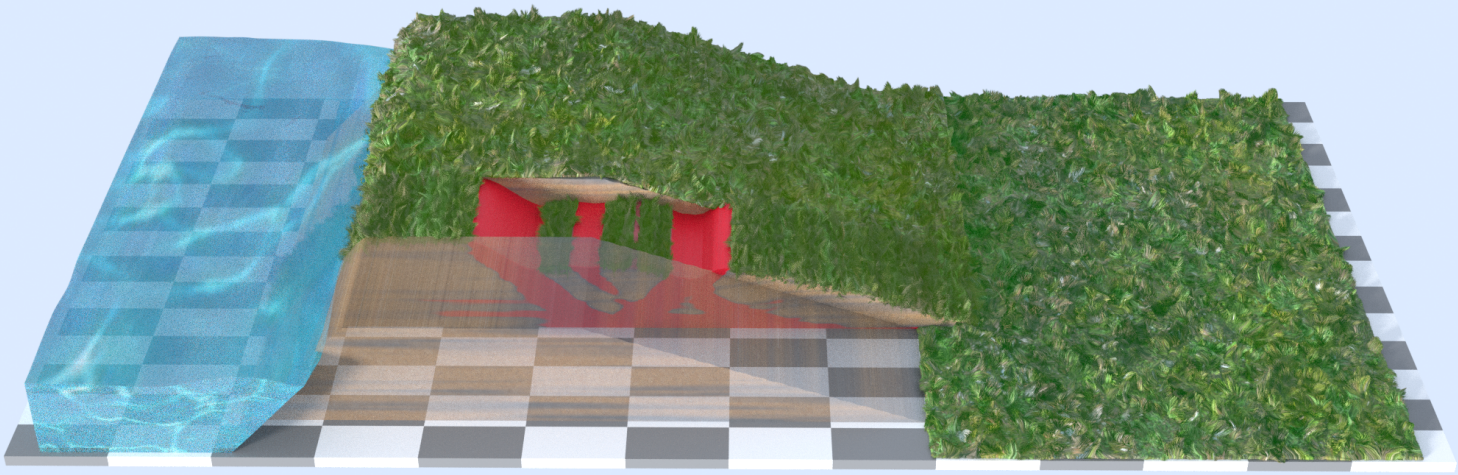
Visualization of the Random Material Point Method for slope stability

MSc Thesis

C.K.L. Man



TU Delft

Visualization of the Random Material Point Method for slope stability

MSc Thesis

by

C.K.L. Man

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4385470	
Project duration:	April, 2020 - September, 2021	
Thesis committee:	Dr. ir. P.J. Vardon,	TU Delft (chair)
	Ir. G. Remmerswaal,	TU Delft
	Dr. K. Hildebrandt,	TU Delft
	Dr. M. Billeter,	University of Leeds
	Ir. P.R.M. Ammerlaan,	Boskalis

TU Delft

Preface

In front of you lies the thesis “Visualization of the Random Material Point Method for slope stability,” in which it tries to find a method to represent a numerical dataset in appealing, which is used for visual communication to different audiences. The thesis was written as the final assessment to obtain the degree of Master of Science at the Delft University of Technology at the faculty Civil Engineering with the track geo-engineering.

Before I started this thesis, this research proposal immediately caught my eye because I saw that it involves something with Disney movies and graphics (which is later explained during the thesis). It combined the things that I like, which are Civil Engineering and being creative. Creativity comes in handy many times during this research, from making and designing the visuals to thinking how to this research, to coding, and I have enjoyed it till the very last bit.

I have learned a lot from the thesis. I have gained much knowledge about computer graphics and the WBI, and how current practice goes. Furthermore, I have developed soft skills: writing such a big story in a non-chaotic manner and managing a large project. I started the research on the first day that COVID-19 has arrived in the Netherlands, so self-discipline is needed more than ever. Luckily, I have found the Pomodoro Technique, which helped me to focus. Although the pandemic made it difficult to meet people physically, I did not felt lonely, and there was support around me.

I am very grateful to have a supportive committee that showed me how much fun research could be. I want to thank Phil Vardon and Klaus Hildebrandt for pointing me in the right direction when needed. I would also like to thank Markus Billeter for your time, patience and enthusiasm to introduce me to the computer graphics world. And to Patricia, for your insight into the engineering practice at Boskalis and your enthusiasm. Last but certainly not least, I would like to thank my daily supervisor, Guido Remmerswaal, for the endless support, patience, kind words and enthusiasm. From the very start to the end of the thesis, Guido always made time as soon as he can no matter how busy he was, which was appreciated a lot.

Finally, I would like to thank family and friends who have helped me during this long journey, especially my mum, who is always by my side and encouraging. Without you all, I am not able to accomplish something that I never thought I could do in the first place.

C.K.L. Man
Delft, September 2021

"Do you want to build a snowman? It doesn't have to be a snowman...
It also could be a thesis." - Adapted from Frozen I, Walt Disney

Summary

To protect the Netherlands better from flooding, and with an eye on sea-level rise in the rest of the world, more accurate assessments are needed for dykes. The calculation for the most occurring failure mechanism in dykes, i.e. macro-instability, is limited by not being able to calculate large deformation when sliding occurs within a dyke. The random material point method (RMPM) is able to capture the complete failure path, including the residual dyke strength, while taking the heterogeneity of the soil into account, thereby improving the assessment of failure processes. However, as the method is not (yet) used in current practice, clear communication of the results is essential for convincing a wider public of the contribution of this method. Visuals are an important tool in communication, as it increases comprehension of the subject matter if the visual is designed efficiently. This research aims to investigate the available software and which techniques are suitable to make realistic and informative visualizations for a given RMPM dataset of slope failure problems.

A method to create visualization with a certain graphical realism in three-dimensional space is developed. The technique uses a computer graphic software (Blender) combined with an add-on, i.e. an extension plugin. The add-on allows to work with VTK software to process scientific data for visualization, thereby maintaining the scientific correctness of the visualizations. Moreover, a rendering pipeline in Blender is created, which transforms the properties from scientific colors into realistic materials, making the visualizations more intuitive.

However, the dataset is too large to summarize in a straightforward illustration. Therefore, a data analysis is obtained to classify each realization into five pre-defined failure profiles, which are determined based on a literature study. Four failure profiles are classified based on the number of retrogressive failures and whether or not the realization resulted in flooding, while the fifth class describes horizontal failures. A technique has been developed to separate the horizontal failures from the other classes based on the plastic deviatoric strain attribute. Additionally, the data analysis aims to characterize the behavior of each failure profile from an early start, such that the findings could be used for current methods, which could not calculate the full failure profile.

Therefore, this thesis needs to investigate the reduction of the dataset to make it more time efficient when doing a data analysis. It is extended on the clustering algorithm, which has the function to detect failure blocks based on the displacement per dyke profile. The reduction method replaces an amount of data by one representative point per cluster. It not only reduced the size of the dataset significantly, from 3000 GB to 6 GB, it also made the comparison of attributes between realizations, and therefore the data analysis, easier.

The data analysis shows that it is hard to distinguish different failure profiles using only data of the initial failure, which shows the importance of using RMPM to account for post-failure behavior instead of using the current assessment i.e. FEM and LEM. One finding is that equilibrium of the initial failure block is often reached before a vertical crest displacement equal to 0.5 times the height of the dyke. This indicates that the crude estimation in the current assessment is highly conservative. Moreover, within the assumption, it is hypothesized that the secondary failure block will only form after the initial failure block has reached its equilibrium, which is shown otherwise within the data analysis of this thesis.

This work proposes a method for data analysis of RMPM using parallel coordinates, which can be extended to other RMPM datasets for macro-instability and can help to improve the prediction of the probability of flooding. Moreover, it proposes a method to visualize the prominent features, determined using parallel coordinates, in Blender-VTK. This work can, in future research, be extended to other geotechnical problems, such as 3-dimensional dyke slope failure.

Contents

1

Introduction

The Netherlands is known for its flood defense system, required to protect half the population living in flood prone areas. All Dutch primary flood defenses have to meet the requirements of the Water Act (Waterwet) [70]. Using assessment guidelines and tools provided by the Ministry of Infrastructure and Environment in 2017 (WBI) [33], the primary flood defenses are periodically tested by the regional water authorities against the standards from the Water Act. The results from the most recent assessment rounds in 2017 show that before 2050, over a length of 1300 km dykes have to be reinforced [33].

Due to this extremely large operation, the government has assembled a group of experts from different regional water authorities, research groups and industries to develop a reinforcement plan for the primary and secondary flood defenses. This project is called the National flood protection program ('Hoogwaterbeschermingsprogramma' (HWBP) in Dutch) [36]. Additionally, the government invested a budget of 7.9 billion euros to strengthen the dykes in the coming ten years [2]. The challenge of this operation arises from the size of the project and the environment around the dyke. Nowadays dykes often have multiple functions besides only retaining water. For example, dykes are used for recreational purposes, and dyke designs are becoming more nature-inspired, which increases the complexity of the design of the dyke compared to dykes with only a retaining function.

Because of the additional features, it is more attractive to live closer to dykes than it used to be and leads to a denser population around the dyke [36]. The complexity of the dyke design therefore increases even further and makes the strengthening of the dyke complicated.

1.1. RMPM for the remaining resistance after dyke failure

In order to optimize dyke strengthening, geo-engineers would like to assess the strength of a dyke more accurately [1]. By optimizing the assessment, dyke strengthening can be carried out only where it is actually necessary, and the weakest links can be strengthened first.Moreover, when strengthening is required, a less impactful and thus a less conservative solution can be applied when designs are optimized. Therefore, the failure mechanisms of dykes are (re)evaluated.

According to the Dutch assessment guidelines, the most occurring failure mechanism in dykes is macro-instability [33]. Macro-instability, i.e. inner slope failure, happens during a high water event, where water can infiltrate the dyke body due to increased external water pressure. The infiltrated water increases the material weight and lowers the strength of the material, causing a roughly circular mass to slide off at the inner slope of the dyke. A large slide may lower the height of the dyke below the water table, triggering flooding. However, the occurrence of a first slide may not always lead to flooding. The remaining parts of the dyke ('rest profiel' in Dutch or 'residual profile' in English) can still have a certain height and strength, which may retain water. In other words, the remaining resistance after initial failure may still prevent flooding [11].

In current practice, geo-engineers use the Limit Equilibrium Method (LEM) or the Finite Element

Method (FEM) to estimate the likelihood of macro-instability by the normative, i.e. most likely, failure surface. The remaining resistance is often ignored or accounted for by a rough conservative assumption [11, 59]. Improving the estimation of remaining resistance can optimize the assessment of macro-instability, potentially reducing the required strengthening. However, the existing methods are limited to modeling the onset of failure since they cannot model the large deformations occurring during failure. Therefore, they cannot be used to predict the process after the onset of failure, making estimations of remaining resistance difficult with the standard methods for dyke assessment. The material point method (MPM) can handle large deformations, and has been shown to model the evolution of retrogressive slope failures in dykes accurately [65], and can be used to include the effects of the remaining resistance on the probability of flooding [52].

The failure processes of a dyke take place in a highly spatially varying soil structure, a variability that is often not present in other materials used in engineering. Spatial variability greatly influences inner slope stability since failure is attracted to the weak zones within the material. In the Random Finite Element Method (RFEM), random fields have been coupled with FEM in a Monte-Carlo framework in order to account for the spatial variability [27–29]. The method has been shown to accurately predict the probability of initial failure by running a set of FEM realizations, each evaluating one of the possible configurations of the spatial variability. The failure process after the initial failure is also significantly affected by the spatial variability [61]. Recently, MPM has been extended with random fields, creating the Random Material Point Method (RMPM) to account for the effect of spatial variability on the entire failure process [51, 65]. RMPM is an effective tool for predicting remaining resistance after the initial failure of a dyke, and can predict the probability of flooding instead of the probability of initial failure.

If remaining resistance is accounted for in the assessment, the assessment becomes more and more complex. Although conveying these complex assessments in a concise manner may be difficult, the geo-engineer is still obligated to communicate the information to other geo-engineers and the stakeholders affected by the dyke assessment. These stakeholders, for example surrounding residents or nature associations, have become more involved in the decision process for dykes and their surroundings in recent years. Therefore, the communication should be efficient for both geo-engineers, who understand the technical background behind dyke assessment, and other stakeholders, who may lack a (geo)technical background. The question is raised: *How do geo-engineers explain complicated remaining resistance assessments efficiently to each other and the general public?*

1.2. Power of visualization

We live in a world where everything has to be efficient and accurate. Ideally, we like to convey the information of dyke assessment in a short amount of time. Luckily, humans are very good at extracting information through visual observation, like the old saying: “a picture is worth a thousand words”. The notion of the term *visualization* in scientific research was introduced by McCormick et al. (1987) [41] and defined as follows:

> *Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights [41].*

Good visual representation should guide the receiver through the problem, rather than overloading them with technical details [5]. Therefore, using graphical realism to represent RMPM results is an effective and appealing method to explain dyke failure to the general public lacking a geotechnical background [55]. Moreover, these visuals can have additional value when used for marketing purposes and may broaden the application of (R)MPM in geotechnics to the game- and film industry. Visuals with a realistic perception are also called *artistic visualizations* within the thesis.

Additionally, a good visual representation of data should distinctly display multiple (complicated) variables, helping engineers and scientists understand, analyze, and communicate key concepts of their findings in research. The term *scientific visualization* refers within this thesis to data visualization in graphical form, which has the aim to aid in seeing and understanding an otherwise abstract set of

numbers[5]. In Figure 1.1 the difference between artistic and scientific visualization are shown, both visuals use the MPM method to visualize, although there is a difference in the target audience.

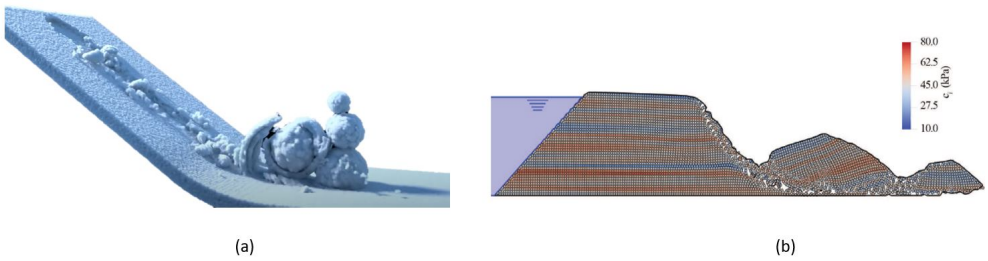


Figure 1.1: (a) Rolling snowball showing the packing effect of snow using a MPM simulation[56]; an example illustration of artistic visualization. (b) Slope failure of a dyke simulated using MPM [50]; an example illustration of scientific visualization

1.3. Challenge of visualization of RMPM for slope stability problems

Implementation of MPM in computer graphics was claimed to be first used in 2013 by Stomakhin et al. [56] to simulate and realistically model snow behavior for an animation movie. Later, several studies created impressive animations with the aid of MPM for different types of materials such as frictional contact in cloths, fracturing behavior in solids, and thermomechanical behavior in food [16, 67, 72]. Although the MPM simulation themselves are often described in detail, the rendering methods for the visualization are described shortly in comparison. For example, Umenhoffer [60] has re-implemented in 2015 the rendering pipeline of Stomakhin et al. [56], but again does not go into details of the individual rendering components. Since studies often do not go into the details of the visualization pipelines, reproducing the used techniques is difficult. Therefore, the components of the pipelines should be investigated themselves.

Computer graphics research shows that it is possible to make artistic visuals from MPM data. However, they prioritize aesthetics over scientific accuracy and may change numerical behavior to improve the aesthetics, which is not an option when attempting to convey accurate information with improved visuals [56, 72]. The term scientific accuracy refers to the meaning of being able to provide correct reading within an illustration, i.e. showing the layering of soils with different strengths. Moreover, none of these simulations are applied in the geotechnical field. Andersen et al. [7] is one of the first with the focus on visualizing and post-processing of MPM data for soils. It addresses the importance and difficulties of attaching physical attributes to the points so that it is possible to query every position in space and have the corresponding attribute value attached to it. Choosing the correct attributes to visualize requires a geotechnical background, on top of computer graphical knowledge, to make the visuals appealing, informative, and physically correct.

Achieving artistic visuals from geotechnical data with scientific accuracy is difficult since most geotechnical visualizations are limited to basic scientific visuals. Additionally, there is no obvious software, which can handle both simulation data and computer graphical designs. Previous research often uses self-developed algorithms [43, 74, 75] in combination with advanced commercial animation software packages, which are too complex for most users of MPM in the geotechnical field. It would be useful if the visualizations can instead be created using visualization software developed for scientists like the Visualization ToolKit [24].

A coherent data structure is useful to create visualizations from data points. An RMPM simulation consist of many MPM simulations, each is a realization and thus a possible outcome. Together is also referred as an *ensemble*, and each realization involves a set of material points (MPs) carrying physical quantities changing over time. In other words, the ensemble involves a staggering amount of data to process, making it difficult to analyze [5]. It is further complicated by the fact that grouping data points with many attributes is difficult, which is a known issue within data science and goes by the name of *'Curse of dimensionality'* [40, 69]. Therefore, the dataset must be reduced in size while maintaining its

quality. Here, geotechnical judgment is essential to reduce the dataset into sensible outcomes, such that understandable and informative visuals can be produced. Visualizing an ensemble involves showing the expected outcomes and the variability within the outcome, and good ensemble visualizations can lead to more confidence in the outcomes of RMPM [5, 47]. Lastly, each realization within the ensemble has its storyline (the simulation), where actions (variables) lead to a certain outcome (failure profile). Since many storylines have similar outcomes, grouping those stories helps find patterns to understand which actions lead to a certain outcome. Additionally, grouping simulations can help select which realizations represent the stories of a group, and artistic visualization can be employed to enhance these stories further.

1.4. Objectives and outline of the thesis

This research aims to investigate the available software and determine which techniques are suitable to make realistic and informative visualizations for a given RMPM dataset of (time-dependent) dyke slope failure. Geo-engineers are the target users for creating the visuals and therefore, a guideline has been written to understand the computer graphical methods behind the creation of the visuals. Additionally, this thesis investigates how to reduce the RMPM data efficiently, while maintaining the prominent features of the datset. Lastly, correlations between the different characteristics within an ensemble are studied, which may improve the prediction of the probability of flooding based on the information gained from an initial failure and assist in visualizing the prominent features of the dataset. The main research question of the thesis will be as followed:

Research question

How to visualize the random material point method for dyke slope stability problems?

The question is answered using six sub-questions split over two chapters, in which Chapter 4 is about how to make the visualizations, and Chapter 5 is about what to visualize :

Chapter 4

1. What are the requirements for scientific and artistic visualization of RMPM?
2. Which visualization software fulfills the requirements for artistic and scientific visualization of RMPM?
3. How can a (R)MPM dataset be visualized with the chosen software?

Chapter 5

4. How to reduce the RMPM data, while retaining core characteristics of the data?
5. How to verify relations within the ensemble to help engineers predict the probability of flooding with the current methods?
6. How to visualize a RMPM ensemble while retaining its characteristics?

1.4.1. Outline

The thesis is composed of six chapters. Chapter 2 of the thesis describes in detail previous research and related background. The given dataset is described in Chapter 3. The next chapter justifies which software will be used based on the requirements. Chapter 5 can be divided into three parts; where the first part shows how to make a more efficient dataset by reducing the amount of data. The second part verifies and analyzes the reduced dataset. The third part classifies the RMPM dataset into different types of failures and finds relations between the probability of flooding in an early stage. In Chapter 6 the conclusions and suggestions for future research are described.

2

Related Background

In order to make a visualization out of RMPM data, it is necessary first to have some further understandings about why the current assessment in engineering practice needs a more accurate calculation method. Thereafter, the literature presents how the numerical method, RMPM, works.

2.1. Dyke assessment: estimating a probability of flooding

It is stated in the Dutch water act that a safe dyke must withstand the normative circumstances, which can be an extreme high water event or a storm causing high waves hitting the dyke for at least 8 hours [70]. The normative circumstances are defined as the allowable probability of flooding for each dyke section, which has been defined based on the two consequences of failure of a specific dyke section: (1) the affected population and (2) the loss of economic value. In other words, the main function of a flood defense is to prevent flooding, and the Water Act defines flooding to be the ultimate limit state (ULS) of a dyke. In Figure 2.1 a concept of a dyke is given.

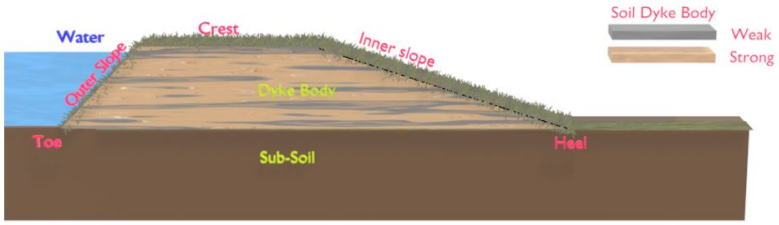


Figure 2.1: Overview of the basic primary clay dyke concepts.

There are different dyke failure mechanisms, for example macro-instability, piping and overtopping, all of which can lead to ULS dyke failure [59]. Within HWBP, the national program for the protection of the primary flood defenses, macro-instability has the largest price tag in terms of both cost and the length of dyke that needs to be reinforced. Macro-instability can occur during an extreme high water event when water infiltrates the dyke body due to the high external water pressures. This infiltration will (1) increase pore pressures, reducing the effective stresses and shear strength in the dyke, and (2) increase the weight of soil. The increased load (higher weight) and reduced resistance (lower strength) can activate a shear zone, potentially causing a circular mass to slide off. In other words, the development of the initial sliding plane is the onset of slope failure, i.e. the start of the failure mechanism macro-stability. In the Dutch assessment guideline for macro-instability (WBI), flooding is assumed to occur whenever an initial slide initiates [33]. So, the assessment guideline defined the initiation of the

macro-instability mechanisms as ULS failure of the dyke, and the probability of flooding is based only on the start of the initial failure [59].

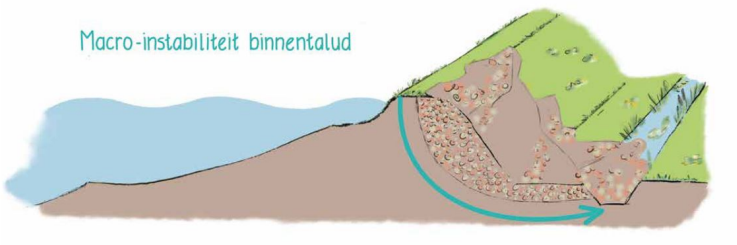


Figure 2.2: Cross section of a dyke profile showing the failure mechanism macro-instability [36].

However, questions are raised about the assessment for the probability of flooding based on the initiation of initial failure. This assessment is over-conservative since the occurrence of an initial slide may not always immediately result in flooding of the dyke [62]. The remaining parts of the dyke (also called the residual profile) may still have a significant height and strength, and may still be able to retain water, as shown in Figure 2.2. After initial failure, which can take a couple of hours to days, a new equilibrium is reached. After that, a possible secondary failure may occur, further damaging the remainder of the dyke. Flooding occurs when the residual profile is no longer able to retain the water [11]. Here, the assessment of initial failure is described first, after which the recent research into residual dyke resistance is discussed.

2.1.1. Analysis of initial failure

According to the guidelines, the probability of initial failure can be approximated based on the Factor of Safety (FoS) using the semi-probabilistic approach or with a probabilistic approach such as FORM or Monte-Carlo analyses [34]. Besides these computational methods, there are also a set of rules of thumbs, which can be applied under specific dyke geometries for which macro-instability is highly unlikely, and as such no flooding due to macro-instability is expected [33, 35, 59].

FoS is a value indicating the safety of the structure, where the resistance (R) is compared to the loads (L) as follows:

$$\text{FoS} = \frac{R}{L} \tag{2.1}$$

The FoS is computed based on design values of the resistance and load, and a conservative empirical formula is used, generated for the guidelines using probabilistic analyses, to converted FoS to a probability of initial failure.

In a probabilistic approach R and L are modeled by their probability density functions, from which a limit state function Z is defined. Hence, failure occurs when

$$Z = R - L < 0 \tag{2.2}$$

The probability of flooding is given by

$$P(F) = P(Z < 0) = P(R - L < 0) \tag{2.3}$$

and can be estimated using one of the probabilistic approaches. As a probability of failure is obtained, the empirical formula does not have to be employed.

For slope failure, R and L are complex, highly non-linear functions of many different parameters. Therefore, in order to compute FoS or to evaluate Z, numerical approaches, such as the limit equilibrium method (LEM) and/or the finite element method (FEM), are often used.

Limit equilibrium method

LEM is the most used analytical method in Dutch practice to calculate FoS or Z for dyke slope stability analysis, which account for the most important types of slope failure (i.e. *rotational slips*) [35]. The

shape of this type of failure may be a circular arc or non-circular curve. Circular slips are in general associated with homogeneous soil conditions and non-circular slips with heterogeneous soil conditions [53].

LEM checks the stability of the slope using the gravitational forces as the driving forces and the forces developed from shearing along the slip plane as the resisting forces[64]. In other words, it compares the load and the resistance at a maximum capacity for horizontal force, vertical force and/or moment. *Failure* is defined as the moment when the load exceeds the maximum resistance in any category, i.e. the FoS is given by the lowest factor between resistance and load for the three categories. As the location of the slip plane is unknown, many slip planes must be evaluated to find the lowest value of the FoS for the given problem. The shape of the sliding plane is limited within defined boundaries to limit the searching space.

Several LEM variants exist, each using slightly different assumptions of the failure geometry, i.e. they use different boundaries to define the search space for the minimum FoS. Most variations assume that the slope fails along a circular surface. The first variants for slope stability analysis are the Bishop method and Fellenius method, also called the method of slices. Both methods check the moment equilibrium of a circular slip plane by splitting the plane into several slices [35, 64]. The guidelines for assessments for dyke slope stability (WBI2017) uses the Uplift-Van model as the default model, which is an extension of the Bishop's method accounting for potential uplift [33]. The traditional Bishop's model and the Spencer-Van der Meij model are suggested as alternatives; the latter model has less restriction on the shape of the failure slip surface and uses a genetic algorithm to find the shape providing the lowest FoS. Due to LEMs assumptions, it is not capable of modeling deformation-dependent behavior, coupled deformation and groundwater, nor slope stability analysis with incorporated structures, i.e. sheet piles walls, diaphragm walls [35].

Finite element method

Although LEM is an often used method, several factors limit the method and make it less accurate. LEM is not able to model changing water level[53]. Additionally, it does not consider the spatial heterogeneity of the soil, and it cannot model the deformation of the slope through time. To account for the limitations of LEM or to increase the accuracy when the estimated probability of flooding is close to the design value, FEM can be used, which is a strength reduction method [53]. FEM is a numerical method, which solves partial differential equations using boundary conditions. In order to solve a problem using FEM, the problem geometry (so-called domain) is divided into elements using a particular space discretization. The elements are connected at the nodes forming a mesh/grid. So specifically for dyke failure, the equations are solved on a discretized 2D dyke body. A constitutive model is used to account for material behavior.

FEM is not often the first choice for a slope stability analysis, since it is more time-consuming and more complex when compared to LEM. Additionally, more parameters are required to characterize the additional features of soil and pore water behavior [35].

2.1.2. Analysis of residual dyke resistance

While the Dutch assessment guideline (WBI) does not take the rest profile after initial failure into account [33], recent studies have investigated whether it may still be able to retain water [11, 59]. If flooding did not occur after an initial failure, a secondary failure mechanism is required to cause flooding, which can be one of the following:

- **Subsequent (secondary) sliding failures**: With similar water pressures, it is not likely that there will be another large sliding failure, since a new equilibrium has been formed. However, a smaller secondary failure can occur because a sharp cut may be created as the first slide deforms.

- **Micro-instability**: After initial sliding, the core material of the dyke is exposed, which reduces the resistance of the material and therefore more easy to be washed out. This failure mechanism is often seen for dykes with a sand core, and is much less likely for a dyke with a clay core.

- **Overtopping**: This failure mechanism happens when there is a high-water event in combination with high waves, causing a flow of water over the top of the dyke. The initial slide can damage the cover of the slope/crest, and an overtopping event after initial sliding can therefore more easily eroded the core of the dyke.

However, there are almost no known cases where secondary failure mechanisms occurred after the initial failure, while rest profiles have been observed [59]. This may be caused by one of the following reasons:

- The residual resistance of the dyke is high and prevents a secondary failure
- The high water event doesn't persist long enough to cause a secondary failure
- After initial failure, enough counter measurements are put in place to stabilize the dyke, preventing secondary failures.
- Once a breach occurs secondary failure mechanisms are difficult to trace back, and the dyke failure may have been accounted to a different failure mechanism.

As a lot of potential strength is ignored by the current guidelines, assessments are over-conservative, especially for dykes where the remaining profile would be large. Therefore, the assessment needs additional measurements to account for the strength in the rest-profile. Despite it being an acknowledged issue, the current assessment methods are unable to analyze the process beyond initial failure, such that relying on the remaining strength can be dangerous. Despite the lack of assessment methods, crude approximations are used to determine the residual resistance of the dyke, based upon the remaining profile [11]. The remaining profile is constructed based on the assumption that the initial failure will result in a crest settlement equal to half the height of the dyke, after which a new equilibrium is found. This settlement is rarely seen in practice, where it is often limited to 1-2m [11], and is therefore most likely too conservative. The likelihood of secondary failures for this remaining profile is approximated. Moreover, the residual crest width of the dyke after initial failure is compared with the required residual crest width to prevent flooding, which takes the following aspects into account:

- The form of the dyke.
- The material, which the dyke is made of.
- The duration of the an high-water event.

Blinde et al. [11] highlights how different reports give different guidelines for the likelihood of secondary failures and the required residual crest, which differs from 1.5 m to 3m for different types of dykes, i.e. sand cored dyke, clay cored dyke. Moreover, most of the guidelines lack a scientific ground. Essentially, there is no obvious method available at the moment to assess the strength of the rest-profile of the dyke after initial sliding, due to the fact that current methods are not able to calculate large deformation.

2.2. Material Point Method

The material point method (MPM) is developed to model large deformations, and can therefore assess the process after initial failure. MPM is first described by Sulsky et al. 1994 [58], it is a relatively new numerical method for dynamic analysis of solids using a (material) point discretization to store information of the material, and a mesh discretization for equation solving. It can be seen as an extension of the finite element method (FEM), and the method has proved to be successful in modeling physical problems with large deformation. In FEM, large deformations are difficult to solve because the geometry of the problem is attached to the mesh. When the mesh experiences large distortions due to large deformations, the analysis cannot continue. MPM separates material from the mesh, and thereby prevents mesh distortion.

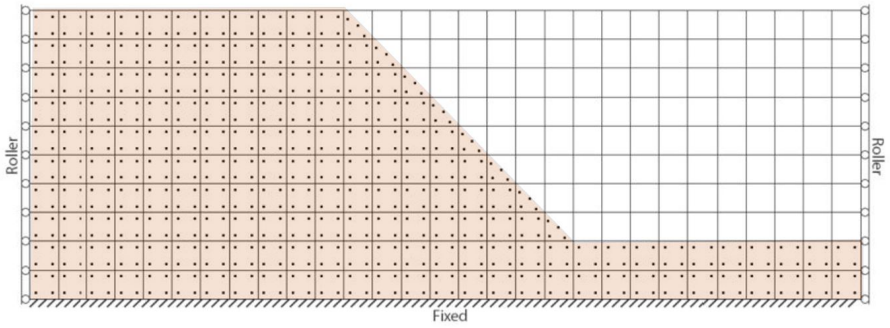


Figure 2.3: A MPM descretization of a soil slope containing the background mesh, material points and boundary conditions [49].

2.2.1. Theoretical basis of MPM

In MPM, a set of material points (MPs) is utilized to track the material through time and space. The MPs are a set of Langrangian points that move through an overlaid Eulerian computational mesh, illustrated in Figure 2.3. It therefore combines the advantages of mesh-based and point-based approaches. The MPs carry all the physical properties of the continuum such as stresses, strains, mass, velocities, strength parameters. The MPs can move freely through a FEM mesh, on which the governing equations can be solved, which makes large deformation modeling possible. As FEM equation can be solved on the mesh, stability issues often seen in meshless methods can be prevented. Moreover, since the computational mesh is only used to solve the balance equation, it can be reset after each step, and mesh distortion is not a problem [7]. This method is promising to analyze geotechnical problems, such as slope stability and micro-instability [6, 18, 51, 63, 65].

Before the start of the calculation, the problem domain is divided over the MPs, i.e. each MP represents a so-called *sub-domain*. MPs carry the mass of a sub-domain, which remains strictly constant during a simulation. MPM therefore automatically satisfies mass conservation. Other state-variables, for example volume, can change over time, enabling material compression or extension. The computational cycle of one time step in MPM, shown in Figure 2.4, is described in three phases [19, 65]. In the first phase the state variables are mapped from MPs to the background mesh (Figure 2.4a). The information required to solve the balance equation on the computational mesh will be transferred from MPs to the nodes of the mesh using mapping functions, i.e. the typical shape functions (SFs) as also used in FEM [65]. The second phase solves the equation of motion on the computational mesh using an incremental time integration scheme. This phase is similar to a time step in traditional FEM (Figure 2.4b). As a result, the mesh is distorted and carries the new updated information. In the third phase, the state-variables are mapped back from the mesh to the material points using the same mapping functions from the first phase (Figure 2.4c). Hereafter, the information associated with the mesh is no longer required, and it can be discarded. The mesh can then be reset to the original position to start a new computational cycle [65].

Similar to FEM, MPM can employ an implicit or explicit time integration scheme. The two schemes have a lot of similarities and a few differences. The explicit scheme was developed first and is used most often because the system of equations is easier to solve. In the explicit scheme, the equations can be solved per node, whereas the full system of equations must be solved in the implicit scheme. The implicit schemes can therefore use larger time steps compared to the explicit scheme, but it is more computationally intensive. The explicit scheme often exceeds the implicit scheme in computation cost, due to the additional time steps, while achieving the same error [19]. However, due to the ease of implementation, explicit schemes are still most often used.

