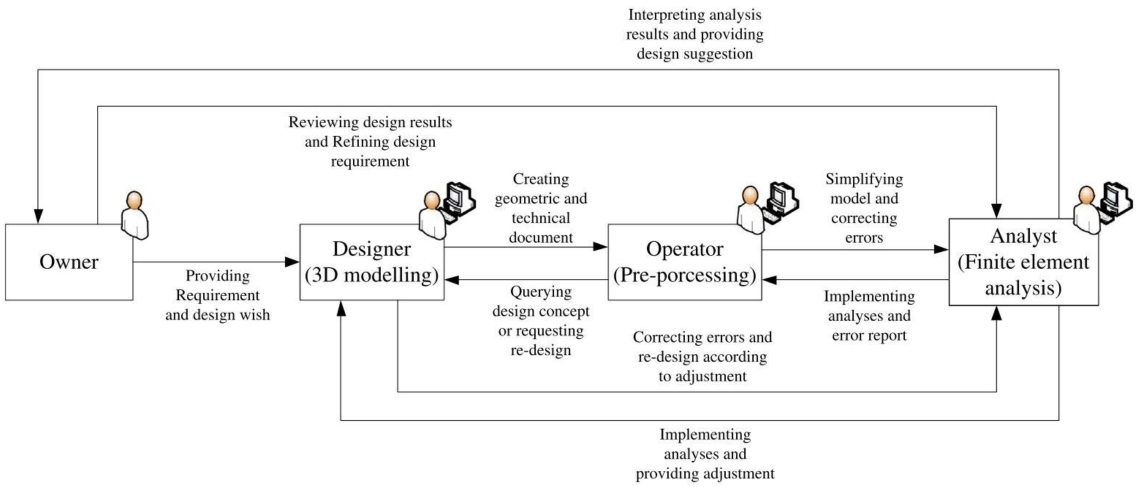


Figure 1-3 Roles in a conventional structural design process (Chi, Wang, & Jiao, 2014)

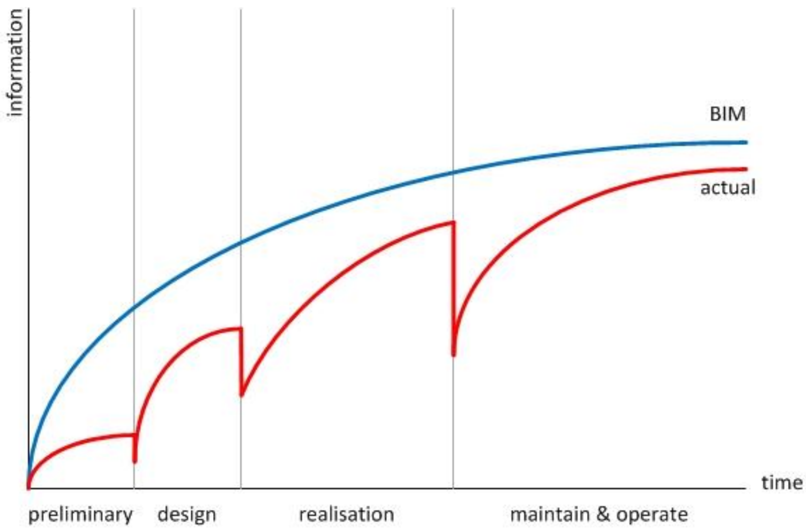


Figure 1-4 Information exchange between design stages (COINS)

At the start of a conventional design process, a tender document is commissioned by an investor (Tol, 2003). Such a document includes a global programme of demands and results of prior conducted investigations on the surroundings and soil conditions. On the basis of a tender document, engineering companies work out many different design alternatives. This must be done to a level that guarantees the structural feasibility. Additionally, the engineering companies must estimate cost, assess the durability and investigate the impact on the surroundings. Based on these estimates, the projected is granted by the investor, to the company with the "best" design. In many civil engineering projects in the Netherlands, the determination of the "best" design is based on an "Economisch Meest Voordelige Inschrijving" (EMVI) *(Economically Most Attractive Tender)* score (Rijkswaterstaat, 2018). The EMVI score is used to find the design with the best cost to quality ratio. The EMVI scores are based on several indicators. EMVI includes among other things: costs, hindrance management, risk management, infrastructure down time and environmental impact. To quantify the environmental impact, Rijkswaterstaat has determined a "Milieu Kosten Indicator" (*Environment Cost Indicator*) value for each building material (Rijkswaterstaat, 2012). This value is used to quantify the environmental impact of a design. This way, the construction may be granted to the company with a relative expensive design with a better cost to quality ratio, compared to cheaper designs.

Many kinds of software packages are used in conventional design, each with different capabilities and purposes. Software for architectural design, structural calculations and software for scheduling and planning are used independently. However, sharing information between these programs is not always possible. "The ability to exchange data between applications is called Interoperability" (Eastman et al., 2011, p. 99). Standardized software formats allow different programs to read data from other programs without translating the file format. Interoperable and standardized data can be combined with parametric design and provide a platform that integrates all data and design processes in a single model. Interoperability and standardization, "smoothens the workflow and facilitates automation" (Eastman et al., 2011, p. 99).

Figure 1-6 BIM technologies

BIM level-3 (Figure 1-6) aims to improve the conventional design processes in the field of construction by adding a new dimension to the parametric design process. In contrast to CAD models and parametric models, BIM level-3 adds an "intelligent" layer to the drawings (Eastman et al., 2011). BIM provides a framework, standardized formats and interoperability between software programs to allow all sorts of data to be integrated and interact with each other. It defines rules to facilitate collaborative design between different contributors and enables stakeholders to work in a single visualized design environment. Collaboratively working within a single model and sharing information from an early stage, increases the involvement of stakeholders.

The complexity and number of automated processes within a model increase the abstraction of the model. According to Harding and Shepherd (2017), many automated systems only show the initial conditions and the end results, such models risk turning into a black box. BIM technologies can help designers to visualize their structural design results and make real-time adjustments in an interactive environment. One of the intentions of BIM is to provide intuitive information (Chi et al., 2014).

If BIM level-3 is fully utilized the prediction is that the computer can be used as a smart tool and make design decisions parallel to those of the designer and engineer. Figure 1-7 shows the history of the development of CAD and BIM. This figure follows a similar trend as Moore's law (Brenner, 1997), which predicts a doubling of computing power every 18 months. Considering this and recent development in the field of artificial intelligence and machine learning, one can ask, what does the future hold?

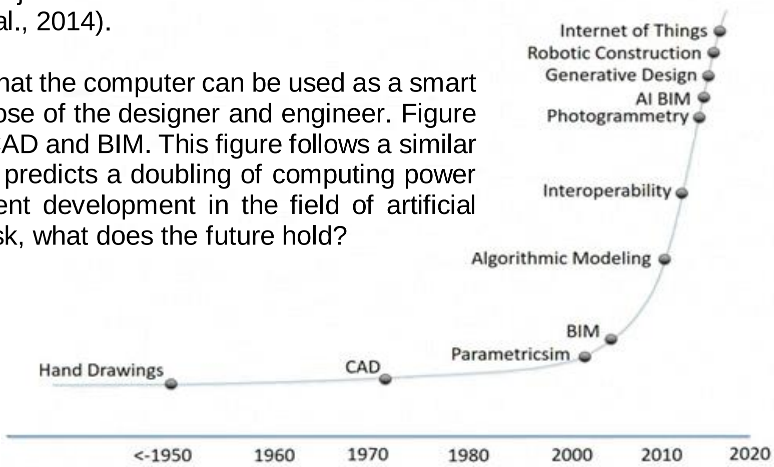


Figure 1-7 The future of BIM, Autodesk, (Allen, 2016)

Although, the problems stated above hold for the entire field of conventual design in civil engineering, the developments of BIM systems are generic. In this thesis, the specific focus is the design of an excavation pit. For many civil engineering projects, the design of an excavation pit is an important aspect of the conceptual design phase. According to Tol (2003, p. 61), “when designing a building pit in urban areas, the most important concern is avoiding inadmissible damage or hinder to adjacent structures,” because the surrounding buildings are owned by third parties and deformations, causing damage to these buildings, happen quick. For an excavation pit, the following boundary conditions are examined (Tol, 2003): (1) Soil layers. In the Netherlands, the data of soil layers, is generally provided in the form of Cone Penetration Tests (CPT). From the CPT the soil properties that influence the excavation pit can be determined. (2) Ground water conditions. These conditions are generally obtained by installed observation wells. From the ground water conditions, the pressures on the excavation pit are derived. Additionally, the ground water conditions influence the construction method of the excavating pit. (3) Surrounding buildings. The vulnerability of the surrounding buildings is investigated based on the type of foundation and the quality of the superstructure. Shallow foundations are susceptible to the deformations of the soil in the upper layers, whereas pile foundations are less susceptible to deformations above the pile tip. Surrounding buildings influence the loads on the excavation pit and are therefore investigated.

Based on the boundary conditions stated above, the design of an excavation pit must meet structural and environmental requirements. Figure 1-8, shows the often-applied structural elements that make up an excavation pit. Each element has its own function in acquiring vertical, horizontal and moment equilibrium. According to Tol (2003), the costs of an excavation pit are mostly determined by the way the groundwater can be kept out. The following three methods are commonly used to avoid groundwater:

1. Retaining walls that extend into the naturally present impermeable layers.
2. Excavating the excavation pit, without pumping out the water, after which an underwater concrete floor is poured.
3. Inject grout or chemicals (silica gel) into the ground to create an artificially impermeable layer.

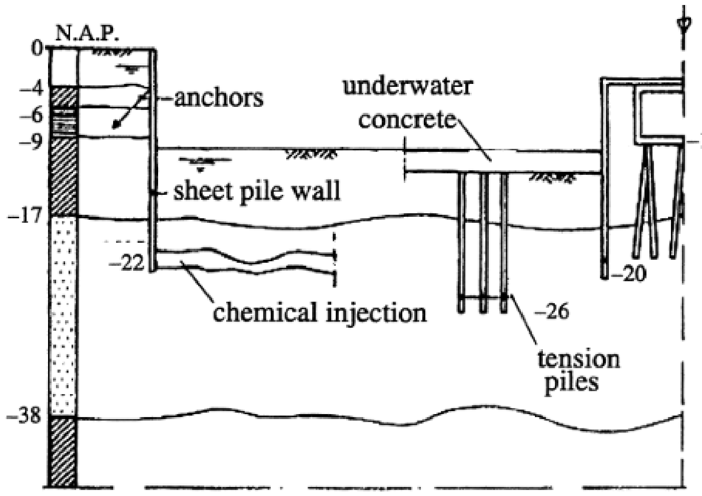


Figure 1-8 Often-applied building elements for excavation pits (Tol, 2003)

For all these methods, the natural or artificially impermeable layer should be at a sufficient depth, that, after the excavation is performed, the weight of that layer and the weight of the soil on top of that layer is in vertical equilibrium with the water pressure. In the case of an Underwater Concrete Floor (UCF), the floor transfers forces to the retaining walls, which is beneficial to the vertical equilibrium. Additionally, vertical grout anchors are generally used to activate the weight of the soil beneath the floor in the vertical equilibrium. According to Tol (2003, p. 72), sheet pile walls, diaphragm walls or pile walls are obvious choices to create a closed earth retaining structure. The equilibria and the magnitude of deformations are determined by the loads, stiffness of the structural elements, installation method and the number of struts or anchors.

1.3 Problem definition

The design process of hydraulic structures or other civil engineering structures is circuitous in many engineering companies. Civil engineering objects consist of many different elements and materials and can be constructed in many ways. With each variable the possible combinations add up exponentially. In conventional design, the amount of possible options that can be evaluated and worked out is limited due to time and labour constrains. As a result, the majority of feasible designs within the design space, a space of all possible human made designs (Bhooshan, 2017), go unexplored, thus the exploration of new and out of the box designs is held back. Among others, this is the result of the complex and manual workflow between the engineers and the stakeholders in a project. Involved stakeholders work in many different models and use different software packages for different analysis. Most often, these applications are not compatible with each other and information must be transferred manually between these programs. On top of that, data and results from analysis are shared and communicated to other stakeholders in the form of reports, pictures and plots. Additional analysis based on this data requires manual reinterpretation by other stakeholders. This process is labour intensive and prone to mistakes and information loss.

Witteveen+Bos also experiences these problems. During the conceptual phase, Witteveen+Bos can currently work-out a handful of different alternative designs. It is difficult to win a tender, because of the large number of competing engineering companies. Innovative and out of the box design are often not analysed because of the lack of time which creates a lot of resistance within the company, against the development of out of the box designs. On top of that, many other possible designs within the design space that could have been cheaper or more environmental friendly with small modifications, are not explored because the design process is labour intensive, and it takes several weeks to work out a design of an excavation pit. Structural and environmental requirements and requirements from the surroundings are time consuming to address because the required data is often scattered over several models, data sets and software applications. Combining this data often happens manually. It is not uncommon that additional boundary conditions, key figures for cost estimates and requirements change because additional ground research is conducted, or new information is made available by the investor or supplier. As result, designs must be reworked halfway the tender phase.

Witteveen+Bos recognizes the potential benefits that BIM systems could have to these problems. They want to develop a BIM level-3 system and experience the potential benefits of such systems in practise. Specifically, they want to improve the design workflow and processes of an excavation pit. In the design of an excavation pit, Plaxis 2D (PLAXIS, 2017), a finite element method application, could be used to computed detailed deformations of the soil and to analyse the stability of an excavation pit. However, setting up such a finite element calculation often consumes too much time and Witteveen+Bos resorts to less detailed calculation software. Additionally, the required data that is needed for such a soil deformation calculation is extracted manually from other software like RockWorks (RockWare, 2018), a program that can be used for the soil properties and the interpolation of CTP's and Geographical information systems (GIS), that can be used to analyse the surrounding buildings and their requirements.

Parametric design and 'smart' designs systems such as BIM level-3 are promising solutions to these problems. By the means of standardization and interoperability, such systems could smoothen the workflow between stakeholders, facilitate automation (Eastman et al., 2011) and improve productivity (Sacks & Barak, 2008). According to (Xue, Shen et al. 2012) new information technologies have great potential to develop computer and internet supported design systems. However, the adaptation by the industry including Witteveen+Bos to these systems is lacking. As of today, only one company in the world is BIM level-2 certified. Additionally, many studies focus on single technologies enveloped by BIM and do not grasp the benefits of integrating all technologies into one tool.

1.4 Thesis objective

"Develop a conceptual, partly BIM level-3, integrated parametric design tool that assists in the exploration of an extended design space of an excavation site, in the context of the surroundings and environmental requirements and visualize the designs and their differences to help the stakeholders in comparing and analysing different designs in the conceptual design phase".

1.4.1 Development steps

For the development of this integrated tool the main objective is divided into five development steps.

1. Develop a design tool that integrates parametric design with external specialized software, through interoperability and standardization.

2. Develop an algorithm that is suitable in different projects and can analyse many feasible designs within an extended design space of an excavation pit, bounded by the requirements and limitations posed by the surroundings, environment and the stakeholders.

3. Develop a suitable digital environment for an interactive collaborative design process between a group of engineers and the stakeholders. This environment must present in a clear, interactive manner, the feasible and non-feasible designs and their pros and cons by visualizing the differences between the designs within the design space.

4. Provide visualizations of the results from computations performed by integrated external specialised software to contribute to a better understanding of the design limitations and possibilities.

5. Validation of the design tool by testing the impact of additional requirements, objects and surroundings to the usability of the model.

The integration of these developments steps into one model, leads to the objective of this thesis. (1) The integrated specialized software automates the input of the design tool. This includes the boundary conditions which are based on the surroundings and the environmental requirements. Additionally, the integration of interoperable calculation software is used to assess structural requirements and soil deformations. (2) A design algorithm explores the design space of the excavation pit, by analysing the possible designs bounded by the boundary conditions and the user input obtained from the digital environment (3). Additionally, the design algorithm controls the interoperable specialized calculation software. The digital environment, visualizes the feasible and non-feasible designs to help the stakeholders compare and analyses the different designs. (4) To do so, the design tool must provide sufficient data to the stakeholders by visualizing the results of the computations and interoperable software. (5) The design tool is tested and validated to reinforce the results and trust in the design tool by the stakeholders.

1.5 Method

For the development of the design tool of this master thesis, a suitable approach is needed that maintains a critical view on the development without going through the full development of several full systems. For this reason, rapid prototyping is chosen (Figure 1-9). This method is useful for the development of a prove of concept (Babich, 2017). The conceptual design tool will be programmed, tested and reviewed and the outcome of the tests will form the basis for the refinement and iterations. When each of the development steps has been executed, they are integrated into a single design tool. This will conclude the prototyping process. Afterwards the integrated design tool is tested and reviewed, and the outcome is used for refinement and iterations.

1. Prototyping (Chapter 2)
   - Literature study on the topic of each development step
   - Prototype according to each development step
   - Test and refine the results of the prototype of the development step
   - Integrate the development steps into one model. This integrated model is called SET

2. Test & Review (Chapter 3)
   - Test the design tool with additional, requirements, objects and different surroundings
   - Test the user-interface of SET
   - Review the user-interface, usability and the test results of SET

3. Refine & Iterate (Chapter 4)
   - Apply the necessary refinements based on the tests
   - Define the long-term refinements as future recommendations.

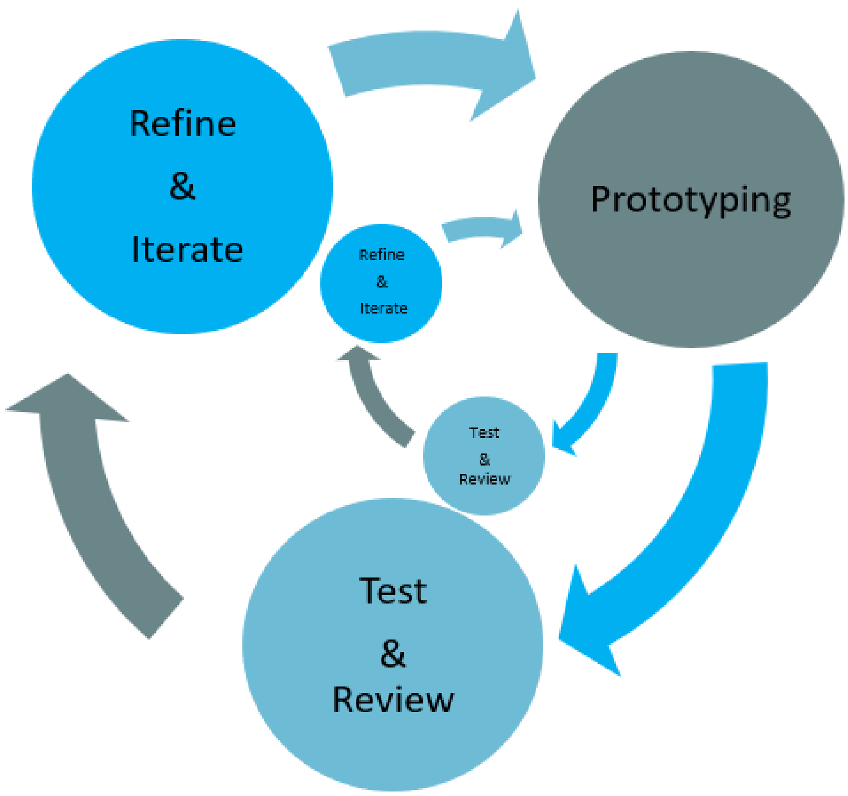


Figure 1-9 Rapid prototyping

The structure of this report is based on rapid prototyping. For this reason, rapid prototyping steps correspond to the headings of Chapter 2, 3 and 4, which is shown in Figure 1-10. In Chapter 2, the prototyping or development of the tool is discussed. Development steps one to four are covered in the three subchapters (2.1 Interoperability and standardization; 2.3 Design algorithm; 2.3.5 Collaboration, interaction and 3D visuals). Each of these subchapters are developed in a small rapid prototyping loop. Subchapter 2.5 covers the integration of the subchapters stated above into one model, which is needed to achieve the main thesis objective. The result of this integration is tested and reviewed in Chapter 3. This chapter corresponds to development step five. In that chapter it is discussed how the tool is tested and reviewed. In Chapter 4, the future refinements and iterations to the design tool are discussed. The refinements and iterations correspond to the recommendations and discussion about SET. In Chapter 6 a conclusion is given, followed by a discussion on the process and execution of this thesis, finally recommendations are given to the industry.

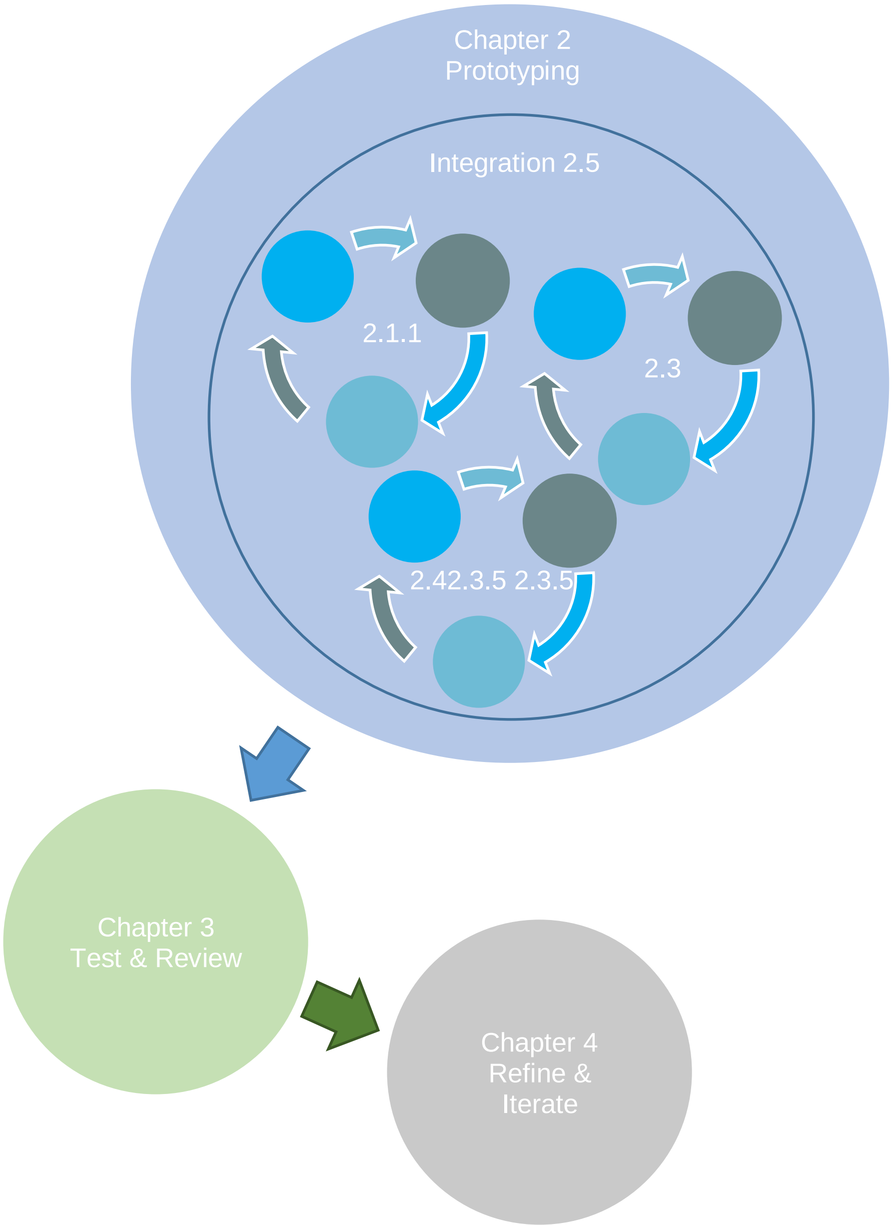


Figure 1-10 Report structure Based on Rapid Prototyping, chapter 2 is based on small rapid prototyping cycles.

1.6 Scope

The scope of this thesis starts wide, to gain a general understanding of BIM technologies, design processes and design tools. The wide range includes: design tool features, design algorithms, design processes, design phases, programming languages, different user interfaces, 3D visuals and 3D object manipulation possibilities. Based on the general understanding the scope is narrowed down, to gain an extensive understanding of design processes of an excavation pit, the conceptual design phase, tender processes and specific BIM technologies, data management, design tool features, user interface, 3D visuals and 3D object manipulations techniques. The development of the design tool is based on the specific knowledge gained within this narrow scope. A schematic representation of the scope is given in Figure 1-11.

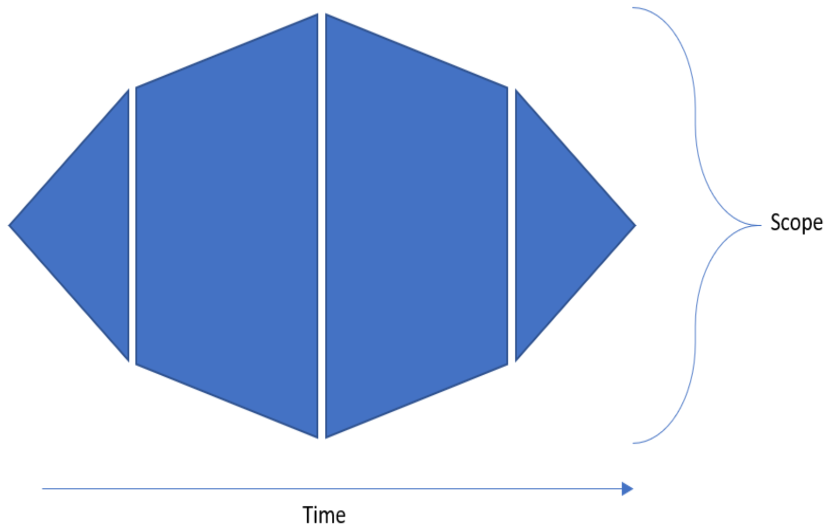


Figure 1-11 Scope throughout this thesis

During this thesis, Witteveen+Bos worked on a tender project for the design of a bus lane next to the existing train tracks, called “Hoogwaardig Openbaar Vervoer in het Gooi” (high quality public transport), in short “HOV in ‘t Gooi”. The tender document of the currently level intersection of the bus lane with the ‘Oosterengweg’ is used during the development of SET.

1.7 Development Limitations

Developing a BIM level-3 model capable of integrating the many aspects of design processes is complex. For the limited amount of time available in this thesis the complexity and difficulty are reduced. This chapter addresses the assumptions and simplifications that are made for this thesis. A balance between the simplifications and the possibility to prove that the integration of design technologies is possible and useful, is made and approved during the kick-off meeting.

BIM level-1 and -2

BIM level-1 and -2 form the basis of a BIM level-3 design tool and because this thesis focuses BIM level-3 and on the integration of many BIM technologies, the support of these BIM technologies is simplified.

Interoperability and Standardization

The design tool must feature capabilities to perform a structural analysis and investigate soil deformations. For this reason, interoperability Plaxis 2D is required. The model must be able to provide Plaxis with input. Automate the calculation process and reuse its output. Additionally, the developed design to must draw conclusions from the Plaxis output.

The BIM standard Industry Foundation Classes (IFC) is complex and comprehensive. IFC is a fully developed standard and will require much insight into the format. As the focus of this thesis is integration of the many BIM technologies, and to reduce the complexity of this thesis, the use of the IFC standard is neglected. However, because standardization is one of the main goals of BIM level-1 and level-2, simple and commonly used file formats are used to standardize the data. The design tool will extract soil layers and their properties from RockWorks models. The surrounding buildings are manually extracted from GIS files. After the extraction of these boundary conditions, the data will be standardized and used by the design tool.

Excavation site

The design tool will be developed for an elongated excavation pit. This includes excavation pits with a horizontal or sloping floor. This includes, land tunnels, open tunnel and cannels. The design tool will neglect the short sides of the excavation pit and transitions between two consecutive excavation pits.

Conceptual design phase

This thesis will only focus on the sketch/conceptual phase of a project. This is one of the early design phases. This thesis will not consider the construction phase or detailed design phase. Accordingly, the functional requirements of the to be constructed object, are simplified.

Calculation cross-sections

Along the length of an elongated excavation pit, especially for an excavation pit with a sloping floor, the boundary conditions and requirements from the environment and surroundings can differ. To capture these different boundary conditions, the user must be able to define 2D calculations cross-sections at normative locations along the length of the excavation pit. The design algorithm must consider these normative cross-sections and perform detailed calculations at these user-defined locations.

Structural objects

Three different building elements of an excavation pit will be taken into consideration: Walls, floors and supporting structures.

- Walls: sheet piles and diaphragm walls
- Excavation types and floors: dry with no floor, wet with UCF, wet in combination with gel injections
- Supports: anchors and struts

Combinations of these building elements and differences in their length can be possible. For wet excavation an impermeable layer for the vertical equilibrium needs to be constructed. SET supports underwater concrete floors and gel injections.

Structural checks by the design algorithm
The structural checks that will be coded into the design algorithm are the following:
- Walls: unity check on the moment capacity in the ULS and SLS based on the results of Plaxis 2D
- Excavation types and floors: vertical stability with the water pressure.
- Supports: the supports are only checked for geometrical recruitments.

The vertical stability of an excavation pit will be calculated with rules of thumb and balance equations.

Structural checks by Plaxis
The deformations of the soil and moments in the retaining walls are calculated with Plaxis. Plaxis performs detailed Finite Element Method (FEM) calculations. From these calculations follow the forces and moments in the structural elements. These are used in the unity checks of moments in the retaining walls. The resulting soil deformations of Plaxis are used for the settlement requirements of the surrounding buildings. The design algorithm uses these deformations to check the relative rotation and horizontal strain of the surrounding buildings.

Interaction with surroundings
The interaction of the excavation pit with, and its effects on, the environment and surroundings corresponds to the three boundary conditions of an excavation pit (Tol, 2003). The tool must be able to interact with the surrounding buildings, ground layers and water level. The interaction with buildings is limited to their settlement requirements and whether anchors can be placed underneath them. The distance to the excavation site of a building can also have a minimum.

Settlement requirements
For the surrounding buildings, the horizontal strain and relative rotation will be examined. The design tool does not make a distinction between different foundation types of the surrounding buildings and assumes shallow foundations.

3D visuals
In this thesis a very basic 3D environment will be constructed. Simplified representation of environment, surrounding buildings and structural objects will be used.

Level of detail
For this thesis the Level of Detail 000 is used (Het Nationaal BIM Platform, 2010). Simple shapes and boxes represent object. Object can contain non-geometric information.

Cost Estimates and Quantification
For the quantification of each design, the costs supplied by Witteveen+Bos will be used. The environmental impact of a design will be quantified with the “Milieu Kosten Indicator” (MKI) (environmental cost indicator) value and material volumes.

Optimisation within infinite combinations
It is assumed that tools to optimize more complex structures will be available in the future. Either by using artificial intelligence and machine learning or through faster computers and quantum computers which can calculate the many combinations in an acceptable amount of time. For this reason, an optimisation will be part of this thesis, however, this will be done with a brute force method combined with smart use of computation power and by reducing the amount of possible designs within the design space.

2
Prototyping

The developed tool in this chapter is called SET, Smart Engineering Tool. As discussed in the thesis method, the development steps of the tool are divided over four chapters. The four sections are the following:

1. Interoperability and Standardization
2. Design algorithm
3. Collaboration, interaction and 3D visuals
4. Integration into one model

In parts one to three, first, some design choices are discussed. These choices were made based on literature or to reduce the complexity of this thesis or because of the limited amount of available time. These choices correspond to the development limitations of this thesis as discussed in Chapter 1.7. Some other choices were made, to give me, as a researcher, a better understanding of programming. Materials that were investigated but not used in the design tool can be found in Appendix A .

During the development of the design tool, tender data from a project in Hilversum called “Hoogwaardig Openbaar Vervoer in het Gooi” (high quality public transport), in short “HOV in ‘t Gooi”, is used. This project is used in the test sessions. The project is a tender for a design of a bus lane next to the existing train tracks. The data used in the development of the design tool is based on the currently level intersection with the ‘Oosterengweg’, as shown in Figure 2-1.

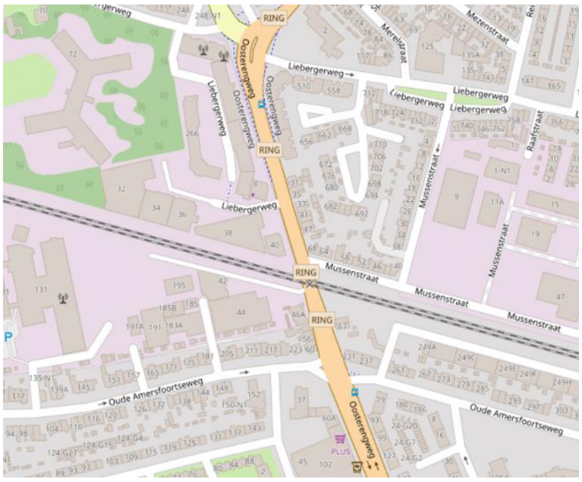


Figure 2-1 Intersection 'Oosterengweg' and the new bus lane (GeoJSON, 2018)

2.1 Development criteria

In this section the criteria, based on the objective and development steps, which the developed design tool should satisfy, are elaborated upon. These criteria are used for the development of the tool in the chapters below. Additionally, conclusions are drawn based on these criteria at the end of this chapter and in Chapter 3.

Interoperability and standardization

a. External specialized software can be called interoperable if the design tool can provide the software of input and use the output in a dynamic manner. The integration does not need to be seamless and fully automatic.
b. Standardization is met if the same data format is used for similar data.

Design algorithm

c. The algorithm is suitable for different projects if it is usable for different projects with different requirements and surroundings.
d. Designs are feasible if they meet all the criteria set by the requirements and limitations posed by the surroundings, environment and the stakeholders.

Collaboration, Interaction and 3D Visuals

e. A digital environment is suitable and interactive if it can clearly visualise design limitations, decisions, possibilities and differences between variants without turning into a black box.
f. The tool is suitable for collaborative design if the same model can be accessed and changed remotely by a group of engineers or stakeholders.
g. Next to graphs, the results of integrated external software should be visualized in a 3D manner.

Integration

The result of the integration of the three parts above is the design tool. This design tool should satisfy the main objective. The following criteria are derived from this main objective:

h. The tool can be called partly BIM level-3 if the tool can make ‘smart’ design decisions in the context of the environment and the surroundings without interference of an engineer. The data should be integrated and interoperable.
i. The requirements of BIM level-1 and level-2 should be met. For BIM level-1 the design tool should support 2D and 3D graphic and is object oriented, every object should contain information about that parametric structural object. For BIM level-2, collaborative design should be possible. The data must be shared from, and saved in, a central repository and the data managed by the model should be interoperable and based on standards.
j. The design space that the tool explores can be called extended if less obvious design solutions are explored.
k. The context of the surroundings and the environmental requirements is sufficient if it is based on some limiting factors of surrounding buildings and when the design tool can include them in calculations and the outcome.

2.1.1 Boundary conditions of an excavation pit

According to Tol (2003), three boundary conditions can be distinguished for the design of an excavation pit. The extent to which these boundary conditions are considered by the design tool are discussed below.

