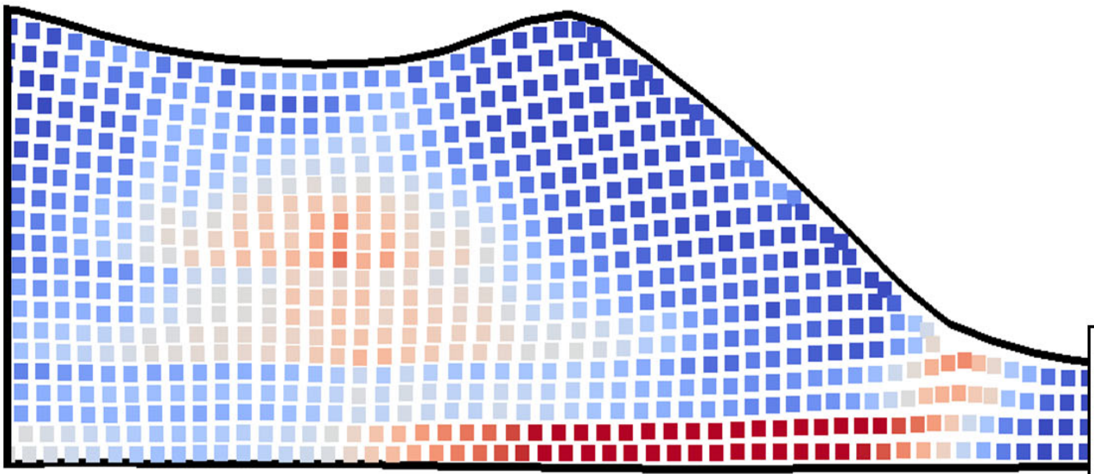
Development and implementation of moving boundary conditions in the Material Point Method

G. Remmerswaal

Technische Universiteit Delft



TU Delft

Development and implementation of moving boundary conditions in the Material Point Method

by

G. Remmerswaal

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday August 24, 2017 at 2:00 PM.

Student number:	4144503	
Project duration:	March, 2016 – August, 2017	
Thesis committee:	Dr. P.J. Vardon,	TU Delft, supervisor
	Prof. dr. M.A. Hicks,	TU Delft
	Dr. ir. R.J. Labeur,	TU Delft
	J.L. Gonzalez Acosta,	TU Delft

This thesis is confidential and cannot be made public until August 23, 2017.

Preface

This master thesis is made as a completion of the master track Geo-Engineering from the Civil Engineering master at TU Delft. After a process of more then a year encountering many positive and negative moments, I am proud to present this thesis. I have been able to get an insight in the Material Point Method and used this insight to design and test a new edge detection method for MPM.

During the past year I have been supported frequently by my head supervisor Phil Vardon. I would like to thank him for our conversations and his support, which kept me focussed on my work and helped me solve the difficult parts of the thesis. I would also like to thank Michael Hicks and Robert Jan Labeur for their time and input through out the entire period. Moreover, I would like to thank Leon Gonzalez Acosta, for the provided technical support regarding the implementation in MPM. Finally I would like to thank Michael Cortis for his assistance with the Implicit Boundary Method.

In my personal life I would like to thank Frank Remmerswaal, Heleen Remmerswaal-de Graaf and Lisette van der Sar for always being supportive and helpful. You will help me in every situation, even the most difficult ones.

G. Remmerswaal
Delft, July 2017

Contents

1

Abstract

A new technique is developed in this thesis, which applies boundary conditions to moving boundaries in the Material Point Method (MPM). While MPM has been proven to be useful in slope stability, foundation and seabed ploughing modelling, the application of boundary conditions is still a challenge, because the location of the boundary is unknown in MPM. The location of the boundary is unknown, due to the fact that the material points, by definition, are not located at the boundary. Boundary conditions have currently been applied to fixed boundaries with a known location. This fixed boundary preferably coincides with the background grid of MPM, in which case the boundary condition can be applied directly onto the background grid. In this thesis the concept of applying boundary conditions to the background grid has been expanded to ’irregular’ boundaries, which do not coincide with the background grid. This technique can thus also be used for moving boundary conditions. Therefore, the research question of this thesis is: Can boundary conditions on moving boundaries be appropriately applied to the background mesh of an MPM in slope stability problems?

The location of the boundary is not exactly defined by current versions of MPM. Thus an edge detection method was constructed to locate the boundary based on the information provided by the MPM model. The Volume of Fluid method (VOF) and the Surface Marker Method (SMM), two edge detection methods used in fluid dynamics, have been tested. Moreover, a new edge detection method has been developed in this thesis. This method is a new implementation of the Level Set Method, which uses a composite Bézier curve to define the boundary. The method has been called the Proximity Field Method (PFM). PFM locates the boundary based on the distance towards a material point, VOF uses the volume of material at a location to determine the location of boundaries and SMM places fake material points at the initial boundary to track it.

The accuracy of VOF proved to be too small for the application of boundary conditions. SMM gave a better representation of the boundary at small strains in comparison to PFM. However, the main advantage of MPM is the possibility of large strain modelling, which SMM could not handle. PFM was able to handle these larger strains and was therefore chosen as the basis for the application of boundary conditions.

PFM’s representation of the boundary had to be improved to be useful in the application of boundary conditions. Smoothing of the surface together with an equal spacing of the material points in the initial condition have improved the representation of the boundary. Moreover, the computation cost had to be reduced, because it was to high after the initial implementation. Within this thesis the computation cost has already been reduced by a factor of 30 and is now equal to the computation cost of an explicit MPM timestep.

Finally PFM has been used to apply a surface traction to MPM. The surface traction must therefore be distributed from the boundary to the nodes of the background grid. This technique has been tested by applying a load to the top of a soil slope. As intended, the load is transferred to the background grid and changes according to the location of the boundary. As expected, the deformation of the slope is increased due to the application of the load. So, the concept of applying boundary conditions on moving boundaries to the background grid has been proven to work. However, additional research is required to determine the accuracy of PFM.

2

Introduction

2.1. Background

The Material Point Method (MPM) is an attempt to use the advantages of both Lagrangian and Eulerian methods by combining the two approaches (Sulsky et al. [34], Sulsky and Schreyer [33]). The numerical dissipation associated with an Eulerian method is removed with a Lagrangian step and the mesh distortion of a Lagrangian mesh is avoided by a convection step. MPM can also be used for grain modelling where it attempts to find a correct mixture between a micromechanical and distinct element approach, to allow for both grain deformation according to continuum constitutive modelling (for which a distinct element approach is a good representation) and grain interactions, such as rolling, sliding and separation (for which a more micromechanical model is needed) (Bardenhagen et al. [3]). MPM modelling is an extention of Finite Element Modeling (FEM), which has been proven to be useful in many soil deformation problems (Smith et al. [31], Esch et al. [16], Felippa [17]). However, modelling individual particles and large strain soil deformations using FEM results in problems regarding the distortion of the solution grid as well as incorrect grain interactions (Ma et al. [21], Phuong et al. [25]). Moreover, generating body fixed FEM meshes can be a complex and time consuming problem in realistic solutions (Bardenhagen and Kober [2]).

2.2. Summary of MPM

To solve all the problems regarding MPM, the MPM algorithm has been developed based on the particle in cell method (FLIP), which combines the Eulerian and Lagrangian descriptions of the material. The granular body is discretized in a number of material points (defined in an Eulerian system), which is connected to a Lagrangian background mesh. Equations of motion are solved on the background mesh, which in is then projected back onto the material points. Afterwards, the background mesh is reset to the initial condition, while the material points store their deformations (as shown in Figure 2.1). FLIP has already been proven to be capable of using this technique to model hydrodynamical problems, by transferring only the nodal velocities from the material points towards the background mesh and vice versa. However, to be able to express soil mechanical problems, which are often history dependent, it is convenient to also transfer the deformation gradient and stress (Bardenhagen et al. [3]). The addition of deformation gradient and stress to the FLIP algorithm has been called MPM. The unmovable mesh of an MPM algorithm cannot be distorted, so any problems regarding a distorted mesh are solved (Sulsky and Schreyer [33], Bardenhagen et al. [3]). Therefore, reordering of the mesh in large strain problems is not necessary. Moreover, grain interaction can be modelled better due to the individual movement of the material points, which can for example be useful in cracking or tearing of the material (Ma et al. [21], Phuong et al. [25], Chen et al. [8]). Finally, generation of a material domain is simpler due to the fact that material points can fill any shape specified by the problem geometry without creating irregular mesh problems. Currently slope stability, foundation modelling and seabed ploughing are a few of the problems in which MPM has been tested and has been proven to be useful (Wang et al. [37], Wang et al. [38], Vardon et al. [35], Cortis et al. [10] and Cortis et al. [11]). As shown by Wang et al, MPM can even be used for modelling of rainfall induced slope failure or retrogressive and progressive slope failure mechanism (Wang et al. [38] and Wang et al. [36]).

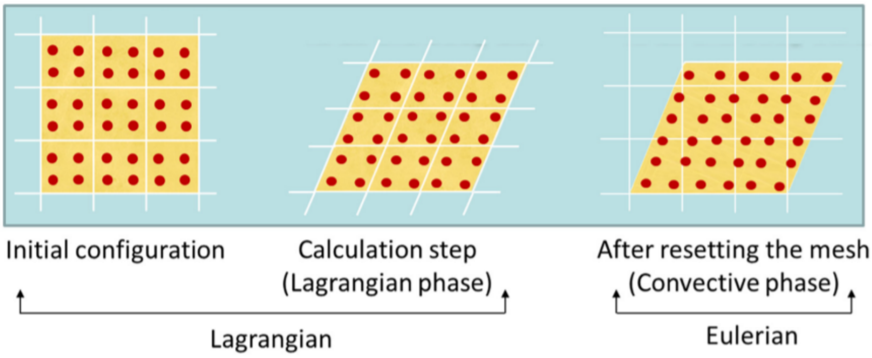


Figure 2.1: MPM displacement Algorithm (Phuong et al. [25])

2.3. Problem description

Applying boundary conditions to moving surfaces of an MPM mesh has proven to be a challenge. In FEM the edge of the material mesh represents the edge of the material domain (Cortis et al. [10]). Boundary conditions can therefore be applied to the FEM mesh. In MPM material domain boundaries, which coincide with the background grid (called 'regular' boundaries in this thesis), can be applied with a similar method used in FEM, namely application of the condition to the background grid, which is comparable to the material mesh. However, many material domain boundaries do not coincide with the background grid (called 'irregular' boundaries in this thesis). These irregular boundary conditions cannot be applied directly to the background grid, because the material edge is not located at this grid. Moreover, the boundary cannot be applied to the material points, due to the fact that the material points are, by definition, not located at the boundary. Developing a solution to this problem is the main focus of this thesis.

Currently boundary conditions in MPM are applied to the material points closest to the boundary, which leads to a loss in accuracy (Cortis et al. [11], Chen et al. [8] and Wang et al. [36]). For example, during the modelling of a rainfall induced slope failure, Wang et al. [36] applied a zero pore pressure boundary to the material points located in the surface background mesh elements. Using this procedure, the zero pore pressure boundary is distributed over a boundary layer with the thickness of roughly one background mesh element, due to the fact that the condition is mapped to all the material points in the surface elements. Moreover, the surface material points are not updated, so the material points initially located in surface elements are maintained as surface points during the entire simulation (Wang et al. [36]). However, the surface points are not bounded to the surface and non-surface points could in fact be closer to the boundary after deformation. Therefore, the boundary condition is not guaranteed to be applied at the surface. It is necessary to either update the surface points to ensure that the boundary is applied to the material points located closest to the surface or not use the material points to apply boundary conditions at all. The application of boundary conditions to material points in a boundary layer has been used to apply surface tractions as well (Chen et al. [8]). Similar problems as in Wang et al. [36] have been encountered with the application of surface tractions.

Only recently, research has started on applying irregular fixed boundary conditions to the background grid in an implicit MPM model (Cortis et al. [11]). Cortis et al. [10] applied a fixed displacement boundary condition at a known location to an implicit MPM model using the Implicit Boundary Method. These boundaries did not coincide with the background mesh, but their location was known during the complete computation. Within this thesis the ideas of Cortis et al. [10], are expanded to moving irregular boundary conditions. This concept might then be used to apply boundary conditions to the surface of a failing soil slope. For example, the zero pore pressure boundary applied by Wang et al. [36] in rainfall induced slope failure, could be applied to the surface without the problems mentioned previously. The main concept is applying a distributed or transformed boundary condition to the background grid based on the location of the domain boundary within the background grid. The application area of this thesis lies on slope stability problems following the work of Wang et al. (Wang et al. [37],Wang et al. [38],Wang et al. [36] and Vardon et al. [35]). So in short the main research question of this thesis is: Can boundary conditions on moving boundaries be appropriately applied to the background mesh of an MPM in slope stability problems?

As mentioned, the material points closest to the surface are not guaranteed to be closest to the surface during the computation. During the literature review, provided in Chapter 3, it was observed that this problem can only be solved by locating the moving boundary. The boundary must therefore be detected to apply the boundary conditions, which can be done with an edge detection method. Therefore a second research question has arisen: Which edge detection method is most suitable for detecting a material boundary in an MPM?

2.4. Objectives

To summarize the previous section, two research questions are investigated within this thesis, namely:

- Can boundary conditions on moving boundaries be appropriately applied to the background mesh of an MPM in slope stability problems?
- Which edge detection method is most suitable for detecting a material boundary in an MPM?

2.5. Outline of the thesis

The thesis consist of seven chapters including this introduction and the abstract presented previously. During this introduction the related literature has been introduced. A detailed description of the related literature is provided in the next chapter. In that chapter the previous research on the application of boundary conditions in MPM and FEM is analysed. At first, FEM is investigated, as it is the foundation of MPM. Moreover, the application of boundary conditions in MPM is expected to be similar to FEM and the concepts of FEM must therefore be clear. Afterwards, the background on MPM is given with a focus on the boundary conditions. Finally, a review has been made of edge detection as it is an important part of the thesis.

In Chapter 4 two existing edge detection methods have been implemented, namely the Surface Marker Method and the Volume of Fluid method. Moreover, a third method, the Proximity Field Method, has been developed in Chapter 4. The usability of the three methods for MPM is tested in Chapter 4 to answer the second research question: Which edge detection method is most suitable for detecting a material boundary in an MPM?

The most usable method has been improved and tested in further detail in Chapter 5. Among other things the stability and computation cost have been improved.

Thereafter, the implementation of a surface traction boundary condition for moving irregular boundaries is discussed in Chapter 6. The application of this surface traction is tested with a bearing capacity problem as well as an additional load on a failing slope. Both of these tests have been compared with FEM during the first seconds of deformation.

Finally, a summary of the conclusions is provided and the research questions are answered in Chapter 7.

3

Related literature and theoretical background

As mentioned in the introduction an attempt has been made in this thesis to apply boundary conditions to 'irregular' boundaries, in the Material Point Method (MPM). Irregular boundaries are boundaries, which do not coincide with the background grid of MPM. Due to convection of the material points through the background grid the boundaries of the soil domain are not guaranteed to coincide with the background grid, resulting in the occurrence of irregular boundaries. Only limited research has been performed on applying conditions to irregular boundaries. In contrast extended research has been performed on applying conditions to regular boundaries, which do coincide with the background grid. Namely, these regular boundary conditions can be applied using the techniques needed to apply boundary conditions in the Finite Element Method (FEM), as MPM is an extension of FEM. Unfortunately these techniques cannot be used directly to apply boundary conditions to irregular boundaries. Therefore, a new technique must be developed.

To develop new techniques for application of irregular boundary conditions or extend the techniques for application of regular boundary conditions, an analysis of FEM is required. Therefore, this chapter starts with an analysis of FEM with a focus on boundary conditions. This analysis of FEM is then used to understand MPM as well as the application of regular boundary conditions in MPM. Finally the background information on both FEM and MPM is used to understand and expand the ideas of the application of irregular boundary conditions. A two phase FEM is discussed, as one of the application areas of this thesis is rainfall induced soil failure, for which a two phase FEM is required.

Moreover, as mentioned in the introduction the material boundary, i.e. the location where boundary conditions are applied, must be detected. An edge detection method for MPM has not been developed. Due to the fact that it is a requirement for the application of boundary conditions, the development of an edge detection method for MPM is part of this thesis. Therefore, a selective review of the existing edge detection methods in literature has been provided in the the second part of this chapter. The methods are reviewed on their applicability for MPM.

3.1. Selective review of FEM

Many partial differential equations can be solved by discretization of the equations of motions in space and time. FEM is one implementation of a discretization of these differential equations. The model is a basis for the MPM model, the main focus of this thesis, and is therefore analysed within this section. FEM is described based on the information found mainly in Smith et al. [31]. However, a comparison is made with some of the other FEM descriptions. In FEM the problem geometry is divided into elements, called an element grid or mesh (Smith et al. [31]). More specified towards this thesis, a 2D soil body will be divided into elements, which are connected at the nodes, for example the soil slope of Figure 3.1. Due to the discretization into elements all propagation problems can be solved using a similar equation.

Within soil mechanics both mechanical and hydro-mechanical descriptions are required, due to the fact that a soil is a two phase material and often contains a fluid as the second phase. The mechanical description is given in the first section as it is required as a basis for the hydro-mechanical description. Furthermore, this thesis mainly focuses on mechanical MPM models, but two of its possible applications lie in the analysis of soil slopes in underwater condition and rainfall induced slope failure and the hydro-mechanical description is required for both. In future research the program designed within this thesis must be expandable to a hydro-mechanical MPM and some information on hydro-mechanical descriptions is therefore given within this chapter.

A short review is made of the difference between implicit and explict time integration schemes as the application of boundary conditions is different in both schemes. The new techniques needed to apply boundary conditions should be applicable to both schemes or one technique should be available per scheme. This review has been placed in Section 3.1.3. In the second to last paragraph of this section the application of boundary conditions in FEM is discussed separately from the general information on FEM, due to the fact that the main focus of this thesis is the application of boundary conditions. Finally most of the formulations are given only for a couple of primary variables in this thesis. However, different primary variables can be used in the discretization process. An explanation for using different primary variables and their impact on the boundary conditions is given in the last paragraph.

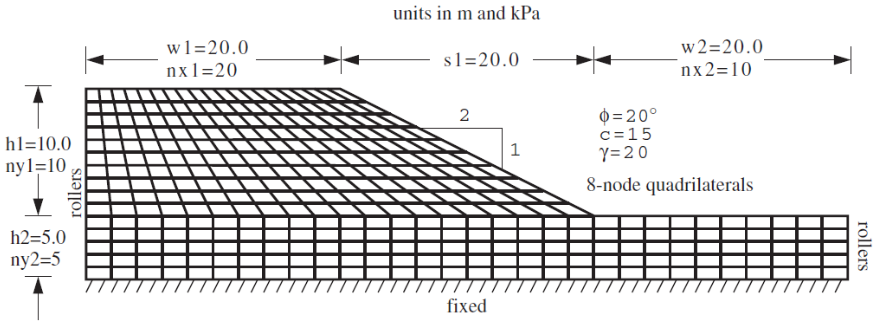


Figure 3.1: FEM Discretization of a soil slope (Smith et al. [31])

3.1.1. Mechanical FEM

The mechanical FEM describes the displacement due to the boundary conditions and the applied forced on the nodes of the FEM grid. The spatial discretization of a propagation problem with linear material properties results in (Section 3, Smith et al. [31]):

$$[\mathbf{k_m}]\{\mathbf{u}\}+[\mathbf{m_m}]\frac{d^2\{\mathbf{u}\}}{dt^2}=\{\mathbf{f}(\mathbf{t})\} \tag{3.1}$$

where $[\mathbf{k_m}]$ is the element stiffness matrix, $[\mathbf{m_m}]$ is the element mass matrix, $\{\mathbf{u}\}$ is the nodal displacement vector and $\{\mathbf{f}(\mathbf{t})\}$ are the nodal forces. The element stiffness matrix $[\mathbf{k_m}]$ is given by:

$$[\mathbf{k_m}]=\int\int[\mathbf{B}]^T[\mathbf{D}][\mathbf{B}]dxdy \tag{3.2}$$

where $[\mathbf{D}]$ is the elastic matrix and $[\mathbf{B}]$ contains the derivatives of the shape function. The element mass matrix $[\mathbf{m_m}]$ is given by:

$$[\mathbf{m_m}]=\rho\int\int[\mathbf{N}]^T[\mathbf{N}]dxdy \tag{3.3}$$

where $[\mathbf{N}]$ is the shape function matrix of this element and ρ is the density of the element. In case of a static equilibrium a reduced form of equation (3.1) can be used as the second-order time-dependency will not be required. This presents a standard stress-strain relationship:

$$[\mathbf{k_m}]\{\mathbf{u}\}=\{\mathbf{f}\} \tag{3.4}$$

The dynamic and static equilibrium discretizations need to be changed to account for the material non-linearity. This is achieved by not using the displacement u as the primary variable, but the change in displacement $\Delta u=\bar{u}=u_{t+\Delta t}-u_t$ (Cundall [12]). For the dynamic form this results in:

$$[\mathbf{k_{NL}}]\{\bar{\mathbf{u}}\}+[\mathbf{m_m}]\left.\frac{d^2\{\mathbf{u}\}}{dt^2}\right._{t+\Delta t}=\{\mathbf{f^{ext}_{t+\Delta t}}\}-\{\mathbf{f^{int}_t}\} \tag{3.5}$$

which reduces to the quasi-static form:

$$[\mathbf{k_{NL}}]\{\bar{\mathbf{u}}\} = \{\mathbf{f^{ext}_{t+\Delta t}}\} - \{\mathbf{f^{int}_{t}}\} \tag{3.6}$$

In which $[\mathbf{k_{NL}}]$ is a non-linear stiffness matrix updated in each step, $\{\mathbf{f^{ext}_{t+\Delta t}}\}$ is the external force applied on the system in this step and $\{\mathbf{f^{int}_{t}}\}$ is the internal force acquired from the initial condition and the previous steps. These element equations can be expanded to the complete soil body by a summation over each element, resulting in:

$$[\mathbf{K_{NL}}]\{\bar{\mathbf{u}}\} + [\mathbf{M}]\frac{d^2\{\mathbf{u}\}}{dt^2}{}_{t+\Delta t} = \{\mathbf{F^{ext}_{t+\Delta t}}\} - \{\mathbf{F^{int}_{t}}\} \tag{3.7}$$

where $[\mathbf{K_{NL}}]$,$\{\bar{\mathbf{u}}\}$,$[\mathbf{M}]$,$\{\mathbf{F^{ext}_{t+\Delta t}}\}$ and $\{\mathbf{F^{int}_{t}}\}$ represent the global versions of the element matrices described before. Detailed descriptions on the assembly of these global matrices can be found in (Smith et al. [31]).

Equation (3.1) can be expanded to also include a damping term. This can either be done by introducing an additional force (Cundall [12]) or an additional term on the left hand side (Section 3, Smith et al. [31]):

$$\begin{gathered}
[\mathbf{K_{NL}}]\{\bar{\mathbf{u}}\} + [\mathbf{C}]\frac{d\{\mathbf{u}\}}{dt} + [\mathbf{M}]\frac{d^2\{\mathbf{u}\}}{dt^2} = \{\mathbf{F^{ext}_{t+\Delta t}}\} - \{\mathbf{F^{int}_{t}}\} \\
[\mathbf{K_{NL}}]\{\bar{\mathbf{u}}\} + [\mathbf{M}]\frac{d^2\{\mathbf{u}\}}{dt^2} = \{\mathbf{F^{ext}_{t+\Delta t}}\} - \{\mathbf{F^{int}_{t}}\} + \{\mathbf{F^{damp}_{t+\Delta t}}\}
\end{gathered} \tag{3.8}$$

This damping could for example be the result of internal or external friction. $[\mathbf{C}]$ is the global damping matrix assembled from $[\mathbf{c_m}]$, which is the element damping matrix and is often assumed to be a linear function of the stiffness and mass matrices:

$$[\mathbf{c_m}] = f_m[\mathbf{m_m}] + f_k[\mathbf{k_m}] \tag{3.9}$$

where f_m and f_k are scalars. Similar descriptions as equation (3.8) are mentioned in different literature, for example, Zienkiewicz mentions the following residual form (Section 2, Zienkiewicz et al. [43]):

$$\{\mathbf{F(t)}\} - [\mathbf{M_m}]\frac{d^2\{\mathbf{u}\}}{dt^2} - P = 0 \tag{3.10}$$

where P is a function of the displacement and represents the stress within the solid. This is incorrect for non-linear solutions in which case P needs to be changed to include this non-linearity. So in fact P is similar to $[\mathbf{K_{NL}}]\{\mathbf{u}\} + [\mathbf{C}]\frac{d\{\mathbf{u}\}}{dt}$ and equations (3.8) and (3.10) are therefore the same.

Using shape functions a local coordinate system can be transformed to a global coordinate system. The local coordinate system of every element has the same size and shape and the elemental stiffness relation given in, for example, equation (3.1) for each element is therefore the same. Additional information on shape functions can be found in Section 3.2 of Smith et al. [31].

Equation (3.2) is numerically integrated for each element using the Gauss-Legendre quadrature (Lowan et al. [20]). Combining this numerical integration and the transformation to local coordinates for equation (3.2) results in (Section 3, Smith et al. [31]):

$$[\mathbf{k_m}] = \sum_{i=1}^{nip} W_i \det|[\mathbf{J}]|_i([\mathbf{B}]^T[\mathbf{D}][\mathbf{B}])_i \tag{3.11}$$

where $[\mathbf{J}]_i$ is the Jacobian matrix at integration point i, nip are the number of integration points per element and w_i is the weight associated with integration point i. Calculating the element stiffness matrix for each element and combining the matrices per node results in a global stiffness matrix. The same technique can be applied to the element mass matrix.

3.1.2. Two phase material FEM

In soil mechanics a hydro-mechanical FEM describes the flow of the pore fluid due to the applied boundary conditions in combination with the effect of this pore fluid on the deformation process. As mentioned previously the information on a hydro-mechanical FEM is included in this thesis, because rainfall induced slope failure is one of the likely application areas of this thesis (Wang et al. [36]). The fluid boundary conditions specify the inflow and outflow of water in the domain, the pore pressure at the boundary of the domain and the pressure gradient over the boundary. At the same time the solid phase is modelled according to the requirements described in the previous paragraph. A hydro-mechanical FEM is a coupled system between the fluid flow and the soil deformation. At first, the uncoupled fluid dynamic FEM is described. Afterwards, this hydraulic formulation is combined with the mechanical FEM to form the hydro-mechanical FEM.

Fluid dynamics FEM

A similar discretization as seen in equation (3.1) can be made for the transient state in fluid dynamics, namely (Section 3, Smith et al. [31]):

$$[\mathbf{k_c}]\{\mathbf{p}\} + [\mathbf{m_m}]\frac{d\{\mathbf{p}\}}{dt} = \{\mathbf{q(t)}\} \tag{3.12}$$

where $[\mathbf{k_c}]$ is the conductivity matrix, $[\mathbf{m_m}]$ is the mass matrix, $\{\mathbf{p}\}$ is the primary variable, which in this case is defined to be the pore pressure and $\{\mathbf{q(t)}\}$ is the inflow and outflow specified at the nodes. The conductivity matrix is given by:

$$[\mathbf{k_c}] = \int\int [\mathbf{T}]^T[\mathbf{K}][\mathbf{T}]dxdy \tag{3.13}$$

where $[\mathbf{T}]$ is a matrix containing the derivatives of the shape functions (similar to $[\mathbf{B}]$ in the mechanical FEM) and $[\mathbf{K}]$ contains the conductivity (permeability) of the material (similar to $[\mathbf{D}]$ in the mechanical FEM). The mass matrix is equivalent to the mass matrix of the mechanical FEM despite of a different density. Equation (3.12) is a simplification of the complete Navier-Stokes equations (Section 2, Smith et al. [31]). The Navier-Stokes equations can also be solved without simplification with a coupled system using both pore pressure and fluid velocity as primary variables. However, in soil mechanics due to the relatively low flow velocities at a macro scale this is often not necessary and the problem can be solved for either fluid velocity or pore pressure.

The mechanical and fluid dynamic FEM discretizations can be observed to be similar, both containing a ’stress-strain’ relationship per element with similar ’stiffness’ and mass matrices. However, a mechanical problem is a second order problem, whereas a fluid dynamics problem is only a first order problem. The fluid dynamics FEM is only a first order problem, due to the fact that the flow can be considered as laminar under the normal fluid velocities present in soils.

Hydro-mechanical FEM

A hydro-mechanical problem can either be solved coupled or uncoupled. In the uncoupled system the pore pressures are often solved first within each time step based on equation (3.12). From the pore pressures distribution the effective stresses are calculated and these effective stresses are used to calculate the displacement. In the next time step the displacement has an influence on the pores and therefore an influence on the fluid dynamics, which is taken into account in the fluid dynamics FEM during the next time step. Even though this seems like a coupled process (both processes influence each other), due to the separation within a timestep equations (3.12) and (3.7) can be solved separately.

A coupled method can also be developed, in which case pore pressure p and displacement u influence each other within each time step according to:

$$\begin{aligned} [\mathbf{k_m}]\{\mathbf{u}\} + [\mathbf{c}]\{\mathbf{p}\} &= \{\mathbf{f}\} \\ [\mathbf{c}]\{\mathbf{v}\} - [\mathbf{k_c}]\{\mathbf{p}\} &= \{\mathbf{q}\} \end{aligned} \tag{3.14}$$

where $[\mathbf{c}]$ is a coupling matrix composed of the shapefunctions and their derivatives (Section 2, Smith et al. [31]). Equation (3.14) uses an quasi-static formulation for both the fluid as the solid. These equations can be expanded to also include dynamic terms. The coupled equation given in equation (3.14) is called a $u-p$ formulation, due to the fact that the solid primary variable is displacement u and the fluid primary variable is pore pressure p. Different primary variables can be used as well, which is addressed in Section 3.1.5.

3.1.3. Implicit or explicit time integration

Equations (3.8) and (3.12) can be solved either implicitly or explicitly with respect to time. To update a state to the next time step in a first-order time dependent problem a weighted average of the gradient at the beginning and the end of the time step is used:

$$\{\mathbf{x}\}_{t+1} = \{\mathbf{x}\}_t + \Delta t((1-\theta)\frac{d\{\mathbf{x}\}}{dt}_t + \theta\frac{d\{\mathbf{x}\}}{dt}_{t+1}) \tag{3.15}$$

where $\{\mathbf{x}\}$ is a time dependent state and θ is a parameter, which can be selected to control the accuracy and stability of the time increment. For higher-order time dependent problems similar integration schemes can be used, for example the Newmark procedure (Section 2, Zienkiewicz et al. [43]):

$$\begin{aligned} \{\mathbf{x}\}_{t+1} &= \{\mathbf{x}\}_t + \Delta t\frac{d\{\mathbf{x}\}}{dt}_t + (\frac{1}{2}-\beta)\Delta t^2\frac{d^2\{\mathbf{x}\}}{dt^2}_t + \beta\Delta t^2\frac{d^2\{\mathbf{x}\}}{dt^2}_{t+1} \\ \frac{d\{\mathbf{x}\}}{dt}_{t+1} &= \frac{d\{\mathbf{x}\}}{dt}_t + (1-\gamma)\Delta t\frac{d^2\{\mathbf{x}\}}{dt^2}_t + \gamma\Delta t\frac{d^2\{\mathbf{x}\}}{dt^2}_{t+1} \end{aligned} \tag{3.16}$$

where θ is replaced by two parameters β andγ, which again control stability and accuracy of the time increment. In a special case of $\beta = 1/4$ and $\gamma = 1/2$ this is equal to an expanded version of equation (3.15) with $\theta = 1/2$, namely:

$$\begin{aligned} \{\mathbf{x}\}_{t+1} &= \{\mathbf{x}\}_t + \Delta t((1-\theta)\frac{d\{\mathbf{x}\}}{dt}_t + \theta\frac{d\{\mathbf{x}\}}{dt}_{t+1}) \\ \frac{d\{\mathbf{x}\}}{dt}_{t+1} &= \frac{d\{\mathbf{x}\}}{dt}_t + \Delta t((1-\theta)\frac{d^2\{\mathbf{x}\}}{dt^2}_t + \theta\frac{d^2\{\mathbf{x}\}}{dt^2}_{t+1}) \end{aligned} \tag{3.17}$$

The dynamic FEM equation (3.7) is expanded in both an explicit and implicit manner in the following paragraphs.

Implicit

Implicit solutions use a $\theta > 0$ and therefore take the gradient at the next time step into consideration. An implicit time integration with $\theta \geq 0.5$ results in a unconditionally stable solution (the error of the integration scheme will be bound) and can therefore use a larger time step compared to an explicit solution. Equation (3.7) can be combined with equation (3.16) with an $\beta = 1/4$ and $\gamma = 1/2$ to form the implicit solution of a dynamic FEM:

$$[\mathbf{K_{NL}}]\{\bar{\mathbf{u}}\} + [\mathbf{M}]\Big(\frac{\{\bar{\mathbf{u}}\}}{\beta\Delta t^2} - \frac{1}{\beta\Delta t}\{\mathbf{v}\}_t - (\frac{1}{2\beta}-1)\{\mathbf{a}\}_t\Big) = \{\mathbf{F^{ext}_{t+\Delta t}}\} - \{\mathbf{F^{int}_t}\} \tag{3.18}$$

Where $\{\mathbf{a}\} = \frac{d^2\{\mathbf{u}\}}{dt^2}$ and $\{\mathbf{v}\} = \frac{d\{\mathbf{u}\}}{dt}$. Equation (3.18) can be rearranged to find the solution given by Wang (Section 9, Bathe [4]):

$$\Big([\mathbf{K_{NL}}] + \frac{[\mathbf{M}]}{\beta\Delta t^2}\Big)\{\bar{\mathbf{u}}\} = \{\mathbf{F^{ext}_{t+\Delta t}}\} - \{\mathbf{F^{int}_t}\} + [\mathbf{M}]\Big(\frac{1}{\beta\Delta t}\{\mathbf{v}\}_t + (\frac{1}{2\beta}-1)\{\mathbf{a}\}_t\Big) \tag{3.19}$$

Which can be reduced to a stiffness relationship if a lumped stiffness matrix is used:

$$[\mathbf{\bar{K}_{NL}}] = [\mathbf{K_{NL}}] + \frac{[\mathbf{M}]}{\beta\Delta t^2} \tag{3.20}$$

and the kinetic terms are combined with the external forces

$$\{\mathbf{\bar{F}^{ext}_{t+\Delta t}}\} = \{\mathbf{F^{ext}_{t+\Delta t}}\} + [\mathbf{M}]\Big(\frac{1}{\beta\Delta t}\{\mathbf{v}\}_t + (\frac{1}{2\beta}-1)\{\mathbf{a}\}_t\Big) \tag{3.21}$$

The total stiffness relationship then becomes:

$$[\mathbf{\bar{K}_{NL}}]\{\bar{\mathbf{u}}\} = \{\mathbf{\bar{F}^{ext}_{t+\Delta t}}\} - \{\mathbf{F^{int}_t}\} \tag{3.22}$$

This results in the requirement to compute the inverse of the stiffness matrix to solve the problem, which can be time consuming and more difficult to implement in non-linear problems. Explicit solutions are therefore still used even though the time step needs to be much smaller for a stable solution.

Explicit

An explicit integration scheme uses a $\theta = 0$ and therefore does not take the variables of the next time step into account when solving that time step. The next time step is only influenced by variables from this time step. Explicit solution often solve for different primary variables as well. The main advantage of using an explicit solution is the lost requirement of computing the inverse of the stiffness matrix (which is a non-diagonal matrix). However, equations (3.8) or (3.12) still need to be solved to find the solution to a problem. Luckily, if equations (3.8) or (3.12) are solved for acceleration or gradient in pressure respectively, only the mass matrix needs to be inverted. The resulting explicit solution is given by:

$$\{\mathbf{a}\}_{t+1} = [\mathbf{M}]^{-1}[\{\mathbf{F(t)}\} - [\mathbf{K_{NL}}]\{\mathbf{u}\}_t - [\mathbf{c}]\{\mathbf{v}\}_t] \tag{3.23}$$

or

$$\{\mathbf{a}\}_{t+1} = [\mathbf{M}]^{-1}[\{\mathbf{F_t^{ext}}\} - \{\mathbf{F_t^{int}}\}] \tag{3.24}$$

Due to the fact that the mass matrix can generally be lumped to a diagonal matrix, with only a small loss in accuracy, its inverse is computed easily and it is therefore advantageous to solve for these primary variables (Section 2, Beuth [5]). Changing the primary variables does change the application of boundary conditions, which will be addressed in Section 3.1.5. In general, the computation of time steps of the explicit solutions are much faster then the time steps of implicit solutions. However, this solution can become unstable. The time step of an explicit solution must therefore be much smaller than the time step of an implicit solution. Dependent on the problem, either implicit or explicit solutions are computed faster over the same time span, because explicit solutions use faster but more timesteps, while implicit solutions use slower but less timesteps. The explicit schemes implementation is less complex and is sometimes implemented in cases where the implicit solution would be faster.

3.1.4. Loads and boundary conditions in FEM

Application of loads and boundary conditions to a material boundary is the main subject of this thesis. In FEM these loads and boundary conditions can be applied directly on the boundary nodes, however this needs to be done differently for each type of condition. In this thesis a surface traction is applied and a fixed primary value condition, i.e. a Dirichlet boundary, and a fixed gradient condition, i.e. a Neumann boundary condition, are taken into consideration (Pathmanathan [24], Section 7, Felippa [17] and Section 2 Zienkiewicz et al. [42]). The fixed primary value condition can either be a fixed displacement,velocity or acceleration in case of a mechanical FEM or a fixed head or flow-rate in case of a hydro-mechanical FEM. These different conditions will be explained below for both the implicit and explicit time schemes.

Surface traction

An example of a two-dimensional surface traction t on a boundary Γ_t can be seen in Figure 3.2. The surface traction can be divided into a normal stress and a shear stress (Section 14, Felippa [17]). In case of a non friction condition on the boundary the shear stress will be ignored and the effect of the traction will be equal to the surface traction decomposed into the normal direction to the surface. Otherwise both normal and shear stresses can be expended on the boundary. The scope of this thesis will only be on problems with a non friction condition on the boundary, but frictional conditions can be tried in the future. The external force applied to the elements will then be given by (Section 2, Zienkiewicz et al. [42]):

$$\{\mathbf{f}\} = \int_{\Gamma_t} [\mathbf{N}]^T \{\mathbf{t_n}\} \tag{3.25}$$

where $\{\mathbf{t_n}\}$ is the traction in normal direction $\{\mathbf{t_n}\} = t \odot n$, with n as the normal to the boundary and the symbol $\odot$ means an element by element multiplication. Due to the fact that the boundary will always coincide with the boundary of an element, this external force can be added to the existing body forces on the right hand sight of equation (3.8) at the boundary nodes. This will be the same for both an implicit or explicit solution. An external flow condition on a hydro-mechanical FEM is treated the same. Only the traction is replaced with the external flow through the boundary.

Dirichlet or fixed freedom boundary condition

A Dirichlet boundary condition, also called a fixed freedom boundary condition, is a condition in which the primary variable is known (Pathmanathan [24]). This could for example mean a constant displacement or a fixed fluid pressure at the boundary. The most simple version specifies the dependent variable

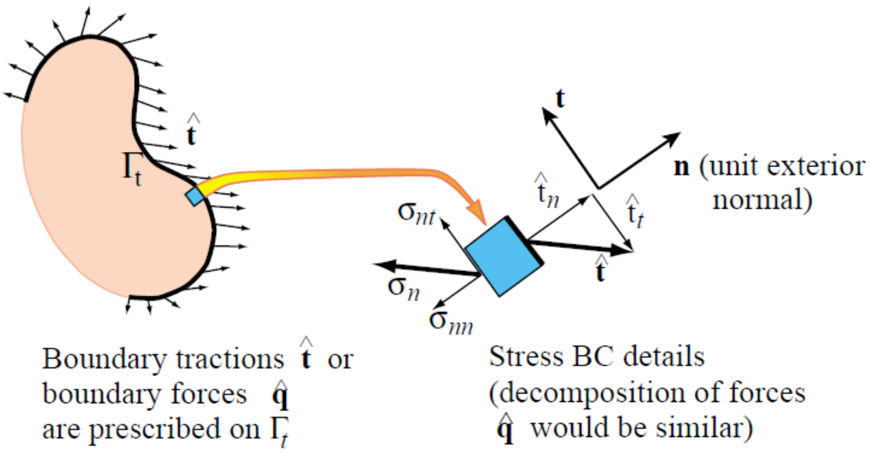


Figure 3.2: Surface traction on boundary Γ_t (Section 14,Felippa [17])

x to be zero on the boundary, which in case of a displacement variable is a completely fixed boundary. In this case the stress-strain equation of this node can be ignored and removed from equation (3.8).

However this is not possible if the variable is a constant, $x = c$, on the boundary. There are two possibilities which are often used in that case. A similar elimination process as with the $x = 0$ boundary can be performed, however in this case the equation needs to be altered instead of removed (Section 2, Zienkiewicz et al. [43]). For example, in case of a static equilibrium problem a variable at the boundary at the eighth node is fixed to be 0.40 m. The stiffness matrix will contain zeros on row eight, with the exception of $K_{8,8} = 1$. The force applied in this row is fixed to $f_8 = 0.40$ kN. This would result in a fixed variable at node eight with a value of 0.40. Performing this 'elimination' process can be tedious. Instead a more practical method is used, in which the diagonal of the stiffness matrix, $K_{8,8}$ in our example, is increased with a 'large' penalty term (Section 3, Smith et al. [31] and Section 2, Zienkiewicz et al. [43]). This penalty must be large enough to diminish the other terms in the equation to be insignificant. A value of 10^{20} is used by Smith. With this technique only the diagonal of the stiffness matrix must be modified to apply boundary conditions, whereas normally the complete equation must be set to zero with the exception of the diagonal. The force should be equal to $f_8 = 0.40(K_{8,8} + 10^{20})$ kN when the penalty term is used. This results in a variable which is roughly equal to the specified value of 0.40 m.

In case of an explicit FEM, the solution in the next time step does not influence the current time step. Therefore, Dirichlet boundary conditions can be enforced by changing the resulting variable after each time step to the required constant at the time step. However, if derivatives of the variable have been calculated and are required within the next time step (for example the velocity and acceleration in a propagation problem), these derivatives might need to be recalculated with the specified boundary value. This can be done easily due to the fact that the velocities, displacements and accelerations of the previous time step are known:

$$\frac{d\{\mathbf{u}\}}{dt}_{t+1} = \frac{\{\mathbf{u}\}_{t+1} - \{\mathbf{u}\}_t}{\Delta t}, \frac{d^2\{\mathbf{u}\}}{dt^2} = \frac{\frac{d\{\mathbf{u}\}}{dt}_{t+1} - \frac{d\{\mathbf{u}\}}{dt}_t}{\Delta t} \tag{3.26}$$

A boundary condition which fixes the derivative of a primary variable on the boundary, for example a fixed velocity, can be rewritten with equation (3.17) to a fixed primary variable. The same can be done in case the primary variable is chosen to be the velocity or acceleration.

Neumann or fixed gradient boundary condition

Applying a fixed gradient boundary condition normal to the boundary introduces a new boundary integral which needs to be applied to the right hand side of equation (3.12) (Section 3, Smith et al. [31]). A fixed gradient normal to the boundary is defined as $\frac{\partial\{\mathbf{x}\}}{\partial n} = C$, with n being the normal to the boundary and $\{\mathbf{x}\}$ is a primary variable of FEM. According to Smith this integral is equal to:

$$\int_\Gamma c_n[\mathbf{N}]^T C\hat{n} d\Gamma \tag{3.27}$$

where c_n is the diffusion property along the normal to the boundary n and $\hat{n}$ is the unit normal to boundary Γ. In case the most common fixed gradient condition $\frac{\partial\{\mathbf{x}\}}{\partial n} = 0$ is applied, this integral can be ignored as it is a natural boundary condition of FEM. This boundary condition therefore is the default boundary condition at any material boundary and does not have to be defined manually.

The integral in equation (3.27) is similar to the integral calculating the force of a surface traction in equation (3.25). Both integrals contain a vector describing the 'force' which has an affect on the domain. This domain affecting 'force' is the normal traction or the specified gradient along the normal in the case of the surface traction or the fixed gradient conditions respectively. Moreover, both integrals distribute the 'force' vector over the nodes using the shape function matrix. In case of the fixed gradient, this integral is multiplied with the diffusion property, which is the main difference between the two integrals. This diffusion property is dependent on the direction of the boundary and is therefore not constant if $c_x \neq c_y$. However, in many cases the diffusion property normal to a boundary can be assumed to be constant. If these assumptions can be made this condition can be applied using the same technique as the surface traction.

3.1.5. FEM formulations using different primary variables

FEM discretizations using many different primary variables are used in practice and the same holds for the MPM model. This section compares these different primary variables models, with a focus on the boundary conditions. A summary of the different possible boundary conditions and the application of these conditions are given at the end of this chapter. This summary can be used to translate the boundary application methods within this thesis to use different time integration schemes and primary variables. Within the section many formulations using different primary variables are described for a quasi-static, dynamic and hydro-mechanical FEM. For each primary variable a short description is given on the application of boundary conditions. The formulations are also separated on time integration scheme (either implicit or explicit). A summary of the application method for the different boundary conditions can be found in Tables 3.1 and 3.2.

Implicit Quasi-static FEM

The standard quasi-static solution is already given in equation (3.6), which uses displacement u as the primary variable. As mentioned before fixed displacement boundary conditions (Dirichlet) can be applied by using for example the penalty method. Force boundary conditions can be applied using the method described in Section 3.1.4 and fixed gradients are not used in the quasi-static solution. Applying fixed velocities or accelerations on the boundary requires a transformation from velocities or accelerations to displacements according to equation (3.16).

Instead of using only the displacement u as an primary variable, a mixed formulation $u-\sigma$ can be used as well, which can increase the accuracy of the stresses significantly (Section 10, Zienkiewicz et al. [42]). This results in the coupled finite element equation:

$$\begin{bmatrix} [\mathbf{A}] & [\mathbf{C}] \\ [\mathbf{C}]^T & 0 \end{bmatrix} \begin{Bmatrix} \{\boldsymbol{\sigma}\} \\ \{\mathbf{u}\} \end{Bmatrix} = \begin{Bmatrix} \{\mathbf{f}_\sigma\} \\ \{\mathbf{f}_\mathbf{u}\} \end{Bmatrix} \tag{3.28}$$

where $[\mathbf{A}]$ is a matrix similar to $[\mathbf{K}]$. $[\mathbf{A}]$ differs from $[\mathbf{K}]$, as it uses the inverse of $[\mathbf{D}]$ to be constructed. $[\mathbf{C}]$ is a couplings matrix, $\{\mathbf{f}_\sigma\} = 0$ and $\{\mathbf{f}_\mathbf{u}\}$ is the normal force vector from equation (3.6). In this model $\{\boldsymbol{\sigma}\}$ is called the primary variable, whereas $\{\mathbf{u}\}$ is called the constraint variable. The constraint variable is necessary to apply the boundary conditions to the primary variable. The boundary conditions can therefore be applied in the same way, using both the penalty method for displacement conditions (Section 3.1.4) as well as the surface traction using the method described in Section 3.1.4.