2.2.2. Stress oscillations

Stresses and strains are vital for elasto-plastic problems, such as dyke failures, to model and understand the behavior of the dyke soil [7]. Although the displacement calculation of MPM is fairly accurate, the accuracy of the stress field is not (yet) at the desired level due to oscillations [19]. There are a num-

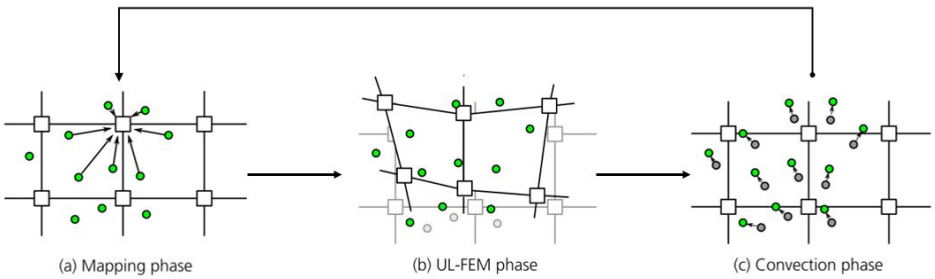


Figure 2.4: Computational cycle of one MPM time step containing three steps: (a) mapping of the state variables of MPs to the background mesh; (b) solving the governing equations on the background mesh; (c) back mapping of the state variables to the material point and reset the background mesh, Figure adapted from Wang et al. [65].

ber of techniques developed to solve this problem, which are mainly attributed to the use of linear FEM Shape Functions (SF) with discontinuous gradients. The discontinuous gradient causes a stress oscillation when the MP crosses the boundary, a problem frequently called the cell boundary crossing problem. Most stress-oscillation solutions change the shape function used in the mapping process. For example, in the Generalized Interpolations Material Point (GIMP) method the finite element shape functions are replaced by shape functions constructed based on linear finite element shape function and a material point support domain [19]. These functions and their gradients are continuous at the element boundary, and can spread the influence of an MPM over multiple elements when their support domain overlaps these elements.

A second problem is only applicable to the implicit MPM scheme. Integration using SF gradients causes oscillations when it is not performed at the Gauss points, while integration using SFs is also stable when performed at MPs. Contrary to most variables, the stiffness matrix required in implicit MPM must be integrated using SF gradients, and therefore the stiffness oscillates when the MPs are used instead of the Gauss points. Double mapping (DM) is used to solve this problem, where MP stiffness is be mapped to the Gauss points from which the stiffness integration is performed. DM ensures a stable integration of the stiffness matrix. The DM-G method combines DM with GIMP to obtain more accurate stress and stiffness [21].

For most constitutive models, the invariants of the stress tensor, i.e. the mean stress (p) and deviatoric stress (q) are particularly useful and are often studied in so-called *p-q diagram*. These diagrams represent the interaction between stresses from which the constitutive behavior can be investigated. The mean stress and deviatoric stress are given by

$$p = \frac{\sigma_{xx} + \sigma_{yy} + \sigma_{zz}}{3} \tag{2.4}$$

in which:

- σ_{xx} = stress in the x direction [kPa].
- σ_{yy} = stress in the y direction [kPa].
- σ_{zz} = stress in the z direction [kPa].

and

$$q = \sqrt{\frac{1}{2}((\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{xx} - \sigma_{zz})^2) + 3(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^3)} \tag{2.5}$$

where the mean stress is negative in compression.

The research of Remmerswaal et al. [51] stated that numerical stress oscillations are more likely to cause failures in weaker zones, which are present in RMPM due to heterogeneity in strength parameters. Therefore, stress oscillation reduction techniques discussed in Section 2.2.1 might be more valuable in RMPM compared to MPM.

2.2.3. Boundary conditions

Application of boundary conditions to MPM, i.e. external water loads, is rather difficult, due to the fact that forces are directly applied to MPs. Moreover, MPs along the boundary are difficult to find and can result in stress oscillation [10, 49]. Secondly, application to the background mesh needs to be at the boundary, however the outer layer of MPs are, by definition, located at the center of the represented material domain. Therefore, the boundary is detected using the Proximity Field Method (PFM) [49], which is based on the level set method [54].

2.3. Incorporating spatial variability in dyke assessment

The material properties of dykes and the subsoil play a large role in the stability of dykes [15, 27, 28]. The material is highly variable, and it is important to include the variability of soil properties into the prediction of initial failure, and modeling of a complete failure process [51, 65].

A limited amount of site investigations can be performed to quantify soil parameters and their spatial variability. Cone penetration tests (CPTs) are a relatively good and easy method to determine geotechnical properties of the soil. During this test, a cone penetrometer is pushed into the ground at a standardized speed and data is recorded. The cone penetrometer measures the resistance and friction at the tip and measures the sleeve friction along the shaft of the cone. Since it is a continuous test, it gives a good quantification of the spatial variability in vertical direction at a specific location[64]. Multiple tests can be used to also quantify the spatial variability in the horizontal direction [15]. While measuring properties using many CPTs can reduce the uncertainty of material properties, measuring the properties for a complete dyke section is unfeasible due to the natural variability of soil. Therefore, a different approach should be used to account for the variability of material properties.

Random fields can account for the local and spatial variability in the underground [28]. This uncertainty is on one hand based on the natural variability of the soil, and on the other hand, the random field can incorporate model and measurement uncertainty. Random fields are generated based on spatial variability, defined by the (local) point statistics and spatial statistics. The point statistics are defined by a probability density function with a mean μ and standard deviation σ. The spatial statistics are modeled using a correlation function, with the horizontal and vertical scales of fluctuation θ_h and θ_v, respectively, as input parameters. These scales of fluctuation describe the distance beyond which minimal correlation is observed. Weakly correlated material has a small value θ. While, a large θ means the material is strongly correlated over longer distances.

The random finite element method (RFEM) combines random fields with FEM within a Monte Carlo framework to account for the uncertainty and impact of soil variability [27–29]. Multiple realizations are performed within the Monte Carlo simulation, each with one random field, which is one possible representation of reality. A random field is one possible outcome for a property field based on the material statistics, and the Monte Carlo analysis is used to evaluate many possible outcomes to compute the reliability of the structure [29, 38, 61]. The result of the Monte Carlo Analysis can be used to assess the reliability of the slope, while including the effect of weak zones, something which is ignored in more standard probabilistic designs using for example LEM. RFEM has to be proven to be useful in geotechnical designs, and especially in dyke assessment [22].

Although RFEM is a significant improvement when compared to FEM, it is still incapable of calculating large deformations due to the limitations of FEM [65]. Therefore, FEM is replaced by MPM to calculate the response of each realization, presenting the Random Material Point Method (RMPM). The random field of a realization of the Monte Carlo simulation is mapped to MPs. Once the material properties from the random field are assigned to each MP, an MPM simulation can be performed. In dyke slope stability analysis, RMPM cannot only compute the probability of initial failure, but also the probability of flooding [52]. Additionally, it can quantify the likelihood of different consequences, which makes it well-suited for risk assessment. For example, in dyke stability analysis, the volume of the failed mass and the total number of failure blocks can be quantified. Moreover, the run-out distance of failed mass in horizontal and vertical direction can be calculated, which helps to give a better prediction of the probability of flooding and better quantification of how much reparation work is needed [51]. However, little is known of how to quantify and represent the likelihood of the different consequences.

2.4. MPM visualization: creating the visualization pipeline

Most visualizations of (R)MPM are limited to assigning colors to a finite volume representing each MP [7], since it is difficult to attach attributes to coordinates within a visualization software. Graphics software is often made to store point coordinates with no attributes because it is often not needed and textures are added separately. On the other hand, visualization using a set of colors is relatively simple with scientific visualization software, such as ParaView [24], which is also used in the (R)MPM research field of geotechnical problems [7, 50, 65]. Visualization using ParaView is so common within the field, that the standard coloring scheme of ParaView can often be recognized within scientific work. However, more research is required to represent (R)MPM data intuitively or generate graphical-realistic visualizations.

Stomakhin et al. [56] is claimed to be the first to use MPM to make renders with a graphcial realism in 2013. The renders have later been recreated by Umenhoffer [60]. Although successful images are created, both mention that the rendering time is a significant limitation of their work due to the high particle count. This high particle count extends the simulation cost, but simplifies the visualization process due to the detail in the particle deformation. In the work by Stomakhin et al. [56] and Umenhoffer [60] simulation attributes were tuned by hand to improve the visualization. This parameter tuning is generally not possible for the visualization of dyke failures, as they would invalidate the outcome of the dyke assessment. Furthermore, it does not visualize any attributes as the snow texture is added in a later stage, the MPM is only used for its coordinates.

While there are visualizations focusing on scientific attributes and other visualizations focusing on realism in renders, there are only a couple of examples where the two are combined. Stomakhin [57] shows how to use MPM to visualize phase changes, i.e. solidifying and melting materials. Ding et al. [16] makes a visualization with graphical realism of the phase changes in cooking, and also visualizes the inherited temperature by using colors to indicate which phase of cooking it is, i.e. a dark-brown cookie indicates an overcooked cookie. While this works shows that it is possible to use MPM data to create realistic visualizations and maintaining a scientific basis, most of the work on visualizations in MPM focus on the numerical method, MPM, instead of the visualization process. Therefore, this process is not described in detail and cannot be easily reproduced. The background behind these visualizations is therefore further discussed.

2.4.1. Visualization pipeline

A visualization pipeline is used to transform raw data into an image, which is first introduced by Haber and McNabb [23]. The pipeline (in general) consists of three steps, as shown in Figure 2.5. Within each step of the pipeline, algorithms are applied to transform data from one form to another, i.e. it can change the dimensionality of the data [24].

According to Hansen et al. [24] the transformations can be categorized into two types, namely structure and type transformations. Structure transformations are the effects that transformation has on the geometry and topology of the dataset. Type transformation refers to the type of dataset that the algorithm operates on, for example, vector data, point data or cell data. These transformations are stored in a library for easy usage and it is possible to adjust when necessary. Connecting varying transformations forms a so-called visualization pipeline.

Figure 2.5: Original visualization pipeline (adapted from Haber and Mcnabb [23]

2.4.2. Libraries

Libraries store the algorithms to create the visualization pipelines, which is also called a visual toolkit [24]. A toolkit/library can be seen as an information center providing algorithms, data representations, and a mechanism to connect these algorithms and transformations. The libraries are often included in application software or visualization tool, which allows the user to apply the algorithms and see the output in a user-friendly manner. Two libraries, VTK and OpenVDb, are chosen to study in more depth because VTK is the framework of ParaView used in the RMPM research community, and VDB is used in various visualization of MPM in the computer graphics community.

VisualizationToolKit (VTK)

The Visualization Toolkit (VTK) is an open-source, object-oriented library and provides functionality for 2D and 3D data visualization [24]. VTK is designed to handle large scientific data for research purposes. The software package must be embedded into an application to see the visual results from the visualization pipeline. It can handle large datasets and contains mechanisms to connect the large datasets and algorithms together to form a working program for data processing. VTK focuses on representations of data for a variety of grid types, including (un-)structured, polygonal and image data. Besides the algorithms for data processing, it contains the rendering algorithms used to finalize the output. VTK's architecture is based on object-oriented principles and implemented in the programming language C++. In a object-oriented scheme, the object is an abstraction, which contains both properties and behavior of an entity in a system. In other words, the object contains both the data and the algorithms useful for this data. VTK's design avoids using small objects to reduce the memory footprint of the data structures. It uses float arrays for point coordinates and a connectivity matrices to represent the topology of a mesh. In order to make VTK easy to use, a number of conventions have been created to ensure code consistency and understandability. For example, all method names are fully spelled out, and abbreviations are avoided, such that method names are easier to remember. Due to these conventions, it can be utilized by a wide spectrum of users.

OpenVDB

VDB is a newly developed C++ library designed by Museth [42] to handle time-varying volumetric data in a more efficient manner. This data is stored within an efficient uniform grid using a unique hierarchical data structure. VDB stands for Volumetric, Dynamic grid, which shares a few characteristics with B+trees (a type of tree data structure). This data structure has the advantage of having no topology restrictions for the sparsity in volumetric data, and therefore it has the possibility to use fast random access. Therefore, it does not require defining specific data patterns to compensate for the slow random access used in other volumetric data structures. To create more innovation using VDB, Museth has made an open-sourced version called "OpenVDB". OpenVDB is recommended to be used for dynamic datasets such as MPM simulations [60, 74] and fluid simulations [43], because it reduces the memory footprint while maintaining fast simulation and visualization.

2.4.3. Visualizing material point data

The basics of the visualization pipeline serve as a backbone to guide through the different techniques from earlier research. These basics show how to create different mesh topology from raw material point data. This thesis aims to not only show the coordinates of the data accurately, but also the attributes of the data points.

Data import

Each visualization pipeline starts with the data import from files [3, 24]. There is a variety of dataset types, and therefore software often have their own file reader and preferred file format. The following terms are used within this thesis with the following meaning:

- **Data type** describes the type of data, for example ASCII (text) or Binary (zeros and ones).

- **Data structure** describes the organization and management of the data, i.e. the geometry and topology of the dataset using data points and their connections, and the general properties of the file.

- **Dataset attributes** describe the data values linked to the data points or data cells (connected points), such as scalar, vector or tensor data.

Generating geometric meshes

The second step is to modify the data and to 'prepare' it for the next step in the visualization process by generating objects on which the point data can be visualized. There are two categories for generating geometric meshes from volumetric data: (1) surface-rendering techniques, where only a mesh is formed from the surface, which could result into a dimension of information is essentially being ignored [24]; (2) volume rendering techniques, which captures the full 3D dataset in a single 2D image. As a result, volume renderings convey more information, but at the cost of increased rendering time due to increased algorithm complexity. Volume rendering techniques are used to visualize clouds, fogs or fire, which cannot be represented by surfaces. It is also used in CT scans, where useful information is contained within the volume and not only on the surface.

An often-used surface rendering technique is Delaunay triangulation, which triangulates any point datasets effectively into a mesh representing the surface of the data [20]. Triangular meshes are most used, due to the fact that other polygons can raise other problems. There is always a plane parallel to a triangle with three vertices, which is not the case when for example using quads. The input is a list of points, which remain as the vertices of the mesh in the output. Using the Bowyer Watson algorithm [48] the triangle mesh is created such that no vertex lies inside the circumcircle of any triangles of the mesh. The circumcircle is a circle which passes through the vertices of a triangle. The output of the Delaunay triangulation is a polygonal mesh. The mesh is written in a certain structure, containing a list of vertices and their location and edges and a collection of ordered pairs of vertices.

This technique is successfully used by Wang et al. [67] to simulate and visualize ductile fractures using MPM. Using tetrahedron mesh to visualize MPM is in contrast to other research [56, 60] where particle-based estimation is used. Using the tetrahedron mesh has the advantage of natural accommodation for textures, while using a particle system extra steps have to be taken [67].

Material points can be assigned a certain volume in space using the volume rendering techniques. Voxels can therefore be used, which are placed in a regular grid. Each cell in this grid can either be filled or empty, and attributes can be attached to each cell in the three-dimensional space [20]. Voxels positions are only encoded from the position in the grid, i.e. they do not have an attribute that encoded their exact position. This technique has been used for particle systems such as MPM, for example in the research of Stomakhin et al. [56]. Material points can be assigned to a single voxel, or multiple voxels can be used to increase the resolution of the outcome, for example by approximating a sphere around each MP with voxels. Data processing can be used to grow, shrink and erode the voxels, which can improve the 3-dimensional representation of the data.

Color mapping

Color mapping is an often used mapping technique to visualize scalar data in colors through a lookup table [24]. The lookup table stores various colors, ranging from a minimum and maximum scalar value, and the color is interpolated to each scalar value within the range. Values are clamped to the maximum and minimum color when they fall outside the range. Scalars data are single values of data associated with each point and/or cell, which is one of the most common types of data in real world. Therefore, there are many different algorithms to visualize it. Color mapping should accentuate important features, while minimizing less important details. The human eye naturally separates similarly colored areas into class regions. One should note the importance of the role of color schemes; it plays a substantial role, which influences the overall expressiveness of the visual representation [5]. The design of different color schemes depends on the task of the user, which can be divided into three levels according to Andrienko and Andrienko [9].

The first level distinguishes the difference individual data values and between sets of data values, illustrated in Figure 2.6a and figures 2.6b and c, respectively. Unsegmented color scales are associated with individual data values because each number can be associated with a unique color. In comparison, segmented color scales are associated with a range of data values. To visualize different sets of data with no underlying correlation a so-called qualitative color palette (Figure 2.6 a) should be used. The second level is the distinction between lookup and comparison tasks. A lookup task is designed of color scales to identify values, which means a colorbar that fits perfectly for that data set. While comparing two or

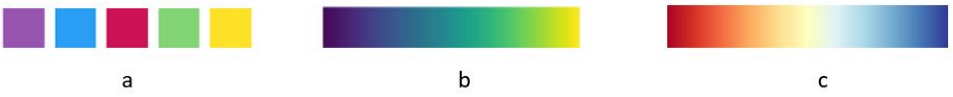
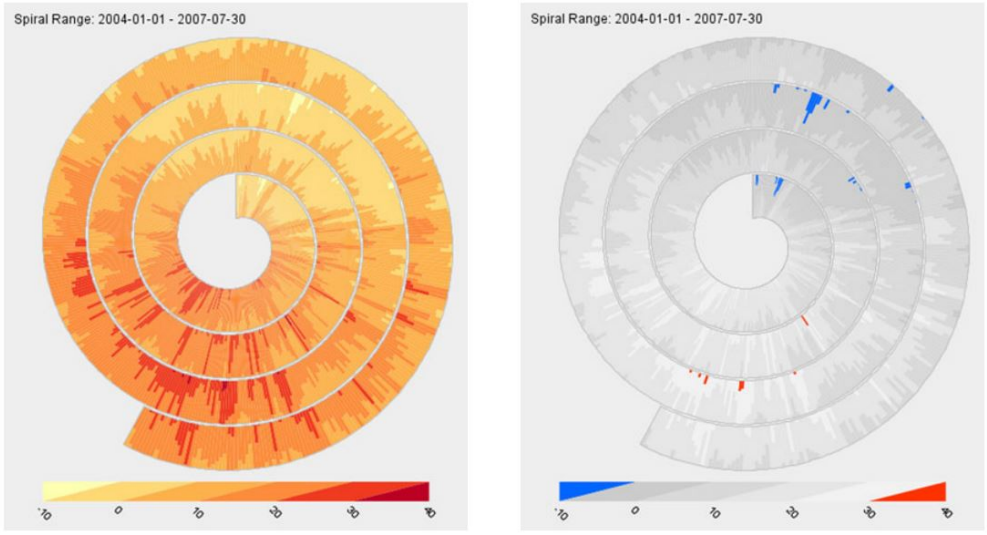


Figure 2.6: Three different colormaps are shown with a purpose to add information to the representation, (a) qualitative palette which is used to express different categorical data. These colors are not correlated with each other; (b) sequential colormap is used to represent numeric data with no boundaries; (c) diverging colormap is used to represent numeric data with boundaries.

more datasets sharing with a common variable of interest requires a global color scale comprising all value range of all datasets that participate for comparison. The third level is to distinguish the difference between identification and localization tasks. An example is shown in Figure 2.7, which shows the daily temperature for 3.5 years mapped to a spiral form. To identify extreme data, i.e. outliers, a diverging colormaps are used (Figure 2.6c), application to the example is shown in Figure 2.7b. It brings attention to certain data values, while a gradient is better for the identification of values within the middle of the range in Figure 2.7a.



(a) Color scale for identification of values. (b) Color scale for localization of extrema.

Figure 2.7: An example where the difference between identification and localization using different colormaps is shown [5], the example uses daily temperature values along a spiral time axis.

Contouring

A natural extension to color mapping is contouring, where similar regions are actively separated with a boundary. A human brain naturally groups data with similar colors, but this automatic grouping of the human brain can be different for each individual, preventing easy data transferring. Contouring separates the data such that everybody observes the visuals in the same way. Examples of 2D contouring are displays of weather maps showing low and high-pressure areas or topological maps with contouring lines representing constant elevation [20]. Three-dimensional contours are called isosurfaces,

which are for example used to display body tissues or temperature in fluid flow. Isosurfaces are a three-dimensional surfaces representing a constant value of a scalar field, and it is usually approximated by polygonal primitives, such as triangles, cubes and hexagons [24]. An example is shown in Figure 2.8.

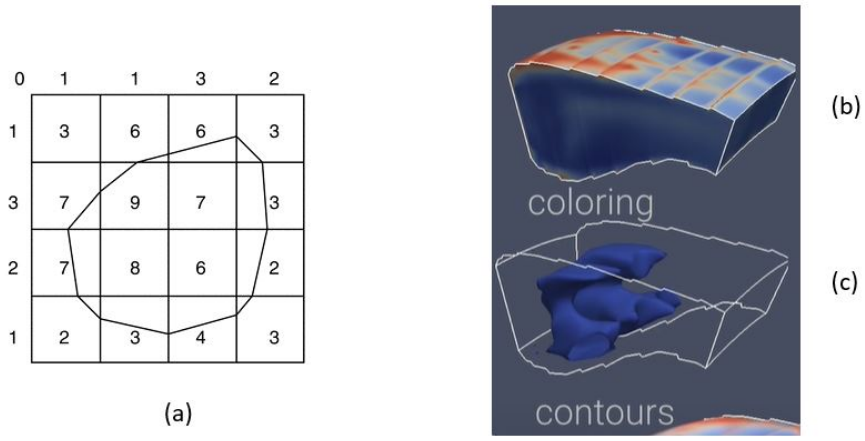


Figure 2.8: An example of contouring; (a) 2D contour with a line value = 5; (b) a colored object; (c) contouring in 3D, which is called isosurface. Figure adapted from[24].

One approach to create the isosurface is to divide the domain into a matrix and to treat each square (for 2D) or cell (3D) independently [24]. This technique is called marching squares in 2D and marching cubes in 3D. This technique is based on the assumption that the surface can pass through a cell in a finite number of ways [54]. A case table is made with all possible topological states of a cell, which is based on the number of vertices in a cell and the number of inside/outside relationships a vertex can have with respect to the contour value. As an example, a 2D cell with four vertices is shown in Figure 2.9. A vertex is considered inside a contour if its scalar value is larger than the scalar value of the contour line, and outside if the value of the vertex is lower than the value of the contour line. Since the example has four vertices, there are $2^4 = 16$ possible ways for the contour line to pass through the cell, which is listed in Figure 2.9. When the case is selected from the case table, and the location of the contour line can be determined using interpolation of in the case 3D it is a cell edge. Thereafter, the algorithm proceeds to the next cell and repeats the same procedure. In the end, it will have created a mesh. A 3D cell (voxel), which has 8 vertices will result in $2^8 = 256$ cases. The output of the marching cube algorithm will also result into a triangulated polygonal mesh

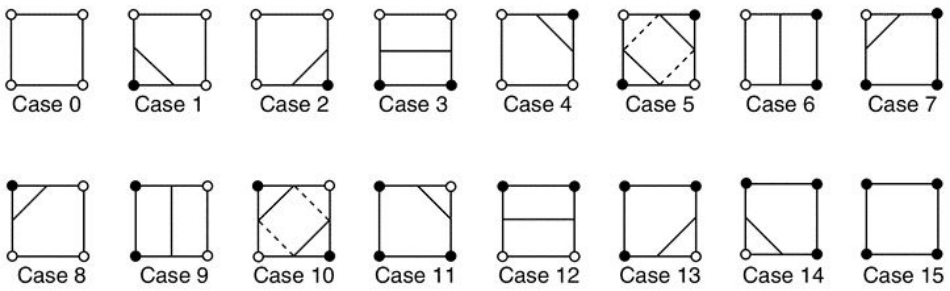


Figure 2.9: A visualized case table of the sixteen marching squares cases. The dark vertices indicate a scalar value above the defined contour values [24].

2.4.4. Visualization software

Scientific visualization

ParaView is a scientific visualization software to analyze large datasets [24]. It can use parallel processing to support large-data visualization. ParaView is built on top of the VTK library and is the foundation of the layered architecture structure of ParaView. The second layer is the parallel extension of VTK, which supports streaming and parallel executions on distributed- and shared-memory machines. The last layer is ParaView itself, which provides a graphical user interface (GUI) and supports real-time visualization and rendering of the data via different techniques like: the Level of Detail (LOD), parallelism and hardware acceleration. This software is made for easy visualization of scientific data and usage of colors, for example has an informative and practical purpose. An illustrative example is shown in Figure 2.10, where the slope is visualized using a sequential colormap representing the undrained shear strength.

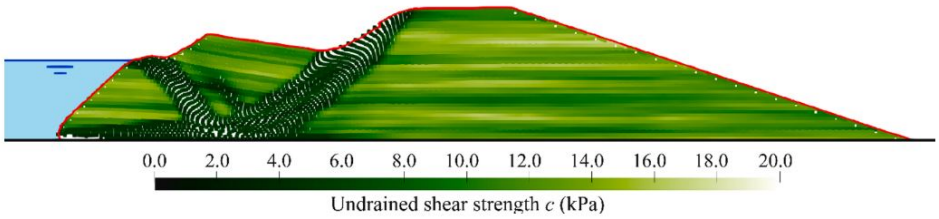


Figure 2.10: An illustrative example of visual representation of a MPM data file from the research of Remmerswaal et al. [52], illustration of water is applied in a post-processing software.

Amira is initially developed by the scientific visualization group at the Zuse Institute Berlin (ZIB) and is a commercial product [24]. The major focus of the software is the visualization and analysis of volumetric data, which is useful in medicine and biology fields. The volumes can be segmented, and the 3D polygonal models can be reconstructed and further processed, including conversion to tetrahedral volume grids. It can handle numerical post-processing and visualize the flow. The finite-element data is generated externally from Amira and defined on unstructured grids. Amira supports the generation of tetrahedral volume grids and triangular surfaces grids for numerical simulations. Scalar quantities can be visualized using color coding on the grid, and displacement vectors can be visualized using glyphs.

Artistic software

Blender is an open-source, 3D graphic software, which includes modeling, animation, simulation rendering and motion tracking. Scripting is supported in Blender by its Python API, which allows for automatizing of some manual procedures. Moreover, it also allows for plugin development, making it possible for users to extend with their own packages for their own needs. Although there is no MPM visualization made using Blender, other types of scientific visualization are created using Blender. In the research of Andrei et al. [8], they have chosen to work with Blender on purpose to visualize molecules in a directly informative way. Based on Blender, BioBlender is developed to visualize elaboration of protein motion of their physical and chemical features. Their additional developed tool enabled them to calculate field lines and export them in an ASCII file, allowing them to be imported in Blender. This tool imports the 3D surface in an object file and the grid file (.dx). In the research of Kretz et al. [37], Blender (v. 2.68) is used to simulate the discrete element method (DEM) and to validate a screw feeder system. They have developed a simulation tool to avoid the common problems in practical applications like: missing stability of material flux over time. The simulation tool can show and implement geometrical and physical properties. Simulation parameters (i.e. shear forces, wall frictions) are fitted to the real experimental data. More than 150,000 spherical particles are used during the simulations with relatively low simulation time (0.04 s each time step). Remarkable is that all the simulations are performed with a Quad-core i7 processing unit with 3.4GHz using a memory of 32 GB RAM.

Houdini and Autodesk Maya are two commercial animation software, which are both used for creating visual effects in games and films. The tools developed within the software include particles, reflections and animations systems [39, 45]. The tools and abilities are till a certain point are similar to the open-source software Blender. However, Houdini is developed originally for simulation purposes and is more advanced in this field than Blender. While Autodesk is more advanced in the field of 3D modeling. Both software are used for visualization and simulation purposes of different scientific research with numerical methods, i.e. FEM [67], FLIP/SPH [12, 14], MPM [56, 60, 72].

Coupled software

The gap between scientific and artistic visualization software is that with scientific software texturing, lighting, and rendering are often lacking. In contrast, artistic software is often not able to handle the simulated data. The scientific visualizations made with Blender require the generation of object files outside of Blender. At the same time, it is desired to handle both post-processing completely within one software or even run the simulation from within the software. This separation of post-processing steps can be solved by the inclusion of libraries into the artistic software.

Blender-VTK: Since Blender is an open-source software, which makes the development of new libraries as an extension of the software an option, Chris Want was the first to combine VTK with Blender released on Github in 2017 [68]. This first attempt of combining VTK with Blender required the Python interpreter of Blender to import the algorithms of the VTK-library, the developed script converts a VTK polygonal mesh to a Blender mesh and vice-versa. Later, Imboden created a VTK add-on, i.e. an external extension, which uses the nodes system of Blender to represent the VTK algorithms [30]. The algorithms are set up as Blender nodes; their input and output parameters can be easily adjusted inside Blender. The outcome automatically updates within the various Blender views.

Houdini-OpenVDB: Another coupled software of libraries within an artistic software is Houdini and OpenVDB. Here the library is incorporated within the software from version 12.5 and not placed as an add-on, resulting in fewer issues with changing versions of either side of the software and less prone to bugs.

OpenVDB is used to represent sparse volume data in an efficient manner, which reduces on computer memory and resulting to faster computations. A sparse volume data is a dataset with high percentage of data cells without a value, i.e. smoke or clouds. OpenVDB is an optimal combination with the software Houdini, since it designed for animation purposes. The closed-source version of OpenVDB is combined with Houdini to visualize MPM in other research to simulate different materials mentioned previously [56, 60].

2.5. Large dataset processing

In order to create good visualization, it usually requires good data structure. The RMPM dataset is a high-dimensional dataset containing both (mostly) dependent and (mostly) independent variables. The separate realizations are mostly independent, and represent an *n-dimensional* domain, with n being the number of realizations [73]. The variables within a realization are mostly dependent and define a *k-variate* dataset, i.e. the number of variables within a realization. If at least one of the attributes within the ensemble is associated with time, the ensemble is called *time-oriented* data. Increasing the number of variables will complicate the identification of patterns. Since Covid-19 has been dominating the news while writing this thesis, it will be used as an example to explain a term and a known issue within data science called: *Curse of Dimensionality*. A group of n-people have conducted the PCR-test after having diner with one Covid-19 positive person and waited for the results in one room with 1.5m social distance. After a while, the results came, which is the first attribute to each group's individual. The group is then split up into two rooms according to the result: positive and negative. The doctors would like to know whether being tested positive is related to gender, so the groups are split further into four rooms. People within the room have 1.5 meter social-distance, while the distance between the rooms is much more. The group is then further split based on age, weight, hours of sleep per night, Etc. until some point, each person has a unique set of attributes and therefore has their personal room, which

is great for the patients and the people in need of quarantine. However, the doctors cannot conclude which group of people are more prone to the virus as there are no clusters anymore and all distances between patients are the same, while there were clusters when the k-numbers of attributes were lower.

2.5.1. K-means sub-clustering method

One of the method, which attempts to give an overview of the data is by using the K-means algorithm. It is desired within each MPM simulation to classify the MPs to different sliding masses, which is useful to evaluate the consequences.

Using visual inspection, one could detect sliding masses during a retrogressive failure simulation of a dyke. However, evaluating all the realizations by hand is impossible. The process is automated in RFEM using the K-means clustering algorithm (K-MCM), which can effectively separate the initial failure from the remainder dyke in RFEM [61]. The sliding body (initial failure) is separated from the stable body (remainder of the dyke) using displacements stored at the material points. This method is effective when the number of groups is known beforehand. However, using this technique to separate multiple retrogressive failures in MPM can be difficult because the number of clusters is unknown. For example, if the number of clusters is assumed to be two, the retrogressive failures can be by accident grouped with the initial failure or the remainder of the dyke. The clustering algorithm helps to focus on the initial sliding mass, which is of importance because it is the largest failure surface. Additionally, understanding the behavior of simulation from the initial failure surface aids in current practice, as it could not calculate beyond this point.

2.5.2. Parallel coordinates

One of the widely used scientific visualization techniques is parallel coordinates, which is made for multivariate data and high-dimensional geometry [5]. It is suitable for the visualization of exploratory data analysis, which guides the user to identify interesting areas for further research, and can help to get an overview of the ensembles. Parallel coordinates can be used for different goals, and the following goals are applicable for the scope of this research [26]:

- **Classification** is to map data to pre-defined classes and to use brushing for the selection of a group of data to analyze further.
- **Clustering** is to identify a set of data items showing similar characteristics. Such that different sets of data can be grouped together, which is different from *classification* as there are no pre-defined classes.
- **Dependency modeling**, where it shows the dependencies between the models, which then can be translated into graphs where a closer look at the dependencies can be made.
- **Summarizing**, where it shows multivariate data in two dimensions.

Different coordinate systems visualize data differently, and therefore it is important to understand how to interpret because the choice of the coordinate system determines what pattern is exhibited. As an example illustrated in Figure 2.11 a, a Cartesian coordinate system is used to show four points data in 2-dimensional space, where it is represented as four polylines in the parallel coordinates space. A parallel coordinate graph is constructed by placing the axis in parallel instead of embedding a 2-dimensional Cartesian coordinate in a plane. Other mappings can also be expressed using the envelope of lines in parallel coordinates. These are described and proven by Inselberg [32], in Figure 2.11b to e, the common patterns in Cartesian coordinates are shown along with their dual representation in parallel coordinates. These above mentioned relations are for ideal situations, which are not always the case when using real datasets. In Figure 2.13 an example is shown with three types of clustering, it illustrates the possibility of using parallel coordinates to detect clusters and even the correlations within these clusters can be determined.

Using a datasets with multiple attributes can be visualized at once using parallel coordinates. A graph N-dimensional is then made by using N-copies of the y-axis are made each plotting a different variable. Often, only vertical axes are used when illustrating parallel coordinates. Other layout choices

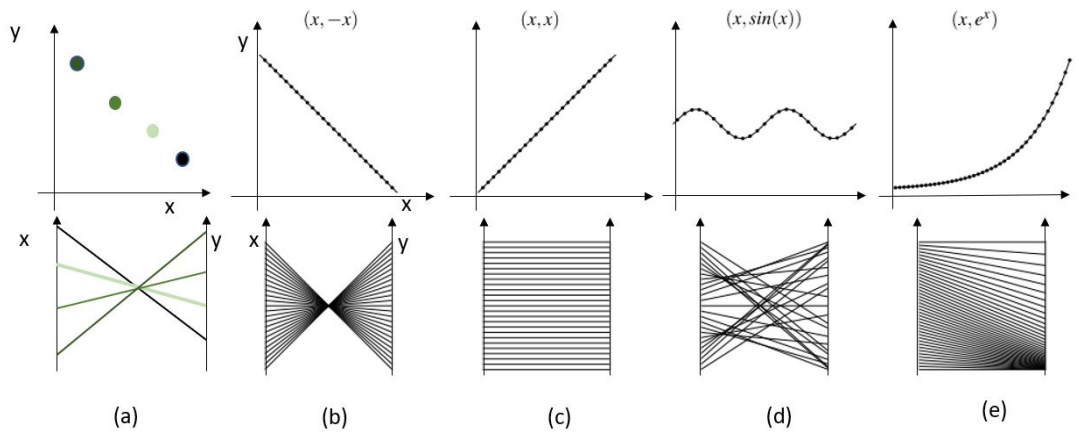


Figure 2.11: Common patterns in Cartesian coordinates (top) and their representation in parallel coordinates (bottom), Figure adjusted from [26]; (a) translation from Cartesian points to lines in parallel coordinates; (b) linear negative; (c) linear positive; (d) sinus; (e) exponential.