1. Soil layers
   A realistic 3D geometry of the soil layers must be supported. This includes, variations in thickness, the depth and a soil layer can be impaired.

2. Water levels
   Water levels can be assumed equal over the whole project area. A table with water pressures of a single borehole suffices.

3. Surrounding buildings.
   The surrounding buildings influence the loads on the excavation pit and impose settlement requirements on the deformations of the soil which are caused by the construction of the excavation pit.

The developed design tool must be able to process these boundary conditions. The user must be able to manually add these to the tool or the tool must be able to extract these for models or databases.

The design tool must take the following into consideration:

1. Horizontal equilibrium of the full excavation pit
2. Vertical equilibrium of the full excavation pit
3. Moment equilibrium of the full excavation pit
4. Maximum internal bending moment of the retaining walls

2.2 Interoperability and Standardization

In this subchapter the development of development step 1 is discussed.

Develop a design tool that integrates parametric design with external specialized software, through interoperability and standardization.

Detailed information and flowcharts about how the tool is programmed can be found in the appendices.

2.2.1 Interoperability, Plaxis 2D

Deformations of the soil around the excavation pit can be calculated with the finite element method software called Plaxis 2D (PLAXIS, 2017). This program is often used by Witteveen+Bos in detailed design phase and not in the conceptual design. The soil deformations have an impact on the surrounding buildings. The results of the finite element calculations are used to check whether the requirements from these buildings are met. Additionally, the stability and equilibria of an excavation pit is checked with Plaxis 2D. For the reasons above, Plaxis 2D is incorporated into the design tool. Plaxis 2D is known for its user-friendly integration with Python. Additionally, Python is one of the most used data processing languages. For this reason, the interoperability of Plaxis is written in Python.

What calculations Plaxis performs is determined by the design algorithm which is discussed in Chapter 2.3. For the design of an excavation pit, the design tool is equipped with the support of several functionalities and objects of Plaxis. For each object, Appendix B.1 shows the supported properties. These objects are:

- Retaining walls
- Struts
- Rotational connections between struts and the wall
- Anchor body
- Anchor rot
- Soil layers
- Impermeable layers

To run a calculation, next to these objects, Plaxis uses construction phases. In each phase, Plaxis activates or deactivates objects or soil layers. An example is shown below. Between each construction phase a new finite element calculation is performed. SET, supports 10 different sets of construction phases by default. These 10 sets are hard coded into SET but can be changed manually. These sets correspond to the used structural objects and define the order in which Plaxis performs calculations. The 10 sets of phases can be found in Appendix B.2. An example of these sets is given below, between brackets is the operation that Plaxis performs, after each phase in such a set, a new Plaxis finite element calculation is run. In this way, each normative situation during construction is analysed.

Phases of a dry excavation pit with no supporting structures:

1. Initial phase, (soil properties and water levels are activated)
2. Installation retaining walls. (retaining walls are activated)
3. Drain bottom, (water level above the bottom of the pit is deactivated)
4. Excavate bottom, (the soil above the bottom of the pit is deactivated)

For each calculation the geometry of each structural element and their properties is determined by the design algorithm. It defines the soil layers and water levels and defines which set of phases it should use. All this information is collected for a large number of calculations and are send to a Python script developed by Witteveen+Bos. For these calculation five types of files are manually transferred to the Plaxis computer:

1. Libraries containing the objects and soil layer properties shown in Appendix B.1
2. The calculations for each cross-section, this file contains the geometry and defines what phasing set and what objects should be used in each calculation.
3. A file containing the calculation order, which prioritises some calculation. In this way, the user can assess results when other calculations are still running.
4. A file containing the surface loads of the surrounding buildings.
5. A construction phases file, this file contains the 10 sets of construction phases.

On the Plaxis computer a script opens these files and executes the following:

1. Open the calculations order file
2. Loop (For each calculation in the calculation order):
    a. Open: the correct calculation of the corresponding cross-section, given by the calculations order
    b. Upload: the objects with their geometry and properties according to that calculation to Plaxis
    c. Add: a corresponding construction phase in Plaxis
    d. Add: surface loads in Plaxis
    e. Create FEM mesh in Plaxis
    f. Start: calculation in Plaxis
    g. Save: the results of Plaxis
    h. Start: at (a.) with the next calculation

From this loop, SET does not use all output which is generated by Plaxis. All output files of a single calculation could take up to 200MB and with a set of around 400 calculations (the amount of calculations during test case Oosterengweg (Chapter 3.1.2) this would take around 16GB. For this reason, it is chosen to only save and transfer the files that the design algorithm uses. For a single calculation, together the files together use around 6MB of data. These files include:

1. Soil deformation
2. Wall deformations, stresses and moments
3. Support forces

An example of a Plaxis calculation is shown in Figure 2-2. How the output is used by SET is discussed in the next chapter.

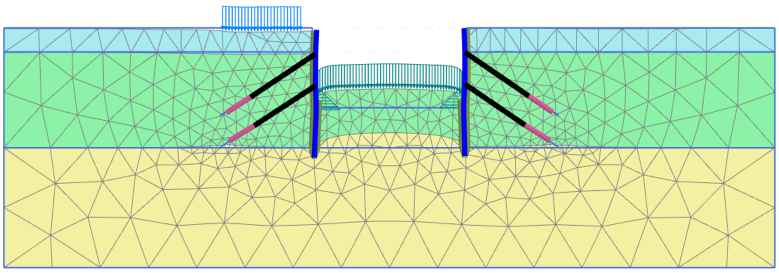


Figure 2-2 Plaxis calculation example

For now, file transfers from one computer to the Plaxis calculation computer and back, requires a manual operation. This is due to the lack of a personal Plaxis user licence. Such a license could not be provided for my personal computer on which the server was running. Otherwise the Plaxis calculations could run on the same computer and would have made the integration fully automatic.

2.2.2 Standardization

For the input of soil layers and properties, the tool has been made compatible with a file format from a program called “RockWorks”. RockWorks is a specialised software package that can analyse CPT’s and interpolate soil layers and define their properties. RockWorks is able to analyse the full topology of soil layers. This means that soil layers can have variations in thickness, depth and soil layer can have holes in it. The files from RockWorks can be saved in a “ESRI ASCII” file format. A script has been incorporated into the design tool that can read this file format and translate it into usable data. Python transforms the file format into a “Wavefront .obj” and stores the data internally. The “Wavefront .obj” is used for the visualization which is discussed in Chapter 2.3.1. The internally stored data is used by Python to provide the design algorithm of input. Appendix B.5 shows how the design algorithm extracts the data for the Plaxis calculations. RockWorks is not interoperable with SET in any other way. It will only supply ‘passive’ input of the ground layers.

For the environmental aspects that are loaded into the tool, Geographic Information System (GIS) is used. From GIS files the surrounding buildings and their requirements are extracted. The first time SET is used, the contours of the buildings need to be traced manually and an arbitrary height must be assigned to the 2D representation of buildings. This can be done in SketchUp, see Figure 2-3. Afterwards SET converts these files into a “Wavefront .obj” file. The building properties are stored in a “Comma Separated Values” (CSV) file. This must be done once and manually. SET stores these properties and afterwards this data can be read, changed, and used automatically. The choice for this file format is made because the CSV file format is widely available and is written in the plain text format ASCII. ASCII can be read and written on any computer without specialised software. The relative rotation, horizontal strain and loads of the surrounding buildings can be added manually after the buildings are loaded into the model. This is possible via the input table as described in Chapter 2.4.2.1.

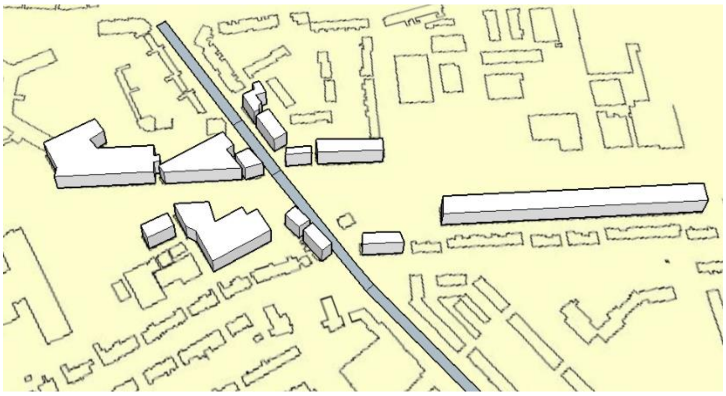


Figure 2-3 From 2D to 3D in SketchUp

2.3 Design algorithm

This chapter will elaborate upon development step 2:

Develop an algorithm that is suitable in different projects and can analyse many feasible designs within an extended design space of an excavation pit, bounded by the requirements and limitations posed by the surroundings, environment and the stakeholders.

The design tool analyses the designs within the design space of an excavation pit based on the input of the user and the boundary conditions. As discussed in Chapter 2.2.2, the input from the boundary conditions is extracted from GIS and RockWorks models. The user has several additional input possibilities, some of these inputs, like costs and material properties can be changed by editing a value. The to be designed object can be defined by loading a 3D model into design tool. From this input, the design tool is able to analyse an extended design space within a relatively small amount of time. To explore an extended design space, a computer algorithm is developed and incorporated into SET. The design algorithm in the back-end of SET, is written in Python and is combined with parametric design. “Parametric design tools can process data and change surfaces and structures accordingly” (Phillips, 2010, p. 32). By interacting with the parametric model, the design algorithm can change the objects within the model. With this analysis the design tool should produce information which helps the stakeholders find an optimal design. “In large-scale structural optimization, the number of design variables and constraints will be very large” (Guo & Cheng, 2010, p. 819). Some of the different methods in finding an optimal design in a set of infinite possibilities are:

1. Brute force, a ‘dumb’ algorithm that examines every combination within a given range.
2. Mathematically
3. Machine learning
4. Artificial intelligence

Machine learning and artificial intelligence are becoming useful and powerful algorithms to find an optimal design. However, these algorithms are complex, and the use of these algorithms could be a thesis topic on their own. The use of these algorithms goes beyond the objective of this thesis. A mathematical algorithm is useful to find a single optimal design in an efficient manner (Kwak & Haug, 1976). However, the answer to an optimal design, is something subjective and the nuance lies in the details of the full context. In the full context of the environment, surroundings and the stakeholders, an optimal design is difficult to find. For one, the definition of optimal is different for each stakeholder or point of view, thus a single solution to an optimal design is ambiguous. For this reason, the design tool will not provide a single optimal design. Instead, it shows the difference between each design based on the structural integrity and quantifications, and helps the stakeholders explore the many design possibilities. To do so, SET allows users to dynamically add requirements, from these requirements, SET must be able to show the difference between each design within the design space. The benefit of a brute force approach is that it will identify all feasible designs within the given design range. SET only excludes structurally unfeasible designs. This allows stakeholders to evaluate many design within the given ranges.

2.3.1 Calculation cross-sections

The design tool bases its calculations on user defined calculation cross-sections, an example is shown in Figure 2-4. By default, the width of a calculation cross-sections is set 100m wide. This width is also used for the Plaxis calculations and can be changed by the user. The number of cross-sections influences the details of a design. For example, when only one cross-section is used, the corresponding design will be used for the whole excavation pit. For each additional cross-section a more detailed design is created, however the amount of Plaxis calculations increate linearly and the amount of combinations between cross-sections increase exponentially. As discussed in Chapter 2.3.4. a maximum of 7 cross-sections was set, this way, the computations can stay within a day's worth of Plaxis computation time. For a tunnel with a slope at both sides, at least six cross-sections were found to have a smooth transition from the lowest point to both ends of the tunnel.

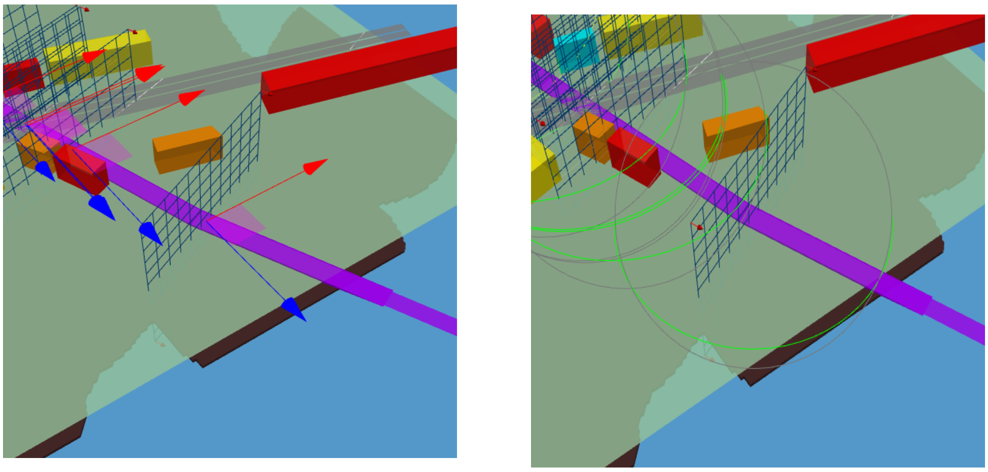


Figure 2-4 User defined calculation cross-sections, left: draggable, right: rotatable

The design tool is equipped with an “align” button, when pressed (Appendix B.4), the calculation cross-sections are positioned exactly in the centre of the to be build object and the angle is set perpendicular to the object, then, for each defined calculation cross-section, the algorithm starts producing possible designs. Each design is checked for several points, these points are discussed in Chapter 2.3.2 and 2.3.4.

2.3.2 Quantification

Each design is quantified by SET based on three values. These values can be defined by a user in a CSV file. Table 2-1, shows how these values are stored in a CSV file:

1. Cost estimates
2. Material volumes
3. MKI values

The cost estimates are based on material cost, and the mobilisation cost. In the material cost, the construction cost is included. The total material cost is based on the volume or running meter of a building material. The volume or the amount of running meters is obtained from the 3D model. The values for the costs can be set by the user accordingly. The mobilizations cost depends on the transport and the assembly of equipment before it is operational. This applies for example to large canes and specialized equipment. A value can be set by the user. The user can define what structural elements use the same equipment, if so, the design tool will use the mobilisation cost only once. For example, when two different kinds of sheet piles can be installed with the same crane the mobilisation costs of that crane will only be used once.

For each structural element, the material volumes are derived from the internal design algorithm of SET.

MKI values are part of the Dubocalc method developed by “Rijkswaterstaat” (Rijkswaterstaat, 2012). These values are used to indicate the impact of material usages on the environment. Dubocalc is part of the EMVI scoring system. For each building material a MKI value for the volume or a running meter can be set by the user.

Material Name	Mobilisation costs	Mobilisation group	Material costs	MKI	Unit

Table 2-1 User input costs and MKI

2.3.3 Pre-processing

The flowchart of the design tool is shown in Chapter 2.6, Figure 2-39. The pre-processing part of the algorithm is the design algorithm that generates the input for Plaxis. This part of the design algorithm considers environmental requirements and requirements from its surroundings. To do this, first, information is extracted from the environment. This process is elaborated upon in Appendix B.6 and B.7. In the following chapter, the functionalities of these corresponding scripts are given in a short textual description. Because different coordinates systems are used by each programming langue, some pictures are mirrored or upside down in this chapter. The different coordinates systems are discussed in Appendix B.3. Because only the front-end in JavaScript is normally visible to the user, the Python coordinate systems of the plots that are only used for in this report are not converted.

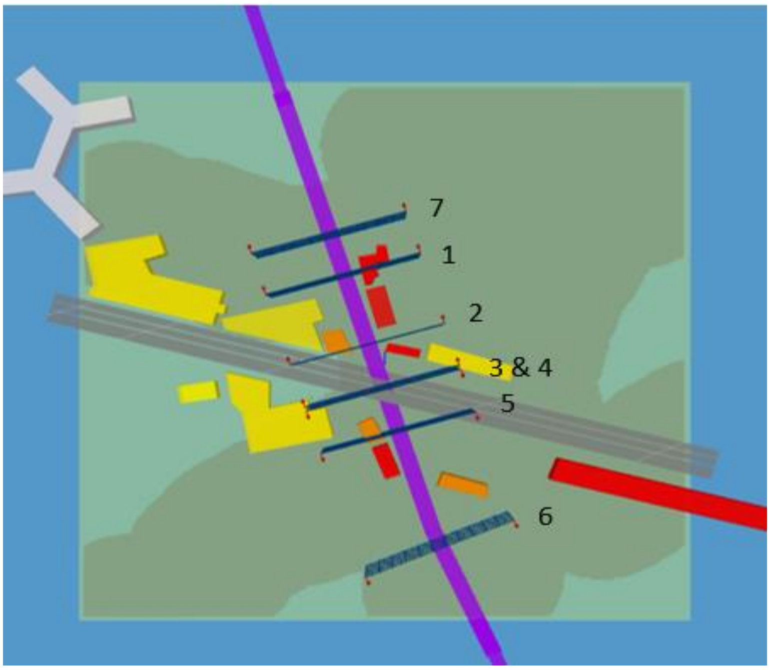


Figure 2-5 Calculation cross-sections (JavaScript)

Ground layers

For each calculation cross-section (Figure 2-5) the ground layers underneath a calculation cross-section are extracted by SET (Appendix B.6) from the ESRI ground layer files (Figure 2-6). This is done with a nearest point interpolation and an example of the results are shown in Figure 2-7. Note that cross-section two starts at the value of around 30, this is because no soil is located underneath the first 30 meters of the cross-section.

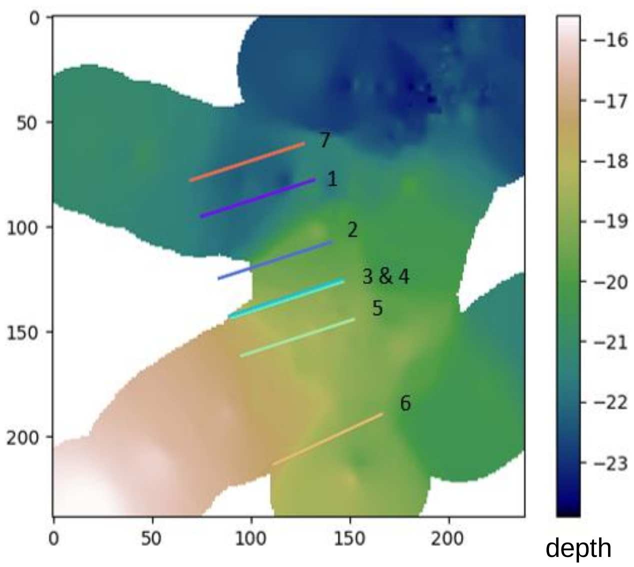


Figure 2-6 The ESRI ground data (clay layer) (Python Plot)

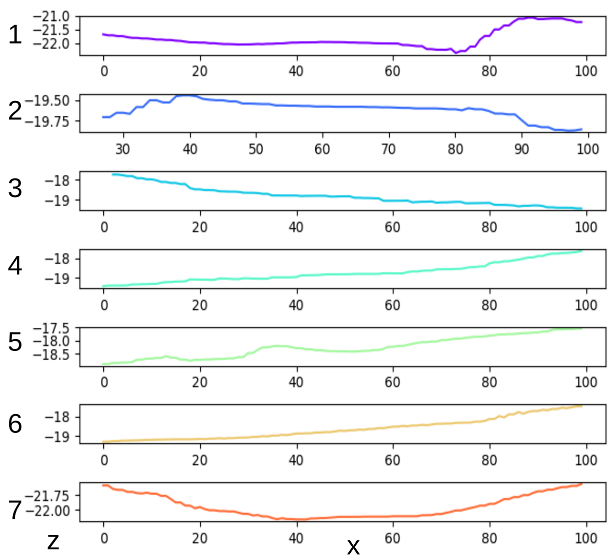


Figure 2-7 Extracted depth values for each calculation cross-section. Normalized coordinates

Surrounding buildings

The information from the surrounding buildings are extracted by SET. A detailed elaboration can be found in Appendix B.6 and B.7. To do this the cross-sections are turned into an imaginary line. The edges of the shapes of the surrounding buildings are read into Python and are turned into an imaginary line as well. A script determines for each of these lines the intersection point and whether this intersection point lies within the boundaries of the building edges. An example of the resulting intersection points is shown in Figure 2-8. For each intersected building, Python stores the corresponding maximum relative rotation, horizontal strain and the load of that building. Afterwards all intersected points are normalized to the coordinates system of the calculation cross-sections (Figure 2-9). This normalized data is used for the geometry of the building loads in the Plaxis calculations. It is also used to determine if anchors are not positioned underneath them. The post-processing also uses this data to determine the relative rotation a horizontal strain of the buildings as discussed in Chapter 2.3.4.

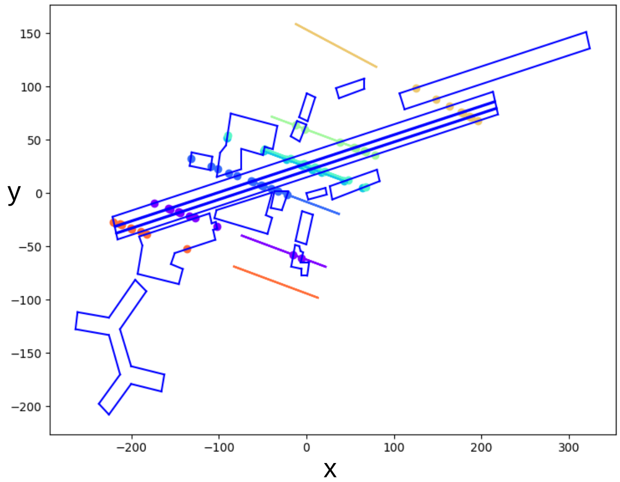


Figure 2-8 Intersection point of the buildings on the calculation cross-sections. (Python Plot)

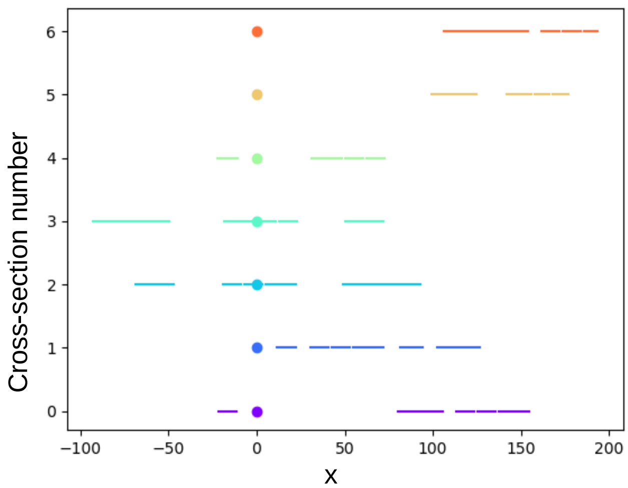


Figure 2-9 The intersected buildings. Normalized x for each calculation cross-section (x in 1D) Python goes from 0 to 6 instead of 1 to 7 (Python Plot)

Building pit extraction

The last data extraction is the data extraction of the building pit. This process is elaborated upon in Appendix B.7. The process of this data extraction is similar to that of the surrounding buildings. Additionally, the following information is extracted from the excavation pit:

1. The x and y coordinates of the outer edges where the retaining walls need to be constructed
2. The depth of the construction pit
3. The slope of the floor
4. The corresponding pit length for each calculation cross-section

The x and y coordinate extraction are similar to that of the surrounding buildings as discussed above. For the dept of the building pit and the slope of the floor the following operations are performed.

a. The begin point and end point, of the intersection point between the edges of the intersected building and the calculation cross-section are determined
b. The deepest points of the downward facing edges are determined
c. The distance ratio is set equal to the depth ratio of these point to find the required depth of the excavation pit at the location of the intersected point
d. From the two deepest points the slope of the excavation pit floor is determined with the distance and depth ratio

An illustration of this process is given in Figure 2-10.

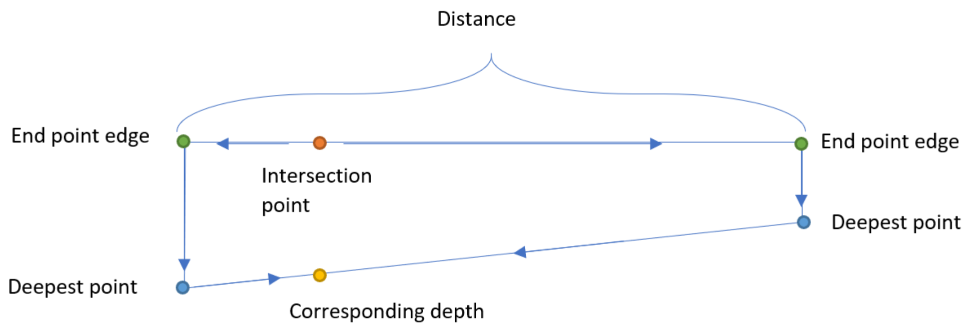


Figure 2-10 Illustration of the depth finding process, longitudinal cross-section of a tunnel

For the pit length the design algorithm must consider three things. First, the length of that particular cross-section must be in the same direction as the direction of the indication arrows (Figure 2-11). Secondly the design tool checks if there is a calculation cross-section in that direction. If not, the length of the pit for that calculation cross-section must stop at the point where the tunnel surfaces. This is achieved with a similar process as the floor slope. For example, the pit of cross-section 7, in Figure 2-12 (next page), has to stop at the surfacing point of the tunnel.

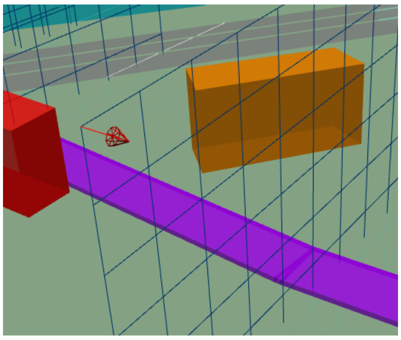


Figure 2-11 Direction indication arrow

Thirdly, if there are calculation cross-sections in the direction of the indication arrows, the pit length must be the distance between two calculation cross-sections. However, to make the design tool suitable for different projects, the user is not forced to add the calculation cross-sections in a particular order. The design tool first finds which cross-sections it meets in the direction of the arrows. From these cross-sections the closest cross-section is determined. For example, the length of the excavation pit for cross-section 1 in Figure 2-12, corresponds to the distance between cross-section 1 and 7. The results of this process are shown in Figure 2-13.

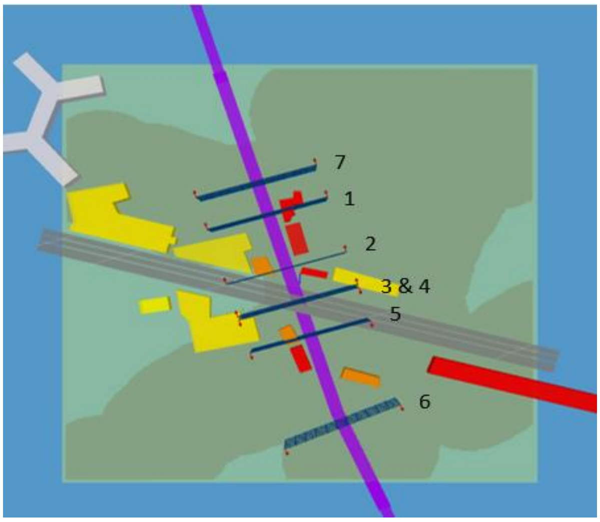


Figure 2-12 Calculation cross-sections (JavaScript)

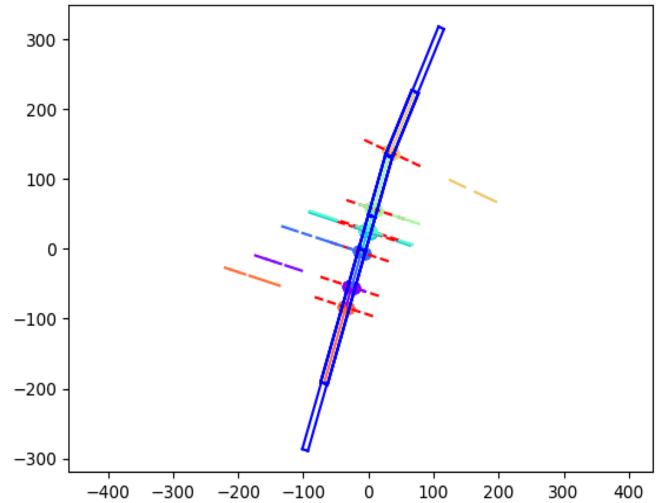


Figure 2-13 Extraction of the excavation pit data. (Python Plot)

In Figure 2-14, the extracted data as discussed above is shown in the 3D environment.

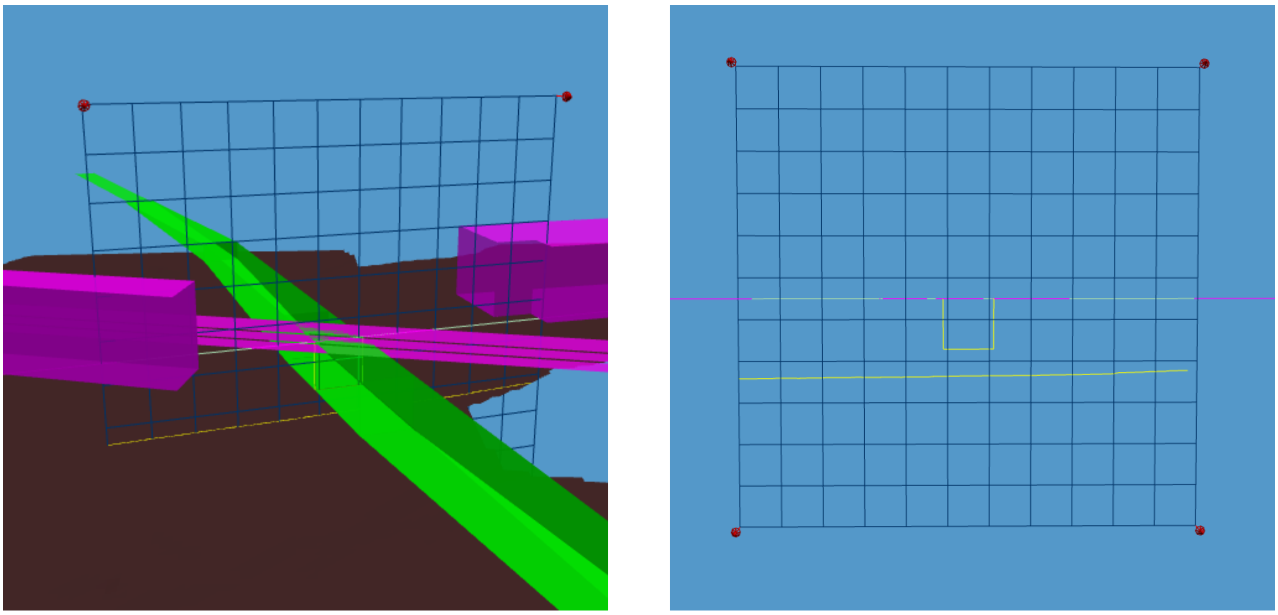


Figure 2-14 3D representation of the extracted data, on the left the intersected objects, on the right the lines extracted from this intersection.

Design generation

From all extracted data, user input and user setting the brute force algorithm creates all possible combinations of each structural element that hold up to the requirements. A flowchart that represents this process is shown in Figure 2-15.

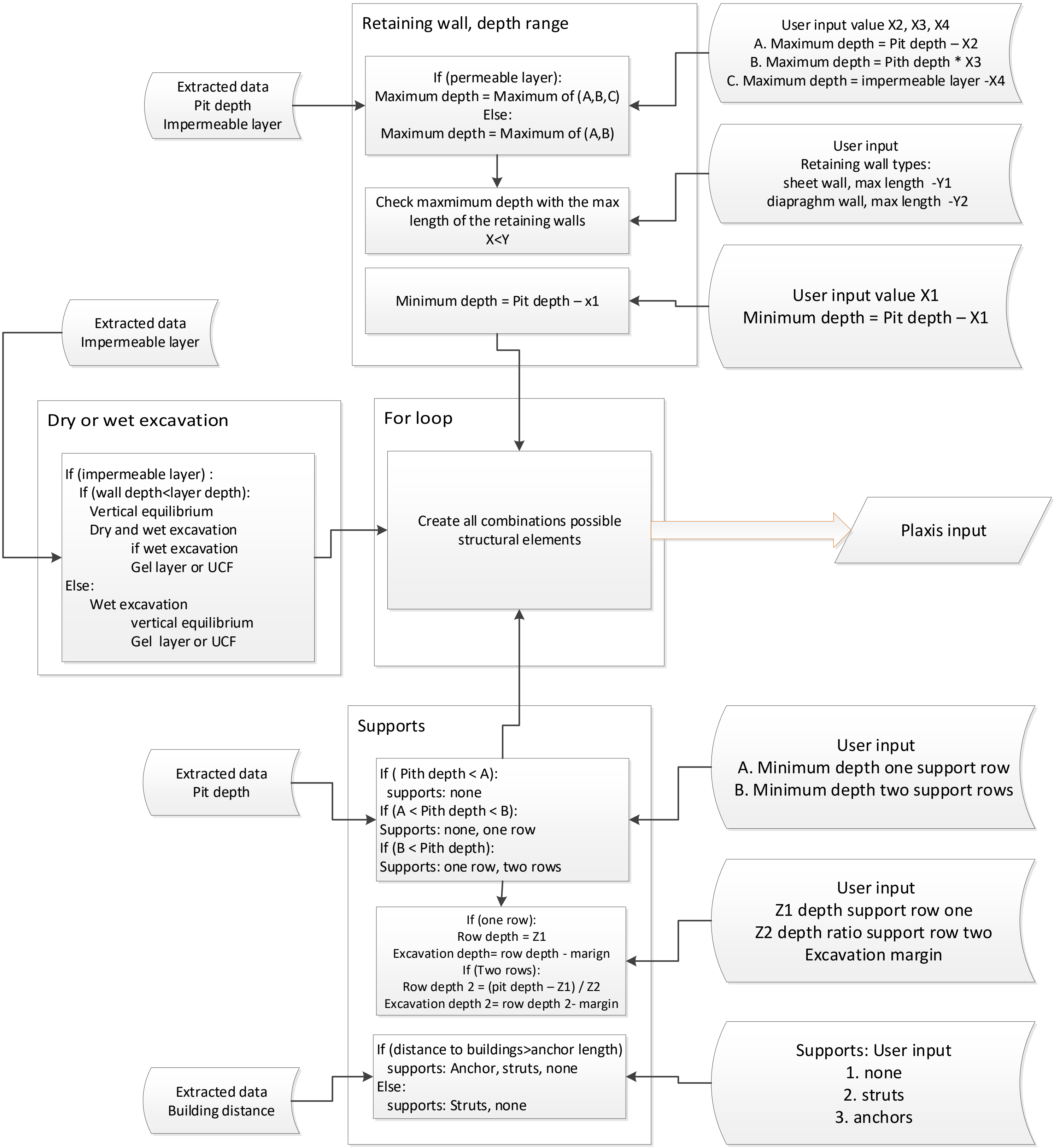


Figure 2-15 Flowchart of the pre-process design algorithm

2.3.4 Post processing

2.3.4.1 Build-in checks

The post processing process starts with the evaluation of the Plaxis results. When calculations turned out to be unstable in Plaxis, no results are generated. These failed designs have to be removed from the feasible designs. The following checks happen in the post-processing algorithm.