Implicit Dynamic FEM

As mentioned before the Implicit dynamic FEM is most often described using a displacement u primary variable equation (3.7). The application of the boundary conditions are the same as the quasi-static formulation. However, a formulation using velocity v as the primary variable can also be made (Esch et al. [16]). The force boundary condition can then still be applied the same, however a displacement boundary needs to be transformed to a velocity boundary condition before it can be applied, according to equation (3.16). The coupled solution $(u-\sigma)$ using both stress and displacement from the previous paragraph can probably be expanded to a dynamic problem as well. However, this has not been done in any of the reviewed literature.

Explicit Dynamic FEM

For a dynamical FEM solution explicit time integration schemes are often used instead of the implicit time integration schemes. As mentioned before in Section 3.1.3 the main difference is the use of different primary variables as well as a different method to apply fixed boundary conditions. For an explicit FEM acceleration is used as a primary variable instead of velocity or displacement. In case acceleration is used as a primary variable, fixed acceleration conditions can be applied by replacing the variables at the fixed nodes. Displacement and velocity conditions need a transformation to acceleration according to equation (3.26). Loads applied to the explicit frame are treated the same as in the implicit frame.

Implicit Hydro-mechanical FEM

As mentioned in Section 3.1.2 one of the common ways to describe a two phase material is with the usage of a $u-p$ formulation. In that case the solid boundary conditions are specified in the same manner as in the mechanical case with a u formulation and they are only applied to the equations regarding the solid structure. A fixed pore pressure can be applied using the penalty method on the equations regarding the fluid mechanics and both a Neumann and external flow condition can be specified by applying an additional integral to the fluid force vector f_p. A $u-p$ formulation is a simplification of the $u-w$ formulation, in which w is the displacement of the pore fluid (Zienkiewicz et al. [41]). The $u-w$ formulation is convenient for the finite element formulation due to the usage of displacement for both phases. However, this formulation does require more equations than the $u-p$ formulation and is therefore more computationally intensive. The $u-w$ formulation is less restrictive and is therefore more applicable to large strain problems (Wang et al. [37]). A fixed fluid displacement $w_\Gamma = c$ is again applied using the penalty method and the derivatives of the fluid displacements can be transformed to a fixed displacement. The boundary conditions are therefore similar to the mechanical boundary conditions.

Explicit Hydro-mechanical FEM

Both the $u-p$ and the $u-w$ formulation can be rewritten to an Explicit formulation, namely $a_{solid}-p$ and $a_{solid}-a_{fluid}$ respectively. However, in this research the main focus is MPM formulations, for which $a_{solid}-a_{fluid}$ is the better solution (Wang et al. [37]). Similar to the explicit dynamical FEM fixed boundary conditions can be applied by changing the nodal values to the correct value and the displacement and velocity conditions again need a transformation according to equation (3.26).

3.1.6. Conclusion

FEM is a type of discretization often used in soil modelling. FEM uses a finite element mesh on which the governing equation of the deformation of the soil is solved. In soil-mechanics both pure mechanical as well as hydro-mechanical FEM solutions are used to model the deformation of dry and wet soils. FEM works well under small strain conditions, but many problems arise once the deformation is large. Due to the large deformations the mesh becomes distorted and re-meshing is required to increase the performance of the model. This re-meshing has been proven to be difficult and the MPM method has therefore been developed, which is discussed in the next section.

Applying boundary conditions to an FEM model is straight forward as the condition can be applied directly to the boundary of the mesh. The application techniques described within this chapter can serve as a basis of the boundary application techniques in MPM, which is addressed in the following chapters.

Table 3.1: Summary of boundary conditions for different primary variables in implicit solutions

Primary variable	Dirichlet	Neumann	Load
Quasi-Static (Mechanical)			
u	u: Penalty method v: Transformation to u a: Transformation to u	-	Add to f
$u-\sigma$	u: Penalty method v: Transformation to u a: Transformation to u σ: Penalty method	-	Add to f_u
Dynamic (Mechanical)			
u	u: Penalty method v: Transformation to u a: Transformation to u	-	Add to f
$u-\sigma$	u: Penalty method v: Transformation to u a: Transformation to u σ: Penalty method	-	Add to f_u
v	u: Transformation to v v: Penalty method a: Transformation to v	-	Add to f
Hydro-Mechanical			
$u-p$	u: Penalty method v_{solid}: Transformation to u a_{solid}: Transformation to u p: Penalty method	$\frac{dp}{dn}$: Add to f_p	Traction: Add to f_u External flow: Add to f_p
$u-w$	u: Penalty method v_{solid}: Transformation to u a_{solid}: Transformation to u w: Penalty method v_{fluid}: Transformation to w a_{fluid}: Transformation to w	$\frac{dw}{dn}$: Add to f_w	Traction: Add to f_u External flow: Add to f_w

Table 3.2: Summary of boundary conditions for different primary variables in explicit solutions

Primary variable	Dirichlet	Neumann	Load
Quasi-Static (Mechanical)			
-	-	-	-
Dynamic (Mechanical)			
a	u: Transformation to a v: Transformation to a a: Change nodal values	-	Add to f
Hydro-Mechanical			
$a_{solid}-a_{fluid}$	u: Transformation to a_{solid} v_{solid}: Transformation to a_{solid} a_{solid}: Change nodal values w: Transformation to a_{fluid} v_{fluid}: Transformation to a_{fluid} a_{fluid}: Change nodal values	$\frac{dw}{dn}$: Add to f_w	Traction: Add to f_u External flow: Add to f_w

3.2. Selective review of MPM

In this section the main differences between FEM and MPM are discussed with once again a focus on the application of boundary conditions. The information acquired from the FEM backgroud is only used as basis of the MPM formulation and in the continuation of the thesis the focus has been placed only on MPM models. MPM is an attempt to use the advantages of both Lagrangian and Eulerian methods by combining the two approaches (Sulsky et al. [34], Sulsky and Schreyer [33]). As mentioned in the previous section large strain conditions are difficult to represent with an FEM model, which is the main reason for the development of MPM.

As can be seen in Figure 3.3 the stress-strain relationship is again discretized on a nodal grid. However, in contrast to the FEM grid of Figure 3.1 this grid will not displace and will not represent the shape of the soil body. The granular body is instead discretized in a number of material points (defined in a Lagrangian system), which is connected to a Eulerian background mesh. These material points describe the soil in close proximity to the points. The displacement of an MPM solution is still calculated on a background grid, similar to FEM, but the displacement is applied to the material points (Phuong et al. [25]). After this application the background grid is reset to the initial condition and the background grid will therefore stay as a regular grid, while the material points move through this grid. This solves the re-meshing problem occurring under large deformations in FEM. The MPM solution is therefore a useful tool in modelling large deformation soil mechanics (Phuong et al. [25]).

In this section a description of the MPM algorithm is given, followed by an analysis of the application of boundary conditions to MPM.

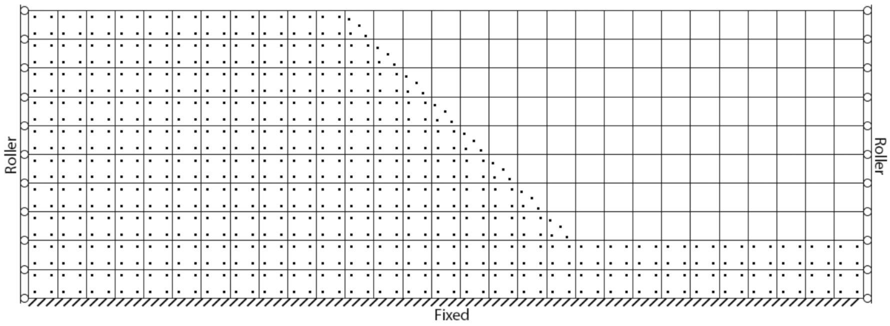


Figure 3.3: MPM Discretization of a soil slope containing a background grid and material points

3.2.1. Governing equation and calculation steps

The governing equations of MPM are often derived from the principal of virtual work (Section 3, Beuth [5]). However, the resulting governing equation is identical to the governing equation of FEM, which is the reasoning behind the long analysis of FEM in the previous section.

The main difference between FEM and MPM is that the state variables, such as stress, are calculated and stored for the material points, in case of MPM, instead of the Gaussian integration points, in case of FEM (Section 3, Beuth [5]). Currently the material points are initialized at the Gaussian integration point, after which the loading step has been separated in three stages (Phuong et al. [25]):

- Mapping of the states of the material points to the background grid (initial step).
- Solve the governing equation, for example equation (3.8), on the background grid (Lagrangian step). This step is equal to the FEM deformation.
- Finally, map the states back to the material points and reset the grid (Convective or Eulerian step).

These steps are visualised in Figure 3.4.

3.2.2. Loads and boundary conditions in MPM

Applying loads and boundary conditions in MPM can be difficult, due to the fact that the boundary of the soil body may not coincide with a boundary of an element. In this thesis boundaries coinciding

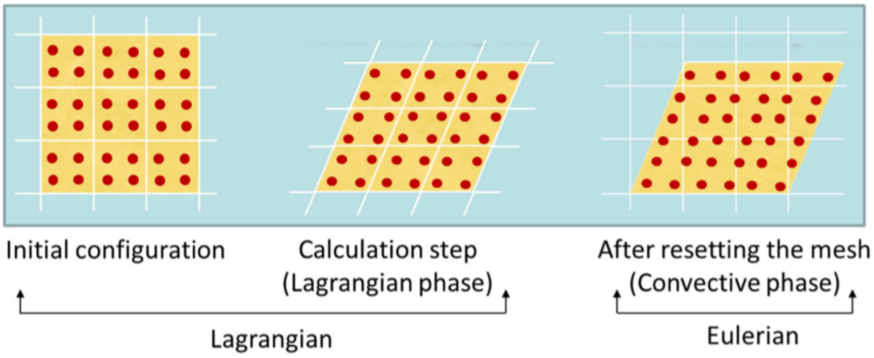


Figure 3.4: Steps of the MPM calculation process (Phuong et al. [25])

with background grid elements are called regular boundaries, whereas irregular boundaries describe boundaries which do not coincide with the background grid. Transferring the boundary condition from an irregular boundary to the nodes of an element is not trivial and will be addressed in Section 3.3. However, applying conditions to regular boundaries is comparable to boundary conditions in FEM and is therefore already addressed in this section.

Surface traction

If the boundary does coincide with a boundary of a background element, the application of a load or surface traction is similar to FEM. In both an explicit or implicit MPM model applying loads (or surface traction) to an element boundary is identical to applying loads to an FEM. The external force can just be added to the boundary nodes of the element during the Lagrangian step according to equation (3.25).

Fixed freedom boundary condition

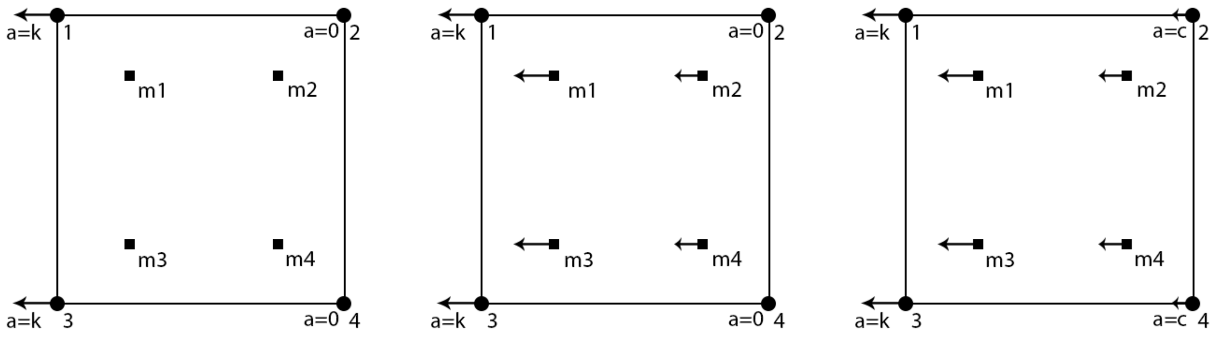
In case of an implicit MPM a fixed freedom boundary condition is also applied in the Lagrangian step, which can probably be done the easiest using the ’penalty-term’ method. This is the same method used in FEM. On the other hand slightly more attention needs to be paid to the explicit MPM. Each time the states of the nodes are mapped back to the material points it is important that all the states correspond to the specified boundary condition. To explain this further the following example is provided: Consider the 4-node element with four material points seen in Figure 3.5. A fixed displacement boundary has been specified on the right boundary, so $u = 0$. From a previous iteration the acceleration on the left side of the element is found to be k, so $a_1 = a_3 = k$ and $a_2 = a_4 = 0$. Due to mapping, all four material points receive an acceleration ($a_{m1} = a_{m3} > a_{m2} = a_{m4} > 0$). During the calculation on the nodes it is then found that $a_2 = a_4 \neq 0$, due to the mapping from the material points with acceleration. This is not a problem, as long as $a_2 = a_4$ are set back to 0 again before mapping to the material points in the new iteration. If this is not done the acceleration of the material points will be to high. This can still be the case even if the boundary condition $u_2 = u_4 = 0$ is forced on the nodes. It is therefore necessary to not only fix the displacement, but also the velocity and acceleration. In case of a non-zero displacement or velocity boundary condition, the fixed velocity and/or the fixed acceleration can be calculated from equation (3.17)

Fixed gradient boundary condition

As mentioned in Section 3.1.4 applying fixed gradient boundary condition is similar to applying a surface traction as long as the diffusion normal to the boundary is constant. Similar to the surface traction, it is therefore logical to apply this boundary condition as a force to the Lagrangian step.

3.2.3. Conclusion

MPM is a relatively new concept in comparison to FEM. However, due to the fact that many algorithms of FEM can be reused a lot of progress has been made on MPM. The governing equation is the same as a governing equation of FEM. However, the variables required for this equation are not stored on the grid, which is the case for an FEM, but on the material points. The points flow through the background grid and their variables, for example the stress and strain, are mapped back to this background grid. The equations of motion are solved on the grid and applied back onto the material points.



(a) Initial condition (b) Mapping to the material points (c) Mapping back to the nodes

Figure 3.5: Example explaining the problem with boundary conditions due to the mapping of state variables

Due to the fact that the equations of motions are solved on a grid, applying boundary conditions is similar to FEM. However, in MPM a irregular boundaries can be present. These boundary conditions are no longer applied with the same technique used in FEM. The application of boundary conditions on these boundaries is therefore a currently unsolved problem and the focus of this thesis. In the next section previous research on this topic is discussed after which the missing pieces are determined.

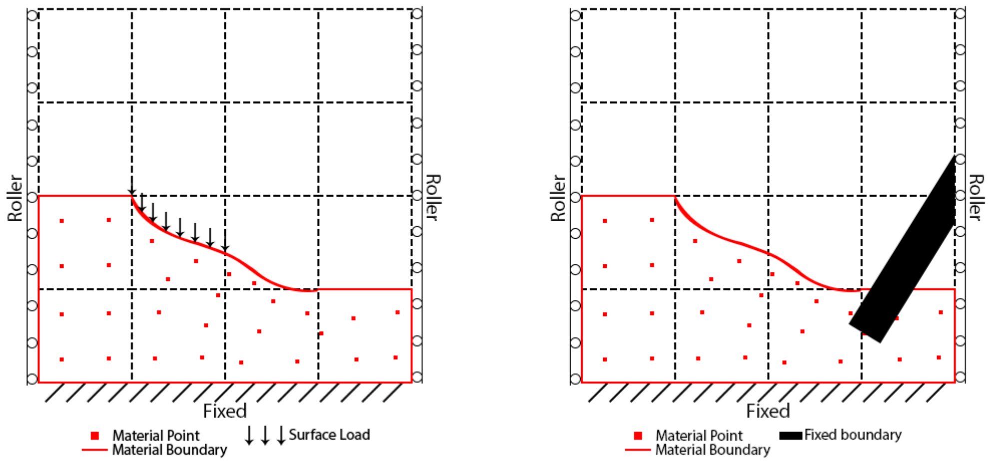
3.3. Loads and boundary conditions on a free boundary in MPM

As mentioned in Section 3.2 applying a load or boundary condition on a free boundary is difficult in MPM due to the fact that irregular boundaries can occur. These are called irregular boundary conditions in this thesis. In case of fixed boundaries the grid can often be aligned to the boundaries, but this might require a non-regular background grid. The MPM method tries to avoid using non-regular meshes and it might therefore be preferred to use a regular grid with a irregular boundary condition. This thesis tries to solve the two problems visualized in Figure 3.6. In Figure 3.6a a surface load is applied to a failing soil slope. In reality this could for example represent a shallow founded structure on top of the soil slope, which applies a gravitational load during the slope failure. Due to the moving surface this boundary condition will be irregular during the deformation. In a hydro-mechanical MPM Figure 3.6a could also represent a pore pressure or flow rate applied to a moving surface. A slanted boundary condition can be seen in Figure 3.6b. In reality this could for example represent a stiff pile foundation or sheet pile, which blockades the failure of the slope. Due to the small size and the angle of this pile in comparison to the background grid this boundary condition is irregular. In this section a inside is provided into previous research on this topic.

3.3.1. Previous research on irregular boundaries

In hydrodynamics, irregular boundary conditions also occurred in a method called Smoothed Particles H ydrodynamics (SPH). The method used to apply irregular boundary conditions in hydrodynamics can perhaps be reused in MPM. In hydrodynamics the irregular boundary conditions are applied by including the residual boundary terms in the calculation or by using virtual or ghost particles (Ma et al. [21]). Due to the fact that the material points and the boundary are moving, it is difficult to include the residual terms, because this requires some hand calculations based on the location of the boundary. Ghost particles also have been used to apply boundary forces or fixed primary variable conditions (Monaghan [22]). However, this method has been applied on SPH instead of MPM and the ghost particles have only been used on boundaries with known location during the analysis. In case of a moving boundary only reflection conditions have applied using this method (Allahdadi et al. [1], Randlesa and Liberskyb [27], Liu et al. [19]). In this case the interest lies with a moving boundary for which the location is not known beforehand on which loads or displacements are applied.

A moving mesh could allow the material boundary to always coincide with the nodes of the background mesh (Wang et al. [40]). However, in case of a failing soil surface, this will result in an irregular background grid, which probably needs to be re-meshed when the deformation becomes too large. The main reason to use an MPM model will then be lost.



(a) Surface traction applied to a failing soil slope (b) A slanted fixed boundary condition

Figure 3.6: Irregular boundary conditions in MPM

The boundary conditions can also be applied directly to the material points within a certain boundary layer h or to the material points in the boundary elements (Chen et al. [8] and Wang et al. [36]). However, in case of moving boundary, finding the material points within this layer is quite difficult as the location of the boundary needs to be known. Moreover, load application on the material points can result in stress oscillation (Section 3, Beuth [5])

Applying loads and boundary conditions defined on a irregular boundary to the background mesh, might be a more simple solution and has only been tested recently. This solution is comparable to regular boundary conditions as they are also applied to the background grid. Therefore, all the requirements for application of irregular boundaries to the background grid are already included in most MPM programs. However, a transformation is required to move from the irregular boundary to the nodes of an element. These transformations are quite difficult and research has only just started on it (Cortis et al. [10] and Cortis et al. [11]). Background information on these transformations are described in the next couple of paragraphs as in recent research a couple of boundary conditions have been applied on irregular boundaries with this technique. To be able to transform the boundary conditions, it is also necessary to find the location of the material boundary. In contrast to FEM, locations of material boundaries are not described by the method. An algorithm to find the domain boundary is therefore required. Most importantly this algorithm should find the free boundaries, because fixed boundaries are often known from the problem geometry. An edge detection method is required to find the material boundaries. Background literature on these algorithms are discussed in the next section.

3.3.2. Surface traction on irregular boundaries

The application of a surface traction to an irregular boundary is an unsolved problem. Regular boundary conditions are transformed from the boundary to the nodes according to equation (3.29) (Section 2, Zienkiewicz et al. [42]).

$$\{\mathbf{f}\} = \int_{\Gamma_t} [\mathbf{N}]^T \{\mathbf{t}\} \tag{3.29}$$

where $\{\mathbf{f}\}$ is the force applied to the nodes, Γ_t is the part of the boundary on which traction $\{\mathbf{t}\}$ is applied and $[\mathbf{N}]$ is the shape function matrix. This equation will also be used for the transformation from an irregular boundary condition to the surrounding nodes. To apply this transformation equation (3.29) needs to be worked out, which is done in Chapter 6.

3.3.3. Fixed primary variables on irregular boundaries

As mentioned before even though the explicit MPM has been proven to be useful in many cases, problems with fixed variables on irregular boundaries have been avoided in many studies (Cortis et al. [11]). In recent studies, irregular boundaries have been taken into consideration in research on the implicit MPM formulation (Cortis et al. [10] and Cortis et al. [11]). In the first part of this section the Implicit Boundary Method (IBM) is discussed, which applies fixed displacement boundary conditions to irregular boundaries with a known location in implicit MPM models. Cortis et al tested this method on a boundary with a fixed location or a moving boundary with a known location during the computation. This technique can hopefully be expanded to moving irregular boundaries with an unknown location. In the second part the requirements of an fixed primary variable boundary condition for explicit MPM models are described.

Implicit MPM

The method used to apply fixed variable boundary conditons to implicit MPM is IBM developed by Burla and Kumar, who based their method on the 50 year old idea of using trial functions for a displacement due to a boundary condition (Burla and Kumar [6] and Kantorovich and Krylov [18]). Burla and Kumar used trial functions containing Dirichlet functions specifying the boundary conditions:

$$\{\mathbf{u}\} = [\mathbf{D}]\{\mathbf{u^g}\} + \{\mathbf{u^a}\} \tag{3.30}$$

where $\{\mathbf{u}\}$ is the displacement, $\{\mathbf{u_g}\}$ is the grid variable, defined using standard MPM approaches, $\{\mathbf{u_a}\}$ is the boundary value function and $[\mathbf{D}]$ are the Dirichlet functions which vanish at the boundaries. u_a is a vector field which takes on the correct values at the boundary, so that $u = u_a$ at the boundary. This can be homogeneous for example if the displacement at the boundary is constant over the boundary. The Dirichlet function matrix $[\mathbf{D}]$ is a diagonal matrix ($[\mathbf{D}] = diag(D_1, ..., D_i, ..., D_{n_d})$) containing functions D_i, which become zero at the boundary at which the displacement in direction i is specified. In our 2D research, $[\mathbf{D}]$ is therefore a diagonal 2×2 matrix. Introduction of the trial function will change the stiffness matrix and introduces penalty terms in the vicinity of the boundary. The stiffness matrix becomes (Burla and Kumar [6] and Cortis et al. [11]):

$$[\mathbf{K_e}] = [\mathbf{K_1}] + ([\mathbf{K_2}] + [\mathbf{K_2}]^T) + [\mathbf{K_3}] \tag{3.31}$$

where $[\mathbf{K_1}]$ is the standard stiffness matrix per element and is computed for every element containing material points. The penalty terms are only calculated on the elements containing an essential irregular boundary condition. This method can also be used on regular boundary conditions, but might be less accurate than the methods originally used for regular boundaries due to the fact that a small boundary layer is still created.

$[\mathbf{K_2}]$ and $[\mathbf{K_3}]$ only need to be evaluated on the boundary layer, due to the fact that they contain the gradients of the Dirichlet functions (Burla and Kumar [6]). These gradients are only non-zero close to the boundary of the domain. According to Burla et al. $[\mathbf{K_2}]$ is then given by:

$$[\mathbf{K_2}] = \int_{V_E} [\mathbf{\bar{B}_1}]^T [\mathbf{\bar{D}_1}]^T [\mathbf{D_e}][\mathbf{\bar{D}_2}][\mathbf{\bar{B}_2}] dV \tag{3.32}$$

where $[\mathbf{\bar{B}_2}]$ and $[\mathbf{\bar{B}_1}]$ contain the shape functions and the derivatives of the shape functions respectively, $[\mathbf{\bar{D}_1}]$ and $[\mathbf{\bar{D}_2}]$ contain the Dirichlet functions and their derivatives respectively and $[\mathbf{D_e}]$ is the standard material constitutive matrix. $[\mathbf{\bar{D}_2}]$ will only contain non-zero entries in a thin band along the boundary due to the fact that the gradients are zero at a small distance from the boundary. The integral in equation (3.32) can therefore be evaluated over a thin band around the boundary (Cortis et al. [11]):

$$[\mathbf{K_2}] = \int_t [\mathbf{\bar{B}_1}]^T \Big(\int_0^\delta [\mathbf{\bar{D}_1}]^T [\mathbf{D_e}][\mathbf{\bar{D}_2}][\mathbf{T}] dn \Big) [\mathbf{\bar{B}_2}] dt \tag{3.33}$$

where $[\mathbf{T}]$ is a rotation matrix, which allows the method to be used on boundaries non-parallel to the background grid (Cortis et al. [11]). The integral is integrated along the boundary in the t-direction and across the boundary in the direction of the normal to the boundary n. δ is a small value (10^{-5} was used by Burla et al.) which describes the thickness of the boundary layer. $[\mathbf{K_3}]$ can be evaluated over the same domain as $[\mathbf{K_2}]$ and is given by:

$$[\mathbf{K_3}] = \int_t [\mathbf{\bar{B}_2}]^T \Big(\int_0^\delta [\mathbf{T}]^T [\mathbf{\bar{D}_2}]^T [\mathbf{D_e}][\mathbf{\bar{D}_2}][\mathbf{T}] dn \Big) [\mathbf{\bar{B}_2}] dt \tag{3.34}$$

A more detailed explanation has been provided by Cortis and is given in Appendix A Similar to the penalty method described in Section 3.1.4, the displacement is applied as a force on the right hand sight of the equation. The applied force due to a prescribed displacement $\{\mathbf{u_a}\}$ on a boundary is given by:

$$\{\mathbf{f_e}\} = ([\mathbf{K_2}] + [\mathbf{K_2}]^T + [\mathbf{K_3}])\{\mathbf{u_a}\} \tag{3.35}$$

Explicit MPM

Unfortunately, a fixed boundary condition on an irregular boundary for an explicit MPM has not been designed in previous research. Similar to the explicit FEM and regular boundary conditions in MPM, the penalty method can not be used in this case. IBM is therefore not applicable to explicit MPM. A new technique must therefore be designed. It might be possible to apply the boundary condition by replacing the calculated primary variable at the nodes by the correct value, based on the boundary conditions. This technique is comparable to the technique used for regular boundary conditions. However, due to the fact that the boundary does not coincide with the nodes, an iteration is required to determine the correct values of the primary variables at the nodes. This iteration should decrease the difference between the defined value at the boundary and the actually applied value at the boundary.

3.3.4. Fixed gradients on irregular boundaries

As described in Section 3.2.2 applying fixed gradients is similar to applying a surface traction in case of an regular boundary with a constant gradient. The same holds for an irregular boundary. The integration along the boundary used in Section 3.3.2 can be used for fixed gradients as well.

3.3.5. Conclusion

As mentioned, the research on irregular boundary conditions has just started. Based on this literature study the choice has been made to design a boundary application technique similar to the application techniques of FEM and regular boundary conditions in MPM. One technique, namely IBM, has already been designed based on these principles. This method applies fixed primary variable conditions to irregular boundaries with a known location. This technique must be expanded to moving boundaries with unknown location as well. Furthermore, a method must be developed for the same boundary condition in explicit MPM models.

3.4. Free boundary detection

As mentioned in Section 3.3 the location of the material boundary needs to be known to be able to apply a boundary condition to an irregular boundary. Previous studies in fluid dynamics have already tried to locate the surface of a material. A couple of methods have been developed for boundary detection in the Smoothed-Particle Hydrodynamics method, which is comparable to MPM. Some of these techniques can possibly be used to detect the boundary in MPM as well. However, the techniques need to be adjusted to fit soil mechanics and the MPM model, in which it tracks the boundary between the soil (possibly a two phase material including solid particles and a pore fluid) and the adjacent substance (probably air, water or a different solid surface).

Morris [23] and Sethian [29] discuss a couple of edge detection techniques from which the Level Set Method, Volume of Fluid technique (VOF) and Surface Marker Method (SMM) appear to be the most applicable to this thesis. A theoretical background behind the methods explained by Sethian is given in the next paragraphs, followed by an explanation of the implemented algorithms in Chapter 4 together with the choices behind these algorithms, the results of the algorithms and a summary of the advantages and disadvantages of the three methods.

3.4.1. Surface Marker Method

In SMM the boundary is discretized into M intervals with usage of $M+1$ markers. A soil slope discretized by markers is shown as an example in Figure 3.7. In the version of SMM described by Sethian [29], the time and space derivatives of the marker points are expanded using Taylor solutions and a discretized equation of motion is solved for each point. The location of the markers is then known in time, based on their initial location and the movement calculated from the equations of motion. Because MPM already uses moving points to describe the material (the material points) it might be possible to replace the marker points with “fake” material points and therefore apply the already calculated motion to the fake points. These fake points do not store any information except for their own location and therefore don't influence the movement of the "normal" material points. The marker points are only used to track the surface they are initialized on. The method is therefore fast and requires a small amount of memory.

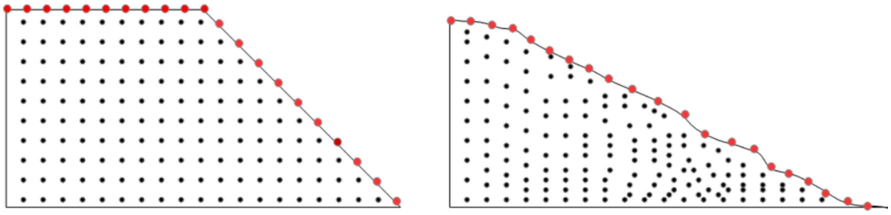


Figure 3.7: SMM – Material points shown in black and markers in red.

3.4.2. Volume of Fluid method

A different approach to boundary detection would look at the volume of material present at a location. The boundary is then present at the edge between non-zero and zero volume of material. This approach is taken in VOF, which determines the amount of material present in each grid cell of a mesh (Sethian [29]). The amount of material per cell can for example be represented by the percentage of the cell which is filled with material. Flow of material can than be described by increasing and decreasing the percentage of filled material. In MPM each cell could possibly receive a value based on the amount of material points in the cell. In the method discussed by Sethian the flow of the material is described by increasing or decreasing the cell value based on the in and outflow of material, but in MPM the movement of the material is already defined by the material points. It is therefore redundant to describe the material based on the in and outflow of material. The cell values are therefore only calculated based on the number of material points within each cell. In Figure 3.8 these values are represented by the strength of the red colour of each cell. The cells can now be interpreted as a grayscale image of the geometry (as seen in Figure 3.8), even though converting to an actual image is not necessary during the algorithm.

Normal image edge detecting could be used to find the edges, for example a Canny Edge Detector (Pound and Riley [26] and Canny [7]). A Sobel operator can be used to find the cells which contain the edge (Pound and Riley [26]). A Sobel operator is a matrix filter, which is applied to a gray scale image to detect edges. Two filters are applied to find the edges in the vertical and horizontal direction Pound and Riley [26]:

$$\begin{aligned} G_x &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * VOC \\ G_y &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * VOC \\ G &= \sqrt{G_x^2 + G_y^2} \end{aligned} \tag{3.36}$$

where VOC is a matrix containing the volume of material in each cell, G_x and G_y are the Sobel operaters in respectively the horizontal and vertical direction, G is the combined Sobel operator giving all edges and $*$ is a convolution operator. G gives a high positive or negative result if the grid cell contains or touches the edge and the operator results in a low or zero value if the cell is not close to the edge. G can therefore be used to find the edges of a figure. G_x and G_y can also be used to find the direction of the edge, which can help to connect different edges. A Canny Edge Detector will only report edges that are connected to other edges to avoid finding only small changes in amount of material and only respond to the correct edges (Pound and Riley [26]). According to Sethian VOF is stable, but has a questionable accuracy (Sethian [29]).

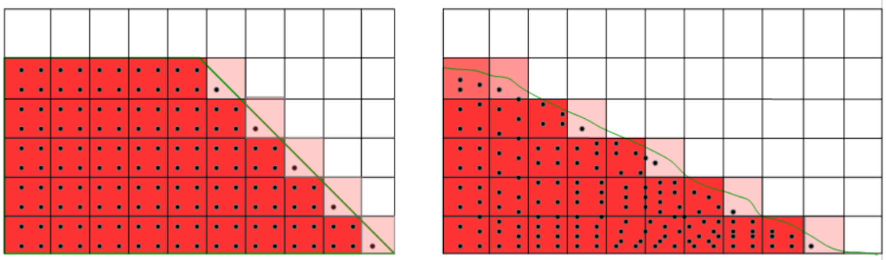


Figure 3.8: Coloured image based on amount of material within each cell (or the number of material points)

3.4.3. Level Set Method

The Level Set Method is a group of methods which uses the same concept, namely defining the boundary as a certain level of a field. Often the zero level is used as the boundary. For the version of the Level Set Method explained by Sethian [29], a boundary is considered to be a curve in two dimensions or a surface in three dimensions. The goal is to track the surface as it moves in the normal direction with a speed function F, as seen in Figure 3.9 (Sethian [28]). Tangential movements are ignored in his case. This speed function is dependent on local geometric information (L), properties depending on the geometric information (G) and independent properties (I). This immediately represents the first difficulty with respect to the MPM model. In the MPM model the speed function is a more complex function than a function only based on geometry and curvature. Moreover, the velocity of the surface is probably not normal to the surface. However, it might be possible to use the velocities calculated from the material points as a speed function.

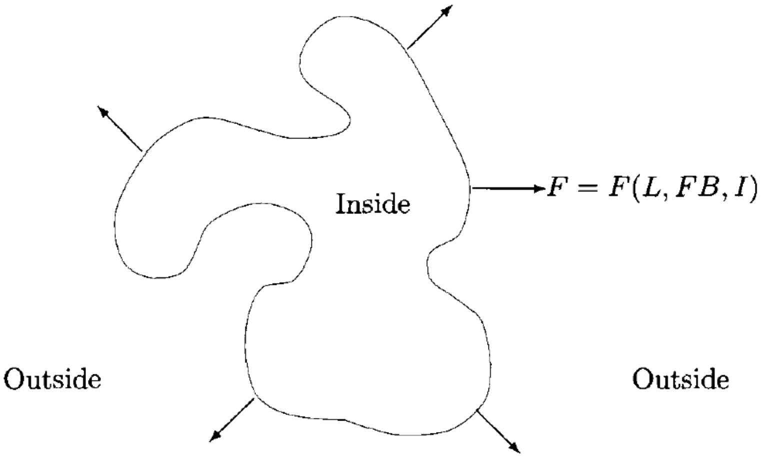


Figure 3.9: Boundary moving in normal direction with a speed function F (Sethian [28])

According to Sethian [28] the solution will be stable, if the initial curve is smooth, non-convex and both the speed function F and the curvature of the curve are twice differentiable. Moreover, under these conditions the change in variation of the curve, as seen in Figure 3.10, can be predicted.

However, due to the large strains the curvature of the boundary will probably be not twice differentiable, at least not during the complete process. For example a sharp corner might occur in the boundary, in which the normal of the surface would be undefined. To solve these problems 'weak solutions' need to be used for these corners as described by Sethian [28]. For this research, the boundary needs to represent a material, which will in fact be smooth (the boundary cannot cross itself). This needs to be taken into consideration when a weak solution is chosen. Luckily Sethian [28] tried to solve similar problems and a similar solution can therefore be used.

As described previously, the boundary $\gamma(t)$ is assumed to move in its normal direction with a speed function F. The velocities of the material points and background grid are known and might be able to act as a known speed function. The idea behind the Level Set Method is to define a function $\phi(x, t)$, where the boundary is represented by the zero level of this function (Sethian [28]). In the initial situation

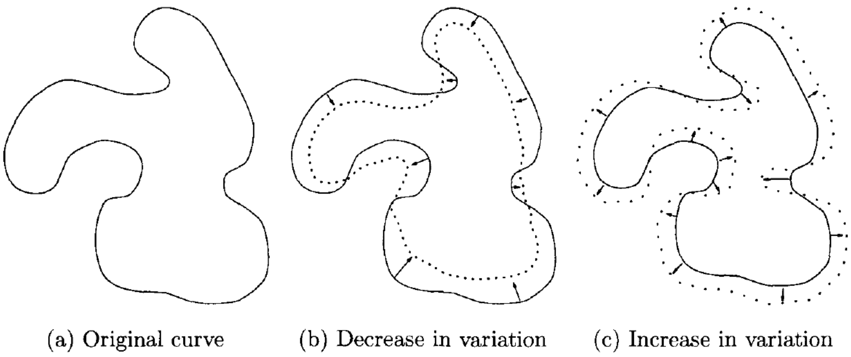


Figure 3.10: Changes in variation of the surface (Sethian [28])

(a) Original curve (b) Decrease in variation (c) Increase in variation

$\phi(x,t) = 0$, is given by:

$$\phi(x, t = 0) = \pm d \tag{3.37}$$

Where d is the distance to the surface. A positive d represents a point outside of the material, where a negative d represents a point inside the material. Sethian [28] then solves the following level aet differential equation on a grid to find the development of each contour of the function in time:

$$\phi_t + F|\nabla\phi| = 0 \tag{3.38}$$

However, due to the fact that F is unknown and not normal to the boundary direction, this differential equation is not easily solvable for an MPM method. Moreover, it seems to be redundant to solve a second differential equation while the position and velocity of the material points and background grid are already found, based on the equations of motion. However, the concept of using a level set function, which is defined as the distance to the boundary, and applying velocity to this function at each time step might be a possibility.

In summary, this version of the Level Set Method seems to be not applicable in combination with the MPM method, due to the fact that most of the advantages of this method are lost once the movement of the boundaries is not only in the normal direction. Moreover the movement function is unknown in the case of MPM. However, the method can be changed to be used in MPM.

3.4.4. Conclusion

SMM and VOF seem applicable for an MPM method without many modifications. Although these methods are uncomplicated, the accuracy and stability of these methods is questionable. The Level Set Method has a higher accuracy and stability then the previous two methods (Sethian [28]), but the version presented above needs to be modified to be applicable to MPM.

3.5. Conclusion

In this literature review the background information behind the application of boundary conditions to moving boundaries in MPM is given. In FEM, the basis of MPM, boundary conditions are applied to the nodes of the FEM mesh. Surface tractions are applied as a force at the nodes in the governing equation of FEM, whereas fixed variable conditions are applied by modifications to the stiffness matrix. In MPM, the same technique is used for the application of boundary conditions to boundaries coinciding with the background grid. However, only recently research has started on the application of boundary conditions to boundaries not coinciding with the background grid.

A couple of techniques to apply boundary conditions to moving boundaries have been discussed in Section 3.3. The application of these conditions to the background grid has been chosen, because of the resemblance to FEM. Moreover, conditions applied on boundaries coinciding with the background grid have already been successfully applied to the background grid. This technique has been chosen over Ghost Particles or application to the material points, due to the inaccuracies present in these methods. These inaccuracies can hopefully be reduced or avoided with the application of conditions to the background grid. Finally the methods has been chosen over a moving mesh, due to the fact that the background mesh is chosen to be fixed and regular to avoid re-meshing issues.

A transformation from the boundary to the background grid is required with this technique. Cortis et al. [10] already designed one of these transformations for an implicit MPM, namely IBM. Different boundary conditions, such as the application of a surface traction, must therefore still be developed. The same formula used in FEM to apply surface tractions can probably be expanded to also be used to apply surface tractions.

Finally, the location of the boundary must be known to compute the transformations. A mathematical description of the boundary is therefore necessary. This mathematical description can than be used as a basis for the development of an edge detection method for MPM. SMM, VOF and Level Set Method, have been reviewed as edge detection methods and seem applicable to MPM with a couple of changes. In this thesis, these methods have been implemented and tested. They are compared on their usability with MPM.

4

Boundary detection algorithms

As mentioned in Chapter 3, applying boundary conditions on irregular boundaries has proven to be difficult in MPM. Within this thesis the application of boundary conditions is proposed to be similar to this task in FEM. The proposed solution first determines the location of a soil boundary, the boundary between the soil and a different material. This location is then used to apply the boundary conditions to the background grid of MPM. The location of the boundary is used to move the boundary condition from the boundary to the background grid. In currently ongoing research, this transformation technique has been tested on fixed displacement boundary conditions at predictable locations (Cortis et al. [10]). In that research the technique has been applied to boundary conditions at fixed locations or moving boundaries which deformation is defined by the user. The solution needs to be expanded to include different displacement conditions. Moreover, the technique can be used on moving boundaries, which location is unknown before the calculation, for example a load on top of a failing slope surface. A start has been made on both of these expansions in this thesis.

The location of the boundary must be found automatically for the application of the boundary conditions to the nodes of the MPM background grid. In FEM the boundary is always represented by an edge of the material mesh. However, as mentioned in Section 3.2, the background mesh is fixed in MPM and the material points moving through the mesh represent the soil. Therefore, the edge of the material mesh cannot be used to represent the boundary in MPM. Moreover, the material points cannot be used directly to represent the boundary, due to the fact that they are not located at the boundary. Therefore, a different method is required. In this section two previously used edge detections method have been tested, namely the SMM and VOF. These methods are compared to a third method, which has been designed specifically for this thesis, namely the PFM. At the start of this chapter, one section is dedicated to general descriptions and definitions of soil boundaries. Composite Bézier curves, the type of curve used to describe the boundary mathematically, are explained within this section as well. Finally, an analysis and a comparison of the three boundary detection methods is given in the last section of this chapter.

4.1. General information on boundaries

A boundary is the edge between two different materials. In this thesis, only boundaries on the macro scale are taken into account. The boundary between soil and air or soil and water at ground level or the boundary between a stiff and soft soil are examples of boundaries at the macro scale. The boundary between soil particles and pore fluid is not taken into account as its scale is considered to be too small. However, extremely large pores, with sizes many times larger than the soil particles, could be seen as boundaries on the macro scale and these gaps are taken into account.

The boundary between soil and a different material has been defined as a set of Bézier curves. The order of the composite curve defines the order of the polynomials used in the curve as well as the continuity of the composite curve. These curves have been chosen for their flexibility regarding the application of boundary conditions. Linear, quadratic and cubic curves can be used as a boundary, without rewriting the code of the application of boundary conditions, due to the fact that one description is used for all these curves. Moreover, the edge detection methods only need to define two points per polynomial, namely the required points for a composite curve using linear parts. The points required for the higher order curves can be determined based on the location of these two points.

The section continues with a further explanation of composite Bézier curves as well as the method to construct the additional points required for the higher order polynomials from the two given points.

4.1.1. Bézier curves

The boundary Γ can be described by a composite Bézier curve. A composite Bézier curve is a continuous curve formed from small polynomial sections (Shene [30]). Each section is defined as a Bézier curve according to the following equation:

$$\Gamma_j = B(t)_j = \sum_{i=0}^{n} b_{i,n}(t) D_i, t \in [0,1] \tag{4.1}$$

where Γ_j is a section of the boundary defined by Bézier curve $B(t)_j$, which is defined by the Bernsteinpolynomals $b_{i,n}$ and the control points D_i. The Bernsteinpolynomals $b_{i,n}$ are defined to be:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, i = 0, ..., n \tag{4.2}$$

The endpoints (D_0 and D_n) of each Bézier curve are connected to each other and a composite Bézier curve is therefore at least C_0 continuous (Cheng [9]). These end points will be found by the edge detection methods described later in this chapter, but the intermediate points (D_1 ... D_{n-1}) are computed based on the continuity of the composite curve. Linear curves ($n = 1$) allow C_0 continuity (endpoints coincide) and intermediate points are not necessary. Quadratic curves ($n = 2$) allow C_1 continuity (tangent vectors at endpoints are the same) and require 1 intermediate point. Finally Cubic curves ($n = 3$) allow C_2 continuity (curvature at endpoints is continuous), which requires 2 intermediate points. Endpoints D_0 and D_n are always on the Bézier curve, while the additional points (D_1 ... D_{n-1}) guide the curve, but are not located on the curve. These points only guide the curve to follow the rules of continuity.

In Figure 4.1, a boundary approximated by a composite Bézier curve of order 1, 2 and 3 is shown. The exact location of the boundary lies somewhere in between the linear and cubic Bézier curve, because the corners of the linear curve are too sharp and the swelling added at the corners of the cubic curve is unreasonable. However, both approximate the shape with decent accuracy, because only the corners are approximated incorrectly in both cases. A cubic Bézier curve gives a smoother surface, with small deviations at the corners, due to the fact that the method is not ideal for the representation of sharp corners. The corners only effect the first segment before and after the swelling occurs. The influence of the corner is therefore small. The influence could be completely removed by removing the C_1 and C_2 continuity at sharp corners. This has not been done during this thesis, but could be investigated in the future. A linear Bézier curve does not result in any swelling at the corners. However, the linear curves are only C_0 continuous.

Due to the low number of freedoms of the quadratic Bézier curve and the enforcement of C_1 continuity, sharp corners are devastating for a quadratic Bézier curve. The deviation inflicted by a corner is larger than the deviation for a cubic Bézier curve. Moreover, this swelling influences all the segments before and after the corner. This approximation is unusable due to the low accuracy. Small local deviations from the exact shape often affect the complete shape. In Figure 4.1 large oscillations of the quadratic curve can be seen around a part of the boundary, which is supposed to be a straight line. In this case the large oscillations occur due to almost straight angles at the corners. However, smaller oscillations will still occur at corners with shallower angles. Quadratic Bézier curves are therefore unusable and only linear and qubic curves will be used in the continuation of this thesis. Higher order curves might be usable as well and could present a smoother surface, but this investigation is left for future research due to the time constraints of this thesis.