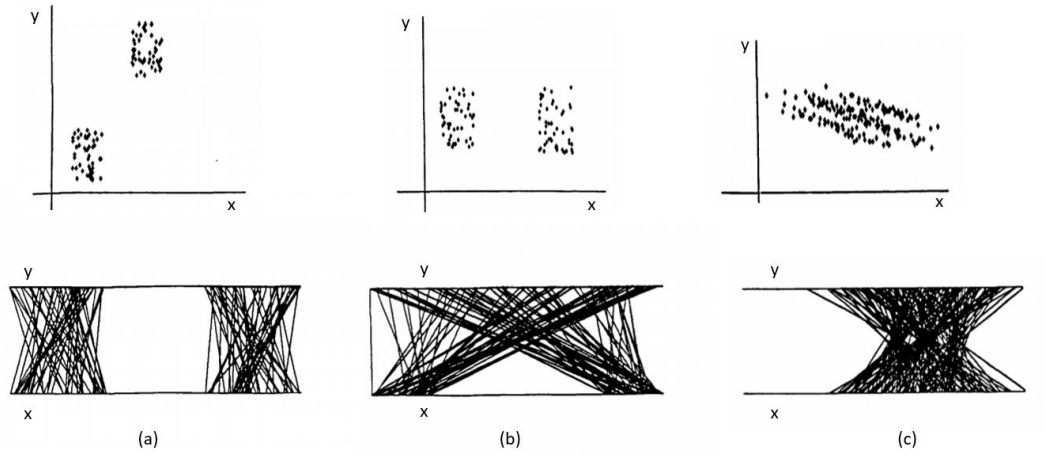


Figure 2.12: Three examples of clusters shown in parallel coordinates; (a) clusters that are separated in both x an y direction; (b) clusters that are separated in x direction, but not in y direction; (c) clusters that are neither separated in x and y direction, Figure adapted from [71].

depend on the number of axes, the range of data and personal preference. The full potential of parallel coordinates only unfolds itself when there is supported interaction, i.e. re-ordering, flipping and scaling the axis. The importance of ordering the axis is shown in Figure 5.15, where the two clusters are presented clearly after ordering in Figure 5.15b. With the help of brushing, which is an interaction technique to highlight a selection of data, the behavior of the clusters are even more clearly shown.

In order to make a visualization using parallel coordinates, which shows certain correlations of the dataset clearly, many decision has to be made by the user [26, 46]. These decisions are often subjective. A good example of a deficiency in parallel coordinates is "the clutter" and the challenge to reduce this. There are objective measures for the clutter, however in practice, it usually depends on the context and individual experience of the receiver. As a result, many researches are facing the following challenges when visualizing data with parallel coordinates:

- Overplotting
- Ordering of axes
- Time series
- Uncertainties

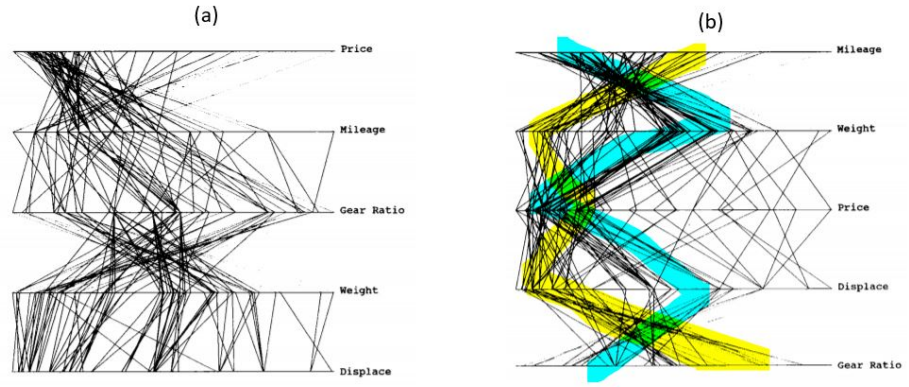


Figure 2.13: Parallel coordinates of a five-dimensional of an auto-mobile dataset; (a) no noticeable correlations; (b) after ordering axis two five dimensional clusters are shown and highlighted, Figure adapted from [71].

Preventing clutter for parallel coordinate for time series typically implies that the complete time series cannot be taken into account. Instead only a couple of moments in time should be plotted.

Outliers

Outliers are often separated from the rest of the data and are considered as a flaw or an error in the measurements [47]. However, outliers can still contain significant information and influence the decision process. Losing an outlier by merging it with the general context means obstructing the visual exploration in the observed data. Therefor, it would lose the truthfulness of the newly produced visuals. It is important on the other hand to treat the outliers separately, to remain the quality of data abstraction. Novotý [47] proposed a method to treat outliers by using a 2D bin technique to preserve visualization of outliers even for large datasets. Binning is frequency-base representation of the original data. Histograms are well known 1D binning techniques, while 2D binning divides a 2D data space with multiple intervals (bins) and colors each square according to its frequency. Novotý's studies shows that it is not necessary to use multidimensional binning. In fact it is not efficient at all, since this type of binning demands enormous memory and not always gives a better result when compared to 2D binning. Each pair of adjacent axes in the parallel coordinate graph represents a pair of dimensions in data and can be binned in a two-dimensional subspace. Once the data is represented in 2D-bins, outlier detection can be started by identifying the isolated bins with low-populations. Once the bins are separated into main bins and outlier bins, the outlier extraction can be performed in parallel coordinates by using brushing, which highlights a certain range of realizations, in this case the range in which the outliers fall, to visualize [26]. As such the behavior of the outliers can be studied for all the variables in the parallel coordinate plot, while in the 2D histograms only two variables can be observed.

2.6. Conclusion

The RMPM field of research can play an important role in making a more accurate calculation for current dykes for the failure mechanism macro-instability, as it accounts for the post-failure behavior. However, geo-engineers are limited in communicating findings to a larger audience, i.e. stakeholders involved in the decision-making process of designing the dyke and often without geo-engineering knowledge. Using visual communication helps to explain the problem and solution in a more intuitive and appealing manner.

In recent years, research in computer graphics has yielded different methods of using MPM to simulate and visualize a variety of materials. Although these research has not been investigated in simulations of the geo-engineering field, it is confident that it is possible to create visualizations with a certain graphical realism from RMPM data.

The challenge of making visualizations from RMPM data arises not only from an investigation of how to visualize the dataset, but also the number of MPM realizations to visualize and the amount of data as soil has compared to other materials more properties to describe the material. Therefore, it requires both geo-engineering and computer graphics background. This thesis investigates the possibilities of how to visualize an RMPM dataset for visual communication purposes, which makes the gap between computer graphics and geo-engineering smaller.

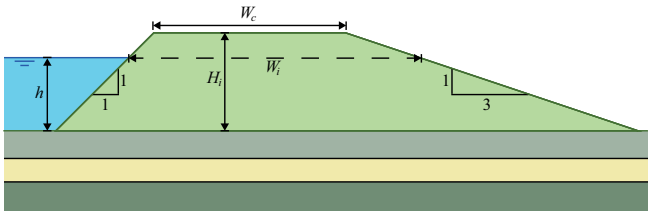
3

Simulation description

The visualizations created within this research are based on a given RMPM dataset performed on an idealized dyke cross-section as described by Remmerswaal et al. [52]. The full RMPM dataset comprises 10 000 realizations, i.e. each realization is a possible outcome of the dyke, performed in parallel on a grid computer system (Spider) at a national computer center in the Netherlands (SURFsara). As this research focuses on the failure process, all realizations with stable dykes, i.e. dykes without any failure, are excluded from the analysis, as they do not provide information on the failure process. There are in total 1189 realizations that experience failure and are of interest to visualize.

3.1. RMPM data description

Geometry: The clay dyke has an initial crest height (H_i) of 5 m and a crest width (W_c) of 10 m. The outer slope of the dyke is 1 on 1, and the inner slope is 1 on 3, shown in Figure 3.1. An external water load is added to the outer slope of the dyke, 0.25m below the crest ($H - h = 0.25m$). Deep slides through the foundation layers below the dyke are limited by a fixed boundary condition at the toe of the dyke. In other words, the dyke is assumed to be on a strong foundation layer.



Geometry	Soil properties	
H = 5m	γ = 15 kN/m³	θ h = 24 m
W = 10m	E = 1000 kPa	θ v = 0,5 m
Inner slope 1:3	v = 0,45	ci = μ_{ci} = 13 kPa, CoV = 0,2
Outer slope 1:1	Hs = -50 kPa	cr = μ_{cr} = 5 kPa, CoV = 0,2
h = 4,75m	Constitutive model = Von Mises	

Figure 3.1: Geometry of the dyke used for the given RMPM dataset and a summary of the material properties and model conditions.

Random fields and material properties: A Von Mises constitutive model with shear strength softening is used to simulate the strength reduction caused by a build up of pore pressures during the failure. This model uses an initial undrained shear strength (c_i) and residual undrained shear strength (c_r). Random fields are used to incorporate the effect of the spatial variability of these shear strength properties. The values of each random field are generated using Local Average Subdivision (LAS) and are mapped onto the MPs in their initial position at the start of the analysis. The spatial variability is considered only for c_i and c_r, which are assumed to be fully correlated. All other material properties are assumed to be deterministic, as presented in Figure 3.1.

The random fields are generated based on the point and spatial statistics of the initial and residual undrained shear strength. Both properties are assumed to be normally distributed with a coefficient of

variation (V) of 0.2, and a mean c_i of 13 kPa and a mean c_r of 5 kPa. The spatial statistics are defined by the vertical scale of fluctuation θ_v = 0.5 m and the horizontal scale fluctuation θ_h = 24 m. This horizontal scale of fluctuation results in a significant layering of the spatial variability.

The unit weight of the dyke clay is γ = 15 kN/m^3 with an elastic modulus of E = 1000 kPa and Poisson's ratio ν = 0.45. The softening modulus of the undrained shear strength is equal to H_s = -50 kPa. In order to create a significant number of failure processes with limited computation cost, the material properties were chosen relatively weak to enforce a higher probability of failure than is typically observed in practice.

Discretization: A mesh of 4 noded (linear) square elements is used. Each element has the size $\Delta x = \Delta y$ = 0.25 m, and four equally spaced material points are placed within each element, which leads to 6400 MPs in total. The nodes at the bottom boundary are fixed to mimic a strong foundation layer. The random field is generated with a cell size corresponding to the material point domain, i.e. $1/2\Delta x$ by $1/2\Delta y$, which captures the spatial variability adequately, since the random field size is four times smaller than the smallest scale of fluctuation, i.e. θ_v = 0.5.

Implicit MPM is used in this study for two reasons: (1) the implicit time-scheme saves high computational costs [66], and (2) algorithms to improve the accuracy of the implicit MPM have been developed, which are not applicable for the explicit MPM. For the implicit scheme, a larger time step equal to Δt = 0.01 s can be used compared to explicit MPM [21]. The DM-G technique is used to reduce the stress oscillations.

Initial conditions: The in-situ stresses are generated under the assumption of an elastic material and are generated quasi-statically using gravity loading. Plasticity and dynamics are considered after the start of the simulation. The plasticity generates unbalanced forces at the nodes of the mesh and may lead to an initial failure. Hydrostatic pressure is applied on the boundary segments at the outer side of the dyke from the base level till $0.25m$ below the crest of the dyke.

End conditions: The simulation is terminated when (1) the crest height is lower than the water level, i.e. the dyke floods, (2) a dyke finds a new stable equilibrium, or when (3) the maximum simulation time is reached, i.e. $t = t_{max}$ = 40 s. In case the maximum simulation time is reached, and the height of the dyke is still above the water level, the realization is assumed to be stable.

To determine if a new equilibrium is reached, the change in average horizontal displacement, average vertical displacement and average plastic deviatoric strain is evaluated over one second. The dyke is assumed to be stable when these changes are smaller than 0.1%. Furthermore, a simulation with a residual crest height lower than $0.1m$ above the water level will not be terminated, even if the dyke is stable according to the conditions mentioned above. In this case, the simulation only stops when the maximum simulation time is reached.

3.2. RMPM Output

The generated data is presented using different file types. Two files store the data of interest to visualize: an MP-file and a mesh-file created for each time step in a .inp file and .vtk file, respectively. The MP-file stores MP coordinates along with attributes per MP, and the mesh-file stores the coordinates of the elements along with the attributes per node. In Table 3.1 an overview of the included attributes is shown. Two additional files which could be of interest are (1) the boundary file, which stores the coordinates of the external geometry in a .vtk file format, and are determined using the boundary detection method, described in Section 2.2.3. And (2) the cluster file, which stores some additional information of each failure block detected by the K-MCM algorithm (Section 2.5.1), the additional attributes are listed in Table 3.2. This file does not contain much information, but it could be extended.

A summarizing overview of the output files of interest is given in Table 3.3, along with the order of magnitude of each file based on an example realization with 180 time steps and two failure blocks.

The K-MCM algorithm is used to detect failure blocks and label MPs in which failure block it belongs, described in Section 2.5.1. The algorithm is modified for RMPM by combining the K-MCM technique with a Euclidean displacement threshold to detect multiple clusters. The threshold for this dataset is defined to be $0.3m$. The first step of the algorithm computes the mean of the Euclidean displacement of all MPs per time step; while the mean does not exceed the defined threshold, the dyke is considered to be stable. If it exceeds the threshold, the K-MCM algorithm subdivides the dyke body into

Table 3.1: An overview of the MP-output file, the attributes describes each MP per time step

MP-File, attributes for each MP:		
x, y, z coordinates [m]	x, y, z, xy stresses [kPa]	Initial undrained shear strength [kPa]
Displacement [m]	Deviatoric stress [kPa]	Undrained shear strength [kPa]
Velocity [m/s]	Mean stress [kPa]	Cluster [-]
Acceleration [m^2/s]	Plastic deviatoric strain invariant [-]	

Table 3.2: An overview of the cluster-output file, the attributes describe each cluster per time step

Cluster-file, attributes per cluster per timestep	
Time [s]	Cluster number [-]
Volume [m^3]	Displacement [m]

two clusters; the initial slide and the remainder of the dyke. The mean of the Euclidean displacement of MPs in the remainder of the dyke is then again computed and compared with the threshold. If this exceeds the threshold, it will be again subdivided into two new clusters, i.e. secondary cluster and a new remainder of the dyke. The process will be repeated until the remainder of the dyke is defined to be stable, i.e. the mean of the Euclidean displacement of the remainder of the dyke is below the threshold. The algorithm is executed for each time step. The advantage of the modification compared to the original is that it is not required to know the number of clusters upfront.

3.3. Failure profiles

Each RMPM dataset, also called an ensemble, has a set of consequences leading to a certain failure profile. These realizations can be classified into a set of failure profiles. Grouping the different types of failure profiles will help reduce the amount of data to visualize. It can therefore help to design representative visualization of the ensemble. While some classification techniques can detect patterns within the data, and thereby find groups automatically. Here it is chosen to construct the groups based on previous experience. In the current guidelines for macro instability, only two groups are created: stable and failed dykes [1, 33, 70]. However, when residual dyke resistance is included, as mentioned in Section 2.1, multiple failure patterns can be recognized for failed dykes, which can either lead to flooding or no flooding, and therefore important to understand the failure process and to recognize a certain failure profile from an early stage.

Table 3.3: Overview of the output files per realization, calculation of data size are based on one example realization with N=180 time steps.

File	Description	File size	Extension
MP file	Coordinates of all MPs, format of data, attributes per MP. One file per timestep.	$N_a N_t$ ~2,8 MB	.inp
Mesh file	Coordinates of all nodes in the mesh, mesh elements, attributes per element. One file per timestep.	$N_a N_t$~5,1 MB	.vtk
Edge file	Coordinates of MPs at edge of the dyke, format of data. One file per timestep.	$N_a N_t$ ~15 KB	.vtk
Cluster file	Calculating average height/width per cluster for each timestep. One file per realization.	1 x ~5 KB	.res
Example calculation of total file size for realization 1 with 180 (=N_t) time steps.		*~ 1,36 GB*	

3.3.1. Types of failures in macro-instability

Five types of failure profiles are defined, which are related to possible macro-instability failure profiles and each MPM realization within the RMPM data should be classified within these types of failure profiles. It is important to understand whether stabilization occurs after the initial failure block forms, or whether retrogressive failure is triggered [33]. Moreover, it is important to separate the cases where flooding occurred, either after an initial failure or after retrogressive failure, from the cases where flooding could be prevented because a stable equilibrium was reached. Finally, RMPM research suggests that failure blocks occur to be not only rotational but also horizontal [52].

Horizontal failures are often not detected as flooding during the simulation because the displacement is lower than rotational failure and therefore defined as stable. Additionally, during horizontal failure, the height of the crest does not decrease much, so it is not defined as flooded. However, when making it a three-dimensional problem, a horizontal failure mechanism almost always leads to flooding because it causes a breach in the dyke segment. Therefore, separating horizontal failure mechanisms in two-dimensional analyses is important.

To summarize, the following types of failure can occur within the RMPM data:

1. Stable after one failure block (**No flood 1**), Figure 3.2b and c, 11 out of 1189 realizations.
2. Flooding after one failure block (**Flood 1**), 150 out of 1189 realizations.
3. Stable after retrogressive failure (**No flood R**), Figure 3.2d, 101 out 1189 realizations.
4. Flooding after retrogressive failure (**Flood R**), Figure 3.2e, 867 out of 1189 realizations.
5. Horizontal failure (**Horizontal**), Figure 3.2f, 60 out of 1189 realizations.

As previously mentioned, the group of stable dykes, see Figure 3.2a, are excluded from the remainder of the analysis and therefore not included in the five classifications summarized above. The retrogressive flooded realizations are the most common failure type within this dataset. The failure type 'Flood 1' is not described in the research of Remmerswaal et al. [52], it distinguishes shallow (Figure 3.2b) and deep (Figure 3.2c) initial failures. For this research to classify simulations with only one failure block, the categories 'flooded' and 'not flooded' are more suited.

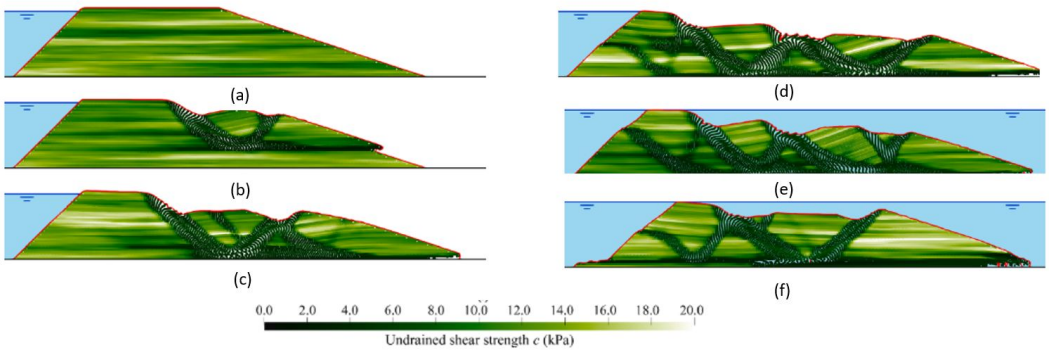


Figure 3.2: Six possible outcomes showing the final deformed position first described by Remmerswaal et al. [52] showing the final deformed position with MPs colored according to the undrained shear strength: (a) stable; (b) shallow initial failure; (c) deep initial failure; (d) retrogressive failure without flooding; (e) retrogressive failure with flooding; (f) horizontal failure triggered by rotational failure.

3.3.2. Identification of failure profiles

The K-MCM algorithm is used to identify how many failure blocks are formed during the simulation, and can therefore classify retrogressive failures. Furthermore, flooding is detected based on the height of the crest. Taken together four out of five failure types are detected.

From the five classes, only horizontal failures are difficult to identify, because it often does not flood and the K-MCM algorithm is not able to identify the form of the failure block. Horizontal failures can appear in both flooded and non-flooded realizations, as previously described. Therefore, a clear characterization is needed to create an algorithm to detect horizontal failures. Remmerswaal et al. [52] describes two different types of horizontal failures seen in the RMPM data, which are:

- Type 1: a horizontal failure in which the width of the sliding dyke stays more or less the same. The height of the sliding failure block decreases slowly, Figure 3.3a.
- Type 2: a horizontal failure triggered by an initial rotational slide, Figure 3.3b.

The characteristic of both types is a horizontal sliding plane to the outer slope, which will lead to an increasing plastic deviatoric strain value along the horizontal plane over time. So, by finding the increased horizontal plane, horizontal failures can be detected. However, since there are two types of horizontal failures in which Type 2 is formed after one or more rotational failure, it is more difficult to detect the horizontal failures. Additionally, the horizontal failure plane can be formed at different depths. The difference between retrogressive failure and retrogressive failure with a horizontal failure is shown in Figure 3.4. The increase of the plastic deviatoric strain is shown with similar patterns, but there is no horizontal plane observed in the retrogressive failure of Figure 3.4b.

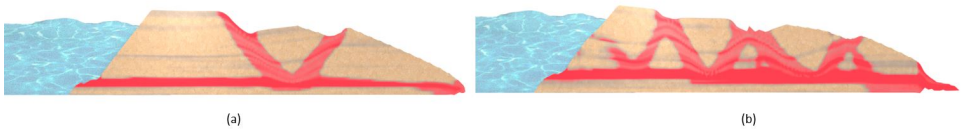


Figure 3.3: Two types of horizontal failures as described in Remmerswaal et al. [52]; (left) Type 1 failure where the moving failure block experience little reduction in undrained shear strength; (right) horizontal failure triggered by an initial failure.

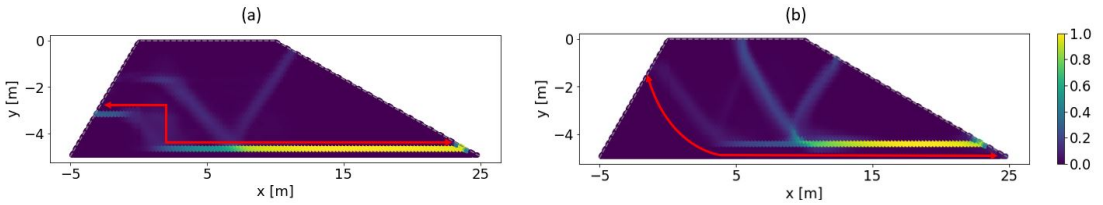


Figure 3.4: Two MPM realizations in a non deformed position with MPs normalized and colored according to plastic deviatoric strain at t_{end}; (a) Horizontal failure type 2; (b) Flooded retrogressive failure.

In order to separate the horizontal failures from the rest, an algorithm is written in which it used the increase of plastic deviatoric strain to separate horizontal failures from other failure types. The algorithm uses the plastic deviatoric strain attribute from the material points at t_{end}. The dyke body at start is rasterized in order to create a heatmap. Only material points with a plastic deviatoric strain value larger than the threshold = 0.5 are registered. The threshold is needed to compensate for the oscillating behavior in the MPM simulations; a value too low will lead to overdetection of horizontal failures and a value too high will lead to underdetection. To detect Type 1 failure, a continuous row of non-zero values must be detected. As for Type 2 failures, two rows of non-zero values together form one row. The full algorithm is in Appendix B.

4

Software selection and implementation

4.1. Requirements for visualization software

The visuals with graphical realism, which this thesis is aiming to create, are often created using a 3D computer graphics software via a *rendering pipeline*. There are a variety of options of software and techniques to create such an image. In order to evaluate the options, the selection criteria for the most fitting techniques and software for visualization of RMPM ensembles must be determined. These criteria are based on design principles [5], and for this thesis the design is based on the following question: *"What is the role of the visual representation of RMPM?"*.

Additionally, one should keep in mind that most geo-engineers, who might create these visualizations, lack an extensive computer graphical background. Therefore, the visualization design methods should be easy to use, and there should be enough guidance for users with little experience. For this reason, an additional guide on how to generate these visualizations is shown in Appendix **??**.

4.1.1. Selecting the target group

In order to answer the question is important to determine for whom an RMPM visual is designed for:

- **Stakeholders**: This group refers to the concerned party affected by the dyke strengthening program, described in Chapter 1, and usually do not know the technical concepts of dyke failure or the numerical methods used to assess them (such as LEM, FEM and MPM). Therefore, visuals should have a more explanatory role. For example, the assessments of WBI indicate that there is 1300 km of dykes that do not comply with the Water Act, which sounds concerning [70]. However, additional reports from WBI and practical cases show that it does not directly imply that civilians are in danger, described in Section 2.1, but instead that often more accurate models, such as RMPM, are required to reduce over conservatism. Additional visuals can explain these concepts efficiently, and creating visuals using accurate data will add trust to the visuals.

- **Geo-engineer**: Geo-engineers are familiar with most of the basic numerical methods. Visuals have the purpose to easily convey and evaluate results [7]. However, RMPM is a relatively new and unknown method [65]. Therefore, visuals of RMPM have both an explanatory and evaluatory role, even for geo-engineers. It may require more than one visual, which is because RMPM consist of multiple realizations. It requires the user to be more creative when designing informative visuals. The visuals should clearly convey the processes within the entire dataset.

There are multiple challenges in making realistic visualizations using (R)MPM, and based on these challenges, a list of requirements is formed to evaluate the visualization software.

4.1.2. Requirements

After settling for whom the visualizations are made for, a first design should be generated to set up a list with requirements for the software. The design of a visualization in general has three main criteria[5]: expressiveness, effectiveness and appropriateness.

Expressive visuals show exactly the information contained in data, nothing more or less. For example, visualization of the subsoil in a detailed manner is unnecessary, because the subsoil is not included in the MPM simulation.

A visual is *effective* when it addresses the cognitive capabilities of the human visual system. For example, the background of a visual is context-related information and is an aid to obtain intuitively interpretable and recognizable visual representation. For dyke slope stability problems, for example, the texture and coloring of the visual are important to understand what the dyke is made of and what happens during dyke failure. Moreover, the environment around the slope is important; for example, the water table is required to understand the location of the inner and outer slope.

Lastly, visuals need to be *appropriate*, which means that the effort of creation of each part of the visual should be equivalent to the added benefit [5]. Therefore, it might, for example, not be needed to simulate the river water in a detailed manner, which costs significantly more computational power compared to using a water texture.

Overview of the list of requirement

Based on literature (Section 2.4.3) and the above-described criteria, which include the needs of the user and receiver, the following list of requirements is formed for choosing the appropriate software for further research, and an overview is given in Table 4.1:

- **Data import:** the software has to be able to import and read the unstructured MP data file containing coordinates, grid type, and attributes for multiple time steps, preferably with limited changes to the MPM code.
- **Existing research on MPM:** this is not a hard requirement, but a preferred requirement. Since using graphics software that already created successful renders of MPM will improve the final outcome.
- **Transformation libraries:** raw data has to be transformed before visualization, so some transformation libraries containing transformation algorithms are required.
- **Modeling**: this feature is needed in order to create more appealing and realistic images by altering and/or adding geometry and features of the mesh.
- **Textures/materials/shaders:** this process materializes the mesh by adding textures and shaders to the object, which are required to convey realistic features of the dyke.
- **Rendering** is the process to generate the image in a graphical realism or non-realistic manner by adding correct lighting for example.
- **Simulation/animation:** Software has to be able to animate the different time steps, and preferably the MPM simulation can be performed from within the software.
- **Python interpreter** is preferred to avoid manual implementation of generating the visuals for multiple realizations.
- **Price:** the cost of the software should be manageable to make it attractive and accessible for future applications in the geo-technical field.
- **Hardware:** software has to be able to be run on laptops/desktops, since most users within the geo-technical field do not have a high-end graphics computer available.

Table 4.1: Requirements the software needs to have based on literature study and design of the dyke.

Scientific requirements	Artistic requirements	Basic software requirements
Data import	Modelling	Python interpreter
Research on MPM	Texturing/materials/shading	Pricing
Transformation libraries	Rendering	Hardware
	Simulation animation	

4.1.3. Selecting software

Based on the different types of software, described in Section 2.4.4, a comparison is made of different types of visualization software is shown in Table 4.2. The table suggests that only coupled software satisfies the list of requirements.

Table 4.2: Comparing different computer graphic software with the requirement list.

	Python Interpreter	Data Import	Research on MPM	Transform ation Libraries	Pricing	Hardware	Modeling	Texturing/ materials/ shading	Rendering	Simulation animation
Blender-OpenVDB	++	++	-	+/-	++	+	++	++	++	+
Blender-VTK	++	++	-	++	++	+	++	++	++	+
Houdini-OpenVDB	++	++	++	+	-	+	++	++	++	++
Amira	+/-	++	-	+	-	++	-	+/-	+	+/-
Paraview	++	++	+	++	++	++	-	-	+	+/-

Note. Each item of each category is rated with positive (++) or (+) and negative (-) valence. A category with a mixed-valence is indicated with the symbol (+/-).

Scientific software, such as ParaView and Amira3D, are not able to handle any modeling and texture requirements, but ParaView has been extensively used in the MPM community. 3D animation software such as Houdini and Blender are originally not created for scientific purposes. However, to meet this additional requirement, libraries (OpenVDB and VTK) can be included. Both Blender and Houdini have integrated OpenVDB into their software. However, the integration of OpenVDB in Blender is not fully developed yet, which makes it more difficult to use and less user-friendly [44].

Blender-VTK (BVTK) and Houdini-OpenVDB both have their advantages and disadvantages. Previous research, described in Section 2.4, has shown that it is possible to transform MPM data to visual representation using Houdini-OpenVDB. However, this previous research does not describe the details behind the visuals using MPM. Blender-VTK has not yet visualized MPM data, and the coupling is also relatively new compared to Houdini-OpenVDB. Nevertheless, the choice of using Blender-VTK is made for the following reasons:

- Blender is an open-source software, making it more accessible. There is a high development rate, which can be seen in the fast developments made in the last year in the field of Blender-VTK [31], which make Blender-VTK a promising tool for the future.

- VTK is the underlying software of ParaView, which is used within the geo-technical RMPM research community. Users are already familiar with the algorithms within the VTK library, and data from MPM can directly be imported into Blender without any conversions. For this reason, there is the confidence that (R)MPM representations in Blender-VTK can be created, even though there is no previous evidence.

- Blender-VTK is free, whereas Houdini-OpenVDB costs more than 6000 dollars per year, limiting the usefulness in geo-technical practice.

- Although there are successful visuals created with VDB using MPM simulation [43] and therefore has a higher chance to create successful renders that suffice the design goals, there are no renders of MPM that have been created with OpenVDB. Instead, only VDB has been used with MPM. As OpenVDB has not yet been fully implemented [43], (R)MPM visualization with Houdini-OpenVDB will not guarantee success.

4.2. Making visuals in Blender

First, one MPM realization will be visualized with Blender-VTK, after which the visualization is extended to multiple realizations. The visualization pipeline is used to transform data to an image, which is based on the theory described by Haber and Mcnabb (1990) [23], it is essentially divided into three parts:

1. Filtering
2. Mapping
3. Rendering

Implementing the visualization pipeline theory to the design goals for the MPM realization gives the design process illustrated in Figure 4.1. The first step of the visualization pipeline, filtering, is completed within the RMPM simulation itself, which is a given dataset and described in Chapter 3. The visualization pipeline is essentially divided into two independent pipelines: (1) *VTK-pipeline*, in which the data is transformed to a mesh, also called the mapping phase. This phase is done within the VTK add-on using the BVTK node system, described in Section 2.4.4. (2) *Rendering pipeline*, in which the mesh, created in the VTK-pipeline, is modeled and textured further in Blender making it graphical realistic, i.e. the rendering phase. Different design goals lead to different rendering pipelines, and as a result, there will be different images. In other words, there is not only one possible design for the rendering pipeline. This chapter shows how to set up the VTK-pipeline and what the options are for the rendering pipeline. After that, the user can alter the pipelines to their own need.

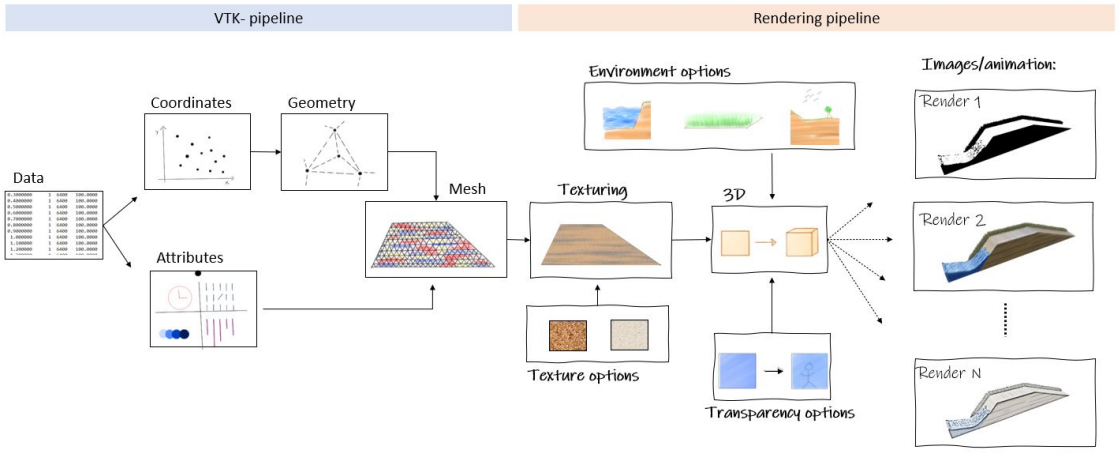


Figure 4.1: A conceptual design of the visualization pipeline showing divided into two pipeline. Left: VTK-pipeline where data is transferred to the mesh. Right: Rendering pipeline where the mesh is materialized.

4.2.1. VTK-pipeline

A basic set up and a brief explanation of two VTK-pipelines from the MPM dataset are shown for a mesh and a particle system. A detailed set up can be found in the guide, which also includes detailed guiding of using the nodes applied to MPM data, additional options to the visualization pipeline, and details of the installation in Windows.

Importing data

The first step of the VTK-pipeline is importing the dataset. This thesis focuses on the import of the MP-file, which carries all information of the material points (Section 3.2). The data file is an unstructured grid file in a *UCD avs* format [4], which includes:

- Material ID
- Coordinates of each MP in x, y, z direction
- Grid type, which in this case is points
- Attributes attached to each MP

Using the VTKucdreader node, designed for this file type, the data can be read and imported. Each date file stores the data for one time step, and each simulation has multiple time steps and therefore multiple files. The whole time sequence can be loaded into Blender allowing the reader to read the time steps accordingly. Standard Blender animation can be used to animate the time steps as discussed further in Section 4.2.2.

Generating a mesh

The next step is mapping the relevant data to appropriate visual variables using structure and type transformations. For unstructured data, it is chosen to use a surface rendering technique (Section 2.4.3) instead of a point rendering technique, because surface rendering requires less computational power and satisfies the current design goals.

Delaunay triangulation is chosen to as the most appropriate surface rendering technique, because the data is unstructured. Other surface rendering techniques like marching cubes may be faster compared to Delaunay triangulation [20]. However, this technique within the VTK library is suited for structured data or 'image data'. Translating the MP dataset will result in extra computational steps, which can be avoided using Delaunay [3].