1. Check successful Plaxis calculations
2. Unity check, retaining wall bending moments
3. Settlement check, horizontal strain, relative rotation

From these checks, many designs turn out to be unfeasible. After these checks, these designs are not evaluated anymore. For all feasible designs for each cross-section, SET creates the design of the excavation pit around the to be design object. Examining all designs of each cross-section would require immense computation power. For example, assuming each cross-section has 750 calculations (Chapter 2.3.5.1), the amount of combinations would be 750^n for a full tunnel design. When 7 cross-sections are used, this would generate 131 * 10^{21} possible combinations of designs. To counteract this, only the six best calculations of each calculation cross-section were used in this process. For a design with seven cross-sections, this already would amount to a quarter of a million possible combinations. This already takes seconds to calculate this. To provide a wide range of variety in the evaluated designs, the best six designs are chosen on the cost and on the uniqueness of the structural elements. For all the combinations of these cross-sections, the total MKI, costs, and mobilisation costs are determined.

Successful Plaxis calculations check

If a calculation is successful, all construction phases are available in the Plaxis result files (Chapter 2.2.1). If construction phases are missing, it means that the calculation failed. In the design algorithm, a loop is set up to go through all the data generated by Plaxis and compare the construction phases with the construction phases in the Plaxis results. If this does not correspond to each other, the calculations can be deemed a failure. The exact reason why calculations fail in Plaxis is hard to recover. This makes the Plaxis calculations a bit of a black box.

Data conversion

Additionally, the data conversion loop extracts the data from the Plaxis files and converts these to usable data in Python.

Unity check retaining walls

The Plaxis results contain the internal moments and deformations (Figure 2-16) that occur in the retaining walls. This data is provided over the whole length of the walls. On the highest values of these moments, a unity check is performed. This is done for a SLS and ULS. Especially for the diaphragm walls, the SLS is checked for the crack width. These SLS and ULS values are given by the users. For the unity check, a value of 1 is used. When the check is above 1, the calculation is deemed as a failure.

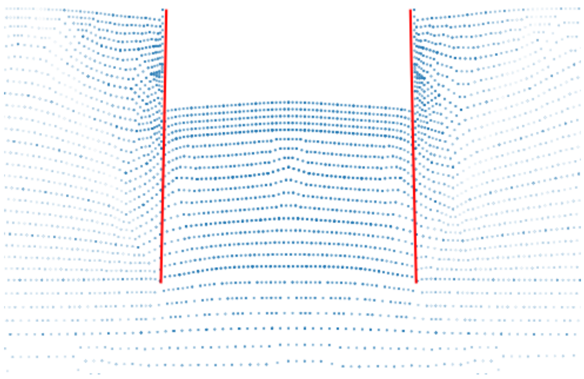


Figure 2-16 Exaggerated deformations of the soil and retaining walls (scale factor 10) (Python Plot)

Settlement check

From the Plaxis calculations the settlement of the soil is known. For each point on the mesh the deformation: U_x, U_y and U_{tot} are extracted. These settlements are used for the deformation checks on of the buildings. The extraction is discussed in Chapter 2.3.2.

Horizontal strain and relative rotation

The horizontal strain and relative rotation of the surrounding buildings is checked. The foundations of these buildings are assumed to be shallow.

The horizontal strain is assed with the normalized data of the intersected buildings (Figure 2-9). Because the calculation cross-sections are aligned, as discussed in Chapter 2.3.1, the centre point of the calculation cross-section corresponds to the centre point of the Plaxis results. Thus, the coordinates of the normalized data of the intersected buildings corresponds to the settlement values of Plaxis. This is utilized to check the horizontal strain.

For the coordinates of the end points of the intersected buildings the nearest deformation of the surface layer is taken (Figure 2-17). The horizontal deformations are subtracted from each other and divided by the distance between these two points. This process is similar for both the horizontal strain and the relative rotation, however for the horizontal strain the values U_x are used and for the relative rotation the U_y is used.

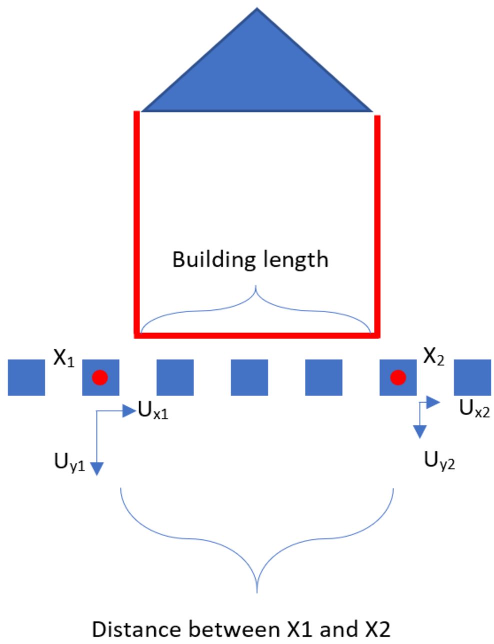


Figure 2-17 Representation of used data

$$\text{horizontal strain} = \frac{U_{x1} - U_{x2}}{(x1 - x2)}$$

$$\text{relative rotation} = \frac{U_{y1} - U_{y2}}{(x1 - x2)}$$

2.3.4.2 Settlement contours

For the 3D representation of the settlements of the Plaxis results, the U_{tot} is interpolated by Python. The FEM data is first turned into grid data. Then a standard Python function can interpolate the data and provides the plot shown on the right. For each line in this contour plot, the data is turned into a "WaveFront .obj" file Appendix B.9.1. This is done in combination with the real coordinate of the corresponding calculation cross-section and the length of the excavation pit.

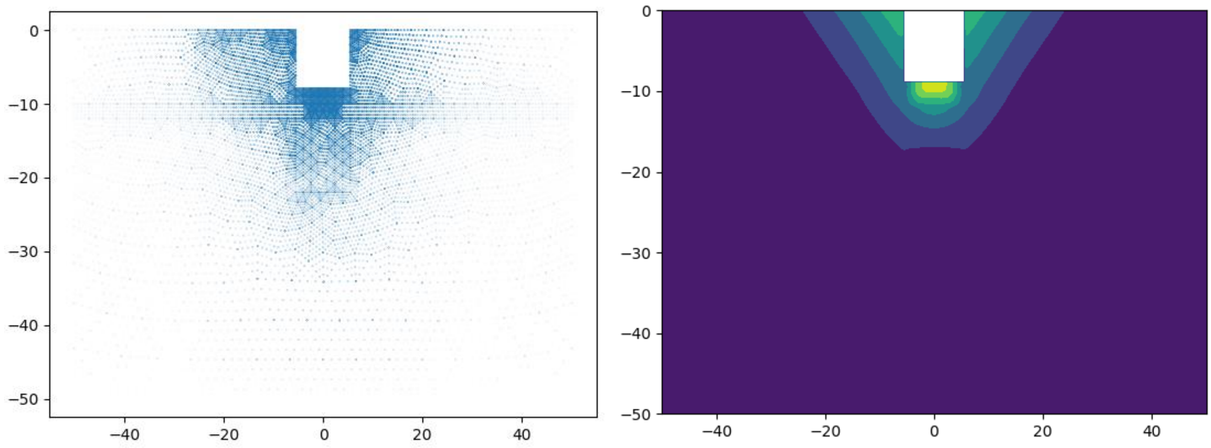


Figure 2-18 Example of settlements, left: FEM mesh, Right: interpolated data (Python Plot)

2.3.4.3 Create interactive plot

SET visualizes the difference between the designs, in an interactive plot of the MKI and costs. Because, all combinations of possible designs add up to too many designs, only six designs of each calculation cross-section are used. For seven calculation cross-sections this adds up to around a quarter of a million feasible designs of the full excavation pit. These six designs are selected on the basis of their costs and on their uniqueness designs. From al designs of each calculation cross-section, first the cheapest is extracted. Secondly the cheapest with a different support or excavation type is extracted. Lastly the cheapest design with unique walls are extracted. This process stops when six designs are extracted.

For the full design of the excavation pit, the six designs from each cross-section, are combined. SET adds for each of these combinations, the costs and MKI values. Mobilisation cost for similar structural elements are added once. When this is finished, the data is send to the front-end where the interactive plot visualises this data.

2.3.5 User Settings

The design algorithm analyses the designs within the design space given by the user. The user can change the settings and has several input possibilities. Some of these inputs, like costs and material properties can be changed by editing a value. Other input can be given by adding 3D models to the design tool. For example, the data from the buildings surrounding the excavation pit is extracted from 3D "Wavefront .obj" models (Appendix B.6).

2.3.5.1 Geometric input

The number of calculations that must be checked by the design algorithm, depend on the number of variables and the range of each variable. When an additional variable is added, the number of possible combinations increases exponentially. For SET to perform its calculation, an advised amount of construction elements should be used, these are:

- 5 retaining walls
- 3 excavation types (dry, wet + UCF, wet + gel layer)
- 2 support elements (5 when including, no support, 1 row and 2 rows)

With an input of 5 wall types, a depth range with 5 steps, wet + UCF, wet + gel and dry excavation and 5 different supports, around 750 different designs for each calculation cross-section are created by the design algorithm. From these 750 calculations many calculations will be eliminated by the design tool on the basis of the Plaxis calculations, environmental requirements and structural requirements.

Retaining walls

For each retaining wall, the depth range can vary. This means that the design tool will include wall depths between a maximum and minimum depth, Table 2-2. The user can define this minimum and maximum and can also define a step size. To reduce the number of calculations it is advised to keep the number of steps to around four or five and thus the minimum and maximum accordingly.

Value	Formula	User input
Minimum depth range	Round (Pit depth – X)	X
Maximum depth range A	Round (Pit depth – X)	X
Maximum depth range B	Round (Pit depth * X)	X
Maximum depth range C	Round (Impermeable layer dept – X)	X
Maximum depth	Max(A,B,C) < Maximum wall length	
Search interval	X	X

Table 2-2 User input, wall depth range

Supports

The supports can have one or two rows depending on the user input and the pit depth. The depth of these rows can be defined by the user (Table 2-3). An excavation depth can be defined relative to the installation depth of the support. This excavation depth will be used in the Plaxis calculations. Plaxis will check if the pit, excavated to this dept, is stable in every construction phases, just before a row of supports are installed.

Value	Formula	Test case input
No supports	If pit depth >= X	X
One row of supports	If pit depth < X	X
Support depth 1 row	X	X
Excavation dept 1 row	Support depth 1 - X	X
two rows of supports	If pit depth<X	X
Support depth 2 row	(Support depth 1 – pit depth) / X	X
Excavation dept 2 row	Support depth 2 - X	X
Support spacing	X	X

Table 2-3 User input, Support depths

2.3.6 User input: Structural requirements

Impermeable layer, dry and wet excavations

To keep the ground water out of an excavation pit, the design algorithm checks whether an impermeable layer is present underneath the cross-section. This is checked for the full width of the excavation pit. The users can define an addition margin (Table 2-4), shown in Figure 2-19.

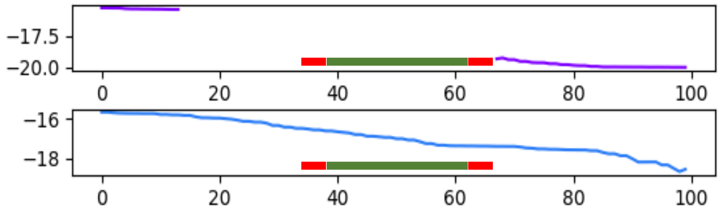


Figure 2-19 Two examples of impermeable layer check, green: pit width, red: margin. The two other colours represent the extracted clay layer data.

Value	Formula	User input
Impermeable layer tests	(Pith Width/2 + X) from middle of the pit	X

Table 2-4 Impermeable layer test

It is possible that an impermeable layer is not present, impaired or that the retaining walls do not extend into the impermeable layer. In that case, the design tool is programmed to create designs with a gel injection or an underwater concrete floor, for these wet excavations, to act as the impermeable layer. In all cases, the upward pressure of the groundwater, that is kept out of the building pit, has to be in a vertical equilibrium with the natural of artificial impermeable layer.

Vertical equilibrium: Gel layer

To achieve vertical equilibrium Figure 2-20, the weight of a gel layer, combined with the weight of the ground on top of this layer must meet the upward water pressure. It is assumed that the gel layer does not add any weight to the soil. The user can define the thickness of this layer (Table 2-5). The depth of the gel layer is determined by the design algorithm, based on the vertical equilibrium. The depth of the retaining walls is checked to make sure they extend into the gel layer. In this calculation no floor is added, and the gel layer does not provide structural stability in the Plaxis calculations.

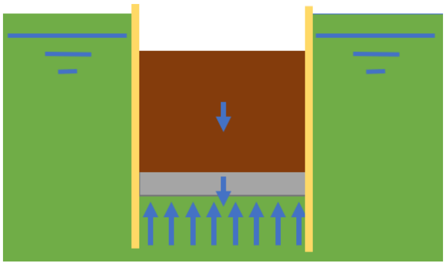


Figure 2-20 Vertical equilibrium, gel layer

Value	Formula	User input
Gel layer thickness	X	X
Minimum wall length underneath layer	X	X

Table 2-5 User input Gel layer

Vertical equilibrium: Underwater Concrete Floor

The vertical equilibrium for the underwater concrete floors is found between the weight of the concrete floor, tension in vertical anchors and the upward water pressure (Figure 2-21). As a rule of thumb, a thickness of the concrete floor can be set for an amount of upward pressure. For example, 1m thickness for every 50kn/m^2 of upward water pressure. The residual upward pressure is transferred to vertical anchors. The number of anchors is determined on by the residual force of the upward pressure multiplied with the floor area. This is divided by a user determined values. It is assumed that the floor of the to be build object is placed on top of the UCF. A drainage layer between these floors is neglected. For this reason, the top of the underwater concrete floor must be at the required bottom of the excavation pit, this requires a deeper excavation. The length of the retaining walls is checked to extend below this additional depth. The user can provide the input in shown in Table 2-6.

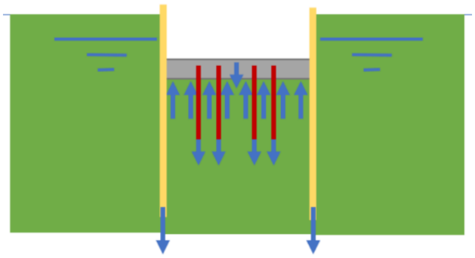


Figure 2-21 Vertical equilibrium, underwater concrete floor and vertical anchors

Value	Formula	User input
Concrete density	X	X
Minimum wall length underneath UCF	X	X
Thickness of UCF	Upward water pressure (kn/m2) / X	X
Number of vertical anchors	Residual upward force / X	X

Table 2-6 user input UFC

Vertical equilibrium: Clay layer

In the case of dry calculations, SET analysis the vertical equilibrium of the weight of the clay layer and the soil above it to that of the water pressure. This is shown in Figure 2-22. No additional checks are performed because a dry excavation is only tested when the retaining walls already extend into a solid clay layer and the impermeable layer is unimpaired. Thus, these checks are already performed. The soil density is taken from the soil properties.

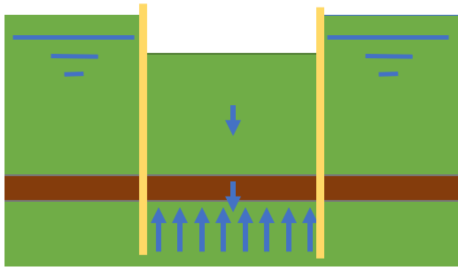


Figure 2-22 Vertical equilibrium, clay layer

2.3.6.1 Horizontal and Moment Equilibrium

The horizontal and moment equilibrium are analysed by Plaxis. Next to the geometrical user input that can be given to achieve these equilibria as described in the section above, the user can define the construction phases and material properties that Plaxis uses for the finite element calculations.

Construction phases

Corresponding to the installation of supports, floors and walls, the user should define what set of construction phases Plaxis should use. In Appendix B.2, ten of these sets are shown and how these are used is discussed in Chapter 2.2.1. An example is shown below:

Phases of a dry excavation pit with 1 row of struts:

1. Initial phase
2. Installation retaining walls
3. Drain to the excavation depth below row 1
4. Excavate to the excavation depth below row 1
5. Install Struts row 1
6. Drain bottom, (water level above the bottom of the pit is deactivated)
7. Excavate bottom, (the soil above the bottom of the pit is deactivated)

Material properties

For each structural element, the user can define the material properties. Appendix B.1 lists the supported properties. Grout anchors require additional input, the user must define the length and the angle of the anchor rod and body. The supported material properties are shown in Appendix B.1. These values can be defined by a user in a CSV file.

2.3.6.2 Structural Checks

Unity check retaining walls

For the internal moments of the retaining walls, the user can define a maximum. For the unity check, a value of 1 is set as default but can be change by the user (Table 2-7).

Value	User input
Maximum internal moment SLS	X
Maximum internal moment ULS	X
Unity check threshold	X

Table 2-7 Unity Check input

2.3.6.3 Settlement checks

Horizontal strain and relative rotation

SET checks the horizontal strain and relative rotation of the surrounding buildings. Therefore, a maximum can be set for each building (Table 2-8).

Value	User input
Maximum horizontal strain	X
Maximum Relative rotation	X

Table 2-8 Settlement Requirements

2.3.6.4 Unity check, Moment capacity

Internal bending moment

SET checks the internal bending moments of the retaining walls. Therefore, a maximum ULS and SLS value can be set for each retaining wall (Table 2-8). The value of the unity check is set 1 as default but can be changed by the user.

Value	User input
Maximum SLS	X
Maximum ULS	X
Unity threshold	X

Table 2-9 Internal Bending Moment

2.4 Collaboration, Interaction and 3D Visuals

In this chapter development step 3 and 4 will be covered.

Develop a suitable digital environment for an interactive collaborative design process between a group of engineers and the stakeholders. This environment must present in a clear, interactive manner, the feasible and non-feasible designs and their pros and cons by visualizing the differences between the designs within the design space

Provide visualizations of the results from computations performed by integrated external specialised software to contribute to a better understanding of the design limitations and possibilities.

Visuals can be generated in many programming languages with many levels of complexity. A balance was found in Web Graphics Library (WebGL), (Khronos Group) for the front-end of SET. This choice was made on the basis of the pros and cons of the following programming languages:

CAD/BIM applications
- Rhinoceros and Revit (Appendix A.1)

Programming language with a graphics library.
- DirectX game engines Unity and Unreal (Appendix A.2.1)
- OpenGL, in Python (Appendix A.2.2)
- WebGL, in JavaScript

The pros and cons of different programming languages and the fully developed applications are elaborated upon in Appendix A .

WebGL is compatible with JavaScript (JS) and a JS module called “Three.js” (Cabello) provides functions to utilize the WebGL capabilities. WebGL is developed by Mozilla Foundation (Mozilla Foundation) and maintained by the Khronos WebGL Working Group and is an open source graphics library that is, in contrast to DirectX and OpenGL, focused on the graphics in web browsers. HTML and CSS are used for the layout of the web browser.

2.4.1 3D Visuals and Interaction in WebGL

2.4.1.1 Cross-Sections

In WebGL cross-section planes are created (Figure 2-23). These planes are used as the calculation cross-sections, which is discussed in Chapter 2.3.1. Each cross-section can be dragged and rotated and positioned above the to be designed object. When the “R” key is pressed, the object can be rotated, to move these cross-sections the “T” key should be pressed. The ability to move or rotate can be toggled with a button. The centre point of the cross-sections is tracked in the front-end and are based on polar coordinates. Each time a plane is moved or rotated, the corresponding values are sent to the back-end.

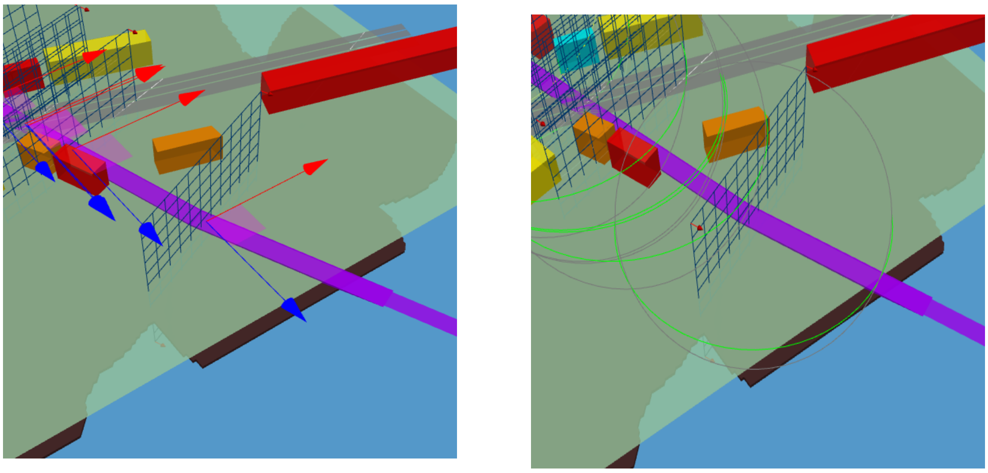


Figure 2-23 User defined calculation cross-sections, left: draggable, right: rotatable

2.4.1.2 Clipping Planes

Clipping planes are planes that define a border in which objects are shown in a 3D scene. In the example shown in Figure 2-24, 4 clipping planes are used to cut out the ground within the excavation site. This creates the illusion of removed ground. All the ground layers are now viable for a bird view perspective. Each clipping plane intersect at the corner with another clipping plane. These clipping planes are used to make longitudinal cross-sections.

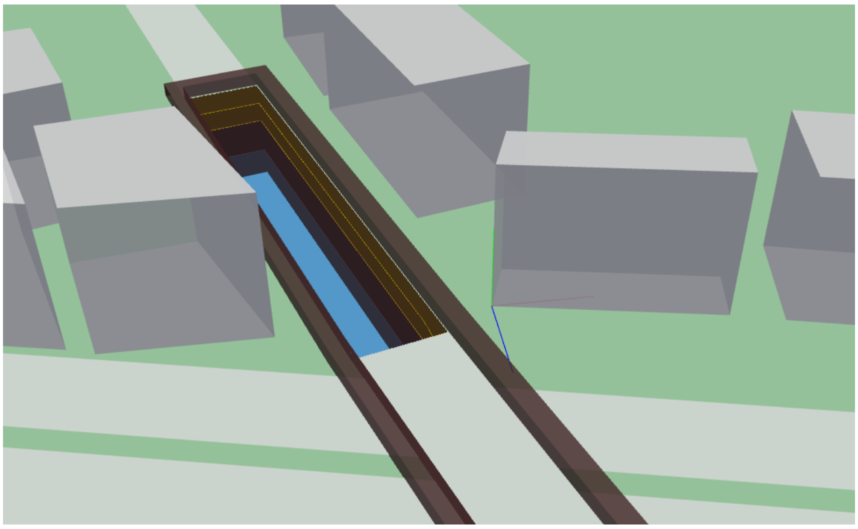


Figure 2-24 Clipping plane example

2.4.1.3 3D Object and Rendering

To create a 3D scene (Figure 2-25), objects need to be rendered. Rendering an object requires two things, the geometry and the material. The geometry is build up from triangles and the appearance is determined by the material that tells the renderer how the object must look.

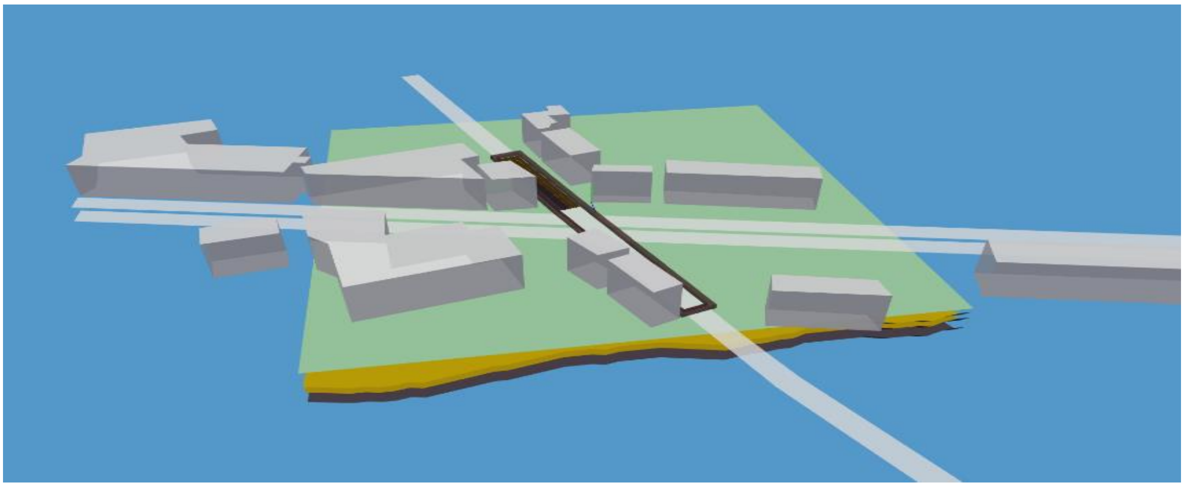


Figure 2-25 Example of rendered scene

The material parameters that are used are:

- Transparency: Boolean, is the object to be allowed to be transparent
- Opacity value: Boolean, in combination with a transparency.
- Colour: RGB hexadecimal
- Lighting: how does the object interact with lights.

The interaction with clipping planes is defined in the material of an object.

- Clipping-plane intersection: Boolean, will clipping planes affect the object
- Clipping-planes array: array of clipping planes the object will interacted with

For the 3D models, instead of the BIM IFC standard (IAI Model Support Group, 1996-2007), the format "Wavefront .obj" is chosen for this thesis. "Wavefront .obj' is widely used in 3D applications and in "Three.js". The format is written in ASCII, a plain text file format, thus it is not encoded or written in binary and can be opened and edited on any computer. This makes the file format very easy to use and other file formats can be easily translated to "Wavefront .obj". The data contained in an .obj file is only used for the appearance of an 3D object. Apart from the outlines and shape of the object, information about the textures and materials can be included in the file format. In Appendix B.9.1, Figure 0-17 an example of the file format is shown together with its visual representation.

2.4.1.4 Ray casting

Ray casting is the process of sending an imaginary laser from the user viewpoint to an object that is clicked in 2D representation of an 3D environment (Figure 2-26) (Linietsky & Manzur, 2014). This imaginary laser can track which objects it intersects with. The information is used to select objects.

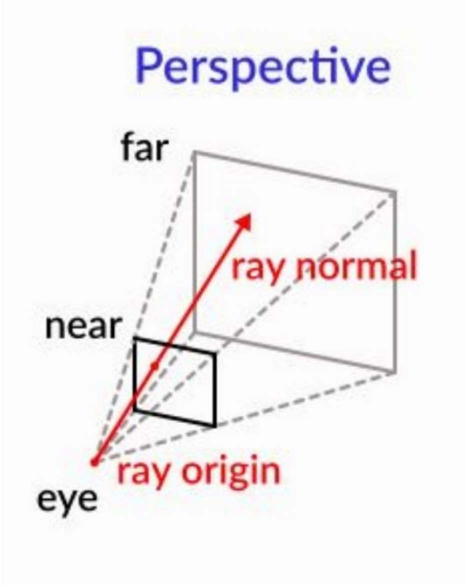


Figure 2-26 Ray casting (Linietsky & Manzur, 2014)

2.4.1.5 Lighting

The lighting gives a sense of depth to objects. In Figure 2-27 a comparison is shown between two objects. On the left is an object with an artificial light source, on the right an object with no lighting. The picture on the left is more realistic and conveys more information. In SET a hemispherical light is added from above and from below. Such a hemispherical light can be compared with light of a cloudy day. Because the design must also be viewable from below, an upward facing hemispherical light source is added.

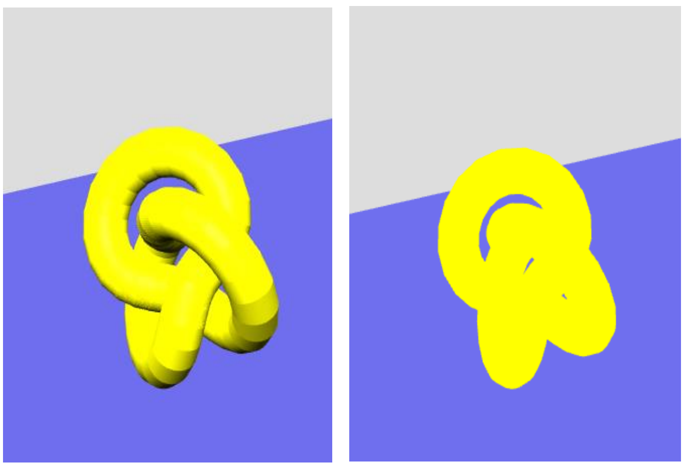


Figure 2-27 Lighting comparison

2.4.1.6 Multiple scenes

The rendering process is done on an imaginary digital canvas. To give the user the ability to view the design from two sides at the same time or to convey different data in two views, multiple canvasses are sometimes used.

2.4.1.7 Level of detail

Limitations like internet bandwidth and performance of internet browsers influence the performance of a web application (Liu, Xie, Tang, & Jia, 2016). In this thesis the amount of data that needs to be sent is reduced with the level of detail (Het Nationaal BIM Platform, 2010) of models. For this thesis the Level of Detail 000 is used. simple shapes and boxes represent objects. Object can contain non-geometric information.

2.4.1.8 Tables

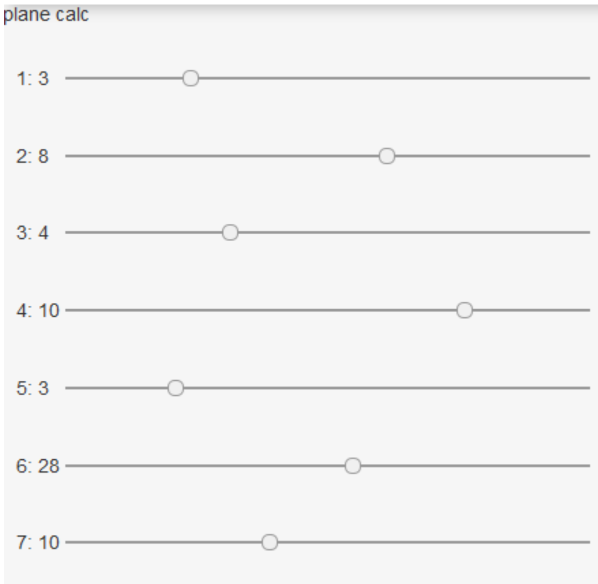
Tables are used to show data of the objects in the 3D environment and update automatically. These tables are written HTML and CSS. An example is shown in Figure 2-28.

ID	3
Name	Hilversum_3
Color	ffee00
Strain	1E-3
Load	10
Rotation	2E-3

Figure 2-28 HTML table, used for object information

2.4.1.9 Sliders

Sliders are used to navigate between different designs. When a slider is dragged, it triggers a function which astatically updates the 3D scene. An example is shown in Figure 2-29.



Wall Type	Wall Depth	Wall Area	Exca type	Impermeabe	Exca soil	Support	# Supports
Dia 1200	-22 m	1337 m2	wet UCF	1237 m3	3606 m3	strut 2r	24
AZ 46	-21 m	2179 m2	wet UCF	3132 m3	8470 m3	strut 1r	21
Dia 1200	-25 m	1564 m2	wet UCF	1935 m3	5719 m3	strut 1r	13
AZ 46	-25 m	1568 m2	wet UCF	1941 m3	5737 m3	strut 1r	13
Dia 1200	-20 m	3416 m2	wet UCF	4895 m3	12156 m3	strut 2r	68
AZ 20	-23 m	4349 m2	dry	0 m3	2695 m3	anchor 1r	38
AZ 20	-20 m	4553 m2	dry	0 m3	3911 m3	strut 1r	46

Figure 2-29 Sliders that can be used to change the visualized design of each of the seven cross-sections. The table on the right shows the corresponding properties of that design.

2.4.1.10 Interactive Plot

Quantified data of the designs are visualised in an interactive plot. The plot in Figure 2-30 is created with Plotly (Plotly, 2018), which is a JavaScript module. Each point in these plots can be clicked, the corresponding design in the 3D scene will update accordingly.

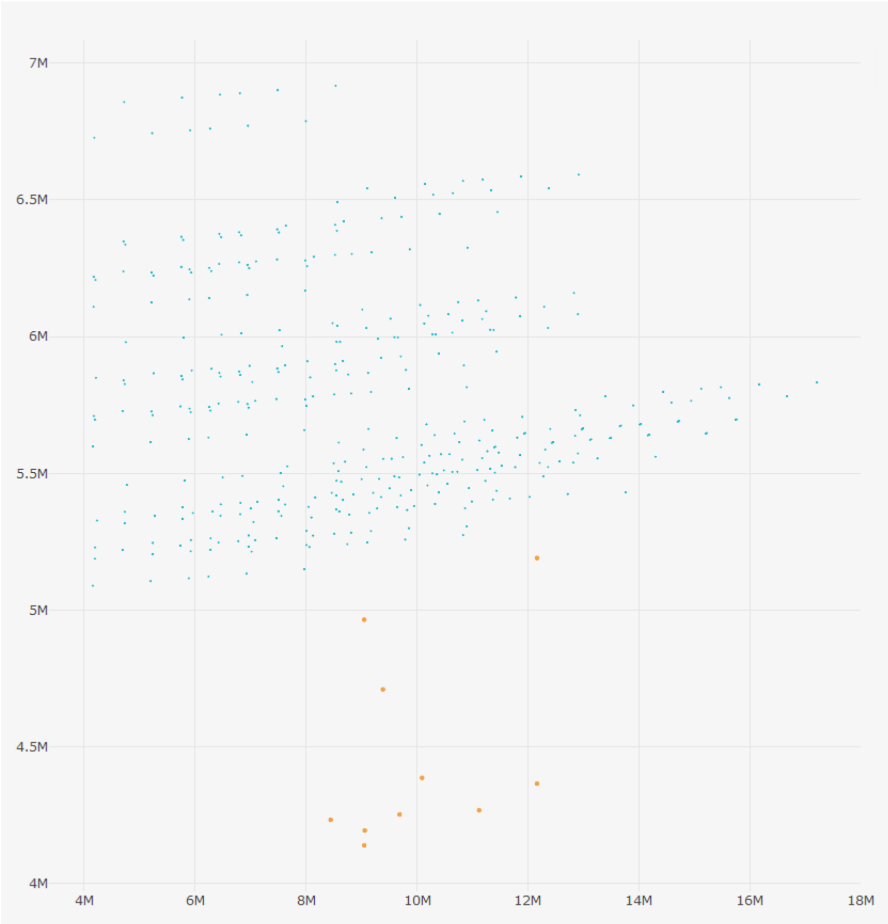


Figure 2-30 Interactive plot (Plotly, 2018)

2.4.1.11 Dropdown menu bar

Navigation between different webpage is possible via the menu bar. The number of cross-sections determine the number of input pages as discussed in Chapter 2.4.2.2. The number of navigation buttons is updated accordingly in a drop-down menu. Such a dropdown menu is shown in Figure 2-31.