4.1.2. Construction of additional control points from the endpoints

To construct the sections of the composite Bézier curves the endpoints of each part need to be found. The control points in between the endpoints can be computed from these endpoints. This construction is not necessary for linear curves and the focus in the continuation of this paragraph will therefore be

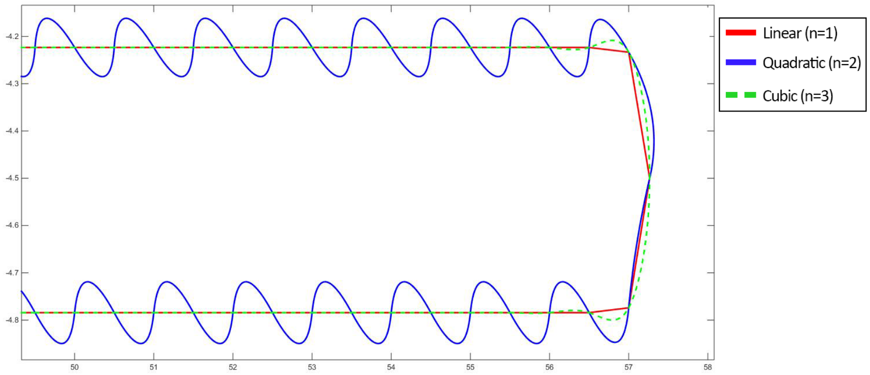


Figure 4.1: Composite Bézier curves of order n = 1,2 and 3 (Linear, Quadratic and Cubic).

on cubic curves. All surfaces will be assumed to be C_2 continuous surfaces, with a starting and end point (P_0 and P_m respectively). P_0 and P_m are the same point if the surface is a closed shape. A closed and open cubic composite Bézier curve consisting of 5 individual pieces can be seen in Figure 4.2. The control points of this curve consist of the endpoints P_0 to P_5 and the middle points Q_1 to $Q_1 0$. In general a n order composite Bézier curve of m pieces needs the control points P_0...P_m and Q_1...$Q_{m(n-1)}$. If the order is $n = 1$ (linear curves) no middle points are needed. If a cubic composite curve is assumed to be C_2 continuous, the locations of the middle points Q_1...Q_{2m} can be found by solving the following system of equations (Eno [15]):

$$\begin{aligned} Q_i + Q_{i+1} &= 2P_{i/2} \text{ for } i = 2, 4, 6 \ldots (2m-2) \\ Q_i - 2Q_{i+1} + 2Q_{i+2} - Q_{i+3} &= 0 \text{ for } i = 1, 3, 5 \ldots (2m-3) \end{aligned} \tag{4.3}$$

For a closed surface the following equations need to be added:

$$\begin{aligned} Q_1 + Q_{2m} &= 2P_0 \\ 2Q_1 - Q_2 + 2Q_{2m} - Q_{2m-1} &= 0 \end{aligned} \tag{4.4}$$

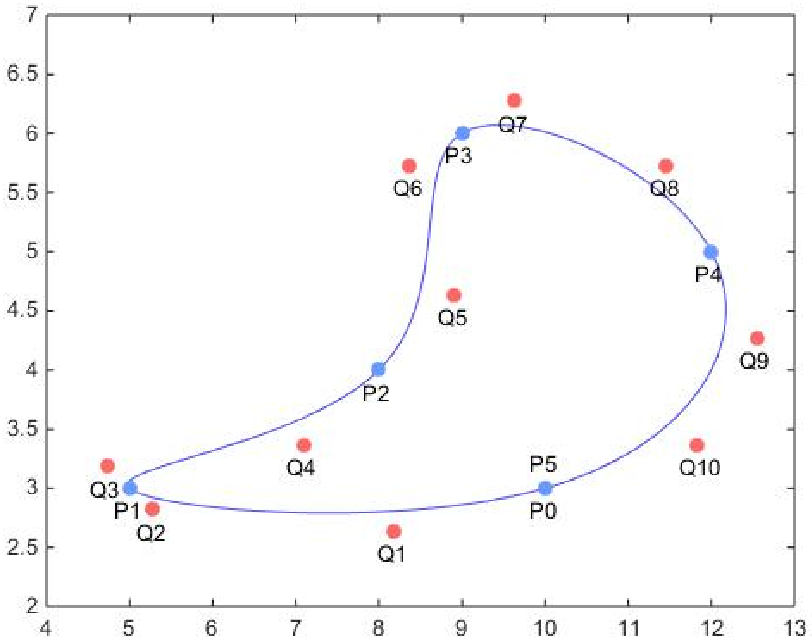
For an open surface an assumption needs to be made for Q_0 and Q_m. They have been chosen to be governed by the following equations:

$$\begin{aligned} Q_1 &= \frac{3}{4}P_0 + \frac{1}{4}P_1 \\ Q_{2m} &= \frac{3}{4}P_m + \frac{1}{4}P_{m-1} \end{aligned} \tag{4.5}$$

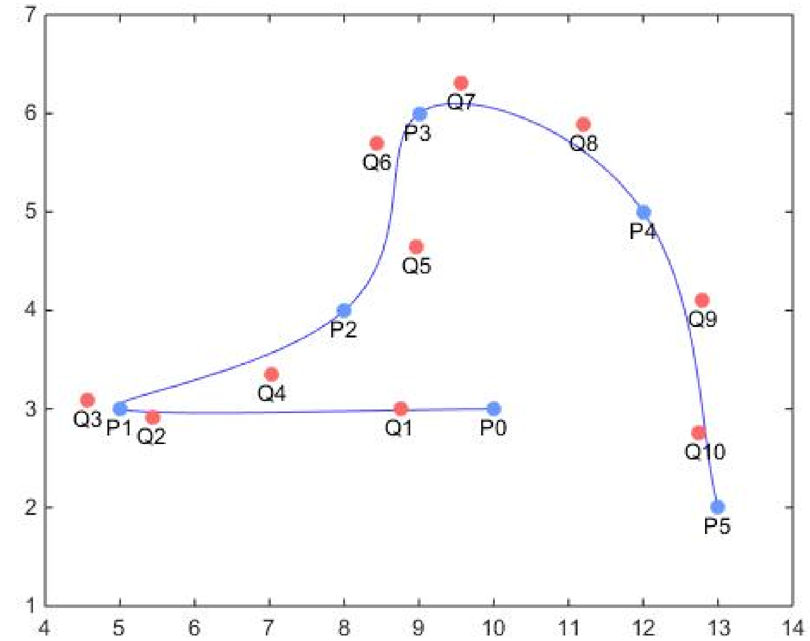
So, only the endpoints P of the composite curve need to be calculated from the material points, after which the middle points Q can be computed from P.

4.2. Surface Marker Method

As mentioned in Chapter 3, the first edge detection method tested is SMM. This method basically reuses the algorithms designed for the material points to find the surface of the soil by placing fake material points (markers) at the surface. These markers move based on the movement of the normal material points, but do not influence the stress-strain relationship. The addition of the markers therefore does not change the location of the material points. The markers are defined to be the end points of the composite Bézier curve and a surface can thus be defined based on the location of the markers. The method mainly reused the algorithms designed for the material points and only a couple of changes must be made to implement the method. Additional background information on SMM is given in the next paragraph. This information is followed by three tests, which determine the usability of the method.



(a) Composite Bézier curve with closed ends



(b) Composite Bézier curve with open ends

Figure 4.2: Two example composite Bézier curves $n = 3$ including their control points (P are the end points and Q are the additional control points).

4.2.1. General information on the implementation of SMM

As can be seen in the flow chart of Figure 4.3 the program consists of three stages, namely the initiation stage, the computation stage and the reporting stage.

Within the initiation stage the fake material points are created based on a location specified in a data file. This can either be done by specifying the location of each surface marker by hand or with the usage of an automatic distribution method. This automatic distribution method is capable of assigning markers to straight surfaces or sloped surfaces as can be seen in Figures 4.4a, 4.5a and 4.6a, but this could be extended to different surface types as well. It is currently possible to create multiple edges and specify the markers for each edge separately. This could be useful for a problem which contains multiple moving boundaries.

The computation stage is run within every iteration of the main program. The movement calculated on the background FEM grid from the real material points is also applied to the fake material points and the fake points therefore follow the material. In the calculation step of the MPM program (the Lagrangian phase) the surface markers move according to the moving background grid, which for an explicit MPM would indicate:

$$x_m^{t+\Delta t} = x_m^t + v_m^{t+\Delta t}\Delta t \tag{4.6}$$

where x_m^t, is the current location of the marker, $x_m^{t+\Delta t}$ is the location of the marker after this time step and the velocity $v_m^{t+\Delta t}$ is given by a summation over the velocity of the background grid:

$$v_m^{t+\Delta t} = \sum_{i=1}^{nod} \frac{(mv)_i^{t+\Delta t}}{m_i^t} N_i(x_m^t) \tag{4.7}$$

where $(mv)_i^{t+\Delta t}$ is the impulse of node i and m_i is the mass of that node. $(mv)_i^{t+\Delta t}$ and m_i are calculated during the MPM procedure and can therefore be reused, without any difficulty. Using the movement described by the 'normal' material points also removes the problems regarding the distance between the markers becoming too small (Sethian [29]). If this distance becomes to small and calculations are made with the description of Sethian [29], oscillations can occur. However, because the markers itself are not used to calculate the velocities, oscillations will not occur. This addition to MPM is fast and uncomplicated as it only reuses previous sections of code needed for the MPM method.

During the reporting stage the location of all the surface markers are combined to form the edge. For this method the assumption is made that the boundary will only be stretch and moved but will always stay as a convex curve. The surface markers are therefore connected to the same neighbouring markers during the entire deformation process. This might need to be changed in more difficult simulations where tumbling might occur. The problem will be addressed in more detail in Section 4.2.2.

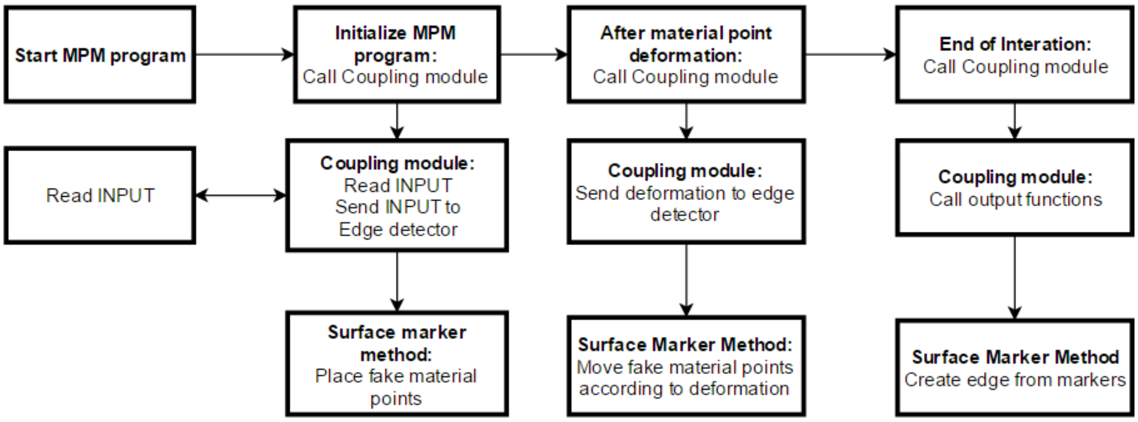


Figure 4.3: Flow chart of SMM

Table 4.1: Muliplication factors of the applied displacement at the left boundary.

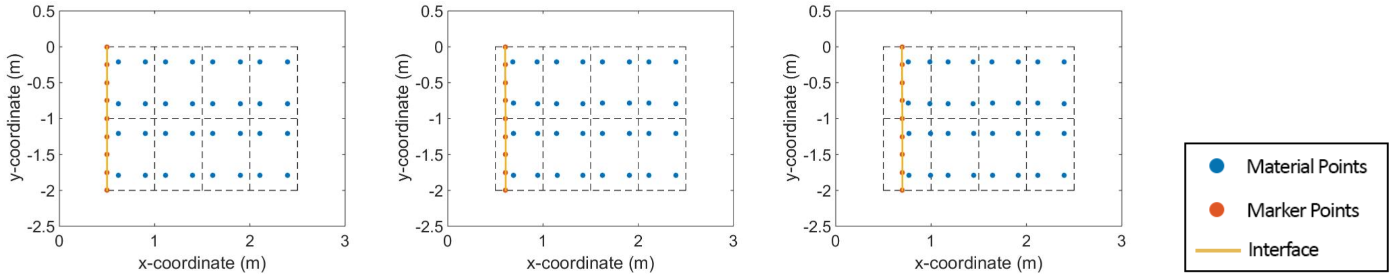
Node i	1	2	3	4	5
Multiplier	1	2	4	2	1

4.2.2. Results of SMM

SMM has been tested on two axi-symmetrical problems as well as a slope stability problem.

Axi-symmetrical fixed displacement

The initial testing was performed on a small axi-symmetrical elastic-plastic MPM program. The plasticity is defined by a Mohr-Coulomb soil model. Only 8 8-node quadrilateral background mesh elements have been used together with 32 material points. Therefore, 4 material points have been used per cell. In this test a fixed displacement was applied to the left side of the domain and the surface markers were initialized at this location as well. The boundary condition is applied to the background nodes without any transformations based on the location of the boundary. The inclusion of a force based on the location of the boundary will be discussed in Chapter 6. The boundary is detected correctly during the complete test as can be seen in Figure 4.4. The marker points move along with the movement of the material points and the distance between the boundary and the first material points decreases as the material compresses. The displacement is small and constant, which indicates that SMM works well under small displacements.

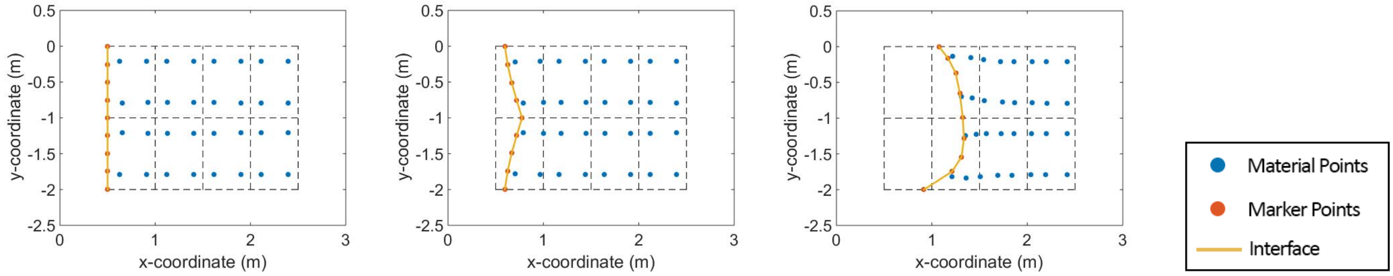


(a) Initial condition (b) Time step 6 (c) Time step 13

Figure 4.4: Axi-symmetrical fixed displacement test

Axi-symmetrical fixed displacement with variable displacements

The second test is almost identical to the first test. However, the fixed displacement of the five boundary nodes was multiplied with a certain factor based on the location from top to bottom, as shown in Figure 4.5. These multiplication factors are given in Table 4.1. So in short, the displacement of the center nodes was increased. In contrast to the previous test the boundary is not detected at the correct location during the complete test. However, as expected, based on the boundary conditions, the material points at the center are compressed further than the material points at the top or bottom. This is also reflected by the marker points in Figure 4.5b. The problem observed during this test is the location of the bottom marker point in Figure 4.5c. This marker should be symmetrical with the marker at the top, which clearly is not the case. Although the origin of this problem is currently uncertain, a logical explanation could be the fact that material points are absent in the cell in which the marker is located. A stiffness relation cannot be constructed in an empty cell, which results in an incorrect approximation of the movement of the marker point. Even though this explanation could explain the incorrect location of this material point, it cannot explain why this problem does not occur at the highest marker. Another reason could be the small deviations from a symmetrical solution in the material points due to plasticity, which unfortunately are to small to be observed from Figure 4.5c. These small deviations might increase to larger deviations at the edges of the cells, resulting in the unsymmetrical shape of the boundary.



(a) Initial condition (b) Correct detection of boundary (c) Incorrect detection of boundary

Figure 4.5: Axi-symmetrical fixed displacement test with an increased displacement at the center nodes

Slope stability problem

As a third and final test the slope stability problem of Figure 4.6 has been evaluated. A 4 meter high slope with a 45 degree slope angle fails under the influence of gravity. The slope is placed within a 30 meter wide trench bounded by roller boundary conditions on all sides. 192 4-node quadrilateral background mesh elements have been used with a size of 0.5 m by 0.5 m, together with 768 material points. The elements along the slope have been deformed to follow the shape of the slope. Marker points have been placed every 0.25 m in the horizontal direction. As can be seen in Figures 4.6b - 4.6d the deformations of the slope are large and the slope collides with the right boundary at the end of the deformation process. The large deformations observed in this problem could occur in reality under for example liquefaction slides. The biggest advantage of MPM over FEM is the performance under large deformations and it is therefore important that the edge detection method can handle these large deformations.

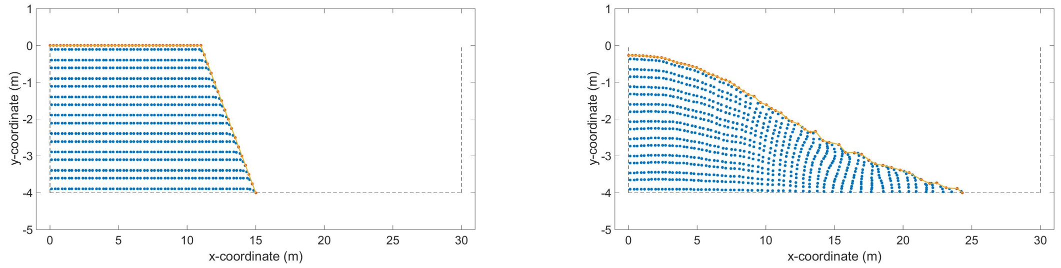
SMM handles the large deformations quite well under the assumption that the material points at the surface remain at the surface. These conditions can be observed in Figures 4.6b and 4.6c. The surface does stretch out and therefore looses the fine details under large deformations, but this problem could be reduced by introducing new marker points if the distance between two material points becomes to large.

However, there are a couple of predictable problems regarding this method, which occur at larger strains. In Figure 4.7, looping of the boundary can be observed. It starts to occur in Figure 4.7a as the second to last marker point overtakes the last material point. This induces a sharp unrealistic corner in the boundary. The problem is more clearly visible in Figure 4.7b, where the markers are completely unable to represent the clash with the right boundary. Looping of the boundary has already been observed by Sethian [29], when he mentioned the boundary detection method. According to Sethian [29], this problem might be solved by smoothing out the velocities to decrease the chance of looping occurring or redistributing the fake points every couple of time steps to ensure a smooth boundary. Smoothed velocities of the markers might reduce the accuracy of the representation of the boundary due to the fact that this smoothing is not be applied to the real material points. Redistributing could solve the problem, however doing this accurately and automatically can be difficult. The redistribution needs to place material points on the real boundary, which is not accurately represented by the markers under large deformations. A different approach would not change the location of the material points, but only change the connection of a material point to different neighbours. This reconnecting is expected to be difficult to do perfectly as well, because it will be difficult to connect the correct markers. A connection based on shorted distance will not work, because this will not connect the correct markers in Figure 4.7.

Moreover, under certain conditions, the marker points do not represent the soil surface, due to the fact that real material points fall on top of the marker points. This occurs when soil falls back on itself during the process, which is a possibility at large strains. This can be seen in Figure 4.7b, where two material points are located above the soil boundary found by the marker points.

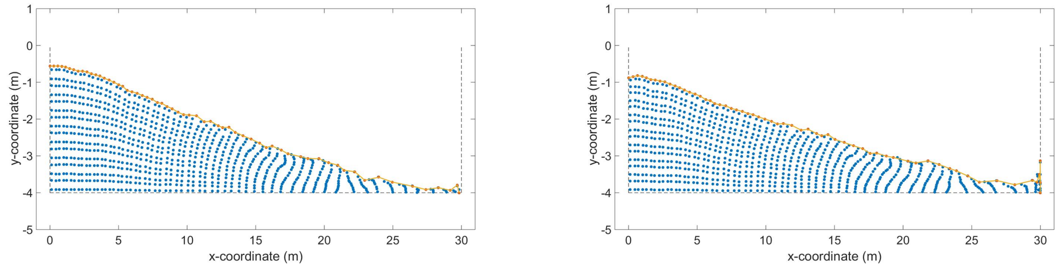
Finally, tearing of the material will also not be tracked, due to the fact that the markers only track the boundary on which they are initialized. New boundaries developing due to the tearing are therefore not tracked. Tearing has not been present in any of the testing cases presented, but this method cannot handle tearing by definition. If tearing occurs from a tracked boundary into the soil, the surface markers might drop into the gap and therefore roughly represent the gap. However, holes forming inside the soil will not be tracked at all.

So although the implementation in MPM might only require a small adaptation of the current programs and will probably be not computationally intensive, it is not certain whether or not this option would indeed give a correct indication of the boundary.



(a) Initial condition

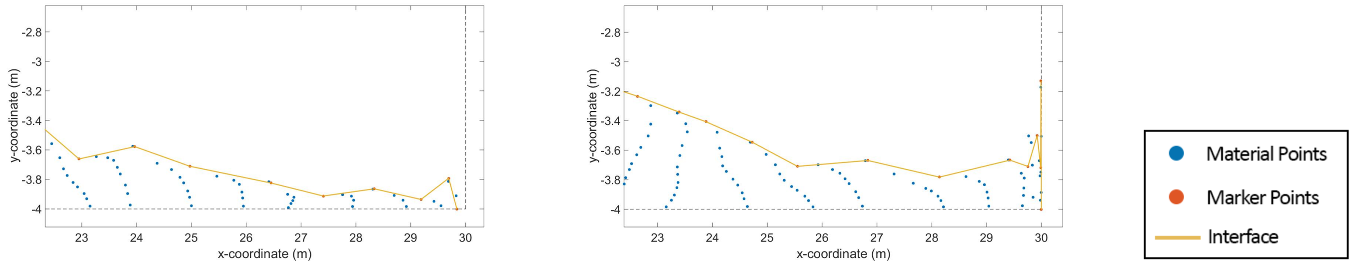
(b) Complete failure of slope



(c) Soil approaches right boundary

(d) Soil collides with the right boundary

Figure 4.6: boundary detection of large deformations of a soil slope with SMM. All the figures have a distorted horizontal to vertical scale.



(a) Detail of the soil reaching the boundary

(b) Detail of the soil colliding with the boundary

Figure 4.7: Detailed figures of problem zones at the right boundary

4.3. Volume of Fluid method

VOF was supposed to consist of three stages as can be seen in Figure 4.8. However, only 2 stages are completed, namely the initiation stage and the computation stage. The third stage could not be completed due to the fact that the method has far too little accuracy. This will be explained in more detail in the results paragraph after the presentation of general background information on VOF.

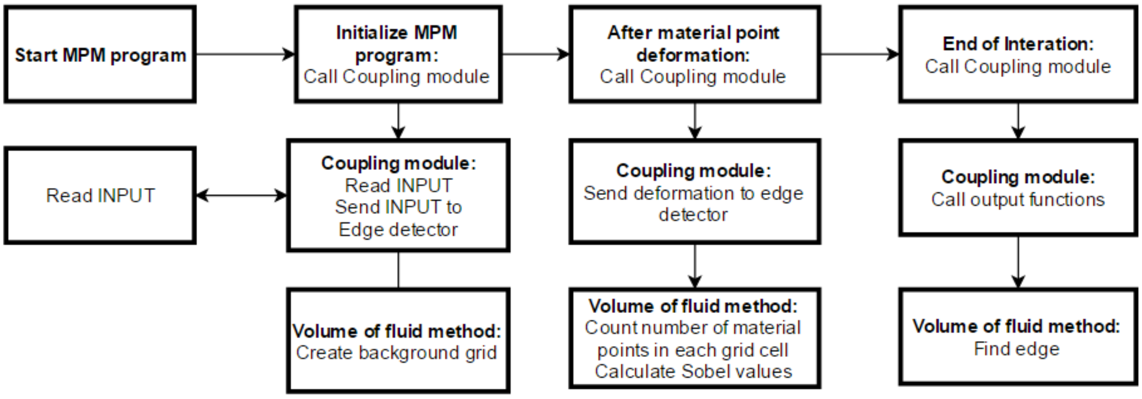


Figure 4.8: Flow chart of VOF

4.3.1. General information on the implementation of VOF

In the first stage the grid of VOF is created. The shape of this grid is the same as the background grid of the MPM computations, but a higher resolution can be used for a slightly higher accuracy. Besides this grid nothing needs to be initialized for this method.

In the second stage both the amount of volume within each cell and the Sobel operators are computed (Pound and Riley [26]). Due to the fact that this method uses the material points to calculate the amount of the material present within each grid cell, the amount of the material present within each grid cell can be approximated by simply counting the number of material points in each grid cell. So the volume of material in a cell (VOC in short) is calculated according to equation (4.8).

$$VOC_i = n_{mp,i} \tag{4.8}$$

where VOC_i is the volume of material in cell i and $n_{mp,i}$ is the number of material points in cell i. In Figure 4.9 a grayscale representation of the VOC of the soil slope tested in the previous section is presented as an example. A more accurate solution could be created by approximating the actual volume of the material present within each cell. This could be calculated based on the location of the material points with respect to the grid and the volume of each material point. Some material points will therefore influence multiple grid cells. Even though this solution will give a higher accuracy the solution is almost identical to PFM and is therefore not implemented. As explained in Section 3.4.2, the Sobel filter is applied to the VOC_i to find the Sobel operator G per cell according to equation (3.36). This can again be represented by a gray scale image as presented in Figure 4.10.

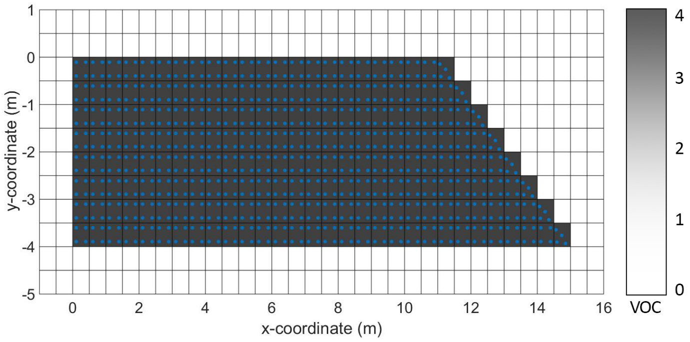


Figure 4.9: Volume of material per cell (VOC)

In the final stage the end points of the composite Bézier curve should be found from these Sobel values. A high G indicates the presence of an edge in the cell. The direction of the edge can be approximated based on G_x and G_y. This direction indicates which boundary cells are connected. Cells with lower G can be chosen as a boundary cell, because of their connection to other boundary cells. In the next paragraph the results of VOF are discussed.

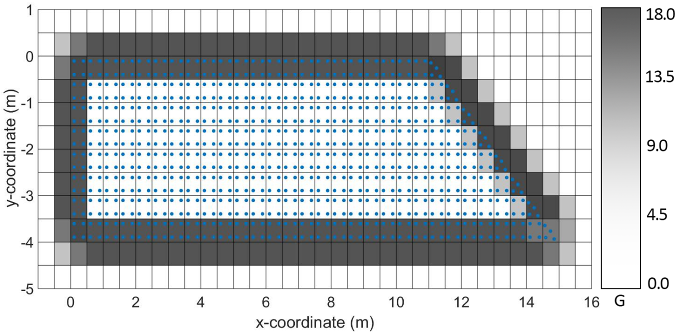


Figure 4.10: Sobel operater per cell (VOC)

4.3.2. Results of VOF

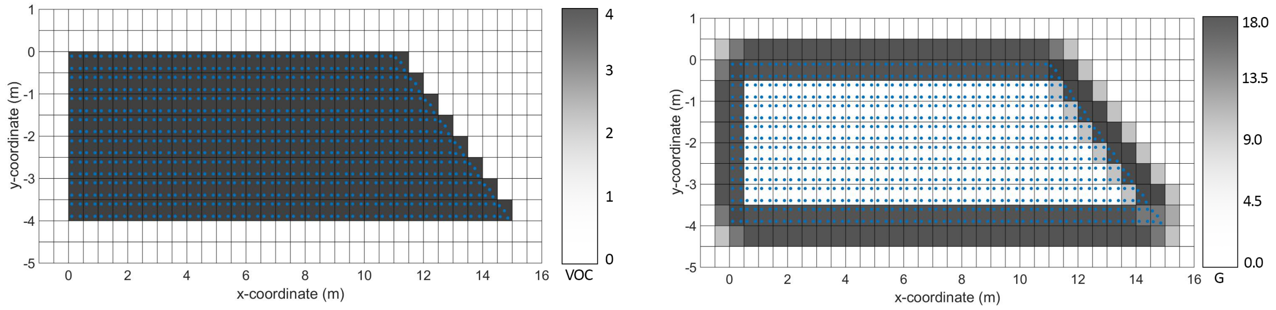
Only the slope stability problem discussed in Section 4.2 will be discussed in this paragraph, due to the fact that it is the only test which provides useful information. In the initial condition, shown in Figures 4.9 and 4.10, the cells which are close to the edge are clearly defined by a high G in comparison to surrounding cells. However, the accuracy is too minimal to locate the end points of the Bézier curves. The method was never supposed to be highly accurate, but its purpose was to find the general location of the edge in a fast and computationally cheap method. Unfortunately the method is not even capable of this task, as it can only find the location of the edge with a resolution of 2 cells as can be seen in Figure 4.10. The method does not have any information where the edge might be within these cells. It could be between the cells or it could lie closer to the edge of one of them. The direction of the boundary from G_x and G_y does not solve this problem, due to the fact that this direction is mainly useful after at least one boundary cell has been detected. The accuracy gives far too little information and is practically useless.

Moreover, G does not define the boundary clearly after deformation as shown in Figure 4.11. Due to the spacing of the material points in the initial condition, the deformed material points are often located along curves. A grid cell can therefore be in between two of these curves and has a $VOC = 0$. On the other hand, a curve can cross one of the grid cells resulting in a large VOC in this grid cell. This uneven distribution in the VOC in the deformed solution results in an incorrect detection with the Sobel operator. For example the Sobel operater finds an edge inside the soil body in Figure 4.11d. This indicates that the method is not only inaccurate in the location of correct edges, but also incorrectly defines cells to be close to an edge.

Moreover, at the start of the development of the VOF method it was expected that the accuracy of the model can be increased by reducing the grid size. This grid size reduction is analysed in Figure 4.12. In the initial condition this appears to work until a minimum grid size, which is equal to the volume represented by the material points in the initial condition, as shown in Figure 4.12b. The method completely breaks at an even smaller grid size, due to the fact that many cells do not contain any material points. The sobel operater than finds gaps through the entire soil body as can be seen in Figure 4.12c. The bandwidth in which the edge will be located can therefore be reduced to a minimum which is twice the size of the volume represented by the material points. This is a result of the minimum grid size (equal to the volume represented by the material points) and the fact that the method finds a bandwidth of two cells in which the boundary could be located.

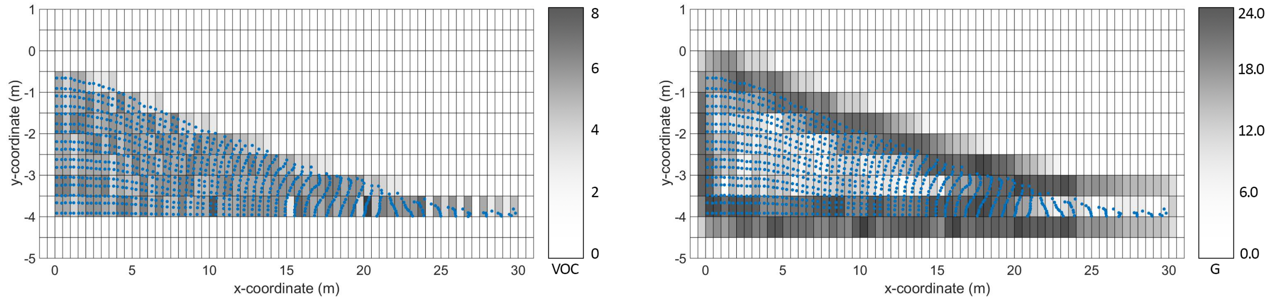
During the deformation process the minimum grid size needs to be even higher, due to the uneven distribution of material points. According to Figure 4.12d, the problem seen in Figure 4.11d increases when the grid size is reduced.

So, in the most favourable situation the bandwidth, in which the edge will be located, will be twice the size of the volume represented by the material points. Additional information is therefore required to place the endpoints of a Bézier curve. Consequently, this method does not meet the requirements for this research and will be ignored in the following chapters. To save computation time in the future this method might be useful for finding a rough location of the boundary so more intensive methods do not need to search through the entire domain. Moreover, the incorrect detection after deformation would still be a problem. Using a different method to speed up a slow method is not part of this thesis due to time constraints and will be left for future research.



(a) Gray scale of VOC in the initial condition

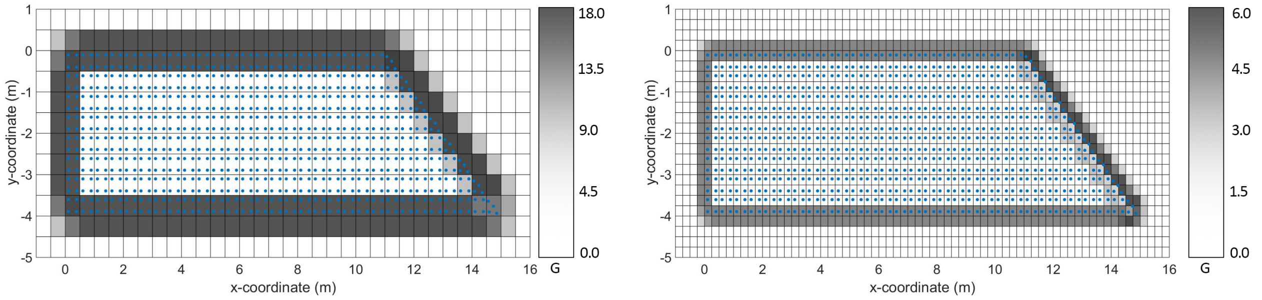
(b) Sobel operator of the initial condition



(c) Gray scale of VOC after deformation

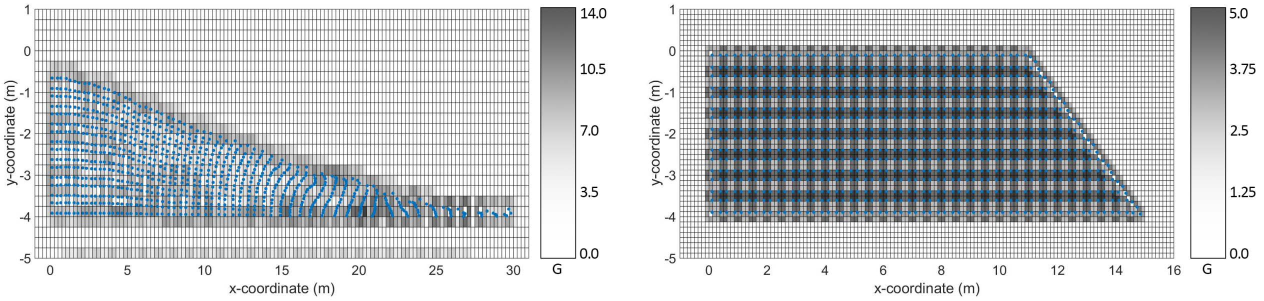
(d) Sobel operator after deformation

Figure 4.11: Boundary detection of large deformations of a soil slope



(a) Sobel operator with grid size: 0.5 m

(b) Sobel operator with grid size: 0.25 m



(c) Sobel operator with grid size: 0.125 m

(d) Sobel operator after deformation with grid size: 0.25 m

Figure 4.12: Effect of the grid size of VOF

The method could possibly be improved to result in an higher accuracy. For example, by not only taking the number of material points in each cell into account, but also the location of each material point within this cell. However, if these modification are included the method will lose its initial purpose of being fast and cheaply computed and the method will more or less represent PFM, which already takes the location of the material points within a grid into account.

4.4. Proximity Field Method

A new implementation of the Level Set Method has been developed for this thesis. The field used in this new Level Set Method is defined by kernel functions around the material points. A zero level of this field of values is found and defined to be the boundary (Sethian [28]). The version of the Level Set Method implemented by Sethian [28] has been discussed in Section 3.4. The new method, called the Proximity Field Method (PFM), uses composite Bézier curve to define the boundary, based on the zero level of the field.

The method for the construction of the field used in PFM is different in comparison to the Level Set Method presented in Section 3.4. The field values are determined based on their proximity (distance) to a material point in PFM. The field is at a local maximum at a material point and is supposed to be zero at the boundary between two materials. So, in a simplified explanation the proximity field will be based on the distance from each material point. A grid is set up on which the field is calculated, after which a contour is drawn through this grid to determine the end points of the composite Bézier curve. A more detailed theoretical explanation of PFM is given in the first couple of paragraphs of this section, after which the results of the method are discussed.

4.4.1. Calculation of the proximity field using kernel functions

In more detail the proximity κ_j around material point j is described by a kernel function, which is assigned to each material point. This kernel describes the influence of this material point to the field. Currently the proximity field is calculated based on a two dimensional kernel function (an Epanechnikov function). The proximity κ_j for material point j is given by a the following kernel function (Soh et al. [32]):

$$\kappa_j(\{\mathbf{u}\}) = C(1 - \{\mathbf{u}\}^T\{\mathbf{u}\}) \text{ for } u^t u < 1 \tag{4.9}$$

Where $\{\mathbf{u}\}$ is a scaled distance vector and C is a constant. The details of these two parameters will be discussed in Sections 4.4.2 and 4.4.4 respectively. This kernel function has a circular baseline. Outside of this baseline the result will be zero and inside the kernel will be calculated. A summation of the proximity κ_j over n material points results in the proximity field K:

$$\mathrm{K}(\mathbf{x}) = \sum_{j=1}^{n} \kappa_j(\mathbf{x}) \tag{4.10}$$

The proximity field K is specified on the coordinate system of the material points, whereas the individual kernel functions use a scaled vector. The transformation between these coordinate system is discussed in Section 4.4.2. Due to the fact that a defined baseline is specified within each individual kernel function, a baseline can also be found in the proximity field. In an ideal situation this baseline would coincide with the edge of the material. The zero level of the proximity field would therefore represent the boundary. However, due to the current configuration of the material points this is not the case and a different base level can be chosen to optimise the localization of the boundary. This will be explained in more detail in Chapter 5.

An simplified example is shown in Figure 4.13, in which four material points and their kernel functions influence three example calculation points. Due to the fact that both calculation point 1 and 2 are inside of a kernel baseline, both points have a non-zero proximity value. On the other hand, point 3 has a zero proximity value, because it is located outside of all the baselines. A points proximity value K can be influenced by multiple material points as is shown by calculation point 2, which is influenced by κ_A and κ_C, i.e. the proximity around material points A and C. Due to the perfect arrangement of the material points in this example no point is influenced by three or more material points. However, this could be the case if the arrangement was less perfect, for example after movement of the material points. Moreover, the kernel functions are circular in this example, while they could be a ellipse in the actual calculations, due to the transformations from local to global coordinates.

The amount of material points in Figure 4.13 is small, but one problem can still be observed, namely the shape of the surface detected with circles. This will not be straight and it is therefore difficult to approximate the exact location of the surface. In reality, the soil surface will not be perfectly smooth, but the roughness size will be much smaller than the roughness of the proximity field. In future research it might be useful to test different kernel functions, to observe which functions can be best used to represent a soil surface.

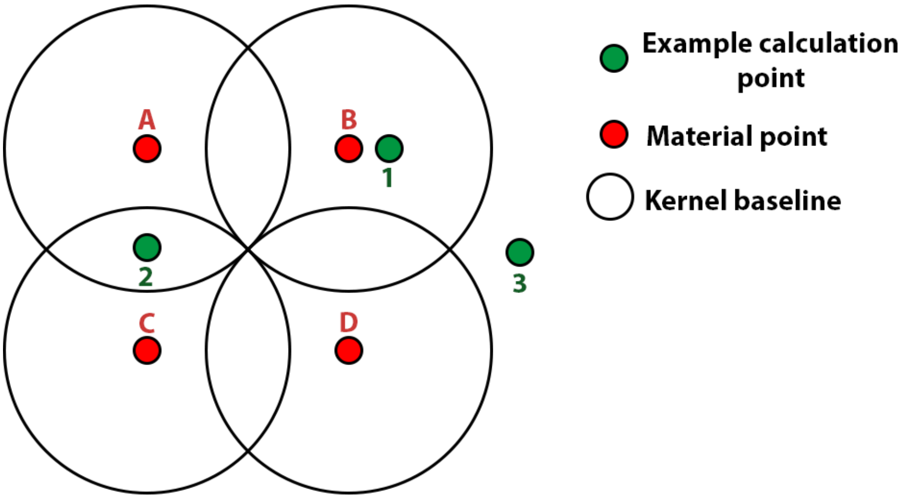


Figure 4.13: Simplified proximity field calculation on 3 random points

4.4.2. Transformation to scaled coordinate system

Normal coordinates can be transformed to scaled coordinates with the shape matrix of the kernel function. The kernel function has an ellipse as baseline in the global coordinate system, which can be transformed to a circular baseline using a scaled coordinate system according to:

$$\{\mathbf{u_{i,j}}\} = [\mathbf{S_j}]^{-1}(\{\mathbf{x_j}\} - \{\mathbf{x_i}\}) \tag{4.11}$$

In which $\{\mathbf{x_i}\}$ is the vector containing the global coordinates of the point i at which the proximity is calculated, $\{\mathbf{x_j}\}$ is the global coordinates vector of the material point j, $[\mathbf{S_j}]$ is the shape matrix of material point j and $\{\mathbf{u_{i,j}}\}$ is the local distance vector between material point j and calculation point i. Due to this transformation, the same formula can be used for each kernel function to calculate the proximity of a kernel, which simplifies the calculation process. A summation of equation (4.9) over all the material points using the local distance vector equation (4.11) results in the proximity of point i. This summation is then repeated for each node of a calculation grid. The position of this grid is the same as the background grid of the MPM method, but a higher resolution can be used to achieve greater accuracy.

The shape matrix $[\mathbf{S_j}]$ defines the shape and size of the baseline in the global coordinate system. The shape of the local kernel function baseline has been defined to be circular with a radius of 1. This local kernel function baseline is visualized in Figure 4.14b. a and b in this figure are defined to be the length of the unit vector or 1 meter in the test executed in this thesis. The shape matrix can transform this circular shape to ellipses of any size and rotation. The result is a global kernel function, which baseline is visualized in Figure 4.14a. In Figure 4.14a, c and d can have any arbitrary length and can be rotated around an angle θ.

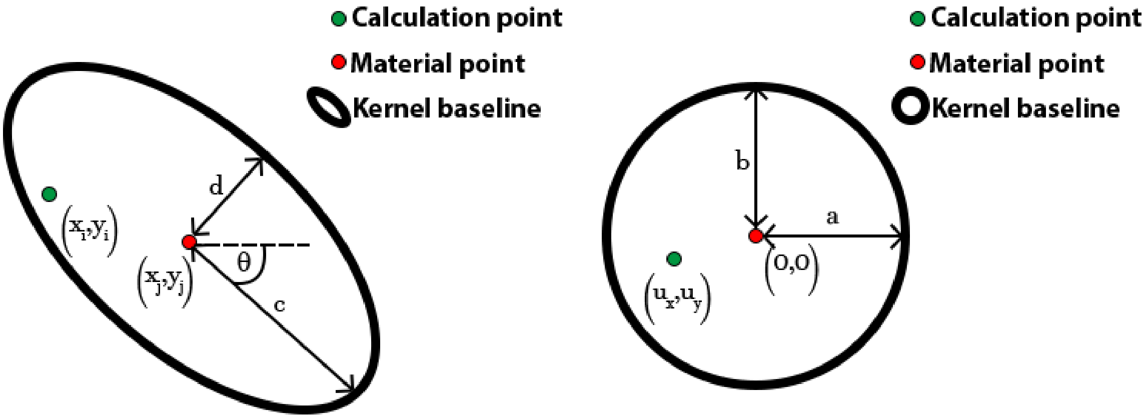
The shape matrix consist of two components, namely a rotation matrix $[\mathbf{R}]$ based on the rotation of the kernel and a diagonal matrix $[\mathbf{A}]$ consisting of the size of the kernel functions baseline, in short $S = R * A$. To transform the global system of Figure 4.14 to the scaled coordinate system the following transformation must be performed:

- To achieve the correct conversion from global to local coordinates matrix $[\mathbf{A}]$ must therefore be filled with the size of the ellipse in both directions:

$$[\mathbf{A}] = \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix} \tag{4.12}$$

- A rotation to this baseline can be applied to the kernel with the rotation matrix $[\mathbf{R}]$, which would rotate the axis of the ellipse in comparison to the cardinal directions of the global coordinates. $[\mathbf{R}]$ describes a counter clockwise rotation, so the rotation angle described in Figure 4.14 is negative. Given that $\phi = -\theta$, $[\mathbf{R}]$ can be defined for this transformation:

$$[\mathbf{R}] = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{4.13}$$



(a) Global Kernel Baseline (b) Local Kernel Baseline

Figure 4.14: Transformation of a global kernel function to a local kernel function

Multiplying $[\mathbf{A}]$ and $[\mathbf{R}]$ results in a full 2 by 2 matrix for every case except the cases when c and d coincide with the cardinal directions. At $\phi = 0, \phi = \pi$ the matrix is filled with c and d on the diagonal and at $\phi = \frac{1}{2}\pi, \phi = \frac{3}{2}\pi$ the matrix is filled with c and d on the off diagonals.

The shape of a kernel is not constant. Depending on the settings of the method, the ellipse can grow and shrink based on strain of the material point and even rotate based on the shear strain of the material point.

4.4.3. Deformation

The difficulty arises once the material starts to compress, expand or distort. Due to the fact that the shape of the material is not stored in the material points, it is difficult to reproduce an exact proximity field. However, a simple shape is assumed, so a proximity field can be constructed based on this simple shape. Based on the strain of the material points the kernel function can be stretched and rotated to approximate the material around the point. The stretching will be dependent on the strain in cardinal directions and a rotation could be used to represent shear strain. A deformation of the example given in Figure 4.13 is shown in Figure 4.16.