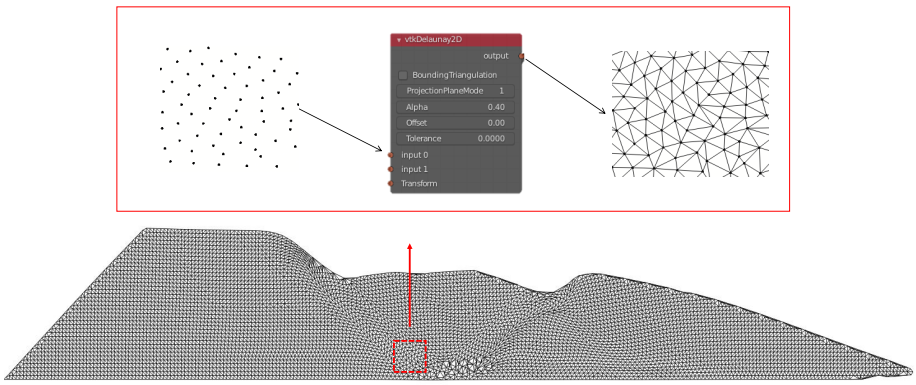


Figure 4.2: Mapping of the Delaunay triangulation for one realization using a VTK class vtkDelaunay2D shown as a node in Blender, with an alpha value 0.4.

The BVTK node vtkDelaunay2D, shown in Figure 4.2, is attached to the input connector: VTKucdreader. As the simulation is a 2D cross-section, a 3D Delaunay triangulation is not required. The Delaunay 2D operation generates a polygonal mesh in the form of triangles. The *alpha* variable is used here to remove unwanted connections in the mesh. It is used so that only edges, vertices and triangles within the alpha radius are outputted. Using a value too small will result in holes in the mesh, and using a value too large will lead to triangulation at certain unwanted locations. A set of alpha values is illustrated in Figure 4.3, which shows that an alpha value between 0.3 and 0.5 gives good meshes. During the research, it shows that using an alpha value between 0.3-0.5 gives plausible results. For simplicity and consistency reasons, it is chosen to use an alpha value of 0.4 for this research.

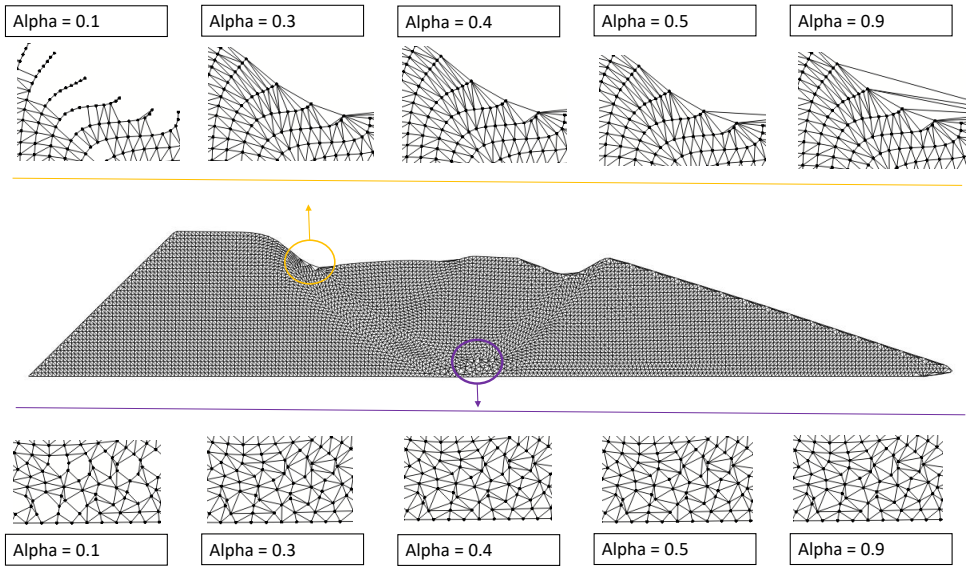


Figure 4.3: Delaunay mesh with alpha values of 0.1, 0.3, 0.4, 0.5, 0.9. Alpha values above 0.5 will result into triangulation at unwanted points, while alpha lower than 0.3 will results into holes in the mesh.

3D Data: The vtkDelaunay3D node is tested by using a self-made 3D dataset, which is created by duplicating the MPs within the MP-file with a depth ($z = 0.5m$) (Appendix B). The usage of the 3D-node is similar to the 2D version with an additional feature for the alpha value. It enables to control which of the output shapes (tetrahedra, triangles, vertices, and/or lines) are influenced by the alpha value [3].

Color mapper

Up till this step, the generated mesh does not yet carry any attribute values, which is necessary when materializing the dyke according to the actual data (Section 2.4) for later use in the rendering pipeline. Therefore, the Color Mapper node maps is used, it adds a selected attribute to a color spectrum and transfers the values on to output mesh and/or glyph. The node's input connector is connected to the VTK-pipeline, and additional VTK nodes can be added to perform basic mathematical operations on the data before color is applied. The color spectrum of the variable is determined by the Color Ramp node connected to the Color Mapper, illustrated in Figure 4.4. A specified value range controls the lookup-table, or one can use an automatic value range within the mapping node. The automatic value range is determined by the minimum and maximum values within the list of attributes. When there is no color ramp node connected to the Color Mapper, an automatic color ramp is used, which is identical to the color range shown in Figure 4.4.

VTK to Blender

The last step within the VTK-pipeline converts the VTK surface mesh along with the attributed values to a Blender mesh. Figure 4.5 shows the converted mesh. Thereafter, the rendering pipeline is attached to the converted VTK-mesh. The complete overview of the visualization pipeline is shown in Figure 4.6 to create a mesh including the filtering steps.

Generating particles

An alternative to converting the material points to a mesh is to assign a particle (a geometry) to each point. This particle VTK-pipeline can generate images to show the material points as spheres or to represent a physical data vector in the form of a glyph [7]. The VTK-pipeline for particles is split into two parts. In the first part, the basic shape of the points/glyphs is defined. In this case, the vtkSphreresource

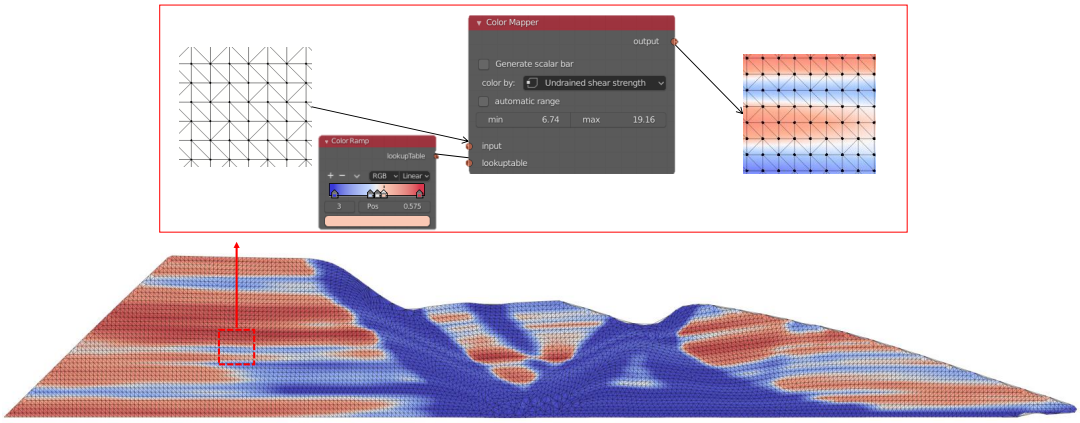


Figure 4.4: A VTK class color mapper is used to attach data values to a MPM mesh. Here the initial undrained shear strength is mapped.

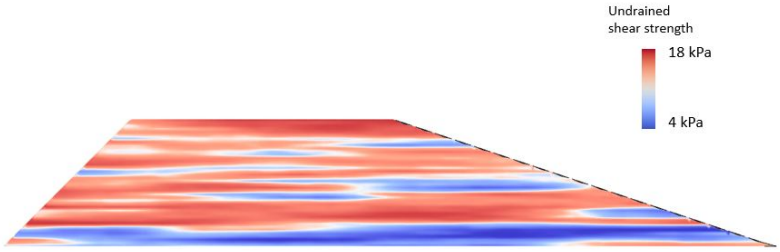


Figure 4.5: The initial undrained shear strength of the dyke; a VTK-object that is created in Blender from the VTK-pipeline

node is used to create small spheres with a certain size and form representing each MP. The output of the node is connected to the VTK to Blender Particles node to create a particle object in Blender. The second part of the pipeline imports the MP data file and combines it with the first part to create a particle system. The complete VTK-pipeline is shown in Figure 4.7.

The VTK to Blender Particles node converts the VTK particles to a Blender particle system using object instancing, which uses little memory and therefore large numbers of points can be visualized at once [31]. The advantage of using the particle system is the possibility of visualizing multiple attributes by adjusting the scale and color of the particles. However, one should note that this node is experimental within Blender and there are known issues with hanging renders.

In the VTK to Blender Particles node, the name of the particle object must be selected. Moreover, the optional functions *Scale Value* and *Color Value* variables of the data file can be coupled to the scale and/or color of the particles. Additionally, the *Direction Vector* function can be used to point glyph objects in the direction of the vector data.

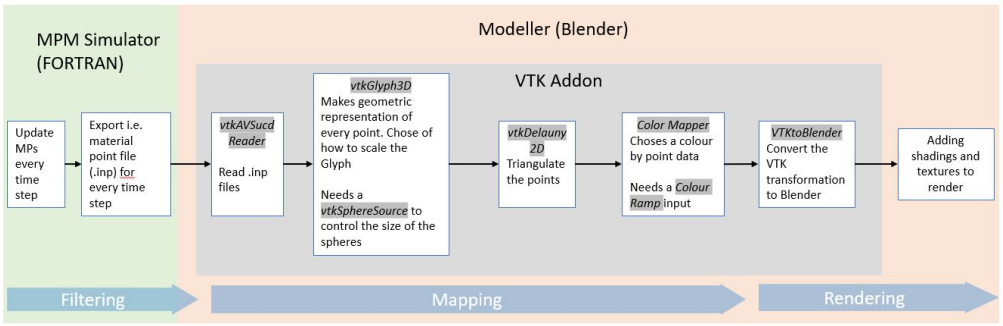


Figure 4.6: Simplified overview of the complete VTK-pipeline to generate a mesh indicated with the basic steps from the pipelines

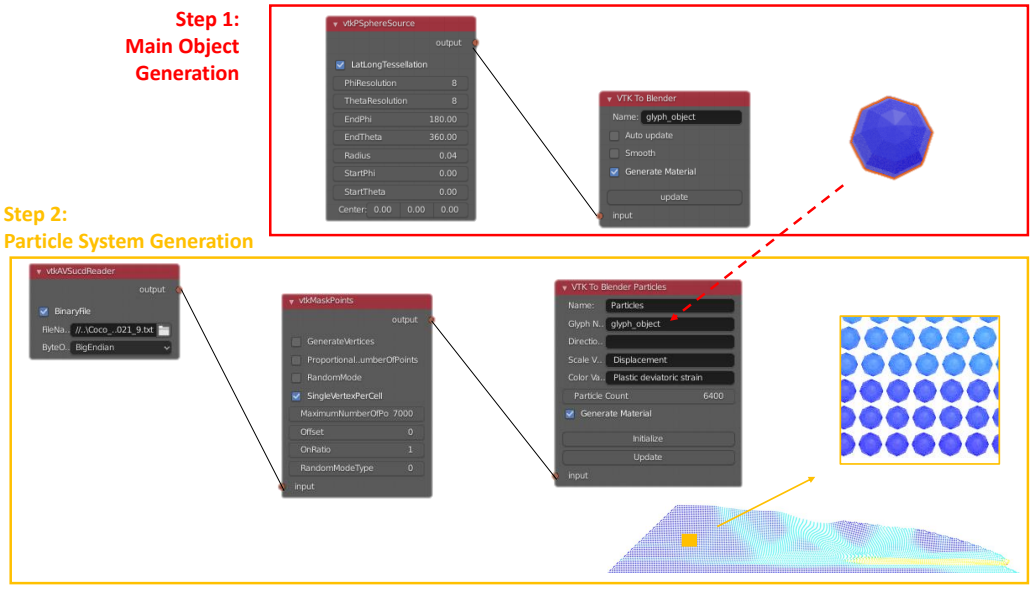


Figure 4.7: VTK-pipeline in Blender software for the particle system; where the first step shows how the main particle is created and the second step shows how to couple individual MPs to the particle.

4.2.2. Rendering pipeline

The output of the VTK-pipelines is a scientific representation of the data, with the color attached as a single value to the vertex. Materializing in Blender starts after converting the VTK-pipeline to a Blender object. It has the goal to make the visualizations more appealing using a material node system, from here on called the *rendering pipeline*. The VTK-pipeline automatically generates a rendering pipeline containing five basic nodes as an output, and forms the basis of further extension of the rendering pipeline, illustrated in Figure 4.8. The purpose of each basic node is as follows:

- Texture coordinate: this node contains color values assigned by the VTK-pipeline.
- Mapping: this node can transform the input vector by applying rotation, scaling and translation.
- Image texture: this node adds an image file to the texture from the UV coordinates. For the general material it takes a RGB value from the image according to the location of the UV map.
- principled BSDF: this node combines multiple properties of materials into one node such as metal, transparency, transmission of light and coloring.
- Material output: this creates the final object from the rendering pipeline.

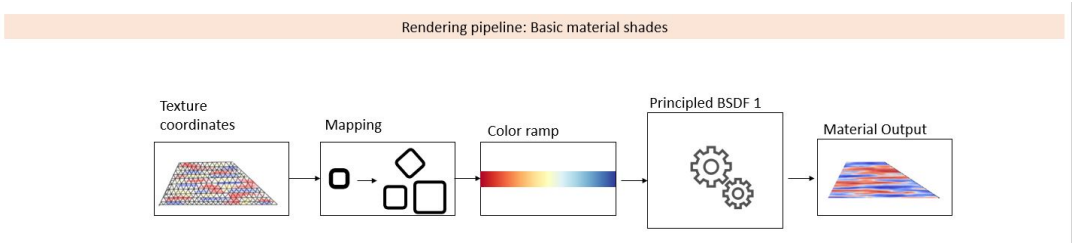


Figure 4.8: Basic rendering pipeline is shown, which is generated automatically after the VTK-pipeline.

Depending on the visualization goal, different rendering pipelines can be created based on the converted object from the VTK-pipeline (Figure 4.5 and 4.7).

Textures

To make the core of the dyke more appealing, different textures can be added according to the colored value from the VTK-mesh. As an example, illustrated in Figure 4.10, the undrained shear strength of the dyke will be represented using two textures: (1) a light sand-colored texture indicating the strong zones; (2) a darker, gray-colored clay texture to indicate the weak zones. The simplified rendering pipeline is shown in Figure 4.9. There are two new nodes added to this pipeline compared to the basic pipeline; (1) the black-and-white color ramp in which the dyke is colored according to the undrained shear strength value given by texture coordinate, i.e. the values provided by VTK, (2) the Mix shader node, to attach the textures to the black and white values from the Color ramp. The ratio between weak and strong material can be chosen according to the user's preference to give a correct representation of the strong and weak zones. After that, each texture is attached to the mix shader and rendered via the material output.

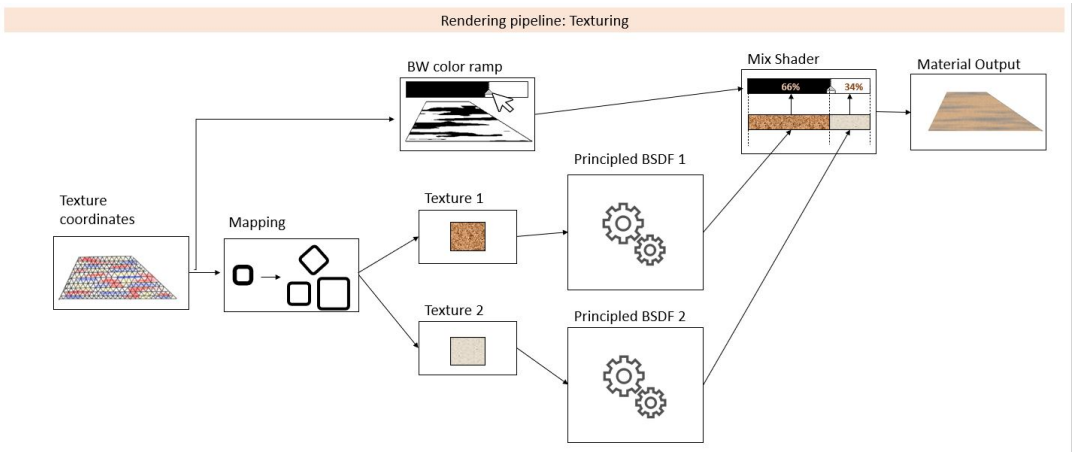


Figure 4.9: Render pipeline for texturing: the mesh created in the VTK-pipeline is transferred to a black-and-white colormap. Thereafter, two different textures are attached to the according to black-and-white values.

It is recommended to be careful when only two textures are used to indicate the weak and strong zones. If the current undrained shear strength is used, the initially weak zones and the softening zones will likely all have the same value and appear to have the same texture, see Figure 4.11a. So it is suggested to add another texture/color to indicate the weakening zones, as shown in Figure 4.11b.

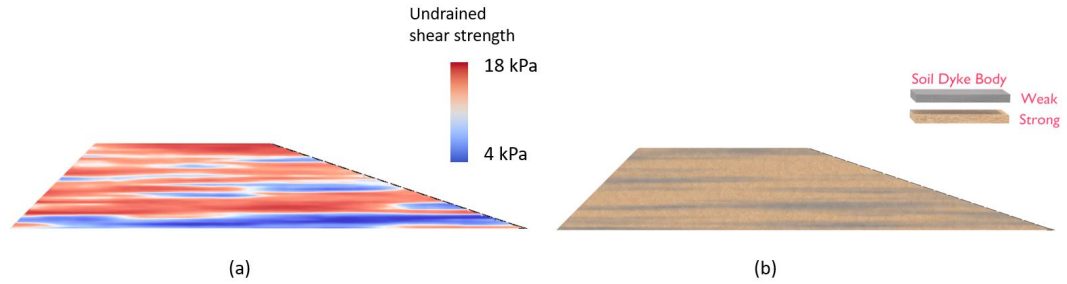


Figure 4.10: One single dyke showing the initial undrained shear strength using (a) scientific visualization by using a color range and (b) artistic visualization by using textures

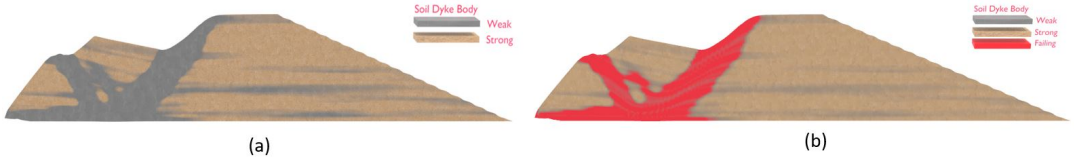


Figure 4.11: Slope failure showing the undrained shear strength using textures (a) the weakening texture gives the illusion that the strength is same everywhere (b) Adding an additional color shows the lowest (critical) undrained shear strength

Three-Dimensional space

Making a visual representation of the dyke in 3D gives the observer the idea of dealing with a slice of the entire dyke with a finite thickness. While, a 2D cross-section can instead seem like an infinite dyke in the third dimension. A 3D slice can be created by replicating the values of the mesh within the data file, as described in Section 4.2.1. Alternatively, the extrusion function in Blender can duplicate the vertices while keeping the new geometry connected to the original vertices (Figure 4.12 (top)). This method does not require any additional modeling in Blender to make it 3D (Figure 4.12 (bottom)).

The Delaunay triangulation results are compared using a 2D and 3D dataset, illustrated in Figure 4.12. The Delaunay triangulation is directly applied to make a 3D mesh, shown in Figure 4.12b, which has a better triangulation with the same alpha value compared to the triangulation in 2D (Figure 4.12a). Additionally, the triangulation from a 3D file gives a finer mesh at the top of the dyke, which helps whenn adding other parameters on top. Although, it costs more time to create the 3D file and run the VTK-pipeline for 3D Delaunay triangulation, it has a better visual representation of the mesh than using the extrusion method.

Transparency

The alpha[1] parameter in the BSDF can be used to control the transparency of the objects (with 0 being fully transparent and 1 being fully opaque). Here the alpha value is used for two purposes. The first purpose of transparency is to remove/reduce parts of dyke, which is a useful tool to get the receiver to focus on a certain aspect. It can, for example, be used to highlight the weakening behavior of the soil throughout time, illustrated in Figure 4.13b, or focus only on initially weak zones in the material. The rendering pipeline for this set up is similar to the texture rendering pipeline; it needs an additional black-and-white Color Ramp node connected to the alpha value in the principled BSDF node, the parts with black color will become transparent. This part of the rendering pipeline is highlighted in Figure 4.13.

The second purpose of transparency is to make the entire dyke cross-section opaque by using one

[1]The alpha-value from Blender material nodes should not be confused with the alpha value of VTK-Delaunay node (Section 4.2.1)

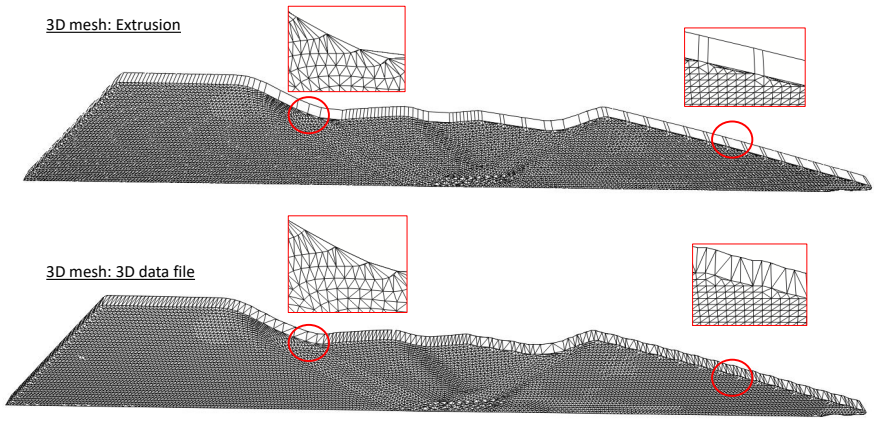


Figure 4.12: Mesh of the slope failure created in two ways; (top) mesh created using extrusion, which emphasizes bad triangulation at the edges of the mesh; (bottom) mesh created using 3D file, which creates triangulation on the top layer showing more accurate triangulation

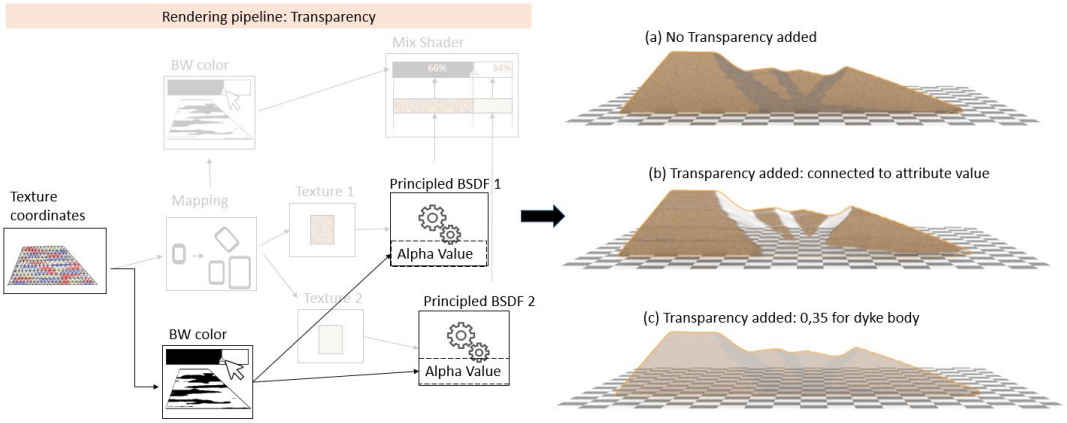


Figure 4.13: Slope failure of the dyke and using transparency to create a different effect. Left: the rendering pipeline to create the middle visualization, only additional nodes are highlighted with the input and output. Right: 3 dyke layers shown; (a) reference with no transparency; (b) transparency is coupled to the undrained shear strength making the weakest part invisible; (c) a transparent layer (alpha =0.35) is added over the whole dyke body.

alpha value for the entire dyke body. It can be used to plot multiple cross-sections after each other. For example, to plot multiple MPM realizations of the same ensemble. As the alpha value is applied to the entire cross-section, it is not needed to extend the pipeline with an additional Color ramp.

Environment

The environment of the dyke consists of the top layer of the dyke, the protected lowland, and the riverside land, which gives additional information to the receiver and an easier link to practicality.

The purpose of representing the riverside is mainly to indicate the location of the water and the water level. Therefore, it is not necessary to create a fluid simulation for this purpose. A simple plane with water texture will suffice to indicate the role of water. However, the fluid simulation can be impressive and may have its use in some circumstances. The top layer of the dyke is a protective layer on top of the dyke body. The layer fractures due to movements during macro-instability and exposes the weakest parts of the soil body, which makes it more vulnerable for erosion and secondary failures such as micro-instability [33]. It is important to visualize the fracturing behavior with scientific accuracy to demonstrate the function of the top layer. A short set-up is shown in Figure 4.15 of how to create this

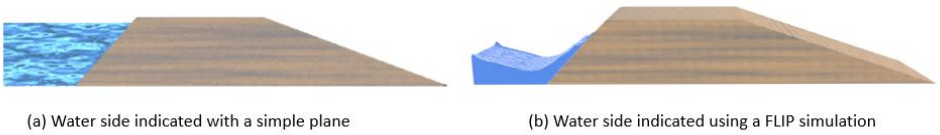


Figure 4.14: Initial dyke showing the river side using two different methods to create.

layer. Individual grass blades are created using a particle system, which is an optional step to make the visual representation more realistic. The duplicated layer is changing according to the parent layer, which is the dyke body, meaning that changing data in the dyke body will automatically update the grass layer, even during animations.

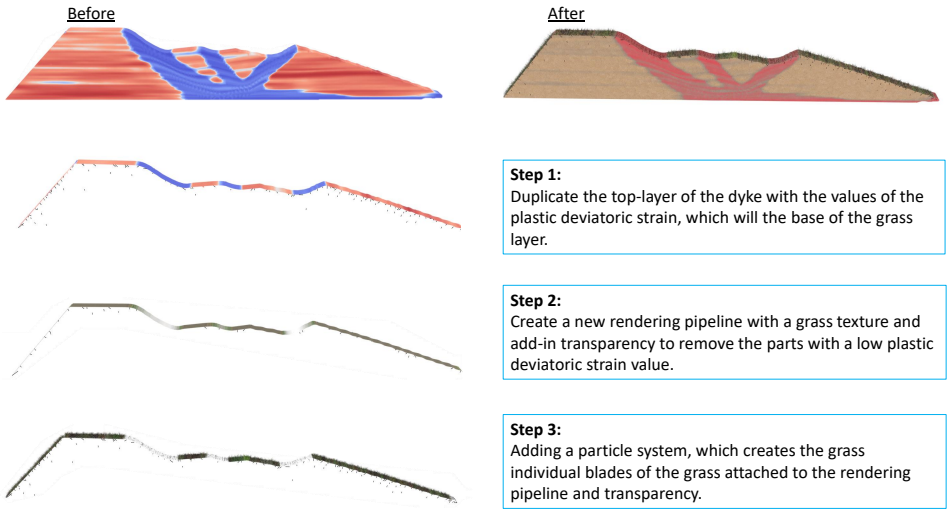


Figure 4.15: Set-up of how the grass top layer is created using three steps. Step 1: duplicates the top of the dyke into a separate layer. Step 2: adding a grass texture and a transparency similar to the rendering pipeline described in figure 4.13. Step 3: Optional step, where a particle system is used to create individual grass blades on top of the dyke for graphical realistic look.

Animation

As the dataset is already a time-series, it is logical to create animations to give the receiver an idea of how the failure proceed through time, illustrated in Figure 4.16. Blender will automatically detect the files within a file directory path in the VTK-pipeline and attach the assigned rendering pipeline. Keeping each realizations' file-name the same and a consecutive number at the end, which represents the different time steps, is recommended to make the process smoother.

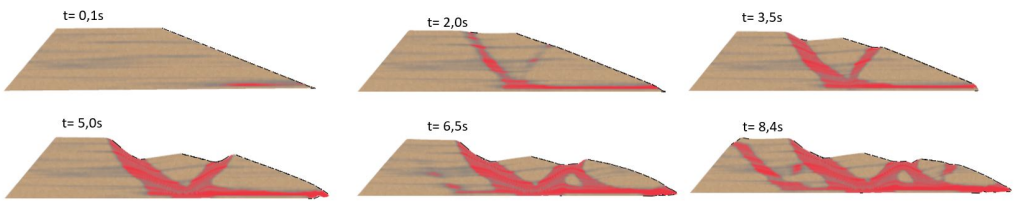


Figure 4.16: Six time steps of one realization from the failure class: flooded retrogressive failure.

4.3. Results in Blender

Using scientific software like ParaView is a popular method for geo-engineers to make visualization because it is easy to use and it does not require computer graphics background to create the visuals. Blender-VTK is a more advanced method in comparison. Both methods can make visualizations with scientific accuracy.

4.3.1. Visuals in three dimensional space

The main difference between the two methods is that using visuals from ParaView constrains the ability to visually communicate with a larger audience, as the representation is often limited to only coloring. In comparison, Blender-VTK creates visuals with a certain graphical realism, which is easier for an audience to link to practicality. Further, it offers a method of making a 2D dataset into a 3D visual, and it gives a first option for visualizing 3D MPM simulations of dyke failure. It shows the receiver that these failures are a slice of the dyke and not the entire dyke. The existing MPM visuals are compared to the visuals created with Blender-VTK in Figure 4.17. In Figure 4.17b, the colorscale is simplified into two types of soil, which will be less overwhelming for the receiver when seeing it for the first time. Moreover, it highlights the parts with failure in a vibrant color, which immediately caught ones attention. As a result, the Blender-VTK shows the value of using graphical realism and three-dimensional space for visual communication.

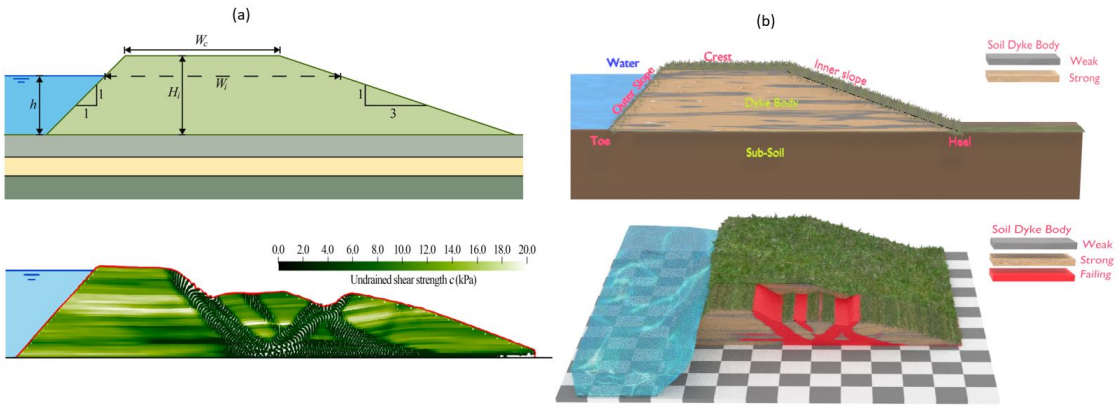


Figure 4.17: Initial dyke and the dyke at the end of the simulation; (a) reference images from Remmerswaal et al. [52]; (b) recreated images using Blender-VTK preserving the scientific accuracy, while making it more appealing and closer to realism.

The different designs of VTK-pipelines and rendering-pipelines result in various examples demonstrating the power of visualization using an (R)MPM dataset for slope stability. Some of these visualizations are more complex to create with more advanced methods and cost more rendering time. An overview is created for all pipelines and compared in Table 4.3, which includes the number material points within the dyke body, the relative time to create and run both the VTK- and rendering-pipeline along with an indication of the difficulty level to create each pipeline.

4.3.2. Visualization of the ensemble

The goal of the thesis is to investigate the possibilities to transform RMPM data to a visual with a certain level of realism. Three features are highlighted, which can be used for visual communication to stakeholders and was not possible to create in the scientific visualization software ParaView:

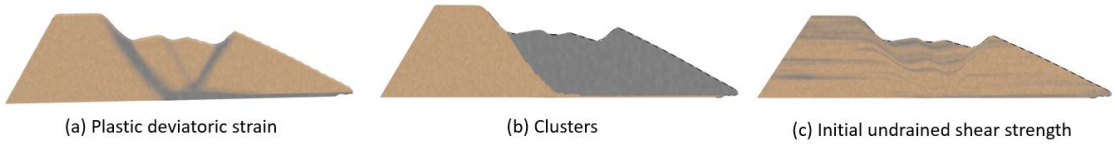
- Recognition of a dyke, by showing water and grass environmental features.
- Explanation of spatial uncertainty, by showing the material strength and the development of different failure paths. Figure 4.19 shows different failure path of five realizations.
- Explanation of the potential damage due to macro-instability, by showing the fractures in the grass cover during animation, which illustrates the reduction of strength in the soil. Figure 4.15 illustrates the fracturing of the grass layer using the undrained shear strength.

Table 4.3: Overview of various examples created in this chapter with particle count, simulation time and an indication of the relative difficulty level to create the pipeline.

	VTK				Blender					
	Dyke body 2D	Dyke body 3D	Bubble Race	Multiple layers	Texturing	Transparency	Grass layer simple	Grass layer advanced	Water Plane	Water FLIP
Figure in thesis	4.10	4.12	4.20	4.19	4.18	4.13	4.15	4.15 step3	4.14a	4.14b
VTK pipeline time /frame	+	+++	++	+*N-layers						
Rendering time/frame					+	+	+	++	+	+++
Particles	6400	12800	6400	6400* N-layer						
Difficulty level	+	++	++	+*N-layers	+	+	+	++	+	+++

Note. Each item of each category is rated with advanced (+++), intermediate (++) , simple (+).

The method shows how to change data to a mesh topology with attribute values attached, illustrated in Figure 4.5. In addition, with the use of Blender, there is more realism in the visuals by adding different textures according to different data values, illustrated in Figure 4.18. Next, it gives a method to not only make illustration but also to make animations, which is valueble for time-dependent data, shown in Figure 4.16.



(a) Plastic deviatoric strain (b) Clusters (c) Initial undrained shear strength

Figure 4.18: Dyke failure showing different attributes values; (a) plastic deviatoric strain (b) clusters (c) Initial undrained shear strength

With an eye to RMPM, the transparency method is able to create visualizations of multiple layers shown in Figure 4.13. Figure 4.19 shows a visualization using multiple dyke layers; each visual representation has a set of five dyke layers and different alpha values. Two challenges arises when creating this image; (1) there is not yet a method to import the layers automatically, meaning that the data files of different realizations needs to be imported manually, which will become an impossible job for RMPM with more than 1000 realizations, (2) a low *alpha* value will result into invisible layers while a high alpha value creates an opaque image blocking most of the simulations.