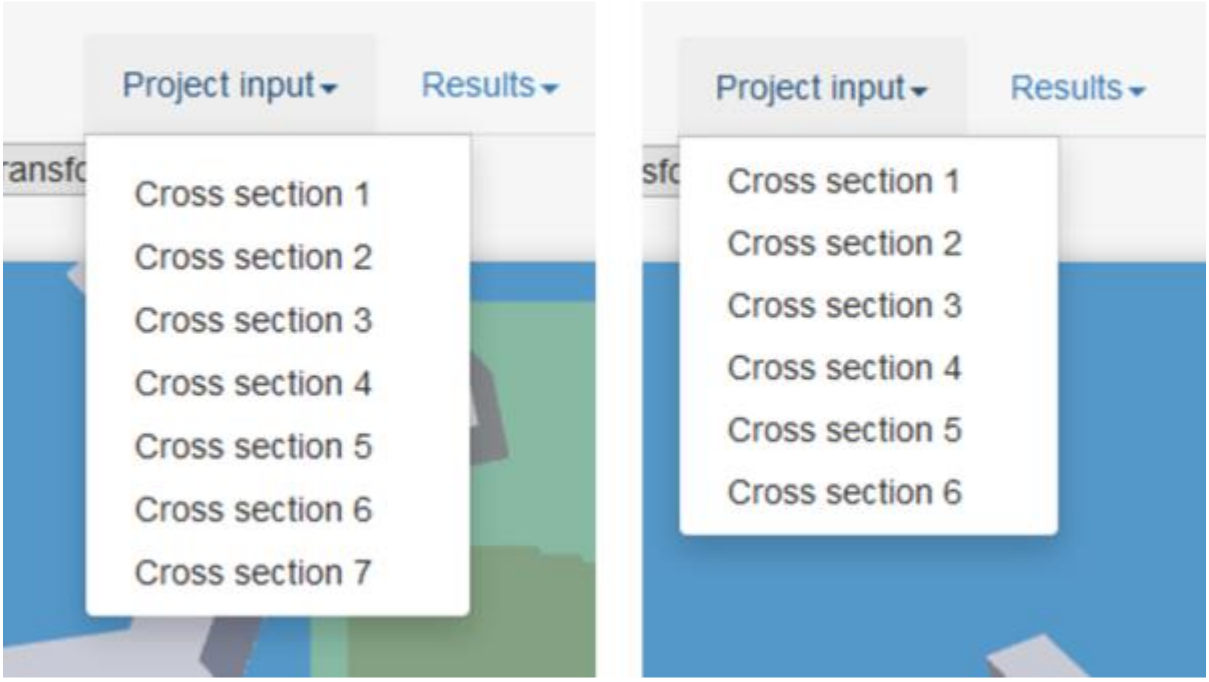


Figure 2-31 Menu bar with dropdown menu. The number of menu buttons change according to the number of calculation cross-sections

2.4.2 Website

The front-end of SET is developed for the web browser. The different pages that can be accessed by the tool are discussed in this section.

2.4.2.1 Input page

The input page, shown in Figure 2-32, is simple. It shows the 3D objects and gives the user the ability to define calculation cross-sections. Four buttons are included in this page that provide all needed functionalities for the cross-section placement.

- Add plane
- Delete plane
- Align plane
- Toggle transform

Additionally, ray casting is used on this page and allows users to click objects in the 3D environment. When an object is clicked, it turns blue and a properties table pops up on the right. The user can adjust the properties of the clicked object form within this table. These adjustable building properties are:

- Maximum relative rotation
- Maximum horizontal strain
- Load of the building
- Colour of the building
- Building ID

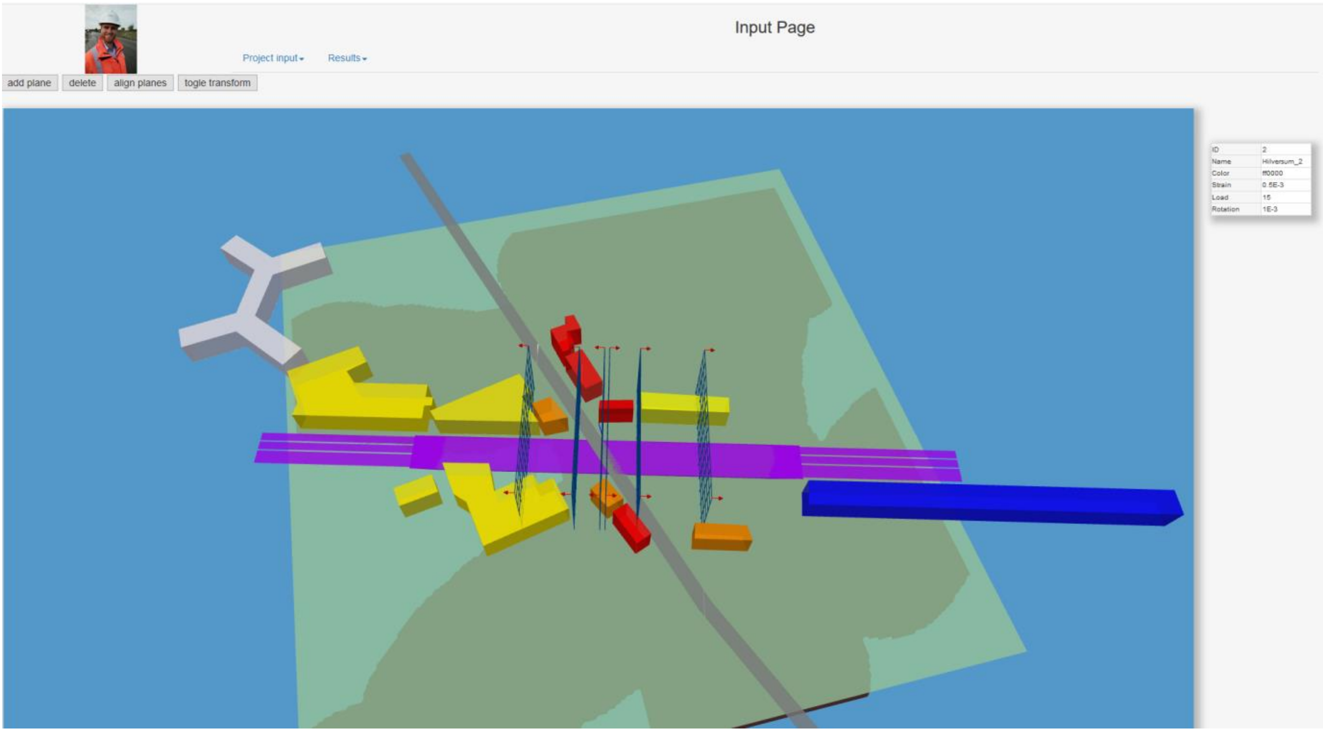


Figure 2-32 Input page

2.4.2.2 Input page for each cross-section

For each calculation cross-section, the Plaxis input calculations can be viewed in a separate input page, an example is shown in Figure 2-33.

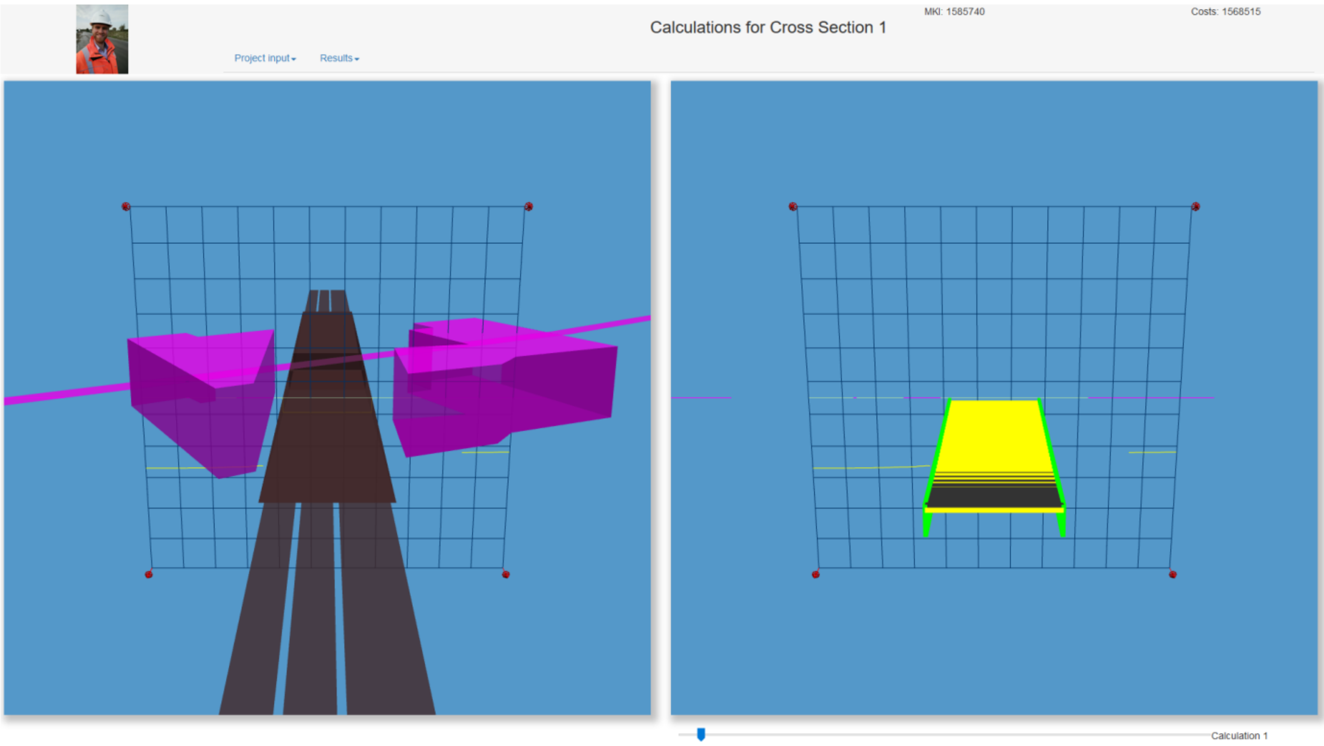


Figure 2-33 Input page for each cross-section

2.4.2.3 Result page

The results of the calculation processes are shown on a results page (Figure 2-34). Four functionalities are offered, and each functionality is linked together.

Navigation options:

- Interactive plot
- Sliders

Data conveyers:

- Properties table
- 3D view of the design results as a whole

The plot in Figure 2-34 on the right is interactive. Whenever a user hovers over a point, the point will exactly show its cost and MKI data. Additionally, when the point is clicked, the 3D view and properties table will update and show the design that corresponds to that point. The sliders will update as well. For each cross-section the designs will change accordingly. Conversely when a single slider is moved, the only a single cross-section will update in the 3D view and in the properties table. Additionally, the plot will update too, as it is very likely that a dot is not plotted as discussed in Chapter 2.3.1. In this plot, only feasible designs are shown, with the sliders, however, unfeasible designs can also be analysed.

A button is present to toggle between a “2D” longitudinal cross-section view and full 3D view in the 3D environment.

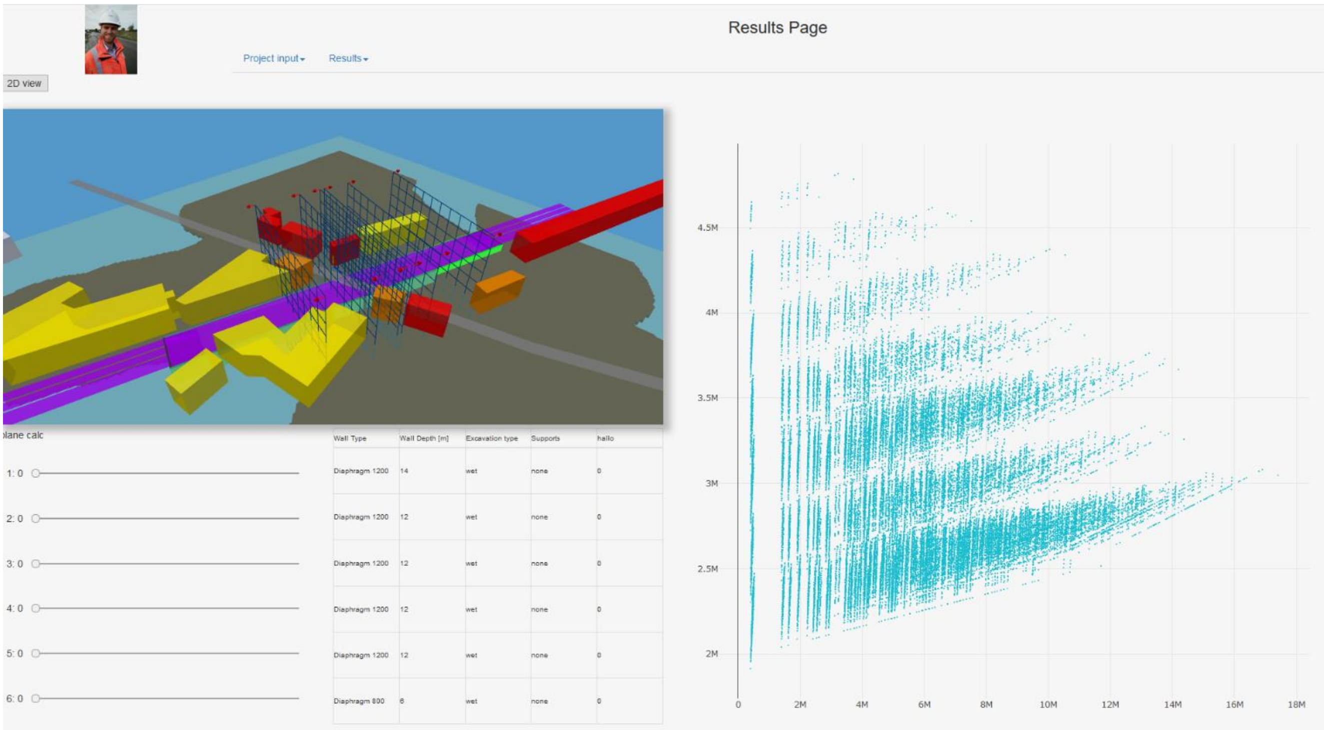


Figure 2-34 Results page

Clipping planes are used in this page to provide a 2D cross-sectional view along the full tunnel design. This is shown in Figure 2-35. The accomplish the 2D view of the bend of the tunnel as shown in Figure 2-36, the extracted data of the to be build object, as described in Chapter 2.3.3, is used to determine the correct location of the clipping planes.

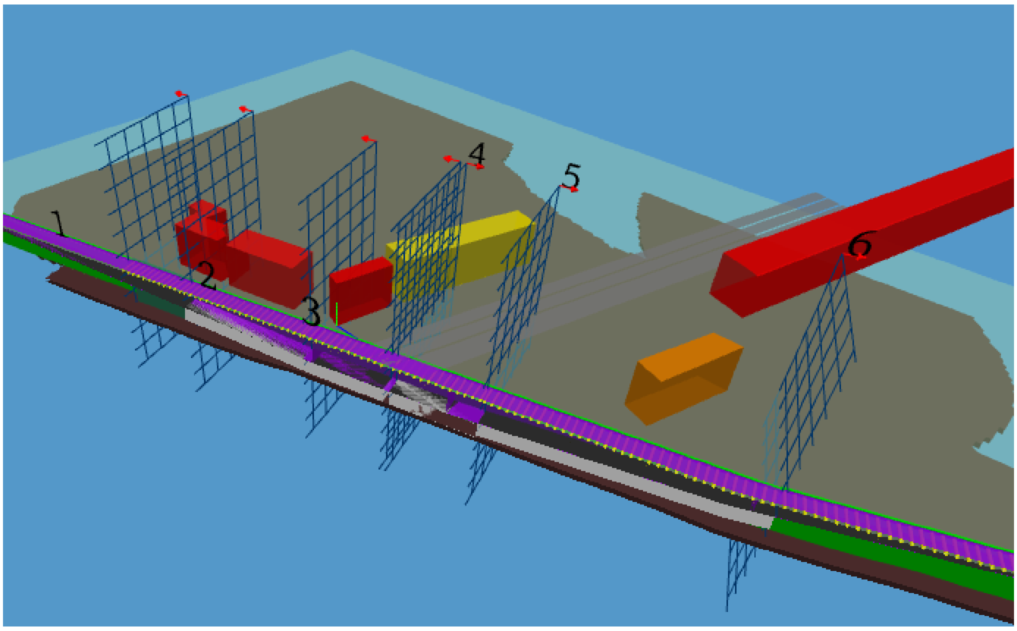


Figure 2-35 2D View of a design. Clipping plane are added along the axis of the tunnel

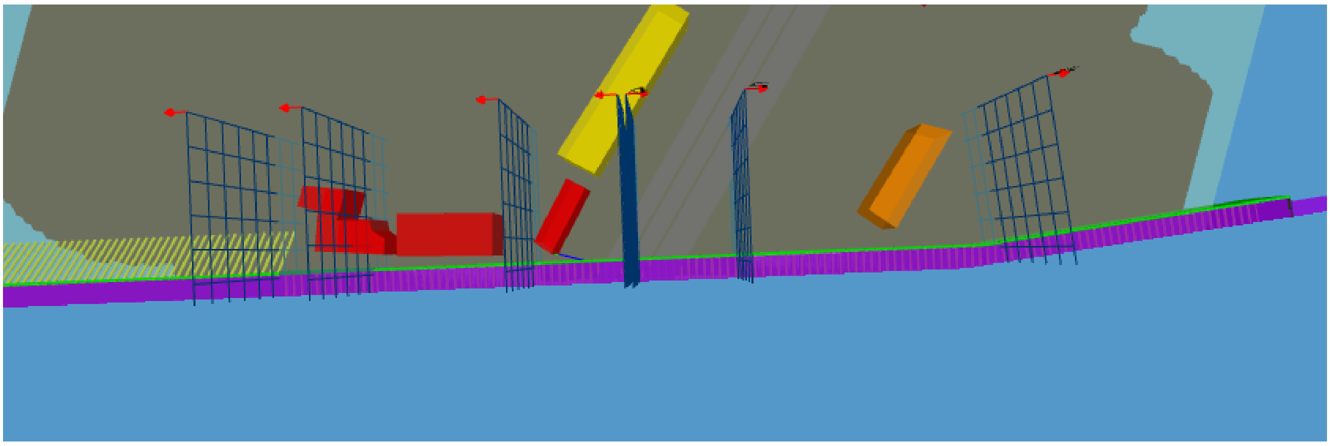


Figure 2-36 Cross-sectional 2D view from above, including the bend of the tunnel

2.4.2.4 Result page for each cross-section

For each calculation cross-section, the results can be analysed for each individual calculation (Figure 2-37). This includes settlements, horizontal strain, relative rotation and bending moments of the walls. In this page a HTML colour bar is used to convey the corresponding values to the colours that are used in the 3D visualization of the settlements.

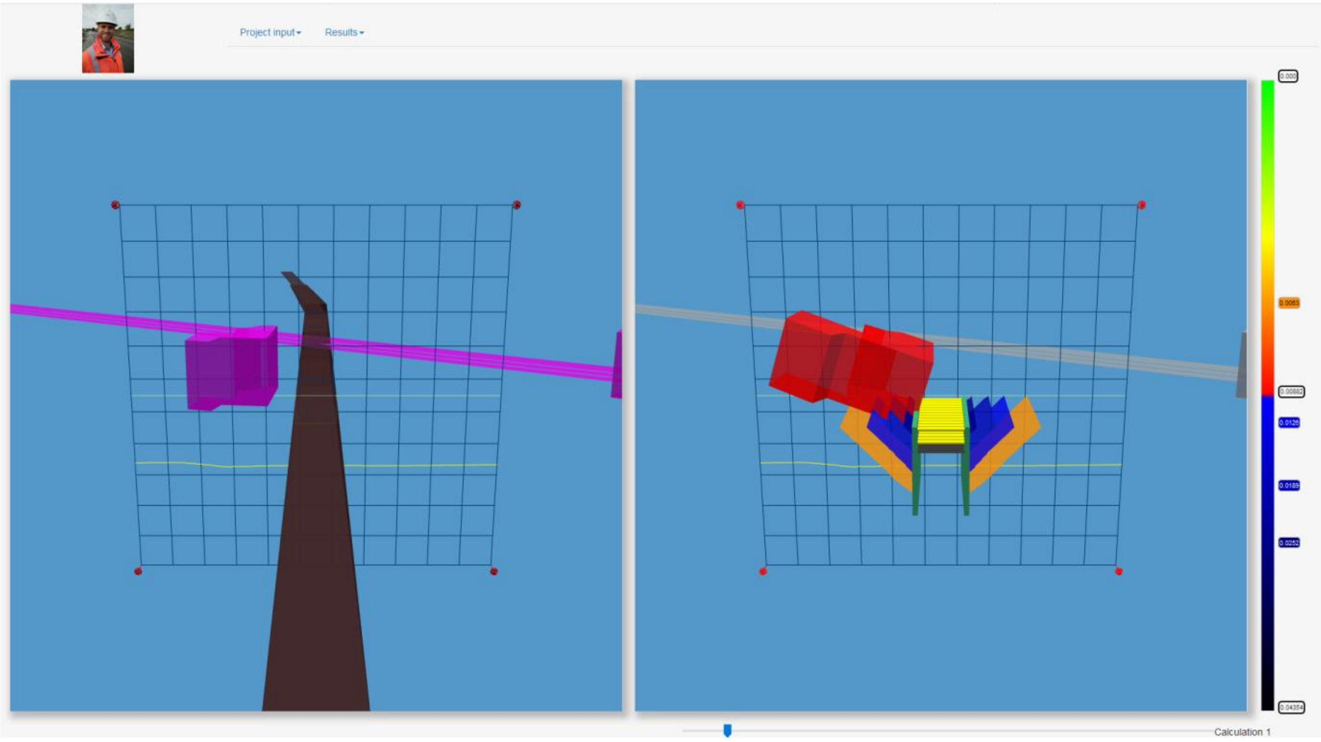


Figure 2-37 Results per cross-section

When the settlements of the soil do not meet the horizontal strain or relative rotation of the buildings, this is visualized by stretch the building or tilting it. This is shown in Figure 2-38.

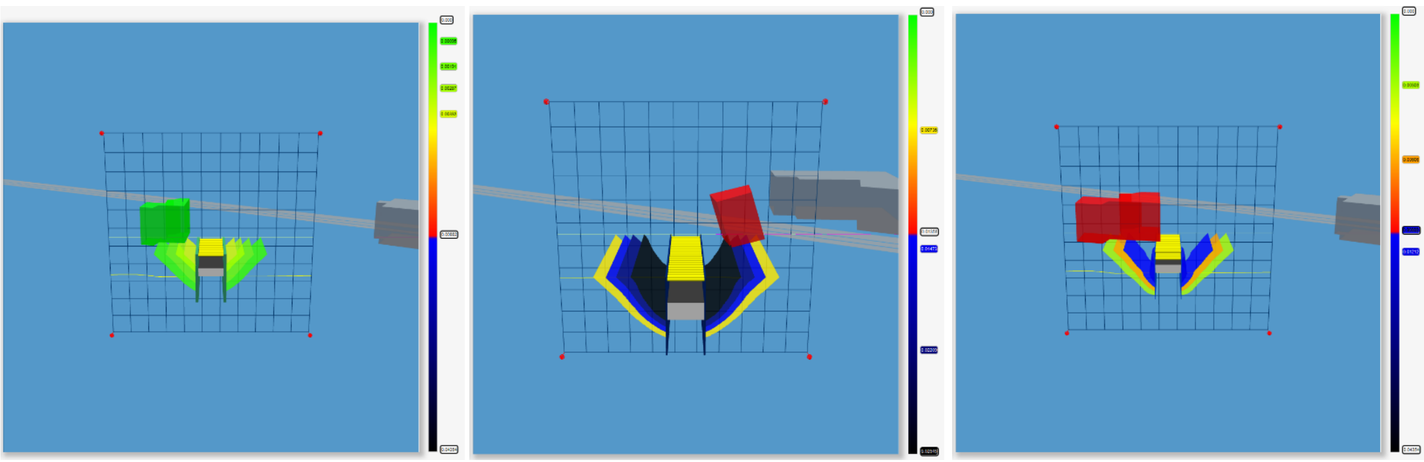


Figure 2-38 Visualization of the settlement requirements. Left: requirements met, Middle: relative rotation not met, Right: horizontal strain not met

2.5 Integration

The integration of the chapters above, requires two programming languages to communicate with each other. To establish this communication, a server is setup. Due to security issues, the server used in this thesis is limited to a local network. When these security issues are addressed the server can be opened to the internet.

2.5.1 Back-end Server

A Python module called “Django” (Django Software Foundation, 2018) is used to setup the server. On the one hand this server satisfies the communication needs between Python and JavaScript, on the other hand it has as benefit that the server can be used to allow people to work at the same time in the same model and satisfy the objective of collaborative design. The server only facilitates the communication between the two languages, however, the actual communication is done with two other modules called “Jinja2” (Ronacher, 2018) and Asynchronous JavaScript and XML (AJAX)(I’m not a fan of the abbreviation either) (Mozilla Foundation).

The communication from Python to JavaScript is facilitated by “Jinja2”. This module secures the communication from the server to the web front-end and does not require additional security measures. As one can imagine that the data, intentionally sent from the server to the web page, will not harm the server’s integrity. The communication from the website to the server, however, requires many safety measures. Otherwise, a hacker could manipulate the sever and gain access to secure information or to the network. The communication from the front end to the back end is facilitated by “AJAX”. For this thesis these security measures are circumnavigated by running the server locally and thus shielded from the internet. This means that only computers on a local network can access the design tool. This way “AJAX” can send data from JavaScript unencrypted.

A benefit to “AJAX” is that it can be used to update parts of the webpage This can be done asynchronous, this means that parts of the webpage can be loaded and changed without reloading the whole page (Mozilla Foundation). This reduces the reload time and enhances the user experience.

The data is exchanged via URL’s. A user at the front-end can add anything to a URL and this could be utilized by a user with harmful intentions.

When the front-end request an “AJAX” action and must send data to the back-end, it asks the server to open an URL. The data that needs to be transferred is included in this URL, in plain text, together with a tag. When the server receives that request, it reads the tag and extracts the data from the URL. The request to open that URL is denied by the server and only the processed information is returned to the front end. An example is shown below:

Front end:
“AJAX” Request URL: Results/graph_update/<graph_values(x,y)>

Django:
    Use function graph_update with values x,y
        Calculate (x,y) = Z
        Return Z

Front end:
        Update graph with value Z

2.5.2 Parallel computation server

A server is built to share data and this property utilized to share workloads between computers. As shown in the flowchart in Chapter 2.6 parts of the design tool run on other computers, in this flowchart three computers are used. Because of the parallel nature the process on one computer does not have to interfere with the processes on the other computers. This makes SET fast and responsive.

2.5.3 Data structure

Static files, like the "Wavefront .obj" files, are both used by Python in the design algorithm and by Django to send these to the front-end. Because of security reasons and the way Django developed, these static files must be in some predefined folders of a computer. To keep the design tool robust, the use of different file names is not sufficient. For example, all "Wavefront .obj" files are stored in a separate map. This way the design tool can load all files in that map without having to track which files contained the right information. When the tool is used for a different project the surrounding buildings has to be loaded into that map and SET automatically uses these buildings.

2.6 Flowchart

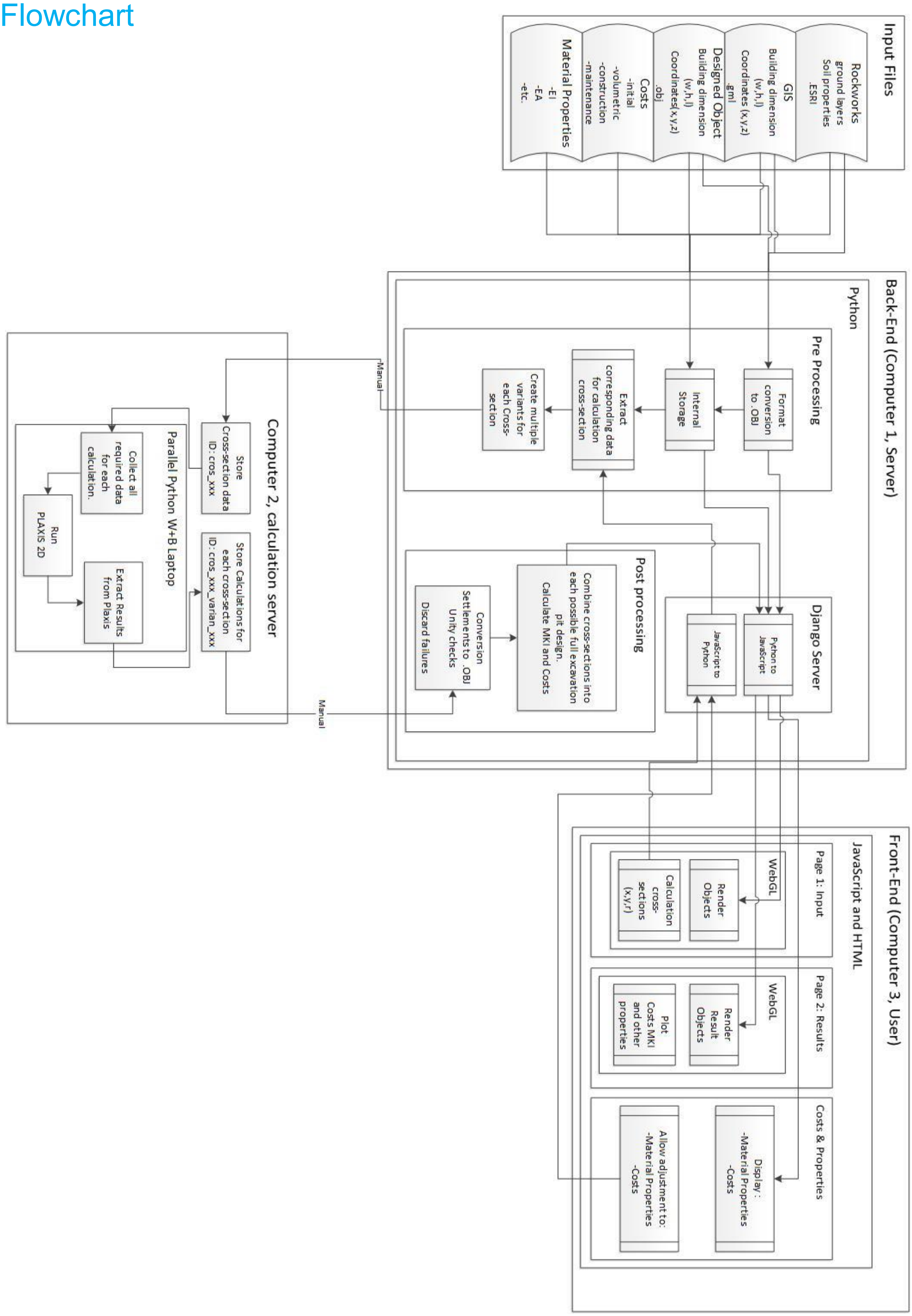


Figure 2-39 Flowchart of SET

2.7 Prototyping Conclusion

Interoperability and standardization

A geotechnical finite element programme (Plaxis 2D) is integrated into SET. Plaxis runs a finite element calculation for each design and from this calculation much data is acquired on deformation of structural elements and the soil. Plaxis runs on a parallel computer, the input generated by the design is transferred manually to this computer. According to the development criteria this should be done in a dynamic manner. This last criterion is not met by the design tool. The output of Plaxis does not influence or generate new input for Plaxis, this is due to the brute force nature of the algorithm. However, because this brute force method checks every possibility within a given input it gives the user the feeling it is dynamic. Whenever a user wants to check a 'new' design the design algorithm already considered that design. RockWorks and GIS models are loaded in to the program and are translated into a standard format. The standard CSV and "Wavefront .obj" are used in this thesis. They are open source and are widely available. They are written in the plain text ASCII format and thus can be opened on any computer without the need of special software.

Design algorithm

Ground layers can be added from RockWorks and the surrounding buildings can be added manually from GIS files. SET considers the surrounding buildings and their loads and will not design anchors underneath a building. Based on cross-sections, a design algorithm, automatically creates designs which differ in structural element, their length and properties. It checks the vertical equilibrium of each excavation pit. It eliminates dry excavations with no impermeable layer or it creates impermeable layer for wet excavations. This layer is made of gel injections or an underwater concrete floor. After these calculations are processed by Plaxis, the settlement requirements like, maximum relative rotation, maximum horizontal strain are checked and the internal moments of retaining wall undergo a unity check.

Collaboration, Interaction and 3D Visuals.

The digital environment of SET gives the user the ability to find the differences between variants. The visualization of feasible designs is suitable and interactive. However, the design limitations should be enhanced to call SET decidedly suitable. Up to a quarter of a million designs are plotted in an interactive graph, with, as a default, environmental impact and costs on the x and y axis. The visualization of this data helps the stake holders find an optimal design in a collaborative manner. The results of the integrated external software Plaxis are automatically visualised. Soil settlements are turned into 3D objects and interact with the deformation requirements of the buildings and this can be viewed in 3D.

Integration

For the integration of the tool, a server is setup. This server provides the communication between Python and JavaScript. The result of the integration is the tool called SET. In the following chapters the tool, as a whole, is tested and reviewed.

3

Test & Review

This chapter corresponds to development step 5

Validation of the design tool by testing the impact of additional requirements, objects and surroundings to the usability of the model

The aim of this thesis is to develop an integrated design tool according to the research objective and development steps given in Chapter 1.4. This chapter discusses the tests and validation process that is used to conclude if this objective is achieved.

A test session is conducted based on two test cases. In these test session, SET is used to explore many designs. The data of the results from this exploration is given in Chapter 3.1. Afterwards, these results are viewed from within SET. Eventually this design process is reviewed based on the criteria. This reviewing process is divided over the following tests:

- Validation of the outcome
- User-friendliness of the tool

The test session is held with Koen de Jong and Jeroen Hendriks, both employees of Witteveen+Bos. Koen de Jong is a geotechnical engineer and his expert opinion is used for the validation of the outcome. Jeroen Hendriks is structural engineer and manager. Together they tested the user-friendliness of the tool. Prior to the test session, the test cases were setup and calculations were already run through Plaxis. This was done because the integration with the calculations back-end is not fully automated and otherwise this would take too long. From the review, recommendations to SET emerged, these are discussed in Chapter 4.

3.1 Test Cases

To test the criteria given in the last chapter, two test cases are setup. The environment and surroundings in both cases are based on “HOV ‘t Gooi’, a project in Hilversum where a new bus lane is going to be constructed along an existing train track. The municipality of Hilversum has written a tender because it wants a tunnel at the currently level junction between with the “Oosterengweg” road. Because of the limited amount of space between buildings Witteveen+Bos designed a double layer tunnel for the “Oosterengweg” road, shown Figure 3-1 on the left. The option to have the train and bus go underneath the road was deemed to be too expensive and further investigation into this design was halted. Because the design tool is developed to investigate “out of the box” ideas, this tunnel, which is shown in Figure 3-1 on the right, was chosen as the second test case, this tunnel is called HOV in ‘t Gooi tunnel.