Mathematically the deformation process can be achieved by modifications of the shape matrix $[\mathbf{S_j}]$. Stretching or compressing the kernel function can be achieved by multiplying the shape matrix with a stretch matrix:

$$[\mathbf{A_j}] = \begin{bmatrix} 1+\epsilon_{x,j} & 0 \\ 0 & 1+\epsilon_{y,j} \end{bmatrix} \tag{4.14}$$

where $[\mathbf{A_j}]$ is the stretch matrix of material point j, $\epsilon_{x,j}$ is the strain increment in the x-direction of material point j and $\epsilon_{y,j}$ is the strain increment in the y-direction of material point j. A rotation can be applied in the same way based on the shear strains:

$$[\mathbf{R_j}] = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{4.15}$$

where $[\mathbf{R_j}]$ is the rotation matrix of material point j, ϕ is the rotation based on the shear strains. Under the assumption that shear strain increments are small, the rotation angle can be approximated with

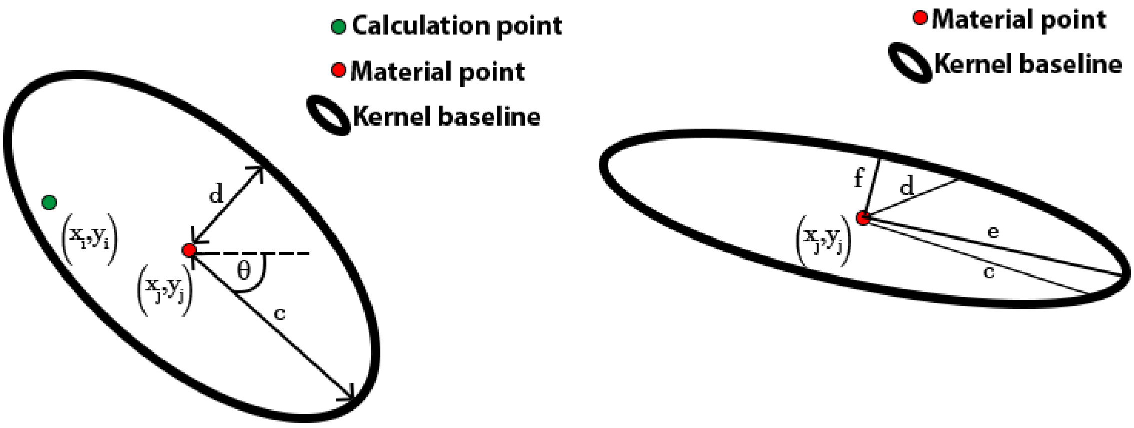
$\phi \approx \epsilon_{xy,j}$. $\epsilon_{xy,j}$ is the shear strain increment of material point j. Under this same assumption the trigonometric functions can be approximated as: $\sin(\phi) \approx \epsilon_{xy,j}$ and $\cos(\phi) \approx 1 - \frac{{\epsilon_{xy,j}}^2}{2}$. The order of the multiplication with the stretch and rotation matrix are important for which a choice needs to be made. A large and small strain option are available, which are described below.

Large strain

The large strain approach does not restrict the strains of the material points. The strain is applied in increments on the shape matrix. So during every time step the shape matrix $[\mathbf{S_{t-1}}]$ is multiplied with the deformation matrices $[\mathbf{A_{j,t}}]$ and $[\mathbf{R_{j,t}}]$ to form the new shape matrix $[\mathbf{S_t}]$:

$$[\mathbf{S_t}] = [\mathbf{R_{j,t}}][\mathbf{A_{j,t}}][\mathbf{S_{t-1}}] \tag{4.16}$$

Deforming a rotated ellipse with this large strain definition is problematic, due to the fact that the stretching operation does not act along the axis of the ellipse. This results in an ellipse shown in Figure 4.15, where vectors $\{\mathbf{c}\}$ and $\{\mathbf{d}\}$ no longer represent the axis of the ellipse. An ellipse is therefore no longer described by shape matrix $[\mathbf{S_t}]$. This can either be ignored and the undefined shape described by this matrix could be used by PFM or the principal axis of the ellipse must be found. The decomposition to principal axis can be seen as vectors e and f in Figure 4.15b. Both approaches have not been tested until this point, but instead the small strain approach given below has been applied.



(a) Timestep: t_{i-1}

(b) Timestep: t_i

Figure 4.15: Large Strain Deformation

Small strain

The small strain definition does assume a restriction on the strains of the material points. It is therefore less usable in the MPM method due to the large strains the model is often used for. However, the implementation of the approach is much simpler. Instead of incrementally applying the deformation, the complete deformation process, from the start of the computation until this iteration, is applied. The deformation of the kernel always starts on the shape of the kernel function in the initial condition. In succession the total normal strains and the total shear strains are applied to the initial condition, with a stretch and rotation matrix respectively:

$$[\mathbf{S_t}] = [\mathbf{R_{j,t-t0}}][\mathbf{A_{j,t-t0}}][\mathbf{S_{t0}}] \tag{4.17}$$

where $[\mathbf{R_{j,t-t0}}]$ and $[\mathbf{A_{j,t-t0}}]$ represent deformation matrices based on total strains. Total strains can still be larger than the incremental strains used in the large strain definition and is therefore not advised to use the small angle trigonometric functions. During every iteration the kernel is deformed from Figure 4.14b to Figure 4.14a. This small strain approach can be implemented incrementally as well, which reduces the required memory of this method to only the shape matrix $[\mathbf{S_t}]$. In the incremental definition the vector c_{t-1} and d_{t-1} are stretched using the horizontal and vertical strain respectively to find c_t

and d_t. Vectors c_{t-1} and d_{t-1} can be obtained from the columns of the shape matrix and c_t and d_t are placed back into the shape matrix. Afterwards the new shape matrix is rotated based on the shear strain:

$$[\mathbf{S_{t,sheared}}] = [\mathbf{R_{j,t}}][\mathbf{S_{t,normal}}] \tag{4.18}$$

The approach is therefore similar as the large strain definition, due to the fact that a incremental normal strain is applied followed by a incremental shear strain during each iteration. The difference is the direction of the normal strain. Stretching along the coordinate axis is used in the large strain definition, where as stretching along the ellipse axis is used in the small strain definition.

Problems during deformation

A couple of problems immediately arise once the material starts to deform as can be seen in Figure 4.16. Gaps in the proximity can occur due to the fact that ellipses are used. The unfilled space can occur even if the initial condition did fill the complete area. In real soil deformation holes can still occur, but they will probably only occur, when the distance between the material is much larger. This therefore needs to be corrected in the proximity field or ignored during the interpretation of the field to find the boundary. This problem is addressed further in Section 5.3

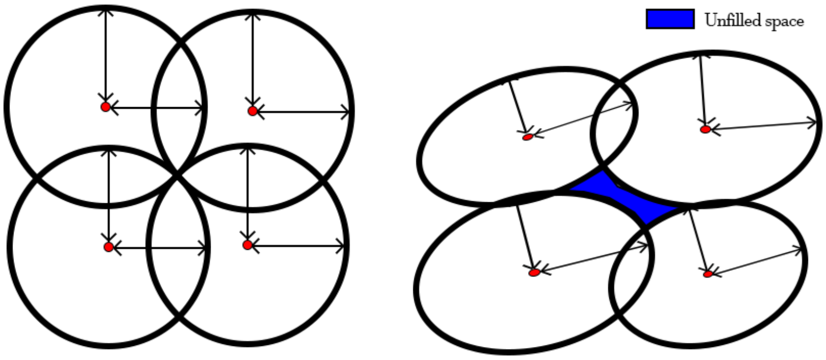


Figure 4.16: Unfilled space under deformed kernel functions.

4.4.4. Kernel function constant

The constant C in equation (4.9) is dependent on the size and shape of the kernel function. Its dependency on the size and shape of the kernel function is a result of the demand to have a constant volume beneath the kernel function (in this case a volume of 1 has been chosen). This constant volume represents the mass of the material described by the material point of the kernel function. The material point is assumed to describe the same material during the computation. Due to the fact that the mass of material remains constant for each material point, the total volume of the kernel should remain constant as well. To achieve a constant volume of the kernel function, the constant C is governed by the following equation:

$$C = \frac{2}{\pi \det(S)} \tag{4.19}$$

where S is the shape matrix specified in Section 4.4.2.

4.4.5. Proximity field grid

The kernel functions define the proximity field in the entire domain. To actually find the boundary, the field is only calculated on a grid. This grid can be the same size and resolution as the background grid of MPM, but the resolution can be increased by decreasing the size of the cells. Another technique, which could be tested in future research, would calculate the proximity of a material point with respect to the other material points. However, this concept was only thought of after the initial programming had been completed and it has therefore not been implemented.

The calculation of the proximity field is uncomplicated. Equation (4.10) is calculated for each node on the grid (an eight node grid has been chosen). An example proximity field cell with proximity values assigned to each node is given in Figure 4.17. A nine node cell is visualized in this example, but only eight nodes are used in the localization of the end points. The ninth node is only used in an improved connection algorithm, which connects the edge points based on the value of this middle node. This algorithm is discussed in Section 5.1. A contour level is chosen and the crossings of this contour level with the grid cell boundaries are found by reducing the proximity field K with grid value z to form H:

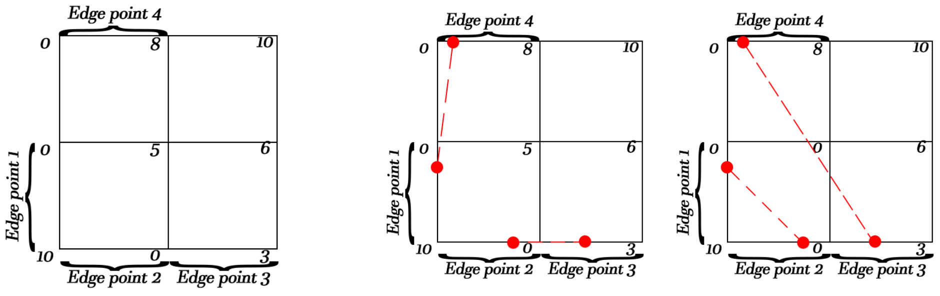
$$H = \mathrm{K} - z \tag{4.20}$$

Theoretically a contour level $z = 0$ is correct, due to the fact that the boundary is theoretically located at the zero level of the proximity Field. However, in practise increasing the size of the kernel functions, therefore increasing the value of the proximity Field, and using a $z > 0$ can result in a better approximation of the boundary.

Based on the nodal values of H in example Figure 4.17 four boundary points are required on the vertices of this cell, because on each vertices which connects a negative or zero valued node ($H \leq 0$) to a positive node ($H > 0$) a boundary point should be created. The method evaluates every grid cell in succession end finds all the endpoints of the Bézier curve.

The exact location of the edge point is found with an iterative process, which is necessary due to the non-linearity of the kernel function. A visualization of the process is given in Figure 4.18. During this iterative process the proximity is calculated at the middle between points a and b. a and b are the edge nodes at the start of the iterative process. After each step either a or b is replaced by the middle point dependent on which section contains the boundary point. In the example during iteration 1 and 2 point b is replaced and point a is replaced afterwards. Only five points are displayed in this example for clarity of the figure, but in reality a maximum number of iterations can be defined as well as a maximum error of H is defined. Currently a difference of 0.0001 in comparison to the exact contour value is allowed, while the proximity field maximum is greater than 10. Moreover, the maximum number of iterations is set to a 1000, resulting in an extremely small location error when this maximum is reached (in the order 2^1000). The method is far from efficient as visualized in Figure 4.18 as well, iteration point 2 almost matches the exact location, but the process moves further from the location in the next iteration, after which it starts to move back towards the exact location. More efficient methods can be developed to optimize the computation cost of the method, which will be addressed in Section 5.2.

Finally the edge points have been connected, which has proven to be complicated if the number of edge points is large than 2. Luckily the number of edge points will always be even, due to the fact that the number of vertices where only one of the two nodes is zero must be even. The connection algorithm for two end points is trivial, because they must be connected to each other. However, up to 8 edge points could be present on an eight node element, without a clear indication which points should be connected. In the version tested in this section the final connection algorithm was already implemented. This algorithm connects two edge points based on the proximity of the middle node (the ninth node) of the cell. In Section 5.1 a further explanation is given on this methods as well as the previously used algorithms and the limitations of all the methods.



(a) Proximity field grid (b) Connection of Bézier curve endpoints

Figure 4.17: Example of a proximity field cell

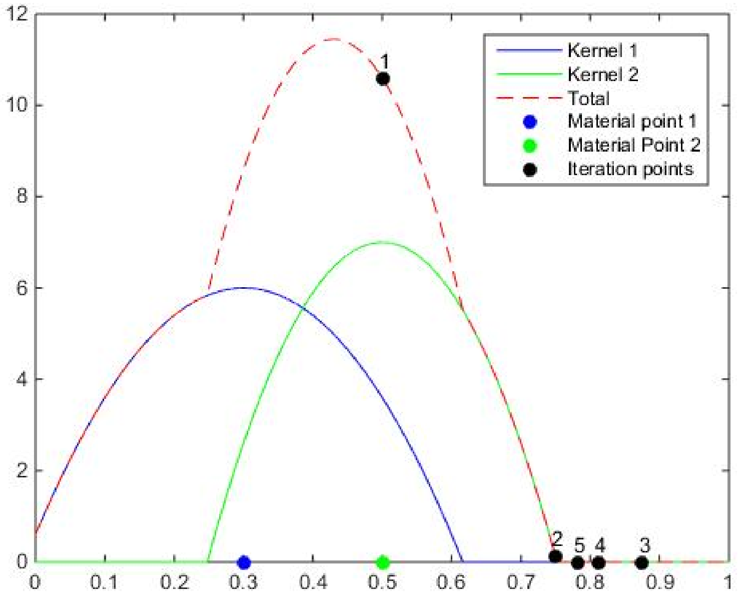


Figure 4.18: Iterative process to find boundary points in PFM.

4.4.6. General information on the implementation of PFM

Unlike the two previous methods this method has been separated into four parts as can be seen in Figure 4.19. In the initialization phase of PFM a grid is constructed on which the proximity is calculated. Besides the initialization of the grid, the size of the proximity kernel for each material point is also calculated. This size is based on the amount of material around the material point in the initial condition and can be modified by a factor in each cardinal direction.

$$c_j = \frac{f_x \text{Cellsize}_x}{/} 2n_x \tag{4.21}$$

where c_j is the size of the kernel of material point j, Cellsize$_x$ is the size of a cell in MPM, f_x is a user specified factor and n_x is the number of material points per cell in the x-direction. The same can be done to calculate d_j for the y-direction. f_x can be used to achieve a maximum coherence of the edge with the initial condition.

The second part of the method updates the size of each kernel based on the strains of the material points during a time step of the MPM method. This is done for each material point individually. The proximity field grid and the size of the kernel are then used in the third phase to compute the proximity field at the end of each MPM time step. On each grid node the proximity is calculated.

In the final phase a contour function finds the boundary from this proximity field grid. A boundary level is specified within a data file to find the location of the edge. In the ideal situation this level is the zero level of the field, but due to the current configuration of the material points this is not the case. A different boundary level, its value depending on the application, often gives a better result. This will be explained in more detail in Section 5.3. The contour function first finds the cells in which a part of the boundary is located. An iterative process is used to pinpoint the boundary to a specific location. The points are then connected to form an boundary with linear connections. Depended on the order of the Bézier curves intermediate points might be placed in between the point found by PFM, but this is handled by the coupling module as mentioned before.

4.4.7. Results of PFM

PFM was initially tested on the same problems as SMM, namely on two axi-symmetrical problems as well as a slope stability problem.

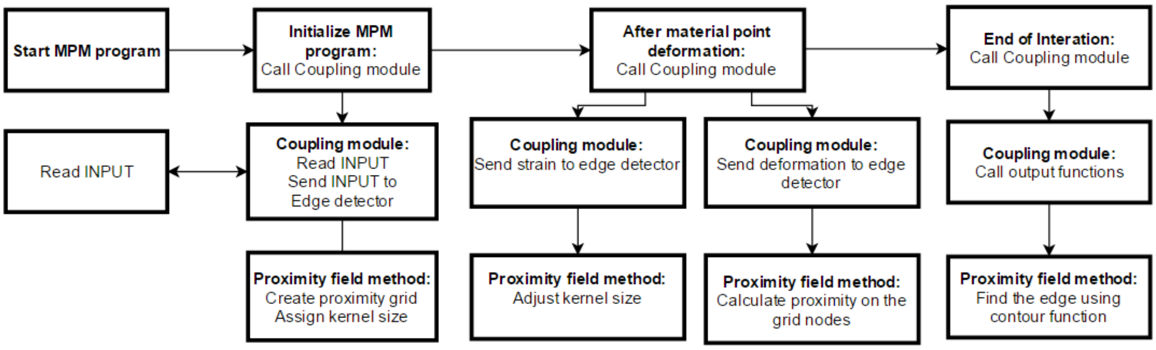
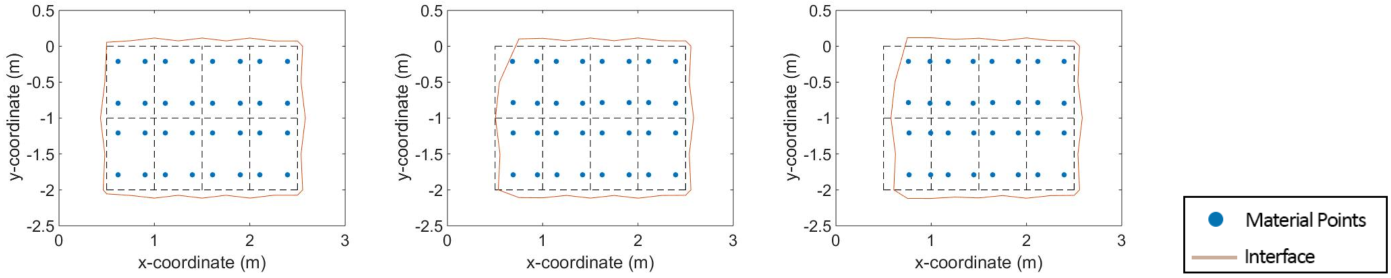


Figure 4.19: Flow chart of the proximityfield method

Axi-symmetrical fixed displacement

The initial testing was performed on a small axi-symmetrical MPM program. The boundary isn't detected as accurately as SMM (Figure 4.20 can be used as a reference). In the initial condition the boundary is not straight, which can be explained with the shape of the kernel functions. Moreover, the boundary is slightly non symmetrical after the deformation process. This indicates that PFM is a decent approximation of the surface, but exact solutions will never be found using this method. Even under these simple load conditions the method fails to approximate the surface accurately.

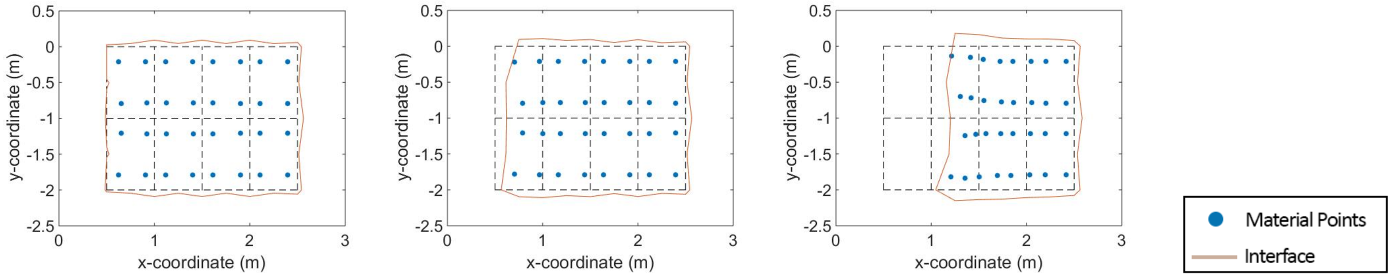


(a) Initial condition (b) Small deviations of boundary (c) Small deviations of boundary

Figure 4.20: Axi-symmetrical fixed displacement test using PFM

Axi-symmetrical fixed displacement with variable displacements

In the second test the center is once again given a higher load as mentioned in Section 4.2. During this test the advantage of PFM starts to be visible as can be seen in Figure 4.21. SMM accurately approximates simple load conditions, while PFM approximates all load conditions with small deviations. As long as the material points are located correctly, PFM will find a surface, which is close to the actual soil surface.

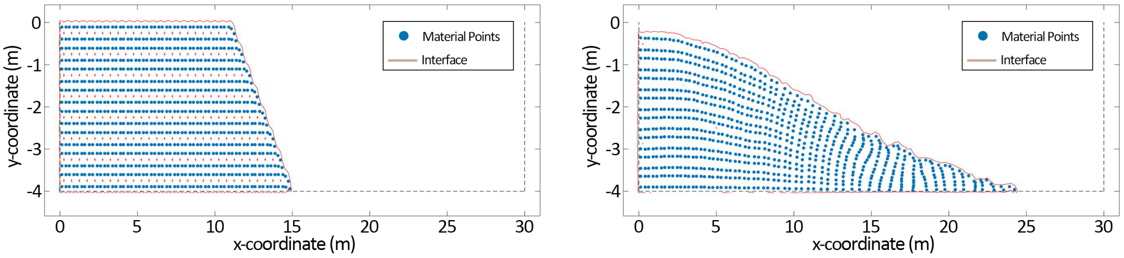


(a) Initial condition (b) Small deviations of boundary (c) Small deviations of boundary

Figure 4.21: Fixed displacement test with an increased displacement at the center nodes using PFM

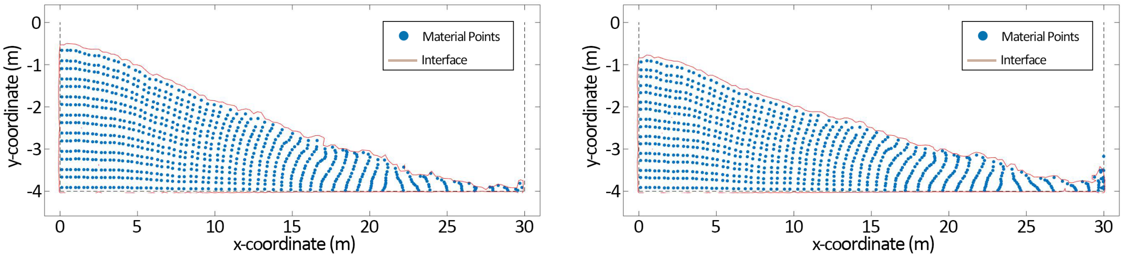
Slope stability problem

As a third and final test the slope stability problem of Figure 4.22 has been evaluated. A 4 meter high slope with a 45 degree slope angle fails under the influence of gravity. Only one problem with the boundary detection can be observed in Figure 4.22, namely the holes in the initial conditions. Apart from these holes only small deviations from the exact surface location are present. All the problems described for SMM are not present with PFM and is therefore much more stable. The method will still work even if the surface is located in empty MPM cells, which is not guaranteed for SMM. Moreover, the method is capable of detecting new surfaces as the method is only defined as a function of the existing material points.



(a) Initial condition

(b) Complete failure of slope



(c) Soil approaches right boundary

(d) Soil collides with the right boundary

Figure 4.22: boundary detection of large deformations of a soil slope using PFM

4.5. Conclusion

As mentioned in the introduction, one of the two research questions was: Which edge detection method is most suitable for detecting a material boundary in an MPM? After the executed tests it can be concluded that the accuracy of VOF was far too low and is therefore not considered as an option. Moreover, it was discovered that SMM has the most potential for simple load conditions, but the PFM has a broader spectrum of possibilities. The broader spectrum of possibilities is especially useful for MPM. Tearing of the material and large strain deformation, two of the advantages of MPM, are tracked more or less correctly by PFM, while SMM fails on both of these points. Due to the fact that MPM is specifically made for difficult load and displacement conditions, the detection method should be able to model these problems. SMM is therefore not applicable.

Additionally ,the current implementation of the PFM can still be improved easily to achieve better results. For example the incorrect hole placement could be fixed with a minimum hole size. On the other hand the problems regarding SMM have been encountered to be difficult to correct. Reordering of the surface is similar to re-meshing a FEM grid, which is one of the main problems the MPM model tries to bypass. Finally, one of the disadvantages of PFM is its computation cost. However, the current configuration is far from efficient, so optimizations of the algorithm can reduce this cost greatly. In the continuation of this thesis, the method has already been optimised and some of the problems have been fixed, as is discussed in Section 5.2.

So in short the PFM has been found to have the most potential, even though difficulties still exist with the method. PFM has therefore been chosen to be used in the continuation of the thesis and boundary conditions have been applied using PFM in Chapter 6.

5

Proximity Field Method: Further analysis and improvements.

In Chapter 4 it has been concluded that PFM best fits the requirements of an boundary detection method for an MPM Method. According to the tests executed within Chapter 4 the method had the best approximation of the boundary under large strain conditions of the three methods tested (SMM, VOF and PFM). However, a couple of problems and questions still exist with PFM:

- Bezier end points connect to wrong neighbours under certain conditions.
- The computation cost of the method is too high.
- Unrealistic small holes are described by the method.
- The surface does not coincide with the expected surface in the initial condition.
- The kernel functions are deformed unrealistically under large strains.
- Should the deformation be applied to the kernel functions using the small or large strain approach?

These problems are discussed within this section and have either been solved or been reduced. To discuss these problems additional testing has been performed to further analyse the method.

Most of the testing has been performed on a slope stability problem as well as a bearing capacity problem. The two-dimensional slope profile is visualized in Figure 5.1. As can been seen, a 4 m high slope is loaded by gravity and no additional load has been placed on the slope during the testing in this chapter. The soil parameters used in the test are given in Table 5.1. The background grid elements are 4-node quadrilateral elements with a size of 0.5 m by 0.5 m. Four material points are placed inside of each grid cell. The problem uses 1150 background grid elements. From the 1150 elements 842 elements contain material points in the initial condition resulting in 3368 material points. The proximity field grid elements have the same size as the background grid. The development of a contour algorithm has been performed on different slope stability problems, due to the fact that the MPM program used in the problem described above was unfinished during the development of a contour algorithm.

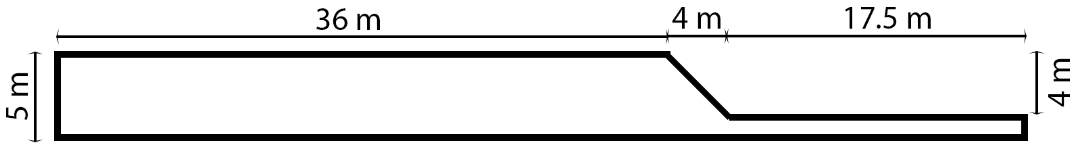


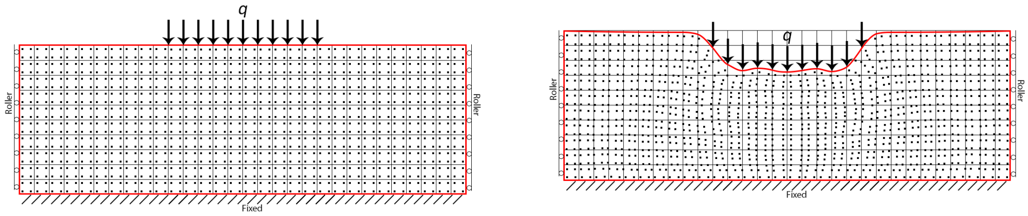
Figure 5.1: Two-dimensional slope profile

The application of a surface traction to a moving boundary will be discussed in Chapter 6. However, a bearing capacity problem, which uses a surface traction on a moving boundary, is used within this

Table 5.1: Parameters used in the slope stability test

Parameter	Symbol	Value
Young's modulus	E	10^3 (kPa)
Poisson's Ratio	ν	0.33 (-)
Volumetric weight	γ	20 (kN/m^3)
Initial Cohesion	c_i	20 (kPa)
Peak Cohesion	c_p	10 (kPa)
Residual Cohesion	c_r	5 (kPa)
Surface traction	q_v	0 (kPa)

chapter to explain problems regarding the deformation of the PFM kernels under a large load. Therefore, a short explanation of the bearing capacity problem is given below. The problem is explained in further detail in Section 6.2. As can be seen in Figure 5.2, a surface traction is applied to the center of a 15 m wide soil body. The height of the soil body is 5 m and is bounded by a fixed bottom boundary and roller boundaries on the left and right of the domain. The soil properties are similar to Table 5.1, but $\nu = 0.15$ and $q_v = 50$ kPa. The background grid of MPM uses 4-node quadrilateral elements with a size of 0.5 m by 0.5 m. Four material points have been placed inside of each grid cell. In total 300 background grid elements are used together with 1200 material points. An additional 150 empty background grid elements have been used on top of the domain to ensure that the material are inside of the background grid. PFM again used the same grid size.



(a) Surface traction applied to the initial condition

(b) Surface traction applied during the deformation process

Figure 5.2: Surface traction on a rectangular soil body

So, in this chapter improvements to PFM are made. The testing required for these improvements is performed on a slope stability or bearing capacity problem. After the improvements, a comparison with FEM is made at the end of this chapter.

5.1. Connection of the boundary points

As mentioned in Chapter 4 the boundary points can be connected in many different figurations and an algorithm has been developed to connect the points correctly. In this section the development process of the algorithm is discussed by explaining and comparing all the algorithms which have been used at one point during the development. Finally, a method called the Triangle Connection Algorithm (TCA), which connects the points based on the proximity of the middle node of the grid cell has been observed to be the most stable. The method is called the Triangle Connection Algorithm due to the fact that it uses the middle node to separate the cell in eight triangles and determine the correct connection based on those triangles. However, a couple of different algorithms have been tested before TCA was developed. All the algorithms are discussed, because TCA has some flaws, as discussed later in this section. In future research ideas from the currently unused algorithms might prove to be useful or on the other hand an algorithm can possibly be ignored due to the flaws mentioned in this section.

A short recap on the iteration algorithm of the boundary points is given, as it was the starting point of these connection algorithms. Moreover, a couple of the algorithms expand the ideas used to locate the boundary points to connect them as well. A more detailed explanation on the iteration of the edge points is given in Section 4.4.5. For each node of the grid cell the proximity field K is calculated, after which it is reduced by the fixed contour level z:

$$H = \mathrm{K} - z \tag{5.1}$$

where H is the reduced proximity field. The contour level z can be negative. The edge points are found on the boundaries of grid cells in a clockwise order starting from the lower left of the grid cell. On each vertex where H shifts between $H \leq 0$ and $H > 0$ an edge point is placed. As mentioned in Section 4.4.5 the exact location at which the proximity value shifts on these vertices is found by iteration over the vertex. In Figure 5.3 the nodes of a proximity grid cell have been coloured based on the value of the H. The green nodes represent $H > 0$ and the red nodes $H \leq 0$. As can be seen, four edge points have been found in this grid in a clockwise order. The numbers next to the edge points indicate the order in which they were found. These four edge points can be connected using the algorithms described in the next paragraphs.

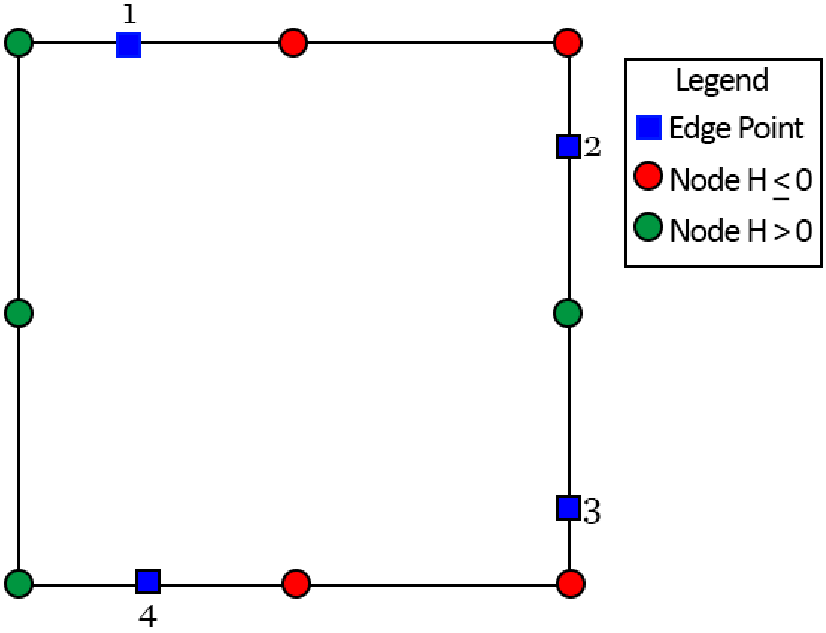


Figure 5.3: Edge points on a grid cell, which need to be connected

5.1.1. Connection based on the order of appearance

The first implemented algorithm connected the boundary points based on order of appearance. In the first version of this algorithm $n - 1$ connections were made between n edge points found on the boundaries of a grid cell. The connections are made between edge point i and $i + 1$, where i ranges from $i = 1$ until $i = n-1$. In Figure 5.4 the algorithm has been executed on the sample cell of Figure 5.3. The algorithm has also been tested on a slope stability problem, which is shown in Figure 5.5. Under most circumstances only two edge points are detected per cell (the trivial connection problem) and the algorithm works perfectly under these circumstances. One of the instances in which the algorithm connects all the points correctly is shown in Figure 5.5a. The only problems visualized in this figure are the small holes in the soil body. However, the holes are not the result of the connection algorithm, because they are the result of holes in the proximity field. A solution for these gaps is discussed in Section 5.3.

However, the algorithm needs further improvements to remove unnecessary connections when more than two edge points are present in a cell, as shown in Figure 5.5b. An additional connection is drawn, which separates the slope in two soil bodies. For true separation a gap should occur in between the two soil bodies. However, in this case the separation is not the result of a gap, because the two soil bodies are still connected. The drawn connection, which separates the two soil bodies, is therefore incorrect. Moreover this jointed contour can break the boundary condition application algorithms resulting in an unreliable model.

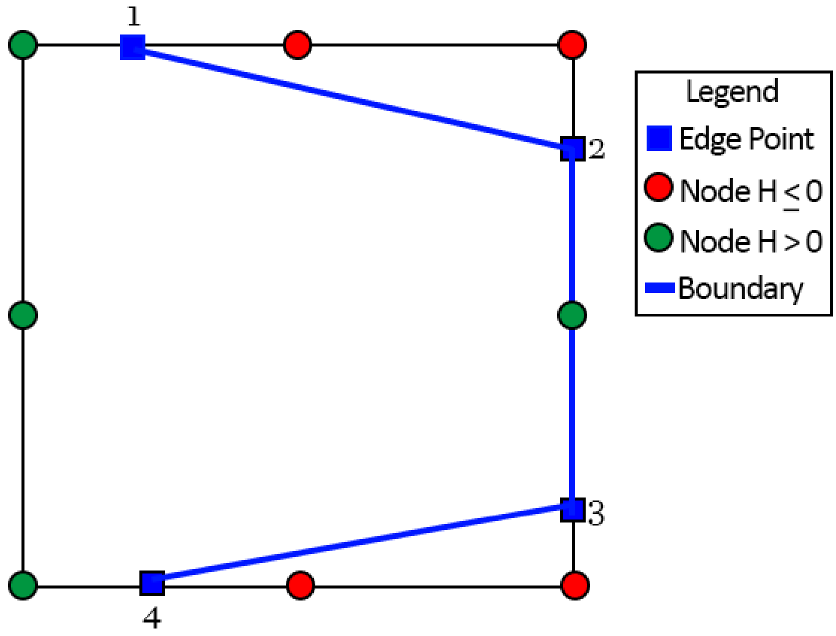
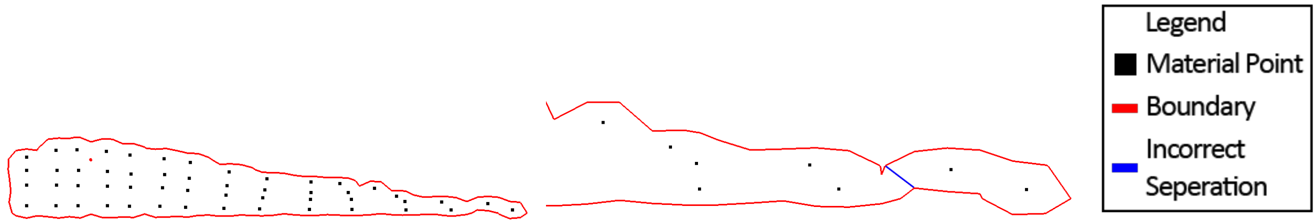


Figure 5.4: Example grid cell for connection of the edge points based on the order of appearance



(a) Overall good connection of the edge points (b) Incorrect separation of the soil bodies (c) -

Figure 5.5: Slope stability test for connection of the edge points based on the order of appearance

5.1.2. Connection based on the proximity along the connection

In an attempt to fix the connection algorithm based on the ordering of the appearance the connections are drawn based on the slope of the proximity field along the connection. The algorithm once again attempts to draw $n-1$ connections for n edge points based on the slope. For each possible connection (six connections are available in the example in Figure 5.3) the slope of the proximity field is calculated. This slope is approximated by a linear approximation of the proximity field:

$$\frac{dH}{ds} \approx \frac{H_m}{s_m} \tag{5.2}$$

where $\frac{dH}{ds}$ is the slope of the reduced proximity field, H_m is the reduced proximity field value at the middle of the connection and s_m is the distance to the middle of the connection.

The algorithm draws the connections with the gentlest slope, because these should represent the connections along the boundary. Along the boundary the proximity field value should stay at the contour value, therefore resulting in $\frac{dK}{ds} = 0$. However, the boundary is approximated as a straight line, whereas a curved boundary is likely to occur, resulting in $\frac{dH}{ds} \neq 0$. The algorithm therefore not only draws the boundaries with $\frac{dH}{ds} = 0$, but draws the $n-1$ connections with the smallest $\frac{dH}{ds}$.

Moreover, a contour can be approximated by a continuous function without joints and the surface found by the algorithm should obey these rules. One additional rule is therefore applied, which only allows a maximum of two connections per edge point.

Table 5.2: Proximity field slope per connection of Figure 5.6

Edge point	Edge point	Slope of the proximity field
3	4	0.05
2	4	0.4
1	4	0.8
1	2	1.2
1	3	2.2
2	3	6.7

The algorithm has been applied on the example cell of Figure 5.3 and the result is shown in Figure 5.6. The slope of the proximity field for each connection of this example is given in Table 5.2 ordered from lowest to highest. According to the gentlest slope connections $3-4, 2-4$ and $1-4$ should be drawn. However, this would connect edge point 4 with 3 connections and connection $1-4$ is therefore replaced by connection $1-2$.

Unfortunately, the addition of the slope of the proximity field to the connection algorithm often removes the wrong connections as can be seen in Figure 5.7. Apparently, the proximity field is two curved for this approach as it approximates the field as having straight boundaries. The slope of the connections along the boundary can therefore be quite steep. This steepness often results in incorrect choices of the connections, in case of a grid cell with more than 2 edge points. This algorithm is therefore even worse than the connection algorithm based on the order of detection.

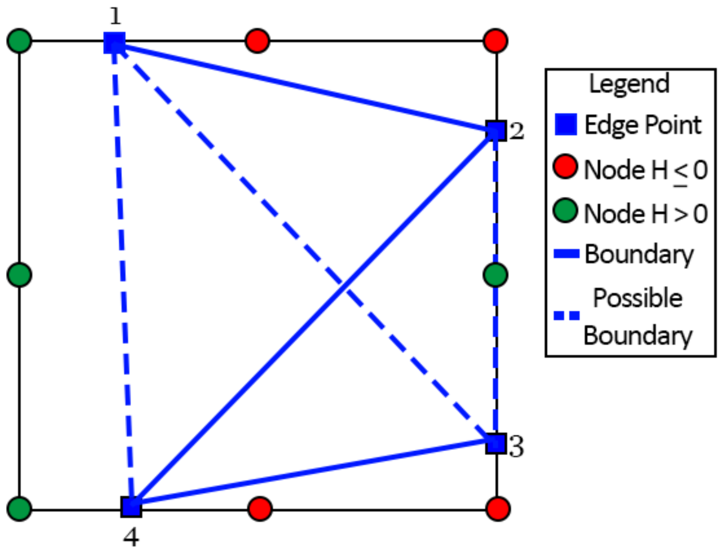
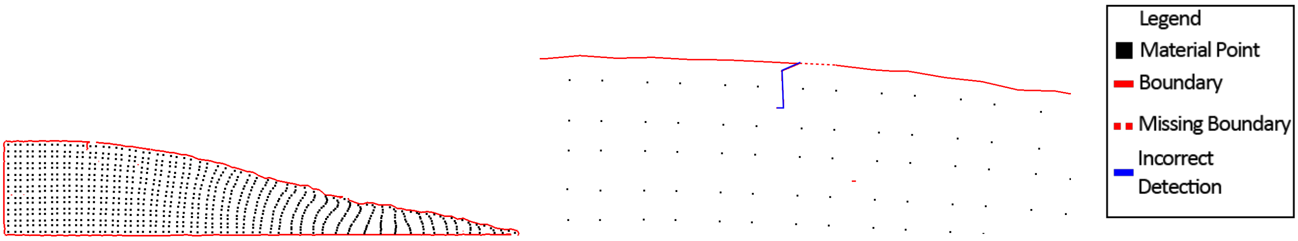


Figure 5.6: Example grid cell for connection of the edge points based on the slope of the proximity field



(a) Incorrect choice in the drawn connections (b) Detail of unconnected surface (c) -

Figure 5.7: Slope stability test for connection of the edge points based on the slope of the proximity field

5.1.3. Triangle Connection Algorithm

Before TCA was developed, a couple of additions were still made to the previous two algorithms. Both the distance between the edge points and the slope of the proximity field normal to the boundary were taken into account. Both of these additions only increased the number of incorrect choices. So in short a completely different method was required to solve the connection problem. TCA has therefore been developed.

TCA splits every cell into eight triangles using a ninth node at the center of each cell (as shown in Figure 5.8a). A connection is still drawn as a straight line between two edge points. However, this straight line represents several edges between the boundaries of the triangles. Four reasonable conclusions have been drawn on the properties of contours to connect the correct edge points namely:

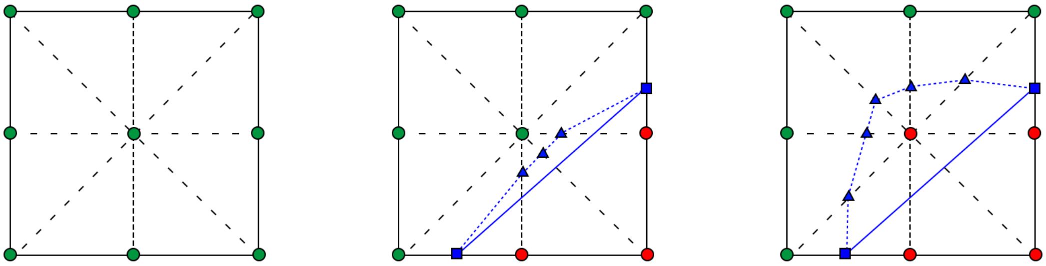
- The number of edge points must be even due to the even number of shifts in the proximity field with respect to the contour level.
- Only $\frac{n}{2}$ edges need to be drawn for n edge points per cell. For example edge points 1-2 and 3-4 would be connected in Figure 5.3. In case that 2-3 must be a connection to form a closed contour this connection can either be made by this cell (therefore connecting 2-3 and 1-4) or by the cell on the right of this figure.
- Edge points can only be connected to the next or previous edge point. So edge 1-3 and 2-4 of Figure 5.3 are not allowed to exist.
- Edge points can only be connected to one edge per cell. Edge points are located on the boundary between two cells and are therefore connected to two edge segments.

Using these conclusions the number of possibilities can be reduced to only two options in case the number of edge points is larger than two and only one option otherwise. Namely edge point i can either be connected to edge point $i-1$ or $i+1$. The decision between these two options is made by the proximity value of the middle node and is based on drawing contours trough triangles. On a triangle the contours are drawn between the two vertices which cross with the contour, or in this case between the two vertices where a shift in the proximity field around the contour level occurs. This can be done until another boundary of the cell is reached. This path is drawn as a dashed line in Figure 5.8, while marking the crossings with the vertices with triangles. A straight connection is drawn between the two points on the boundary of the cell which are connected using this path.

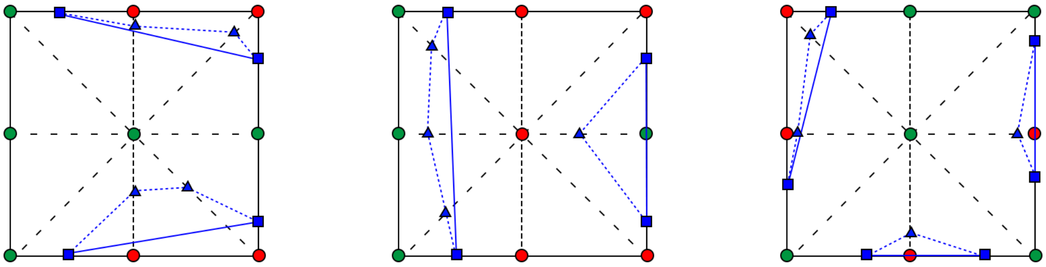
A simplified approach assigns the correct connection with the information of only one triangle. Suppose edge point i is detected between nodes j and $j+1$, for example edge point 1 of Figure 5.8b is located between node 5 and 6. A connection is made according to the following rules:

$$\text{Connection:} \begin{cases} i, i+1, & \text{if } \mathrm{sgn}(H_{j+1}) = \mathrm{sgn}(H_9) \\ i-1, i, & \text{otherwise} \end{cases} \tag{5.3}$$

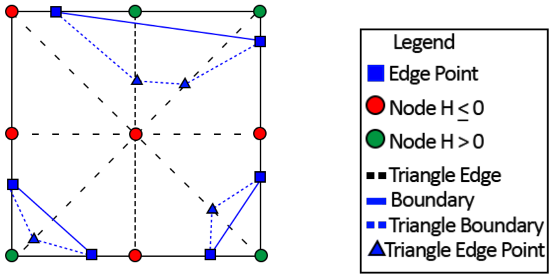
where H_{j+1} and H_9 are the reduced proximity value of node $j+1$ and the middle node respectively and $\mathrm{sgn}(x)$ is the sign function of x. In all the different examples given in Figure 5.8 the contours separate the negative and positive nodes correctly, with the exception of the middle node. It is less important that the middle node is separated correctly, as it is only used to make a decision between the two previously mentioned options. PFM is still reliable if the middle node is separated incorrectly, but breaks down if any of the other nodes are separated incorrectly. Luckily, the latter is not the case.