Another method to visualize the RMPM ensemble is to show an average of an attribute of multiple simulations and show in one visualization by using the particle method described earlier (Section 4.2.1). Figure 4.20, where on the background an dyke is shown and the size and color of the spheres at the front indicates plastic deviatoric strain developed over time, normalized, and added together in a file, in other words, a weighted average deviatoric strain is computed. Each sphere represents one MP, with a larger sphere means a higher amount of plastic deviatoric strain developed in this material point within the ten realizations. The results clearly show a preference for a specific failure path. This method results in much lower computational cost within Blender than plotting all realizations after each other, but it requires a larger amount of pre-processing. Moreover, it is uncertain whether or not the results will be visible for many realizations.

4.4. Discussion and conclusion

A list of requirements for software (Section 4.1) is created based on the needs of the user and the receiver, and Blender-VTK is chosen as the most optimal combination of visualization software and scientific

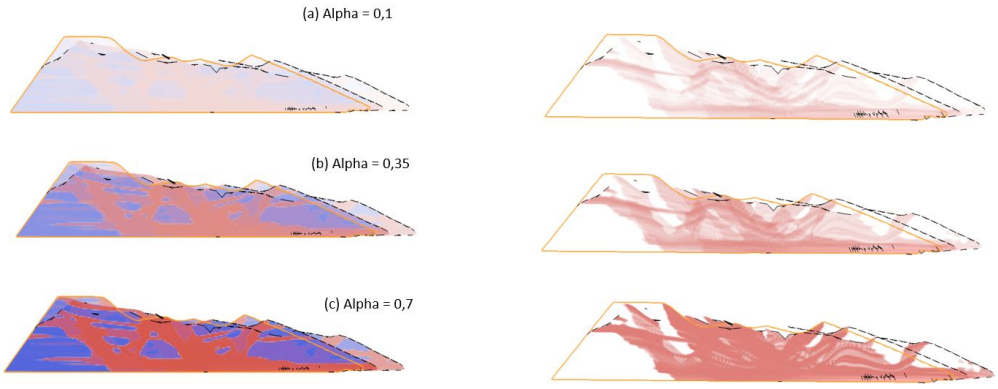


Figure 4.19: A set of five dyke failures with different alpha values with weakening behavior of the dyke shown in red; (a) alpha = 0.1, (b) alpha = 0.35, (c) alpha = 0.7. Right: only shows the weakening behavior

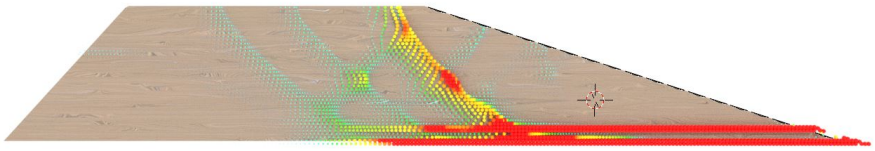


Figure 4.20: Visualization of the dyke with plastic deviatoric strains shown in front indicating the common failure path from 10 MPM realizations

library. Visuals with graphical realism are created to represent the (R)MPM data using a VTK library and the Blender software.

Hansen and Johnson [24] mentioned the difficulty of making computer graphics when lacking knowledge in this field. This method comes with a detailed description on how to make graphical-realistic images specifically for dyke failure targeting geo-engineers as users. This guideline will save time learning VTK and/or Blender, simplifying the creations of Blender VTK-pipelines. As a result, geo-engineers will not be dependent on illustrators to create visuals. Moreover, information will not be lost when a geo-engineer designs visuals compared to transferring the ideas to an illustrator first.

The method shows several compelling results, but there are still improvements to be made. The rendering in Blender is still separated from the VTK-pipeline, and it is therefore more difficult to generate appropriate materials based on VTK variables within Blender. The colormaps need to be consistent in order to make the realizations intuitive as described in Section 2.4.3. As the automatic ranges can update over time, it is not recommended to use the automatic updates of the value range. Because it will make the comparison between realizations and even between time steps not possible.

There are, however, still limitations regarding the final rendering in Blender. Some features, for example, the colors of particles show correctly using the Cycles render engine, while others work better with other engines. It can therefore be hard to reach optimal results.

Creating the environment needs more time comparing to other procedures as it needs more manual processing to create the environment and therefore requires more familiarity with the software, which both cost more time. Moreover, the rendering time to make the advanced grass and water texture is longer as it both has a particle system. However, it is recommended to use the full version to create more appealing visual representations as shown in Figure 4.14,4.15. Table 4.3 shows the relative difficulty of creating these visuals.

Difficulties arise when visualizing RMPM data, as another dimension is added to the raw data since there are multiple realizations. One should ask again: *"What is the role of the visual representation of*

RMPM?". This leads to the following purposes of the visualizations and solutions:

- It serves to explain the role of random fields in RMPM to the receiver by showing the different consequences.
- It serves to give an overview of the ensemble for research purposes, indicating the expected behavior of the various consequences.

The method to create the visuals is developed, but the RMPM dataset is too large and there is no obvious design of how to visualize this dataset. Therefore relations between the realizations data should be explored more extensively. From these relations, clusters can be formed within the ensemble to represent the failure of different possible consequences more efficiently, while maintaining the general behavior of the cluster. The original visualization pipeline is therefore refined according to dos Santos and Brodlie [17] to meet the requirements of higher dimensional visualization problems. The original filtering step from Haber and McNabb [23] is split into two steps: data analysis and filtering. In order to create a visualizations that represents the RMPM ensemble, a data analysis should be conducted.

5

Classification and visualization of ensembles

To classify failure processes of the five failure groups, see Chapter 3, the ensemble is further investigated. The dataset has many attributes describing the simulation, but not all are of interest because it does not give additional information. A wide selection of attributes will be studied in detail, which is based on literature described in Section 2.1.2. The current assessment aims to classify each realization and assess the residual dyke more accurately after initial failure. The residual strength of the dyke is expected to be highly dependent on the initial failure's characteristics because it is expected to be the largest failure and therefore has the largest impact. The following attributes are chosen to investigate based on literature:

1. Displacement of the failure blocks: the displacement of the dyke body can be an important indicator for residual strength.
2. Size of the failure blocks: the size of the failure blocks determine the remaining profile, and can thereby be an indication of the residual strength.
3. Kinematics of the failure blocks: velocity and acceleration can be a valuable indicator of the development of failure.
4. Stresses and strains: stresses and strains of the material points can indicate whether the constitutive model is behaving correctly.
5. Duration of failure: time plays an important role in the behavior of failure, each realizations has a different time span and duration. A simulation with a longer duration is hypothesized to indicate that there is no flooding.
6. Undrained shear strength: the undrained shear strength of the material is an indication of the strength of the dyke, which is important of how the failure planes develops, as it searches for the weakest plane.
7. Plastic deviatoric strain: the plastic deviatoric strain indicates where the boundaries of the cluster are, which can be used when searching for a specific material point at the failure surface.

Even after a selection of attributes, the dataset is still too large and the computational power limits an extensive and quick data analysis of the full ensemble. Therefore, a pre-selection of nine realizations is manually chosen for data analysis. This dataset is assumed to be representative of the ensemble, as it contains at least one realization from each failure class. The goal of the data analysis for the small

dataset is to reduce the final dataset even further. After reduction, correlations between the parameters are identified using parallel coordinates, such that the entire ensemble can be studied at once. Lastly, an overview of the representative parameters and correlations are presented for the complete ensemble, and representative realizations of each failure group are visualized using Blender-VTK.

5.1. Analysis of the subset of the ensemble

5.1.1. Displacement of the failure block

The displacement process of the failure blocks is important in order to understand the failure process and to estimate the resistance of the failure block more accurately, because a failure block can still add resistance force to the overall dyke, and in particular the first failure block because it is estimated to be the largest failure block. Moreover, in current practice, the assessment assumes that a secondary failure only occurs after the initial failure block has reached its equilibrium (Section 2.1. Therefore it is important to understand the displacement of the first failure block.

In practice, the amount of displacement cannot be computed, and must therefore be assumed. The vertical movement of the crest of the failure block is, in the current assessment, often estimated to be 1/2 times the height of the failure circle [11, 33], after which a new equilibrium is assumed. However, this crude assumption is highly uncertain as described in Section 2.1.2; the remaining resistance of the failure block also largely depends on the amount of horizontal movement. This horizontal movement also indicates how much the failure block itself has been weakened. The horizontal and vertical movements are hypothesized to be related to the residual strength, and thus the probability of flooding.

Since, the first failure block is often the largest failure block, it will have the largest influence on the probability of flooding [11]. Moreover, the first failure block is the only block for which the failure surface can be computed accurately with the current most used methods, such as LEM and FEM. Therefore, within this research the focus is on analyzing the first failure blocks, as these observations can help engineers in practice to better estimate the failure profile (based on the position of the failure surface as calculated with LEM or FEM). However, it may also indicate that predictions of the failure process based only on the initial shape will be unreliable.

Displacement of Center of Mass

The displacement of the failure block can be estimated using the adapted K-MCM algorithm described in Section 2.5.1; the algorithm detects a new failure block during the simulation, and the corresponding material points are clustered into the failure blocks. Figure 5.1 illustrates the material points clustered into different failure blocks. The K-MCM algorithm allows calculation of the average displacement of the material points per failure block, shown in Figure 5.1, i.e. the *center of mass* (CoM^i) is tracked, which is computed in each time step with the following formula per failure block:

$$CoM(x,y) = \frac{\sum_{p=1}^{nmp} x_p m_p}{\sum_{p=1}^{nmp} m_p}, \frac{\sum_{p=1}^{nmp} y_p m_p}{\sum_{p=1}^{nmp} m_p} \tag{5.1}$$

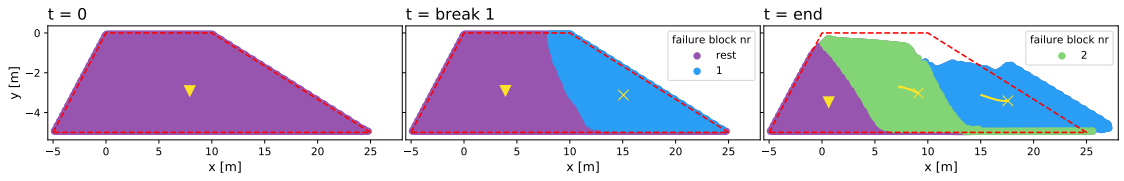


Figure 5.1: Three time steps (t_0, t_{br1}, t_{end}) through an example of retrogressive failure mechanism for one realization showing how the K-MCM clustering algorithm computes the number of failure blocks and their location in each time step. At t_{end}, each failure block shows its CoM (indicated with a yellow cross) along with the traveled path from the start of the failure block. Initial geometry shown in red dotted line.

When the K-MCM algorithm has determined the presence of the initial failure block for the first time during the simulation, the time will be marked as the first break (t_{br1}). t_{end} marks the end of the

simulation, which occurs when the dyke floods, the dyke is stable, or the maximum simulation time of 40 seconds is reached. See Section 3 for further details on the end of the simulation.

For the small subset, CoM^1 is tracked per simulation from t_{br1} to t_{end} shown in Figure 5.2, in which four MPM realizations are highlighted in different colors, along with its corresponding failure profiles of the realizations at t_{end}. For these examples, tracking CoM^1 is a good indication of the failure extend, as small displacement paths indicate limited deformation of the dyke (Figure 5.2c), which results into a stabilization and no flooding occurs.

A larger horizontal displacement appears to indicate a larger number of failure blocks or the occurrence of a horizontal failure (Figure 5.2 b and c). The large horizontal movement is either caused by follow-up failure blocks pushing from behind, or the water pressure pushing the entire dyke. So, horizontal displacement can be a good indicator for residual dyke resistance.

Comparing realizations c and d in Figure 5.2, a higher starting point CoM_y^1 (at t_{br1}) indicates a shallower failure block, while a lower starting point indicates a deeper failure block. Furthermore, the x-position of CoM^1, when looking at realization a and b in Figure 5.2, correlates with the width of the failure block. The correlation indicates that the size of the failure block can be back-calculated from CoM_x^1 and CoM_y^1. This back-calculation is further investigated in Section 5.1.2.

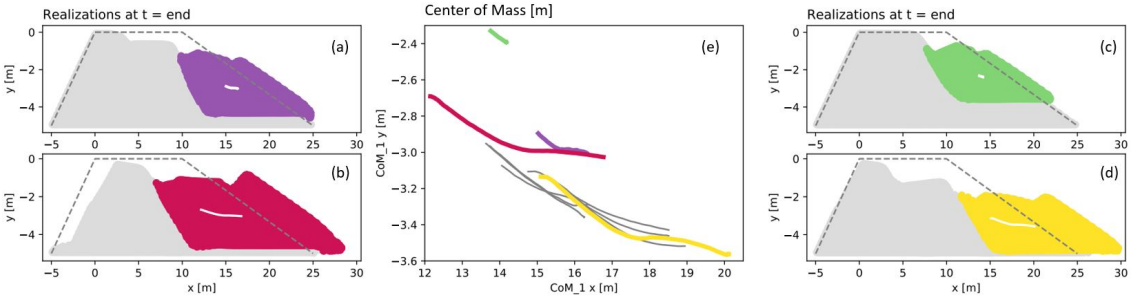


Figure 5.2: Displacement of the center of mass of the first failure block from t_{br1} to end of all realizations in the small dataset. Four realizations are highlighted in color within the graph and the failure profiles are shown in the figures next to it.

Crest settlement

While the horizontal deformation and the size of the failure surface are accurately correlated with the CoM^1, the crest settlement is underestimated by CoM^1. For example, in realization d, shown in Figure 5.2, the vertical settlement of almost 2 m equates to a small vertical displacement of CoM_y^1 of roughly 0.5 m. The horizontal movement of the block is represented accurately using CoM_x^1, while the significant vertical movement of dyke crest results in almost zero CoM_y^1 displacement. Therefore, the average vertical displacement of the crest (per failure block) is recorded over time, as illustrated in Figure 5.3. The crest settlement calculated with CoM_y^1 is compared to the newly introduced crest tracking method in Figure 5.4, which demonstrates in the dyke visualization that tracking the crest indeed gives better results. Therefore, the horizontal displacement of the failure block is tracked using CoM_x^1 and the vertical displacement of the failure block is tracked with the crest tracking method.

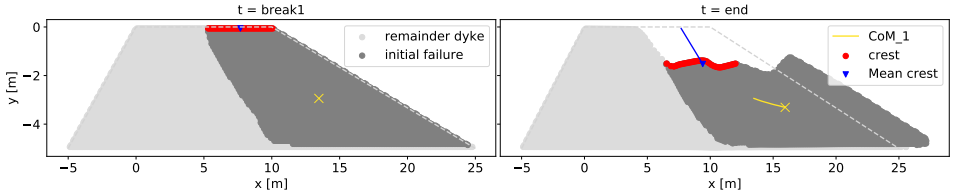


Figure 5.3: Two time steps (t_{br1}, t_{end}) of one example realization illustrating the average path of the crest and CoM for the first failure block. Initial geometry shown in gray dotted line.

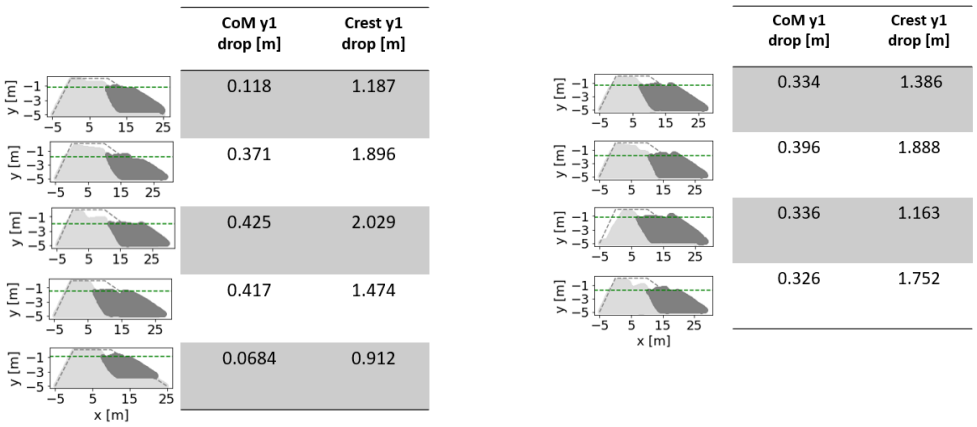


Figure 5.4: Difference in crest settlement calculation using the CoM and the crest tracking method of the small ensemble, calculation with the crest tracking method indicated with the green dotted line.

5.1.2. Size of the failure blocks

The maximum and minimum position of the failure block can be determined using the K-MCM clustering algorithm, which gives the height and width of the failure block. The number of material points within one cluster gives the total mass/volume of the material. However, this can be error-prone, as one material point can significantly impact the outcome.

The calculated position of CoM^1 at t_{br1}, as presented in Section 5.1.1, indicated that CoM^1 is hypothesized to be able to back-calculate the location and the area of the failure block. So, together with the assumption of the form of the initial failure block, i.e. rotational (Section 2.1), the height and width are estimated from CoM^1. The estimated area of the failure block will be called the theoretical slip surface (A_{theo1}) from here on.

The center of mass can be calculated using vector addition of a weighted position vector [25]:

$$CoM_1^x = \frac{m_1 \cdot x_1 + m_2 \cdot x_2 + ... + m_n \cdot x_n}{m_1 + m_2 + ... + m_n} \tag{5.2}$$

$$CoM_1^y = \frac{m_1 \cdot y_1 + m_2 \cdot y_2 + ... + m_n \cdot y_n}{m_1 + m_2 + ... + m_n} \tag{5.3}$$

The boundaries of the theoretical slip circle are called $bound_x^1$ and $bound_y^1$ respectively, which are bounded to the maximum height and width of the slope design. The area of the theoretical slip surface is divided into three parts, shown in Figure 5.5. A1, A2, and A3 are assumed to be a rectangle, triangle and parabola, respectively. Each area is computed using Eqs. (5.4) - (5.6).

The equations are dependent on three unknowns: the $bound_x^1$, $bound_y^1$ and b, which are the boundaries of the theoretical slip surface and the width of the parabola. The width of the parabola is set to 3m for a first estimation. The equations (5.7) and (5.8) are solved using Python (version 3.6.6).

$$A_1 = (10 - bound_x^1 - b)(-bound_y^1) \tag{5.4}$$

$$A_2 = 3/2(bound_y^1)^2 \tag{5.5}$$

$$A_3 = 2/3b(-bound_y^1) \tag{5.6}$$

$$CoM_1^x = \frac{A_1(5 - 1/2bound_x^1 - 1/2b) + A_2(10 - bound_y^1) + A_3(bound_x^1 + 5/8b)}{A_1 + A_2 + A_3} \tag{5.7}$$

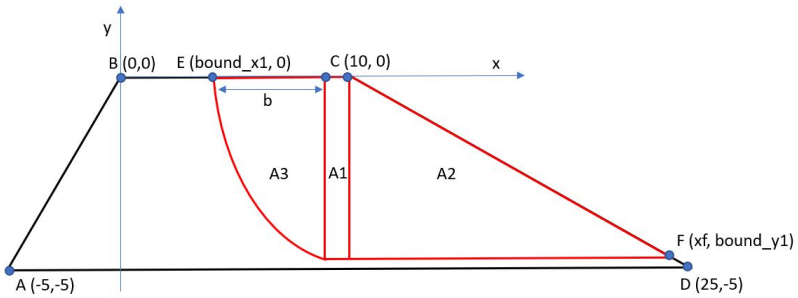


Figure 5.5: Simplified layout of the dyke and a rotational failure block divided in three parts A1, A2 and A3, coordinates A-D are at the border of the dyke. Coordinate E is the intersection with of the failure block with the crest and point F is the intersection of the failure block with the inner-slope.

$$CoM_1^y = \frac{A_1(-1/2 bound_y^1) + A_2(-2/3 bound_y^1) + A_3(-2/5 bound_y^1)}{A_1 + A_2 + A_3} \tag{5.8}$$

A visual inspection of the dataset and the theoretical in Figure 5.6 shows that the accuracy of the computation of the theoretical boundaries is acceptable. Especially the horizontal boundary ($bound_y$) is computed accurately, while the vertical boundary ($bound_x$) often cuts through the parabola. It suggests that the hypothesis of using CoM to calculate the location and size of the failure block gives valuable results and should be further investigated for the entire ensemble.

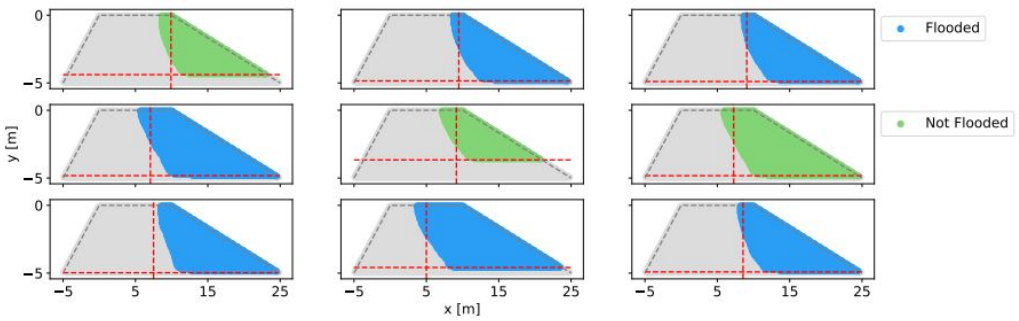


Figure 5.6: Nine realizations showing the size initial failure along with the theoretical boundaries with the red dotted lines. Flooded simulations indicated with blue and not-flooded simulations indicated with green.

5.1.3. Kinematics

The average velocity of the failure blocks is analyzed. In Figure 5.7 the horizontal velocity of the first failure block is shown for the small subset.

The jump in each simulation occurs when the K-MCM algorithm forms the initial failure block. Before the jump, the entire dyke is used to calculate the velocity of the first cluster. When the failure block is created, the average velocity of the first cluster suddenly increased since the roughly stable material is no longer included in the first cluster.

The large oscillations, shown in some simulations, are caused by the creation of subsequent failure blocks, which push the first failure block, thereby again increasing the velocity. Flooded realization end immediately, and therefore movement can still be present in the initial failure block, while the velocity decreases over time in stable realizations. Two realizations have an oscillating behavior till the end of the simulation ($t_{max} = 40s$) due to inaccuracies within MPM computations, leading to an oscillating behavior. The low amount of damping slowly decreases the amplitude of these oscillations. Due to the

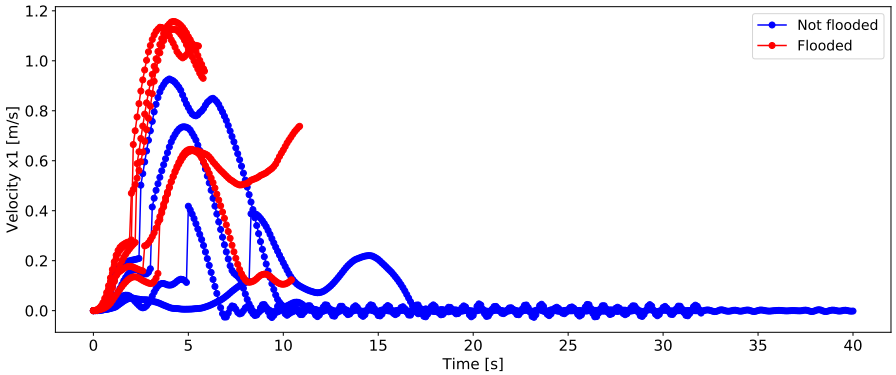


Figure 5.7: Averaged velocity in x-direction of the first failure block from t_{start} to t_{end} of all realizations in the small dataset. Flooded simulations are indicated with red and not-flooded in blue.

oscillations, some simulations cannot meet the requirement of a stable realization, and therefore reach 40 seconds of simulation time.

The small dataset shows that it is more useful for velocity to investigate the full path of the diagram instead of one single point (in time). Because one single point in time is not cannot give information of the whole path. As this is not feasible for the full ensemble, the velocity (and the acceleration, for which the same behavior was found) will not be further investigated.

5.1.4. Stresses and strains

Stress-strain diagrams (or p-q diagrams) are, unlike the previous analyses in Section 5.1.1 and 5.1.3, not interesting when averaged over a complete failure block as different material points in a failure block may have completely different stress/strain behavior. Therefore, in this preliminary study, six points are chosen that should represent the various behaviors. The p-q diagrams verifies whether the constitutive model behind works correctly. For the entire ensemble, the material points of interest have to be automatically detected as their location depends on the failure surface's (unknown) location.

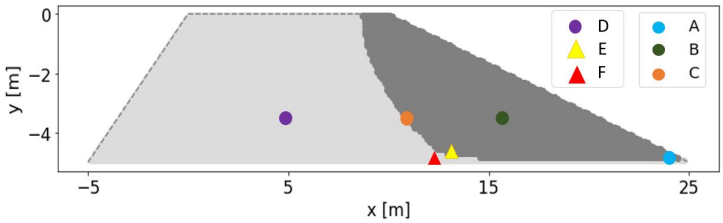


Figure 5.8: Six material points selected to plot the stresses in p-q diagram.

The material points A to D are chosen to analyze key positions in the dyke body; see Figure 5.8. Material points A, B, C and D are located at the toe of the dyke, the middle of the failure block, the edge of the failure surface, and the middle of the residual dyke body, respectively. The p-q stress paths of material point A-D are shown in Figure 5.9 along with the initial and residual position of the yield surface for a Von Mises material (F^0_{vm}, F^{res}_{vm}), which are computed using the following formula:

$$F^0_{vm} = C_i * \sqrt{3} \tag{5.9}$$

and

$$F^{res}_{vm} = C_r * \sqrt{3} \tag{5.10}$$

The development of the undrained shear strength is plotted as well. In the Von Mises model, the deviatoric stress governs the plastic behavior, as the model is mean stress independent. Figure 5.9 A and C

show that the deviatoric stress increases at these points until the yield surface is reached. Afterwards, softening causes the undrained shear strength to decreases until it reaches its minimum. Due to the softening, the yield surface shrinks, and the deviatoric stress changes accordingly. On the other hand, points B and D have a constant undrained shear strength, as the yield strength is not reached, i.e. yielding did not occur.

A clear stress/strain path cannot be observed, especially before yielding the stress patterns are chaotic. So, this indicates that even though GIMP and double mapping (together called DM-G [21]) have been used to reduce the oscillations, the individual material points still oscillate. The dynamic nature of the simulations may partially cause this, but some oscillations are likely to be nonphysical. As previously discussed in Section 2.2.2, the displacement patterns are realistic even with these nonphysical oscillations present.

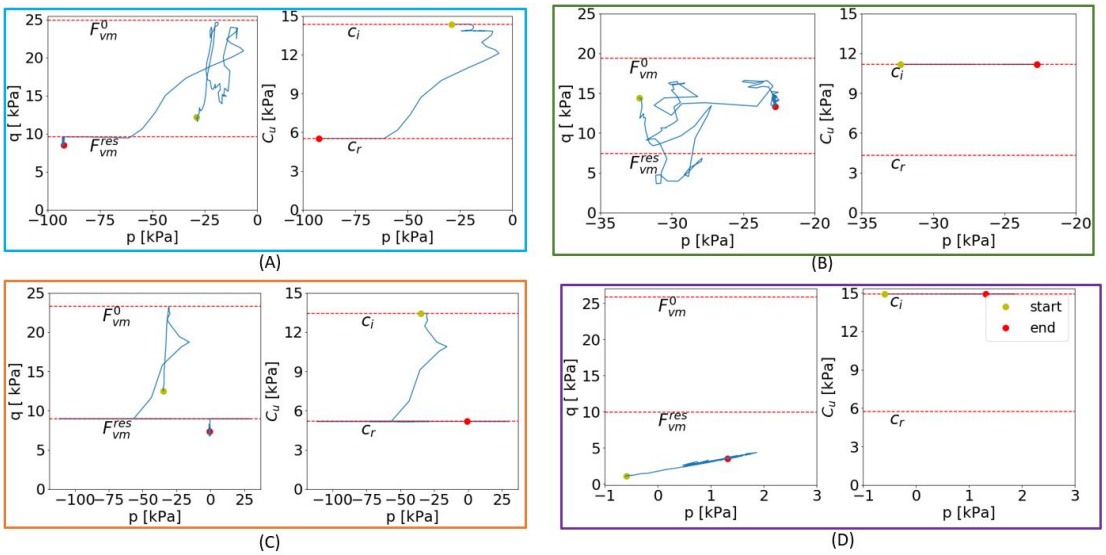


Figure 5.9: Four p-q diagrams for material points A, B, C, D.

The nonphysical oscillations can more clearly be observed in points E and F in Figure 5.10, which have been placed at opposite sides of the failure surface, see Figure 5.8. These p-q diagrams show an unnaturally high increase and decrease in mean stress at opposite sides of the failure surface. This extreme and contradictory behavior is caused by volumetric locking [13]. Andersen et al. [7] shows a promising post-processing method to visualize stresses more accurately by smoothening, i.e. averaging. In other words, it averages the stresses over the background grid elements, which is done by mapping back the material point stress to the grid node, which happens as a post-processing procedure and therefore does not influence the computation itself. Another method to reduce stress oscillations is proposed by Coombs et al. [13], where an extension is proposed on existing implicit MPM for any constitutive models. This method suggests taking the geometric center of the MPs in each element rather than using the center of the background grid cell.

Here, a new post-processing method is introduced and investigated to reduce stress oscillation: the average is taken over the material points within a defined radius around the point selected for the stress-strain diagram, illustrated at the bottom of Figure 5.11. Comparing this to the post-processing method of anderson et al. [7], the method in this thesis take the average over MPs rather than grid elements. Each time step, the material points within this radius may change, and different points may therefore be used to compute the average within each step. Two different radii have been used to investigate the effect of the smoothening algorithm as shown in Figure 5.11a and b. The radii are chosen such that eight and four material points lie within the radius in the initial condition.

The method reduces the (extreme) increase/decrease of the mean stress, and therefore the stress becomes more realistic for large radii when the residual undrained shear is reached. However, the p-q diagrams still show oscillating behavior at the end of the simulation. Moreover, comparing Figure 5.10E

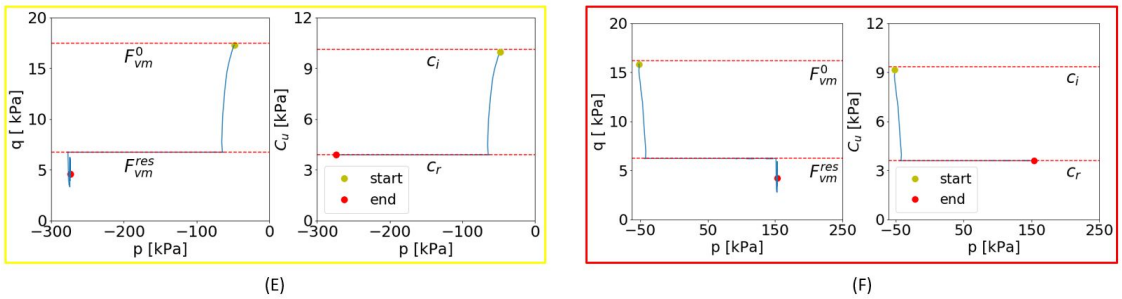


Figure 5.10: Two p-q diagrams for material points (point E and F) on the edge of the failure block showing extreme behavior in contrary direction.

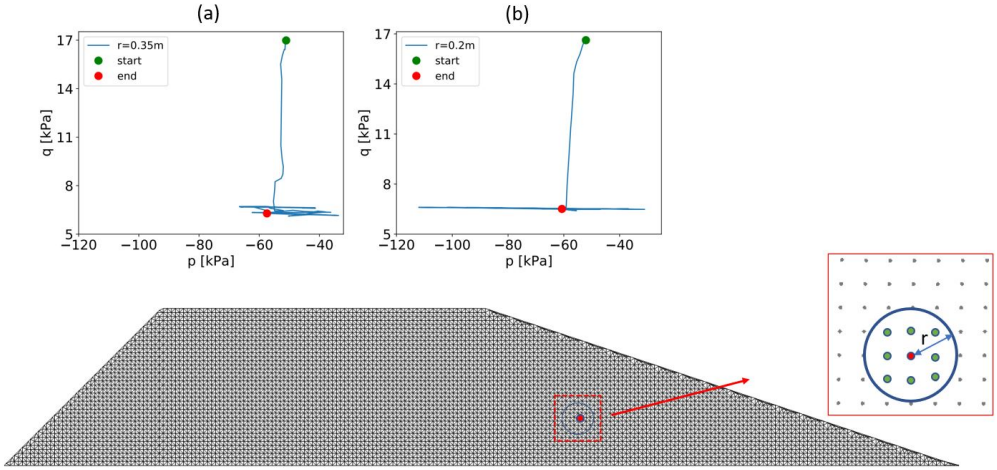


Figure 5.11: Smoothened p-q diagram by averaging the values of multiple MPs which falls within the radius around point E. Two different radii are tested; (a) 0.35m; (b) 0.2m.

and 5.11 before the residual strength is reached, shows that the behavior up to the yield stress changes when the radius is increased, and becomes more chaotic.

Classification according to the stress-strain paths is difficult due to; (1) the unclear stress-strain paths, (2) the volumetric locking, which is not solved by the averaging technique, and (3) the difficulty of automatically selecting the interesting points for stress-strain paths. Therefore, p-q diagrams will not be further investigated within this thesis but suggested as future work.

5.1.5. Time

Comparing the duration and different moments in time of the realizations can, for example, help identify whether flooded realizations failed quicker. Time is one of the additional new features within RMPM, as the whole time span from start to flooding was not captured in FEM. The time feature is not much investigated in previous research. There are a few moments in time, which are of importance to set focus. The first is when the first failure block is formed ($t = br_1$), since it can indicate the strength of the dyke, since a weaker dyke probably fails quicker than a stronger dyke. Moreover, the required time to reach failure could be computed with a FEM calculation. So, it would be advantageous to be able to relate the probability of flooding to the moment of the first failure block occurring (Section 5.1.3).

Additionally, to describe the failure process, the time at which subsequent failure blocks is formed is detected and the moment when the simulation ends, i.e. the duration of the simulation, is of interest to analyze. All of these timestamps may be related to the probability of flooding [52]. Each time step where a new failure block breaks off is captured in a continuous time-scale, shown in Figure 5.12. In this figure, the average time until the first failure blocks is larger for the simulation, which do not result into flooding. This finding should be confirmed for the entire ensemble. Also, the period between the first

and follow-up failure blocks increases for realizations that prevent flooding. These results suggest that the different time steps can be a good indication of the overall strength of the dyke. Furthermore, the area of the initial failure blocks at t_{br1} is slightly negatively correlated with the required time to reach failure, i.e. large failures tend to form earlier. These results indicate that, the probability of flooding seems to be correlated with the duration, and the time of forming the initial failure. Simulations with a longer duration and/or later break off point do not lead to flooding. However, further investigation of the ensemble is required to confirm these correlations.