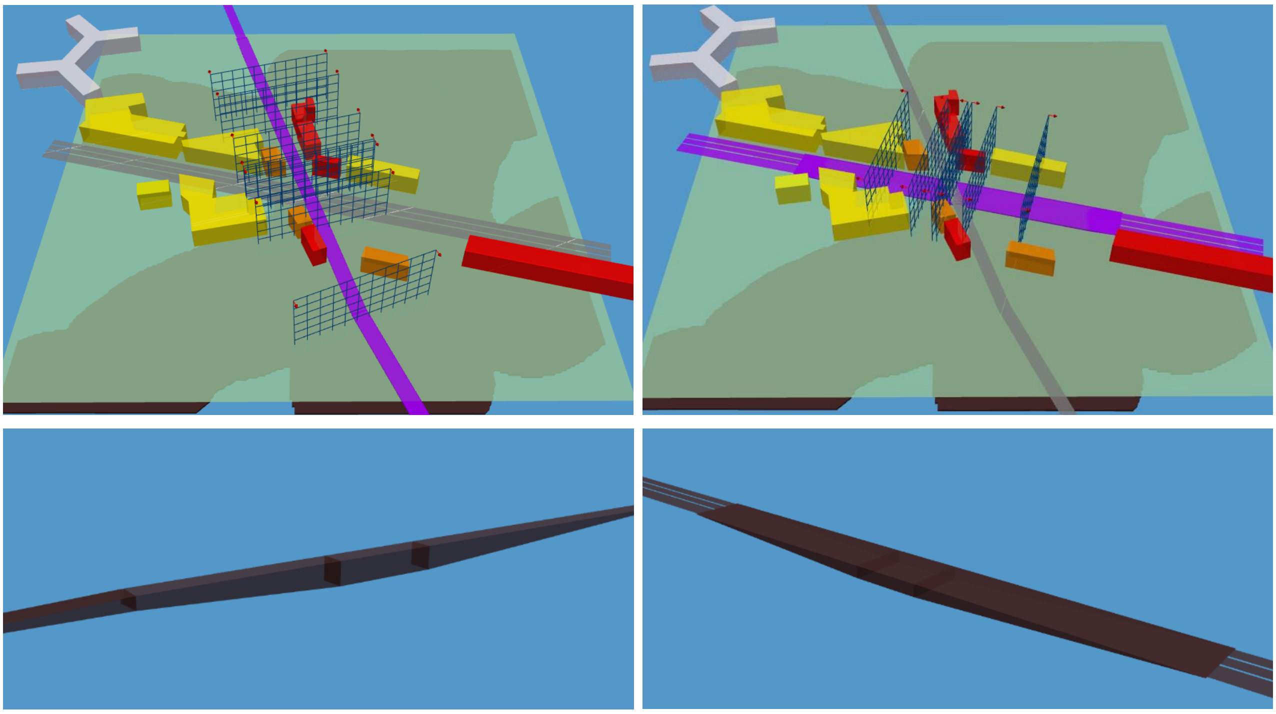


Figure 3-1 On the left, "Oosterengweg tunnel, on the right, bus and HOV in ‘t Gooi tunnel

Oosterengweg tunnel		HOV in ‘t Gooi tunnel	
Length	450 m	Length	250 m
Width	11 m	Width	24 m
Maximum depth	-11 m	Maximum depth	-6 m

Because of problems with the integration of an underwater concrete floor in Plaxis as discussed in Appendix C , many designs of the tests cases failed. To make a comparison between the two test cases, the costs of underwater concrete floors were removed and replaced with the costs of gel injections. Additionally, the support for these gel injections was added to SET. This includes a check for the vertical stability of the gel layer with ground on top. From this vertical stability the required depth of the gel layer was calculated. Additionally, the retaining walls were checked to extend at least 1 meter into this gel layer. This resulted in the elimination of some designs.

3.1.1 Test case User Settings and Requirements

For the two test cases the settings and the surroundings were set equally. The settings are shown below. Properties, costs and other values are based on numbers provided by Witteveen+Bos and from the tender HOV in 't Gooi.

3.1.1.1 Input values

Walls

For the testcases the user input is set at five types of retaining walls and is shown in Table 3-1. Three different sized AZ sheet profiles and two diaphragm walls. A thin, medium and stiff sheet profiles is chosen. For cross-sections that require heavier solutions two diaphragm walls are used. The material properties of the AZ are taken from the Plaxis userbase.

Material Name	Diaphragm 800	Diaphragm 1200	AZ18	AZ20	AZ46
Maximum length	9999	9999	20	25	30
Max moment ULS	1400	2100	640	690	1630
Max moment SLS	800	1200	-	-	-
EA	$1.20\ 10^7$	$1.80\ 10^7$	-	-	-
EA_2	$1.20\ 10^7$	$1.80\ 10^7$	-	-	-
EI	$6.40\ 10^5$	$2.16\ 10^6$	-	-	-
d	0.8	1.2	-	-	-
w	8.3	8.3	-	-	-

Table 3-1 User defined retaining wall properties

Ground layers

For the test cases, sand and clay layers are present. The material properties for these layers are shown in Table 3-2

Material Name	sand	clay
Drainage Type	drained	drained
$\gamma_{saturated}$	20	14
$\gamma_{unsaturated}$	18	14
E_{50}^{ref}	25000	3000
E_{oed}^{ref}	35000	1500
E^{urref}	140000	12000
POP	0	10
R_{inter}	0.67	0.67
C'_{ref}	0	2
kh	10	0.01
kv	10	0.01
nu	0	0
φ'	32.5	17.5
ψ	2.5	0

Table 3-2 Soil layer properties

Cost and MKI

Cost and MKI values are provided by Witteveen+Bos and are shown in Table 3-3.

Material Name	Mobilisation costs	Mobilisation group	Material costs	MKI	Unit
Sheet wall AZ 18	€ 10.000	1	€ 100	11,5	m^2
Sheet wall AZ 20	€ 10.000	1	€ 150	17	m^2
Sheet wall AZ 46	€ 10.000	1	€ 200	24	m^2
Diaphragm wall 800	€ 80.000	2	€ 225	1610	m^2
Diaphragm wall 1200	€ 80.000	2	€ 300	2415	m^2
Dry excavation of sand	€ 0	3	€ 1,50	6	m^3
wet excavation of sand	€ 2.500	3	€ 4	6	m^3
Struts	€ 0	4	€ 1.250	0	m
Grout anchor	€ 7.500	5	€ 2.500	30	a piece
Gel injections	€ 2.500		€ 125	25	m^3

Table 3-3 Material costs and MKI

Requirements from the surroundings

The properties from the surroundings are based on the values provided in the tender document of “HOV in ‘t Gooi”. For each colour of the buildings in Figure 3-2 the corresponding requirements are given in Table 3-4. Grey buildings do not have any requirements, and purple is the to be build object.

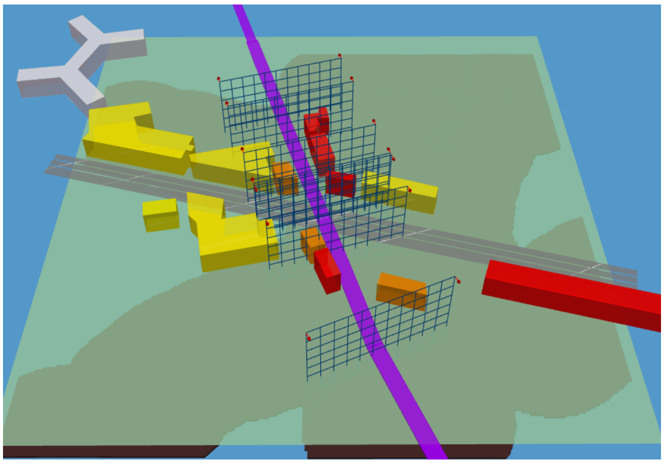


Figure 3-2 Requirements from the surroundings

Colour	Max horizontal strain	Max relative rotation
Yellow	0.001	0.002
Orange	0.00075	0.0015
Red	0.0005	0.001

Table 3-4 Building requirements

3.1.1.2 Design algorithm settings

The design algorithm settings for the two test cases are shown below. For each setting the build in formula is given, the used settings in the test cases is the user inputs.

The impermeable layer settings are shown in Table 3-5. This test checks weather an unimpaired impermeable layer is present underneath the calculation cross-sections. The user input can define the margins outside the excavation pit. For example, in a pit which is 11 meters wide, the design tool will check if there is an impermeable layer present and fully closed between – X to 11 meters + X.

Value	Formula	Test case input
Impermeable layer tests	Pith Width / 2 + X	X = 2

Table 3-5 Impermeable layer test

The range of wall depts is determined with the formulas given in Table 3-6.

Value	Formula	Test case input
Minimum depth range	Round (Pit depth – x)	X = 2
Maximum depth range A	Round (Pit depth – x)	X = 5
Maximum depth range B	Round (Pit depth * x)	X = 2
Maximum depth range C	Round (Impermeable layer dept – x)	X = 2
Maximum depth	Max(A,B,C) < Maximum wall length	
Search interval	X	X = 4

Table 3-6 Minimum and maximum wall range

The settings for the supports are shown in Table 3-7.

Value	Formula	Test case input
No supports	If pit depth >= X	X = - 6
One row of supports	If pit depth < X	X = -6
Support depth 1 row	X	X = -1
Excavation dept 1 row	Support depth 1 - X	X = -0.5
two rows of supports	If pit depth<X	X = -6
Support depth 2 row	(Support depth 1 – pit depth) / X	X = 2
Excavation dept 2 row	Support depth 2 - X	X = -0.5
Support spacing	X	X = 2.5

Table 3-7 Support formulas

3.1.2 Oosterengweg Tunnel

3.1.2.1 Pre-process Results

The results of the “pre-Plaxis” design algorithm, based on the input values and user setting, are 411 calculations. For each calculation cross-section in Figure 3-3, an average of 58 calculations are found. With an estimated search range of 750 calculations (Chapter 2.3.5.1), around 80% of the calculations are eliminated based on the requirements. The properties that are calculated by the design algorithm that are constant for each calculations of a single cross-section are shown in Table 3-8.

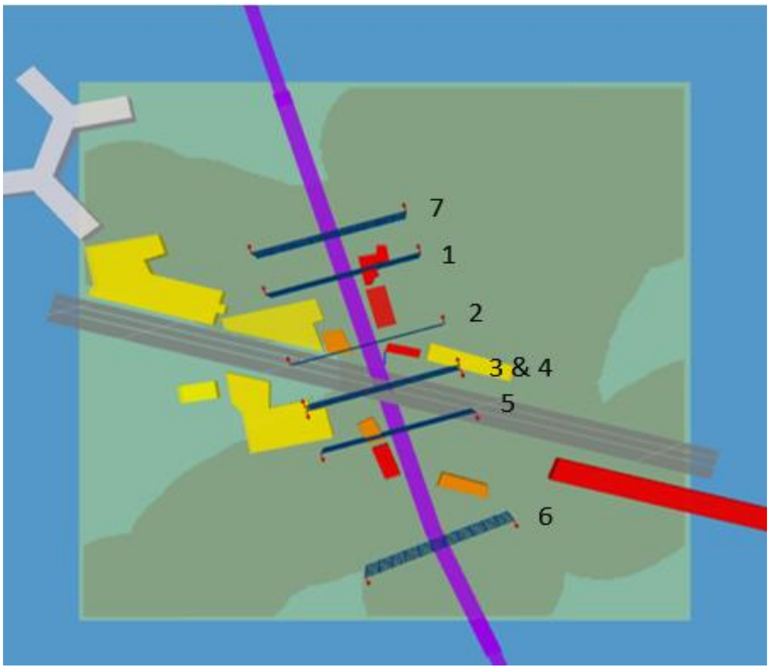


Figure 3-3 The defined calculation cross-sections for the Oosterengweg tunnel

Cross-section number	1	2	3	4	5	6	7
Number of calculations	48	46	46	46	50	87	88

Cross-section number	1	2	3	4	5	6	7
Pit depth	-7.92	-10.78	-11	-11	-10.34	-5.19	-6.25
Pit length	30.38	51.89	31.27	31.37	85.4	94.54	113.84
Floor angle	0.055	0.055	0	0	0.061	0.055	0.055
Clay layer depth	-20.35	-17.873	-17.45	-17.42	-16.87	-17.29	-20.15
Min dept range	10	13	13	13	12	7	8
Support depth 1	-1	-1	-1	-1	-1	-1	-1
Excavation depth 1	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5
Support depth 2	-4.21	-5.64	-5.75	-5.75	-5.42	-2.85	-3.38
Excavation depth 2	-4.71	-6.14	-6.25	-6.25	-5.92	-3.35	-3.88

Table 3-8 Number of calculations for each cross-section and their corresponding properties

An overview of what building elements are used for each of these calculations is given in Table 3-9. For example, cross-section 1 contained 14 calculations which included an AZ 46 profile. These calculations are manually transferred to the Plaxis server. On this computer the calculations are processed, and their results are stored, this took around 35 hours, approximately 5 minutes per calculation. The process of opening a calculation, running it through Plaxis, saving the results and starting a new calculation, was fully automated.

Cross-section	1	2	3	4	5	6	7
Diaphragm 800	10	12	12	12	12	15	16
Diaphragm 1200	10	12	12	12	12	15	16
AZ 18	4	2	2	2	2	9	8
AZ 20	10	8	8	8	8	21	16
AZ 46	14	12	12	12	16	27	32
Wet excavation	38	32	32	32	34	66	80
Dry excavation	10	14	14	14	16	21	8
No struts	-	-	-	-	-	29	-
Struts 1 row	24	23	23	23	25	29	22
Struts 2 rows	24	23	23	23	25	-	22
Anchors 1 row	-	-	-	-	-	29	22
Anchors 2 rows	-	-	-	-	-	-	22

Table 3-9 Overview of the calculations for each cross-section

3.1.2.2 Post-process Results

After Plaxis was run, around 151 of the 411 calculations were successful. These calculations consisted of the structural elements shown in Table 3-10.

Cross-section	1	2	3	4	5	6	7
Total calculations	14	14	14	14	16	52	27
Diaphragm 800	4	4	4	4	4	10	5
Diaphragm 1200	4	4	4	4	4	10	5
AZ 18	-	-	-	-	-	-	-
AZ 20	2	2	2	2	2	16	3
AZ 46	4	4	4	4	6	16	14
Wet excavation	4	-	-	-	-	34	21
Dry excavation	10	14	14	14	16	18	6
No struts	-	-	-	-	-	22	-
Struts 1 row	7	7	7	7	8	18	11
Struts 2 rows	7	7	7	7	8	-	3
Anchors 1 row	-	-	-	-	-	12	12
Anchors 2 rows	-	-	-	-	-	-	1

Table 3-10 Successful Plaxis calculations (Oosterengweg)

After the unity check and settlement requirements the 88 calculations that meet all requirements are shown in Table 3-11.

Cross-section	1	2	3	4	5	6	
Total calculations	2	2	2	2	1	52	27
Diaphragm 800	1	1	1	1	1	10	5
Diaphragm 1200	1	-	-	-	-	10	5
AZ 18	-	-	-	-	-	-	-
AZ 20	-	-	-	-	-	16	3
AZ 46	-	1	1	1	-	16	14
Wet excavation	-	-	-	-	-	34	21
Dry excavation	2	2	2	2	1	18	6
No struts	-	-	-	-	-	22	-
Struts 1 row	-	-	-	-	-	18	11
Struts 2 rows	2	2	2	2	1	-	3
Anchors 1 row	-	-	-	-	-	12	12
Anchors 2 rows	-	-	-	-	-	-	1

Table 3-11 Successful calculations after Unity check and settlement requirements (Oosterengweg)

From these 88 calculations, SET would normally take the best 6 designs from each cross-section and make every possible combination to create the design for the whole tunnel. In total a quarter of a million designs would emerge from this process. However, most of the cross-sections did only have 1 or 2 designs that met every requirement. From these calculations, the design tool only managed to create 576 combinations. From these combinations, SET has created an interactive plot which is shown in Figure 3-4. Al dots on this plot represent the total costs and MKI values of the full and feasible designs for a tunnel. The point in the lowest left corner is the most economical and most environmental friendly design. For this reason, this design is elaborated upon and is shown in Figure 3-5.

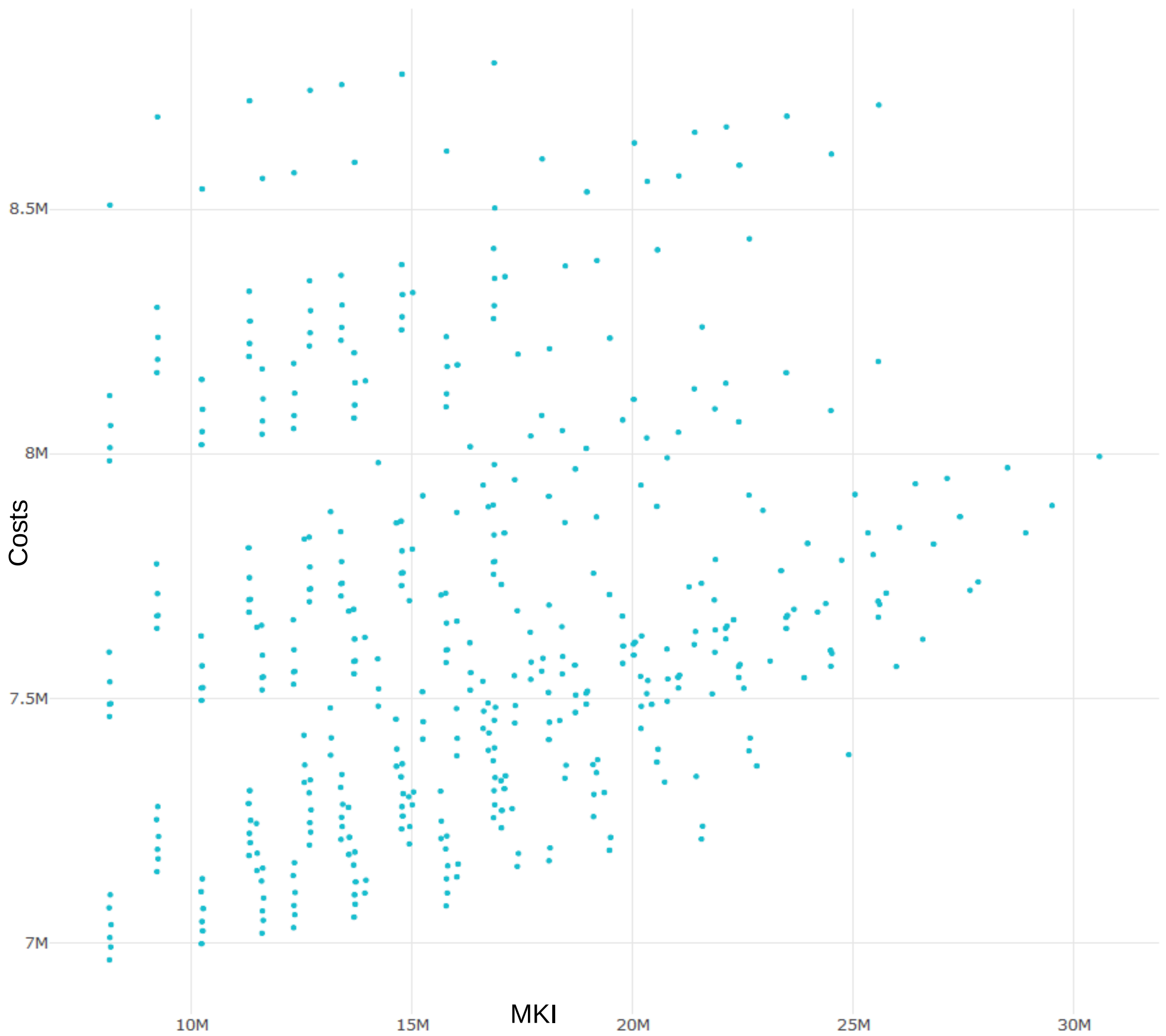


Figure 3-4 Oosterengweg tunnel 576 best designs

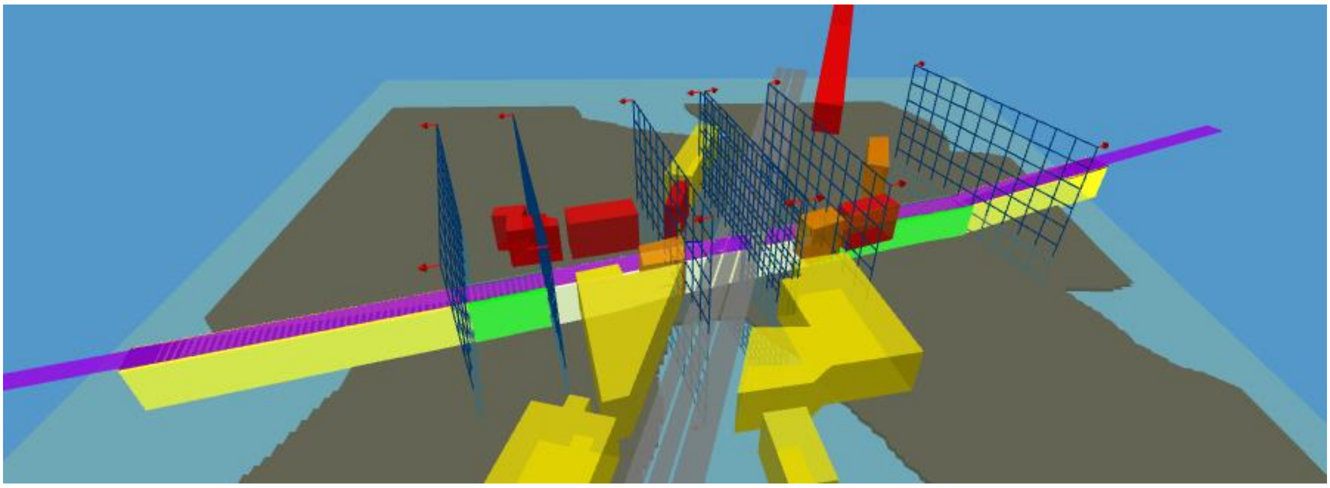


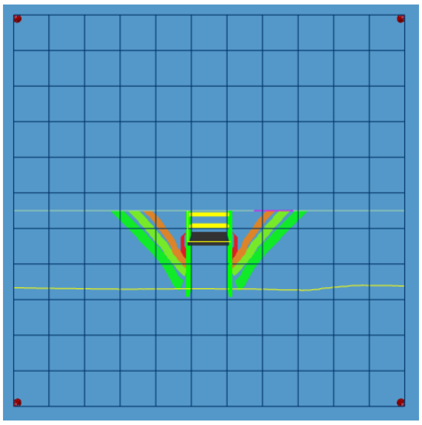
Figure 3-5 Oosterengweg tunnel, most economical and most environmental friendly design

According to SET the design shown above, costs 7 million euros and has an MKI value of 8 million. This is determined on the basis of 7 calculation cross-sections. In Table 2-2 the amount of building materials and structural elements are listed for each cross-section of this design

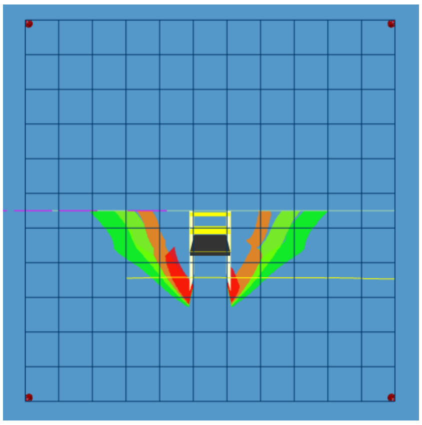
Cross-section	1	2	3	4	5	6	7	
Pit length	30	52	31	31	85	94	113	[m]
Wall Type	Diaphragm 800	AZ 46	AZ 46	AZ 46	Diaphragm 800	AZ 20	AZ 46	
Wall depth	22	21	21	21	20	19	20	[m]
Wall area	1337	2179	1313	1317	3416	3592	4553	[m^2]
Excavation method	Dry	Dry	Dry	Dry	Dry	Dry	Wet Gel	
Excavated m^3 soil	2369	5338	3784	3796	7260	2695	3911	[m^3]
Support type	Struts	Struts	Struts	Struts	Struts	none	Struts	
Rows	2	2	2	2	2	0	1	
Total struts	24	42	26	26	68	0	46	
Gel layer m^3	0	0	0	0	0	0	2504	[m^3]

Table 3-12 Properties of Oosterengweg tunnel

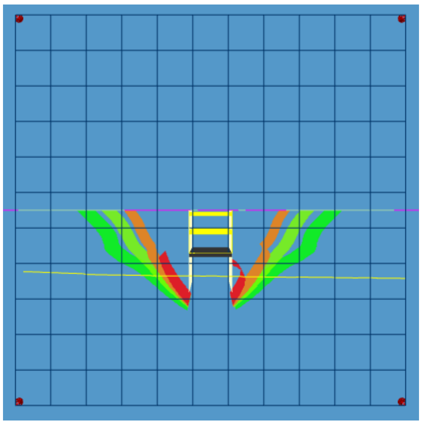
In Figure 3-6 the corresponding settlements for each cross-section of the most economical design are given.



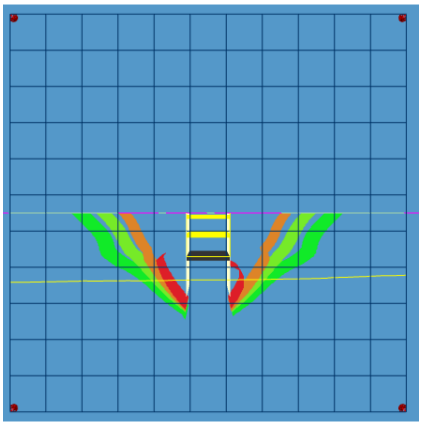
cross-section 1 calculation 13



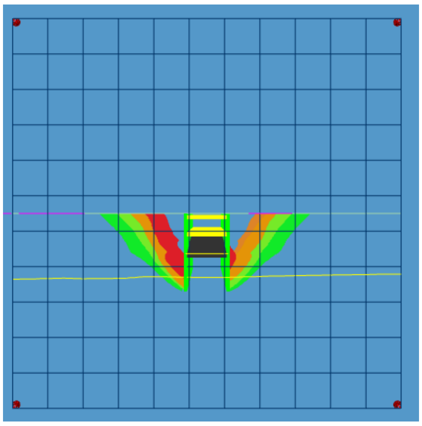
cross-section 2 calculation 9



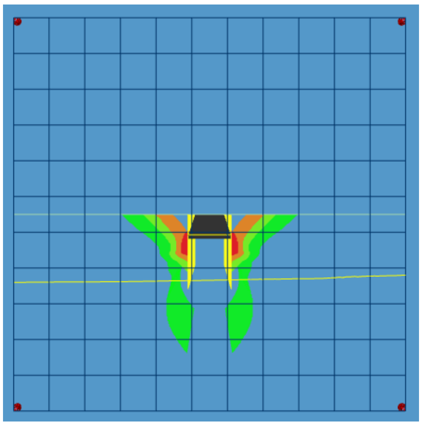
cross-section 3 calculation 9



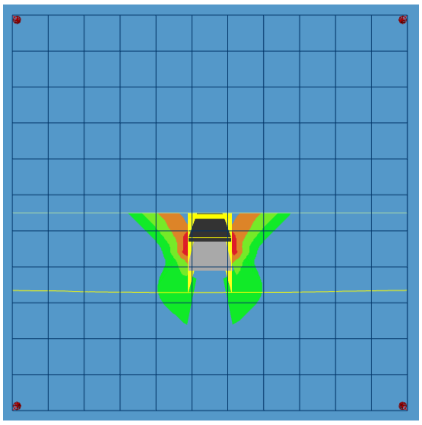
cross-section 4 calculation 9



cross-section 5 calculation 15



cross-section 6 calculation 23



cross-section 7 calculation 10

Figure 3-6, Oosterengweg settlements

3.1.3 HOV in ‘t Gooi Tunnel

3.1.3.1 Pre-process Results

The pre-process of HOV in ‘t Gooi tunnel resulted in 244 calculations. For each calculation cross-section in Figure 3-7, an average of 40 calculations were found. With an estimated search range of 750 calculations (Chapter 2.3.5.1), around 84% of the calculations are eliminated based on the requirements. The properties that are calculated by the design algorithm that are constant for each calculations of a single cross-section are shown in Table 3-13

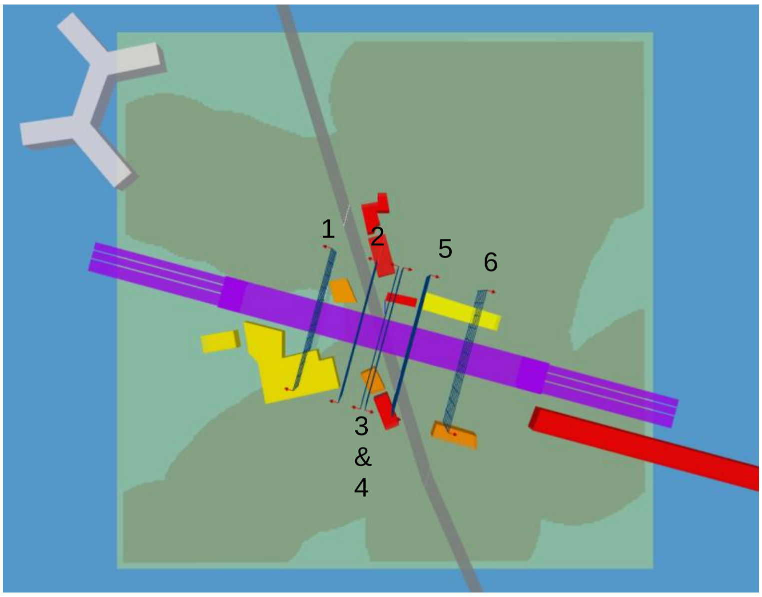


Figure 3-7 The defined calculation cross-sections for the HOV in ‘t Gooi tunnel

Cross-section number	1	2	3	4	5	6
Number of calculations	30	44	44	44	44	38

Cross-section number	1	2	3	4	5	6
Pit depth	-4.10	-5.83	-6.00	-6.00	-5.74	-3.59
Pit length	76.19	31.60	15.66	19.04	39.49	66.88
Floor angle	0.0545	0.0545	0	0	0.0545	0.0545
Clay layer depth	-19.98	-18.66	-17.63	-17.59	-17.82	-18.13
Min dept range	6	8	8	8	8	6
Support depth 1	-1	-1	-1	-1	-1	-1
Excavation depth 1	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5
Support depth 2	-2.30	-3.16	-3.25	-3.25	-3.12	-2.04
Excavation depth 2	-2.80	-3.66	-3.75	-3.75	-3.62	-2.54

Table 3-13 Number of calculations for each cross-section and their corresponding properties HOV in ‘t Gooi tunnel

An overview what building elements are used for each of these calculations is given in Table 3-14. These calculations are manually transferred to the Plaxis server. On this computer the calculations are processed, and their results are stored, this took around 20 hours, approximately 5 minutes per calculation.

Cross-section	1	2	3	4	5	6
Total calculations	30	44	44	44	44	38
Diaphragm 800	6	10	10	10	10	8
Diaphragm 1200	6	10	10	10	10	8
AZ 18	6	4	4	4	4	6
AZ 20	6	10	10	10	10	8
AZ 46	6	10	10	10	10	8
Wet excavation	30	36	36	36	36	38
Dry excavation	-	8	8	8	8	-
No struts	15	22	22	22	22	19
Struts 1 row	15	22	22	22	22	19
Struts 2 rows	-	-	-	-	-	-
Anchors 1 row	-	-	-	-	-	-
Anchors 2 rows	-	-	-	-	-	-

Table 3-14 Overview of the calculations for each cross-section

3.1.3.2 Post-Process Results

The Plaxis results of HOV in ‘t Gooi are shown in Table 3-15, this table gives an overview of what elements are used in the successful calculations. Of the initial 244 calculations, 140 calculations were found to be stable and were calculated successfully in Plaxis.

Cross-section	1	2	3	4	5	6
Total calculations	16	24	24	24	24	28
Diaphragm 800	4	6	6	6	6	7
Diaphragm 1200	4	4	4	4	4	7
AZ 18	-	-	-	-	-	-
AZ 20	4	7	7	7	7	7
AZ 46	4	7	7	7	7	7
Wet excavation	16	18	18	18	18	28
Dry excavation	-	6	6	6	6	-
No struts	8	16	16	16	16	16
Struts 1 row	8	8	8	8	8	12
Struts 2 rows	-	-	-	-	-	-
Anchors 1 row	-	-	-	-	-	-
Anchors 2 rows	-	-	-	-	-	-

Table 3-15 Successful Plaxis calculations (HOV)

In Table 3-16 the calculations, that meet all settlement requirements, and meet the ULS and SLS requirements, are given. An additional 26 calculations were eliminated in this process.114 designs were successful.

Cross-section	1	2	3	4	5	6
Total calculations	6	24	16	16	24	28
Diaphragm 800	4	6	4	4	6	7
Diaphragm 1200	4	4	4	4	4	7
AZ 18	-	-	-	-	-	-
AZ 20	4	7	4	4	7	7
AZ 46	4	7	4	4	7	7
Wet excavation	16	18	12	12	18	28
Dry excavation	-	6	4	4	6	-
No struts	8	16	16	16	16	16
Struts 1 row	8	8	-	-	8	12
Struts 2 rows	-	-	-	-	-	-
Anchors 1 row	-	-	-	-	-	-
Anchors 2 rows	-	-	-	-	-	-

Table 3-16 Successful calculations after Unity check and settlement requirements (HOV)

From all the feasible designs for each cross-section given in Table 3-16, SET has created a plot with the most feasible combinations, shown in Figure 3-8. To give a wide variety of designs, the design tool has also made a distinction between designs with unique building elements. This means that not only the most economical designs are combined but also designs with different building elements. All possible combinations of 6 calculations were examined. This amounts to 46.656 feasible designs.