(a) No edge (b) One edge - Center Positive (c) One edge - Center Negative



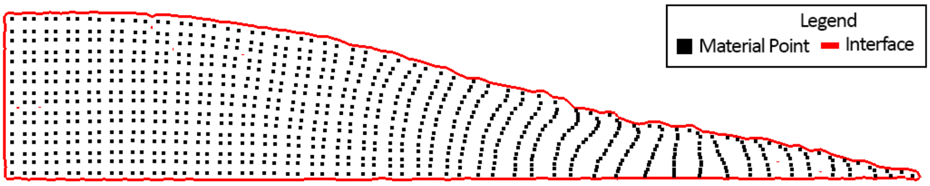
(d) Two edges - Center Positive (e) Two edges - Center Negative (f) Three edges - Center Positive



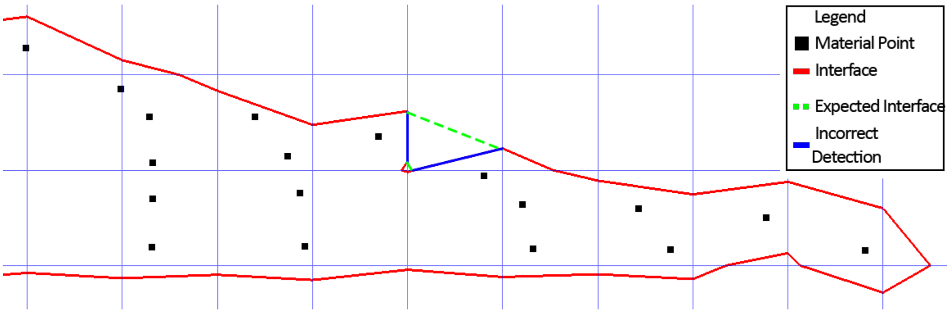
(g) Three edges - Center Negative (h) -

Figure 5.8: Examples of TCA with 1, 2 or 3 edges

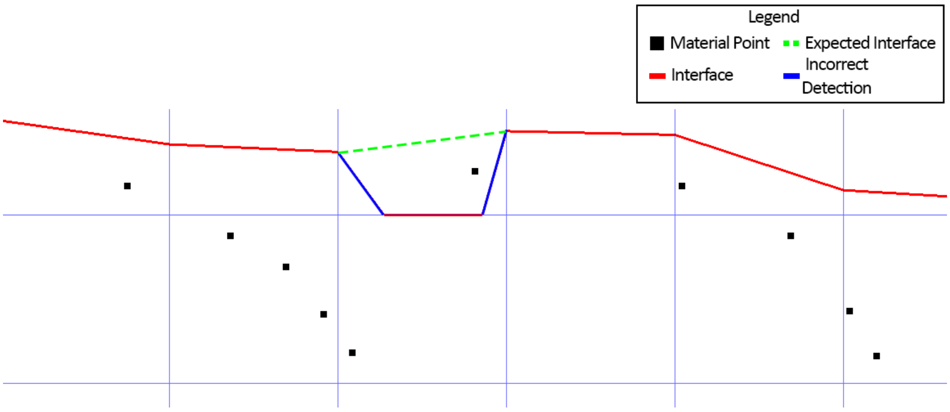
Similar to the previous algorithms, a test has been executed on a slope stability problem, which is shown in Figure 5.9. During the entire failure of the slope the soil body is either completely enclosed by one connected contour or by two completely separated contours, which are both fully connected. The algorithm is therefore stable and will not result in the problems encountered by the previous algorithms, namely gaps in the contour or jointed contours. However, the algorithm is not perfect as seen in Figures 5.9b and 5.9c. In both cases the ninth node falls outside of all the kernel functions and the surface is therefore approximated with strange indents. In Figure 5.9c one of the material points falls outside of the boundary due to this effect. Luckily the algorithm still results in a completely connected boundary and can therefore be considered as reliable. Moreover, both of these issues are the result of the incorrect detection of small holes in the soil body. The occurrence of these holes can be reduced greatly as described in Section 5.3. TCA will provide a stable contour algorithm with the presence of small holes. Moreover, the accuracy of TCA increases without small holes.



(a) Complete connection of the boundary



(b) Unrealistic indent in edge



(c) Material point located outside of boundary

Figure 5.9: Slope stability test for connection of the edge points using TCA

5.2. Reduction of the computation cost

The computation cost of the PFM is relatively high in comparison to the other edge detection methods as well as compared to the regular MPM method. A timestep of the regular explicit MPM method only takes roughly 0.01 - 0.1 seconds per time step, while the PFM method extends this to roughly 2 - 3 seconds. This indicates an increase in computation time of more than 30 times, which is much more than acceptable. However, the efficiency of the method can be improved drastically using the approaches described within this section.

5.2.1. Reusing edge points calculated in previous grid cells

One of the slowest parts of the method is the iteration required for the locating of the edge points, as described in Section 4.4.5. One reduction method therefore reduces the number of times that this method is required. In the original version almost every edge point was localized twice, due to the fact that every edge point lies within two grid cell. Each grid cell calculates the location of every edge point within the cell, doubling the required computation cost. This reduction method skips the calculation of the edge points which are already calculated. However, this reduction technique requires additional memory as it needs to store the grid cell in which the edge points are located.

In the final version of the PFM, the exchange of memory for computation cost was not beneficial, due to the fact that the locating iteration has been optimized drastically. However, this reduction method could be applied in case further computation costs reductions are required. It halves the computation cost of the iteration to find the edge points.

5.2.2. Using shape functions for the locating iteration

A second method to reduce the computation cost of the locating of the edge points uses shape functions to calculate the proximity field values during the iterations. A short recap on this iteration algorithm is given within this section using Figure 5.10. Section 4.4.5 is referenced for additional information on this subject. The edge points are assigned between two cell nodes on which H switches from positive to negative. On this vertex the exact location of the contour level is found by an iterative process. During this process the interval containing the edge point is halved at every iteration step. This interval is shrunk until a user specified error margin is reached.

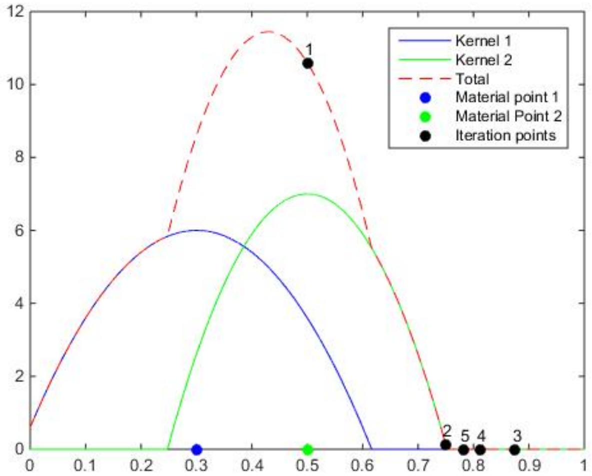


Figure 5.10: Iterative process to find boundary points in PFM.

Unfortunately the iteration process required the proximity field value to find out in which interval the edge point is located. Previously the proximity field value was recalculated using equation (4.10). This resulted in a summation over every material point during each iteration step. This iteration process can take a maximum of 1000 iterations for each edge point, while the complete time step can require the calculation of over 100 edge points. In short, a quite intensive calculation is repeated many times, resulting in the large computation cost of the edge point locating (in the order of 25 times higher then the cost for an explicit MPM timestep).

The material points which influence a certain grid cell can be stored to reduce the recalculation process. During the initial calculation of the proximity field in each time step the material points are stored. The iteration process can then only use the material points which influence the grid cell to calculate the proximity field value of the interval. Using this technique the computation cost is reduced drastically. The calculation cost of the locating is reduced from 25 times the cost of an explicit MPM timestep to only a quarter of the same MPM computation cost.

However, once again the solution described above exchanges computation cost for storage capacity as the method requires the storage of the influence material points for each grid cell of the proximity field. The current program therefore uses shape functions to calculate the proximity field value during the iteration process. From the nodal proximity value the proximity field value can be calculated using shape functions. This requires less computations compared to the technique, which only took the relevant material points into account. The computation cost is reduced to only a tenth of the requirements for an MPM timestep. Furthermore, using shape functions does not require additional storage for the relevant material points for each grid cell. Finally this technique is inherent to the MPM method and the methods used for the MPM method can be reused using this technique and the method is therefore uncomplicated.

A problem with this technique is the shape of the proximity field, shown in Figure 5.10, which does not match the quadratic shape of a eight node grid shape function. The quadratic shape functions approximate the shape of the proximity field based on the proximity at -1, 0 and 1 in Figure 5.11. The six other nodes of the grid do not influence this shape, due to the fact that the calculation is performed on a grid vertex. The shape functions cannot represent the proximity field as is visualized in both Figure 5.11 and Figure 5.12. The boundary is approximated much further outwards in both cases.

One of the largest influences on the accuracy of the representation using shape functions is the shape outside of the kernel functions. Due to the fact that this is kept at zero the shape is far from polynomial even if only one material point influences the edge. It is therefore advantageous to allow the influence of a material point on a grid node to be negative. However, this negative influence is only allowed when all material points influence the grid node negatively. In case all material points influence a grid node negatively only the influence of the closest material point is taken into account. The removal of this bottom boundary of the proximity field and therefore allowing the proximity field to drop below zero, removed the error in the approximation as shown in Figure 5.12.

Even though the shape functions theoretically should not be able to represent the shape of the proximity field, in practice the representation achieves a similar accuracy as a full recalculation of the proximity field. However, the accuracy of this technique has not been proven and misplacement of the soil surface might occur due to the shape functions under curtain conditions. No misplacements due to the shape functions have been observed during the testing of the PFM discussed later in this section.

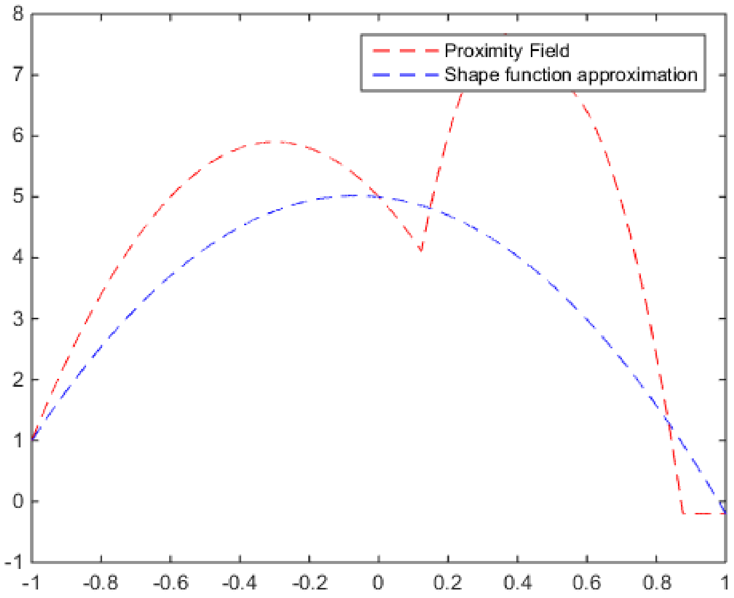


Figure 5.11: Proximity field approximation using shape functions

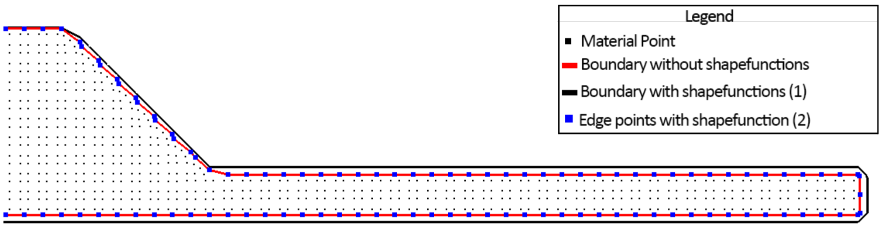


Figure 5.12: Slope profile using shape functions vs regular recalculation of the proximity field inside of grid cells. (1) This shape function boundary used an unchanged proximity field. (2) This shape function boundary used a proximity field with negative values.

5.2.3. Reducing the computation cost of the calculation of the proximity field

Even with the usage of the shape functions described in the previous paragraph the total calculation cost of PFM is still relatively high. The compuation cost is still roughly 4.5 times higher than the cost for an explicit MPM time step. However, the iteration algorithm no longer provides the largest contribution to the computation cost. The current computation cost is mainly the result of the calculation of the proximity field on the grid at the start of the time step. Therefore, a reduction of the cost on this calculation is required.

Currently equation (4.10) is performed for each grid cell, which requires the transformation to the local coordinate systems of all the material points according to equation (4.11). The inverse of the shape matrix $[\mathbf{S_j}]$ is calculated again for each grid cell or stored for all the material points. However, equation (4.10) can also be calculated for each material point, while the contribution from this material point to each grid cell are added to the total proximity field. This removes the requirements to calculate the inverse of $[\mathbf{S_j}]$ multiple times per material point and allows for a smarter selection of the grid cells which need to be included in the calculation.

The grid cell in which the material point is located is found, after which all the neighbouring grid cells are selected which could possibly be influenced by the material point. A larger kernel function results in a larger selection of neighbouring cells. The proximity field influence of the material points is then calculated for only these cells, which reduces the number of calculations in case of large problems.

This final addition to PFM reduced its total computation cost to being roughly equal to a normal timestep in an explicit MPM calculation. In the slope example visualized in Figure 5.12, PFM took 0.047 seconds, while the normal MPM timestep took 0.048 seconds. The inclusion of PFM therefore roughly doubles the computation cost of an explicit MPM timestep. The computation cost does not increase for longer time steps and PFM is therefore only a small addition to the larger computation cost of an implicit MPM.

5.3. Removal of small holes and location of the detected boundary

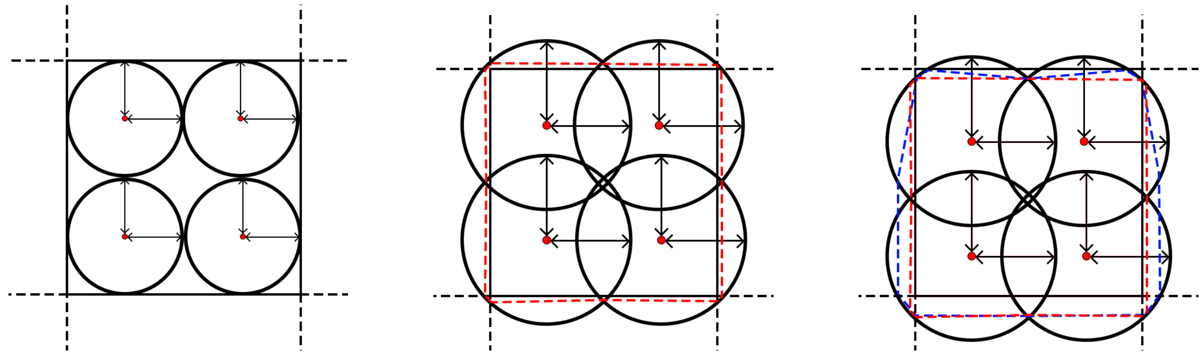
Due to the shape of the kernel functions, the proximity field cannot be created without either gaps or overlaps between the different kernel functions. Filling a space has been proven to be difficult in the initial condition as shown in Figure 5.13 and explained in the continuation of this paragraph. The problems regarding the location of the detected boundary and the appearance of small holes are both the results of the difficulties of filling a shape with a circular kernel function. Within this paragraph these problems are further analysed and a couple of solutions are proposed.

The initial size of the kernel functions in Figures 5.13 is defined according to equation (4.21). f_x and f_y must at least be larger than $\sqrt{2}$, due to the fact that only at this size the eight nodes of the grid are influenced by the kernels. For example, the kernel size used in Figure 5.13a is too small for the detection of a surface as all the nodes of the square fall outside of the kernel function boundary. Increasing the size of the kernels does allow for the detection of a roughly square boundary as shown in Figure 5.13b. The edge is detected slightly to far outwards and is detected by the neighbouring grid cells, but this can be fixed by using a higher contour level, which brings the edge inwards.

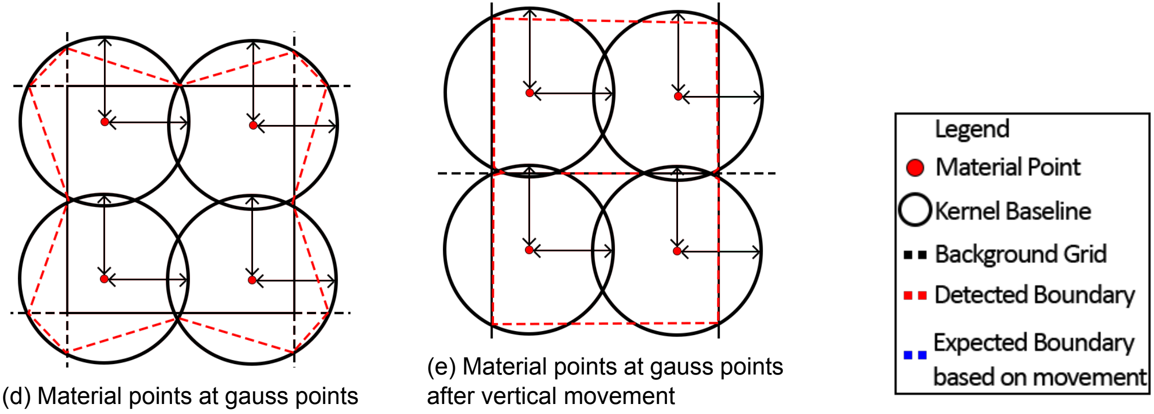
Unfortunately the correct detection of Figure 5.13b is influenced greatly by the location of the material points. After some small vertical movement the boundary is detected once more in Figure 5.13c. Unfortunately the boundary is distorted from a square into a unrecognisable shape. Hopefully these mistakes as well as the potential occurrences of holes in the proximity field can be fixed or reduced by the methods described in this section.

Finally, an additional problem arises from the placement of the material points in the current state of the MPM model. Often the material points are placed on the gauss points of the background cell. This results in an even worse approximation of the boundary, as shown in Figure 5.13d. Using this placement a hole is present at the center of each cell, while the surface is no longer straight in the initial conditions. The surface does become flatter once multiple cells are used, which is the reason why the surface detection was decent according to Chapter 4. The hole at the center is not a problem in the initial condition as it is not detected by the grid, however onces the hole meets a grid vertex the square is no longer correct as a gap is detected in the middle. This gaps is displayed as a straight line due to the fact that it occurs around one of the vertexes instead of a grid corner. At the corner this hole would be detected as an actual hole.

In the continuation of this section an attempt has been made to address these problems based on the



(a) $f_x = f_y = 1$ (b) $f_x = f_y = 1.5$ (c) Small vertical movement



(d) Material points at gauss points (e) Material points at gauss points after vertical movement

Figure 5.13: Attempts to fill a square using ellipsoidal kernels around material points

example slope shown in Figure 5.14. In this figure the kernel functions are shown as the background, indicating the gaps present within the proximity field. The surface is much smoother than seen in Figure 5.13d due to the influence of multiple grid cells. However, as indicated in Figure 5.15 holes do still occur and the focus of this section is therefore on the removal of these holes. Finally, a method is required to reduce the unnecessary roughness of the surface after deformation shown in Figure 5.16. This roughness occurs due to the shape of the kernel function and the size of the background grid elements. The shape of the surface depends on the crossing of a kernel function with the grid and the shape of the surface is therefore unstable without smoothing.

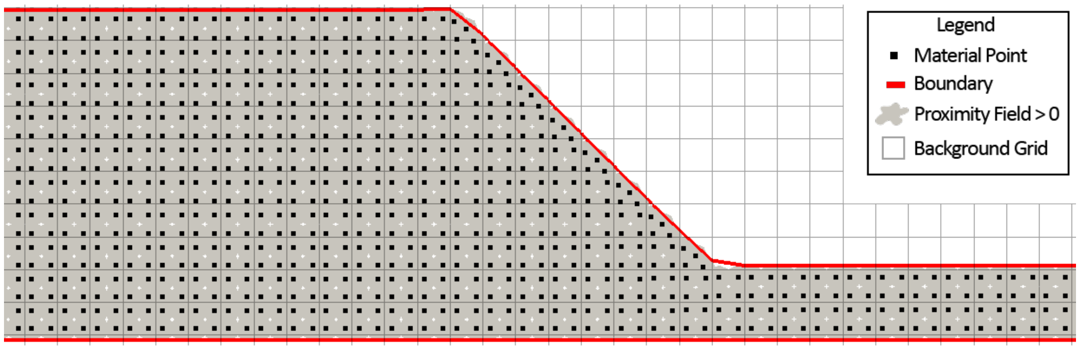


Figure 5.14: Initial condition of a slope stability problem used to address the problems regarding proximity field holes and boundary roughness. A visualisation of the proximity field larger > 0 is included.

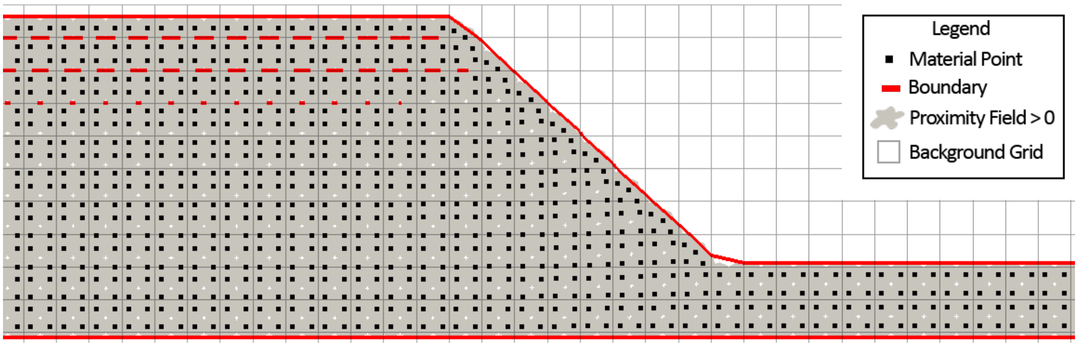


Figure 5.15: Small deformation of a slope stability problem used to address the problems regarding proximity field holes and boundary roughness. A visualisation of the proximity field larger > 0 is included.

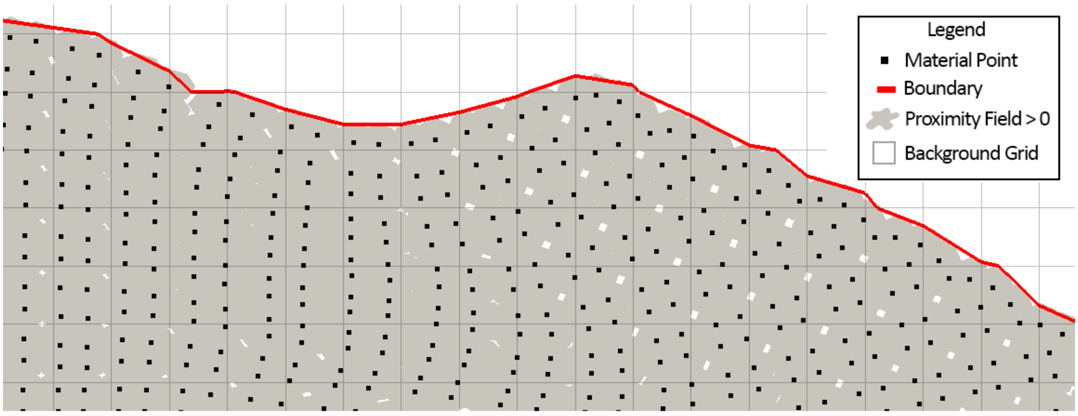


Figure 5.16: Large deformation of a slope stability problem used to address the problems regarding proximity field holes and boundary roughness. A visualisation of the proximity field larger > 0 is included.

5.3.1. Smoothing the proximity field using a Gaussian filter

To both remove small holes at the center of the soil body as well as smooth the soil surface a Gaussian filter can be applied to the proximity field grid (Delmas [13]). A Gaussian filter is a weighted averaging filter, where the weights are determined based on a normal distribution. The filter is applied as a 2D convolution function to the proximity field grid, in which each nodes is treated separately. The iteration algorithm of the edge points must therefore depend completely on the proximity field values at the nodes and this filter can therefore only be applied if shape functions are used to localize the edge points. The filter is constructed using the weights of a normal function:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{5.4}$$

where $G(x,y)$ is the normal distribution at x and y with the average at $x = 0$ and $y = 0$. $x = 0$ and $y = 0$ is the center of the filter and σ is the standard deviation of the normal distribution. A larger σ results in a wider Gaussian filter and therefore more averaging of the proximity field. This two-dimensional filter can also be applied as two one dimensional filters with the same kernel function:

$$\begin{aligned} G(x) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x^2}{2\sigma^2}) \\ G(y) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{y^2}{2\sigma^2}) \end{aligned} \tag{5.5}$$

Separation of the directions saves computation cost as the amount of computation for a two-dimensional kernel filter including n nodes in each direction is in the order of $O(n^2)$, while two one-dimensional kernel filters are in the order of $O(n+n)$. The exact same kernel filter can be used for each direction, so no additional computation cost or storage is required on this front either. Due to these advantages, two one-dimensional kernel filters are used.

Equation (5.5) is a continuous function, while the proximity field is only defined on nodes. Equation (5.5) therefore needs to be transformed to an discontinuous grid, where each grid cell represents the interval defined by:

$$k-\frac{1}{2}\leftrightarrow k+\frac{1}{2},\quad \text{where } k=-m,-m+1...-1,0,1...m-1,m \tag{5.6}$$

with $m=\frac{n-1}{2}$ and due to the fact that n is defined to be odd, m must be an integer. This can be either be achieved by selecting the values of the normal functions at the center of each filter cell or the normal distribution could be integrated over the filter cell for a better approximation. However, first of all one must be certain that the discontinuous Gaussian filter still represents the infinitely long continuous Gaussian distribution. Luckily 99.7% of the area under the Gaussian distribution lies within the interval $[-3\sigma\ \ 3\sigma]$. The size of the kernel function must therefore follow from the following equation:

$$m+\frac{1}{2}=3\sigma\rightarrow m=\lceil 3\sigma-\frac{1}{2}\rceil\rightarrow n=2(\lceil 3\sigma-\frac{1}{2}\rceil)+1 \tag{5.7}$$

where $\lceil x\rceil$ depicts the ceiling function of x and is required to maintain m as an integer. In the current state of the program, n can either be specified by the user, possibly resulting in a incomplete representation of the Gaussian distribution, or n can be calculated based on equation (5.7). After the size of the kernel convolution has been defined the individual weights are calculated based on an integration over the interval, as it is more precise as using the value of the Gaussian distribution at the center of the kernel cells. For each cell the following integral defines the weight of the cell:

$$G(k)=\int_{k-\frac{1}{2}}^{k+\frac{1}{2}}\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{x^2}{2\sigma^2})dx \tag{5.8}$$

This results in the following kernel convolution vector for $\sigma=1$:

$$\begin{bmatrix}0.00598 & 0.0606 & 0.242 & 0.383 & 0.242 & 0.0606 & 0.00598\end{bmatrix} \tag{5.9}$$

A convolution operation is applied to the nodes of the proximity field using this vector in both the horizontal and vertical direction.

Gaussian filters using different standard deviations have been applied to the slope stability problem of Figure 5.14. As visualized in Figures 5.17 - 5.19 holes are no longer present in the detected surface using these Gaussian filters. However, the assumed boundary based on these filters is far from perfect. Sharp corners are 'rounded' as shown in Figures 5.18 and 5.19. Moreover, the surface is detected further into the soil body, often even below the material points. These problems might be solved using a lower contour value, but this could reduces the effectiveness of the filter with respect to the removal of holes. A different method is therefore required to remove small holes from the soil body.

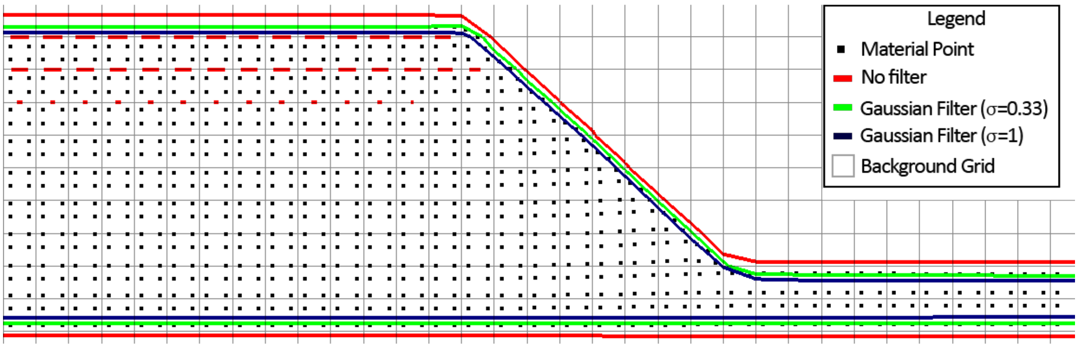


Figure 5.17: Edge detection of time step 1 of a slope stability problem using different Gaussian filters ($\sigma=1$ and $\sigma=1/3$) - Detection of the slope

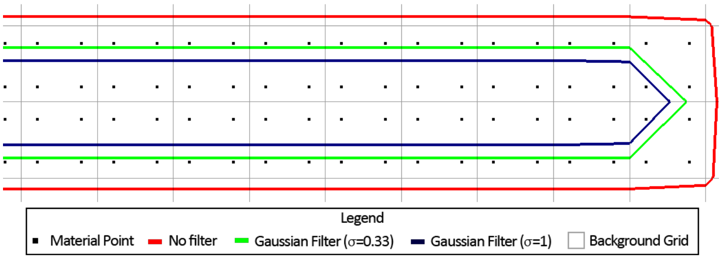


Figure 5.18: Edge detection of time step 1 of a slope stability problem using different Gaussian filters ($\sigma = 1$ and $\sigma = 1/3$) - Detection of the right boundary

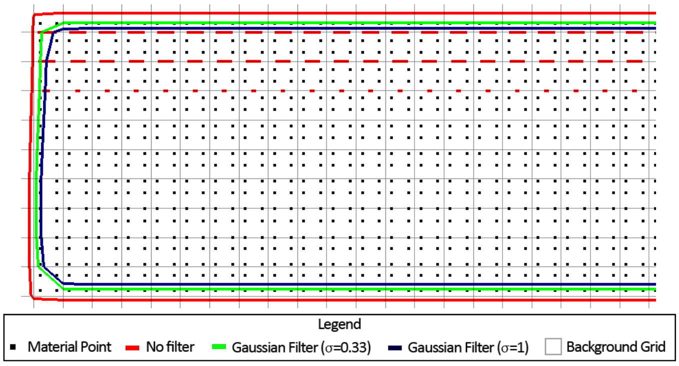


Figure 5.19: Edge detection of time step 1 of a slope stability problem using different Gaussian filters ($\sigma = 1$ and $\sigma = 1/3$) - Detection of the left boundary

Even though the smoothing creates problems at the corners of the solution, the surface of the deformed soil slope is smoother as shown in Figures 5.20 and 5.21. While some of the detail is therefore lost, specifically sharp corners are rounded, the surface might be more appropriate for the application of boundary conditions. Applying boundary conditions to rough surfaces is difficult due to the rapidly changing conditions and the smoothing of the surface can therefore be considered useful. However, the surface of a filtered proximity field will need to be moved outwards for a correct approximation. Moving the filtered surface outwards can be achieved by choosing a lower contour value as done in Figure 5.22. The surface is smoother and at the same location as the unfiltered boundary. The exceptions are at the sharp corners at the right and left boundary condition. The filtered proximity field can be used to apply boundary conditions as the smoother surface will result in a smoother boundary condition. However, at sharp corners the boundary condition will be distorted. Application of boundary conditions to corners is not possible using a Gaussian filter.

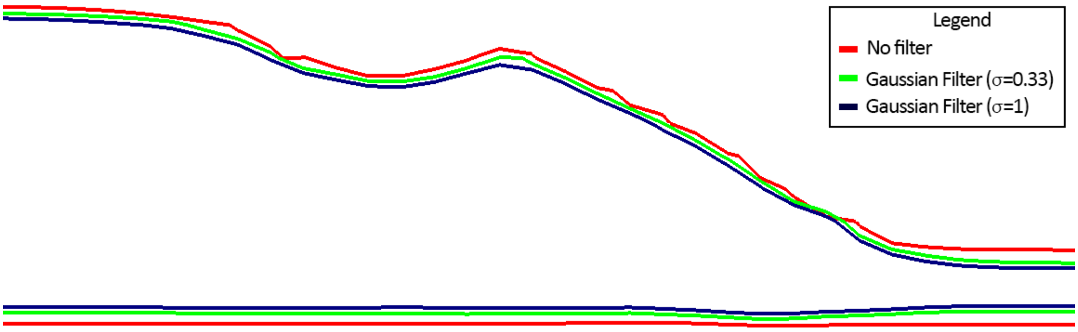


Figure 5.20: Edge detection after deformation of the slope using different Gaussian filters - Time step 24

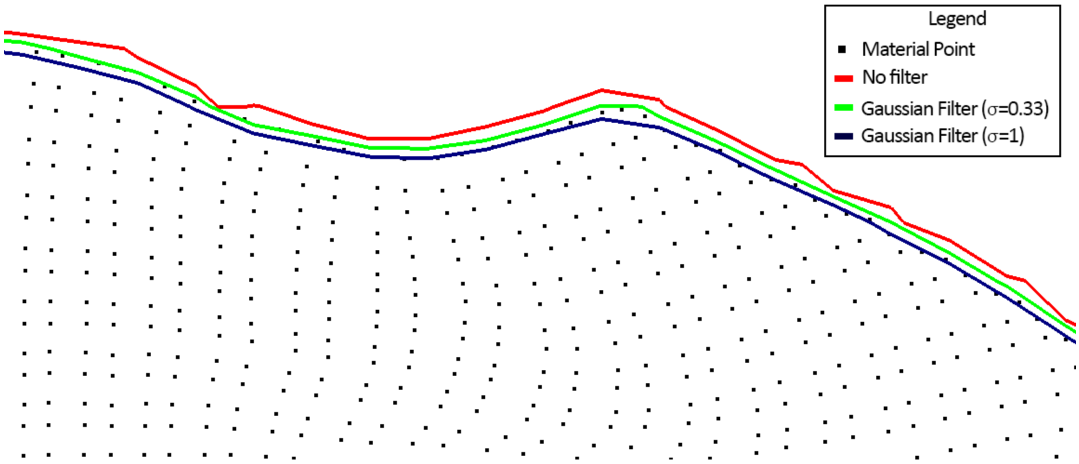
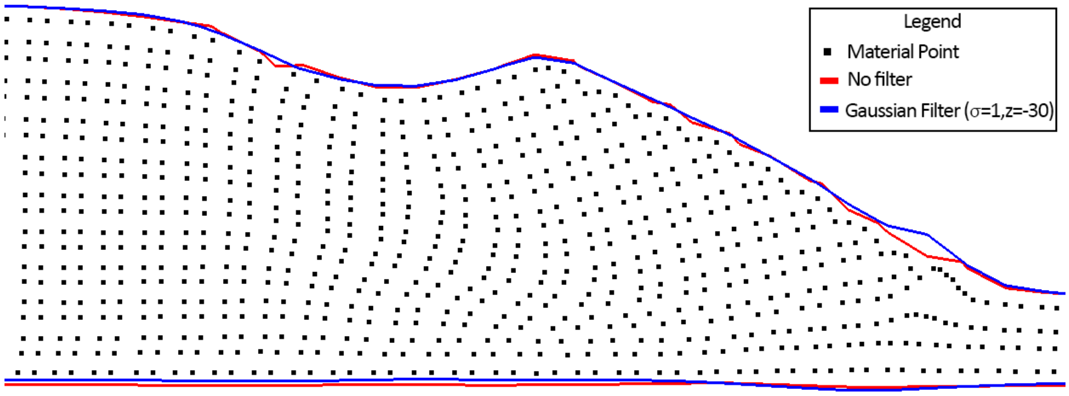
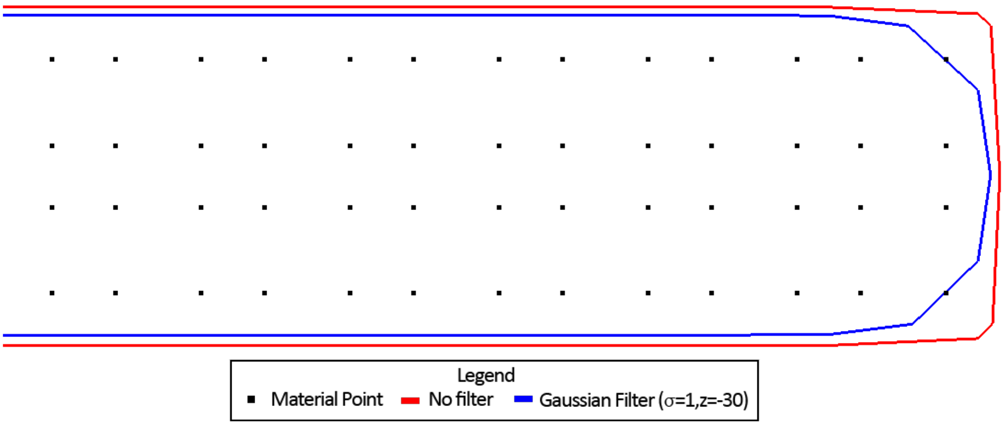


Figure 5.21: Edge detection after deformation of the slope using different Gaussian filters - Detail of time step 24



(a) Overview of the slope in time step 24



(b) Detection at the right boundary

Figure 5.22: Edge detection using an unfiltered proximity field vs a filtered proximity field (Gaussian Filter with $\sigma = 1$) with a lower contour value ($z = -30$)

5.3.2. Smoothing the proximity field using a Median Filter

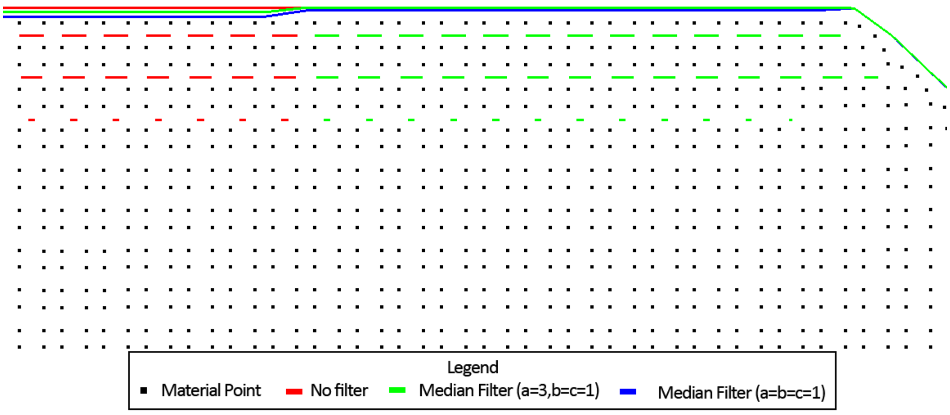
A second attempt to remove holes and smooth the surface was the application of a median filter. For each node the eight surrounding nodes were taken into account and the median of the proximity values of the nine nodes is used as the new proximity value of this node. Similar to the Gaussian Filter, this filter can only be applied when the shape functions are used to localize the edge points. Moreover, weighting can be used to increase the likeliness of for example the middle node to be picked as the median value. To use this weighting some of the nine nodes will be used multiple times according to the following weight matrix:

$$\text{Weight matrix} = \begin{bmatrix} c & b & c \\ b & a & b \\ c & b & c \end{bmatrix} \tag{5.10}$$

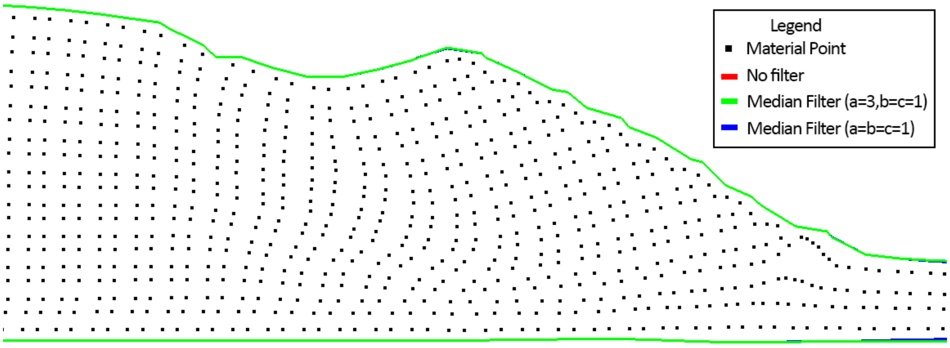
where a, b, c are user specified and represent the number of times the specific node is taken into account. a is applied to the central node, b to the neighbours and c to the nodes at the corners of the median filter. This filter is then applied to each node individually. At the boundaries of the proximity field grid the proximity of the missing nodes are filled in by the proximity of the central node.

Different median filters have been used but all of them experience two problems which have been visualized in Figure 5.23. While the median filters do remove some or all of the holes detected without a filter the surface is distorted unevenly by the filter. While a Gaussian filter detects the surface at a roughly constant distance from the unfiltered surface within one timestep, a median filter can suddenly lower the surface at certain location and maintaining a correct surface detection at other locations. This effect is reduced when a is increased with respect to b and c. However, once a is increased with respect to b and c holes occur more often.

Finally, a hypothetical solution to these uneven distortions due to a median filter has not been found and this filtering method is therefore no longer used.



(a) Incorrect detection of the straight top surface



(b) Overview of the slope in time step 24

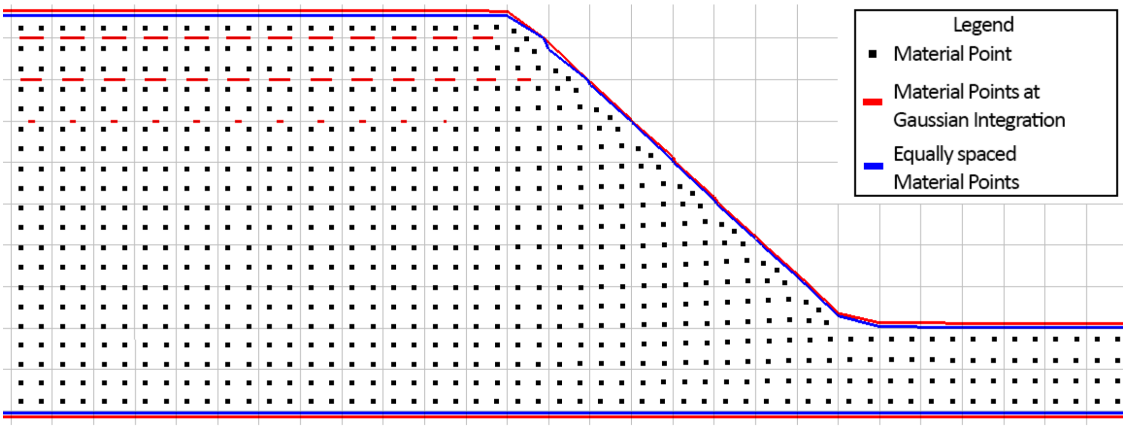
Figure 5.23: Edge detection of a slope using Median filters

5.3.3. Equally space material points in grid cells

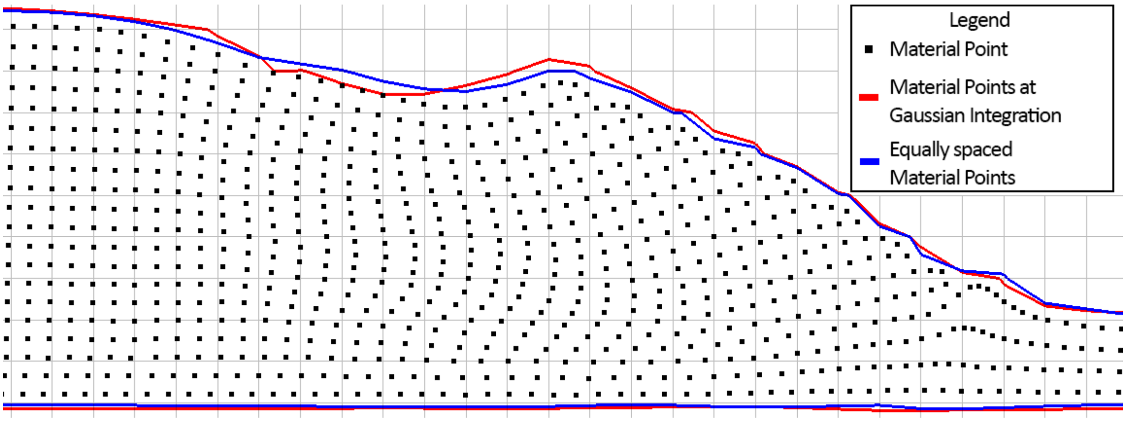
Besides filtering of the proximity field, a completely different solution may change the position of the material points within the grid cells. As seen in Figures 5.13b and 5.13d, a much better filling can be achieved if the material points are equally spaced over the soil body. The material points are currently placed at the Gaussian integration points of the background grid. They are initialized at these locations, because the FEM method used these locations as their integration points. However, due to the movement through the background grid in MPM, the material points do not coincide with the integration points after one time step and are therefore not required to be at the integration points during initialization as well. A clear explanation of the advantages of the placement at the Gaussian points locations is not present. Distributing the material points differently should therefore not be a problem.

A better distribution of material can be achieved by equally spacing the material points, which results in a removal of any holes as visualized in Figure 5.24a. At the same time a slightly smaller soil body is detected using equally space material points, due to the fact that the outer material points are moved slightly inwards (as explained in Figures 5.13b and 5.13d). This surface location appears to be a better representation, due to the fact that the bottom, left and right surfaces are closer to the specified boundary conditions.

However, the deformation process has been observed to be different, as shown in Figure 5.24b. It is currently unknown whether the material points can be equally spaced and the differences in the deformation process might therefore be incorrect. Future research on this topic is required to observe the effects of the initialized location of the material points. As a final note for the detection of edges, equally spaced material points present a better approximation of the boundary and are therefore preferred.



(a) Removal of small holes due to the equally spaced material points



(b) Not matching soil surfaces due to different locations of the material points

Figure 5.24: Comparison between equally spaced material points and material points placed at the Gaussian integration points

5.3.4. Removal of small holes along vertices

The solution presented within this paragraph removes small holes manually during the detection process and does not change the actual proximity field. The solution is therefore mathematically incorrect, whereas the previous solutions are mathematically correct as it actually changes the proximity field. However, the two filtering methods have been unable to completely remove the holes from the soil body without allowing for incorrect detection at other locations. Equally spacing material points in the initial condition did solve all the problems regarding small holes and can be considered mathematically correct. However, it is uncertain whether or not this spacing results in the correct deformation profile. Only equally spacing material points removes all the small holes from the proximity field, but this material point placement might be incorrect. Therefore, the removal of small holes with an algorithm is necessary.