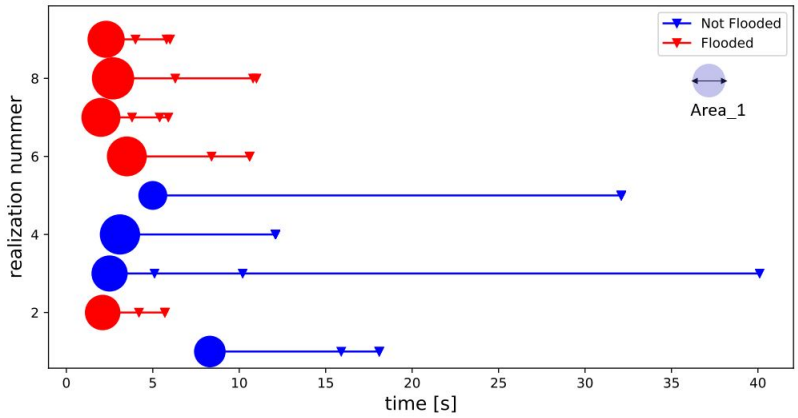


Figure 5.12: A continuous time-scale with the the small dataset is presented starting from t_{br1}. The triangular point indicates the following breaks detected by the K-MCM algorithm, the color indicates whether the simulation is flooded or not and the radius of the circle at the start of each simulation represents the area of the initial failure block at t_{br1}.

The relation between the traveled distance of the initial failure block and the duration of the simulation is investigated in Figure 5.13 using the radius of the circle and a continuous timescale, respectively. One would expect that over a longer duration results larger displacement can take place. However, this is not seen in Figure 5.13, which indicates that longer simulations usually displace less. The displacement does appear to be correlated to the probability of flooding, as large displacement results in flooding while small displacement does not.

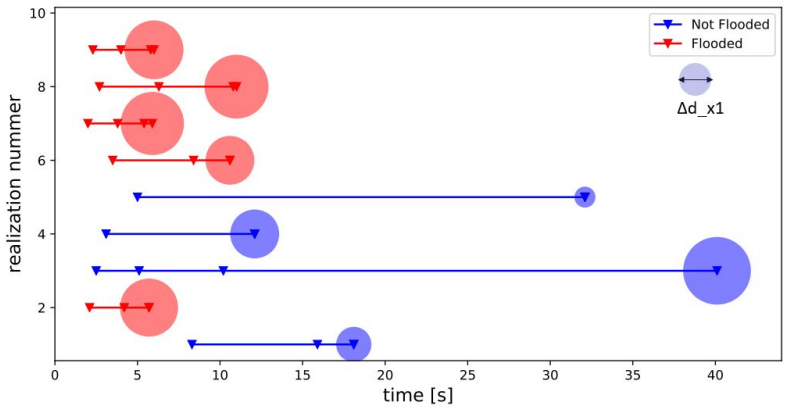


Figure 5.13: A continuous time-scale with the small dataset is presented starting from t_{br1}. The triangular point indicates the following breaks detected by the K-MCM algorithm, the color indicates whether the simulation is flooded or not and the radius of the circle at the start of each simulation represents the traveled of the initial failure block at t_{end}, which shows that the traveled distance is not related to the duration of the simulation.

5.1.6. Hypothesis/conclusion

The analysis of the subset made it possible to do a data analysis over a set of parameters of interest from literature in a time efficient manner. The goal of this analysis is to reduce the dataset such that further analysis to find correlations is possible for the entire ensemble in a time efficient manner. Moreover, the data analysis of the subset gives a preliminary insight of the correlations between different attributes and how it influence the probability of flooding, which forms the base for further analysis.

A summary of the attributes included in the reduced dataset is shown in table 5.1 and 5.2. The mp-file output is reduced most as the data per file is reduced and also the number of files, i.e. only two time steps (t_0 and t_{end}) per realizations is outputted. For the aim of this data analysis, mesh files are not needed. As mentioned at the start of the chapter, the results in this section may be heavily influenced by the selection of the subset of realizations. Therefore, in the next section, parallel coordinates are introduced to analyze the whole ensemble and to justify the findings.

Table 5.1: An overview of the attributes of the MP-file of the reduced ensemble, each attribute has an output per material point for 2 time steps; t_0 and t_{end}.

2 MP-Files (t_0 and t_{end}), attributes for each MP:		
x, y, z coordinates [m]	Plastic deviatoric strain invariant [-]	Initial undrained shear strength [kPa]
Displacement [m]	Deviatoric stress [kPa]	Undrained shear strength [kPa]
	Mean stress [kPa]	Cluster [-]

Table 5.2: An overview of the attributes of the cluster-file of the reduced ensemble. Each attribute has an output per cluster per time step.

Cluster-file, attributes per cluster per timestep		
Cluster number [-]	CoM x, y [m]	Mean Velocity x, y [m/s]
Number of material points [-]	Min position x, y [m]	Mean Acceleration x, y [m/s2]
Cluster volume [m3]	Max position x, y [m]	Time [s]

5.2. Analysis of the ensemble using parallel coordinates

To further interpret the set of attributes for the entire ensemble, parallel coordinates plots are created as described in Section 2.5.2 because this form of data visualization made it possible to show multiple attributes within one graph. From these plots, assumptions for the pre-selected dataset can be justified, and correlations for further research can be identified such as the behavior of different failure profiles. Although many variables can be visualized in a single parallel coordinates plot, it is impractical to plot all possible variables simultaneously. Therefore, two sets of attributes are chosen which describe specific parts of the simulations: 1) the shape and size of the first failure block, which may be computed immediately after the block forms i.e. at t_{br1}, and 2) the deformation process of the first failure block. The following approach, also described in Section 2.5.2, is used to investigate the two sets:

1. Select the appropriate attributes.
2. Determine the order of the axis of the parallel coordinates to clear up clutter.
3. Use brushing to classify the different poly lines.
4. Further investigate patterns in the parallel coordinates using histograms.

The visualizations are made using the Plotly package in Jupyter notebook for Python v3.6.6.

5.2.1. Set 1: Shape and size of first failure block

The first set of attributes (Set 1) is created to verify the back-calculation of the size and shape of the failure block based on CoM^1, see Section 5.1.1. Moreover, Set 1 is to investigate the correlation between the size and shape of the first failure block with the remainder of the failure process.Figure 5.14 presents the polyline of one realization as they appear in the dyke geometry. The following attributes are used:

- The total number of failure blocks.
- The size of the initial failure block (A^1), i.e. number of material points in this block.
- The center of mass of the initial failure block at t_{br1} in horizontal direction (CoM_x^1).
- The center of mass of the initial failure block at t_{br1} in vertical direction (CoM_y^1).
- The left boundary of the initial failure block ($bound_x^1$)
- The lower boundary of the initial failure block ($bound_y^1$).
- Did the simulation result in dyke flooding?

The challenge of plotting the ensemble in parallel coordinates arises not only from choosing the correct attributes but also from placing the axis in a logical order, described in Section 2.5.2 and shown in Figure 5.15. The pre-analysis contributes to choosing which attribute to visualize, it is chosen to visualize the number of clusters per realization and whether it floods because it is related to the failure profile. Additionally, it is chosen to visualize the size of the failure block as the failure area is correlated to all the parameters it is handy if brushing can be applied using this parameter. Thereafter, logically ordering the axis will reveal the correlation between data and the clusters of data in parallel coordinates. Each realization is colored according to the size of the initial failure block, shown in Figure 5.16, because all attributes are correlated to the area of the failure block. The polyline color helps to find a proper order of the axis and shows the dependency of data. For example, Figure 5.15a shows a linear correlation between $bound_x^1$ and CoM_y^1, the colors of the lines changes gradually, and the lines at the lower part of the graph show a similar slope.

The order of axes is chosen based on the dependency between data, Figure 5.15a emphasize the correlation between CoM_y^1 and $bound_x^1$, which is less logical to place next to each other because it is not related to one another as described in Eq.(5.8). Therefore, CoM_x^1 and CoM_y^1 are switched in places shown in Figure 5.15b, it verifies the linear correlation between CoM_x^1 and $bound_x^1$, and between CoM_y^1 and $bound_y^1$. However, the correlation is less obvious between CoM_y^1 and $bound_y^1$, which could be due to the failure profiles' different behavior and leading to mixed correlations.

The arrangement of numbers within each axis should be logical for the receiver to interpret. Therefore the axis of the $bound_y^1$ and CoM_y^1 of Figure 5.15a are flipped, shown in Figure 5.15c, such that the height of the axis corresponds with the depth of the dyke, which is a more logical perception to the receiver, i.e. a deep failure block corresponds to a low number of $bound_y^1$ and shows in the bottom part

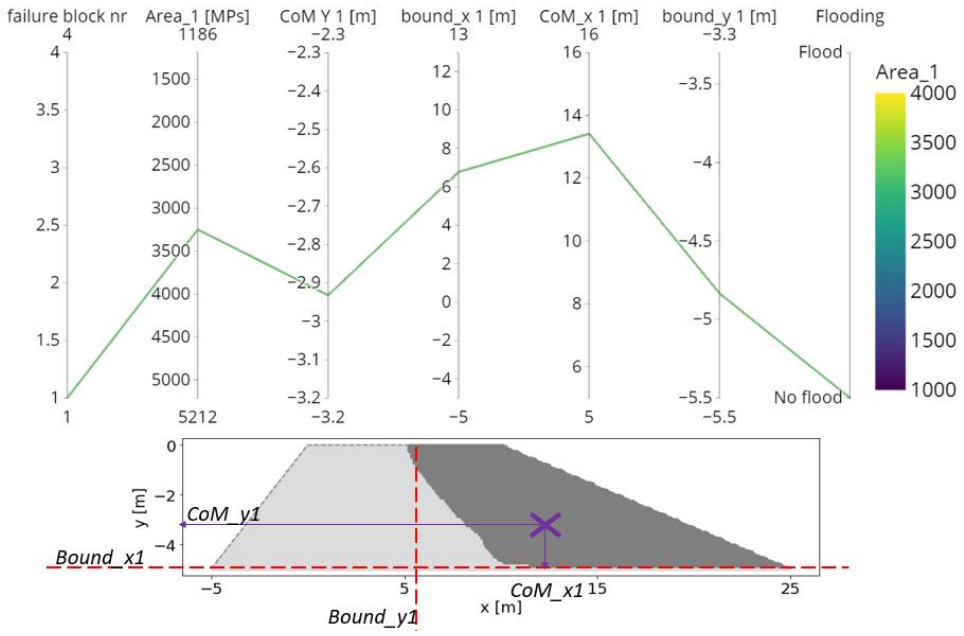
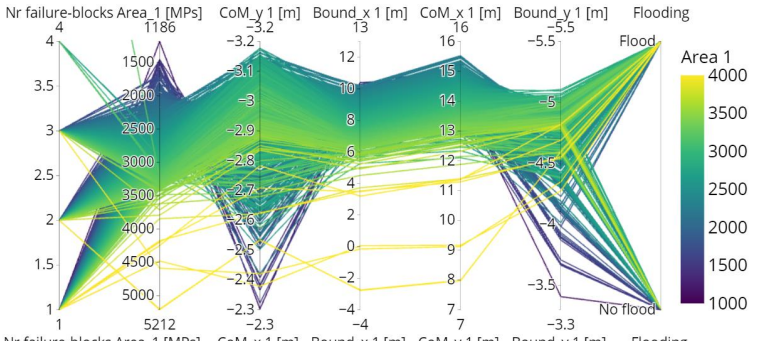
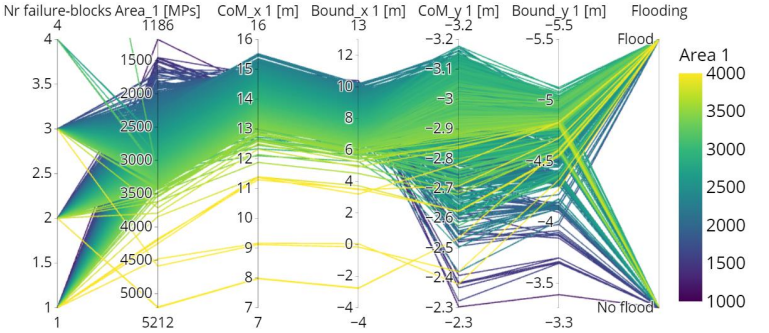


Figure 5.14: Parallel coordinate set 1 with one polyline to represent one realization, and at the bottom the failure profile at t_{br1} with all attributes from Set 1 indicated.

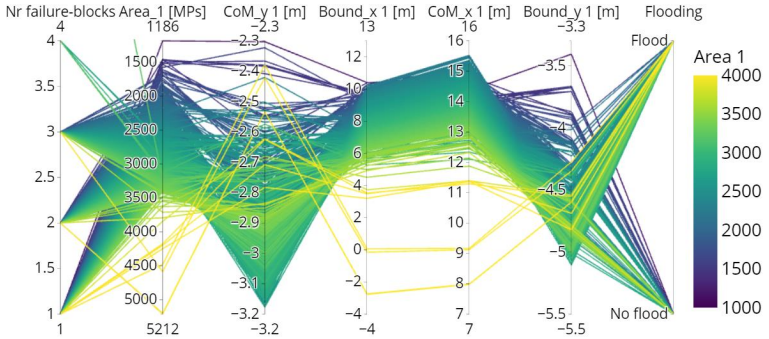
of the axis in parallel coordinates. Taking adjustment of the ordering of the axes and the arrangement of numbers within the axis together of Figure 5.15b and 5.15c, it gives graph shown in Figure 5.15d. It shows the correlations between the parameters as described in Eq. (5.7) and (5.8) and the size of the failure area. However, to have an even more optimized ordering of the axes, the places of $bound_y^1$ and CoM_y^1 are switched compared to Figure 5.15d. It shows the correlation between $bound_y^1$ and $bound_x^1$ in Figure 5.15e and the dependency between the boundaries and also between CoM and area.



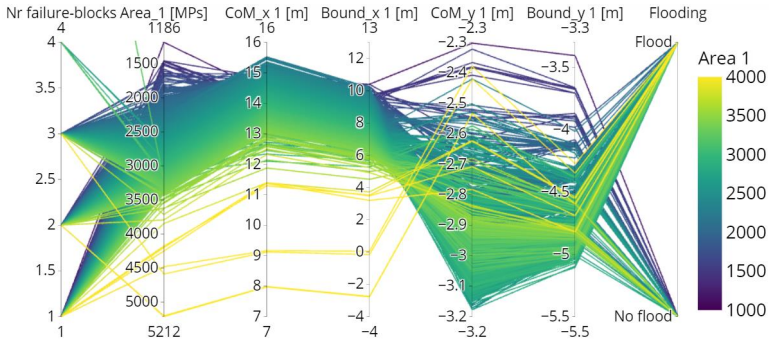
(a) First attempt of choosing the order of axis based on the size of the initial failure area.



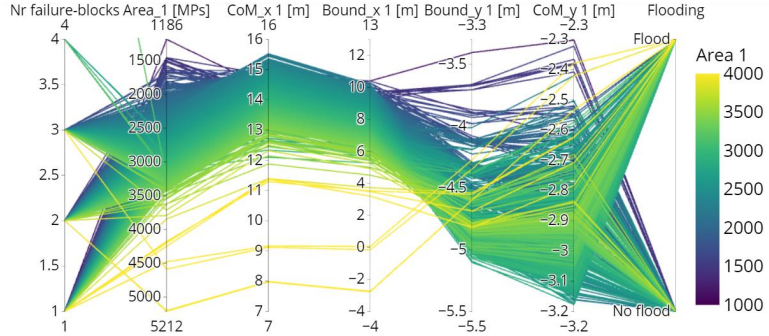
(b) Changing the order of axes compared to Figure a; CoM_x1 and CoM_y1 are switched.



(c) Flipping the numbering of the axis of Bound_y1 and CoM_y1 compared to Figure a.



(d) Combining the adjustment done in Figure b and c, which are changing the ordering of axes and flipping the ordering of number of axis.



(e) Switching the order of axes compared to Figure d. CoM_y1 and Bound_y1 are switched.

Figure 5.15: Parallel coordinates Set 1 showing the importance of choosing the order of axis.

After selecting the order and numbering of the axes, Cartesian graphs are used to verify correlation shown in Parallel coordinates and for more detailed analysis, which is shown in Figure 5.16.

Figure 5.16 reveals the correlation between CoM^1, the back-calculated boundaries, and the size of the initial failure block, i.e. the number of MPs clustered into this block, shown in Figure 5.16b and 5.16c. The width of the failure surface is correlated to the size of the failure block and thereby also the horizontal position of the center of mass. The correlation is revealed by the continuous color transition parallel lines of the polylines in the parallel coordinate graph. The depth of the failure surface does not influence wide and large failure surfaces. However, it does affect smaller failure surface which can be seen by the spread of the graph towards smaller larger numbers of $bound_x^1$ in Figure 5.16b.

While an almost linear relationship exists between the width of the failure surface and the center of mass, bands can be observed in the vertical boundary position in Figure5.16a and 5.16c. Each band in Figure 5.16c corresponds to a certain depth of the failure block, i.e. deeper blocks increase the size of the failure as expected. The correlation within these bands are further explored in Section 5.2.1. Moreover, deeper blocks can have wider failure surfaces, which further increases the size of the surface but also the spread of data per band.

The correlation between the width of the failure surface and the size of the failure block has more influence compared to the correlation between the vertical position of the center of mass and the depth of the failure surface (Figure 5.16c). In this figure, bands can again be observed, and the distance between each band is roughly $0.25m$, corresponding to the distance between the material points at the start of the simulation.

Finally, from this parallel coordinate plot, one cannot yet find correlations between the remainder of the failure process, i.e. the total number of failure blocks that form and/or whether or not flooding occurred, except for some extreme cases.

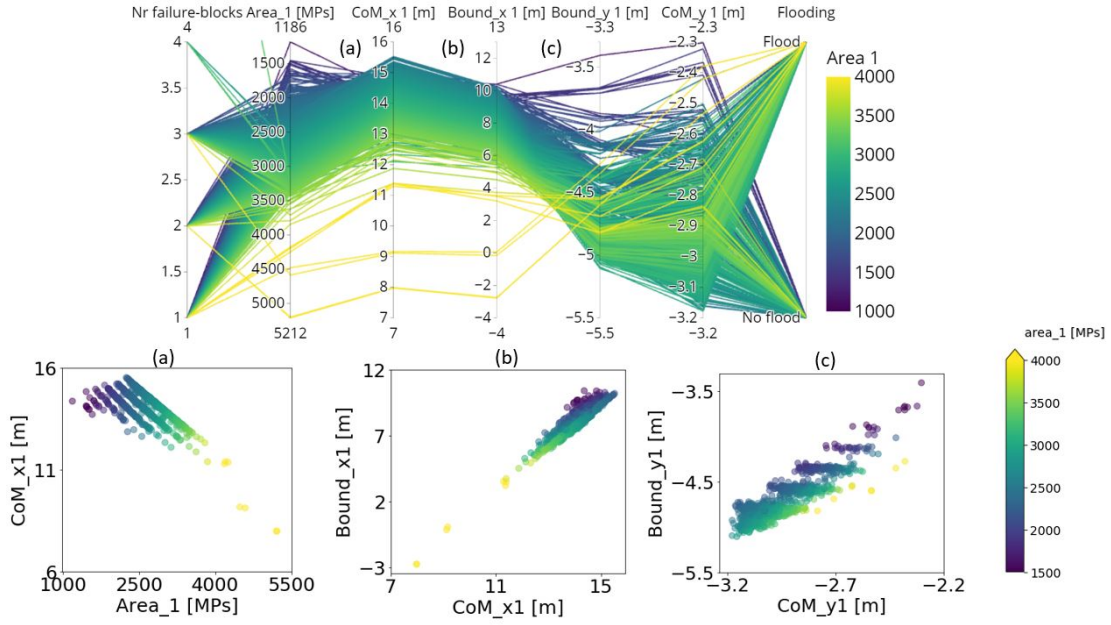


Figure 5.16: Parallel coordinate Set 1 showing 1189 realization with failing behavior, showing seven attributes of the dyke and colored according to the size of the initial failure surface. Graphs [a-c] shows the correlation of attributes in Cartesian coordinates.

Verification of back-calculated size of the failure block

A further analysis is conducted in Figure 5.17 to explain the correlations of the bands seen in the previous Cartesian graphs. While CoM_x^1 and CoM_y^1 are clearly correlated, most of the correlation has been removed when $bound_x^1$ and $bound_y^1$ are back-calculated from CoM^1, as can be observed in the Figure 5.17a and Figure 5.17b. The correlation of CoM_x^1 and CoM_y^1 can be explained by the form of the initial

failure block. For example, as the width of a failure block increases at a certain depth, it affects CoM^1 in both horizontal and vertical directions due to its shape, which is illustrated with the dyke in Figure 5.17. The different bands in Figure 5.17a, correspond to different depths of the failure blocks, and should therefore be more or less horizontal in Figure 5.17b. It indicates that the back-calculation is working as intended. A small correlation can still be observed between $bound_x^1$ and $bound_y^1$ since shallow failures are less likely to generate wide failures, which can be expected due to the roughly circular failure surfaces. The ranges of failure widths are therefore larger for deeper failure surfaces, which is also observed in Figure 5.16.

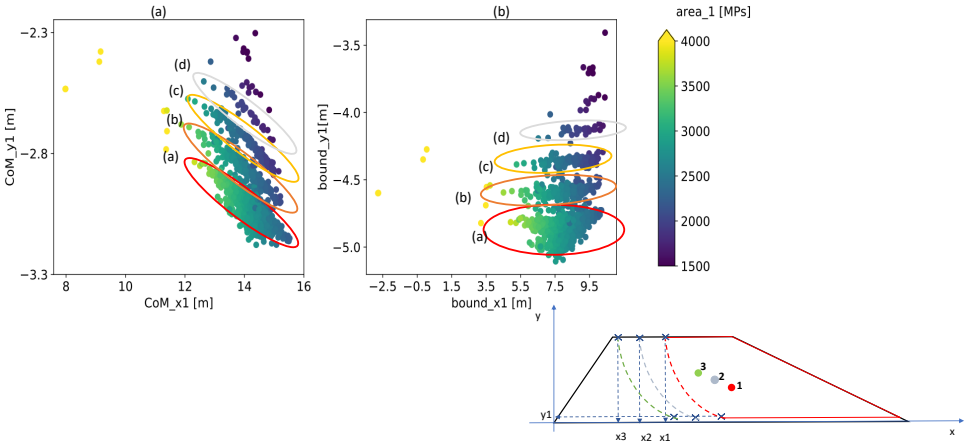


Figure 5.17: The correlation between CoM_x^1 and CoM_y^1 explained using the boundaries in the right figure and a visualization of the dyke with different width of failure blocks at a certain depth y_1.

In Figure 5.18 the back-calculated boundaries are further verified against the coordinate of the absolute minimum material points in both horizontal and vertical direction of the initial failure block. The results prove that the back-calculated boundaries are accurate for both x and y-direction. Moreover, they always present the failure block's overall response (since they are based on an average position). In contrast, a minimum/maximum position can be affected a lot by a single material point. However, in the x-direction, a shift of the data of around 1 meter is present, shown in Figure 5.18a. This shift is also observed in Figure 5.6, where $bound_x^1$ cuts the failure block in area $A3$. This shift may be minimized by changing the width of area $A3$, which will also affect the y-direction. Therefore, the width of $A3$ has not to be modified.

At t_{br1}, the MPs will still mostly be in their original position as the movement of the simulation just started. Therefore, gaps of $0.25m$ high are expected in the verification data due to the distance between MPs discretization, seen in the right of Figure 5.18. As the CoM^1 computes the average position of all MPs, it is less restrained to these gaps, and it causes the differences in the vertical direction. In Figure 5.19, the data in y-direction is split into flooded and non-flooded simulation. It indicates that for non-flooded realizations, vertical boundaries are more restrained to the gaps than the flooded realization. Especially the deeper surfaces show a larger spread. Moreover, wider failure blocks (which are more likely to reach flooding) or horizontal failures may not accurately fit the failure surface assumed for the back-calculation. Therefore, differences of the form of the failure surfaces can be larger for flooded simulations.

Correlation between failure block and failure profile

The secondary goal of the analysis is to observe differences between the different failure profiles, described in Section 3.3, using the size and shape of the initial failure block. Brushing has been used to separate the results of the five classes as shown in Figure 5.20, where only parts of the realizations are selected to visualize. The 'horizontal' class has a relatively different behavior pattern, i.e. it has more extreme behavior than the other failure classes. Overall, the parallel coordinate plots show similar behavior between the different classes. Only in the extremes, the differences can be observed more easily.

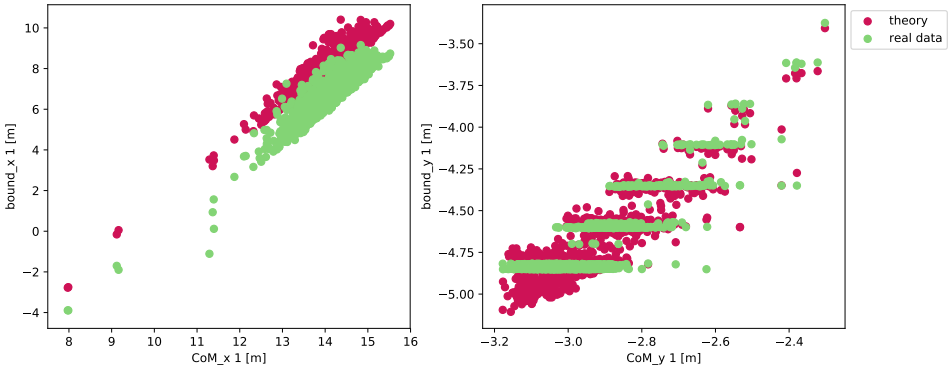


Figure 5.18: Comparison between theoretical and actual data for boundaries of the initial failure block; (right) horizontal data corresponding to the width of the failure surface; (left) vertical data corresponding to the width of the failure surface.

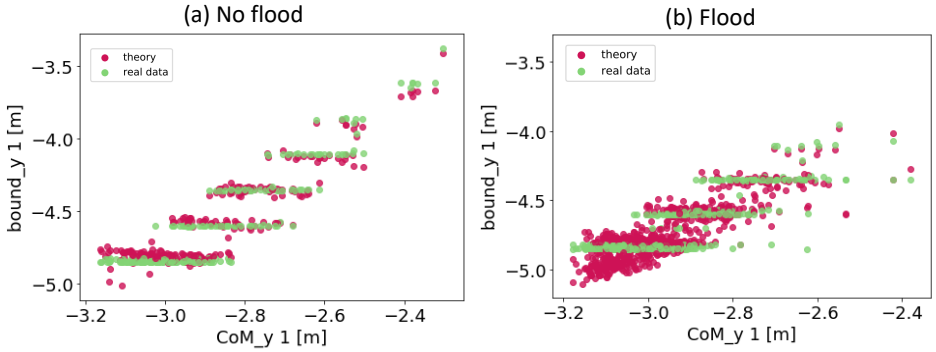
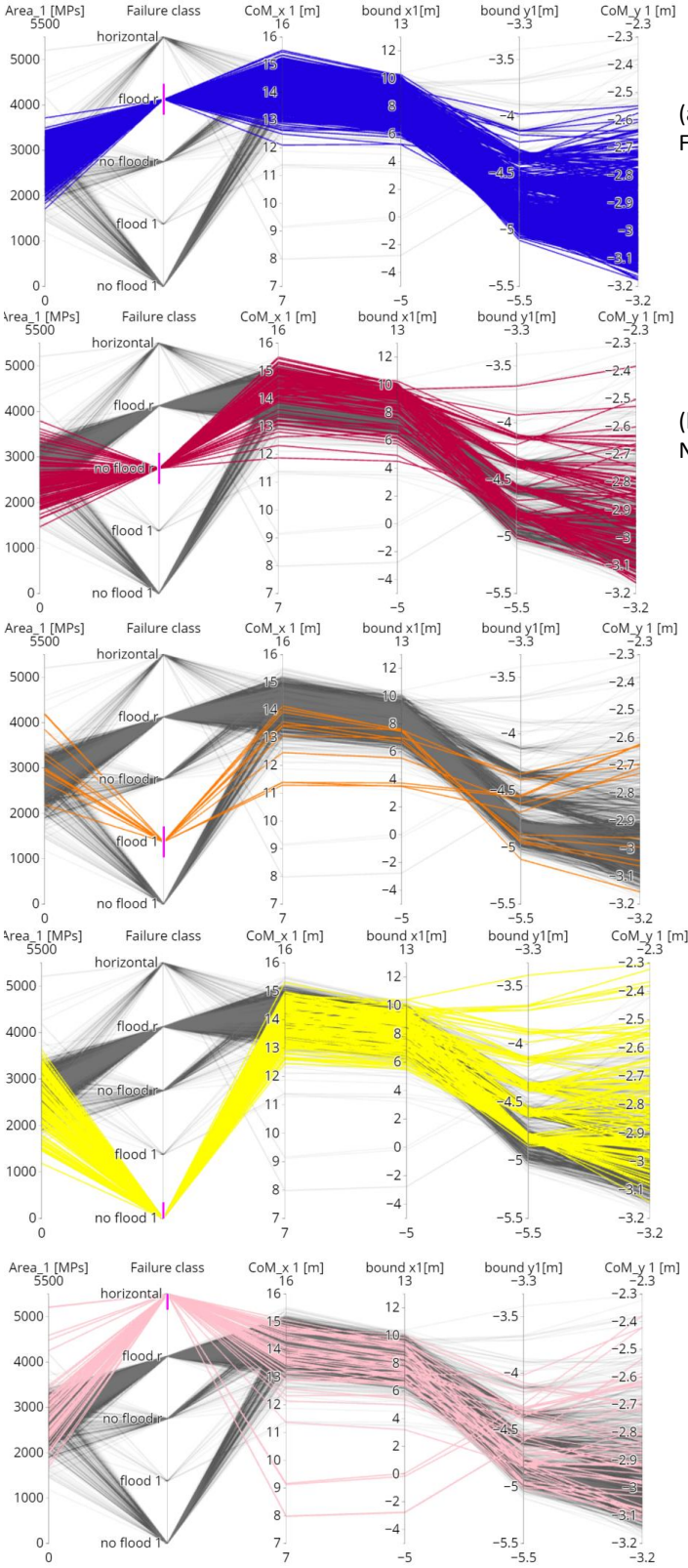


Figure 5.19: Comparison between theoretical and verification data for y-boundary of the initial failure blocks; a) non-flooded realizations; b) flooded realizations.

For example, when flooded and non-flooded realizations are compared, floods are less likely for shallow failures, while failure along the base almost always triggers flooding. Moreover, large failure widths are required to trigger flooding after a single failure, as expected, but retrogressive failure seems more likely for small failure widths.

Observing differences within the most likely outcomes is difficult in parallel coordinates, even when a different color scheme is used (for example, the failure depth seen in Figure 5.21). In Figure 5.21 a clear correlation between CoM_y^1, failure area, and $bound_y^1$ is shown. However, correlations with the failure groups are more difficult to observed (except for the extreme values of $bound_y^1$).



(a) Multiple failure blocks (retrogressive)
Flooded

(b) One failure block
Not flooded

(c) One failure block
Flooded

(d) Multiple failure block (retrogressive)
Not flooded

(e) Horizontal
Flooded

Figure 5.20: Comparison of behavior for five failure profiles in parallel coordinate Set 1; (a) Retrogressive flooding; (b)No-flooding with one failure block; (c) Flooding with one failure block; (d) retrogressive no-flooding; (e) Horizontal failure.

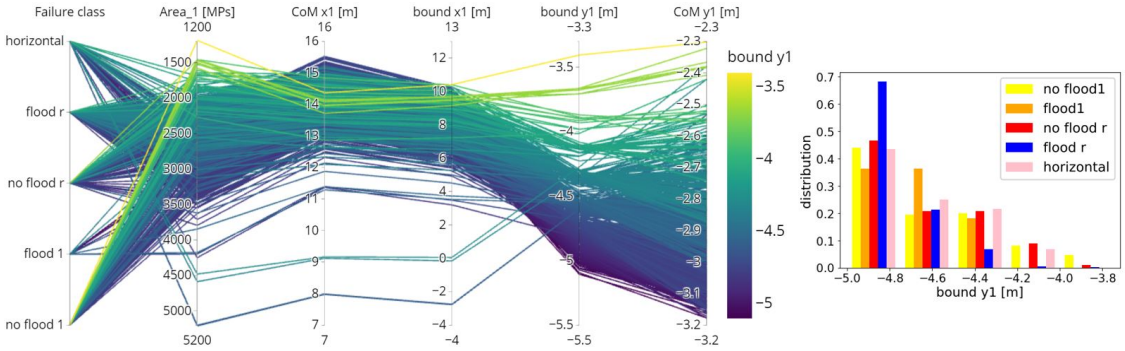


Figure 5.21: Parallel coordinate Set 1, colored according to $bound_y^1$ of the initial failure block and corresponding histogram for the five failure classes.

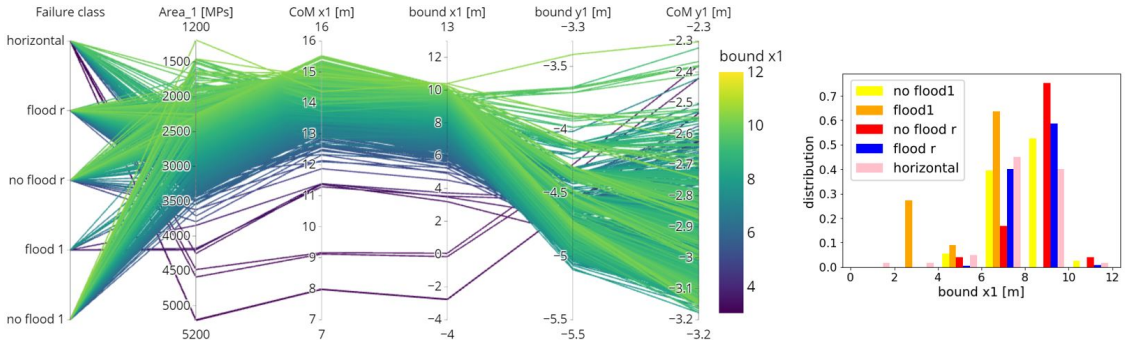


Figure 5.22: Parallel coordinate Set 1, colored according to $bound_x^1$ of the initial failure block and corresponding histogram for the five failure classes.

Therefore, more standard histograms can be used to explore further correlations found in the extreme data values in parallel coordinates. Figures 5.21 and 5.22 plot the shape of the failures subdivided into failure categories. The total number of failures within each category is different, and the histograms are weighted against the number of realizations per failure class.