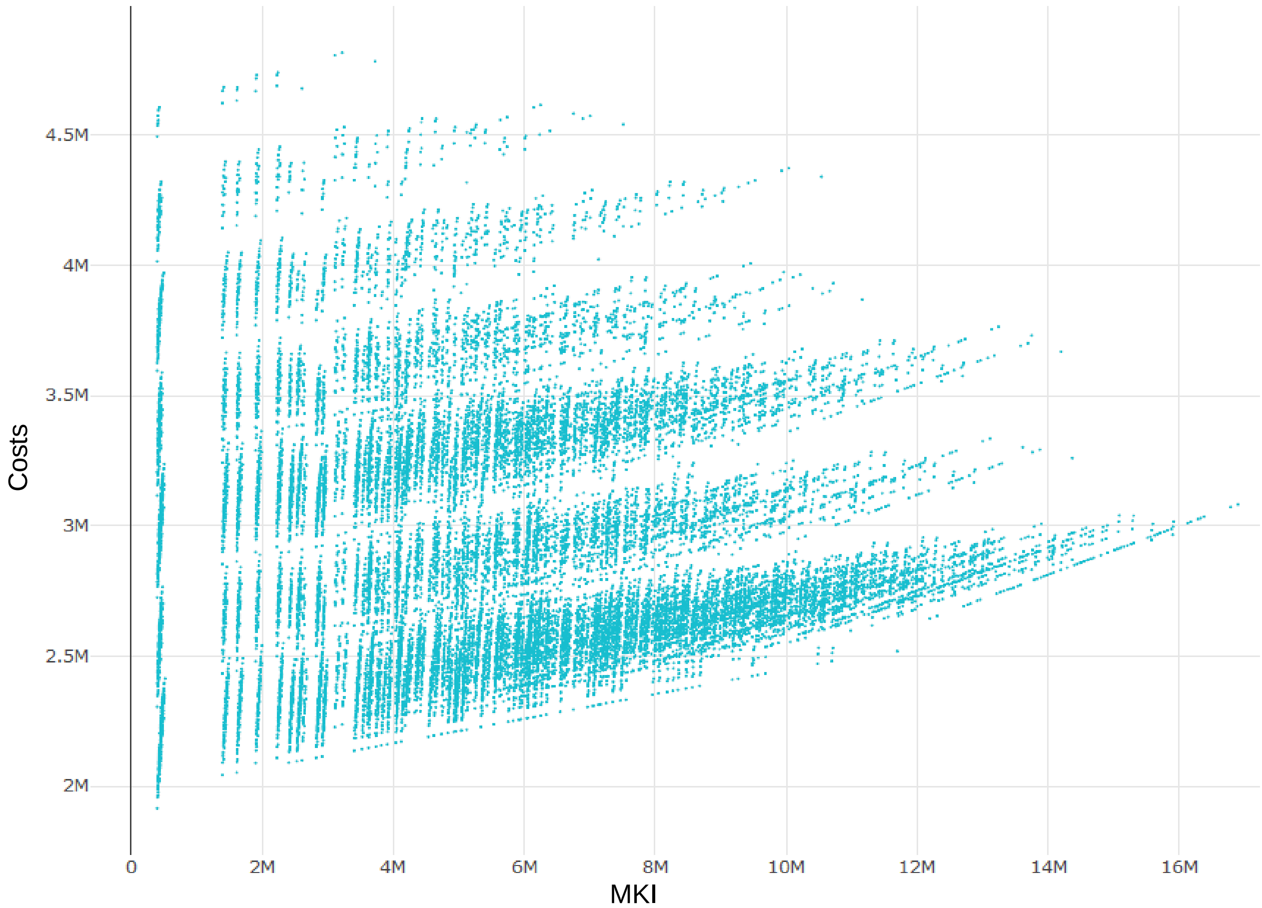


Figure 3-8 HOV in 't Gooi, feasible designs plotted

From the point cloud, shown above, the most economical and lowest MKI scoring design is shown in Figure 3-9. The cost estimate of this design is around 2 million euros and it has an MKI score of half a million. This design did not need any supports and is mostly build in dry conditions. All retaining walls are AZ 20 profiles. Other data about this design is given in Table 3-17.

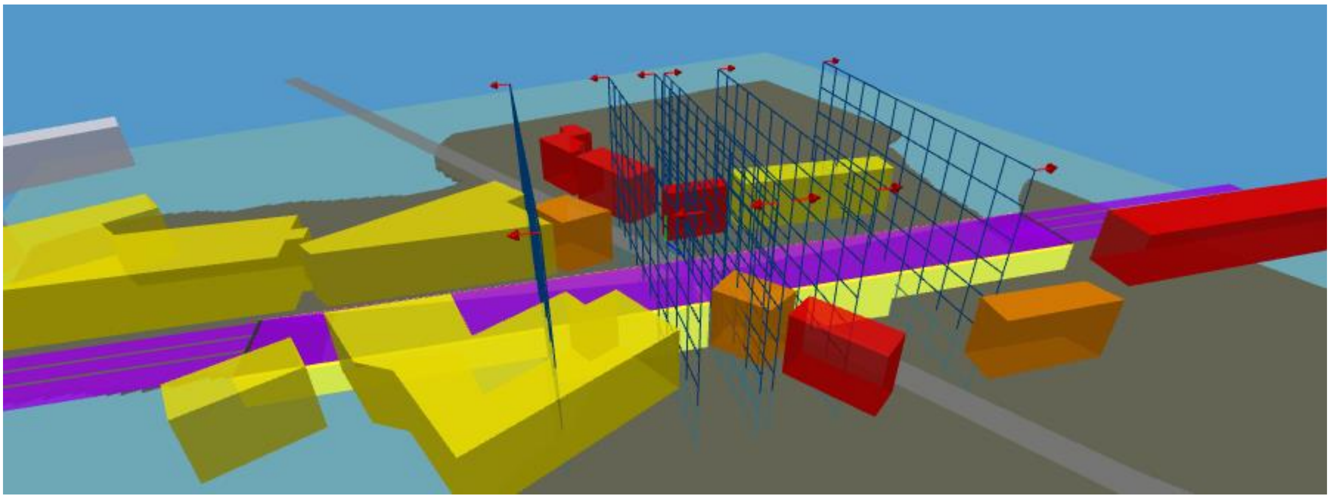
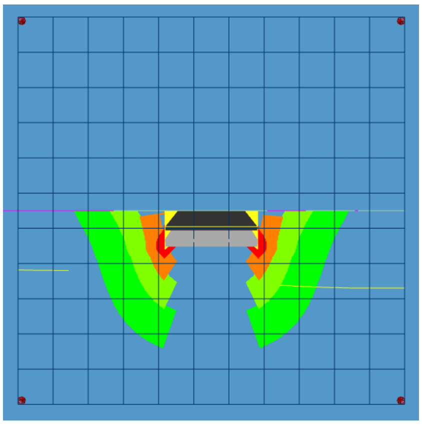


Figure 3-9 HOV in 't Gooi, most economical full design

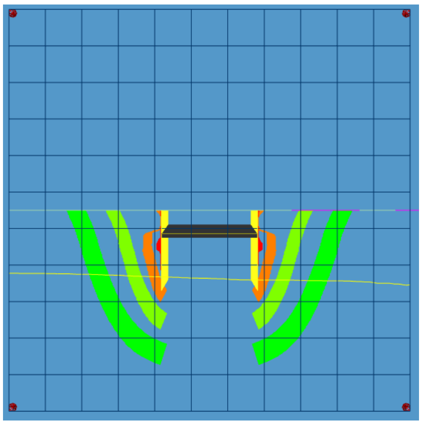
Cross-section	1	2	3	4	5	6	
Pit length	76	32	15	15	40	68	[m]
Wall Type	AZ 20	AZ 20	AZ 20	AZ 20	AZ 20	AZ 20	
Wall depth	10	20	20	20	20	10	[m]
Wall area	1524	1264	626	762	1580	1338	[m^2]
Excavation method	Wet Gel	Dry	Dry	Dry	Dry	Wet Gel	
Excavated m^3 soil	3705	3766	2255	2743	4421	2832	[m^3]
Support type	None	None	None	None	None	None	
Rows	-	-	-	-	-	-	
Total struts	-	-	-	-	-	-	
Gel layer m^3	3658	-	-	-	-	2549	[m^3]

Table 3-17 HOV in 't Gooi, properties of most economical design

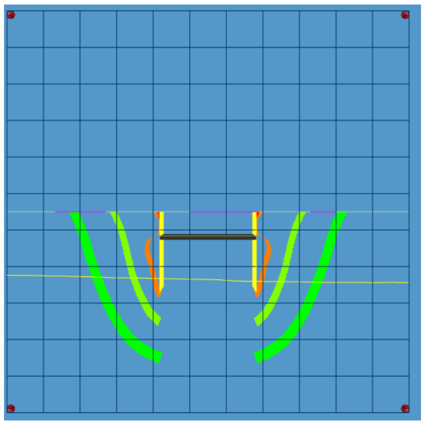
For each cross-section, the corresponding settlements are shown in Figure 3-10.



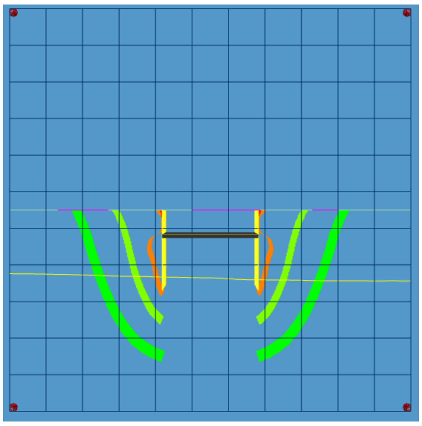
cross-section 1 calculation 3



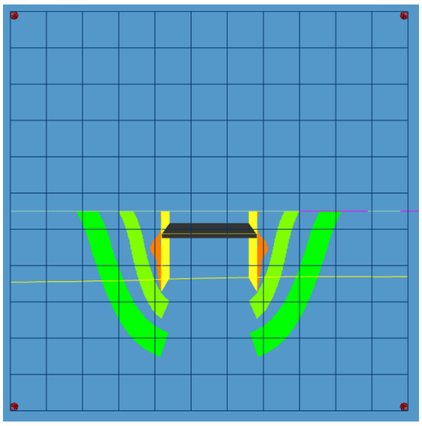
cross-section 2 calculation 10



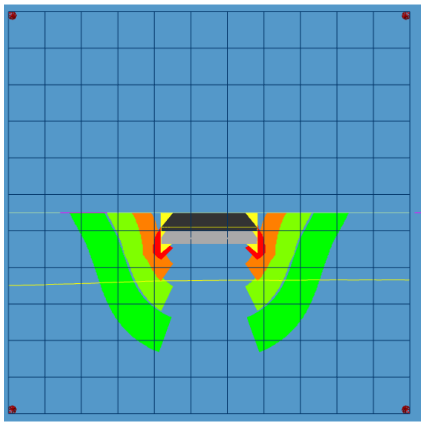
cross-section 3 calculation 10



cross-section 4 calculation 10



cross-section 5 calculation 10



cross-section 6 calculation 9

Figure 3-10 HOV in 't Gooi, settlements

3.3 Comparison between the test cases

In this chapter a comparison between the two test cases is made. It must be kept in mind the wet calculations of the Oosterengweg all failed in Plaxis because of the error in the Plaxis integration as described in Appendix C . From these failed calculations, favourable designs could have emerged.

The cost estimates of the best designs of the two test cases differ almost € 5 million, additionally, the MKI values almost differ 8 million points. The properties of each excavation pit are shown in Table 3-18. These designs are shown in Chapter 3.1.3.2 and 3.1.2.2.

	Oosterengweg	HOV in 't Gooi
Excavated soil m^3	29153	19722
Excavated soil €	€ 53.000	€ 46.000
Walls m^2	17707	7094
Walls €	€ 3.500.000	€ 1.000.000
Struts	232	0
Struts €	€ 3.190.000	€ 0
gel layer m^3	2504	6207
gel layer €	€ 313.000	€ 775.875
Pith length	450	250
Pit width	11	24
Depth	11	6

Table 3-18 Comparison between the two test cases

The largest cost difference is due to the retaining wall of the Oosterengweg, which costs € 2.5 million more compared to the HOV tunnel. This difference can be explained by two factors. The retaining wall area of the Oosterengweg is 10000 square meters larger than HOV in ‘t Gooi. This is because the length of the Oosterengweg tunnel 200 meters longer. Additionally, the depth of the tunnel is almost double. Because of this depth, the corresponding depth of the retaining walls is larger for the Oosterengweg. The average depth of these retaining walls is 20 m for the Oosterengweg (Table 3-19) and 14m for HOV in ‘t Gooi (Table 3-20). This partly explains the cost differences of the walls; however, the wall types are a big influence on the costs as well. The Oosterengweg is mainly constructed with Diaphragm walls 800 and AZ-46 sheet profiles. These are € 50 to € 75 costlier compared to the AZ-20 profiles that are used for HOV in ‘t Gooi (Table 3-3).

Cross-section number	1	2	3	4	5	6	7
Pit length	30	52	31	31	85	94	113
Wall Type	Diaphragm 800	AZ 46	AZ 46	AZ 46	Diaphragm 800	AZ 20	AZ 46
Wall depth	22	21	21	21	20	19	20

Table 3-19 Oosterengweg

Cross-section number	1	2	3	4	5	6
Pit length	76	32	15	15	40	68
Wall Type	AZ 20	AZ 20	AZ 20	AZ 20	AZ 20	AZ 20
Wall depth	10	20	20	20	20	10

Table 3-20 HOV in 't Gooi

The second major cost difference is due to the struts. The struts are quite expensive and cost € 1250 for a running meter. Because of the depth, the construction of the Oosterengweg requires 232 struts. With a width of 11 meters the total costs of the struts are around € 3 million.

The difference in the MKI scores is almost entirely caused by the diaphragm walls. The MKI score of a diaphragm wall is around 100 times that of a sheet wall. With a MKI value of 1610 for diaphragmwall-800, cross-section 1 and 5 of the Oosterengweg (Table 3-20) already have a score of 7.6 million MKI points.

3.4 Validation of the outcome

Based on the user input, the successful results from the post processing are concluded to be logical. However, an unexpected number of calculations failed in Plaxis. After the tests case was concluded it was found that underwater concrete floors were not added correctly to Plaxis. This meant that only an impermeable layer was added but not a solid structure. For this reason, many calculations failed in Plaxis, this elaborated upon in Appendix C . In Figure 3-11 it is clearly visible that the concrete floor is missing. The exact reason for this bug was not recovered.

During the validation of the outcome the following points were found:

- The depth range of the supports corresponds to the user input. However, a lower depth would make more sense and should have been used as user input.
- No designs are found to have anchor underneath buildings, which is a requirement from the tender HOV in 't Gooi.
- The depth range of the retaining walls of each design start from a sufficient depth.
- Only when a dry excavation is tested, the retaining walls extend into the clay layer, as can be expected from the design algorithm.
- The visual representation from the extracted data from the environment is visually assed to be correct and to correspond to the actual 3D objects.

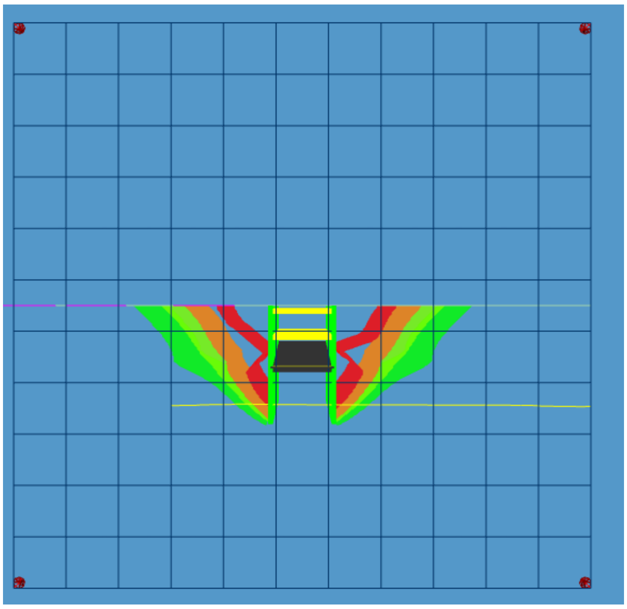


Figure 3-11 Absence of UCF

Oosterengweg tunnel

For the Oosterengweg tunnel, the following was found:

- The amount of calculations, that were found to be feasible by the pre-process algorithm, correspond to the expected amount of calculations.
- For most of the cross-sections, one row and two rows of supports were found feasible. Only calculations of cross-section 6 of the Oosterengweg tunnel allowed no supports (Table 3-9). This was the only cross-section of the Oosterengweg tunnel that had a lower pit depth than the maximum user input depth.
- Most of the cross-sections did not allow anchors.
- The ratio between dry and wet excavations is as expected. As the walls have to extend into the clay layer for dry excavation, which is located almost twice as deep as the pit depth.
- From the remaining building elements (5 wall types with different depth, 2 excavation types and 2 support types) an average of 50 calculations can be expected, this corresponds to the pre-processing results.
- The summation of the length of the design pit was 439 meters. Which was 11 meters shorter than the to be build object. However, two reasons for this deviation could be found in the 3D representation (Figure 3-12). The middle and back to back cross-section were not positioned close enough to each other and another cross-section should have been placed on a bent of the excavation pit.
- The post processing results are plausible. As could be expected from the settlement plots, many designs did not hold to the building requirements and unity check.
- The costs estimate and MKI values of each plot do show logical values. The most economical designs have the least structural elements, and the most expensive designs have the most elements or have anchors.
- Designs with the lowest MKI values are made of AZ profiles compared to the heights values with almost only 1.2m thick Diaphragm walls. The settlement plots show reasonable settlement contours.

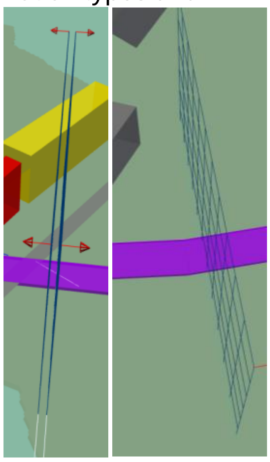


Figure 3-12 Length deviation

HOV in 't Gooi

During the test session, a mistake in the design algorithm was found. Cross-section 5 (Table 3-14) had no wet calculations in the pre-processing results. This mistake was due to the impermeable layer test. Other findings on the results were:

- The length of the tunnel added up to 248 meters.
- The number of designs proposed by the pre-processing algorithm is as expected.
- Most of the feasible design had wet excavations because the pit is shallow.
- The use of supports is correct, only one row of struts and no anchors are used. This corresponds to the depth limitations and the surrounding buildings.

- The results from the post-processing are valid.
- A reasonable amount of calculations was successful in Plaxis and the missing UCF did not pose a problem. This is expected because the depth of the excavation pit was at max only 6 meters. As a result, the forces acting on the retaining walls were much lower.
- For the assumption of a gel layer a dozen calculations did not extend into the required depth of this layer and were eliminated from the results.

- The costs estimate and MKI values of each plot do show logical values. The most economical designs have the least structural elements, and the most expensive designs have the most elements or have anchors.
- Designs with the lowest MKI values are made of AZ profiles compared to the heights values with almost only 1.2m thick Diaphragm walls. The settlement plots show reasonable settlement contours.

3.5 User friendliness

The user-friendliness was reviewed based on the front-end. From the session some bullet points emerged for each page of the front-end. These bullet point are shown below.

3.5.1 Input page

- The input page is clear and provides sufficient data for the conceptual phase.
- The use of calculation cross-sections is intuitive. However, the use of the key board shortcuts is not clear. This should be pointed out or told in a manual.
- The ability to click buildings is nice and intuitive, the colour change of the clicked object is a positive feature.
- The corresponding pop up table, provides sufficient data about the object, however when a value is changed in this table, it is not clear that the user should press enter to apply these changes
- It would be nice if the site automatically refreshes when the align button is pressed. Now it looks like nothing happens.
- Numbers to each calculation cross-section should be added.

3.5.2 Input page for each cross-section

- It would be a great addition if all user settings that can be adjusted in the design algorithm can be changed from within these pages and automatically update the proposed designs.
- The ability to add single designs manually would give the user better possibilities to find unique design that lie outside of the design range.
- More feedback should be given about unfeasible designs.
- The user setting for the design algorithm should be shown in this page.
- In the 2d view, the ground layers should be shown as a solid area.
- The ability to turn the surroundings off or on would give a better understanding of the calculation.
- For each calculation, a table should be provided with the amount of building materials and what structure elements that calculation consists of.

3.5.3 Results page

- The interactive plot gives a good impression of the estimated costs. However, when a dot is selected the dot should be highlighted.
- Labels should be added to the axis of the plot.
- The possibility to select other values on the axis could give more freedom to analyse different data.
- The sliders are a nice addition to scroll to single calculations of each cross-section.
- Unfeasible design should turn red when selected with a slider.
- The properties table should include the amount of building materials and number of structural elements.
- Numbers of each calculation cross-section should be added to the 3D view. It is unclear what data corresponds to what cross-section.
- The longitudinal 2D view could be improved.

3.5.4 Results page for each cross-section

- The ability to view settlement within the full 3D environment provides additional context. This could have a positive effect in the interpretation of the impact of the settlements. Especially the interaction with buildings is nice.
- The failed designs should be given a different colour. What designs did fail is not clear.
- Reference values to the settlements must be added. Without these values the settlement contours are useless.
- A link from the main results page to the corresponding cross-section would make the analysis of the results easier.

3.6 Review

This review is based on the comments of Koen and Jeroen during the test session.

The tool could have great potential in the interaction with potential clients. The ability to create 3D visuals right from the beginning, instead of in later phases, can help convey information and make more creative designs. From these 3D visuals limitations become visible and possibilities emerge. However, the limitations could be visualised better. Additionally, giving the client the ability to analyse designs from the comfort of their office could give them a better idea of the possibilities, especially because the environmental and surrounding requirements are immediately considered. SET provides the ability to let the client analyse designs without sharing valuable data and scripts and because the calculations are automatically executed on a remote server this would not cost anything. Although this visualization could be a benefit, the possibility exists that the simplicity, both from the made calculations as well as the representation of the 3D objects, could convey misleading information. This could give the client the impression that some designs are feasible when they are not, or the other way around.

Witteveen+Bos is interested in further development of the design tool. Future improvements would be to include drawing tools and to provide additional means for the input of the surroundings by full interoperability of GIS. Additionally, it would be interesting to make a connection with “Relatics” and “ANT”. “ANT” and “Relatics” are data bases and shared information platforms already in use by Witteveen+Bos.

4
Refine & Iterate

In this chapter, future refinements and iterations to the design tool are discussed.

4.1 Collaboration, interaction and 3D Visuals

Collaborative design

The design tool is developed to design a structure collaboratively and allowing each stakeholder to access all the data could have negative implications on the usage of the tool. Stakeholders might withhold to upload sensitive data because of the fear of it ending up in the wrong hands. To counteract these implications, a user management system could be introduced. This system could enable a stakeholder to decide to what extend their data is available and to which of the other stakeholders. Together the stakeholders can create user access levels that provide different levels of access to the designs and the data. Additionally, a tracking system can be added that registrates what data is accessed by whom. For now, when changes are made to values or calculations within the design tool, these changes are applied permanently and to all data for all users. A tracking system could register a history of each change and allow the users to go back to older data. A user management system could save these different design scenarios for each stakeholder. A communication system could be added to share the newly explored designs, add comments to designs, or request information from other stakeholders.

Pre-processing

The interaction with the design tool can be improved by additional possibilities to make real time adjustments. This would improve the usability of the tool and could give the possibility to define unique designs, which do not fall in the range of the design algorithm.

Post-processing

The possibilities to post process data could be improved to provide a better view of each design. A lot of data is generated within the design tool and this data is useful in the exploration of the design space. One possibility could be to give the users the opportunity to create their own analysis of the available data by allowing users to download the data or provide easy access to this data. As a second possibility, many more plots could be pre-defined in the front-end, or the data that is displayed should be selectable by the user. Button's and tick boxes could be added to let the user define what information should be shown on what axis. Thirdly, the possibility to program could be incorporated into the front-end. A user is then able to implement their own post-processing without openly sharing all data. The extent to which these options are applicable to each stakeholder could be managed by the access levels, as discussed above.

Draw application
A draw application should be included to quickly define the to be build object. Within this draw application the user should be able to draw a rectangular excavation pit or define a longitudinal road axis with depth and width dimensions.

4.2 Interoperability and Standardization

IFC
In later development of SET, the standardization format IFC should be incorporated into the model. The Python module IfcOpenShell (IfcOpenShell, 2017), which is currently in development, can be used to utilize this format. This module offers functionalities to read and write information contained in a IFC file.

GIS
The GIS support of the design tool could be upgraded to allow a user to quickly setup the tool for other projects. Most of the cities in the Netherlands provide GIS files that are openly available online, this data could be used to extract the surrounding buildings in other projects. The tool should acquire the functionalities to fully read these file, select the design area and automatically turn the 2D information into 3D objects.

4.3 Design algorithm

Dynamic design algorithm
A dynamic design algorithm will enable the user to search in “real time” to new and unique design solution. To do so, in future developments of the tool, a dynamic system should be introduced instead of the brute force algorithm. A dynamic system can only work properly if the integration of Plaxis is seamless. A seamless integration enables the design algorithm to analyse the results of Plaxis directly. From these results the design algorithm can decide what other calculations it should prioritize. For example, when a particular wall depth turns out to be stable and will fulfil all requirements, deeper walls do not need to be checked. The depth range that is used for the brute force algorithm can be abandoned and whenever a design is evaluated the Plaxis calculation can be run on the parallel computer.

Quickly analyse many designs
As discussed with Koen de Jong and Jeroen Hendriks, the potential of SET is not to find the optimal design within the large design space but having the ability to quickly analyse many designs in a short amount of time. Thus, being able to check out of the box ideas. The limitation of the amount of calculation cross-sections, is heavily based on the generation of the many combinations of designs of the full design pit. Each added calculation cross-section would exponentially increase the amount of possible designs. For this reason, the amount of calculation cross-sections was limited as discussed in Chapter 2.3.1. In future development

Object oriented programming
When more structural elements are added in the future and the structure of the design tool becomes more complex, the programming should be more object-oriented. To deal with more structural elements and the corresponding increase in complexity, the data management should be improved. Easier data management can be achieved with object-oriented programming. In object-oriented programming, data, functions and properties are assigned to digital objects. This give the programmer to manage the data, functions and properties of objects without having to reference to data bases. For example, for the building intersection with the cross-section (Appendix B.6), instead of a function that extracts data from two intersecting objects, the function can be incorporated in to the cross-section object. The intersected data is stored in the cross-section and can be processed by a function assigned to the object. This process happens for object, in this case the cross-section, only. If the design algorithm is assigned to an object, the corresponding calculations for each cross-section can be updated separately. This could enable more real time adjustments.

Retaining walls between two dry and wet excavations
Perpendicular retaining walls should be added to make the designs feasible. These are needed to protect one excavation pit from flooding or collapse while another is in construction. Perpendicular retaining walls might also be needed to seal an impermeable layer and prevent water seepage in the ground layers.

Parallel computation
When parallel computation is fully enabled many Plaxis calculations can run automatic and simultaneously at a dedicated server. Depending on the cost the operator is willing to pay for the server, all calculations could run at the same time, thus reducing the required calculation time to the time it takes to run a single calculation which is around 5 to 10 minutes.

Narrow down algorithm
When a brute force method is chosen in further development, an addition to the brute force method could be a narrow down algorithm. For each feasible design, this algorithm could analyse designs with small difference to the feasible designs to find better designs.

Buildings between two cross-sections
SET only extracts data from the buildings that the cross-section intersects with. However, many more buildings are affected by the settlements along the length of the excavation pit corresponding to a cross-section. SET can be improved if it incorporated all buildings between two cross-sections.

Reduction factor
The sloping floor of an excavation pit is not taken inconsideration in combination with the required building materials. Because the depth of the walls and the amount of supports, are considered at the location of the calculation cross-section, this could be significantly overestimated over the length of the pit. The corresponding depth of the excavation pit could be quite significantly higher at one end compared to the other. A reduction factor could be used to reduce this estimate. This could also be done in combination with values of the next calculation cross-section.

4.4 Integration

Security issues
When the server security issues are addressed the server can be opened to the internet, this would fully enable collaborative design.

5

Conclusion

In this thesis, as the objective stated, a conceptual partly BIM level-3 excavation site design tool has been developed. The developed tool, integrates parametric design with Plaxis, GIS and RockWorks. The developed tool helps stakeholders to visually explore many design solutions in a short amount of time, based on the costs, MKI, structural- and environmental requirements and the characteristics of the surroundings. The name SET (Smart Engineering Tool) was given to this tool.

The objective of BIM level-3 requires SET to be founded on BIM level-1 and -2. The design tool supports 2D and 3D visuals. For the design of the excavation pit, parametric design is used in an object-oriented manner, therefore the requirements for BIM level-1 are met. BIM level-2 requires standardization and collaborative design. SET has proven to be able to use standardization. Collaborative design is facilitated in the web based front-end. Data is managed and stored on a server, which acts as a central repository and ascertains that all stakeholders are working in a single model. Additionally, this server provides the communication between Python and JavaScript. A stakeholder is able to remotely adjust requirements and properties and is able to individually analyse the impact of these changes in the web browser. For this reason, SET can be classified as BIM level-2.

A design algorithm is built into SET, which is suitable for the design of excavation pits in different projects with different surroundings and requirements. The design algorithm can be used to design elongated excavation pits and excavation pits with a sloping floor. The design algorithm automatically creates excavation pit designs, based on user defined 2D calculation cross-sections. The user can add these calculation cross-sections along the excavation pit, preferable at normative locations. Based on the boundary conditions corresponding to these calculations cross-sections, the design algorithm creates designs which differ in structural elements, the geometry of these elements and their properties. For each design, SET takes the loads, maximum relative rotation and maximum horizontal strain of the surrounding buildings that intersect with calculation cross-sections into consideration. SET extracts the information of the surrounding buildings from a GIS model. The soil stratification and soil properties are extracted from RockWorks (geotechnical software). For each of the designs, the design algorithm of SET checks the vertical equilibrium of the excavation pit. It eliminates dry excavations without an impermeable layer, or it creates an impermeable layer made of gel injections or an underwater concrete floor. SET checks, based on Plaxis results, if the maximum relative rotation and the maximum horizontal strain of buildings is satisfied and performs a unity check on the bending moments of the retaining walls. The design algorithm is interoperable with Plaxis 2D, a geotechnical finite element programme, which automatically performs detailed calculations of each design. Plaxis runs on a parallel computer and its input is provided by the design algorithm. Apart of a manual transfer of files from the server to the parallel computer, the integration of Plaxis is fully automatic. SET has shown that the interoperability with Plaxis is possible and can function correctly.

SET can explore a large amount of designs in a short amount of time. For each calculation cross-section, up to 750 designs are examined by SET with default settings. Based on the maximum of seven calculation cross-sections, a quarter of a million combinations of excavation pit designs over the full length of the to be design object are examined, which is a fraction of the possible 131 * 10^{21} combinations. The results of Plaxis, the integrated external software, are automatically visualised. Soil settlements are turned into 3D objects, interacting with the deformation requirements of the buildings, and can be viewed in 3D. The costs, material usage and MKI values for each design are determined automatically and are shown in an interactive plot. Stakeholders can access the design tool from a web browser to compare and analyse the designs and draw conclusions on their own preferred designs. The 3D visualization of the designs helps the stakeholders to find an optimal design in a collaborative manner. Although, SET visualizes data in several ways, some processes in SET can be perceived as a black box. Especially the reason why calculations fail in Plaxis is not conveyed by SET, this is not shown because it is difficult to recover from Plaxis.

Based on the tender HOV in 't Gooi, two test cases were made for the level intersection of the train and bus lane with the 'Oosterengweg'. These test cases took around one hour to prepare in SET. For both test cases, the corresponding Plaxis calculations took around a day of computation time to finish. As a result, SET found 88 and 114 feasible designs for the calculation cross-sections for both test cases. The combinations of these calculation cross-sections amounted to a quarter of a million feasible designs. In total these test cases were fully worked-out in two days. The train and bus tunnel, that was deemed unfeasible, was found to be almost a third of the cost compared to the Oosterengweg tunnel. However, it must be noted that the costs to redirect the train tracks and the maximum obstruction time of the train tracks was not included in these designs. Additionally, an error in the integration of an underwater concrete floor in Plaxis occurred, the exact cause of this error is not resolved.

Almost all development criteria of this thesis are met, but it remains speculative whether SET can be labelled as 'smart'. The strength of the design tool lies in the considerable amount of designs that can be worked out in a short amount of time and the integration of automatic processes in the workflow. Variations in designs of the calculations cross-sections are not 'out of the box designs', but by combining all the cross-sections into one design, less obvious designs emerge. Even though it can be discussed whether the design tool can be labelled as being smart, the developed design tool does hold up to its potential and has shown six major benefits:

Extended design space

During the conceptual phase, the feasibility of a relatively large design space can be analysed within a short amount of time without being labour intensive. Within the large analysed design space, possible designs can be found, which in a conventional process, on the basis of expert experience and intuition, could have been deemed unfeasible or go unexplored because of labour and time constraints.

Decisions making based on full context

The decisions concerning the final design can be postponed to the very end. This allows the stakeholders to make design decisions at a later stage and keep a large design space in consideration throughout the design stages. Designs, deemed unfeasible in the conventual conceptual design stage, can emerge as feasible in later stages. Information in the design tool is parametric and each adaptation can propagate through the model. In this adaptation process, the model takes the context of the environment and the surroundings in consideration and, when additional context is added, calculations and their results change accordingly. The 3D visuals provide feedback to the stakeholders on what implications the alterations will have on the design.

Collaborative and shared decision making

The possibility of the collaborative platform stimulates shared and parallel decision making. Stakeholders can incorporate various requirements, limitations and responsibilities into the model throughout the design process. This input utilizes the knowledge and expertise of each stakeholder.

Realtime and remote evaluation of designs

The design tool can be used in meetings and discussions with clients to convey information in real-time; from the start, 3D visuals show the impact of each design together with their possibilities and limitations. The client can be granted remote access to SET and is thereby able to evaluate design solutions from the comfort of the office. SET gives the ability to embed expert knowledge into the designs and extends this knowledge to stakeholders, without sharing valuable scripts and sensitive or private data. The client can evaluate and investigate many designs, based on calculations, requirements, limitations and the quantifications provided by SET.

Automated parallel calculations
The automation of design processes allows the shift of many tasks to a server or remote computers. Calculations can be processed in parallel and automatically without the supervision of an engineer. The automation reduces the amount of required labour and increases the productivity of an engineer. This reduces the costs of the design process and allows engineers to focus on the designs that are more difficult to automate or to investigate “out of the box design”.

Integration of multiple design tools
The integration of design tools that are generally used separately reduces the manual labour that is normally needed to transfer data from one design tool to the other. SET can automatically use the soil stratification determined with RockWorks for the FEM calculations in Plaxis and the results are checked with the settlement requirements of the sorrowing buildings that is extracted from GIS.

Witteveen+Bos sees future possibilities in the developed design tool and has shown interest in the incorporation of the design tool with other automation projects they are currently working on. In the following years, smart design systems like SET and other digital developments that automate design processes by integrating many different models, software applications, the requirements and the characteristics of the environment into one model, will change the engineering practises in civil engineering.

6 Discussion and Recommendations

This chapter will provide a generic discussions and recommendations. A specific discussion about SET in the form of refinements and iterations is given in Chapter 4. Chapter 3.4 discusses the validity of the test results.

6.1 Discussion

In this master thesis, a BIM level-3 design tool has been developed by rapid prototyping. This tool was tested by 2 test cases in collaboration with Witteveen+Bos. The development of 'smart' systems and particularly Building Information Modelling (BIM) technologies are changing the conventional structural design processes. (Chi et al., 2014, p. 135). New information technologies show exponential opportunities regarding civil engineering (Xue, Shen et al. 2012). Several people in the industry speculate about what kind of impact of parametric modelling and BIM level-3 could have (Azhar, 2011; Chi et al., 2014; (Sacks & Barak, 2008). In this thesis an attempt is made to provide insight in the possible benefits of a smart BIM level-3 tool regarding a civil engineering project.