This algorithm only influences the small holes inside of the soil as it removes them from the detection algorithm. Currently the rule to remove small holes is only applied to the straight holes present on a vertex of a grid cell, as seen in Figure 5.15. This rule specifically removes any edges which only connect two edge points on a grid cell vertex. As long as other edges are connected to the edge points on the vertex the edge is allowed to exist. Otherwise the straight lines within the deformed surface of Figure 5.16 would be removed as well. Therefore, the algorithm only removes unrealistic holes. The method could be expended to also remove other small holes based on the size of the holes. However, additional research is required to separated the holes with a realistic size from the unrealistically small holes. No new figures are given as examples for this solution as the solution only removes small edges. Figure 5.15 basically turns into Figure 5.14, where only the outside of the soil body is considered as an boundary.

5.4. Deformation of the kernel functions

As mentioned in the literature study, MPM modelling is especially useful in large strain deformation problems. While the slope stability problem tested above has considerable strains, strains can be bigger in large strain deformation problems. During the application of a force boundary condition, explained in further detail in the next chapter, problems occurred under large strain conditions. In this example a force is applied on top of the center third of the surface, as shown in Figure 5.25.

At first, PFM was used without allowing the kernel function to deform as shown in Figure 5.26. While the solution is physically incorrect as the material points are assumed to represent the same volume of material during the entire solution, a surface can still be detected. Unfortunately, due to the fixed shape of the kernel functions a large hole can be observed in soil body, which is not present in reality.

To hopefully achieve a physically better solution the kernel functions are allowed to deform according to the small strain description given in equation (4.17). As visualized in Figure 5.27 allowing the kernel functions to deform under these large strain conditions results in extremely thin but long ellipsoidal kernel functions. This is a result of the assumption that the strain defined at the material point is representative for the entire kernel function. This assumption is not correct especially in large strain conditions. In Figure 5.27 rotation has not been applied to the kernel functions for the clarity of the figure. However, even with rotation, the kernel functions are deformed to unrealistic sizes. According to the kernel functions of Figure 5.27 the material points influence material located at physically unrealistic distances. In some cases the kernel function even describes an influence of the material point to points far beyond the boundary of the domain. These large kernel functions, which stretch beyond the boundary of the domain eventually break the edge detection algorithm and an edge is therefore not drawn in Figure 5.27.

A rotation is also applied in Figure 5.28. The edge detection algorithm did not break completely in this case, but the problems with the large deformation are clearly visible. Many small holes occur in the proximity field due to the extreme deformation of the kernel functions. Whenever a node of the proximity field grid is located in one of these holes the edge detection algorithm assumes a hole to be located at this location. The abundance of these small holes leads to many nodes being located inside one of them. A connecting surface is drawn according to the connection algorithm described previously, resulting in unrealistic hole formations in the detected surface. Moreover, the kernel functions cross the boundary of the domain. In this case the bottom boundary has already been crossed by two kernel functions, but none of the nodes of the proximity field grid were located in these kernels. Therefore, the connection algorithm ignored the kernels and did not break. If one of the nodes would have been located in these kernels the algorithm would have been broken.

Perhaps, the applied force is too large even for an MPM model to handle. Therefore, the resulting strains might be unrealistic. However, the MPM still works using these strains as long as the surface is detected, as shown in Figure 5.26. Therefore, a solution must be found to reduce the large deformation of the kernel functions. To prevent the model from breaking under large strain conditions, a maximum and minimum size of the kernel functions can be specified by the user in both directions. The maximum size should represent the maximum size of the kernel at which the strain of the material point can still roughly represent the complete kernel function. The minimum size prevents the kernel function from becoming too small to be detected by the proximity field grid. A smaller proximity field grid size allows the minimum kernel size to be smaller as well. The amount of strain applied to the kernel function is linearly decreased the closer the size of the kernel is to the maximum or minimum size. The strain is given by:

$$\epsilon_x = \begin{cases} \epsilon_{x,m} * \min(\max(\frac{s_{x,max}-s_x}{s_{x,max}-s_{x,0}}, 0)1), & \text{if } \epsilon_m > 0 \\ \epsilon_{x,m} * \min(\max(\frac{s_x-s_{x,min}}{s_{x,0}-s_{x,min}}, 0)1), & \text{otherwise} \end{cases} \tag{5.11}$$

where ϵ_x is the strain applied to the kernel, $\epsilon_{x,m}$ is the strain of the material point, s_x is the size of the kernel axis originally placed along the x axis. $s_{x,max}$ and $s_{x,min}$ are the maximum and minimum kernel size specified by the user and $s_{x,0}$ is the initial size of the kernel. The same equation can be used for the strain in the y-direction. The results of this approach have been visualised in Figure 5.29. The approach reduces the number of holes in comparison to allowing no deformation, while the kernel functions remain reasonable approximations for the influence of the material points.

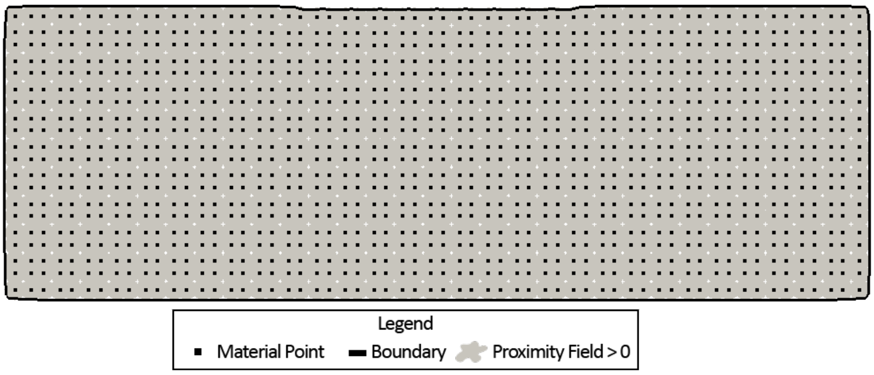


Figure 5.25: The initial condition of a large strain deformation problem. A force of 50 kPa is applied to the center third of the top surface.

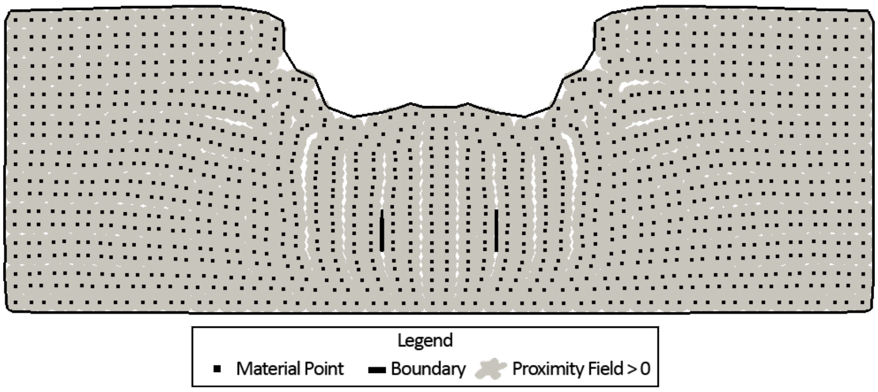


Figure 5.26: Deformation under a force of 50 kPa applied to the center third of the top surface without deformation to the kernel functions.

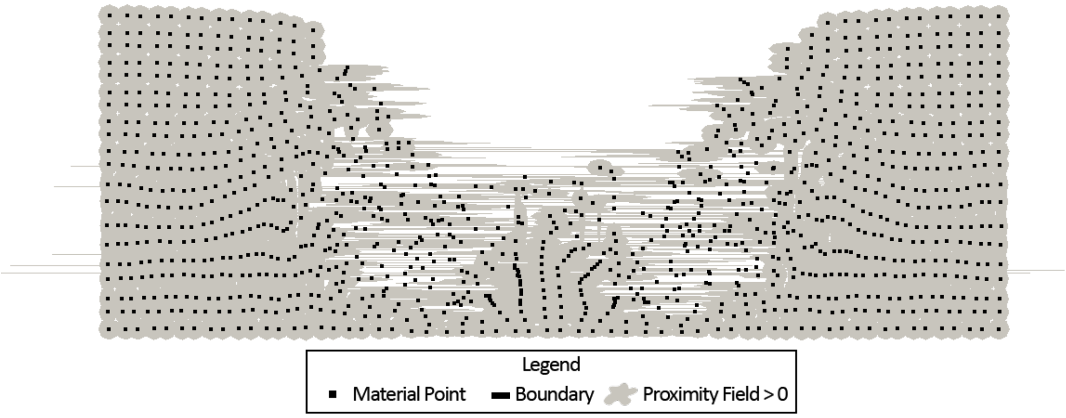


Figure 5.27: Deformation under a force of 50 kPa applied to the center third of the top surface with deformation and without rotation to the kernel functions.

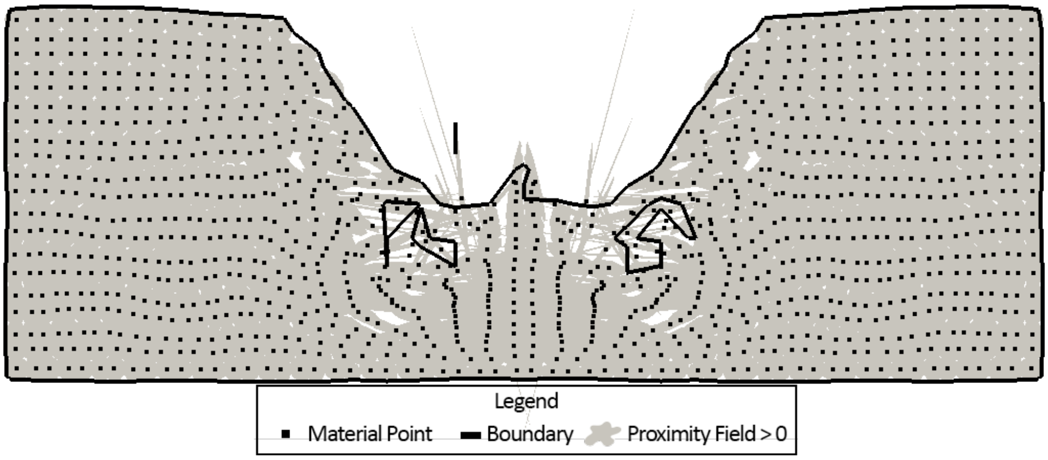


Figure 5.28: Deformation under a force of 50 kPa applied to the center third of the top surface with deformation and rotation to the kernel functions.

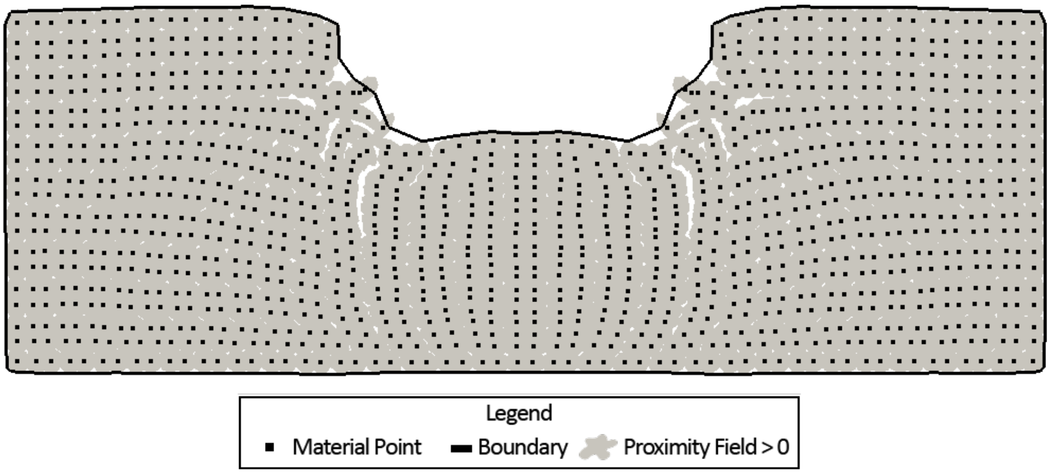


Figure 5.29: Deformation under a force of 50 kPa applied to the center third of the top surface with a maximum deformation to the kernel functions with $s_{max} = 2s_0$ and $s_{min} = \frac{2/3}{s}{}_0$

5.5. Comparison to FEM

Finally, the slope stability problem tested within this section is compared to FEM. The FEM calculation has been performed with an updated Lagrangian version of Plaxis. Unfortunately the constitutive model, softening after the peak strength is reached, used in the MPM test was not available in Plaxis. In an attempt to counter this difference the strength (cohesion) of the material used in Plaxis is reduced. The comparison has been made with Plaxis, due to its frequent use in practice. Besides the comparison with Plaxis an FEM has been build from the MPM program used in this thesis. The Eulerian part of the MPM program has been removed, which leaves a FEM with the same constitutive model as the MPM program. In this section these two comparisons have been analysed.

5.5.1. Comparison to Plaxis

Due to the different constitutive model the strength of the Plaxis FEM has been halved (to 10 kPa) in comparison to the initial strength of the MPM program. This strength reduction is similar to the reduction in strength due to softening. However, this reduction has been applied to the entire slope instead of the deformation zone. Moreover, the initial stress conditions in the MPM test are zero as the computation starts without any pre-loading. The updated Lagrangian version of Plaxis is only capable to work with pre-loading resulting in smaller deformations. However, even with these differences a comparison could be made between the two models.

The differences between FEM and MPM in the initial condition are shown in Figure 5.30. The top and bottom boundary of both models were a relatively close fit. The close fit is the result of equally spacing the material points as well as smoothing of the proximity field. While the representation with PFM is decent, the difference between the two was still roughly 0.05, which is quite a lot considering that it was the initial condition. Further matching of the PFM boundary to the actual boundary location is therefore necessary. The difference at the slope was even larger, roughly 0.1 m. Moreover, the sharp corners at the crest and toe of the slope cannot be represented with PFM, due to smoothing of the surface. However, it is unclear, whether the sharp corners are a better representation of reality. The differences between MPM and FEM reduce with a smaller cell size as the error in the approximation of the surface with PFM depends on the cell size. A smaller cell size, results in a more even spread of the material points. Due to this more even spread, the boundary can be placed closer to the material points without the occurrence of holes inside of the material or at the surface.

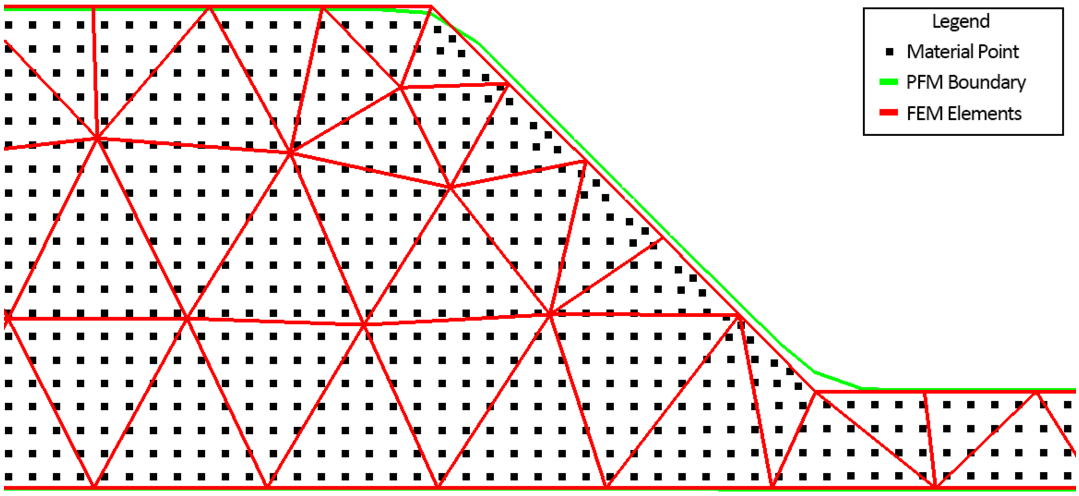


Figure 5.30: Comparison between FEM and PFM boundary at modelling a slope stability problem - Initial condition

The deformation according to the two methods have been compared as well in Figures 5.31 and 5.32. In Figure 5.31 the final deformation step of FEM has been compared with timestep 15 of MPM as the slope angle is similar in both cases. A larger vertical deformation of the crest is present in MPM due to the incorrect initial stress deformation. However, besides this larger deformation the crest profile is similar. A drop of the soil behind the crest is present in both MPM and FEM. Additionally, the crest of the slope is raised slightly in comparison with this drop, due to the rotational failure mechanism. A larger difference is present at the toe of the slope. The toe is raised due to the circular failure mechanism in both models, but this is more clearly present in the FEM model. The increase in toe height is partly due to the limited number of background elements. Which model gives the best approximation is currently unclear, but the large mesh distortion of FEM indicates that FEM modelling is not suitable for this type of deformation.

The FEM model failed to model the deformation after the time step shown in Figure 5.31. However, the deformation process could still be modelled with MPM as shown in Figure 5.32. A new gentler slope angle is created due to the failure process. The height increase of the toe is almost unobservable in this new slope as it is part of the new slope. However, the soil at the toe has been compressed in MPM, which resulted in the mesh distortion in FEM. Mesh refinement at this location would therefore be necessary in FEM, but this is not the case in MPM.

One clear problem with MPM is located at the bottom boundary below the slope. Due to Gaussian Smoothing the surface is lifted at this location. A solution to this problem currently has not been found and this problem must therefore be solved in future research.

In short, the PFM surface approximates the initial condition decently, but improvements to the initial phase can definitely be made. The deformation process is similar in both models with the largest difference occurring at the toe of the slope. Finally, the MPM model is capable of handling much larger deformation without failure of the model.

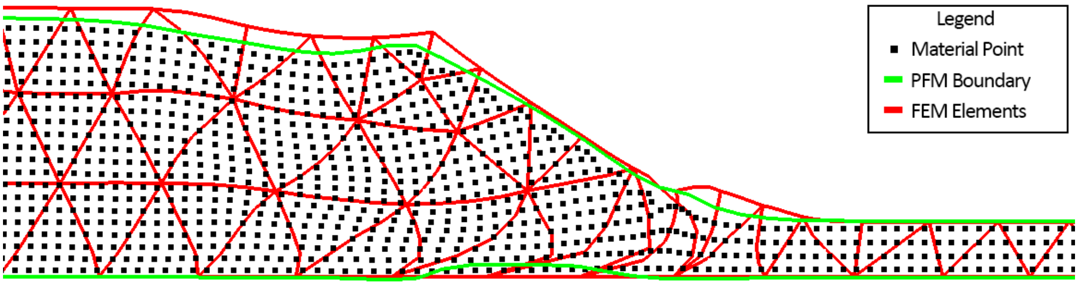


Figure 5.31: Comparison between FEM and PFM boundary at modelling a slope stability problem - Final deformation FEM, timestep 15 MPM

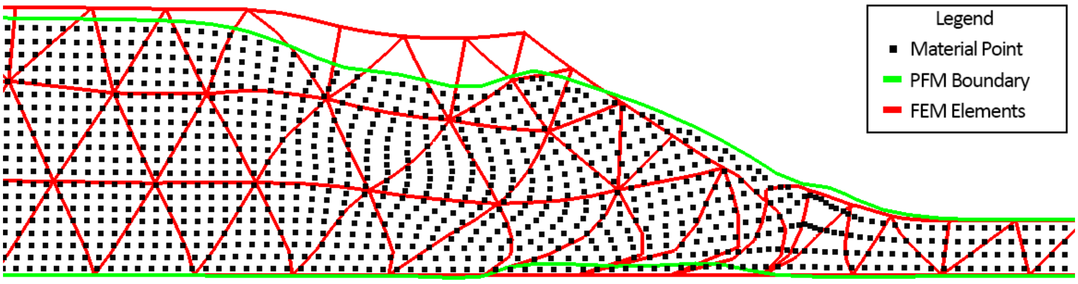


Figure 5.32: Comparison between FEM and PFM boundary at modelling a slope stability problem - Final deformation FEM, timestep 50 MPM

5.5.2. Comparison to FEM extracted from MPM

A better comparison between the edge detection with PFM and the boundary in FEM can be made when the constitutive model in FEM and MPM are the same. An FEM has been extracted from MPM by removal of the Eularian phase. Therefore, the constitutive model is the same. Moreover, the initial stress condition is the same in this FEM. This FEM has only been used to compare with PFM and complete testing has not been performed. Therefore, no guarantee can be made on the accuracy of the FEM. The computational parameters of FEM have been kept the same as for MPM, so the mesh size, shape functions and the number of gauss points are unchanged.

From the initial condition shown in Figure 5.33 three observations can be made, namely:

- The top and bottom boundary are modelled without a large error. The difference between these boundaries in FEM in comparison to the detected boundaries by PFM is only 0.05 m. This difference can be reduced further by an optimization of the contour level and size of the kernel functions of PFM.

- The error at the slope is large (roughly 0.3 m) due to the fact that the material points are initialized close to the surface instead of further below the surface. Especially the material point normally located in the upper right of a square cell is placed much closer to the boundary in the triangular cells. Further investigation is necessary on the placement of the material points to ensure a constant density. Moreover, the size and shape of the initial kernel function of PFM must be consistent with this ideal placement of the material points.
- As expected the smoothing of the proximity field reduces the accuracy at the crest and toe of the slope.

After 0.625 s of deformation the differences between the FEM and PFM boundary are comparable to the differences in the initial condition, as shown in Figure 5.34a. However, behind the crest of the slope the difference has increased to roughly 0.15 m, while the difference at the bottom of the slope has decreased to roughly 0.2 m. These changes are the result of a larger deformation of the FEM slope:

- For example, at the toe of the slope the material points are located at the left side of the elements due to a smaller horizontal deformation of the material points in comparison to FEM. The material points are further from the slope surface of FEM, which decreases the difference between the FEM and PFM boundaries.
- For example, at the crest of the slope the vertical deformation of the material points is smaller compared to FEM, which reduced the distance between the FEM boundary and the material points. Therefore, the difference between PFM and FEM increases.

The effects described above increase with time as shown in Figure 5.34b. So in short, the error of PFM in comparison to FEM is mostly the result of differences in the initial condition and differences between the deformation of MPM and FEM. The differences in the initial condition can be decreased with a better distribution of the material points and a further investigation of the size of the initial kernel functions and contour level. On the other hand, the differences between MPM and FEM are expected once deformation occurs and are also present in MPM without PFM. The error must be defined in the future, but for now PFM is promising.

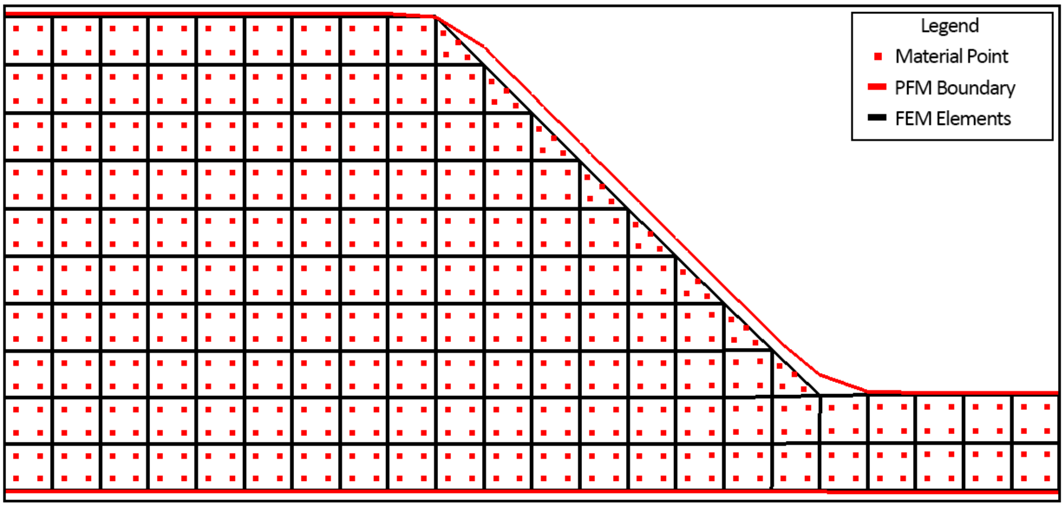
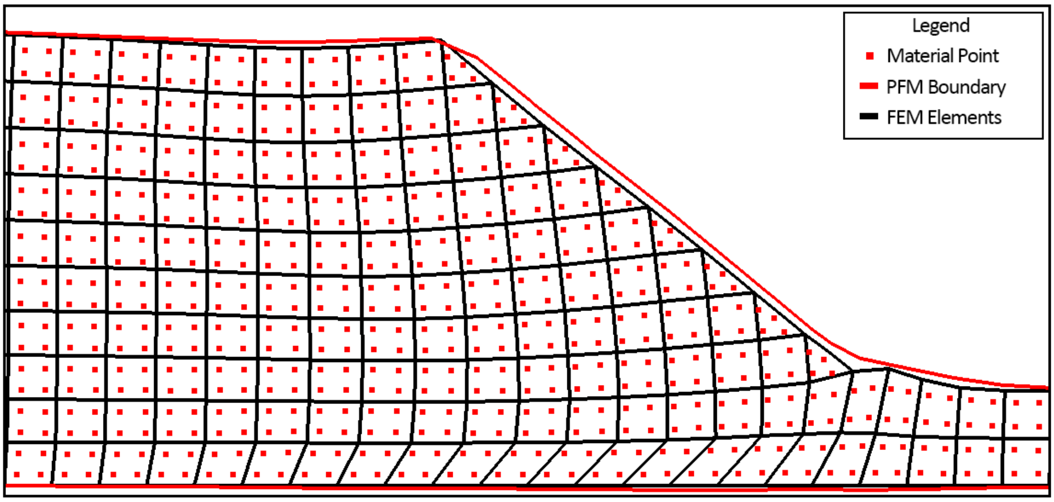
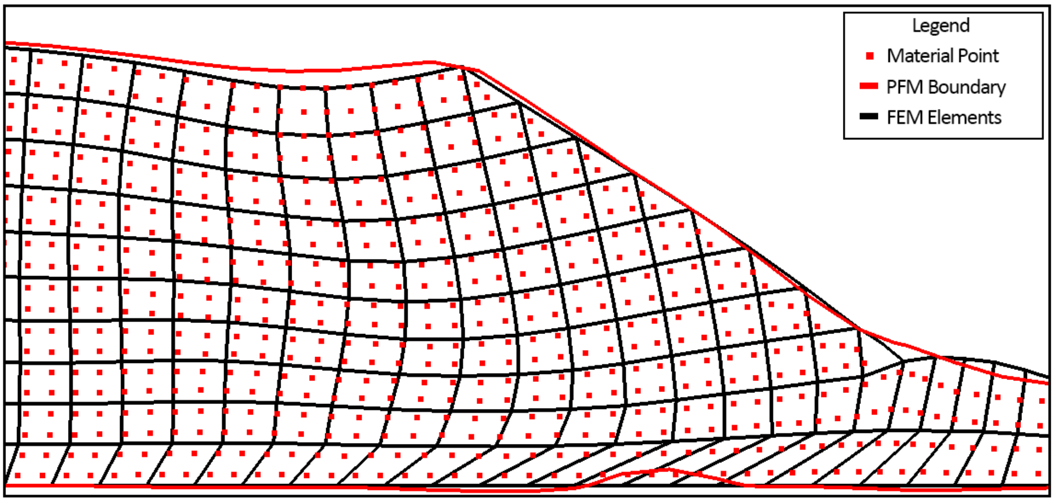


Figure 5.33: Comparison of the initial condition between an FEM extracted from MPM and edge detection with PFM for a slope stability problem



(a) After 0.625 s



(b) After 1.25 s

Figure 5.34: Comparison of the deformation between an FEM extracted from MPM and edge detection with PFM for a slope stability problem

5.6. Conclusion

The current version of PFM is more stable, faster and more accurate due to the improvements made within this section. TCA ensures a fully connected contour without any joints. To apply boundary conditions the boundary must be a complete separation between two or multiple materials. Moreover, the contour cannot have any joints when boundary conditions are applied. TCA separates different materials completely and the contour from TCA does not have any joints. TCA made PFM more stable as an unconnected or jointed contour can no longer break the application of boundary conditions. However, TCA is not perfect as it highly depends on the value of the proximity field at the ninth or center node. This dependency resulted in a couple of strange indents in the boundary and material points being located outside of the boundary. Further improvements are therefore necessary on this topic.

PFM's computation cost has seen a great reduction within this chapter. The computation time of the contour algorithm was reduced by using shape functions to calculate the complete proximity field once the proximity field value at the nodes is known. Furthermore, the calculation of the value at the nodes was sped up by changing the order of the summation. Instead of summing over each material point at each node, a summation over each node at each material point is made. Most nodes are not influenced by a certain material point and the order in the summation allows these nodes to be ignored. This lead to the final reduction in computation cost. The cost of the current state of the method is roughly equal to the cost of an explicit MPM time step.

Many unrealistically small holes where present according to the boundary computed at the start of this chapter. A couple of techniques have been mentioned to reduce the number of these holes. The best technique requires a small changes to the MPM model, namely the original placement of the material points. The points should be spaced equally through the domain to acquire a better approximation of the boundary. This also allows the boundary to be closer to the material points, thereby improving the accuracy of the model. However, further testing is necessary on MPM models if a different spacing of the material points is possible. A small addition is therefore made to the model, which manually removes some of the small holes based on their location and size. Finally, two filter techniques have been tested to both smooth the surface and reduce the number of holes. While a median filter did not prove to be useful, a Gaussian filter gives a smoother surface and removes most of the unrealistic holes. However, the surface found after the Gaussian filer is applied, is too close to the material points. It can even be located below the first layer of material points. This effect can be reduced by lowering the contour value, but it cannot be removed at sharp corners.

A final addition to PFM fixed a problem regarding the extreme deformation of kernel functions by introducing a maximum and minimum size of the kernel function. Unfortunately this is a physically incorrect solution and further improvements must be made on this topic to solve the problems. For example different shapes of the kernel function might solve of the issues and perhaps the strain over the entire kernel function could be taken into account to deform the kernel.

Unfortunately a comparison between MPM and FEM has proven to be difficult. As mentioned in Section 5.5, a slope stability problem has been implemented in Plaxis with a couple of adjustments to the soil model. Moreover, an FEM has been extracted from the MPM program, which uses the same constitutive model, but its accuracy is unknown. The approximation of the boundary with PFM is decent in comparison to FEM, but differences are clearly present. Especially at the slope surface the differences are large due to the distribution of the material points.

The deformation profile of the slope is comparable between the MPM and the Plaxis FEM. The main differences are a smaller toe height increase in MPM and the capability of modelling larger deformations with MPM in comparison to FEM. The deformation of the extracted FEM is larger in comparison to MPM, which explains the changing differences between the boundary found with PFM and the boundary of FEM. Finally, the troubling locations for FEM, due to mesh distortion, could be observed in MPM as highly compressed soil regions. Whether or not PFM and MPM provide an accurate representation of boundaries in comparison to reality is currently unclear due to the difficulties encountered when comparing MPM to FEM. T herefore, future investigations are definitely required to determine the accuracy of MPM and PFM.

6

Application of boundary conditions on irregular boundaries

After the boundary is detected by PFM, a boundary condition can be applied along its location. As mentioned in the previous two chapters, PFM has been determined to best fit the requirements of a boundary detection method. However, the application of boundary conditions does not depend on the detection by PFM. Instead the boundary could also have been detected by other methods. As long as the boundary is detected as a composite Bézier curve, the application of boundary conditions described within this section will be the same.

As mentioned within Section 3.3 irregular boundaries, boundaries not coinciding with the background mesh, are applied to the nodes of the background mesh. This method is comparable to the application of boundary conditions in FEM as well as conditions in MPM on regular boundaries, boundaries which do coincide with the background grid.

However, to apply irregular boundary conditions to the background mesh a transformation is required to move the condition from the location of the boundary to the nodes of the background grid. In this chapter this transformation is described for a surface traction. At first, the theoretical background behind this transformation is given, after which a surface traction is applied to a bearing capacity and slope stability problem.

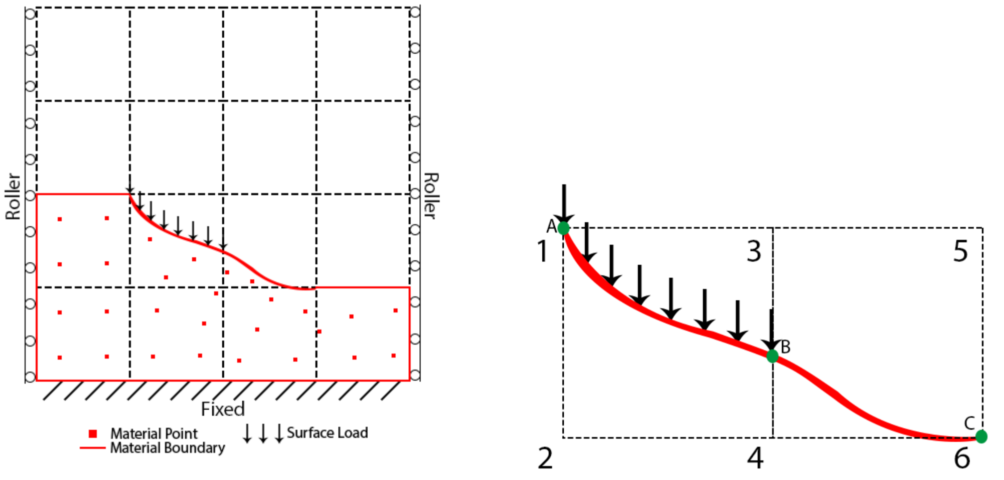
An irregular boundary is described by a composite Bézier curve, i.e. a curve consisting of polynomial sections. The curve is controlled by points found with the edge detection method described in Chapter 4. The transformations of the boundary conditions must therefore use the Bézier curve as their basis. The description of composite Bézier curve can be found in Section 4.1. The application of the boundary conditions on this composite Bézier curve is discussed within this chapter.

6.1. Surface traction on irregular boundaries: Theoretical background

As mentioned before, the boundary conditions along irregular boundaries are transferred to the nodes of the background grid in which the boundary is located. In this paragraph the transformation from a surface traction $\{\mathbf{t}\}$ on an irregular boundary Γ_t to a external nodal force $\{\mathbf{f}\}$ is given. Figure 6.1a is used as an example and a close up of this situation has been provided in Figure 6.1b. The equation used to transfer regular boundaries can be applied to irregular boundaries as well. So, to find the external nodal force based on the applied surface traction on an irregular boundary the following equation needs to be worked out:

$$\{\mathbf{f}\} = \int_{\Gamma_t} [\mathbf{N}]^T \{\mathbf{t}\} \tag{6.1}$$

where Γ_t is the part of the boundary which is under surface traction, $[\mathbf{N}]$ is the global shape function matrix and $\{\mathbf{t}\}$ is the effect of the applied traction on the surface in both the normal and tangent direction. An assumption is made that only normal forces on the boundary influence the soil body, $\{\mathbf{t}\}$ can therefore be replaced by $\{\mathbf{t_n}\} = \{\mathbf{t}\} \odot \{\mathbf{n}\}$, with $\{\mathbf{n}\}$ being the outward unit normal vector of the boundary.



(a) Surface traction applied to a failing soil slope (b) Close up of the irregular boundary within 2 cells

Figure 6.1: Surface traction on an irregular boundary

The assumption is valid if the friction between the reason behind the applied traction (for example a structure) and the soil body is assumed to be zero. Moreover, the influence on the soil body is assumed to be equal to the load $\{\mathbf{q}\}$ applied on the soil body. The traction $\{\mathbf{t_n}\}$ is then given by $\{\mathbf{t_n}\} = \{\mathbf{q}\} \odot \{\mathbf{n}\}$. The boundary has been chosen to be represented by a Bézier curve. This Bézier curve spans from point A to point B on the edge of the element containing the nodes for which the force is calculated, as can be seen in Figure 6.1. Therefore, the boundary curve is given by (Shene [30]):

$$\Gamma_t = B(t) = \sum_{i=0}^{n} b_{i,n}(t) P_i, t \in [0, 1] \tag{6.2}$$

in which the Bernsteinpolynomals $b_{i,n}(t)$ are defined to be:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, i = 0, ..., n \tag{6.3}$$

P_0 and P_n are the points A and B and define the endpoints of the Bézier curve. Points $P_1 ... P_{n-1}$ define the shape of the Bézier curve, but the curve will not cross these points. n is the order of the Bézier curve. Only Bézier curves of the first and third order will be used and therefore a maximum of 4 points are needed to describe the boundary. The description of the boundary is given in more detail in Section 4.1.

Equation (3.29) can therefore be changed to

$$\{\mathbf{f}\} = \int_a^b [\mathbf{N}]^T \{\mathbf{q}\} \odot \{\mathbf{n}\} d\Gamma \tag{6.4}$$

The outward unit normal vector $\{\mathbf{n}\}$ of the boundary is a function of the Bézier curve $B(t)$, due to the fact that the slope of the normal line of the Bézier curve is given by:

$$m = \frac{-1}{B'(x)} = \frac{-1}{dy/dx} \tag{6.5}$$

As long as the boundary is not vertical $(dx/dt \neq 0)$ this can be expanded to be:

$$m = \frac{-1}{\frac{dy/dt}{dx/dt}} \tag{6.6}$$

In case $dx/dt = 0$ the outward normal is known, due to the fact that the boundary is vertical. From m the outward normal can be found to be $[1 \quad m]^T$, which needs to be normalized to find $\{\mathbf{n}\}$. The

Table 6.1: Gaussian integration points and weights for $k = 5$

Integration Point i	Weight w_i	Location u
1	$\frac{128}{225}$	0
2,3	$\frac{322+13\sqrt{70}}{900}$	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$
4,5	$\frac{322-13\sqrt{70}}{900}$	$\pm\frac{1}{3}\sqrt{5+2\sqrt{10/7}}$

derivative of a Bézier curve is found to be a Bézier curve as well (Shene [30]). The derivative $B'(t)$ is given by:

$$B'(t) = \sum_{i=0}^{n-1} b_{i,n-1}(t)[n(P_{i+1} - P_i)], t \in [0,1] \tag{6.7}$$

Or in other words $B'(t)$ is given by a Bézier curve of order $n-1$ defined by the points $n(P_1 - P_0)$,...,$n(P_n - P_{n-1})$ (Shene [30]). Using the definition of the boundary by a Bézier curve given in equation (6.2), equation (6.4) can be changed to:

$$\{\mathbf{f}\} = R\int_0^1 [\mathbf{N}(\mathbf{t})]^T\{\mathbf{q}\} \odot \{\mathbf{n}(\mathbf{t})\}dt \tag{6.8}$$

where R is the length of the segment A,B of the boundary, $[\mathbf{N}(\mathbf{t})]^T$ is the shape function matrix along the boundary. Another remapping to an interval of $[-1,1]$ will allow for an approximation using the Gaussian quadrature. The remapping of equation (6.8) is given by:

$$\{\mathbf{f}\} = \frac{R}{2}\int_{-1}^1 [\mathbf{N}(\frac{\mathbf{1}}{\mathbf{2}}(\mathbf{u}+\mathbf{1}))]^T\{\mathbf{q}\} \odot \{\mathbf{n}(\frac{\mathbf{1}}{\mathbf{2}}(\mathbf{u}+\mathbf{1}))\}du \tag{6.9}$$

R can be computed using the arc length $A(t)$ of a geometric equation (Wang et al. [39]):

$$A(t) = \int_{t_0}^t \left(\frac{dx}{dt}(t)^2 + \frac{dy}{dt}(t)^2\right)^{\frac{1}{2}} dt \rightarrow R = A(1) = \int_0^1 \left(\frac{dx}{dt}(t)^2 + \frac{dy}{dt}(t)^2\right)^{\frac{1}{2}} dt \tag{6.10}$$

Equation (6.10) is again mapped to a domain of $[-1,1]$, giving:

$$R = \frac{1}{2}\int_{-1}^1 \left(\frac{dx}{dt}(\frac{1}{2}(u+1))^2 + \frac{dy}{dt}(\frac{1}{2}(u+1))^2\right)^{\frac{1}{2}} du \tag{6.11}$$

Both equations (6.9) and (6.11) can be aproximated using the Gaussian quadrature resulting in (Duraiswami [14]):

$$\{\mathbf{f}\} = \frac{1}{4}\sum_{i=1}^k w_i\left(\frac{dx}{dt}(t)^2 + \frac{dy}{dt}(t)^2\right)^{\frac{1}{2}} \sum_{i=1}^k w_i[\mathbf{N}(\mathbf{t})]^T\{\mathbf{q}\} \odot \{\mathbf{n}(\mathbf{t})\} \tag{6.12}$$

where $t = \frac{1}{2}(u+1)$, k is the number of integration points, w_i are the weights corresponding to the integration points and u is the location of the integration points. w_i and u are given in Table 6.1.

6.1.1. Error of the Gaussian quadrature

The integrals of equations (6.9) and (6.11) are generally not polynomials and the integration using the Gaussian quadrature is therefore not exact. To reduce the error, the number of integration points is chosen to be five. Higher number of integration points can be used for a higher accuracy, but the computation cost increases as well. u and w_i for $k = 5$ are given in Table 6.1. Even though the force distribution of equation (6.12) is not exact, the total force per element, given by the sum of the forces per element, can be calculated exactly. The total force in the horizontal $f_{h,exact}$ and vertical direction $f_{v,exact}$ are given by:

$$\begin{aligned} f_{h,exact} &= q_h(y_b - y_a) \\ f_{v,exact} &= q_v(x_b - x_a) \end{aligned} \tag{6.13}$$

The error e of the distributed force can now be calculated as:

$$e = \sum_{k=1}^{dim} \{\mathbf{e}\} = \sum_{k=1}^{dim} \left(\sum_{i=1}^{nod} \{\mathbf{f_i}\} \right) - \{\mathbf{f_{exact}}\} \tag{6.14}$$

where nod is the number of nodes per element and dim is the number of dimensions. The following test has been executed to approximate the error of the distribution of the force on one element. One element has been created with a boundary given by a Bézier curve. During the first test the Bézier curve is of order 1 with a fixed point A and a moving point B both located on the edges of the element. A couple of the tested boundaries can be seen in Figure 6.2. The error in the force during these test was negligible (in the order of $10^{-14}\%$ in comparison with the applied load). This is logical due to the fact that equation (6.12) becomes a polynomial if the boundary is linear (dx/dt, dy/dt and $\{\mathbf{n}\}$ are constant and $[\mathbf{N}]$ is a polynomial under these circumstances). Therefore, the Gaussian integration is exact.

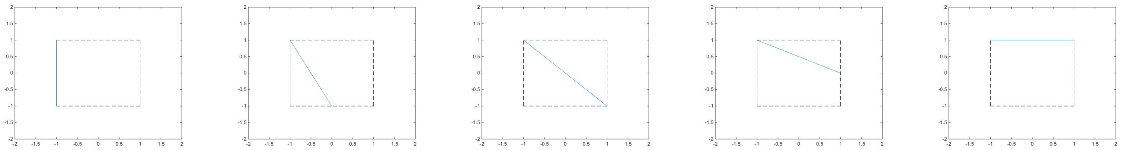

(a) (b) (c) (d) (e)

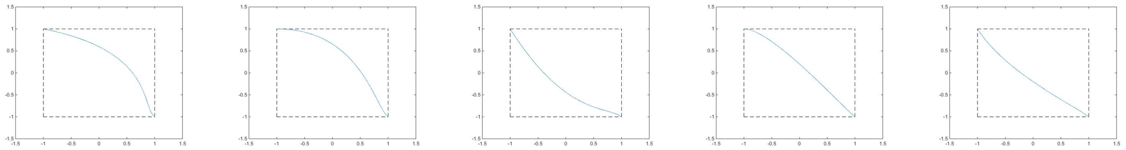
Figure 6.2: Linear boundaries used in testing the error of the distributed force

Besides the linear boundary also cubic Bézier curves have been tested. A couple of examples of cubic boundaries, which have been tested, can be seen in Figure 6.3. Point A has been fixed to $[-1, 1]$, point B has been fixed to $[1, -1]$, while point C (the first control point defining the Bézier curve) varies within this domain. Finally the second control point D is fixed to $[0.75, -0.75]$. For each point C within the domain the percentage error has been calculated and a contour plot of this error is shown in Figure 6.4. Along the line A,B the error is zero as expected, due to the fact that the Bézier curve will then be linear. Moreover, the error is relatively small if point C is close to point D. The reason for the small error, is the mostly linear shape, i.e. a small curvature, of the boundary over most of its length. The curvature is higher only over a small length close to points C and D. Due to the mostly linear shape, the Gaussian integration presents decent results.

The increase in error is large when point C is not located close to either point D or line A,B. The maximum error has been found to be roughly 14%. This error can increase even further when point D is no longer fixed. Finally, while points A and B will always be on the boundary of the elements due to the assumption made for the boundary in Section 4.1, the control points C and D could be located outside of the cell shown in Figure 6.3. This increases the error. This could be resolved by splitting the cubic Bézier curves at the cell boundaries when point C and/or D is located outside of the element. Currently due to the potential of this large error only linear boundaries have been implemented for the application of boundary conditions. Additional research is required to determine whether cubic curves are an improvement of the linear curves. Perhaps, a different description besides Bézier curves is necessary to include non linear discriptions without the error found in this section. As a first advice to reduce the error of the force distribution on non-linear boundaries, the distributed force $\{\mathbf{f}\}$ can be scaled down to the required total force $\{\mathbf{f_{total}}\}$:

$$\{\mathbf{f_{scaled}}\} = \{\mathbf{f}\} \frac{\{\mathbf{f_{total}}\}}{\left(\sum_{i=1}^{nod} \{\mathbf{f_i}\} \right)} \tag{6.15}$$

where $\{\mathbf{f_{scaled}}\}$ is the scaled force. The total force will be correct due to this scaling. However, the distribution might still be incorrect.