As shown in Figure 5.21, it is difficult to distinguish the failure groups based on the depth of the initial failure. For all classes, shallow failure is less likely. Focusing on the histogram in Figure 5.21, the non-flooded realizations decrease of shallower failures cases, and is more gradual than the flooded realizations. As a result, the failure surface with a shallowest depth results in 'No Flood r' and 'Flood 1' realizations. Figure 5.22 shows the correlation for $bound_x^1$ in both parallel coordinate plot and histogram graph. Failure occurs most in the range of 6 to 10 m for all, so for most realizations, more than half of the crest remains after initial failure. Smaller failures width are highly unlikely, as they require a shallow depth of the failure block. Additionally, retrogressive failures are less frequent at wider initial failures compared to non-retrogressive failure classes.

It is surprising that flooding after a single instability, i.e. failure class 'Flood 1', can occur for $bound_x^1$ > 2 m. One would expect that a wide initial failure block cutting through the outer slope is required to lead to flooding. This however, does not appear in Figure 5.22, where a set of realizations are in the range of 6-8 m. Investigating a single realization belonging to this group, indicates the cause in Figure 5.23. The left figure shows the moved material points, and the darker color marks the initial failure formed by K-MCM at t_{end}. This failure indeed does not intersect with the outer-slope. However, a secondary failure has formed, which is undetected by the K-MCM algorithm due to the limited movement required to reach failure from this point. In the right figure, the plastic deviatoric strain at t_{end} again shows the presence of a secondary failure block intersecting with the outer-slope. So, the class 'Flood 1' consist also of realizations with a secondary failure with limited deformations. A similar observation was already observed for horizontal failures. Chapter 3 describes that within the horizontal failure class,

two different types of failures exist: 1) complete horizontal sliding or 2) initial rotational failure followed by horizontal sliding. The relatively small failure width in Figure 5.22 indicates that failure Type 1 is less common compared To type 2.

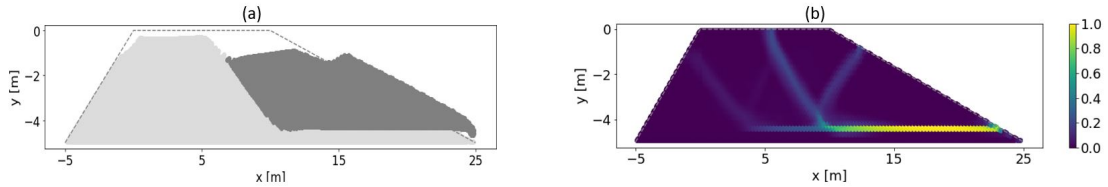


Figure 5.23: Realization with a small initial failure width: (a) MP location at the end of simulation (t_{end}), darker color indicates the initial failure block; (b) normalized plastic deviatoric strain (t_{end}).

Finally, Figure 5.22 is less predictable compared to Figure 5.21. The extremes in this figure appear to be outliers, which follow different behavior from the majority of the realizations. The behavior of these realizations can indeed be observed as outliers in the parallel coordinates by using the brushing method (see Figure 5.24). This indicates that extremely wide initial failures will lead to flooding, while shallow failures width a small width never lead to flooding. It is important to note that both a small width and a shallow failure are required to prevent flooding, so a small width combined with a deeper slide may still lead to flooding.

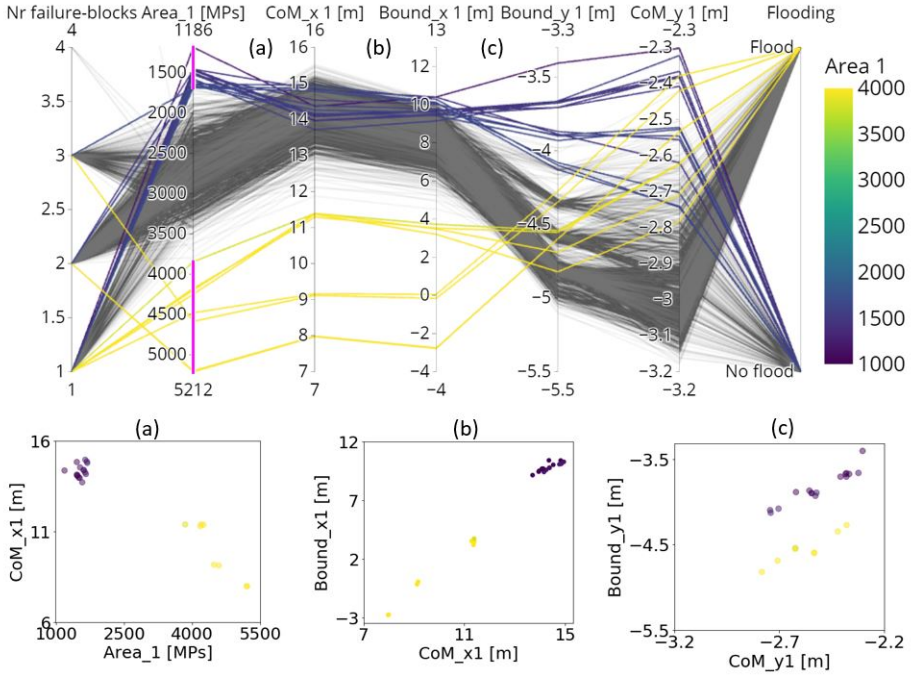


Figure 5.24: Outliers from the parallel coordinates Set 1 with extreme small (purple) and extreme large (yellow) initial failure blocks, with corresponding Cartesian graphs.

5.2.2. Set 2: Displacement of first failure block and final failure profile

The second set of attributes is created based on the findings in Section 5.1.5, which suggest that there is a correlation between the duration of the simulation, displacement of the initial failure, and the final failure profile. It aims to investigate the proposed correlations. The parallel lines are colored according to flooded and not-flooded realizations to highlight correlations with the probability of flooding. The second set of attributes (Set 2) are as follows:

- The size of the initial failure block (A^1), i.e. number of material points in this block.
- The lower boundary of the initial failure block ($bound_y^1$).
- Duration of simulation from t_0 to t_{end}.
- Displacement of the initial failure block in horizontal direction (Δx_1) using to discrete points in time t_{br1} and t_{end} for CoM_x^1.
- Displacement of the initial failure block in vertical direction (Δy_1) using to discrete points in time t_{br1} and t_{end} for CoM_y^1.

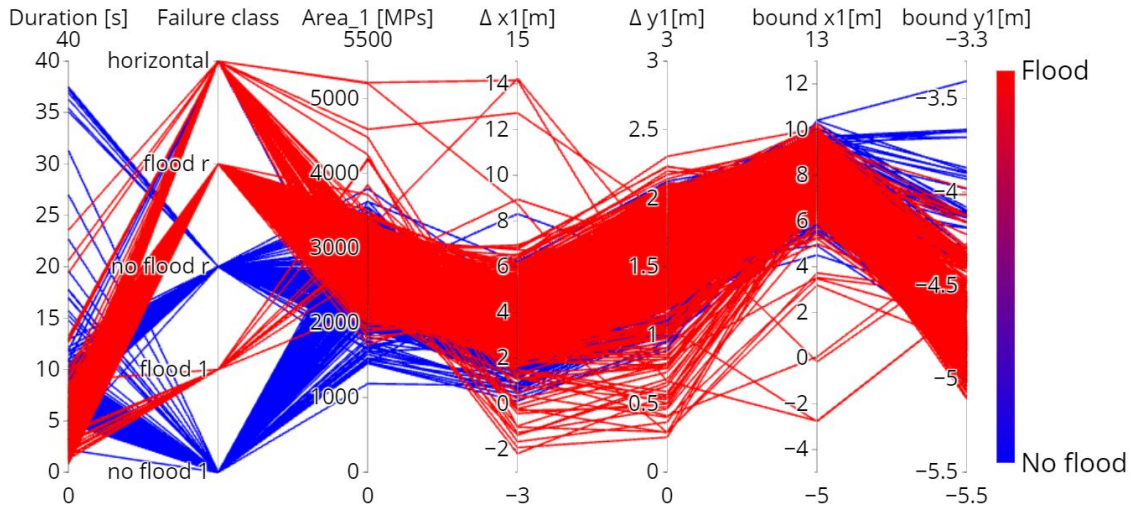


Figure 5.25: Parallel coordinates Set 2 realizations colored according to flooded and non-flooded realizations.

In Figure 5.25, the displacement and duration of the first failure block of the five classes are investigated to observe differences between the failure modes. Most simulations end within 15 seconds, and after 15 seconds, only horizontal failure can still lead to flooding as the dyke is slowly being pushed until it drops below the water level. Retrogressive failure also has a large horizontal displacement, but it occurs quickly comparing to horizontal failure. The duration is further investigated in Figure 5.26. Flooding occurs within 10 seconds in most cases, while most realizations find a stable equilibrium later on. This indicates that if flooding occurs, retrogressive failure occurs as the first failure is still in development. So, simulations which lead to flooding are highly unstable, leading to larger velocities and thereby causing flooding, something which could also be seen in Section5.1.3 and Figure 5.25.

In Figure 5.27 the time between the first and the follow-up failures is shown, excluding the realizations without secondary failure blocks. The probability of flooding decreases when the time between two failures becomes longer, confirming that the realizations leading to flooding are highly unstable. Moreover, horizontal failures of type 2 (where a horizontal failure develops after a rotational first slide) develop quickly due to the presence of the weak layer required to trigger horizontal failure.

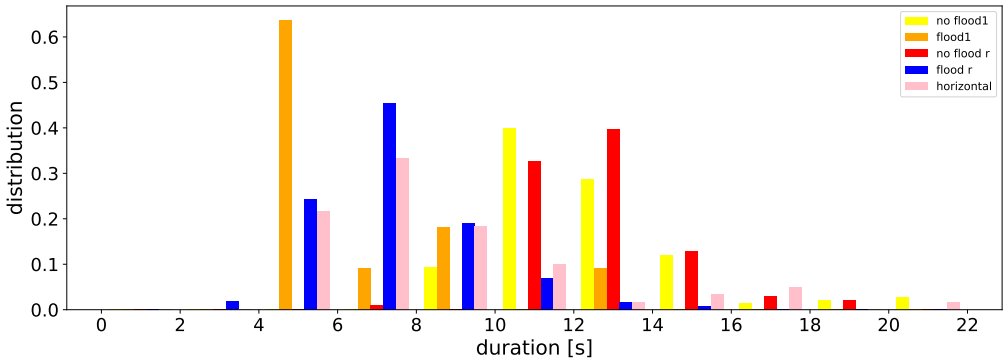


Figure 5.26: Histogram of duration of the failure from t_0 to t_{end} for five failure classes, weighted against the number of realizations per failure class.

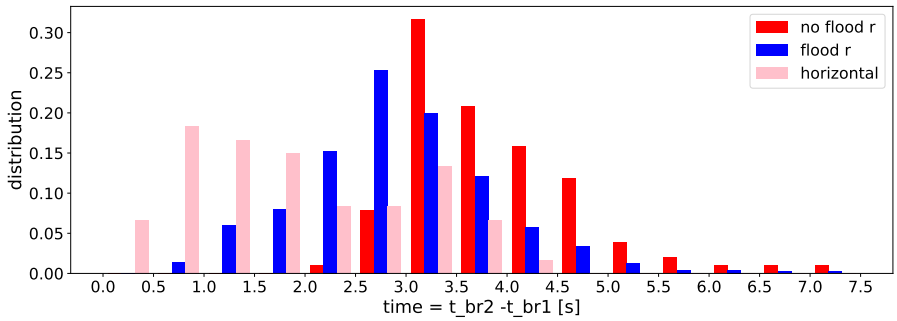


Figure 5.27: Histogram of time between the formation initial failure block t_{br1} and the following failure block t_{br2} for three failure classes, weighted against the number of realizations per failure class.

Figure 5.25 explores further the displacements and reveals an unexpected finding: realizations with a very low displacement cause flooding, while one would expect the contrary, i.e. large displacement should flooding. Therefore, Figure 5.28 focuses on the horizontal displacement of the ensemble. In Figure 5.28b, only realizations with low displacement are selected: these realizations are from the failure categories 'Horizontal' and 'Flood 1'. The initial failure blocks are large of these categories, such that these failures only require limited deformation to cause flooding. This selection of parallel coordinates shows how multiple attributes are connected to each other, to represent a certain failure profile.

In Figure 5.28c, the realizations with larger horizontal displacements are selected, which all fall in the retrogressive failure and horizontal failure categories. Again, as described in Section 5.1.1, the development of a secondary failure forces the initial failure block to travel further in both horizontal and vertical directions. Moreover, as flooding after initial failure is not triggered for these simulations, significant deformation can be observed before the simulation ends.

In Figure 5.29 and 5.30 the displacement is further investigated using histograms. The horizontal and vertical displacement behavior is similar for the retrogressive and horizontal failure classes, while the simulations with a single failure block present significantly less deformation.

Another observation is that the failure classes 'Flood 1' and 'No flood 1' show a counter intuitive behavior in the histograms. It shows a decreasing probability of flooding with larger displacement for realizations, which is correlated with the size of the initial failure blocks and illustrated in Figure 5.31. Retrogressive failures, on the other hand, have similar behavior for both flooded and non-flooded realizations within the histogram of Figure 5.29. This behavior is due to the lack of information about the secondary failure and causing a large uncertainty. Therefore, it is impossible to distinguish the behavior between retrogressive flooded and non-flooded realizations based on the displacement of initial failure blocks.

The last remark is that the largest vertical displacement of the initial failure block is at a maximum

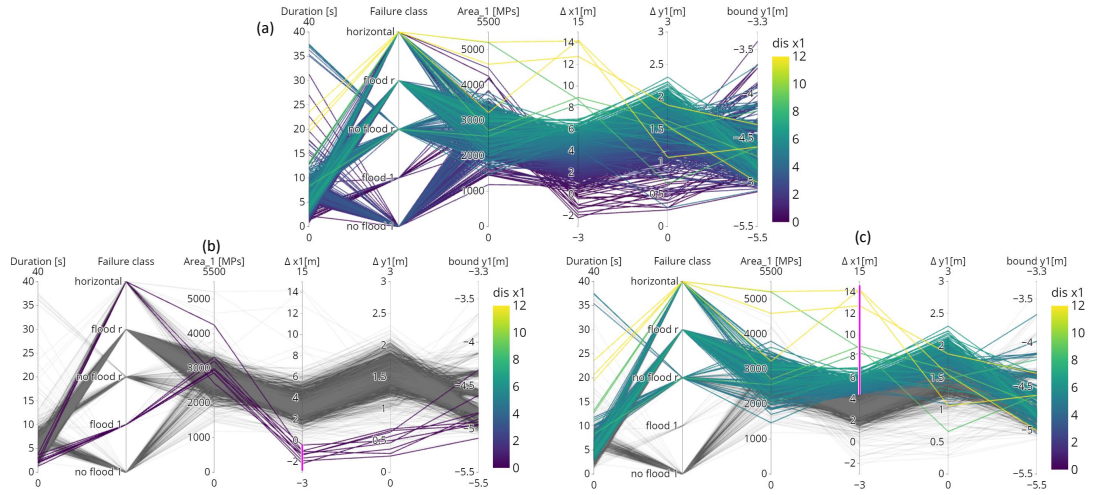


Figure 5.28: Parallel coordinates Set 2 realization colored according to horizontal displacement of the initial failure block; (a) no brushing is applied; (b) low displacement selected consist of failure class 'Flood 1' and 'Horizontal'; (c) high horizontal displacement selected consist of failure class 'Flood r', 'No flood r' and 'Horizontal'.

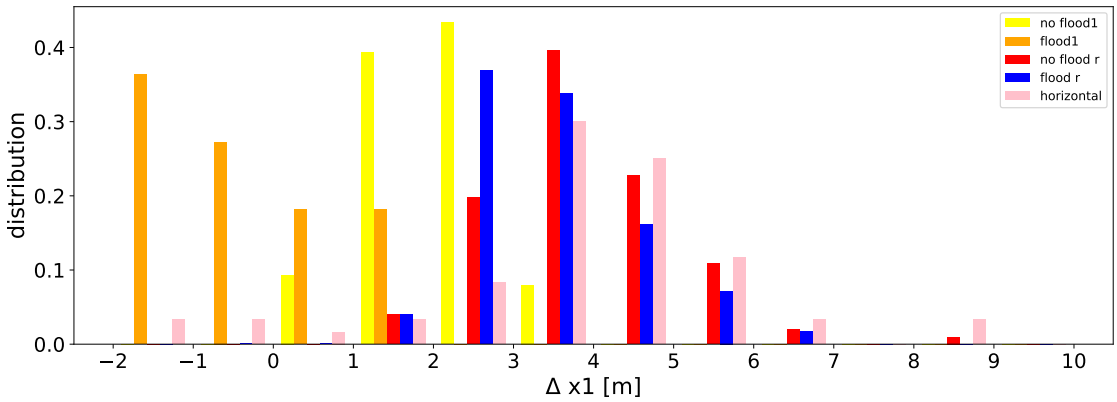


Figure 5.29: Histogram of the horizontal displacement for the five failure classes, weighted against the number of realizations per failure class.

of 2.3 m, while most simulation experiences much less vertical displacement of the crest. So, the assumption of the vertical displacement of half the dykes' initial height in the current assessment, i.e. 2.5 m, is conservative for this simulation. However, even though the vertical displacement is smaller compared to the crude assumption in the current assessment, flooding can still occur relatively easily due to retrogressive failures along with the weak layers.

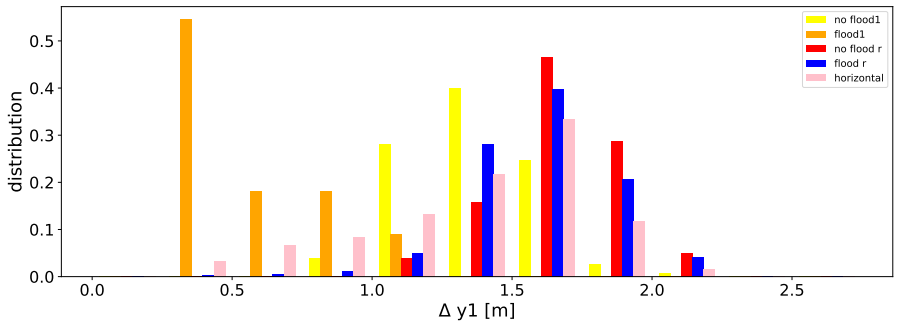


Figure 5.30: Histogram of the vertical displacement for the five failure classes, weighted against the number of realizations per failure class.

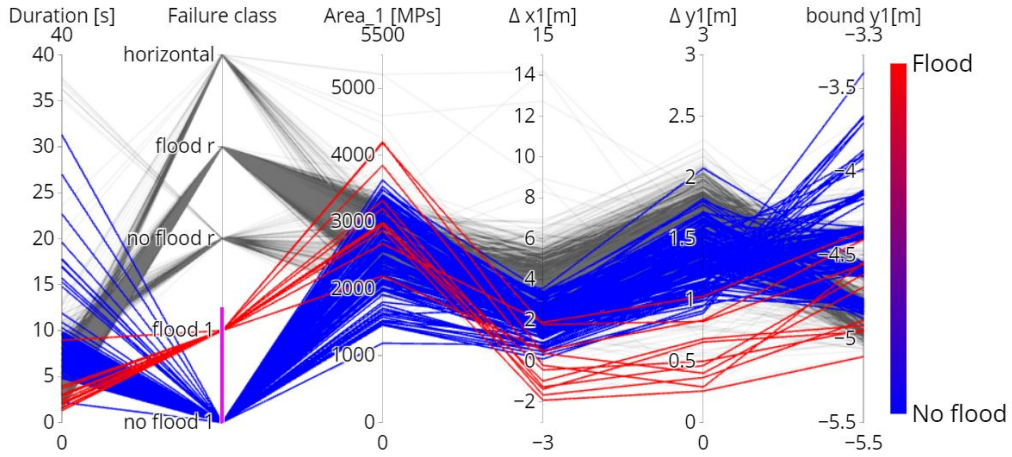


Figure 5.31: Parallel coordinate Set 2, colored according to flooding with only realizations selected with one failure block are selected.

5.3. Classification and results

To answer the question: *"What is the role of the visual representation of RMPM?"*. A potential visualization is created, which has the goal to give a summary of the RMPM dataset with both scientific values and illustrations with a certain realism, shown in Figure 5.32. The visualization is based on one RMPM dataset; it shows a representative dyke realization of each failure profile using the Blender method of Chapter 4, which shows the dyke profile in a graphical-realistic manner. Moreover, it shows the characteristic values of each failure profile using Parallel coordinates graphs, in which the representative realization is highlighted. The graphs gives an indication of how many realizations are in each profile, for example, this RMPM dataset have many realizations in failure profile 'Flooded retrogressive', while it is less in the failure profile 'Flooded one block'. It also gives an indication of how much the values are spread within each failure profile.

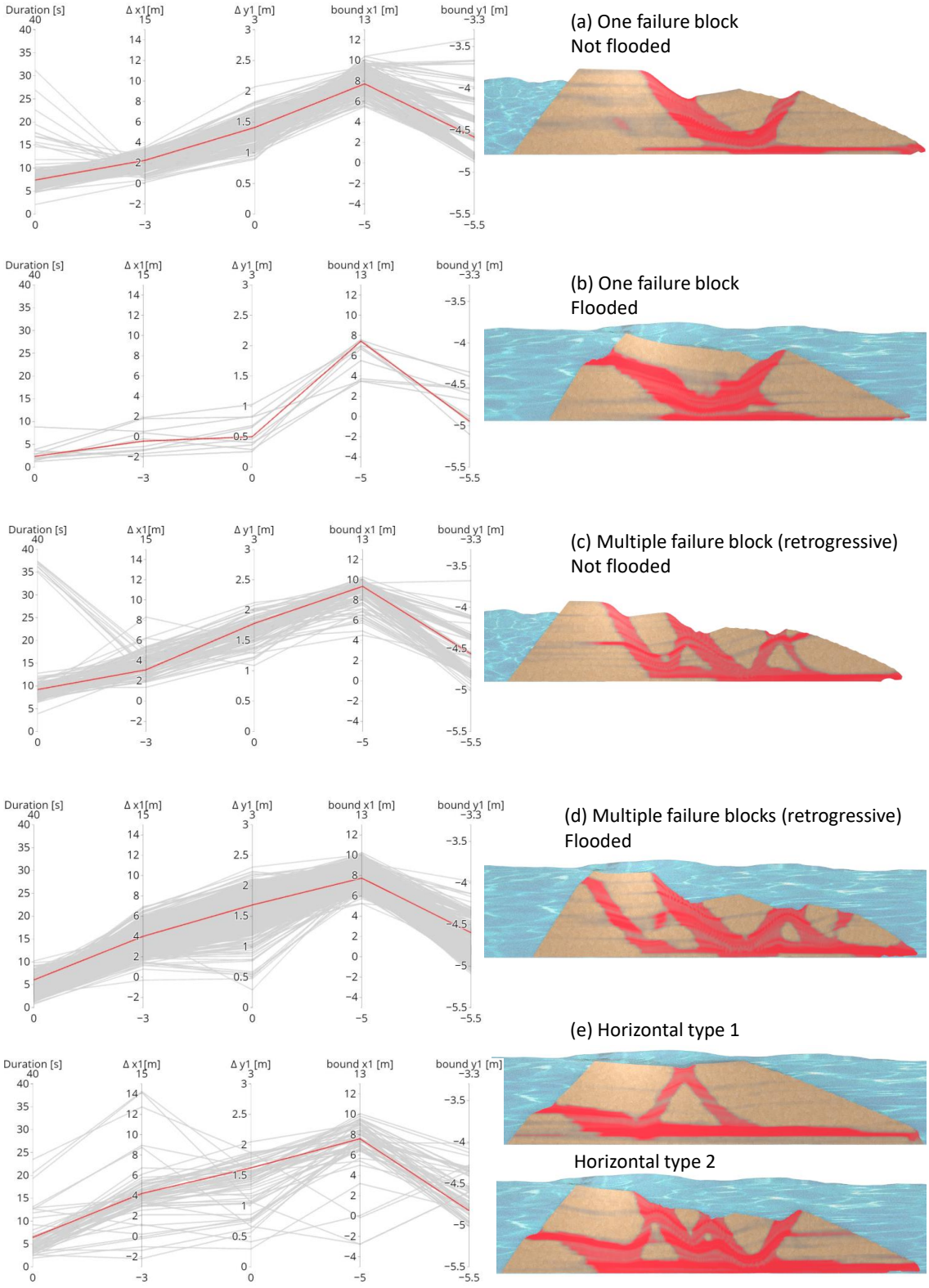


Figure 5.32: A possible RMPM illustration with the goal to summarize what kind of failure profiles are present within the dataset and their characteristic value using parallel coordinate graphs.

5.4. Conclusion and Discussion

This thesis presents a data analysis with focus on reduction of data in the RMPM ensemble by finding characteristic parameters to identify each realization as one out of five failure classes. The average behavior of the failure groups has been summarized by selecting a single representative realization using parallel coordinates, which is then visualized using Blender. These visualizations can be used to explain the results of RMPM intuitively.

Only a selected set of properties is tested, as a complete data analysis is too time consuming, even if strategies such as parallel coordinates are used. Unfortunately, the results show that many parameters related to the first failure block are uncorrelated to the probability of flooding. Machine learning can further improve the data analyses and perhaps better correlation can then be found.

Furthermore, the analysis have demonstrated the value of computing the CoM instead of tracking MP individually. Only a couple of parameters have to be tracked for a single point for each failure block found in the K-MCM algorithm, which results in a significant reduction of data size. In this ensemble the original data size of roughly 3000 Gigabytes, has been reduced to 6 Gigabytes. This reduced dataset is also more efficient for post-processing. For example, the displacement graph in Figure 5.2 is reproduced using the reduced dataset, which reduces the computation time from 4 minutes and 4 seconds to 0.2 seconds.

The results show that correlations and causations are observed easier when multiple attributes are studied at once using parallel coordinates. Observing many parameters at once might be even more important in the field of geo-engineering, because of the larger number of parameters compared to other fields. To verify and explain the correlation histograms remain vital, as parallel coordinates can lead to blending of clusters for each attribute. Histograms are also required to observe probabilities. The histograms are weighted according to the number of realizations of each failure class for easier comparison and one should keep in mind that the number of realizations in each class are different within the RMPM model.

Within this research only one RMPM ensembles is investigated. Therefore, results may change for different simulations, and more studies on different simulations should be performed in the future. Parallel coordinates can help in these studies to find correlations faster. For example, other simulation may find a larger vertical displacement, and therefore the conclusion that vertical displacements of 0.5 times the original height of the dyke are rare should be interpreted with caution. However, as the modeled dyke is weaker than typically observed much larger vertical displacements are unlikely. the finding is still convincing.

Stress/strain diagrams have not been used in the latter parts of this study due to the oscillations. It is recommended to include a technique to reduce volumetric locking in MPM, after which further investigation is required to automatically detect points for stress/strain diagrams in each realization.

The K-MCM algorithm determines the size of the failure block separately in each time step. Therefore, the material points clustered in a block can change over time. This can effect the calculated displacement over time. This phenomena clarifies the negative numbers for the horizontal displacement of Figure 5.28. In other words, these negative displacements are not caused by outer slope failures.

The back-calculated boundaries using CoM reduces the uncertainty compared to other methods finding boundary. The back-calculation can be further improved by removing the shift of the vertical boundary. In other words, failure blocks are expected to be wider than the computed boundary. The approximation of non-flooded data appears to be better compared to flooded data, seen in Figure 5.18, which is a possible indication that flooded realizations have a different form of failure and therefore a larger uncertainty. This could be an interesting area for further research.

6

Conclusion and outlook

In order to optimize dyke strengthening, geo-engineers would like to assess the strength of a dyke more accurately by including the post-failure behavior in the assessment. The random material point method (RMPM) can model the post-failure behavior while also accounting for the spatial variability of the strength properties of the soil. However, the geo-engineer is unable to clearly and efficiently communicate the results of RMPM. After MPM has been used successfully for visual creations within the computer graphics field, it sparks new interest in using more advanced visualization techniques to visually communicate detailed assessments with RMPM to a broader audience and within the field.

A given RMPM dataset is used to investigate the possibility of making visualizations with graphical realism using the computer graphic software Blender while being able to show of attributes from scientific data. A VTK add-on has been included to transform the data to a mesh and maintaining the data attributes of the visualizations. This add-on is still under development, and the visuals created within this thesis could not have been created a year earlier as the add-on was not developed far enough back then. Moreover, leaps of progress have been made within the Blender software during this thesis.

Finding a method to visualize the whole ensemble is complicated as the data is high-dimensional and with too many realizations to design one visualization to represent the datset in an efficient time frame. Therefore, each realization is characterized into five different failure profiles: four failure profiles are classified based on the number of retrogressive failures and whether or not the realization resulted in flooding, while the fifth class describes horizontal failures. The modified K-MCM algorithm divides the material points over multiple retrogressive failures and characterizes the first four groups. A newly developed algorithm detects the fifth failure group, which separates the horizontal failure based on the plastic deviatoric strain algorithm. The K-MCM algorithm is also used to compute the center of mass and its deformations, which significantly reduces the amount of data within the ensemble while maintaining the characteristics of the ensemble.

An investigation of the reduced ensemble with parallel coordinates has discovered relations between realizations. As the method allows showing multiple attributes at once, it is advantageous for geotechnical problems, which usually have more properties than other materials. Clear correlations between the size and shape of the initial failure and the total failure profile have not been found. Better correlations are observed with the deformations of the first failure block, but many properties still overlap. So, extrapolation from the first failure to the rest of the failure profile is difficult, which shows the importance of taking account of the full post-failure behavior using RMPM. One interesting finding is that equilibrium of the initial failure block is often reached before a vertical crest displacement equal to 0.5 times the height of the dyke. This indicates that the crude estimation in the current assessment is highly conservative. Moreover, within the assumption, it is hypothesized that the secondary failure block will only form after the initial failure block has reached its equilibrium, which is shown otherwise within the data analysis of this thesis.

To illustrate the method, a visualization have been designed, in which it highlights the characteristics using parallel coordinates for each failure profile along with an explanatory visual, each representing a failure profile. It shows the effectiveness of visual communication by combining scientific data with an appealing representation of a dyke, even for complex computations such as RMPM ensembles.

This research have demonstrated how geo-engineers could be in charge of their own creative designs and appealing visual representations without communicating through another layer i.e. an illustrator, which may lead to miscommunication. Taken together, the results suggest that the method is not limited by RMPM in slope stability problems, it can also be applied to other point-data and also other geo-technical problems.

6.1. Contributions

The main contributions of this thesis are as follows:

- A design for a scientific visualization pipeline in VTK, which used Delaunay triangulation to transform material points from one RMPM realization to a mesh while keeping the properties of material points attached to the mesh.
- A design for an artistic visualization pipeline in Blender, which transforms the mesh to a realistic visual while maintaining the option to show scientific attributes from the dataset.
- A method to detect horizontal failures using plastic deviatoric strain.
- Categorizing of the ensemble into one out of five described failure profiles.
- RMPM data analysis showing the effectiveness of describing the failure process using the center of mass.
- A method to analyze RMPM data using parallel coordinates and histograms, which can categorize failure outcomes and verify certain expected correlated behavior. Parallel coordinates is also a method, which can give an overview of multiple attributes simultaneously and can therefore show the characteristics per failure type.

6.2. Suggestion for future work

The MPM method is relatively new compared to FEM and LEM, and RMPM is even newer. There are promising results within the data analysis, which indicates that the current assessment can be improved. However, the RMPM is still in development, and more investigation is needed to, for example, account for the oscillations within the stress-strain diagrams. Additionally, different types of soils need to be tested before the method can be used in practice. This study is still useful for different RMPM datasets and even for other point-cloud datasets, which is also of interest to other researchers and companies. The findings within this thesis raise new questions for future research:

- The accuracy of the visualization of the dyke may be improved by using the edge detection file (PMF) [49]. The current edge is placed at the material points, which represent the middle of a material domain.
- The definition of flooding within the RMPM simulation is strongly recommended to be re-evaluated. Some realizations with large failure blocks remain just above the water level, and are most likely highly unstable in reality. Perhaps the detection of softening of the soil, i.e. reduction of undrained shear strength, in the outer-slope can be implemented in the RMPM simulation, such that realizations can be defined as flooded when the outer-slope is damaged during a simulation.
- Further research on the characterization of the realizations is recommended. The focus within this thesis is mainly on the initial failure, which could be extended to look at retrogressive failure blocks and the kinematics of these blocks. Perhaps the usage of machine learning algorithm may help determine which parameters characterize the failure profile and can look at the failure path over time instead of a fixed position in time.
- The accuracy of the center of mass may be improved by modifying the K-MCM algorithm. The size of the large failure block in some realizations changes over time as material points are added or removed from the failure block. This makes a comparison of different time steps within a realization difficult. Perhaps the size of the failure block can be fixed after a couple of time steps

after the failure block is created. In this way, the failure block must fully develop before being fixed but cannot change anymore afterward.

- Testing more datasets to broaden the scope of this investigation can inspire more findings and verify the current findings in this thesis. Moreover, testing more datasets can help to generalize the data analysis and the visualization method. The methods proposed in this thesis should be applicable to datasets with different soil conditions and three-dimensional dyke failures.

Nomenclature

Abbreviation

$CPTs$ Cone Penetration Tests

FEM Finite Element Method

FoS Factor of Safety

$GIMP$ Generalized Interpolations Material Point

$HWBP$ 'Hoogwaterbeschermingsprogramma' National flood protection program

LEM Limit Equilibrium method

MPM Material Point Method

MPs Material Points

$OpenVDB$ Open-source library for Volumetric, Dynamic grid with characteristics of B+trees

$RFEM$ Random Finite Element Method

$RMPM$ Random Material Point Method

SFs Shape Functions

ULS Ultimate Limit State

VTK Visualization Toolkit

WBI Wettelijk Beoordelingsinstrumentarium, Dutch assessment guideline and tools to enforce the wateract

Artistic visualization Visuals made with a realistic perception of the reality, so how it would look like in the real world.

BVTK Blender and VTK coupled computer graphic software

DM Double Mapping

Flood 1 Failure profile: one failure block with flooding.

Flood R Failure profile: retrogressive failure block with flooding.

Horizontal Failure profile: horizontal flooding.

No flood 1 Failure profile: one failure block without flooding.

No flood R Failure profile: retrogressive failure block without flooding.

Scientific visualization Visualization of abstract set of numbers (data) in a graphical form.

Symbols

Δx_i The horizontal displacement of number 'i' failure block

Δy_1 The drop of the crest height (vertically). The "i" indicates the number of the failure block.

A_{theo1} The theoretical area of the initial failure block based on CoM_i.

$bound_x$ The horizontal boundary condition of the dyke for estimating A_{theo1}, which is equal to the length from the toe of dyke till the start of the inner slope.

$bound_y$ The vertical boundary condition of the dyke for estimating A_{theo1}, which is equal to the maximum height of the dyke.