BIM technologies

A list of 5 predicted future development trends of BIM-enabled structural design is given in a review by Chi et al. (2014), which includes over one hundred articles about various BIM technologies and BIM tools. In the next sentences, these predictions by Chi et al. (2014), are quoted throughout the text. The first two of these predictions, were fully utilized in SET. These are: (1) Parameterised structural design and the (2) "possibility for early adoption of structural optimisation at conceptual stages". It is to be discussed whether the utilization of the following prediction holds for SET: (3) "Provide an innovative development of an effective decision-making support tool and related visualization technologies". Many points of consideration that are used during the decision-making process, are not captured by SET. Additional visualizations, next to the interactive plot and 3D visualizations could make SET a more effective decision-making tool. Moreover, the possibility to dynamically include additional requirements could increase the decision-making abilities of SET more effective. (4) "Strengthened cooperative works with data exchange abilities of BIM" is something that should be researched regarding SET. Whether or not SET provides strengthening cooperative design abilities cannot be concluded from this thesis as this is not researched. However, attempts are made to provide collaborative design possibilities. The most valuable collaborative design ability is the fact that the tool can be accessed from a web-browser. The use of a web-based system has as a benefit that web-browsers are free and are available on almost any computer or digital media. This could reduce the threshold for stakeholders to use the tool and to engage in collaborative design. Lastly, Chi et al. predict the development of (5) "high performance numerical methods for large-scale optimisation problems". In this thesis the brute force algorithm that is used for the optimisation is proven to be effective but is not of a high performance numerical nature. It is sufficient for this version of SET, with the current capabilities. However, for an upscaled version of SET, it is recommended to research a more suitable and 'smart' algorithm. What algorithm this might be should follow from this future research.

Features and Capabilities

In addition to the speculation about the effect and impact of BIM level-3, the literature proposes different features that should be included in a BIM tool. All definitive features of a BIM model are not yet established. Azhar (2011) identifies eight different features that should be included in a smart design tool. It can be concluded that three proposed features are fully included in SET. These are (1) “Visualization, 3D renderings with little additional effort”, (2) “Cost estimating and automatic material quantities updates and extraction”, (3) “Construction sequencing”. The following feature is included in SET to a lesser extent. (4) “Conflict, interference, and collision detection”. SET has a simplistic collision detection, only anchors beneath buildings can be detected and the interference of settlements on buildings is supported. Additionally, collision detection is used to extract the building information as discussed in Chapter 2.3.2. No other form of conflict or collision detection is used. The following three possibilities are not addressed in SET: (5) “Code reviews for fire departments and other officials”, (6) Forensic analysis: graphically illustrate potential failures”, (7) “Facilities management, usable fore space planning, and maintenance operations”. Although these features are not assessed in SET, the way SET is developed makes the integrations of these features easily possible. In the future development of SET, the 3D environment that is included in SET could be utilized to facilitate these features. For the forensic analysis, marginal feasible designs could be highlighted. For the management facilities, planning and scheduling tools should be added. Lastly it is recommended that the following feature, proposed by Azhar (2011), is to be included in SET first: (8) “fabrication/shop drawings: to generate shop drawings for various building systems”. Based on development criteria and the available time, the features that have been added to SET satisfy the objective of this thesis. The features correspond to the main objective and the scope of this thesis. On top of that, without the proposed features, SET has shown to have promising benefits, as discussed in the conclusion. When the tool is going to be further developed it is recommended to add two additional features: full GIS support and parallel computation. The GIS support of the design tool could be upgraded and automated to allow a user to quickly setup the tool for other projects. This way the tool could be used during meetings in real time without prior preparation. When parallel computation is fully enabled many calculations can run automatic and simultaneously at a dedicated server without the supervision of engineers or operators. This could give the opportunity for clients to evaluate new and unique design without sharing sensitive data and scripts.

User interface

For the user interface of a BIM tool, Singh, Gu, and Wang (2011) gives 4 criteria which are: (1) “model tree view position and 3D viewer position”, (2) “support for online real-time viewing, printing and mark-ups”, (3) “the ability to click on an object, and check what sub-sets it belongs to”, (4) “the ability to click on an object, and switch between the sub-sets it belongs to, for another sub-set selection”. From these criteria, SET partly satisfies the first three criteria. From SET, a model tree and sub-sets cannot be viewed, however, sub-sets are somewhat used in SET. For example, the excavation pit is a set and the retaining walls, floors, and supports are its sub-sets. The ability to click an object is made possible, however the sub-set is not shown in the front-end. Such a viewer and the ability to switch between sub-sets can be added together with the model tree.

Development method

In this thesis the method of rapid prototyping is used. This method is widely used for the development of prove of concepts. Because the tool is developed from scratch, the rapid prototyping approach is chosen. One of the benefits of rapid prototyping is that every time something small is prototyped, the addition is tested and iterated immediately. This guarantees that everything is working correctly before new features are added. However, the downside is that other features rely on something that is prototyped in an earlier stage. The developed parts from earlier stages have to be redesigned to incorporate these new additions. This means that some work during the development of SET is reworked several times. A method like the V-model (Consulting, 2001-2018a) that focusses on detailed project definitions, analysis of the requirements and software architecture, prior to the coding could prevent these reworks. However, such project definitions should be detailed, and the software architecture must provide a solid foundation for the coding and further development. At the start of this thesis, the required coding skills and software development experience to use the V-model was not present. This also hold for the waterfall method (Consulting, 2001-2018b), a linear method from which the v-model was derived (Consulting, 2001-2018a).

Feasibility of the designs

In the problem definition it was stated, that as of today, only a handful of different alternative designs can be worked-out within the given amount of time during the tender phases. It takes several weeks to work out a design of an excavation pit. With the design of SET this time problem is tackled, as seen in the test-case results in Chapter 3 SET makes it possible to create a huge amount of designs in a relative short amount of time taking in to account all the different requirements. SET also reduced the amount of manual labour to combine all the different software outcomes. However, in the test-cases, as described in Chapter 3, an unexpected number of calculations failed in Plaxis. After the test case it was found that underwater concrete floors were not added correctly to Plaxis. This meant that only an impermeable layer was added but not a solid structure. It was tried to run the calculations several times but still a bug occurred. The exact reason for this bug was not recovered but, in the future, this need to be evaluated. SET has shown that the interoperability with Plaxis is possible and can function correctly. The feasibility of the produced designs is correct within the limited evaluated structural and environmental requirements and the characteristics of the surroundings.

6.2 Recommendations

Data structure

In future development of SET, the data structure should be based on the extensive analysis of an excavation pit in combination with an extensive analysis of the required processing steps in SET. For each processing step the required input and the processed output should be analysed, as well as what data should be visualised. From this analysis, a data tree should be constructed in detail. In this study, a flowchart was created at the start with the knowledge available at that time. It is recommended to create a detailed data tree at the start as it is of added value for the entire development of the design tool. This could lead to easier implementation of new features and make the interpretations of data in the post-processing easier. The data of SET, that could have been combined in a single file, is sometime split over several files because in later development stages additional data, that was not included in the files, was needed.

Object-oriented programming

The program language that was chosen to design this tool was based on several consultations with members of Witteveen+Bos and on internet research and tutorials. The use of different programming approaches is clearly noticeable throughout the development of SET and can be correlated with a personal learning curve. Scripts written in the beginning are highly based on procedural and functional programming; in the scripts written in the end of this thesis, object-oriented programming starts to emerge. Because of the relative limited complexity of SET and manageable amount of processed data, the use of procedural and functional programming is manageable and will not create many bugs or lag.

Standardization

Standardization is a key element in a BIM level-3 tool and for this reason the standards CSV and "Wavefront .obj" are used. However, the main standardization format in BIM is Industry Foundation Classes, IFC (Eastman et al., 2011). IFC is a complex and detailed standard format that can contain almost every property or information about objects within a model. This includes: the 3D model of that object, the materials, requirements and many other properties. Full integration of IFC was not part of the main objective and would have made the development of a conceptual design tool to difficult for the available amount of time. For this reason, the IFC standard is not used in this thesis. With additional time and effort, the IFC standard can be adopted by the design tool. In a later development of SET, the standardization format IFC should be incorporated into the model. The Python module IfcOpenShell (IfcOpenShell, 2017), could be used to utilize this format. This module offers functionalities to read and write information contained in an IFC file.

Optimization algorithm

One of the main goals of SET is to help find optimal designs in a set of infinite possibilities. Some of the different methods that are used to find an optimal design in these circumstances are: Brute force, a 'dumb' algorithm that examines every combination within a given range, mathematically, machine learning and artificial intelligence. For the development of SET a brute force method was chosen, because it was relatively straightforward to implement in SET. The design algorithm was one of the BIM technologies that had to be incorporated. The use of more efficient algorithms could be a thesis topic of its own. Additionally, the brute force method was sufficient for the need capabilities of SET as the structural elements that are supported by SET are limited. In future developments, the need for a more elaborate design tool could emerge. However, this could lead to problems when SET is not redesigned. On the one hand, upscaling in the number of structural elements will pose several problems to SET. The brute force method that is incorporated into SET has as a drawback that each added variable exponentially increases the amount of calculations. This could lead to high calculation times and would make the tool lag.

Programming langue

In this thesis, a combination of Python and JavaScript with WebGL was used for the development of SET. The js module called “Three.js” (Cabello) provides functions to utilize the WebGL capabilities. Because WebGL is browser based and almost all browsers support WebGL (Fyrd, 2018) , the support is independent of the operating system of a user. However, because WebGL is fairly new, it is still gaining support by developers and by hardware manufactures. The hardware support by different Graphics Processing Units (GPU), specialized hard ware for graphics and visuals, is limited to GPUs of the last decade. Not many programming languages work with WebGL; software support is limited to mainly JavaScript. A mid-level JavaScript module called “Three.js” has predefined some basic functionalities and, thus, makes the use of WebGL easier. An additional benefit of this library is that it is browser based, an online browser is free, and the use of web-based applications do not require the installation of additional specialized software. This makes the tool easily accessible and boosts simplified collaborative design. Other computer languages each have their own pros and cons. A low-level programming language forces the developer to start from the fundamentals and grants full control over development of features. To use these low-level programming languages, a full understanding of the computer on a hardware level and knowledge of the theory behind 3D graphics and rendering is required. On the other hand, a high-level programming language, a language with strong abstraction from the details (Run Digital, 2007), simplifies the process of developing a program, as many tasks are pre-developed. The downside of a high-level programming language is the loss of control and computing efficiency. Aside from these languages, existing 3D application software could provide a solid foundation with complex functionality but limits the flexibility and room for modifications.

Feasibility of the output

In this thesis, it is chosen to have the highly simplified designs consisting of the minimal amount of building elements, because that was not the main topic of this thesis. The challenge for the future development assesses all structural requirements to make SET generate fully feasible designs. For example, SET should evaluate transition between two calculation cross-sections. Especially the transition between a wet and a dry excavation might need additional measures. These excavation pits are constructed in a completely different manner and cannot be constructed simultaneously.

Integration with databases

Singh et al. (2011) argues that a BIM-server should provide a centralized data repository for the building project. In SET this is done by the Django server, which was setup to integrate Python with JavaScript. Additional research is required into the integration of SET with databases. From this data, requirements from the surroundings and environment could be used. Reference projects can be used to determine likely feasible solutions and the design algorithm could prioritise these in the calculation order. Additionally, the data from reference projects can improve cost estimates. Not only the cost estimates of the designed structural elements could improve but also a cost estimate could be made of other elements like installations, road surfaces, safety measures and other non-structural works. The option to write data from the design tool to the databases could also have benefits. For example, calculations that have been performed in a prior project could be used when conditions are almost equal. This could speed up the calculation process.

6.3 Future development recommendations

Additional requirements

Upscaling in the number of requirements posed by the surroundings and environment could have a positive influence on the calculation time. Most of the scripts that are written for this purpose are not computationally expensive and take in the order of milliseconds to compute. Especially the requirements that can be checked before the Plaxis calculations can reduce the number of possible design. This would have little impact on the usability of the tool and will not slow it down. It could even speed up SET because especially Plaxis calculations are computationally expensive and take in the order of minutes to compute. Additional environmental requirements could eliminate calculations and thus, such eliminates heavy computation.

Simplicity

Although, it is very apparent to add additional functions and elements to the model, the simplicity of the tool has shown some interesting benefits. Especially during the conceptual design stage, the ease and speed of the tool can be very useful during meetings and discussions with a potential client in which, the possibilities and limitations can be conveyed in 3D and for many designs. Designs ideas of a client can be underpinned on the spot. For this reason, I recommend keeping SET simple and small. An all-enveloping programme should be redesigned from scratch.

Level of detail

Accurate representations of structural elements can give additional information and these detailed drawings could raise additional points of attention. The advantage of the simplified representation by SET has as benefit that the design tool is fast and responsive. This benefit could be nullified when drawings become to detailed. Additional research might find a nice balance.

Safety factors

In the future, the possibility to clearly define safety factors from the user interface, should be built into the tool.

Modular and programmable tool

Every project is different and has its own requirements and surroundings. This makes it very difficult to develop a simplistic tool that is usable in almost any project. If such a tool would be developed many options and exceptions should be added to the tool. Even then, a requirement or customer wish might not be evaluated correctly. For this reason, a modular tool could provide the agility and adaptability that is needed for different project. This could be achieved by giving the users and operators of the tool the ability to write small lines of code and insert these in the design algorithm in the desired location.

Integration with data bases

Additional research is required into the integration of SET with databases. From this data, requirements from the surroundings and environment could be used. Reference projects can be used to determine likely feasible solutions and the design algorithm could prioritises these in the calculation order. Additionally, the data from reference projects can improve cost estimates. Not only the cost estimates of the designed structural elements could improve but also a cost estimate could be made of other elements like installations, road surfaces, safety measures and other non-structural works. The option to write data from the design tool to the databases could also have benefits. For example, calculations that have been performed in a prior project could be used when conditions are almost equal. This could speed up the calculation process.

Cooperative design

In future research, the benefits of cooperative design should be researched.

Design algorithm

It is recommended to research a more suitable and ‘smart’ algorithm, that is usable for more complex and detailed designs.

Programming

Last but not least, I recommend engineering companies to invest in education in programming. Many small tasks and problems can be completed simple and fast with a programming script. Whit some basic understanding of programming, an engineer can write small lines of code to automate repetitive tasks. Many new programming languages like Python, Julia and languages with a node editor, have a large abstraction level and can be used without detailed understanding of the underlying processes. These new programming languages can be written in a sentence like syntax and filter out syntax errors.

Appendices

A
Programming languages

A.1 Cad Programmes

Rhinoceros or Revit, two examples of fully developed CAD applications, provide an extensive set of drawing rendering and modelling capabilities. Both have add-ons that fully support parametric design. These are Grasshopper and Dynamo. Galapagos, an add-on to Rhinoceros, provides support for algorithmic design, however, the flexibility in the use of Rhino or Revit is limited. The difficulty of creating an integrated design tool will lie largely in the conventions that are used and set out by Rhino and Revit combined with the complexity of these applications. Knowledge of the internal functioning of these programs is required and as these programs are everything but open source this might prove to be very challenging. On top of that the knowledge gained is very specific and can only be used in combination with these applications. To meet the thesis objectives a flexible approach is needed. The establishment of interoperability requires full control of the design tool and collaborative design calls for a cheap and accessible platform. For this reason, low-level and high-level programming languages were examined and will be discussed in the next paragraph.

A.2 Graphics library

Graphics libraries are APIs that facilitate the communication between the application and the GPU. A dozen different graphic libraries are available. Three of these libraries and commonly used development software are discussed.

A.2.1 DirectX

DirectX is developed by Microsoft and is one of the most used libraries for games and graphics in general. DirectX is almost exclusively supported by Microsoft's operating systems. The library has a large developers base and much documentation is available. DirectX is not open source and development software is expensive. Two examples of development software are the game engines, unity and unreal. Both engines have extensive functionalities that simplify the use of graphic libraries. This gives the engines many possibilities and complete control over the graphics library. The high-level game engines have large abstraction from the low-level graphic libraries. This means that the knowledge gained about these programs are not interchangeable with other programming languages. A personal objective is to learn about programming in general. The use of these game engines would make the development of the tool for this thesis much easier but would limit the understanding of the underlying processes.

A.2.2 OpenGL

OpenGL is developed by Silicon Graphics and maintained by the Khronos Group and is an open source graphics library that is, in contrast to DirectX, supported by many operating systems and almost any GPU and can be used independently of the programming language. For that reason, OpenGL is used for many CAD software application as a wide variety of operating systems, like Linux or MAC OS, are used by the engineering companies that use the software. Because OpenGL is independent of the programming langue it can be used in combination with Python. However, the combination of Python with OpenGL might be useful, the Python development support for OpenGL is limited. This would require almost everything to be written in a low-level and would make it very difficult and complex.

A.3 OpenGL with Python

At the start of this thesis, an attempt was mad to use OpenGL in combination with python. The core processes of OpenGL were studied, and this formed the basis on which SET was developed. Some of the studied processes are shown below.

A frustum is the area in which objects are visible in the 2D window of 3D environment (Figure 0-1). The objects that are within this area are processed by the GPU, all other objects are not rendered. This reduces the computation load of the 3D representation. The green and red plane, are the near and far clipping plane. These clipping planes filter objects that are too close to the camera or too far away and, thus do not contribute to the scene. The field of view is the angle of the lens. A large field of view a large scenery just like a fish eye lens. In this thesis an angle of 45 degrees was used.

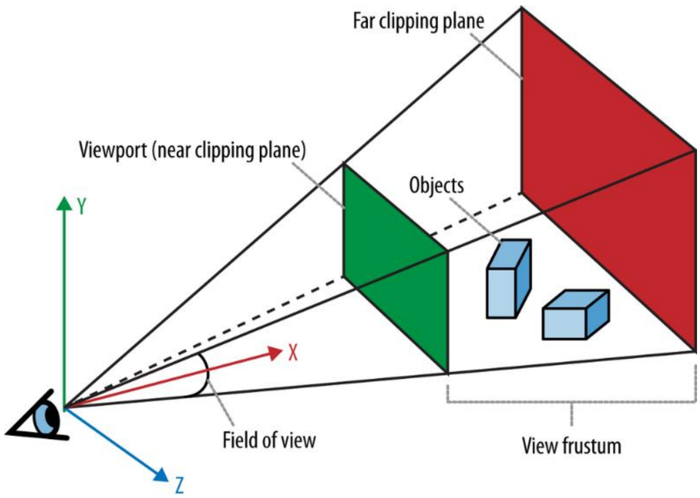


Figure 0-1 Frustum OpenGL https://www.safaribooksonline.com/library/view/programming-3d-applications/9781449363918/ch01.html

OpenGL has several stages in processing data in the GPU (Figure 0-2). The local space is the space that corresponds to the origin of an object. In this space the object can be manipulated. These manipulations only happen to that object. The object has a position towards other objects. They all have a position in the world space. In this world space all object come together to create a scene. manipulations to objects in the world space are relative to the origin of the world. However, to observer the scenery, a view space has to be determined. This view space is what an imaginary camera would see without perspective. To create the perspective, the clip space or the frustum as discussed above is determined in the view space and leads to the clip space. The coordinate system of this clip space is from -1 to 1 in the x and y direction with its origin in the centre. However, many windows or screen where the data is displayed, have their origin in one of the corners. For this reason, the data has to be converted to the space of the window.

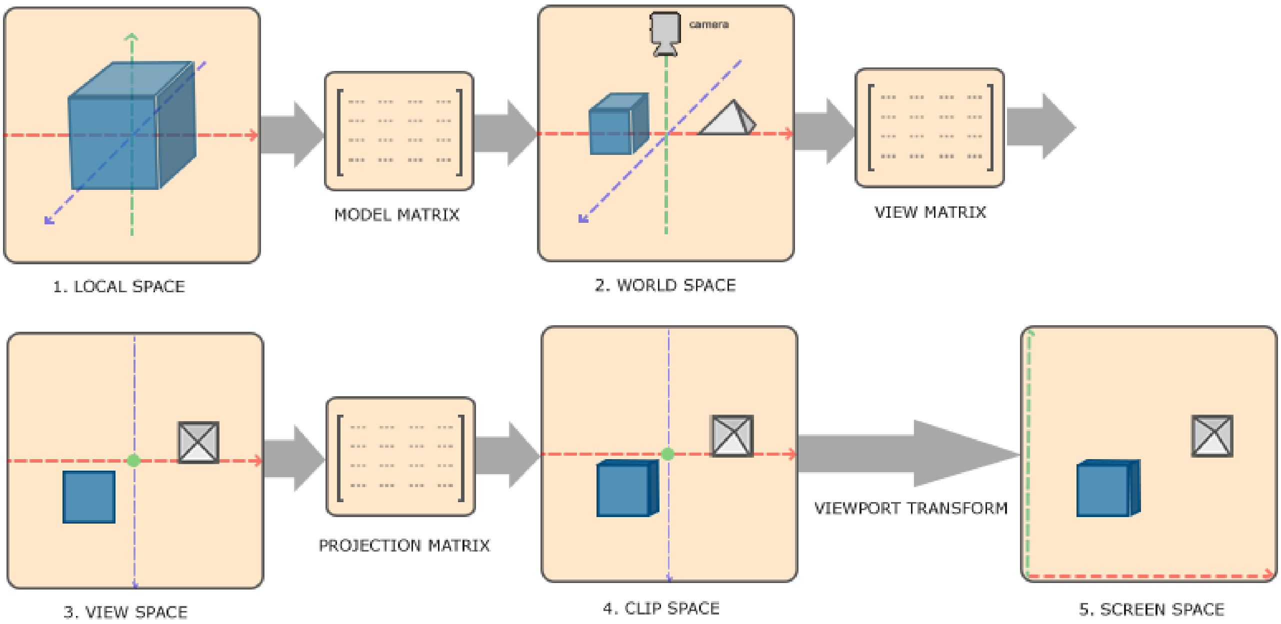


Figure 0-2 OpenGL spaces (Vries, 2018)

Alpha blending is used to determine the colour of two overlapping transparent objects. This is shown in Figure 0-3. On the left the alpha value is almost 0, in the middle the alpha value is 0.5, on the right the alpha value is almost 1.

$$\text{displayed color} = \text{alpha} * \text{colour object 1} + (1 - \text{alpha}) * \text{colour object 2}$$

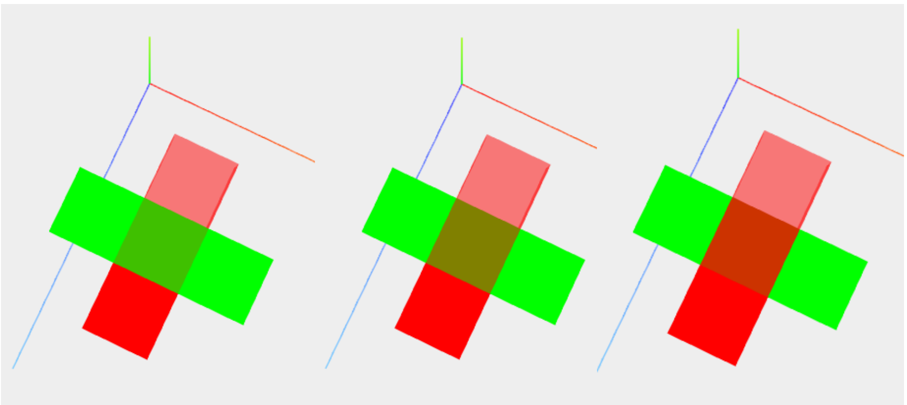


Figure 0-3 Alpha blending.

Object normals are lines that are calculated by the GPU and are perpendicular to the edges and faces of an 3D object. These normals are used by the GPU to determine light reflection of objects. Additionally, these lines are used for many more applications. In this thesis, these lines are used in the intersection of the calculations cross-sections.

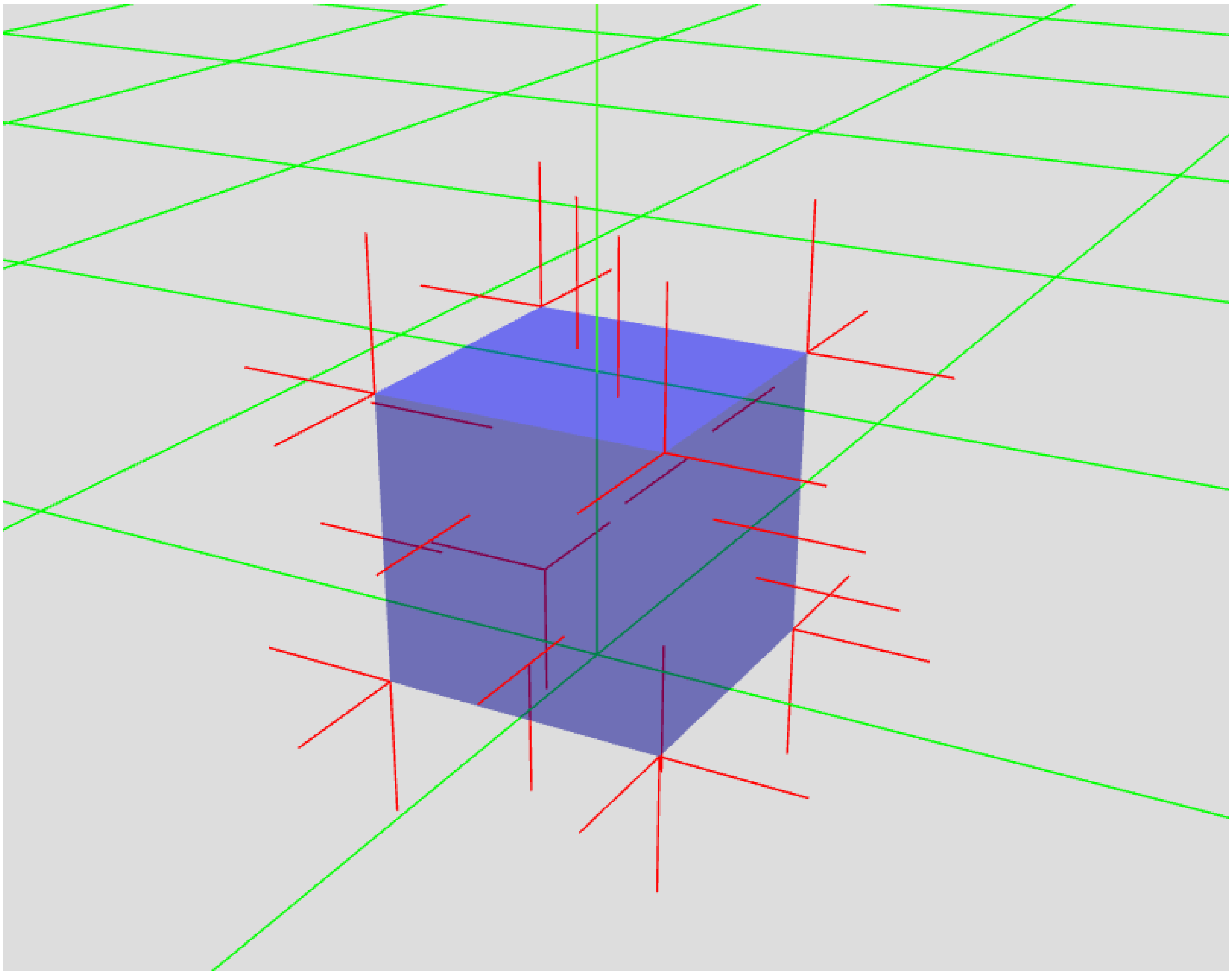


Figure 0-4 Object normals

B
SET features and functionalities

B.1 Supported objects and their properties

Object	Properties
Retaining walls	End bearing, EA, EA_2, EI, d [m], nu, w [kN/m/m], Mp, Np, Np_2, Rayleigh Alpha, Rayleigh Beta, maximum length
Anchor Rod	EA, spacing
Anchor Body	E, spacing, T_{end}, T_{start}, d [m], w [kN/m/m]
Struts	EA, EA_2, EI, w
Rotational connections	Rotation degree, stiffness, Mp
Soil	DrainegeType, E_{oed}^{ref}, E_{50}^{ref}, E^{ref}, Interface Strength, R_{inter}, C'_{ref}, $\gamma_{saturated}$, $\gamma_{unsaturated}$, φ', ψ, ν, kh, kv,

B.2 Supported sets of construction phases

Nr	objects	Construction steps as calculated in Plaxis
1	Dry, no supports	Initial phase, install retaining wall, drain bottom, excavation bottom
2	Dry, 1 row of struts	Initial phase, install retaining wall, drain level 1, excavation level 1, install strut level 1, drain bottom, excavation bottom
3	Dry, 2 rows of struts	Initial phase, install retaining wall, drain level 1, excavation level 1, install strut level 1, drain level 2, excavation level 2, install strut level 2, drain bottom, excavation bottom
4	Dry, 1 row of anchors	Initial phase, install retaining wall, drain level 1, excavation level 1, install anchor level 1, drain bottom, excavation bottom
5	dry 2 rows of anchors	Initial phase, install retaining wall, drain level 1, excavation level 1, install anchor level 1, drain level 2, excavation level 2, install anchor level 2, drain bottom, excavation bottom
6	Wet, no supports	Initial phase, install retaining wall, excavation bottom, install floor, drain bottom
7	Wet, 1 row of struts	Initial phase, install retaining wall, excavation level 1, excavation bottom, install floor, drain level 1, install strut level 1, drain bottom
8	Wet, 2 rows of struts	Initial phase, install retaining wall, excavation level 1, excavation level 2, excavation bottom, install floor, drain level 1, install strut level 1, drain level 2, install strut level 2, drain bottom
9	Wet, 1 row of anchors	Initial phase, install retaining wall, excavation level 1, excavation bottom, install floor, drain level 1, install anchor level 1, drain bottom
10	Wet, 2 rows of anchors	Initial phase, install retaining wall, excavation level 1, excavation level 2, install floor, drain level 1, install anchor level 1, drain level 2, install anchor level 2, drain bottom, excavation bottom

B.3 Coordinate system conversion

To supply Plaxis with the needed information, data must be extracted from the design tool. All data shown in the design tool are in three dimensions and Plaxis requires two-dimensional data. This requires some data conversions. Because Python and Three.js both work with different coordinates systems, the plots shown in this chapter might look mirrored or upside down, however, they do show the same data. These plots are not corrected because they are normally not visible and only used in this report to show that the data extraction works correctly. For this reason, these coordinates systems are not adjusted. The origin points of the 'ESRI ASCII' coordinate system, however, had to be converted to the Three.js coordinate system. The coordinates systems are shown in Figure 0-5.

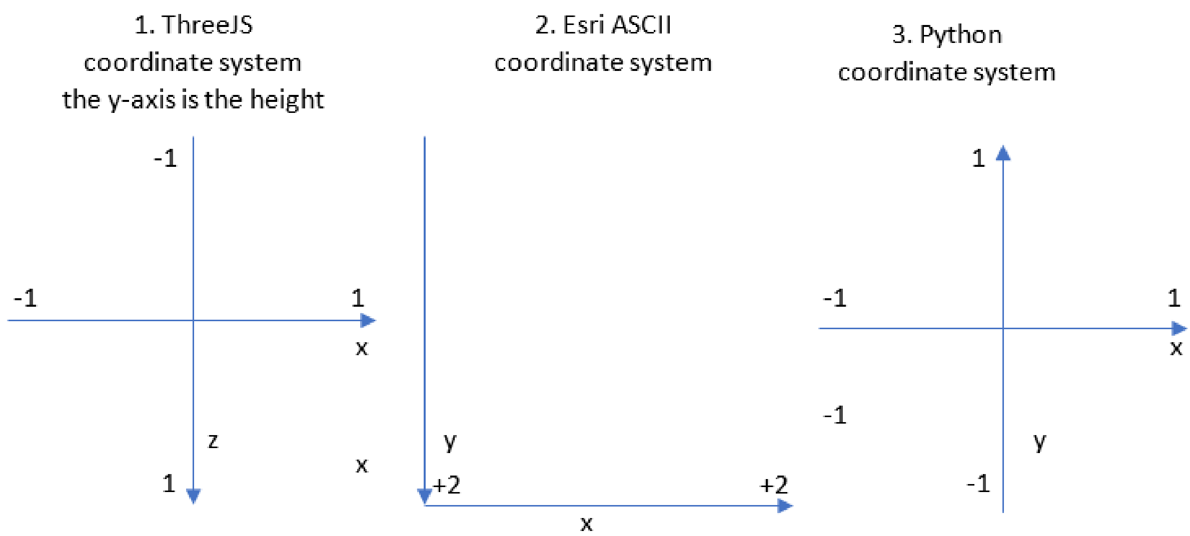


Figure 0-5, Different coordinate systems

B.4 Align calculation cross-sections

When the align button is clicked, the front-end sends an AJAX request to the server. The server intersects all the calculation cross-sections with the to be design object as described in Appendix B.7. From this intersected point, a Python script find the centre point of the to be object and the perpendicular angle to the object (Figure 0-6). These are send back to the front-end.

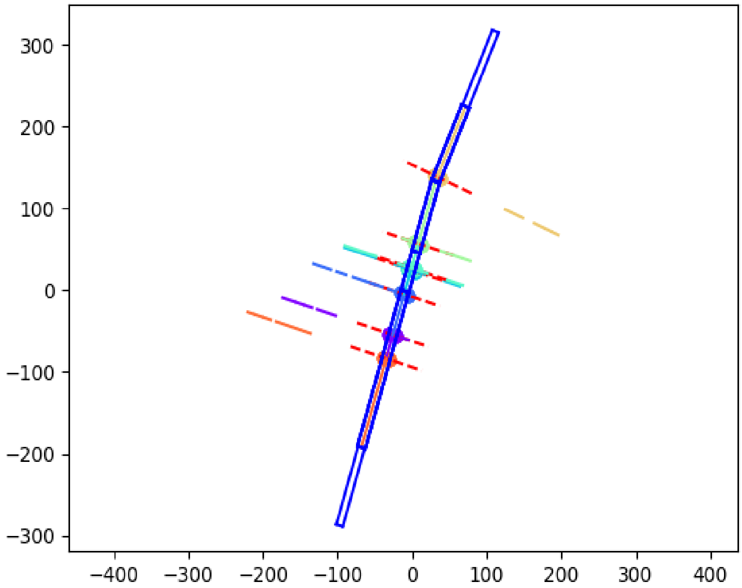


Figure 0-6 Centre point of the to be constructed object

B.5 Ground layer data extraction

To collect the ground layer data, information extracted from a ESRI file. Each time a cross-section is moved or added the script is updated. The following steps are coded in the design tool:

1. For or each calculation cross-section, shown in Figure 0-7, the polar coordinates are read from JavaScript and written to Python.

(Note!! the red arrows on each calculation plane indicating the plane direction, planes 3 & 4 are in the same location but each face a different direction).