(a) (b) (c) (d) (e)

Figure 6.3: Examples of cubic boundaries used in testing the error of the distributed force

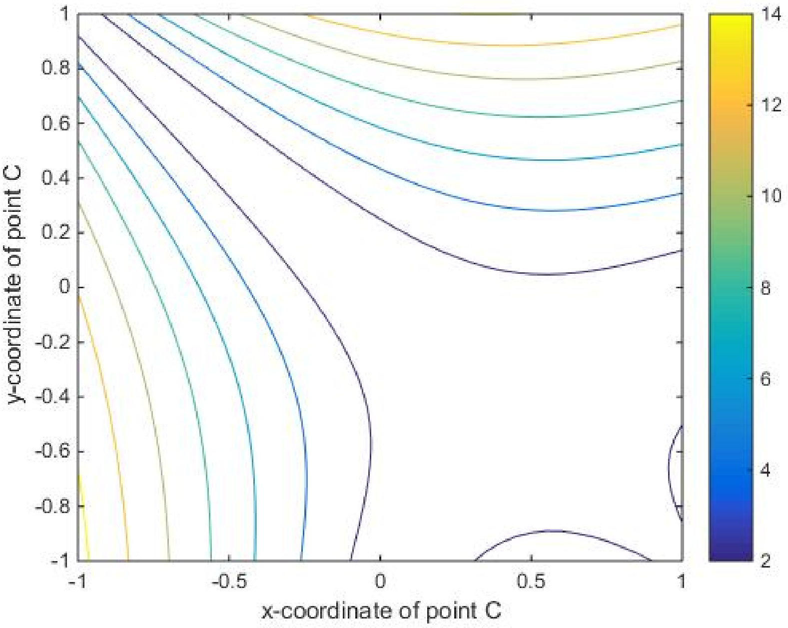
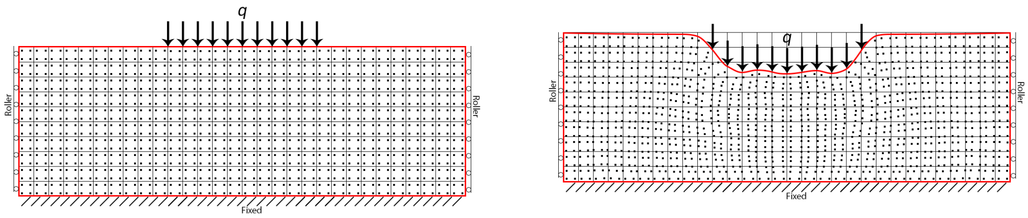


Figure 6.4: Contour of the error percentage of the force distribution for different locations of point C for cubic boundaries

6.2. Surface traction on a rectangular soil body

As a first test of the application of a surface traction to a boundary detected with PFM, a vertical load has been applied to the top of a rectangular soil body. For example, this test could be used to represent the bearing capacity of the soil beneath a loaded foundation. The soil body has also been subjected to gravity and was bounded by a fixed bottom boundary and two roller boundaries at the right and left boundary of the domain. The problem description is given in Figure 6.5. Both an initial condition as well as a deformed condition are given within this Figure. The soil body was 15 m wide and 5 m high and a vertical surface traction q_v was placed on top of the center 5 m. This surface traction is always applied as a vertical load to the soil surface even after deformation has taken place, as shown in Figure 6.5b. The load is applied at the surface of the soil and is capable of moving vertically. However, the horizontal location of q_v is fixed. Due to the fact that PFM is tested, it is used to detect the surface.



(a) Surface traction applied to the initial condition

(b) Surface traction applied during the deformation process

Figure 6.5: Surface traction on a rectangular soil body

Different surface tractions q_v have been tested using different soil parameters to observe the effect of the surface traction on the soil body. These tests will be discussed within this section. In this introduction the figures used to display the deformation process are explained using a surface traction $q_v = 100\ kPa$. The background grid of MPM uses 4-node quadrilateral elements with a size of 0.5 m by 0.5 m. Four material points have been placed inside of each grid cell. The material points are initialized either at the Gauss points or with an equal spacing between them of 0.25 m. The initial location of the material points is discussed further in Section 6.2.1. In total 300 background grid elements are used together with 1200 material points. Dependent on the height increase to either side of the surface traction, additional background elements can be necessary. An additional 150 elements have been placed on top, which ensures that the material points stay inside of the background grid. PFM uses the same grid size, but instead uses 9-node quadrilateral elements as mentioned in Section 5.1.3.

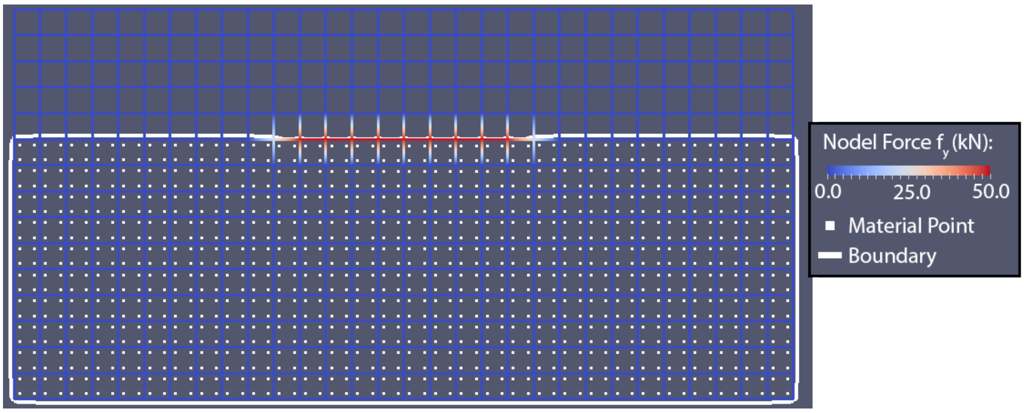
The nodal forces due to the surface traction are shown on the background grid to show the force applied to the soil surface during the deformation process. A clear distribution of the applied load to the background grid is visible in Figure 6.6b as the surface is located roughly at the center of a background grid cell. On the other hand the load is concentrated along a straight line or at two distinct points in Figures 6.6a and 6.6c respectively, due to the fact that the boundary is close to the background grid at these locations.

The distribution of the load is not problematic as long as material points are present within one of the grid cell over which the load is distributed. If all of the grid cells surrounding a loaded node are empty some of the distributed force will not be applied to the material points as a stiffness relationship is not calculated for the empty cell.

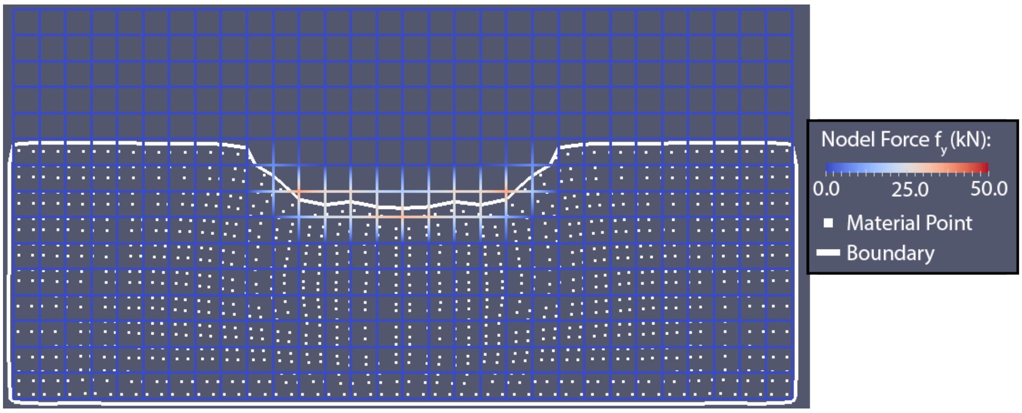
As an example, a close up is made of the problem described earlier in Figure 6.6d. The nodes in a red circle are surrounded by empty cells. Luckily, the load calculated at these nodes is almost zero. However, the nodes in a green circle are loaded significantly and have only one cell containing material points. One could imagine the last cell to be empty similar to the cells around the red nodes. For example, if the two center material point are 0.1 m lower. The location of the boundary would drop as well, but the boundary would still be located inside the empty cells. A significant part of the load would still be applied to the green nodes, which would not be distributed to the material points. Luckily, this problem only occurs when the surface is relatively close to a grid cell boundary. The lost force is therefore only a small portion of the applied force. Moreover, this often only occurs during a couple of time steps and the loss of force during the entire deformation process is therefore small. However, the effect of this lost force must be taken into account in future research.

Mesh dependency of PFM can be clearly observed from Figure 6.6. Especially, in 6.6d the boundary is detected along the edge of a background grid cell. The end points of the composite Bézier curve are detected on the edges of the proximity field grid, which uses 8-node quadrilateral elements. Between each of the nodes of the grid one end point can be detected. Due to the fact that on each vertex of the quadrilateral element three nodes are located, two end points can be detected per vertex. Therefore, a straight boundary along the vertex is possible. The boundary appears to be attracted to the grid boundary due to the mesh dependency. A smaller proximity field grid, in this case a grid equal in size to the MPM background grid is used, can reduce the effect of the mesh dependency. Higher order Bézier curves can also remove the effect, due to the fact that only the end points of the curves are fixed to a grid.

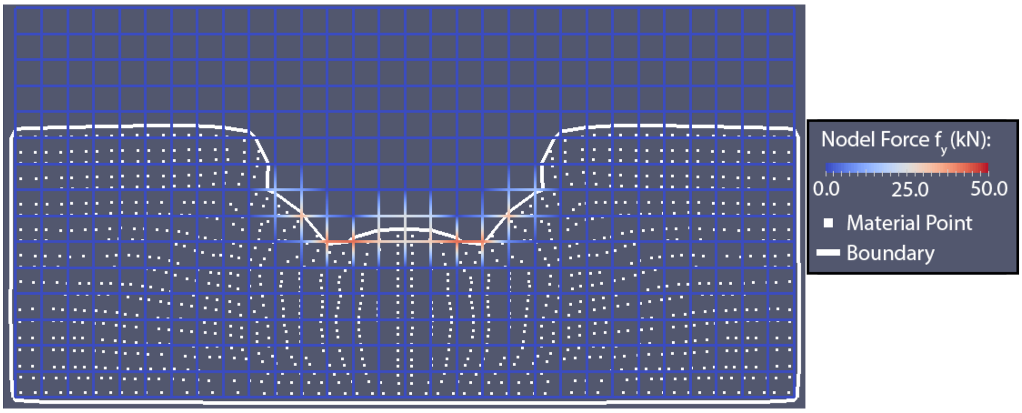
A parametric study of the bearing capacity problem has been executed. The results are given within this section. At first, the soil has been defined with the parameters in Table 6.2. In the parametric study the effect of different parameters on the solution has been compared with this general case. It is therefore important to first analyse the general case and observe the effects of the force on this general case.



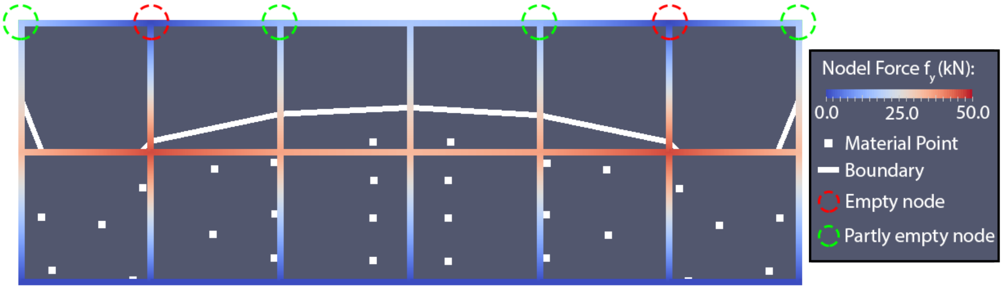
(a) Time step 1



(b) Time step 25



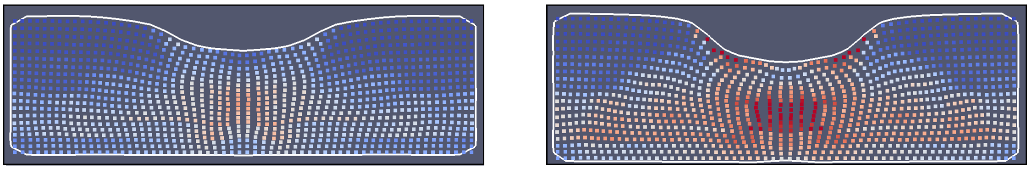
(c) Time step 50



(d) Detail of time step 53 with distribution of load over empty cells

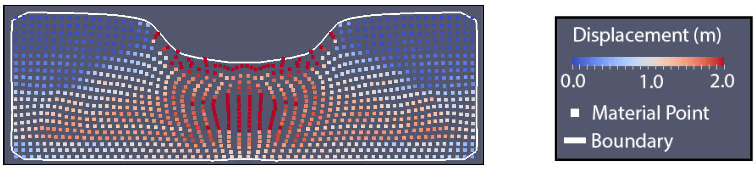
Figure 6.6: Nodal forces due to a surface traction on a rectangular soil body

As visualized in Figure 6.7 a shear band developed due to the applied load $q_v = 50kPa$. However, the largest plastic strains are vertical strains directly below the surface and at the center of the soil body. The size of these strains can be explained due to the load applied at the surface. The maximum amount of strains visualized in this figure are bounded to 1, because the strains at the surface are much larger than the strains in the shear band. The unbounded plastic deformation is shown in Figure 6.8, where the plastic strains at the surface are more than four times as high as the plastic strains below the surface. The extreme deformations are the result of rolling material points, which will be addressed after the displacement and total stress profiles are given.



(a) After 0.75 seconds

(b) After 1.625 seconds



(c) After 3.125 seconds

(d) -

Figure 6.7: Plastic displacement due to a load $q_v = 50$ kPa. The colouring of the plastic strains only displays a maximum displacement of 1 m

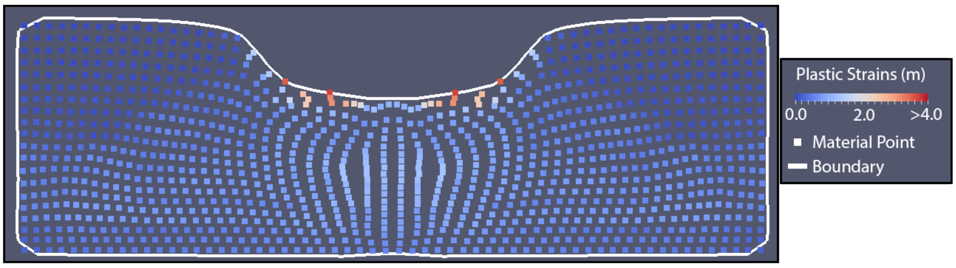


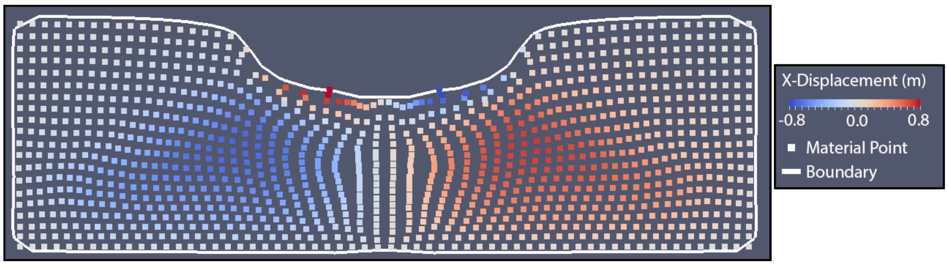
Figure 6.8: Plastic strains after deformation without a maximum to the colouring of the strains

The total horizontal and vertical displacement profiles are given in Figures 6.9a and 6.9b respectively. As expected the vertical displacement is concentrated below the location of the load. The horizontal displacement is mainly located below the surface which is not directly loaded. The reasoning behind these profiles is the load displaced the soil beneath the load, which increased the vertical and horizontal stress beneath the load. The increase in horizontal stress results in a horizontal deformation to either side of the load.

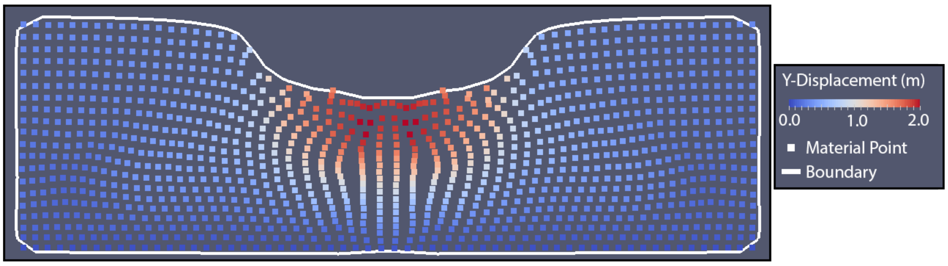
One of the largest problems with the current MPM model is the stress distribution at the start of the deformation process. Currently the initial total stress distribution is 0 kPa over the entire soil domain.

Table 6.2: Parameters used in the baseline test

Parameter	Symbol	Value
Young's modulus	E	10^3 (kPa)
Poisson's Ratio	ν	0.15 (-)
Volumetric weight	γ	20 (kN/m^3)
Initial Cohesion	c_i	20 (kPa)
Peak Cohesion	c_p	10 (kPa)
Residual Cohesion	c_r	5 (kPa)
Surface traction	q_v	50 (kPa)



(a) Horizontal displacements due to a load q_v = 50 kPa

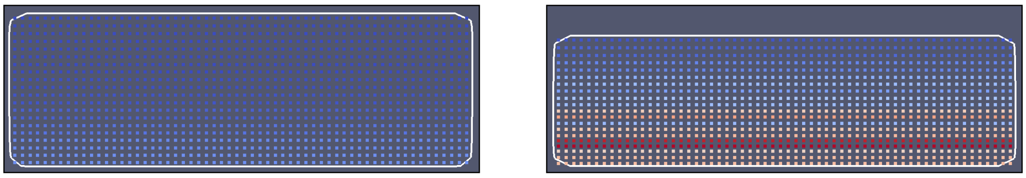


(b) Vertical displacements due to a load q_v = 50 kPa

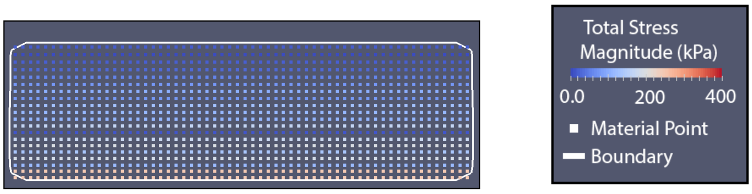
Figure 6.9: Displacements due to a load q_v = 50 kPa

Due to the fact that part of the load is the result of gravity, which is not in balance with the internal stress distribution, deformation already occurs due to only the gravitational load. In Figure 6.10 the stress distribution of the first seconds after the initial loading by gravity is visualized. An increase in load over the height of the soil domain, due to the weight of the soil, lead to a displacement of the soil. This displacement further increased the stress in the soil, due to damping. This damping induced further plastic deformation. The stress is no longer linearly increasing with depth during this deformation as shown in Figure 6.10b. After this downward deformation the additional internal stresses lead to an expansion of the soil, which reduces the internal stresses. The soil basically acts as a spring, which oscillates until the system reaches an equilibrium. An equilibrium is reached after roughly 3 seconds. The oscillations also produce some plastic deformation.

An additional surface traction obviously increases the stresses in the soil body as shown in Figure 6.11. However, even though a large load is applied the initial deformation is still dominated by the application of the gravitational load. The same oscillations can also be observed to either side of the load. Beneath the load the oscillations are reduced due to the higher load. Stress oscillations are still present within the solution and this problem must be solved in future research to observe reasonable solutions.

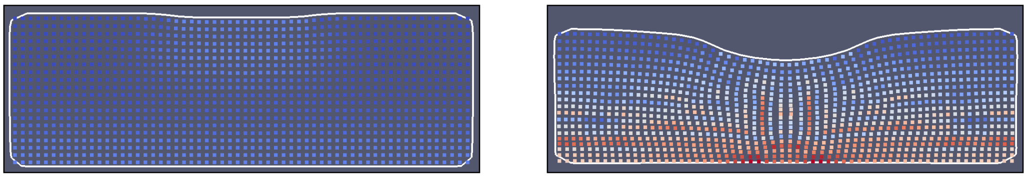


(a) After 0.125 seconds (b) After 0.625 seconds

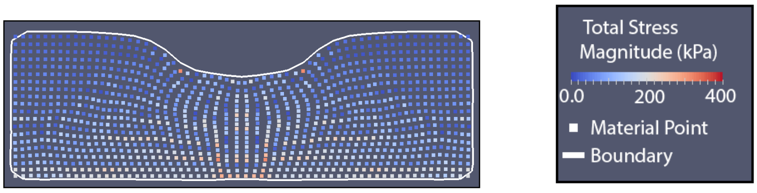


(c) After 1.25 seconds (d) -

Figure 6.10: Total stress distribution and deformation due to only gravity. No additional surface traction is applied.



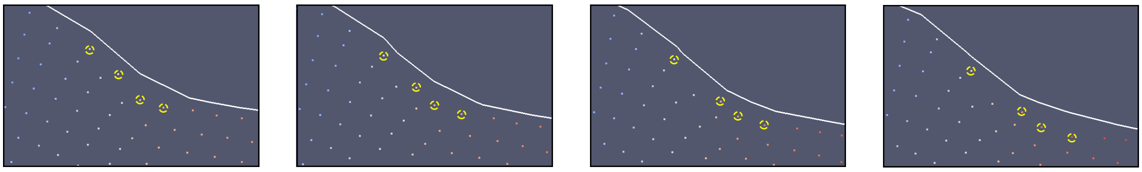
(a) After 0.125 seconds (b) After 0.625 seconds



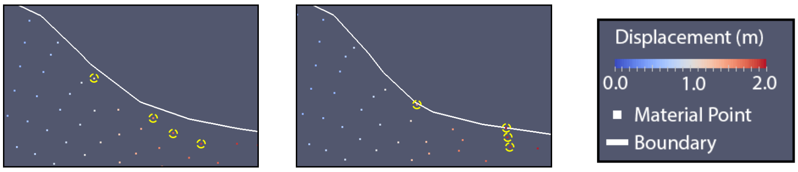
(c) After 1.25 seconds (d) -

Figure 6.11: Total stress distribution and deformation due to gravity and an additional surface traction.

Finally, one additional feature of MPM is the possible rolling of material points. This rolling is clearly visible in Figure 6.12, where the material points at the surface roll over the material points below the surface. While FEM would have large problems describing this behaviour, the behaviour is modelled appropriately in MPM. Moreover, SMM would be unable to detect the surface due to this rolling as the markers would no longer represent the surface. However, with PFM the surface is detected appropriately.



(a) After 1.25 s (b) After 1.375 s (c) After 1.5 s (d) After 1.625 s



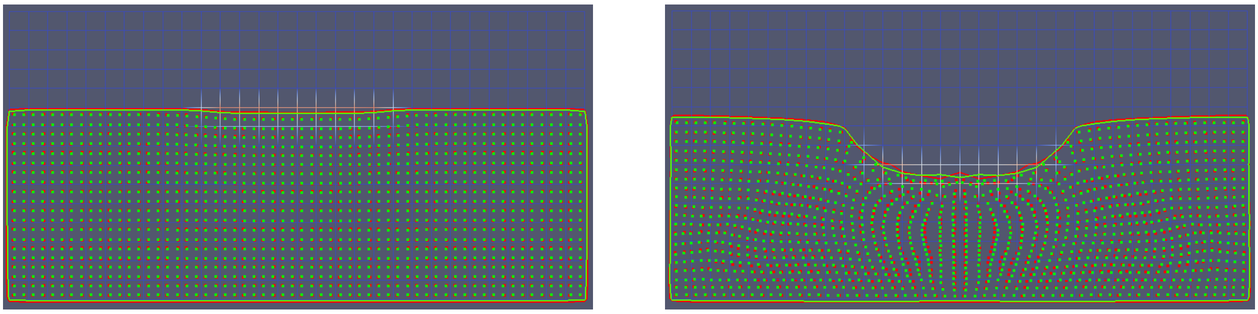
(e) After 1.75 s (f) After 3 s (g) -

Figure 6.12: Detail of the soil surface indicating rolling material points

In the paragraphs of this section a parametric study has been executed. At first, the influence of the smoothing methods described in Section 5.3 on the applied force is discussed. From these test a decision is made, which smoothing method should be applied during the parametric study. In the parametric study the Poisson's ratio, stiffness and soil strength have been varied under different loading conditions.

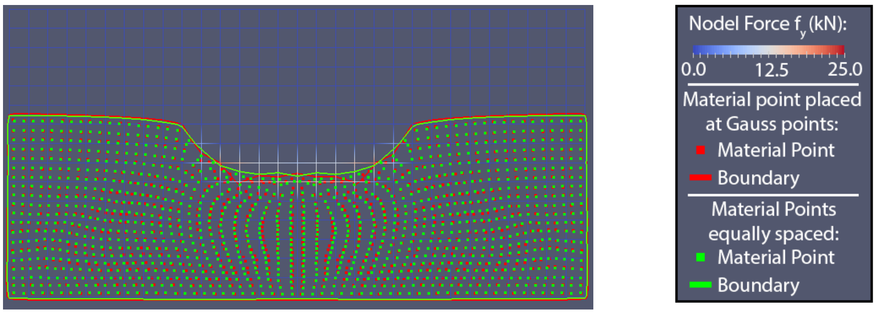
6.2.1. Effects of smoothing of the proximity field surface

To test the effects of the smoothing of the proximity field surface the two most promising techniques, namely equally spacing the material points and a Gaussian filter with a decreased contour value, have been compared to a normal proximity field. The soil parameters and the applied surface traction q_v used in this test are presented in Table 6.2. Even though equally spacing the material points has not been proven to be accurate, it did present a smoother surface as can be seen in Figure 6.13. Moreover, the location of the surface with equally spaced material points is closer to the actual surface in the initial condition. Due to this smoother surface the applied force is smoother as well, resulting in an smoother load distribution. Whether or not this smoother solution is correct is currently unknown. However, besides a better boundary approximation in the initial condition the approximation is also better at the end of the deformation process. While the surface detected from material points placed at the gauss points is detected below the material points in multiple time steps, this is not the case for the detection of the surface using equally space material points. Therefore, equally spacing the material points is chosen.



(a) After 0.125 seconds

(b) After 3.125 seconds



(c) After 6.250 seconds

(d) -

Figure 6.13: Comparison between equal spaced material points and material points placed at the Gaussian integration points.

However, to completely remove the inconsistency in the surface of the proximity field a Gaussian filter is also required. As shown in Figure 6.14 an inconsistency is still possible even if equally spacing of the material points is used. This can be fixed by using a gaussian filter, which further smooths the surface of the proximity field. The only problem created with a Gaussian filter is a loss of sharp corners even at locations where they should be present. However, as long as no force is applied at these corners it is advised to use a Gaussian filter.

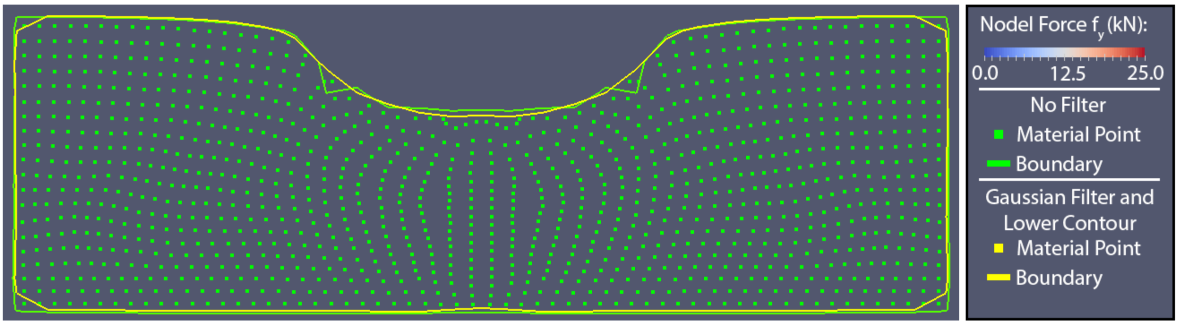


Figure 6.14: Comparison of the applied force between using a Gaussian filter on the proximity field with a field without a filter

6.2.2. Effect of the Poisson's ratio

In the first part of the parametric study the Poisson's ratio was varied from $\nu = 0.15$ to $\nu = 0.45$. The results of these changes are given in Figure 6.15, where it is clear that the displacement is reduced when ν is increasing. This mainly is the result of less plastic strains as shown in Figure 6.16. In this figure the plastic strains for $\nu = 0.45$ are given, which are orders smaller than the plastic strains for smaller Poisson's ratios. Rolling of the boundary is not observed due to the smaller displacement height in the test with larger Poisson's ratios. A shear band can still be observed in Figure 6.16, but the width of the band is much smaller in comparison to the original tests.

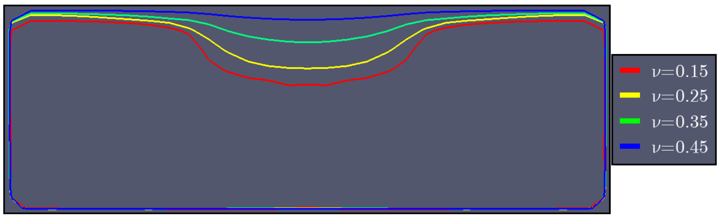


Figure 6.15: Comparison between four different Poission's Ratios under the same load

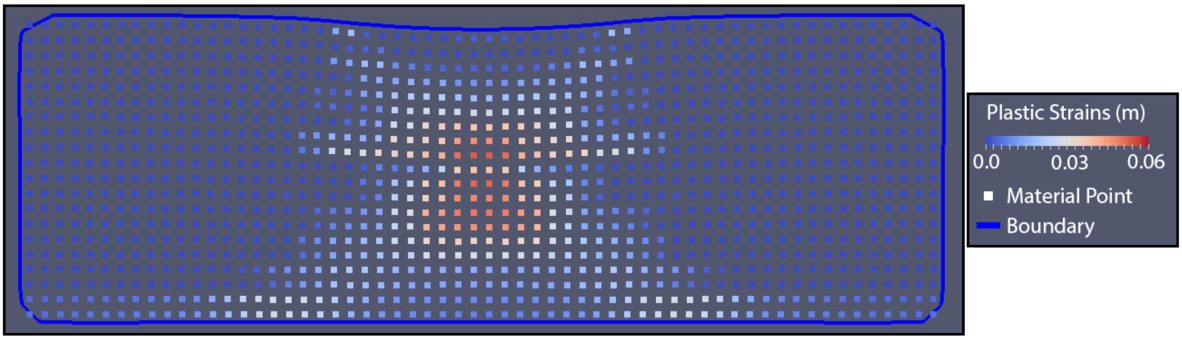
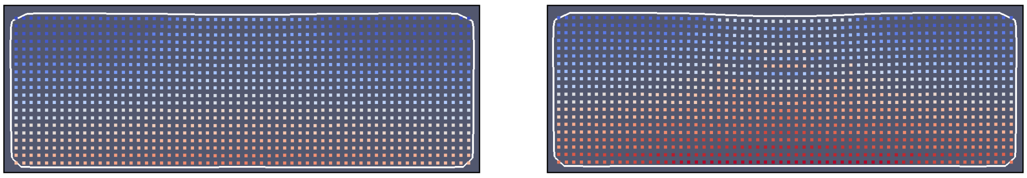


Figure 6.16: Plastic strains for $\nu = 0.45$

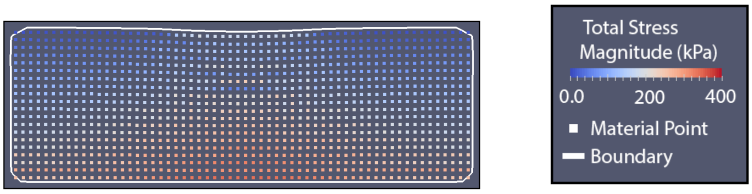
6.2.3. Effect of the stiffness

As expected, similar to an increase of the Poisson's ratio, a decrease of displacement is observed after an increase of the stiffness of the material. Higher stiffness obviously decreases deformation. Due to the initial total stress distribution this effect is enhanced. Due to the higher stiffness the oscillations occurring as a result of a gravitational load are smaller resulting in a smaller increase in stress, as shown in Figure 6.17. The oscillations are also much faster, fading completely after 1 second. Moreover, the smaller oscillations are almost completely elastic deformations and only the surface load induces plastic deformations.



(a) After 0.125 seconds

(b) After 0.25 seconds



(c) After 1.25 seconds

(d) -

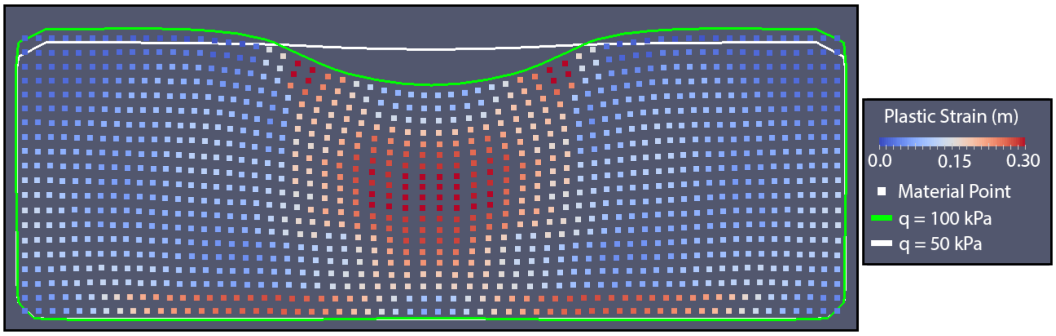
Figure 6.17: Total stress distribution and deformation due to gravity and an additional surface traction with a Young's modulus $E = 10^4 kPa$.

A higher surface traction $q_v = 100kPa$ combined with a higher Young's modulus $E = 10^4 kPa$ clearly shows two shear bands along the bottom of the domain (see Figure 6.18). However, the largest plastic strains are still located at the soil surface as shown in Figure 6.18b. Due to fixed bottom and side boundaries the soil is pushed upwards above the original soil surface. Some of the material points lifted above the original surface roll back into the loading zone and a circular motion starts to occur.

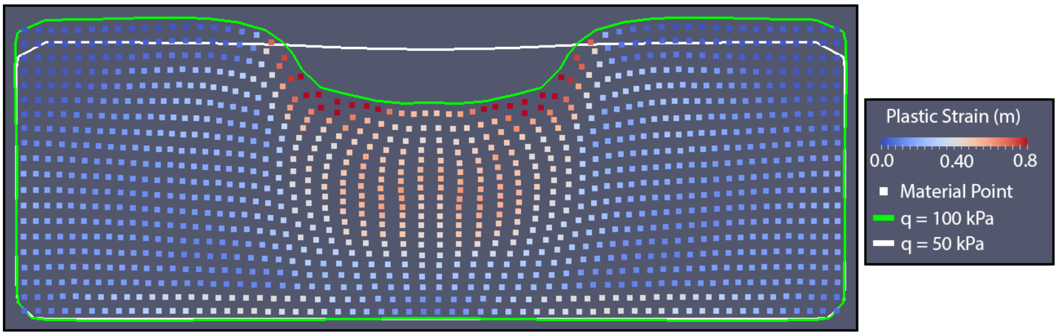
Unfortunately the current MPM model is unable to model the high stresses and plastic strains beneath the load and the deformation process after the timestep shown in Figure 6.18b is unknown. The model is unable to compute a correction stress and therefore stopped at this timestep. The exact cause is unknown, but three hypotheses are given below:

- The relatively small soil body with the fixed bottom and side boundaries could be a cause. The bottom boundary prevents further vertical deformations and the side boundaries do not allow the material to expand horizontally. The soil is therefore unable to deform plastically after the timestep shown in Figure 6.18b to reduce the stresses.
- MPM might be unable to handle the extreme deformation of the surface material points correctly.
- The applied load is too high for MPM to handle in comparison to the stiffness and strength of the material due to the small imperfections occurring with plastic deformations.

A further increase of the stiffness results in an even smaller deformation, which is influenced even less by the initial gravitational loading. A further increase of the load at this higher stiffness to $q_v = 150kPa$, presents similar results as seen previously in Figure 6.18. Due to the higher load and therefore higher stresses the model stopped working even earlier. Figures of the further increase in stiffness are not presented due to their similarity to Figures 6.17 and 6.18.



(a) After 1.25 seconds

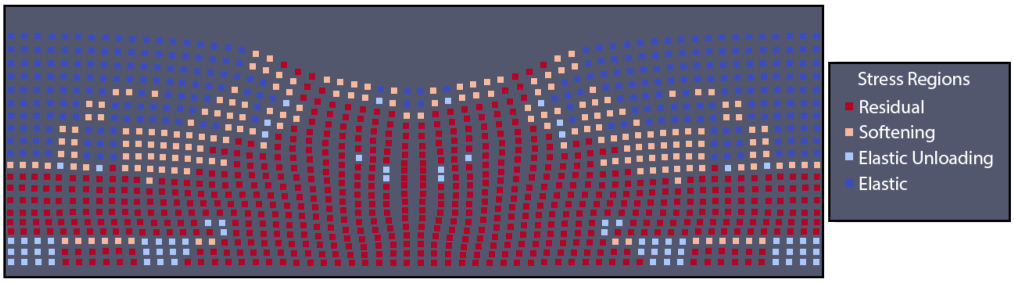


(b) After 2.5 seconds

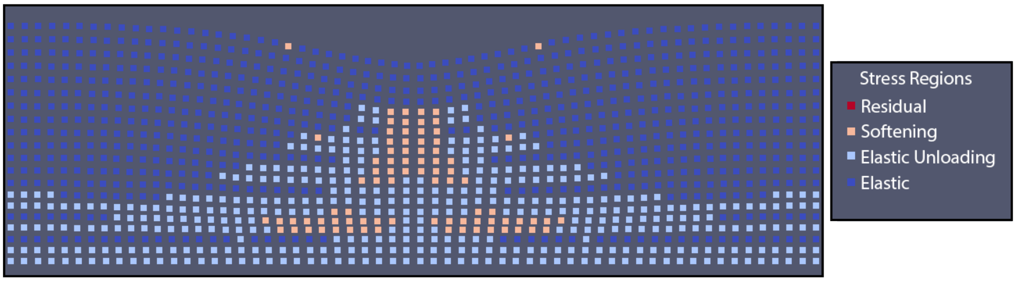
Figure 6.18: Plastic strains due to an increased surface traction $q = 100kPa$ and a higher stiffness $E = 10^4 kPa$.

6.2.4. Effect of the soil strength

The deformation is also reduced by increasing the strength of the soil as shown in both Figure 6.19 and Figure 6.20. With a stronger soil strength the residual strength is never reached even though the stresses are in the same order of magnitude. On the other hand with a weaker soil, during the initial gravitational loading more than half of the soil body reached the residual stress stage. After the oscillations some material points return to elastic unloading, but the strength loss is present through most of the domain. This results in larger deformation due to the surface traction in comparison to the stronger soil. The stronger material is only loaded until the softening stage and a smaller deformation profile due to both the gravitational loading as well as the surface traction is the result.

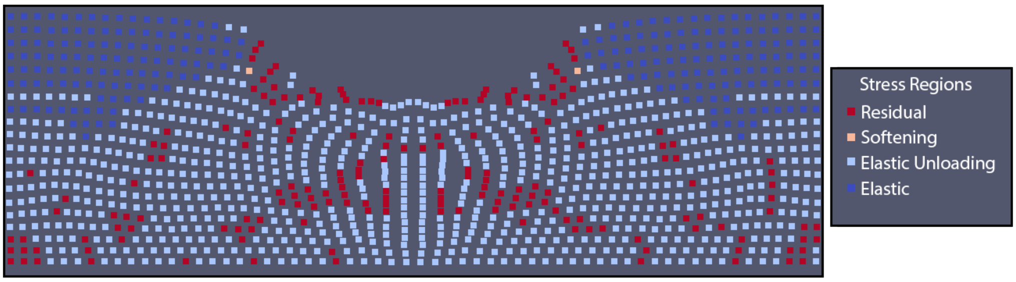


(a) $c_i = 20kPa, c_p = 10kPa, c_r = 5kPa$

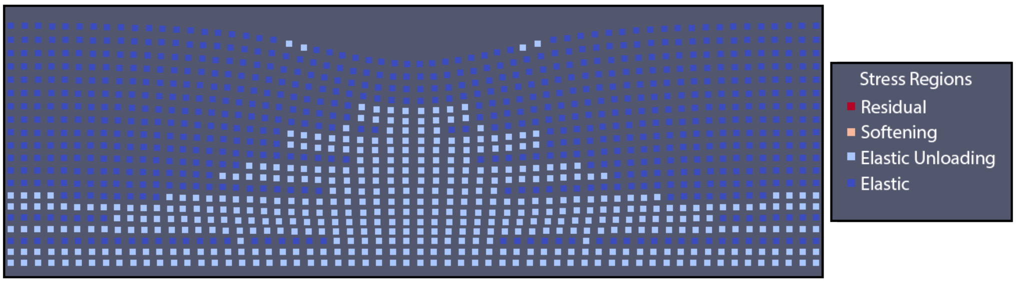


(b) $c_i = 40kPa, c_p = 20kPa, c_r = 10kPa$

Figure 6.19: Stress regions after 0.625 seconds for two different soil strengths.



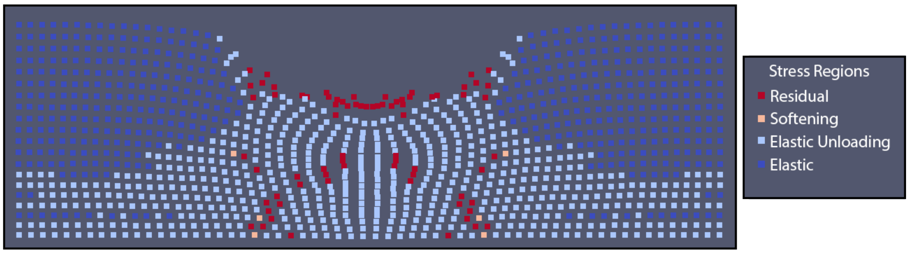
(a) $c_i = 20kPa, c_p = 10kPa, c_r = 5kPa$



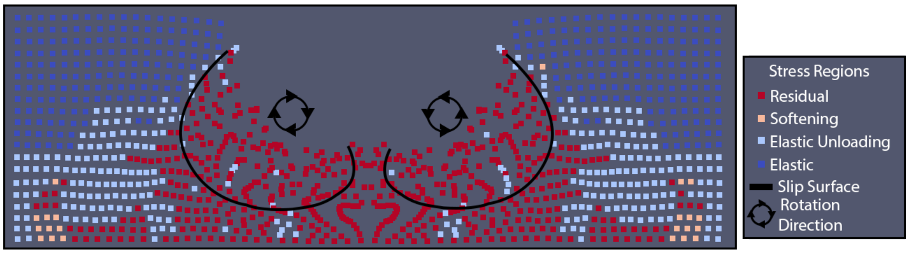
(b) $c_i = 40kPa, c_p = 20kPa, c_r = 10kPa$

Figure 6.20: Stress regions after 2.50 seconds for two different soil strengths.

The stronger soil has been loaded by a large load as well as shown in Figure 6.21. Surprisingly, increasing the load did not break MPM using these higher soil strengths as seen in the previous section. A doubling of the load to $q_v = 100kPa$ increases the material points, which have been loaded into the residual stress stage. The deformation therefore is then comparable to the deformation of the softer soil under a smaller load. A further increase of the load to $q_v = 150kPa$ drastically increases the deformation. The residual stress region is reached for a large portion of the domain, which is comparable to the result for the weaker soil. However, the residual stress region is reached due to the surface traction in the case of the stronger soil, while it was due to gravitational loading in the case of the weaker soil. A rotational slip surface is formed as a result of this deformation. The material points directly below the load are pushed along this surface and eventually fall back on top of the surface. In reality this will most luckily not occur, due to the fact that the load must be applied by a physical object and the material points are therefore unable to fall on top of the soil surface beneath the physical object, but this once again proves that the MPM model is capable of modelling rolling material points.



(a) Surface traction $q_v = 100kPa$



(b) Surface traction $q_v = 150kPa$

Figure 6.21: Stress regions after 2.50 seconds for a soil with increased strength: $c_i = 40kPa, c_p = 20kPa, c_r = 10kPa$

Even though the MPM model did complete the modelling of the soil under a load $q_v = 150$ kPa, a mistake is clearly visible in Figure 6.22. One material point has been lifted far above the soil surface. Further research is necessary to pinpoint the origin of this problem, but the material point appears to be launched from the surface to this location. At this location the material point is fixed due to the fact that the nodes surrounding this material point do not compute any deformation. Finally, similar to the stiffness, a further increase of the strength results in smaller deformations under the same load.

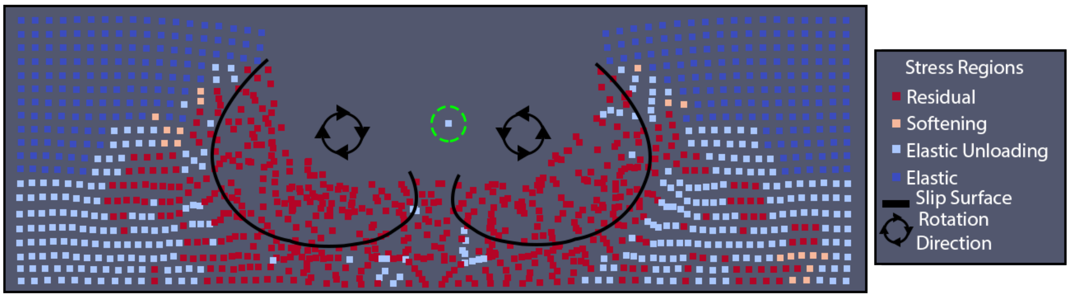
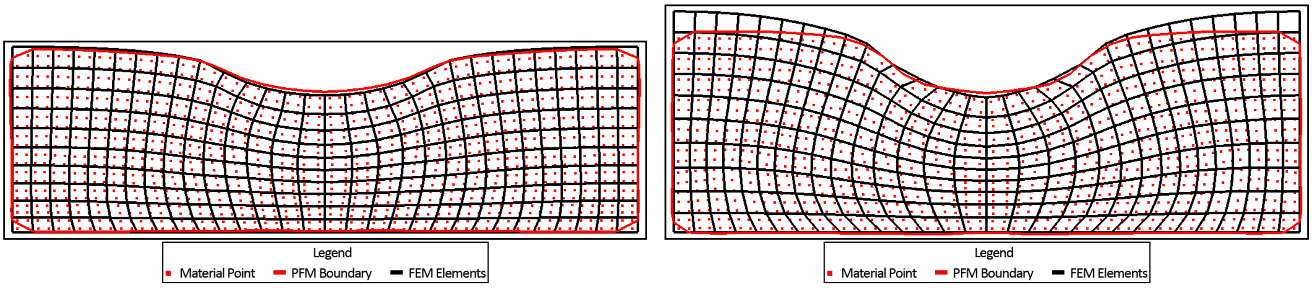


Figure 6.22: Stress regions after 6.25 seconds for a stronger soil under a surface traction $q_v = 150kPa$. Lifted material point marked by the green circle

6.2.5. Comparison to FEM

Similar to the slope stability problem in Section 5.5 a comparison has been made with FEM for the application of a surface traction. Unfortunately a comparison with Plaxis was even more difficult than the comparison made in Section 5.5. As the comparison with Plaxis did not provide any useful information these results are not provided. However, a comparison with the extracted FEM was possible and the results are shown in Figure 6.23. The FEM mesh has the same size as the background grid of MPM and uses 4-node quadrilateral elements, but the elements along the slope have distorted to follow the shape of the slope. Due to the fact that the differences in the initial condition are similar to the slope stability problem provided in Section 5.5, the initial condition has not been shown for the surface traction. The deformation is again slightly larger for FEM after 0.625 s. The PFM surface is therefore higher than the FEM surface. The differences are even larger after 1.25 s. This is mainly the result of a larger vertical expansion due to the stress oscillations at either side of the surface traction in FEM. Moreover, material points roll under the surface traction in MPM, which reduces the total vertical deformation of the surface. This process does not occur in FEM, resulting in a larger vertical deformation below the surface traction.