CoM Center of Mass

F_{vm}^0 Initial yield surface [kPa]

F_{vm}^{res} Residual yield surface [kPa]

H_i initial height

L Loads

R Resistance

t_0 start time of the simulations

t_{br1} Point of time where the K-MCM algorithm have determine the first failure block

t_{end} End time of the simulations

W_c Crest width

p Mean stress [kPa]

q Deviatoric stress [kPa]

A

Appendix A

This guide shows how to make the visuals seen within this thesis using Blender-VTK for those who have never used graphics software. It is made for Windows users, and most of the visuals are made based on one MPM realization.

A.1. Software installation

Blender is an open-source software and can be found in: https://www.blender.org/ [fixme website]. Version 2.82 is used for this thesis, the installation process might differ for a newer version. Install Blender preferably in a default location, for example, **C: Program Files Blender Foundation**. Startup Blender to confirm correct installation and close Blender for the next step. Within the guide, many features of Blender will be used, and in order to be not confused, an overview is given in Figure A.1.

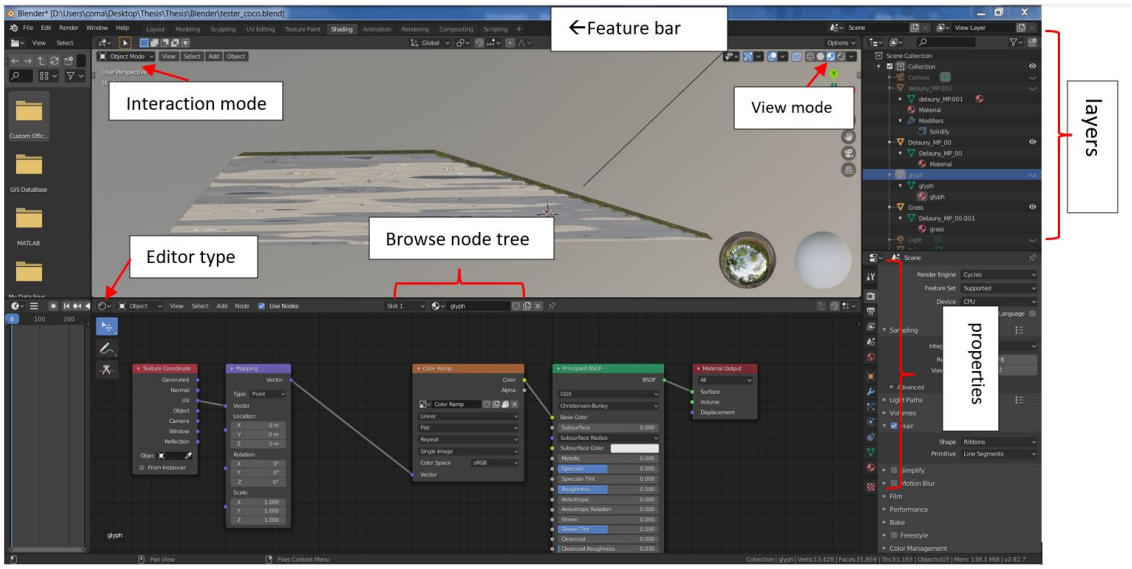


Figure A.1: An overview of the basics in the viewport window for navigation

A.1.1. Add-on: VTK

The VTK library is an add-on, which is needed to import the RMPM data. To install the add-on the following steps are needed:

1. Go to https://github.com/tkeskita/BVtkNodes to download the add-on, by selecting **"Code"**, then **"Download Zip"** and a *.Zip* file will be downloaded. There is no need to unpack the files.
2. Start up Blender and go to the feature bar: **"Edit"**, **"preference"**, **"Add-Ons"**, **"Install"**. Open the zip file from step 1.
3. Activate the add-on by selecting the add-on, which can be found under **"Community"**.
4. It is suggested to click **"Save User Setting"** before closing the settings.

A.1.2. Add-on: Poliigon (optional)

The Poliigon add-on is optional. This add-on helps to create a more realistic grass plane easily. Without this add-on, the grass plane will easily have a checkered look.

A.1.3. Installation of pipelines:

The pipelines made during the thesis can be imported, so it is not needed to recreate the pipelines. The pipelines can either be imported to your own project or you can use it from the example project. In order to import it to your own project the following steps are needed:

1. Download the example project: [fixme drive].
2. Open your own project and go to **"File"** in your feature bar, then click **"Append"** and a new window opens.
3. Select the just downloaded example project and click **"Append"**.
4. Click on the **"filter"** symbol and select the material properties under the Blender ID, shown in Figure A.2.
5. Four folders should appear: select the **"material"** folder to append the rendering pipelines and select **"NodeTree"** for the VTK-pipelines.
6. Add a workspace in the Feature bar of Blender, by clicking on the **+** symbol and selecting **"General"**, then **"Shading"**.
7. The new workspace is used for the VTK-pipelines, rename the workspace by double clicking on the name.
8. Go to the Editor bar and select the VTK editor, which is called **"BVTK Node tree"** and activate the node tree by clicking on **"+ New"** in the Browse node tree and now a **"B"** symbol appears which allows you to go through the different node trees you have created and added to the project.

A similar procedure is for the material, and there is already a workspace called **"Shading"** to add the rendering pipelines.

Tip: Protecting node trees

Non-activated node trees are automatically deleted by Blender when closing the software, and to make sure that does not happen, click on the **"fake user"**-symbol, which looks like a shield in the editor bar. When the fake-user is activated, i.e. it turns blue, the node tree is protected and will not be deleted by Blender.

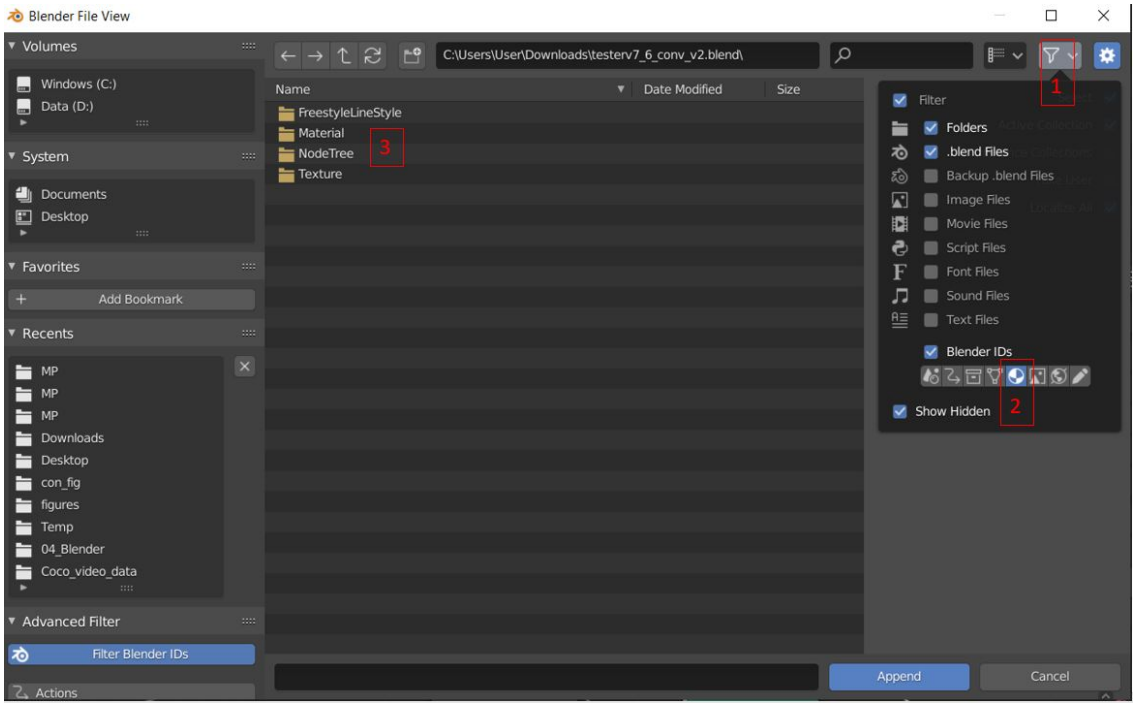


Figure A.2: Window of Blender 2.82 to append the needed pipelines to other projects.

A.2. Creating visuals

A.2.1. Basics

The get even more familiar to the Blender software, it is recommended to do a tutorial series which give you familiarity to the software within one day: https://youtu.be/NyJWoyVx_XI [fixme]. This guide is however still useful for those without any familiarity with Blender. The following basics are essential in Blender:

- *Rotating* the view in Blender: Click on the scroll of the mouse and move.
- *Moving* the view in Blender: Click shift + scroll and move the mouse.
- *Grabbing* an object: Select the object and press G to grab and move the object.
- *Scaling* an object: Select the object and press S to scale the object.
- *Rotating* an object: select the object and press R to rotate the object.

Tip: Move/scale/rotate an object in certain direct

To move/scale/rotate an object in certain direct press X, Y or Z after selecting the object and indication of the operation. For example: scaling the object in x-direction; (1) select object (2) click S (3) click X to scale in x-direction.

Viewport: There are different viewports available within Blender, which display the material of Blender in different shadings for various purposes. Each viewport will be explained:

- **Wireframe** shows only the edges of the material. For example, for editing mesh.
- **Solid** shows the solid form of the material.
- **Material preview** shows the material using the "evee" engine, which allows the user to see the material textures. For example, to see whether the correct texture is used. This mode is most often used.
- **Rendered** shows the material with the scene render engine for interactive rendering. For example, to see how the final renders looks before rendering the image. This mode cost most computational power.

Tip: Selecting the top half for 3D

It is recommended to use the x-ray mode of the view mode bar to select the entire top-half of an object, which allows the user to select the backside of the material. An example is shown in Figure A.3.

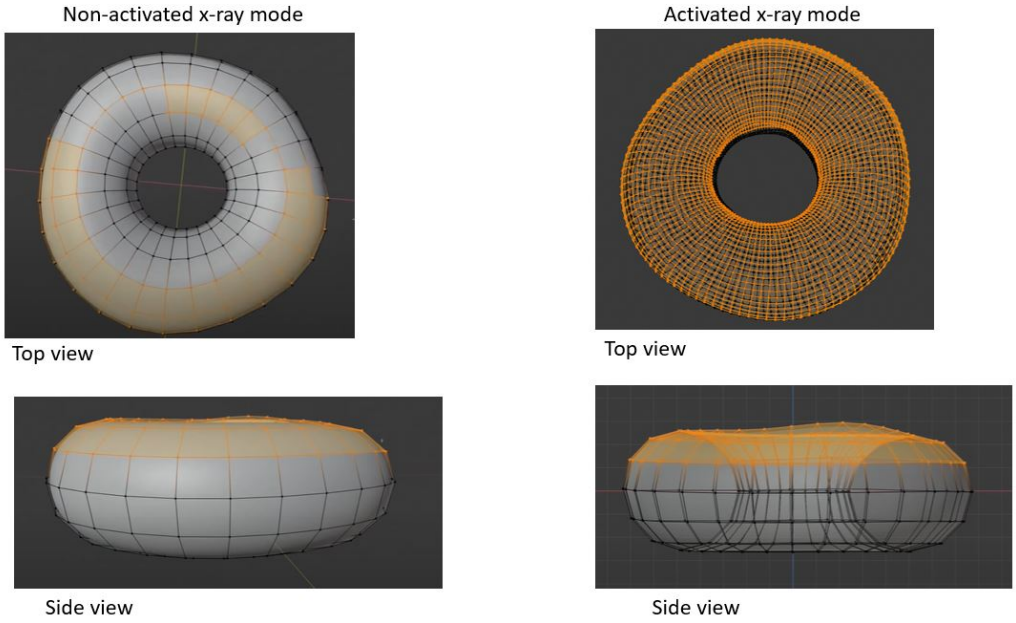


Figure A.3: A comparison of selecting the top half of an example object with (left) or without (right) the x-ray mode.

A.2.2. The first VTK-pipeline

The first VTK-pipeline creates a mesh out of an MPM dataset.

1. In the feature bar go to the VTK-pipeline workspace, in the example project it is called **"BVTK"**.
2. Select the correct VTK-pipeline (**"Delaunay mesh"**) in the browse node tree.
3. The first node of the VTK-pipeline, called *"vtkAVSucdReader"*, loads the data. Select the correct path to the mp-file of the MPM dataset by clicking on the **"file map"** symbol in the node.
4. Click **"Update"** in the next node, which is the info-node, to make sure the file is loaded correctly.
5. Go to the last node (*"VTK to Blender mesh"*) and give a name for the material and click **"Update"** to activate the whole VTK-pipeline. The drop-down menu in the second-last node (*"Color mapper"*) becomes active and a material-layer has been formed in Blender.
6. Select in the *"Color mapper"* node the correct attribute for illustration and a proper range of values.
7. Update the last node once again so it shows the correct material attribute.

Tip: Selecting the correct value range

The largest values of an attribute often appear at the end of the simulation, and to have the correct value range in the *"Color mapper"* node, open the last time step of the simulation in the vtk-pipeline and select **"Automatic range"** in the *"Color mapper"* node. It will automatically find the largest and smallest value, then deselect **"Automatic range"** to fix the value range.

A.2.3. Texturing the dyke

The output from the VTK-pipeline is shown in the scientific color range, which is not realistic. Attaching material to the values gives a more realistic look to the dyke. Three materials are given within the ex-

ample file: two different types of soil and a grass image. These files are essentially high-quality images and can be found and downloaded in [Fixme]. To attach the mesh to the materials, the following steps should be taken:

1. Go the *Shading* workspace and activate the material you want to change by clicking on the material. If activated, it shows an orange border in the material mesh, and an automatic generated rendering pipeline.
2. Select the correct rendering pipeline in the browse node tree, and it will automatically attach to the activated material. There are three options to choose from:
   - **"Db mix material"**: double mix material pipeline, which not only gives two materials to the dyke, it also highlights the weakening parts during the simulation with red.
   - **"Material"**: which gives two materials to the dyke without the red coloring.
   - **"Material transparency"**: Combine the material with different transparency, which will be explained later in the guide.
3. Make sure that the material is selected with the correct path in the *"Image texture"* node, which is the third node in the rendering pipeline.
4. Adjustments can be made to personal preference in the black and white color ramp by dragging the arrows.

A.2.4. Animation

Creating animation in Blender is relatively easy when the name of the files are easy to detect. Make sure that all time steps of one realization are within one file map. The filename should be consistent, for example; an MPM dataset is used with the realization number 2380, and it has 84 time steps, the first time step is then called: "MP_2380_1", and the second time step is called: "MP_2380_1" and so on.

1. Go to the animation workspace in the feature bar, and a timeline will appear in the bottom half of Blender.
2. Make sure the correct viewport is selected in Blender. It can be changed in the view mode and select **"Material preview"**. Moreover, make sure that the **"Auto-update"** is selected and **"Generate material"** is deselected in the VTK-pipeline.
3. Change the editor type to **"Timeline"** and change the number in the time frame.

A.2.5. 3-Dimensional

There are two options to make a 2D mesh to 3D; the extrusion method is easier to apply but not suitable for animation purposes as it has to be done manually. The second option takes more time as the 2D MP-files have to be re-written to 3D files. However, this option is suitable for animation and gives a more stable result.

Extrusion:

1. In the interaction menu (top-left) change the interaction mode from **"Object mode"** to **"Editor mode"**.
2. Select the material and click on **"Face"** and then **"Extrude Faces"** and drag the material to a preferred thickness and right click. A box in the bottom right corner will appear in which you can change the thickness to the exact number.

3D mesh:

The 3D mesh will be created in two parts; In the first part, the 2D MP-file will be rewritten to 3D, done in Python. The second part is similar to making a 2D dyke in Blender, done in the previous sections. The Python algorithm to transform the file is given within the download example package. The only input needed for the function is the path to the file. After rewriting the files, the VTK-pipeline can be used again within the Blender software, select the **Delaunay 3D mesh** node tree to create a 3D mesh. Note that this takes longer than making the mesh in 2D.

A.2.6. Transparency

Transparency helps the receiver to focus on certain aspects of the dyke. Within the research, two methods to create transparency are shown. The first method is giving transparency over the whole dyke layer, such that multiple layers of dykes can be shown. The second type is to attach transparency to a certain range within an attribute.

Method 1: transparency over the whole dyke body

1.

Method 2: transparency over a certain range of the dyke

1. Create and texture the dyke according to personal preferences.
2. Select the *Shading* workspace within the feature bar and the correct material node tree, which could also be imported from the example Blender file **"Material transparency"** or created by yourself.
3. Go to the node *"Principled BSDF"*, which should be attached to the material texture that will become transparent.
4. Go to the **Alpha**-value within the node and select an preferred value for the transparency. Note that the value is only between 0 and 1, and more towards 0 gives more transparency.

Tip: Duplicating node trees

Make a copy of the pipeline by clicking on the numbers left to the shield symbol (fake user) in the browse node tree bar. The number is an indication of how many objects are attached to the pipeline. If there is no object attached to the pipeline, there will be no number indication and deleted if it is not protected.

A.2.7. Adding grass

The grass is one of the environmental features of the dyke, which can be created once again in a simple manner or a more advanced method depending on the user's needs. For more photo-realistic visuals, the advanced method is recommended. However, it cost more computational power and more time to create the advanced version. The grass feature in both versions changes according to the simulation, and through time, i.e. the weakened parts of the dyke body are exposed and are not covered with any grass.

The grass layer is attached to an attribute. First, the correct attribute must be shown by creating a dyke body using the VTK-pipeline with the correct attribute. This thesis chooses to visualize the fracturing of the dyke layer using the plastic deviatoric strain.

Basic:

1. Create a dyke body using the VTK-pipeline (**"Delaunay Mesh"**) and make sure the plastic deviatoric strain attribute is selected and generates an automatic material, which could be deleted later.
2. Rotate the material such that only the top of the dyke is shown and select the top. Make sure that the x-ray mode is not activated.
3. Click Shift+ D to duplicate the top layer, which will function as the top layer.
4. Click Esc to snap the duplicated top layer back to position, which is on top of the dyke.
5. Click P to make the grass layer a separate object and give it a preferred name. The original dyke is not needed anymore for the grass layer and can be changed to the preferred attribute for the dyke.
6. Select the grass layer and attach the imported texture to it, which is called: *Grass*. Execute the procedure as described in Section A.2.3.

The basic grass layer is now formed, and since the attributes are attached to the object, during animation, the grass layer will change simultaneously with the dyke attributes. The advanced grass layer builds on top of the basic steps.

Advanced:

7. Select the grass and

A.2.8. Adding water

The main purpose of showing the water is an indication of where the water is. Water is also an environmental feature, which can be added in various ways. Therefore, it is unnecessary to make an advanced simulation of the water particle for static illustrations, and a textured box might be sufficient.

Basic: water plane

1. Click Shift + A and add a plane.
2. Attach the **Water** material texture to the plane.
3. Scale and place the water plane at the correct place.

B

Appendix B

B.1. Horizontal failure algorithm

This algorithm is used to detect horizontal failures within the RMPM dataset using the plastic deviatoric strain value; it runs through the dataset and outputs the realization number with horizontal failure and the type of failure.

The MPs are divided over a zeros matrix based on their location. For each MPM file, if the material point has a larger plastic deviatoric strain value than 0.5, a value of one is added to the matrix(m,n). This threshold value is set at 0.5 because MPM simulations have an oscillating behavior, so there will be error measurements when there is no threshold. Hereafter, a clustering algorithm is used to detect the different groups within the matrix i.e., connecting non-zeros values are labeled as one group. In the end, there are two conditions, where a simulation can be defined as a horizontal failure:

- Type 1: if there is one full row (m) that is non-zeros.
- Type 2: if there are two rows(m_1, m_2) that together make one row non-zeros.

Algorithm 1 Detection of horizontal failures

Data: MP-file for MPM(a)
Result: Which of the MP files are detected
initialization **for** *MPM (a)* **do**
  for *plastic deviatoric strain for each* MP_i **do**
    if MP_i *larger than 0.5* **then**
      $matrix_{m,n}$ add 1
    else
      Do nothing
    end
  end
  Search which clusters are connected to each other
  for *each row of the matrix* **do**
    if *one row in matrix(m,n) is non-zero* **then**
      Horizontal failure
    else if *two rows together are one row non-zeros* **then**
      Horizontal failure
    else
      No horizontal failure
    end
  end
end

References

[1] Hoogwaterbeschermingsprogramma. URL https://www.hwbp.nl/over-hwbp/wie-we-zijn-en-wat-we-doen/hwbp-en-hwbp2.

[2] Rijkswaterstaat. URL https://www.rijkswaterstaat.nl/water/waterbeheer/bescherming-tegen-het-water/waterkeringen/dijken.

[3] VTK Class reference. URL https://vtk.org/doc/nightly/html/annotated.html.

[4] UCD file, 2007. URL https://people.sc.fsu.edu/{~}jburkardt/data/ucd/ucd.html.

[5] Wolfgang Aigner, Silvia Miksch, Heidrum Schumann, and Christian Tominski. *Visualization of time-oriented data*. Springer, 2011. ISBN 9780857290786.

[6] Søren Mikkel Andersen and Lars Vabbersgaard Andersen. Modelling of landslides with the material-point method. *Computational Geosciences*, 14(1):137–147, 2010. ISSN 14200597.

[7] Søren Mikkel Andersen and Lars Vabbersgaard Andersen. post processing in the material point method, 2013.

[8] Raluca M. Andrei, Marco Callieri, Maria F. Zini, Tiziana Loni, Giuseppe Maraziti, Mike C. Pan, and Monica Zoppè. Intuitive representation of surface properties of biomolecules using BioBlender. *BMC Bioinformatics*, 13(SUPPL.4):1–12, 2012. ISSN 14712105. doi: 10.1186/1471-2105-13-S4-S16.

[9] Natalia Andrienko and Gennady Andrienko. *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media, 2006.

[10] Lars Beuth, Thomas Benz, Pieter A Vermeer, and Zdzisław Więckowski. Large deformation analysis using a quasi-static material point method. 2008.

[11] Jan Blinde, Cor Bisschop, Marieke De Visser, Ruben Jongejan, and Jan Tigchelaar. Factsheet: Afweging ter bepaling glijvlak voor faalmechanisme macrostabiliteit binnenwaarts (translated in English: Consideration for determining failure surface for failure mechanism macro-instability), 2018.

[12] Jeff Budsberg, Michael Losure, Ken Museth, and Matt Baer. Liquids in The Croods. *DigiPro*, 2013. URL http://www.museth.org/Ken/Publications{_}files/Budsberg-etal{_}DigiPro13.pdf.

[13] William M. Coombs, Tim J. Charlton, Michael Cortis, and Charles E. Augarde. Overcoming volumetric locking in material point methods. *Computer Methods in Applied Mechanics and Engineering*, 333:1–21, 2018. ISSN 00457825. doi: 10.1016/j.cma.2018.01.010. URL https://doi.org/10.1016/j.cma.2018.01.010.

[14] Jens Cornelis, Markus Ihmsen, Andreas Peer, and Matthias Teschner. IISPH-FLIP for incompressible fluids. *Computer Graphics Forum*, 33(2):255–262, 2014. ISSN 14678659. doi: 10.1111/cgf.12324.

[15] Tom de Gast, Philip J Vardon, and Michael A Hicks. Estimating spatial correlations under man-made structures on soft soils. In J Huang, Gordon A Fenton, L Zhang, and D V Griffiths, editors, *Proceedings of 6th International symposium on geotechnical safey and risk: Geo-risk 2017*, pages 382–389, Denver, Colorado, 2017. doi: 10.1061/9780784480717.036.

[16] Mengyuan Ding, Xuchen Han, Stephanie Wang, Theodore F. Gast, and Joseph M. Teran. A thermo-mechanical material point method for baking and cooking. *ACM Transactions on Graphics*, 38(6), 2019. ISSN 15577368. doi: 10.1145/3355089.3356537.

[17] Selan Dos Santos and Ken Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3):311–325, 2004.

[18] E James Fern, Dirk Arie de Lange, Cor Zwanenburg, Johannes Antonius Maria Teunissen, Alexander Rohe, and Kenichi Soga. Experimental and numerical investigations of dyke failures involving soft materials. *Engineering Geology*, 219:130–139, mar 2017. ISSN 0013-7952.

[19] E James Fern, Alexander Rohe, Kenichi Soga, and Eduardo E Alonso. *The Material Point Method for Geotechnical Engineering: a Practical Guide*. CRC Press, Boca Raton, FL, 2019.

[20] James D Foley, Andries van Dam, Steven k Feiner, and John Hughes. *computer graphics: principles and practise*. 2014.

[21] José Len González Acosta, Philip J Vardon, Guido Remmerswaal, and Michael A Hicks. An investigation of stress inaccuracies and proposed solution in the material point method. *Computational Mechanics*, 65:555–581, 2020. ISSN 1432-0924. doi: 10.1007/s00466-019-01783-3. URL https://doi.org/10.1007/s00466-019-01783-3.

[22] D V Griffiths, J Huang, and Gordon A Fenton. Probabilistic slope stability analysis using RFEM with non - stationary random fields. *Geotechnical Safety and Risk V*, (January):1–6, 2015.

[23] Robert B Haber and David A McNabb. Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing*, 74:93, 1990.

[24] Charles Hansen and Christopher R Johnson. *The visualization handbook*. 2013.

[25] Coenraad Hartsuijker and Johannes Wijnand Welleman. *Engineering Mechanics: Volume 2: Stresses, Strains, Displacements*, volume 2. Springer Science & Business Media, 2007.

[26] J Heinrich and D Weiskopf. State of the Art of Parallel Coordinates. *Eurographics Conference on Visualization (EuroVis)*, pages 95–116, 2013.

[27] M A Hicks and W A Spencer. Computers and Geotechnics Influence of heterogeneity on the reliability and failure of a long 3D slope. *Computers and Geotechnics*, 37(7-8):948–955, 2010. ISSN 0266-352X. doi: 10.1016/j.compgeo.2010.08.001. URL http://dx.doi.org/10.1016/j.compgeo.2010.08.001.

[28] Michael A. Hicks and Kristinah Samy. Influence of heterogeneity on undrained clay slope stability. *Quarterly Journal of Engineering Geology and Hydrogeology*, 35(1):41–49, 2002. ISSN 14709236. doi: 10.1144/qjegh.35.1.41.

[29] J Huang, D V Griffiths, and Gordon A Fenton. System Reliability of Slopes by RFEM. *SOILS AND FOUNDATIONS*, 50(3):343–353, 2010. ISSN 1881-1418.

[30] Silvano Imboden and Lorenzo Celli. VTK for Blender, Blender conference, 2018. URL https://youtu.be/KcF4LBTTyvk.

[31] Silvano Imboden, Lorenzo Celli, and Paul Mcmanus. Bvtknodes, 2020. URL https://bvtknodes.readthedocs.io/en/latest/BVTKNodes.html.

[32] Alfred Inselberg. *Parallel coordinates: Visual multidimensional geometry and its applications*. 2009. ISBN 9780387215075. doi: 10.1007/978-0-387-68628-8.

[33] Ruben Jongejan, Marcel Bottema, and Robert Slomp. WBI 2017, Code calibration. *ICFM7 Leeds Presentation*, (September), 2017.

[34] Sebastiaan N Jonkman, R D J M Steenbergen, O Morales-Napoles, A C W M Vrouwenvelder, and J K Vrijling. *Probabilistic Design: Risk and Reliability Analysis in Civil Engineering*. Delft University of Technology, Delft, fourth edition, 2017. ISBN 9780784412558.

[35] Sebastiaan N Jonkman, R.E. Jorissen, Timo Schweckendiek, and J.P. van den Bos. FLOOD DEFENCES L ECTURE NOTES CIE5314. 3rd editio:376, 2018.

[36] Bert Kappe, Tom den Boer, and Kirsten Jeurink. Dijken voor beginners. page 68, 2021. URL https://www.runinfo.nl/schemabeginners.htm.

[37] D. Kretz, S. Callau-Monje, M. Hitschler, A. Hien, M. Raedle, and J. Hesser. Discrete element method (DEM) simulation and validation of a screw feeder system. *Powder Technology*, 287:131–138, 2016. ISSN 1873328X. doi: 10.1016/j.powtec.2015.09.038. URL http://dx.doi.org/10.1016/j.powtec.2015.09.038.

[38] M. Lloret-Cabot, G. A. Fenton, and M. A. Hicks. On the estimation of scale of fluctuation in geostatistics. *Georisk*, 8(2):129–140, 2014. ISSN 17499526. doi: 10.1080/17499518.2013.871189. URL http://dx.doi.org/10.1080/17499518.2013.871189.

[39] Robert Magee. *Houdini Foundation - For film, tv & gamedev*. 2019. ISBN 9781775333807.

[40] K Mario. The curse of dimensionality. 1.

[41] Bruce H McCormick, Thomas A DeFanti, and MD Brown. Visualization in scientific computing and computer graphics. In *ACM SIGGRAPH*, volume 21, page 95, 1987.

[42] Ken Museth. VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics*, 32(3), 2013. ISSN 07300301. doi: 10.1145/2487228.2487235.

[43] Ken Museth. A Flexible Image Processing Approach to the Surfacing of Particle-Based Fluid Animation (Invited Talk). pages 81–84, 2014. doi: 10.1007/978-4-431-55007-5_11.

[44] Ken Museth, Peter Cucka, Milhai Alden, and David Hill. OpenVDB documentation. URL https://academysoftwarefoundation.github.io/openvdb/.

[45] Paul J T A T T Naas. Autodesk Maya 2014 essentials, 2013. URL http://site.ebrary.com/id/10718855.

[46] M. Novotny. Visually Effective Information Visualization of Large Data. *8th Central European Seminar on Computer Graphics (CESCG 2004)*, pages 41–48, 2004.

[47] Matej Novotný and Helwig Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006. ISSN 10772626. doi: 10.1109/TVCG.2006.170.

[48] S Rebay. 16_Efficient_Mesh_Rebay.pdf, 1993. URL http://www.sciencedirect.com/science/article/pii/S0021999183710971.

[49] Guido Remmerswaal. *Development and implementation of moving boundary conditions in the Material Point Method (MSc thesis)*. Msc thesis, Delft University of Technology, 2017. URL https://repository.tudelft.nl/islandora/object/uuid{%}3Ad5d89599-1e45-4ac1-b966-49a1a785a350.

[50] Guido Remmerswaal, Michael A Hicks, and Philip J Vardon. Influence of Residual Dyke Strength on Dyke Reliability Using the Random Material Point Method. In *Proceedings of the 7th International Symposium on Geotechnical Safety and Risk (ISGSR)*, pages 775–780, Taipei, 2019. ISBN 9789811127250.

[51] Guido Remmerswaal, Michael A Hicks, and Philip J Vardon. Slope Reliability and Failure Analysis using Random Fields. In E James Fern, Alexander Rohe, Kenichi Soga, and Eduardo Alonso, editors, *The Material Point Method for Geotechnical Engineering*, chapter 16, pages 287–310. 1st edition, 2019.

[52] Guido Remmerswaal, Philip J. Vardon, and Michael A. Hicks. Evaluating residual dyke resistance using the Random Material Point Method. *Computers and Geotechnics*, 133(February):104034, 2021. ISSN 18737633. doi: 10.1016/j.compgeo.2021.104034. URL https://doi.org/10.1016/j.compgeo.2021.104034.

[53] Timo Schweckendiek, Mark G van der Krogt, Ben Rijneveld, and Ana Martins Teixeira. Handreiking Faalkansanalyse Macrostabiliteit Handreiking Faalkansanalyse Macrostabiliteit. *Deltares*, (version 3), 2017.

[54] James Albert Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999.

[55] Marc Steinberg. Realism in the Animation Media Environment. *Animating Film Theory*, (March): 287–300, 2014. doi: 10.1215/9780822376811-017.

[56] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Snelle. A material point method for snow simulation. 2013. doi: 10.1109/DSN.2008.4630068.

[57] Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. Augmented mpm for phase-change and varied materials. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014.

[58] D Sulsky, Z Chenb, and H L Schreyer ¢'. A particle method for history-dependent materials. Technical report, 1994.

[59] Robert 't Hart, Huub de Bruijn, and Goaitske de Vries. Fenomenologische beschrijving - Faalmechanismen WTI (translated in English: Phenomenological description: Failure mechanisms WTI), 2016.

[60] Tamas Umenhoffer. Simulating Snow with the Material Point Method. 2015.

[61] A P van den Eijnden and Michael A Hicks. Efficient subset simulation for evaluating the modes of improbable slope failure. *Computers and Geotechnics*, 88:267–280, 2017. ISSN 18737633.

[62] A.w. van der Meer. POVM actuele sterkte, macrostabiliteit buitenwaarts een verkennende studie. (april), 2017.

[63] Philip J Vardon, Bin Wang, and Michael A Hicks. Slope failure with the material point method : An investigation of post-peak material behaviour. In *Proceedings of the 15th International Conference of the International Association for Computer Methods and Advances in Geomechanics*, page ??, 2017.

[64] Arnold Verruijt. *Soil Mechanics*. 2001. ISBN 9780123693969. doi: 10.1016/B0-12-369396-9/00228-8.

[65] Bin Wang, Michael A Hicks, and Philip J Vardon. Slope failure analysis using the random material point method. *Géotechnique Letters*, 6(2):113–118, jun 2016. ISSN 2045-2543.

[66] Bin Wang, Philip J Vardon, Michael A Hicks, and Zhen Chen. Development of an implicit material point method for geotechnical applications. *Computers and Geotechnics*, 71:159–167, 2016. ISSN 18737633.

[67] Stephanie Wang, Mengyuan Ding, Theodore F. Gast, Leyi Zhu, Steven Gagniere, Chenfanfu Jiang, and Joseph M. Teran. Simulation and Visualization of Ductile Fracture with the Material Point Method. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(2):1–20, 2019. doi: 10.1145/3340259.

[68] Chris Want. VTK Blender code, 2017. URL https://github.com/cwant/VTKBlender.

[69] Kevin Warwick and Miroslav Kárný. *Computer-intensive methods in control and signal processing*. Springer, New York, second edition, 1997. ISBN 9781461273738.

[70] Waterwet. Waterlaw: BWBR0025458, 2009. URL http://wetten.overheid.nl/BWBR0025458/2018-01-01{#}.

[71] Edward J Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[72] Joshuah Wolper, Yu Fang, Minchen Li, Jiecong Lu, Ming Gao, and Chenfanfu Jiang. CD-MPM: Continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics*, 38(4), 2019. ISSN 15577368. doi: 10.1145/3306346.3322949.

[73] Pak Chung Wong and R Daniel Bergeron. 30 years of multidimensional multivariate visualization. *Scientific Visualization*, 2:3–33, 1994.

[74] Joel Wretborn, Rickard Armiento, and Ken Museth. Animation of crack propagation by means of an extended multi-body solver for the material point method. *Computers and Graphics (Pergamon)*, 69:131–139, 2017. ISSN 00978493. doi: 10.1016/j.cag.2017.10.005.

[75] Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. Continuum foam: A material point method for shear-dependent flows. *ACM Transactions on Graphics*, 34(5), 2015. ISSN 15577368. doi: 10.1145/2751541.