2. These polar coordinates of each calculation cross-sections are converted to the ESRI coordinate system, shown in Figure 0-9

3. 100 points are generated between the begin and end of each calculation cross-section represented by lines in Figure 0-9

X = linspace(x0, x1, 100)
Y = linspace(y0, y1, 100)

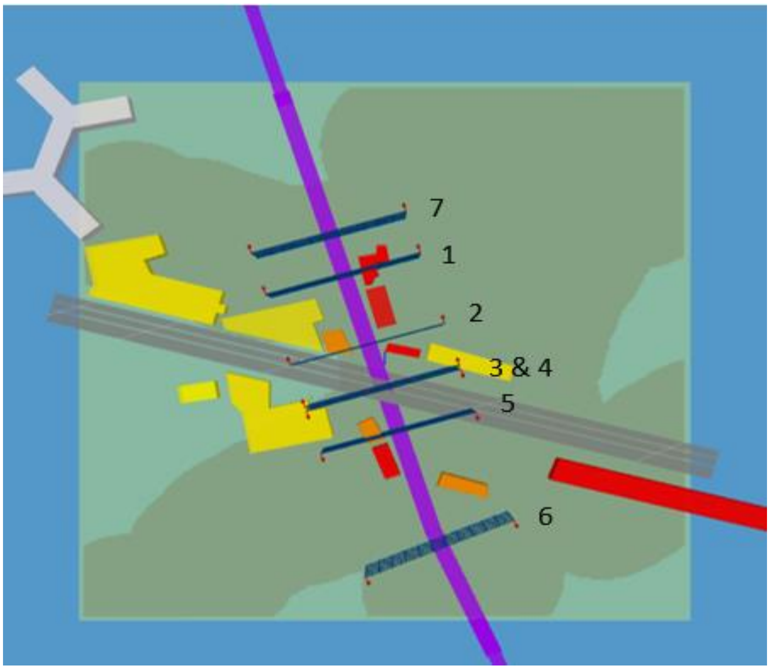


Figure 0-7 Calculation cross-sections in the Three.js coordinate system

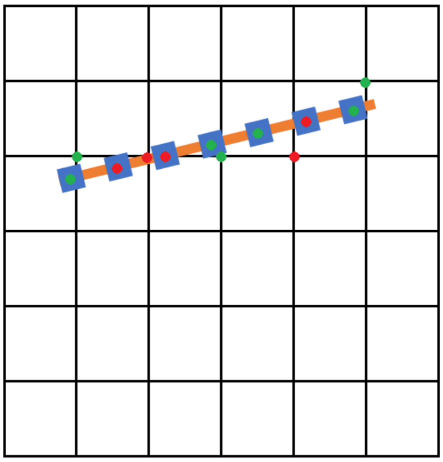


Figure 0-8 Nearest point interpolation

The data in an ESRI file is based on a grid. Each point on that grid contains a depth (Z) value. For the interpolation that is required to extract the corresponding Z values of this grid to that of the line, the grid point nearest to one of the 100 points are used as Z value for that point (Figure 0-8. The results of this interpolation process is shown in Figure 0-10. Afterwards the x and y coordinates are converted from the ESRI coordinate system to the Three.js coordinate system. Figure 0-10 shows the results from this interpolation process. Below a part of cross-section 2, no clay layer is present shown in Figure 0-9. In the results in Figure 0-10 this can be observed from the 100 points underneath the plot.

1. A “Wavefront .obj” file is created from these from the lines and from the ESRI ground data, this way the can be displayed in the 3D environment. This is discussed in Appendix B.9. The intersected ground data is also stored in Python internally, this way this data can be used in the following steps.

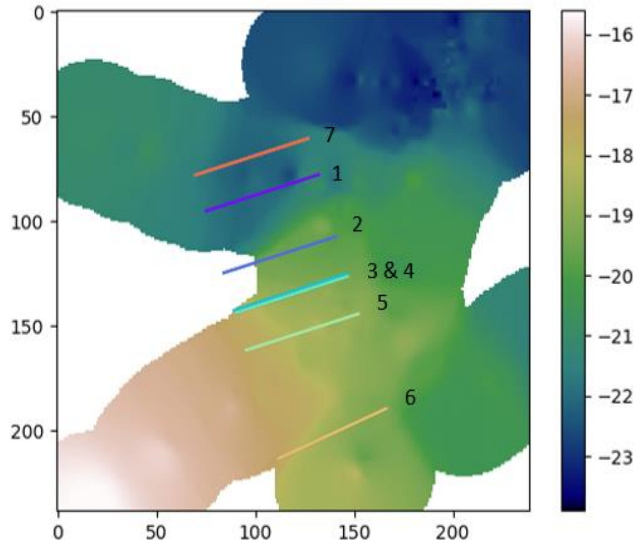


Figure 0-9 The ESRI ground data (clay layer) and the calculation cross-sections in the ESRI coordinate system (Python Plot)

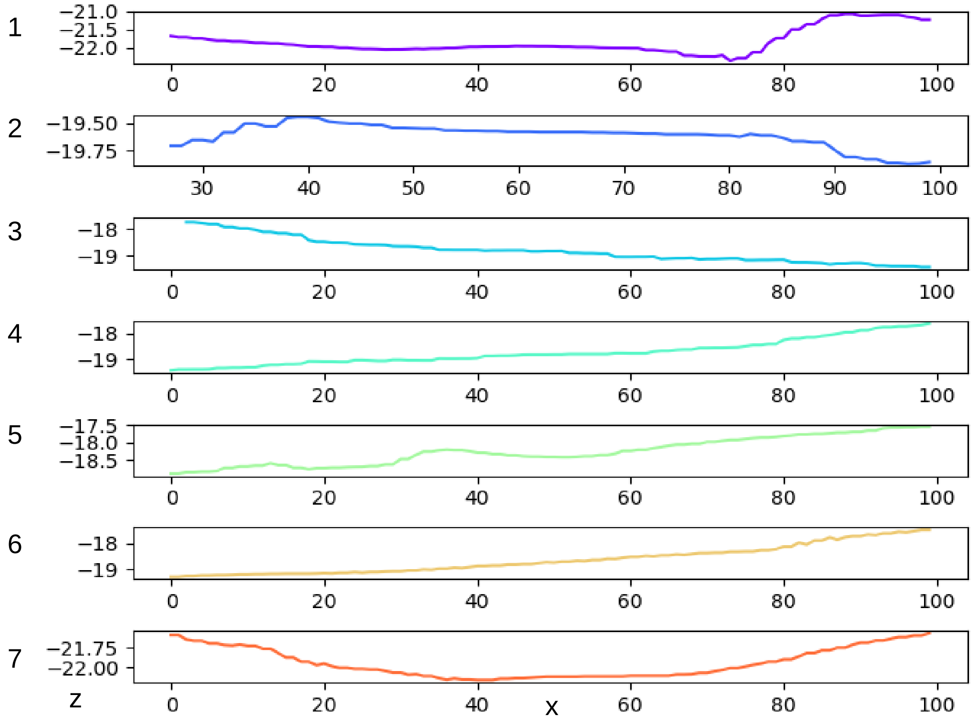


Figure 0-10 Extracted depth values for each calculation cross-section, the x value corresponds to a point on the line which has a x and y value in 3D. (Python Plot)

B.6 Data extraction from the surrounding buildings

The properties of each building are used by the design tool in several ways.

1. For a calculation with anchors, the design tool checks if there is a building above an anchor.
2. For each Plaxis calculation the loads and the coordinates are extract.
3. After the settlement results from Plaxis are generated the extracted data is used to check the horizontal strain and relative rotation.

The data from these buildings is extracted in the following way:

1. All buildings are, one by one, read into Python.
2. The lines of the edges that make up the building are determined
3. For each line the intersection with the calculation cross-section is found. This can be seen in Figure 0-12. The longer lines are the exact size of the calculation cross-sections.
4. The building properties, as discussed above, are extracted from the intersected buildings.
5. The coordinates of intersected points are normalized to that of the calculation cross-sections. This can be seen in Figure 0-13. As you can see cross-sections 3 and 4 (2 and 3 in Python) are the same but mirrored because the face the opposite direction.

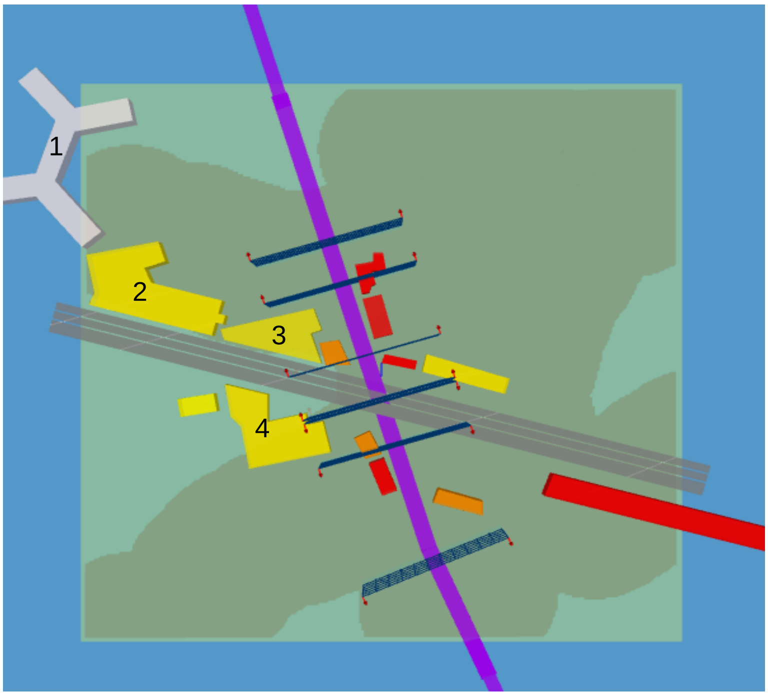


Figure 0-11 Calculation cross-sections in the Three.js coordinate system

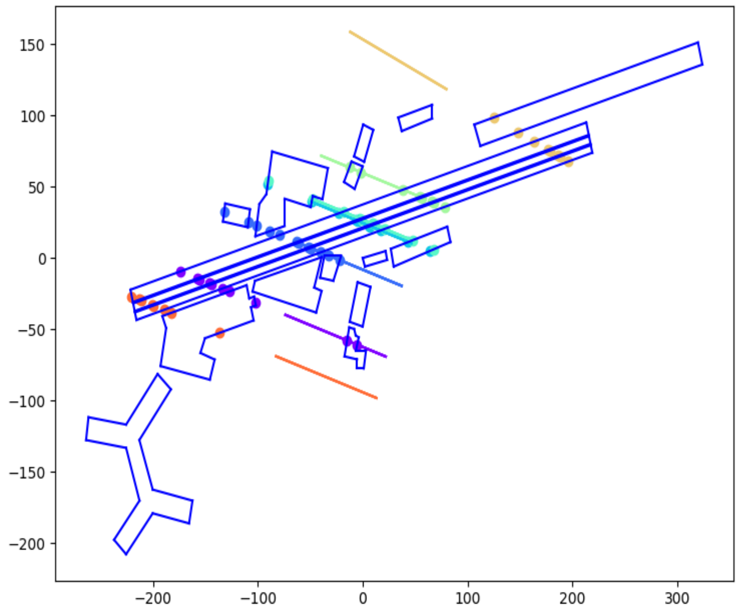


Figure 0-12 Intersection point of the buildings on the calculation cross-sections. Each line represents a calculation cross-section. Each dot an intersected point on an edge of a building. Python coordinate system.

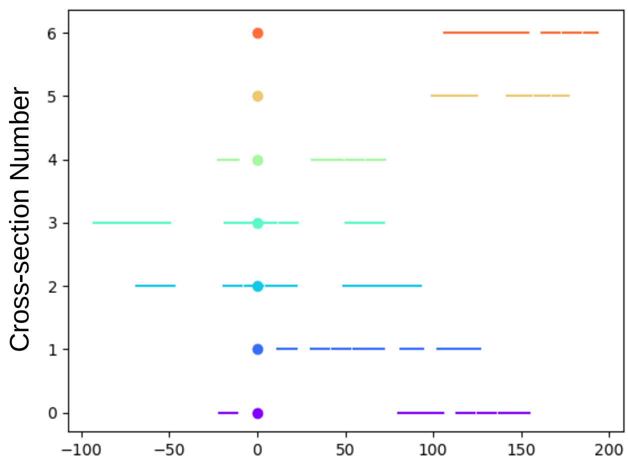


Figure 0-13. The intersected buildings, each line corresponds to a building. The dot corresponds to the origin of the calculation cross0sections (Python Plot)

B.7 Data extraction from the to be constructed structure

1. The object is read into Python.
2. The lines of the edges that make up object are determined
3. For each line the intersection with the calculation cross-section is found. This can be seen in Figure 0-16.
4. The coordinates of intersected points are found
5. The corresponding pit depth point is found (Figure 0-15) along with the floor slope
6. The length of the excavation pit is determined. This is the distance between two calculations cross-sections or from a calculation cross-section to the surfacing point of the to be build object.

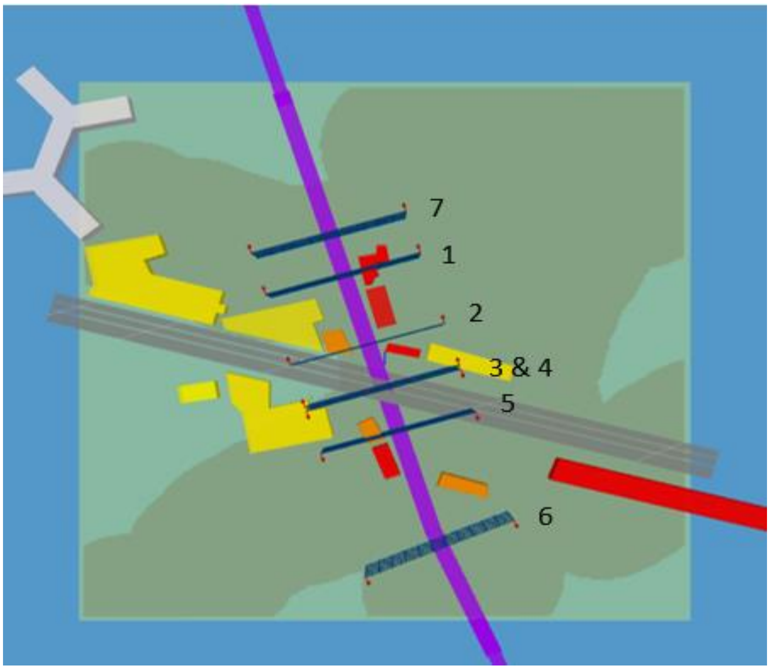


Figure 0-14 Calculation cross-sections in the Three.js coordinate system

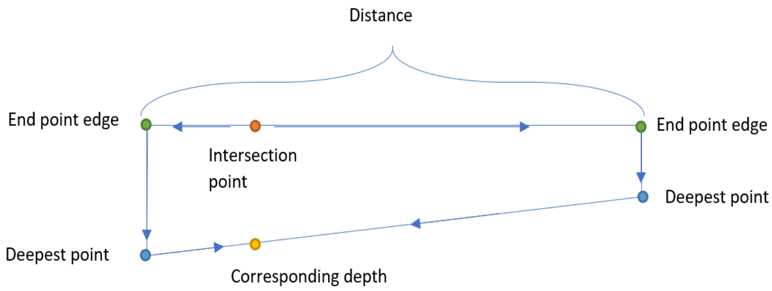


Figure 0-15, Illustration of the depth finding process, longitudinal cross-section of a

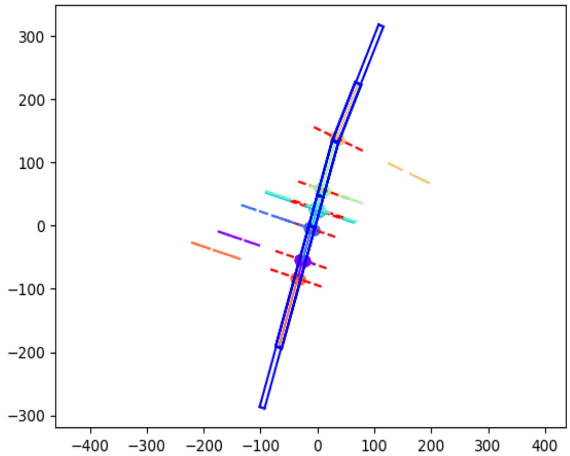


Figure 0-16 Aligned calculation cross-sections (Python Plot)

B.8 Pre-processing design algorithm

1. When the align button is pressed, the calculation cross-sections are centred and are positioned perpendicular to the to be build object.

2. Data extraction
    a. The polar coordinates of a calculation cross-section are extracted from the 3D environment.

    b. The corresponding data from the ground layers is extracted as described in Appendix B.5

    From each ground layer the depth lines (Figure 0-10) are used by the design tool

    c. The corresponding data from the surrounding buildings is extracted as described in Appendix B.6

    From this extracted data the following values are used by the design tool
        i. The distance of each building to the retaining walls
        ii. The load of the building
        iii. The maximum allowed horizontal stress
        iv. The maximum allowed relative rotation

    d. The corresponding data from to be constructed structure is extracted as described in Appendix B.7
    From this extracted data the following values are used by the design tool:
        i. The x and y coordinates of the outer edges where the retaining walls need to be constructed. From these coordinates the width of the excavation pit is derived.
        ii. The depth of the construction pit
        iii. The angle of the floor
        iv. The distance between two calculation cross-sections is determined. This is the length of the excavation pit for that calculation.

3. A range for the wall depths is defined
    a. A minimum wall depth is defined
        i. Minimum wall depth = excavation depth – user input
    b. A maximum wall depth is defined, the lowest value of the following three depths is used by the design tool:
        i. Maximum wall depth 1 = excavation depth – user input
        ii. Maximum wall depth 2 = excavation depth * user input
        iii. If (impermeable layer is present)
            1. Maximum wall depth 3 = impermeable layer dept – user input
    c. Maximum depth retaining wall is lowest value of the three maxima stated above.
    d. The maximum is adjusted when it is deeper as the maximum length of the retaining wall.
4. Possibility of wet and dry excavations are determined on the basis of the ground depth lines, the depth lines underneath the excavation pit must show a solid impermeable layer. This layer must extend outside of the excavation pit as well.
    a. When a no impermeable layer is present: wet excavations
        i. for wet excavations two additional impermeable layers are taken into consideration. The height of this layer is defined by the user. The depth of this layer is based on the horizontal equilibrium. Additionally, the wall lengths are checked if the extend deep enough into this layer.
        ii.
    b. When an impermeable layer is present: wet and dry calculations
        i. Dry calculations when the retaining walls extend into the impermeable layer and a vertical equilibrium is reached
        ii. Wet calculations if the walls do not extend into the impermeable layer. For these calculations an impermeable layer is created by an UCF or gel injections.

5. Supports
   a. first it is check if anchors can be applied, this depends whether or not an anchor is positioned underneath a building.
   b. The number of rows and their installation depth is determined for each support
      i. No supports when pit depth > user input
      ii. On row if pit depth < user input
      iii. Support depth 1 row = user input
      iv. Excavation level underneath support 1= Support depth 1 row – user input
      v. Two rows if pit depth < user input
      vi. Support depth 2 rows = (Support depth 1 – pit depth) / X
      vii. Excavation dept 2 rows = Support depth 2 – X
   c. The number of supports is determined
      i. (Pit length – support spacing)/ support spacing * number of rows
6. Based on the volume or number of structural elements, the costs, MKI and mobilisation costs are calculated.
7. For each calculation cross-section the design algorithm creates a calculation order. This order is based on the costs of each excavation pit. This is done to be able to assess Plaxis results before every calculation has been processed. To increase the change that data is available for each cross-section, the calculation order starts with the cheapest calculations of each cross-section. Then the second cheapest calculations of each cross-section and so on.
8. All required files are stored and can manually be transferred to the parallel Plaxis computer.

B.9 Data conversion

B.9.1 Wavefront .obj

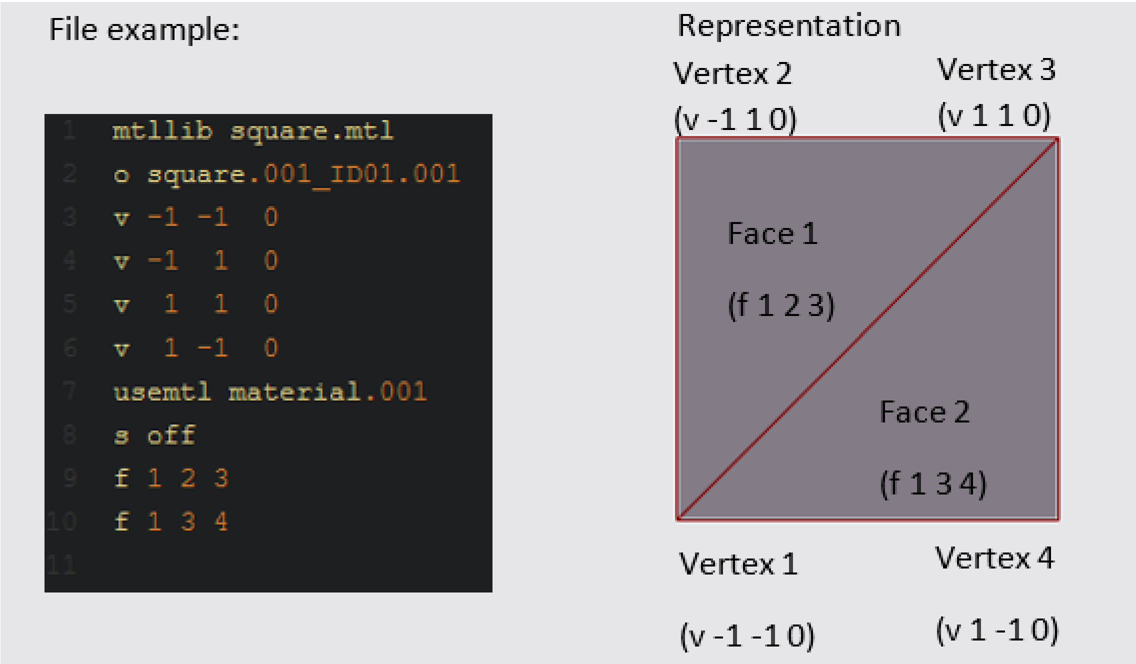
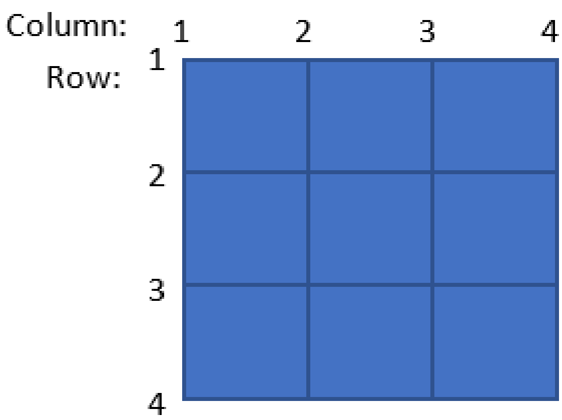


Figure 0-17 On the left: Example of "Wavefront .obj" file; on the right: digital representation

B.9.2 ESRI to Wavefront .obj

The conversion from ESRI to "Wavefront .obj" start with reading the ESRI file. The ESRI file contains three different sets of data.

1. (m,n) square grid, every point on this grid contains a depth (z) value.
2. a cell size, by multiplying this cell size with m,n, this will give a relative x and y coordinate
3. Origin data, the longitudinal and latitudinal coordinates of one of the corners to the corresponding coordinates in the world.



To convert this ESRI ACSII file to a "Wavefront .obj" file. First the squares are replaced with triangles. This is done by a for loop, a graphical representation is show in Figure 0-19.

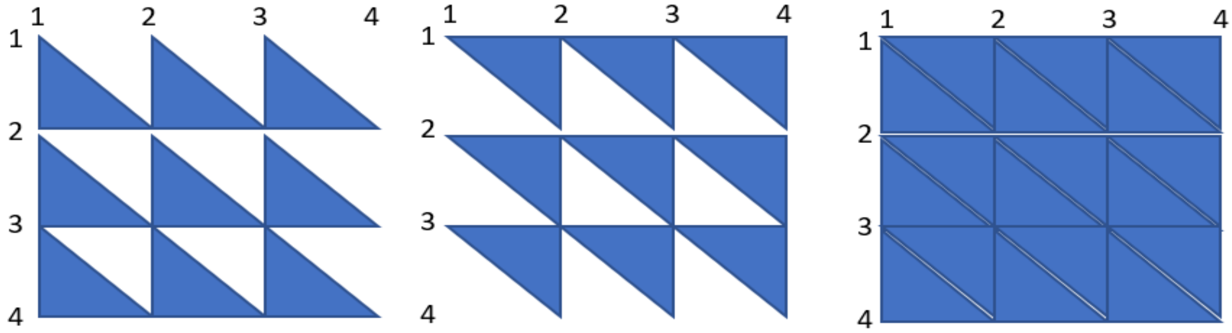


Figure 0-19 From square grid to triangulated mesh

After the mesh is created, the data is written in the "Wavefront .obj" file format. To do this, all points (vertices) are given a number starting with 1. Vertices 1, 4, 5 make up face A in Figure 0-18. All vertices and their corresponding coordinates x,y,z are stored. Then for each triangle (face) the corner points are stored in the "Wavefront .obj" file (Appendix B.9.1).

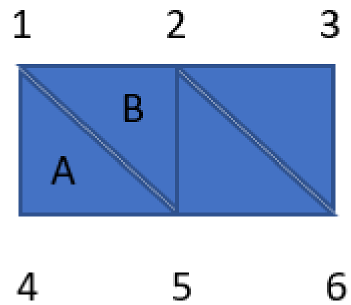


Figure 0-18 Mesh triangles

C
Plaxis

C.1 Error in the Plaxis integration

Almost all calculations with a wet excavation of the Oosterengweg test case (Chapter 3.1.2) failed. The same calculations failed after alterations were made to the scripts, after the green light meeting. The exact cause of these failures was not found. In Figure 0-20, the results of one of the calculations that failed during the first draining phase, is shown. In the settings of Plaxis the UCF was added, however in this figure it is clearly visible that no horizontal forces are acting on the walls. There is no pivot point of the retaining walls visible at this location. Many other calculations did already fail at the phases where only the retaining walls are installed. This could be due to some issue with the script that uploads the geometry, however the exact cause was not found.

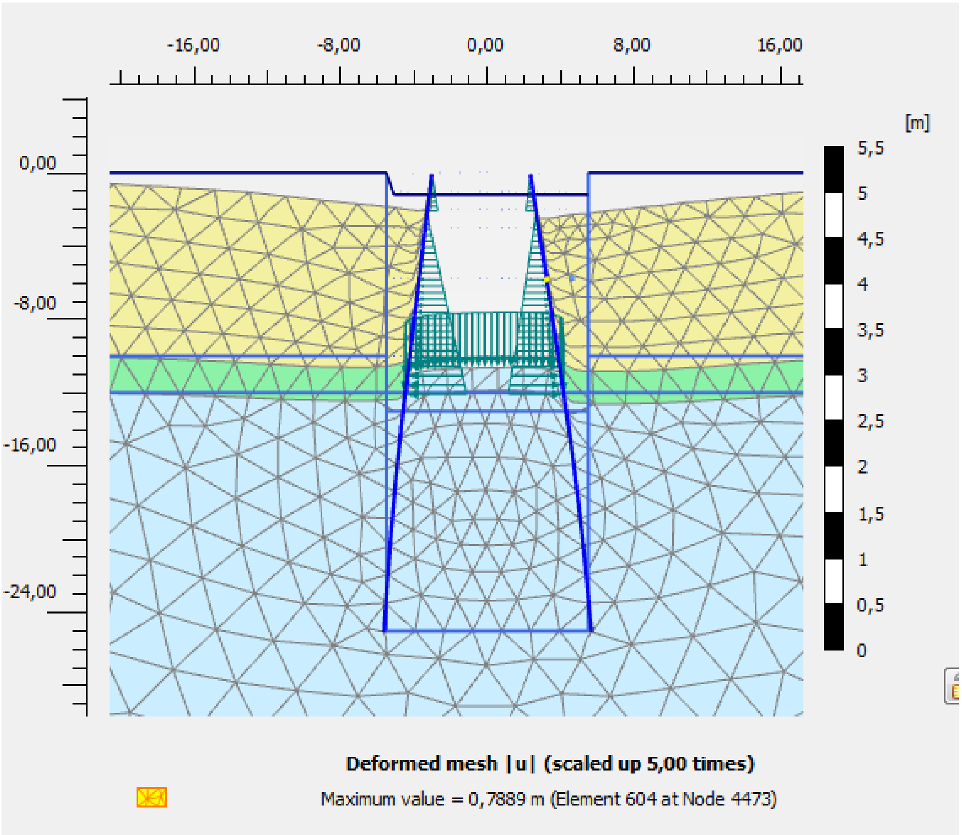


Figure 0-20 Example of failed calculation

D
References

References

Aish, R., & Hanna, S. (2017). Comparative evaluation of parametric design systems for teaching design computation. *Design Studies, 52*, 144-172.

Allen, J. B. (2016). The Future of BIM Will Not Be BIM—and It's Coming Faster than You Think. Retrieved from https://www.youtube.com/watch?v=xq6yKyauu-o&t=1333s

Azhar, S. (2011). Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. *Leadership and Management in Engineering, 11*(3), 241-252.

Babich, N. (2017). Rapid Prototyping: The Most Efficient Way To Communicate Your Ideas. Retrieved from https://theblog.adobe.com/rapid-prototyping-efficient-way-communicate-ideas/

Bew, M., & Richards, M. (2008). BIM Maturity Diagram. In.

Bhooshan, S. (2017). Parametric design thinking: A case-study of practice-embedded architectural research. *Design Studies, 52*(1), 115-143.

Brenner, A. E. (1997). Moore's Law. *Science., 275*(5306), 1549.

Chi, H.-L., Wang, X., & Jiao, Y. (2014). BIM-Enabled Structural Design: Impacts and Future Developments in Structural Modelling, Analysis and Optimisation Processes. *Archives of Computational Methods in Engineering : State of the Art Reviews, 22*(1), 135-151.

COINS. Information exchange in the life cycle. In. http://www.coinsweb.nl/wiki2/index.php/Welcome_to_the_COINS_wiki_pages.

Consulting, C. (2001-2018a). V-Model Development. Retrieved from https://www.coleyconsulting.co.uk/testtype.htm

Consulting, C. (2001-2018b). Waterfall Model. Retrieved from https://www.coleyconsulting.co.uk/waterfall-model.htm

Eadie, R., Browne, M., Odeyinka, H., McKeown, C., & McNiff, S. (2013). BIM implementation throughout the UK construction project lifecycle: An analysis. *Automation in Construction, 36*, 145-151.

Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). *BIM handbook : a guide to building information modeling for owners, managers, designers, engineers and contractors* (Second edition. ed.). Hoboken: John Wiley & Sons.

Fyrd. (2018). WebGL - 3D Canvas graphics. Retrieved from https://caniuse.com/#feat=webgl

GeoJSON. (2018). GeoJSON. Retrieved from http://geojson.io/#map=2/20.0/0.0

Guo, X., & Cheng, G.-D. (2010). Recent development in structural design and optimization. *Acta Mechanica Sinica, 26*(6), 807-823.

Harding, J. E., & Shepherd, P. (2017). Meta-Parametric Design. *Design Studies, 52*, 73-95.

Het Nationaal BIM Platform. (2010). Levels of Detail. Retrieved from https://hetnationaalbimplatform.nl/levels-of-detail.php

Horst, v. d. A. Q. C. (2014). *CT4170 Construction Techonolgy of Civil Engineering Projects* TU Delft.

IAI Model Support Group. (1996-2007). Industry Foundation Classes IFC2x Edition 3 Technical Corrigendum 1. Retrieved from http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm

IfcOpenShell. (2017). IfcOpenShell. Retrieved from http://ifcopenshell.org/python.html

Linietsky, J., & Manzur, A. (2014). 3D ray casting from screen. Retrieved from http://docs.godotengine.org/en/3.0/tutorials/physics/ray-casting.html

Liu, X., Xie, N., Tang, K., & Jia, J. (2016). Lightweighting for Web3D visualization of large-scale BIM scenes in real-time. *Graphical Models, 88*, 40-56.

Mileham, A. R., Currie, G. C., Miles, A. W., & Bradford, D. T. (2007). A Parametric Approach to Cost Estimating at the Conceptual Stage of Design. *Journal of Engineering Design, 4*(2), 117-125.

Mozilla Foundation. (2014, May 20, 2014). Asynchronous JavaScript + XML. Retrieved from https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX

Mozilla Foundation. (2018, Jul 3, 2018). Getting started with WebGL. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL

Ooms, B. (2017). BAM Infra eerste bouwbedrijf ter wereld BIM Level 2 gecertificeerd. Retrieved from https://www.bam.com/nl/pers/persberichten/2017/11/bam-infra-eerste-bouwbedrijf-ter-wereld-bim-level-2-gecertificeerd

Phillips, S. (2010). Parametric Design: a Brief History
ARCCA, 10(1), 24-30.

Rijkswaterstaat. (2012). DuboCalc. Retrieved from https://www.rijkswaterstaat.nl/zakelijk/zakendoen-met-rijkswaterstaat/inkoopbeleid/duurzaam-inkopen/duurzaamheid-bij-contracten-en-aanbestedingen/dubocalc/index.aspx

Rijkswaterstaat. (2018). Selectie- en gunningscriteria. Retrieved from https://www.rijkswaterstaat.nl/zakelijk/zakendoen-met-rijkswaterstaat/inkoopbeleid/aanbesteden/selectie-en-gunningscriteria.aspx

Run Digital. (2007). RD Glossary Retrieved from https://web.archive.org/web/20070826224349/http://www.ittc.ku.edu/hybridthreads/glossary/index.php

Sacks, R., & Barak, R. (2008). Impact of three-dimensional parametric modeling of buildings on productivity in structural engineering practice. *Automation in Construction, 17*(4), 439-449.

Singh, V., Gu, N., & Wang, X. (2011). A theoretical framework of a BIM-based multi-disciplinary collaboration platform. *Automation in Construction, 20*(2), 134-144.

Tol, A. F. v. (2003). *CT5330 Foundation Engineering and Underground Construction* Delft University Of Technology.

Vermeij, P. (2006). *Parametric associative design for free form architecture.* (Master), TU Delft, WorldCat.org database.

Vries, J. d. (2018). Learn OpenGL. Retrieved from https://learnopengl.com/

Software

Cabello, R. (2018). Three.js (Version r90). Retrieved from https://threejs.org/

Django Software Foundation. (2018). django, the web framework for perfectionists with deadlines (Version 2.0.4). Retrieved from https://www.djangoproject.com/

Khronos Group. (2011). WebGL (Version 2.0): the kronos group. Retrieved from https://www.khronos.org/webgl/
PLAXIS. (2017). Plaxis 2D (Version 2017): PLAXIS. Retrieved from https://www.plaxis.com
Plotly. (2018). Plotly. Retrieved from https://plot.ly/
RockWare. (2018). RockWorks Retrieved from https://www.rockware.com/product/rockworks/
Ronacher, A. (2018). Jinja2 (Version 2.10). Retrieved from http://jinja.pocoo.org/docs/2.10/