Even though small differences are present between the PFM and FEM boundary the general shape is consistent. This indicates that the application of the surface traction in MPM has been designed and implemented correctly.



(a) After 0.625 s

(b) After 1.25 s

Figure 6.23: Comparison between an FEM extracted from MPM and edge detection with PFM for a bearing capacity problem

6.3. Surface traction on a soil slope

The surface traction has also been applied to the failing soil slope described in Figure 5.14. The soil parameters of the slope are given in Table 6.3 and are mostly the same as the parameters used in previous tests. However, the Poisson's Ratio is increased from 0.15 to 0.33. The background grid elements are still 4-node quadrilateral elements with a size of 0.5 m by 0.5 m. Once again four material points are placed inside of each grid cell. The problem uses 1150 background grid elements. From the 1150 elements 842 elements contain material points in the initial condition resulting in 3368 material points. Multiple tests have been carried out on this slope in which the location of the surface traction is changed. These tests are compared to the baseline test, where no additional load is applied and the only load on the soil is the gravitational load. The surface traction is applied in combination with the gravitational load.

Table 6.3: Parameters used in the slope stability test under the application of a surface traction

Parameter	Symbol	Value
Young's modulus	E	10^3 (kPa)
Poisson's Ratio	ν	0.33 (-)
Volumetric weight	γ	20 (kN/m^3)
Initial Cohesion	c_i	20 (kPa)
Peak Cohesion	c_p	10 (kPa)
Residual Cohesion	c_r	5 (kPa)
Surface traction	q_v	50 (kPa)

In each case the surface traction q_v has been applied over a distance of 5 meters. The horizontal location is again fixed during a computation. However, the horizontal location of the surface traction was varied between computations. For the surface tractions located at least 5 meters behind the top of the soil slope, the slope failure is similar to the slope without an additional load. This is visualized in Figure 6.24. However, the slope is pushed forward by the horizontal displacement due to additional load. Moreover, this additional load decreases the height drop due to the failure of the slope surface, due to the fact that additional material is pushed into the location of the failure. In all cases a gentle downward slope angle can be observed from the original crest height to a height slightly below the new crest height. Due to the location of the slip surface, the soil originally located at the crest deforms less than the surrounding soil and is therefore located higher then the surrounding soil forming a new crest.

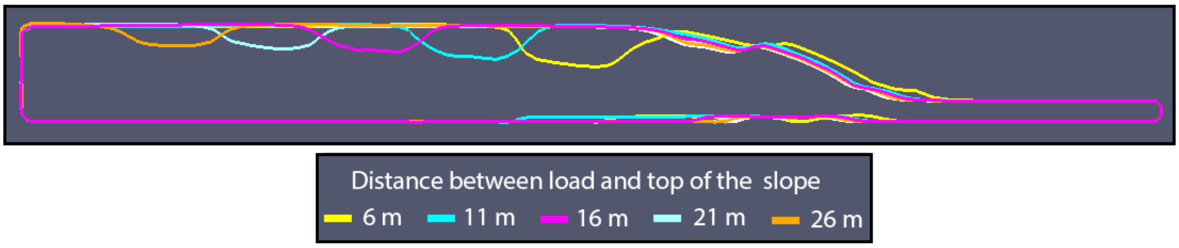


Figure 6.24: Surface deformation due to surface load behind the top of the slope

A different deformation profile occurs when the load is placed directly on top of the crest of the soil slope. In Figure 6.25 the surface traction has been placed 6 m, 1 m and 0 m behind the crest of the soil slope. The deformation profile without an additional load (shown in white) can still be recognized within the deformation profile with a load placed 6 meters behind the crest. In both cases the original slip surface is recognized, but the deformation is increased due to the additional load. However, an even closer load, for example a load 1 m or 0 m behind the crest, does no longer represent the same deformation profile. Due to the fact that the force is placed on top of the slip surface, a gentle slope is no longer present. Instead the load deforms the start of this gentle slope into an steep slope, due to the fact that the deformation of the slope failure and the additional load are combined. A small crest of a height similar to the unloaded case is still present if the load is placed 1 m behind the original crest. On the other hand a much wider crest is formed if the load is placed directly on top of the original crest. During the deformation process a small crest is still created, but the air behind the crest is then filled by the soil displaced by the load.

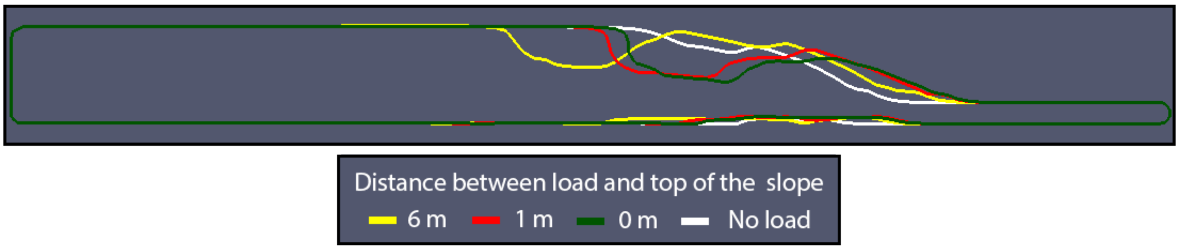


Figure 6.25: Surface deformation due to surface load close to the top of the slope

The load has finally been placed both on the slope and in front of the toe of the slope. The results of these two test are shown in Figure 6.26. Placing the load on top of the slope completely destroys the slope profile. Due to both the gravitational force and the additional surface traction the material points are even propelled through the air and land on top of the surface afterwards. This behaviour is shown in Figure 6.27. On the other hand the shape of the deformation hole due to the surface traction is still comparable to the results found previously. The main difference is the depth of the hole and the angle of the bottom of the hole. A completely different result is found by placing a load in front of the toe of the slope. Due to this load the soil slope gains stability as the deformation is decreased. This could be a feasible solution to reduce slope failure. However, the stress beneath the toe does increase, which could be a problem if the bottom boundary is not completely stiff or located further beneath the toe of the slope. Further research is therefore necessary to determine if this is a practical solution for the reduction of deformation.

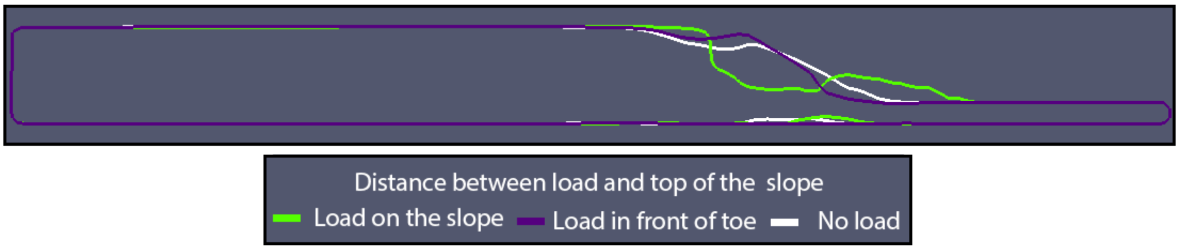


Figure 6.26: Surface deformation due to surface load on top of the slope or in front of the toe of the slope

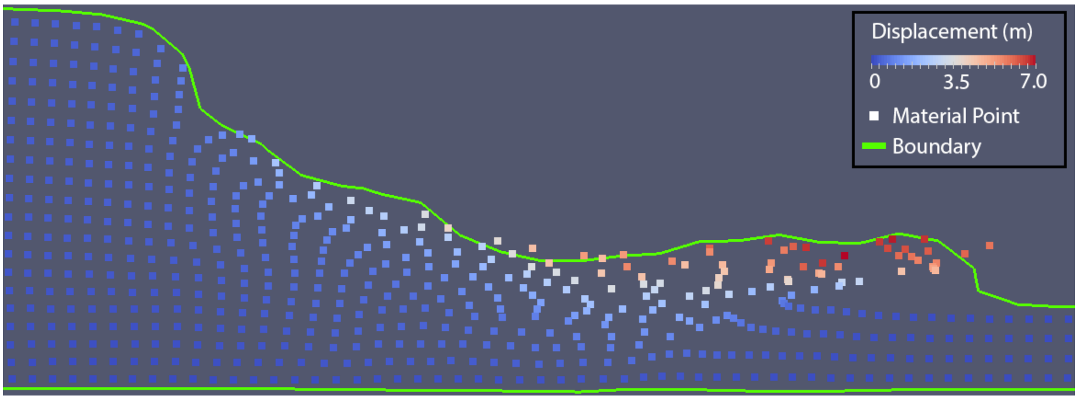


Figure 6.27: Propelled material points due to a surface load on the soil slope

The results of the previous figures are combined to visualize the effect of the location of the surface traction. As a start, the depth of the hole formed due to the surface traction is given in Figure 6.28 with respect to the location of the load. The distance between the load and the slope is a comparison between the crest of the slope and the right edge of the force. So a load with a distance of 0 m has been placed on top of the crest of the slope, while a load with a distance of 5 m has been placed on the slope itself. Negative distances represent loads to the left of the crest, while positive distances represent loads to the right of the crest. The location of the bottom of the hole has been plotted against the location of the load. The depth of the hole appears to increase quadratically in comparison to the distance to the crest. Surprisingly this correlation appears to be present even for loads applied close to or on top of the slope. The correlation is bounded by an upper and lower boundary. The lower boundary is given by the depth of the deformation hole at infinite distance from the slope. On the other hand the top boundary is given by the depth of a deformation hole by a load in front of the soil slope. Due to the low height of the soil in front of the soil slope, the soil deformation is small due to a load at this location. The depth of the hole at this location is therefore mainly given by the height of the soil slope, due to the fact that this height is not subtracted from the depth of the hole.

The additional deformation of the crest and the toe due to a surface traction in comparison to the unloaded situation is given in Figure 6.29. Only a load in front of the toe of the slope has reduced the horizontal deformation. Every other surface traction, either on top of the slope or behind the slope, increased the deformation. The deformation of the crest reaches a maximum at a force on the soil slope, while the toe received a maximum deformation with a load directly behind the crest. The additional deformation decreases once the distance increases.

Finally, the additional vertical deformation of the crest has been given in Figure 6.30. Surprisingly the new crest increases in height if a load is applied roughly 6 meters behind the crest. The load applied at this location pushed the material beside the load upwards, a behaviour also observed in the test on the rectangular soil body. A load in front of the toe decreased the crest deformation even further. On the other hand loads close to the crest should be avoided as these loads drastically decrease the height of the newly formed crest.

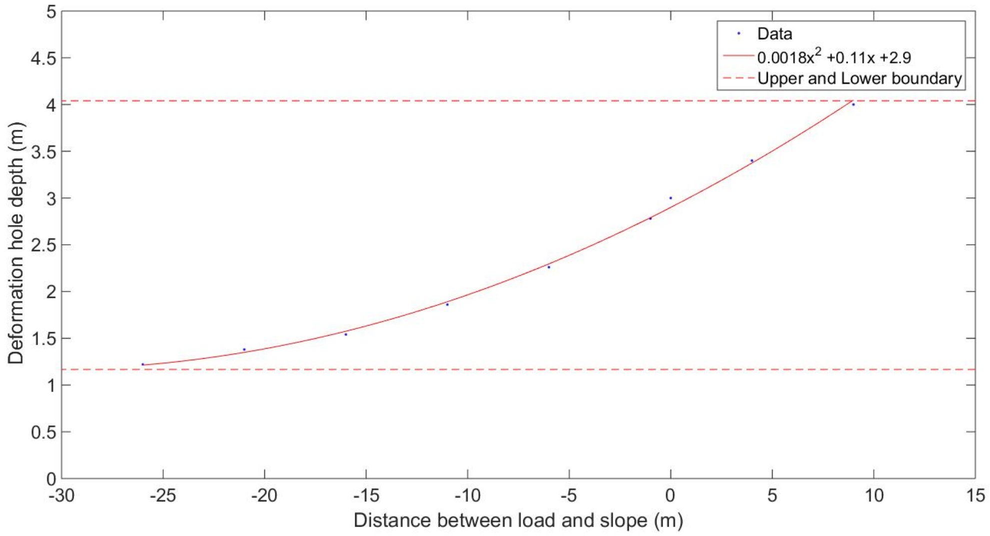


Figure 6.28: Depth of the hole formed by the surface traction

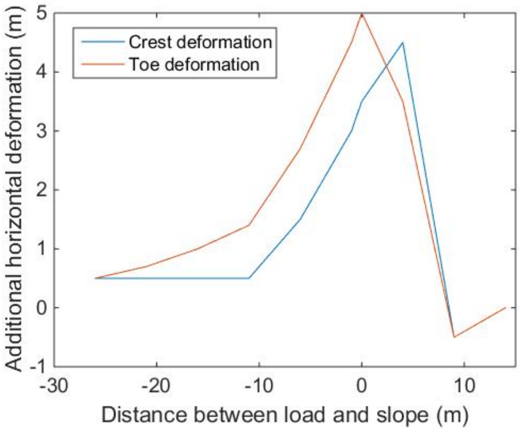


Figure 6.29: Surface deformation due to surface load behind the top of the slope

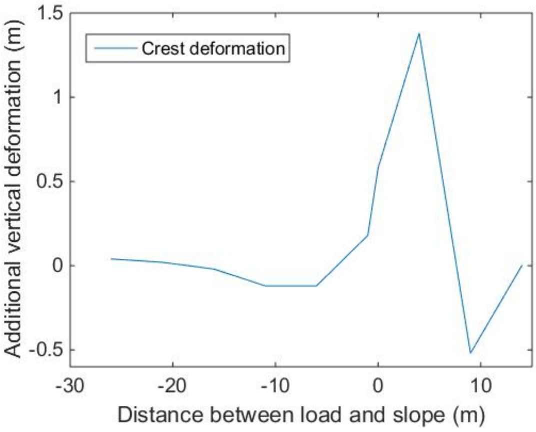


Figure 6.30: Surface deformation due to surface load behind the top of the slope

While the solutions found within this section appear to be decent a couple of problems have been observed during this test. Firstly, the material points do not always lay within the boundary drawn by PFM. On the one hand this is a result of the Gaussian filter applied to the proximity field and on the other hand the size of the proximity field grid excludes fine details. The Gaussian filter also lifted the boundary from the fixed bottom boundary as show in Figures 6.24, 6.25 and 6.26. Additionally some of the material points being launched from the problem domain into the air remain stuck due to the absence of a stiffness relationship. All these problems require further investigation to fix. Moreover, the stress and strain at the surface beneath the load is extreme in comparison to the rest of the domain. This might resemble reality but once again additional research is required on the deformation at the boundary.

6.3.1. Comparison to FEM

Once again, a comparison has been made between an FEM extracted from MPM and a normal MPM. In both cases the surface traction has been applied 1 meter behind the original location of the crest. The FEM mesh uses the same elements as the background grid, but the slope elements are deformed to match the shape of the slope. The results of the comparisons can be seen in Figure 6.31.

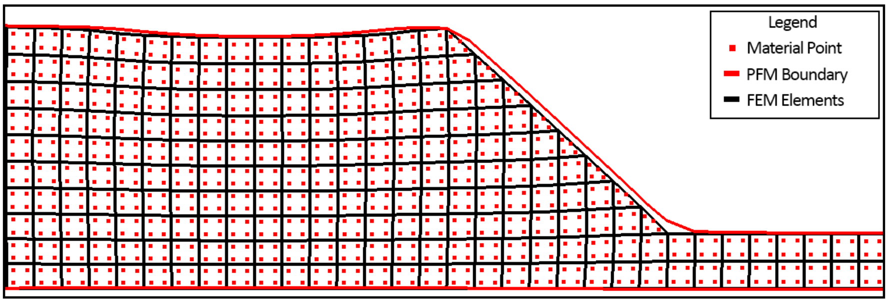
The differences between FEM and PFM are comparable to the differences found for the slope without a surface traction discussed in Section 5.5. The slope surface is not detected in the right location due to the initial location of the material points. However, due to the large deformation of the FEM slope the surfaces of FEM and PFM are at the same location after 0.625 s (see Figure 6.31b). There are a couple of observations which differ from Section 5.5, namely:

- The difference between FEM and MPM occurs faster due to the combination of the slope failure and the applied surface traction. For an unloaded slope the deformation was comparable between MPM and FEM in the 1.25 seconds, whereas the difference betweem MPM and FEM is sever for a loaded slope in the same period.
- PFM is unable to detect the gentle slopes to either side of the surface traction present in Figure 6.31a. Due to the Gaussian filter the surface is detected closer to the material points. This is effect is removed with a lower contour value. However, this does not effect every part of the surface the same. The effect of the smoothing and the lower contour value is dependent on the angle α between two parts of the surface shown in Figure 6.32. If $\alpha < 180$ the boundary is detected further from the material points. This occurs at the toe of the slope and on either side of the surface traction. A smaller α enhances the effect as can be seen at the toe. If $\alpha > 180$ the boundary is detected closer to the material points. This can be seen at the corners of Figure 6.23. In this case the effect enhances with a larger α.

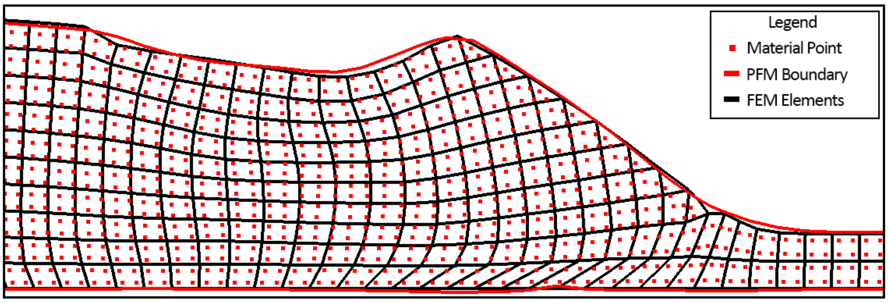
6.4. Conclusion

Within this chapter a surface traction has been applied to two MPM problems both based on a large strain relationship. The first test applied a surface traction to the top of a rectangular soil body. This surface traction deformed the top surface after which the load has been applied to the deformed boundary. According to the tests the transformation of the load from the surface to the nodes works as intended and reasonable deformation patterns occur. Unfortunately due to the MPM method currently used the stresses oscillate, resulting in large plastic strains under only the application of gravity. The addition of the load is therefore inaccurate. However, the behaviour is generally coherent to the expected behaviour, as two shear surfaces are created. The material points are also capable of rolling under the load. In comparison to FEM the same soil elements are loaded when a load is applied on the nodes, while in this case the load is applied to a specific location without specifying the soil elements on which the load should be applied. The parametric study also gave the expected results as the stronger or stiffer soils deformed less. Moreover as expected, a higher surface traction on a stronger soil was capable of achieving similar deformation as a lower traction on a softer soil.

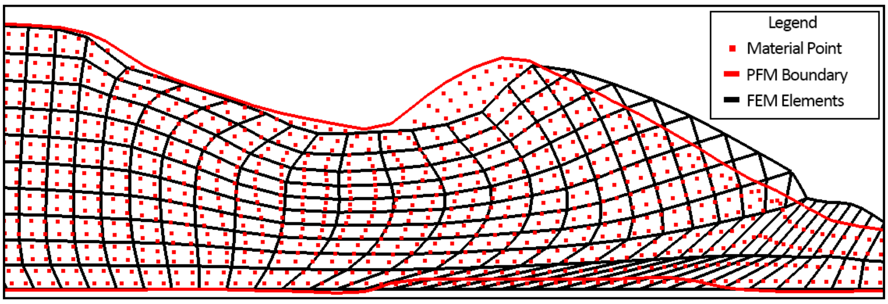
Afterwards the surface traction has been placed on a slope stability problem to answer the research question of this thesis: Can boundary conditions on moving boundaries be appropriately applied to the background mesh of an MPM solution in slope stability problems? It is once again difficult to prove the correctness of the solutions, but the behaviour obtained from the surface traction acted as expected. A load closer to the slope increased the deformation of the slope. Especially the horizontal deformation of



(a) After 0.125 s



(b) After 0.625 s



(c) After 1.25 s

Figure 6.31: Comparison between an FEM extracted from MPM and edge detection with PFM for a surface traction on a slope.

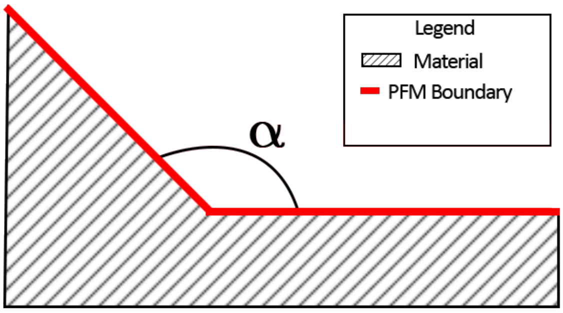


Figure 6.32: Angle of the detected surface

the soil slope was increased drastically. Moreover the slope also effects the shape of the deformation due to the load as the depth of the deformation increased with a load closer to the surface. However, the slope stability problem also showed a couple of problems. The shape of the boundary in comparison to the location of the material points is questionable under certain circumstances, due to the fact that material points can be located outside of the surface. Furthermore, extreme deformation are observed close to the surface and it is unknown whether these deformations are reasonable. Additional research is required to either solve these problems or investigate whether or not this is a representation of reality.

Moreover, the force is currently fixed to a specific horizontal location, whereas the load might move along with the deformation in reality. Therefore, this must be added to the model in the future.

A comparison between a surface traction in MPM and FEM presented similar results to the comparison made in the previous chapter. In the initial condition the differences are small, with the exception of the slope surface. At the slope surface the differences are larger, roughly 0.3 m, due to the placement of the material points. During deformation the differences between MPM and FEM increase, similar to the previous comparison. However, the differences appear faster due to the surface traction. The general shape is comparable, but the deformation of FEM occurs faster and results in larger final deformations in comparison to MPM. Moreover, an additional problem with the Gaussian filter has been observed. With the filter non-linear parts of the boundary can be detected incorrectly. Depending on the curvature the boundary is either detected further outwards or inwards.

As a final note, a fixed displacement boundary conditions is mentioned. Due to the time constraints placed on this thesis this currently has not been implemented. The theoretical background has been provided by Cortis et al. [11], but unfortunately the changes required to apply IBM together with PFM are not included. This is definitely a necessity in the future to be able to apply conditions in two phase MPM models. However, most fixed displacement boundary conditions are straight and can therefore be solved with the standard theory provided by Cortis et al. [11]. The priority of the changes to IBM was therefore smaller than the implementation of the surface traction. The implementation of a fixed displacement boundary for the explicit MPM is also not included and must be developed in further detail and implemented in future research.

7

Conclusion and recommendations for future work

In this section the two research questions of this thesis are answered with a summary of the conclusions provided in previous chapters. Additionally an outlook to future work are provided.

7.1. Concluding remarks

In this thesis a new method for the application of moving boundary conditions in MPM has been developed and implemented. As mentioned in the introduction, the main research question of this thesis was: Can boundary conditions on moving boundaries be appropriately applied to the background mesh of an MPM in slope stability problems? To answer this question and apply boundary conditions to moving boundaries the location of the boundary first had to be located. Therefore, a second question needed to be answered: Which edge detection method is most suitable for detecting a material boundary in an MPM?

From the literature review it can be concluded that the existing methods for the application of boundary conditions in MPM are either not applicable to moving boundary conditions or introduce inaccuracy due to a boundary layer. To reduce the size of the boundary layer and reduce the inaccuracy Cortis et al. [11] developed IBM. IBM is similar to the 'penalty method' used in FEM and applies the boundary conditions to the background grid of MPM. IBM has been developed for fixed boundaries not located on a grid boundary. A modification to the stiffness matrix includes the fact that the boundary does not coincide with the grid. In this thesis the concept of applying boundary conditions to the background grid has been expanded to moving boundaries. Similar to IBM an transformation of the boundary condition is required to apply it on the background grid.

A second problem regarding MPM is the unknown location of moving boundaries. To perform the transformation of the boundary condition, the location of the boundary must be known. An edge detection method for MPM has therefore been developed. Three methods, namely SMM, VOF and PFM, have been tested. As mentioned within Chapter 4 PFM appears to be the most suitable for MPM. PFM is a new version of the Level Set Method, which uses composite Bézier curve to define the boundary. In PFM a proximity field is computed based on kernel functions around each material point. A summation of each of these kernel functions at the nodes of a grid results in the proximity field. Based on the proximity at the nodes a contour can be drawn at a specified contour level. From this contour a composite Bézier curve can be constructed, which represents the boundary. Deformation of the material around the material points can be modelled by a deformation of the kernel functions. The deformation of the functions is based on the strains of the material point.

From the tested methods, PFM was the only method capable of reproducing the boundary at larger strains and was also capable of detecting the surface after rolling and tearing of the material. SMM is more accurate at small strains, but is unable to represent the boundary at larger strains. The accuracy of VOF was too low for the detection of a boundary and large deformations only decreased the accuracy. Therefore, both SMM and VOF proved not applicable for edge detection in MPM. On the other hand, PFM gives a reasonable prediction of the boundary based on the information provided by the material points and the detected boundary has been successfully used to apply a surface traction on a failing soil slope.

So, it can be concluded that PFM is the most suitable edge detection method for a material boundary in MPM. However, after the initial implementation of PFM, the method was not capable of applying boundary conditions in MPM, due to a large computation cost, a limited stability and a limited accuracy. Therefore, improvements have been made to PFM.

At first, the high computation cost of PFM has been tackled. The computation cost has been reduced by a factor of 30 to roughly the computation cost of an explicit MPM time step. As the method preferable needs to run during every time step, the computation cost of the total program doubles when an explicit MPM is used. This reduction has been achieved by using shape functions which approximate the proximity field, during the contour algorithm. A better order of the calculation of the proximity of nodes during each timestep provided reduction of the computation cost. However, even with these reductions PFM has a considerable computation time in comparison to the explicit MPM. Therefore, PFM might be more suitable for an implicit MPM as it uses larger and more intensive time steps, which reduces the impact of the addition of PFM on the implicit MPM.

Secondly, the stability and accuracy of PFM has been increased and the model is now capable of modelling any deformation problem as long as the material points are still inside the proximity field grid. To improve stability and accuracy, unrealistic holes in the proximity field have been removed. This is preferably done by equally spacing the material points in the background grid. Additional testing is required to observe the effects of the equal spacing. However, as of now the results are comparable to material points located at the Gaussian integration points and have therefore been accepted. Equally spacing material points also presents a smoother surface as material is no longer concentrated at certain locations, which is the case for material points located at the integration points. The shape of the kernel function is an ellipse, which introduces unrealistic roughness to the boundary. Concentrated material increases this unrealistic roughness. Therefore, spacing material points equally is preferred. Further reduction of the roughness is achieved with a Gaussian smoothing filter. This unfortunately also smooths sharp corners, which are supposed to be sharp. Therefore, a filter should be used with care. Due to the increase in accuracy and stability, it is advised to both use a Gaussian filter and space the material points equally. Finally, to improve the stability an improved contour algorithm has been developed.

After the removal of the previously mentioned problems, PFM was suitable for the application of boundary conditions in MPM. Therefore, PFM has been used to apply a surface traction on a slope stability problem to answer the main research question. The surface traction has also been used to model a bearing capacity problem to provide additional information required to answer the research question.

The effects of a surface traction modelled with MPM on the soil slope seem intuitive as it increases the plasticity and deformations. As expected, the deformation of the slope increases as a vertical surface traction is placed closer to the top of the slope and reaches a maximum when the load is placed on top of the slope itself. The applied load is distributed correctly to the background nodes and the deformation according to this load appears reasonable.

Furthermore, the bearing capacity problem seems reasonable. A parametric study has been performed, in which the strength, stiffness and Poisson's ratio have been varied. As expected, a higher strength, stiffness or Poisson's ratio reduces the deformation. Moreover, the influence of the load in comparison to the gravitational load increases under stronger and stiffer soil conditions. Finally, a larger load has been applied on the stronger soils and a larger deformation occurred as expected.

The detected boundary for both the bearing capacity problem as well as the slope stability problem have been compared with FEM. During the first couple of time steps the comparison between PFM and FEM is decent. The difference on most of the boundary is only 0.05 m and can be further reduced by an optimization of the contour level. However, the detection of the slope surface is inaccurate, but this is the result of the placement of the material points and is not caused by PFM.

The high loads in the bearing capacity problem revealed one of the problems regarding PFM. The method assumes a constant strain over the material influenced by the material point. This assumption is decent until the size of the kernel becomes too large. Under the high load conditions observed in the bearing capacity problem, the length of some of the kernel functions approached the size of the entire soil domain. To remove these unreasonably long and thin kernel functions, a maximum deformation of the kernels has been included. This also improved the stability of PFM, due to the fact that PFM cannot find a boundary if a kernel function expands outside of the domain.

To summarize, PFM has been used within this thesis to detect the boundary of a soil domain. From the three tested methods PFM was the most applicable edge detection method for MPM. A surface traction has been applied to the background grid based on the detected boundary and reasonable deformations have been observed. While it is difficult to prove the correctness of the solution, PFM has been used to appropriately apply a surface traction on a moving boundary. Therefore, boundary conditions on moving boundaries can be appropriately applied to the background mesh of an MPM in slope stability problems.

7.2. Recommendations for future work

Applying boundary conditions to the background grid in MPM has been shown to be appropriate. At the same time PFM has been shown to be a useful edge detection method in MPM and a useful tool to apply moving boundary conditions. However, research has only just started on this topic and PFM has been developed in this thesis. Therefore, additional research is required and a few recommendations for future research are as follows:

- The accuracy of PFM is questionable. Therefore, additional testing is required on the accuracy of PFM. PFM should be implemented on a new version of MPM, which initializes the stresses before deformation. With a better initial stress the comparison between MPM and FEM will be easier as most FEM use an initial stress. Perhaps, PFM can also be compared to an analytical solution to further define the accuracy of PFM. The effect of a Gaussian filter on the accuracy should also be investigated.

- PFM is based on a field defined on a mesh. Therefore, mesh dependency is present in PFM. MPM attempts to remove mesh dependency and a meshless method might be more applicable. Perhaps the proximity field can be defined on the material points as well. This could remove the requirement of a mesh for PFM and remove the mesh dependency. However, it is currently unclear if this is a possibility.

- PFM has been applied to an unfinished Implicit MPM with a reasonable initial stress condition. However, due to the fact that this program was currently unfinished, it was unusable for the testing performed in this thesis. Therefore, PFM should be implemented further in an Implicit MPM. Using PFM with an implicit program reduces the computational cost influence of PFM in comparison to the cost of MPM.

- Due to a lack of time only a surface traction has been applied on a soil stability problem and a bearing capacity problem. The theory given in Chapter 3 on the application of fixed primary variables should be implemented in the future. This has partly been done by Cortis et al, but IBM should be expanded to moving boundaries as well (Cortis et al. [11]).

- A two phase MPM has not been used in this thesis, whereas one of the application areas is rainfall or pore pressure induced failure (Wang et al. [36]). It is therefore necessary to also apply the concepts described in this thesis to two phase materials. The boundary would still be located based on the location of the material points, but the pore pressure condition is applied at this boundary to the fluid phase.

- The boundary is currently described by composite Bézier curves and the application of a surface traction has only been tested using linear Bézier curves (first order). It is currently unclear whether composite Bézier Curves are the ideal representation of the boundary, as a small deviation of one control point can effect the shape of the complete boundary. Moreover, the error in the distribution of a surface traction associated with a higher order curve is not negligible. A different curve, for example a B-spline, might therefore be more applicable to describe the boundary. Trying different curves is a subject of future research.

A

Implicit Boundary Method

In this appendix a further explanation of the constructions of the matrices used in the Implicit Boundary Method (IBM) is given. The appendix is on currently unpublished information provided by Cortis (Cortis et al. [11]). A 2D version of IBM is shown in this appendix. However, 3D versions are also available.

The element stiffness matrix $[\mathbf{K_e}]$ is still defined by equation (3.2):

$$[\mathbf{K_e}] = \int_{V_e} [\mathbf{B}]^T [\mathbf{C}][\mathbf{B}] dV \tag{A.1}$$

However, the shape function matrix $[\mathbf{B}]$ is redefined to also include the boundary conditions:

$$[\mathbf{B}] = \begin{bmatrix} D_1 \frac{\partial N_j}{\partial x_1} + N_j \frac{\partial D_1}{\partial x_1} & 0 \\ 0 & D_2 \frac{\partial N_j}{\partial x_2} + N_j \frac{\partial D_2}{\partial x_2} \\ D_1 \frac{\partial N_j}{\partial x_2} + N_j \frac{\partial D_1}{\partial x_2} & D_2 \frac{\partial N_j}{\partial x_1} + N_j \frac{\partial D_2}{\partial x_1} \end{bmatrix} \tag{A.2}$$

where D_1 and D_2 are the Dirichlet functions which apply the boundary conditions. According to Figure A.1 the boundary is applied to a thin band δ. The Dirichlet function associated with this thin band depends on the applied condition. Two typical Dirichlet functions are defined as

$$D = \frac{2n}{\delta} - \frac{n^2}{\delta} \tag{A.3}$$

to apply a fixed variable boundary conditions, or

$$D = 1 \tag{A.4}$$

to allow free motion of the boundary. The boundary layer δ must be small and according to Burla and Kumar [6] a thickness of $\delta \approx 10^{-5}$ is appropriate.

Combining equations (A.1) and (A.2) will result in the following expansion of the stiffness relationship:

$$[\mathbf{K_e}] = \int_{V_e} [\mathbf{B_1}]^T [\mathbf{C}][\mathbf{B_1}] + \int_{V_e} [\mathbf{B_1}]^T [\mathbf{C}][\mathbf{B_2}] + \int_{V_e} [\mathbf{B_2}]^T [\mathbf{C}][\mathbf{B_1}] + \int_{V_e} [\mathbf{B_2}]^T [\mathbf{C}][\mathbf{B_2}] \tag{A.5}$$

The individual parts of equation A.5 can be calculated separately forming a modified stiffness matrix:

$$[\mathbf{K_e}] = [\mathbf{K_1}] + ([\mathbf{K_2}] + [\mathbf{K_2}]^T) + [\mathbf{K_3}] \tag{A.6}$$

where $[\mathbf{K_1}]$ is the standard local stiffness matrix and $[\mathbf{K_2}]$ and $[\mathbf{K_3}]$ contain the modification for cells containing an edge. $[\mathbf{K_2}]$ and $[\mathbf{K_3}]$ can be computed along the boundary due to the fact that they contain derivatives of the Dirichlet functions. $[\mathbf{K_2}]$ and $[\mathbf{K_3}]$ are given in equations (A.7) and (A.8) respectively. The stiffness matrix $[\mathbf{K_e}]$ is unchanged for cells not containing an edge.

$$[\mathbf{K_2}] = \int_{\Gamma} [\mathbf{\bar{B}_1}]^T \Big(\int_0^{\delta} [\mathbf{\bar{D}_1}]^T [\mathbf{C}][\mathbf{\bar{D}_2}][\mathbf{J}] dn \Big) [\mathbf{\bar{B}_2}] dt \tag{A.7}$$

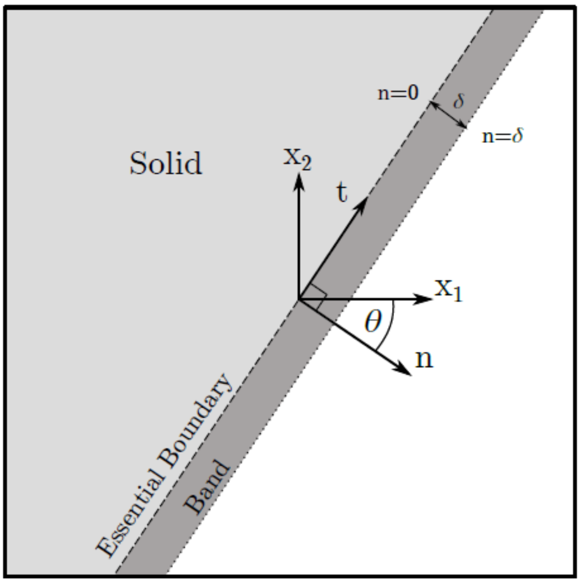


Figure A.1: An inclined boundary showing the thin band on which a essential condition is applied (Cortis et al. [10])

$$[\mathbf{K_3}] = \int_{\Gamma} [\bar{\mathbf{B}}_\mathbf{2}]^T \Big(\int_0^{\delta} [\mathbf{J}]^T [\bar{\mathbf{D}}_\mathbf{2}]^T [\mathbf{C}][\bar{\mathbf{D}}_\mathbf{2}][\mathbf{J}] dn \Big) [\bar{\mathbf{B}}_\mathbf{2}] dt \tag{A.8}$$

where $[\mathbf{J}]$ is the Jacobian transformation matrix between boundary coordinates and global coordinates. The other matrices of equations (A.7) and (A.8) are given below:

$$[\bar{\mathbf{B}}_\mathbf{1j}] = \begin{bmatrix} \frac{\partial N_j}{\partial x_1} & 0 \\ 0 & \frac{\partial N_j}{\partial x_2} \\ \frac{\partial N_j}{\partial x_2} & 0 \\ 0 & \frac{\partial N_j}{\partial x_1} \end{bmatrix} \tag{A.9}$$

$$[\bar{\mathbf{B}}_\mathbf{2j}] = \begin{bmatrix} N_j & 0 \\ 0 & N_j \end{bmatrix} \tag{A.10}$$

$$[\bar{\mathbf{D}}_\mathbf{1}] = \begin{bmatrix} D_1(n) & 0 & 0 & 0 \\ 0 & D_2(n) & 0 & 0 \\ 0 & 0 & D_1(n) & D_2(n) \end{bmatrix} \tag{A.11}$$

$$[\bar{\mathbf{D}}_\mathbf{1}] = \begin{bmatrix} \frac{\partial D_1(n)}{\partial n} & 0 \\ 0 & \frac{\partial D_2(n)}{\partial t} \\ \frac{\partial D_1(n)}{\partial t} & \frac{\partial D_1(n)}{\partial n} \end{bmatrix} \tag{A.12}$$

Bibliography

[1] F.A. Allahdadi, T.C. Carney, J.R. Hipp, L.D. Libersky, and A.G. Petschek. High strain lagrangian hydrodynamics: A three dimensional sph code for dynamic material response. *Journal of Computational Physics*, 109:67–75, 1993.

[2] S.G. Bardenhagen and E.M. Kober. The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–495, 2004.

[3] S.G. Bardenhagen, J.U. Brackbill, and D. Sulsky. The material-point method for granular materials. *Comput. Methods Appl. Mech. Engrg*, 187:529–541, 2000.

[4] K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, Pearson Education, Watertown, second edition, 2014.

[5] L. Beuth. Formulation and application of a quasi-static material point method. Report, Institut für Geotechnik der Universität Stuttgart, 2012.

[6] R.K. Burla and A. V. Kumar. Implicit boundary method for analysis using uniform b-spline basis and structured grid. *International Journal for Numerical Methods in Engineering*, 76:1993–2028, 2008.

[7] J. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–697, 1986.

[8] Z. Chen, W. Hu, L. Shen, X. Xin, and R. Brannon. An evaluation of the mpm for simulating dynamic failure with damage diffusion. *Engineering Fracture Mechanics*, 69:1873–1890, 2002.

[9] F. Cheng. 15 curves and surfaces, 2016. URL `http://www.cs.uky.edu/~cheng/cs535/Notes/CS535-Curves-1.pdf`.

[10] M. Cortis, W.M. Coombs, and C.E. Augarde. Implicit essential boundaries in the material point method. In *UK Conference of the Association for Computational Mechanics in Engineering*, page 4. Cardiff University.

[11] M. Cortis, W.M. Coombs, C. E. Augarde, S. Robinson, M. Brown, and A. Brennan. Modelling seabed ploughing using the material point method. *Procedia Engineering*, 00:7, 2016.

[12] P. Cundall. Distinct element models of rock and soil structure. *Analytical and Computational Methods in Engineering Rock Mechanics*, pages 129–163, 1987.

[13] P. Delmas. Gaussian filtering, 2016. URL `https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf`.

[14] R. Duraiswami. Gaussian quadratures, 2016. URL `http://www.umiacs.umd.edu/~ramani/cmsc460/Lecture16_integration.pdf`.

[15] D. Eno. Bezier curves and splines, 2006. URL `http://www.oswego.edu/Documents/mathematics/Bezier%20Curves%20and%20Splines.ppt`.

[16] J.M. van Esch, D. Stolle, and I. Jassim. Finite element method for coupled dynamic flow-deformation simulation, 2011. URL `publications.deltares.nl/EP2225.pdf`.

[17] C. A. Felippa. *Introduction to finite element methods*. Department of Aerospace Engineering Sciences and Center for Aerospace Structures, Boulder, Colorado 80309-0429, USA, 2004.

[18] L.V. Kantorovich and V.I. Krylov. *Approximate Methods of Higher Analysis*. Interscience Publishers, 3th edition, 1958.

[19] M.B. Liu, G.R. Liu, and K.Y. Lam. Investigations into water mitigation using a meshless particle method. *Shock Waves*, 12:181–195, 2002.

[20] A.N. Lowan, N. Davids, and A. Levenson. Table of the zeros of the legendre polynomials of order 1-16 and the weight coefficients for gauss' mechanical quadrature, 1941. URL http://www.ams.org/journals/bull/1942-48-10/S0002-9904-1942-07771-8/S0002-9904-1942-07771-8.pdf.

[21] S. Ma, X. Zhang, and X.M. Qiu. Comparison study of mpm and sph in modeling hypervelocity impact problems. *International Journal of Impact Engineering*, 36:272–282, 2009.

[22] J.J. Monaghan. Simulation free surface flows with sph. *Journal of Computational Physics*, 110:399–406, 1994.

[23] J.P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33:333–353, 2000.

[24] P. Pathmanathan. The finite element method, 2012. URL http://www.cs.ox.ac.uk/people/pras.pathmanathan/umd_lecture3.pdf.

[25] N.T.V. Phuong, A.F. van Tol, A. Rohe, and A.S.K. Elkadi. *Modelling of pile installation using the material point method*. CRC Press/Balkema, Leiden, The Netherlands, 2014. URL https://books.google.nl/books?hl=nl&lr=&id=31DvBQAAQBAJ&oi=fnd&pg=PA253&dq=on+the+application+of+the+material+point+method+for+offshore+foundations&ots=BoMe5wj-Kt&sig=Iu3Sbf8hblz5sTTi-PHwipwGASs#v=onepage&q&f=false.

[26] M. Pound and S. Riley. Canny edge detector, 2015. URL https://www.youtube.com/watch?v=sRFM5IEqR2w.

[27] P.W. Randlesa and L.D. Liberskyb. Smoothed particle hydrodynamics: Some recent improvements and applications. *Comput. Methods Appl. Mech. Engrg*, 139:375–408, 1996.

[28] J.A. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica*, 5:309–395, 1996.

[29] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving intefaces in Computational Geometry Fluid Mechanics, Computer Vision and Material Science*. Cambridge University Press, Cambridge, 1999. URL https://books.google.nl/books?hl=nl&lr=&id=ErpOoynE4dIC&oi=fnd&pg=PR1&dq=Sethian+JA.+Level+Set+Methods:+Evolving+Interfaces+in+Geometry,+Fluid+Mechanics,+Computer+Vision,+and+Materials+Science.+Cambridge+University+Press:+Cambridge,+1996&ots=N_igB2TXM6&sig=VxSsrF4Fy34OSwqRygf1VNasO84#v=onepage&q&f=falseandhttps://math.berkeley.edu/~sethian/2006/Publications/Book/2006/OnLine/page.037.html.

[30] C.K. Shene. Derivatives of a bézier curve, 2011. URL https://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/Bezier/bezier-der.html.

[31] I.M. Smith, D.V. Griffiths, and L. Margetts. *Programming the Finite Element Method*. John Wiley and Sons Ltd., Chichester, West Sussex, PO19 8SQ, United Kingdom, 5th edition, 2014.

[32] Y. Soh, Y. Hae, A. Mehmood, R. H. Ashraf, and I. Kim. Performance evaluation of various functions for kernel density estimation. *Open Journal of Applied Sciences*, 3:58–64, 2013.

[33] D. Sulsky and H.L. Schreyer. Axisymmetric form of the material point method with applications to upsetting and taylor impact problems. *Computer methods in applied mechanics and engineering*, 139:409–429, 1996.

[34] D. Sulsky, Z. Chen, and H.L. Schreyer. A particle method for history-dependent materials. *Computer methods in applied mechanics and engineering*, 118:179–196, 1994.

[35] P.J. Vardon, B. Wang, and M.A. Hicks. Slope failure simulations with mpm. *Procedia Engineering*, 175:258–264, 2017.

[36] B. Wang, P.J. Vardon, and M.A. Hicks. Preliminary analysis of rainfall-induced slope failures using the material point method. In S. Aversa, L. Cascini, L. Picarelli, and C. Scavia, editors, *12th International Symposium on Landslides*.

[37] B. Wang, M.A. Hicks, and P.J. Vardon. *Slope failure analysis using the Material Point Method*. Thesis, 2016.

[38] B. Wang, P.J. Vardon, and M.A. Hicks. Investigation of retrogressive and progressive slope failure mechanisms using the material point method. *Computers and Geotechnics*, 78:88–98, 2016.

[39] H. Wang, J. Kearney, and K. Atkinson. Arc-length parameterized spline curves for real-time simulation. Report, Department of Computer Science - The University of Iowa, 2003.

[40] L. Wang, W.M. Coombs, C.E. Augarde, M. Brown, J. Knappett, A. Brennan, D. Richards, and A. Blake. Modelling screwpile installation using the mpm. *Procedia Engineering*, 00:8, 2016.

[41] O.C. Zienkiewicz, C.T. Chang, and P. Bettess. Drained, undrained, consolidating and dynamic behaviour assumptions in soils. *Geotechnique*, 30(4):385–395, 1980.

[42] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Butterworth-Heinemann, Oxford, 6th edition, 2005.

[43] O.C. Zienkiewicz, R.L. Taylor, and D.D. Fox. *The Finite Element Method for Solid and Structural mechanics*. Elsevier Butterworth-Heinemann, Oxford, 2014.