Development and preliminary evaluation of the main features of the Particle Finite Element Method (PFEM) for solid mechanics

Vasilis Papakrivopoulos

TU Delft Delft University of Technology

Challenge the future

Development and preliminary evaluation of the main features of the Particle Finite Element Method (PFEM) for solid mechanics

by

Vasilis Papakrivopoulos

to obtain the degree of **Master of Science**
at the Delft University of Technology,
to be defended publicly on Monday December 10, 2018 at 2:00 PM.

Student number:	4632567	
Project duration:	February 15, 2018 – December 10, 2018	
Thesis committee:	Dr. P.J. Vardon,	TU Delft (chairman)
	Prof. dr. M.A. Hicks,	TU Delft
	Dr. F. Pisanò,	TU Delft
	J.L. Gonzalez Acosta, MSc,	TU Delft (daily supervisor)

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

TU Delft Delft University of Technology

PREFACE

This work concludes my academic career in the Technical University of Delft and marks the end of my stay in this beautiful town. During the two-year period of my master studies I managed to obtain experiences and strengthen the theoretical background in the civil engineering field that was founded through my studies in the National Technical University of Athens. I was given the opportunity to come across various interesting and challenging topics, with this current project being the highlight of this course, and I was also able to slowly but steadily integrate into the Dutch society. In retrospect, I feel that I have made the right choice both personally and career-wise when I decided to move here and study in TU Delft.

At this point, I would like to thank the graduation committee of my master thesis. First of all, Dr. Phil Vardon for trusting me with undertaking such a challenging topic, leaving me the liberty to discover the vast field of numerical modelling and guiding me through the course of the project, when needed. Also, I would like to express my gratitude to Professor Michael Hicks and Dr. Federico Pisanò for their productive feedback during the meetings we had. Of course, I could not neglect to thank Leon Acosta, who, as my daily supervisor, would stop whatever he was doing when I walked into his office and discuss with me for as long as it was required; his calmness during feedback and brainstorming has helped me a lot in times when things were not going so well. Finally, last but not least, I would like to thank Guido Remmerswaal for his useful input and conversations, which were not strictly limited in technical matters.

Finally, I could not put into words how much I appreciate the support I have enjoyed during these last two years from people in my life, whether they have been physically present or not. Besides my close friends, both old and new ones, I want to thank here my family; my parents and my brother, Giannis, have always acted as cornerstones and role models in my life, unconditionally, through thick and thin.

Vasilis Papakrivopoulos
Delft, December 2018

ABSTRACT

The Particle Finite Element Method is a numerical tool that has been introduced more than a decade ago for the solution of engineering problems involving large deformations. The method falls under the category of mesh-based particle methods, meaning that all information is stored on moving particles that represent the domain under analysis and a computational mesh is used for the solution of the governing equations. Although the method was initially developed for simulating fluid-structure interaction problems, owing to its versatility in handling large deformation and constant changes in domain boundaries and contact interfaces, it has been recently employed for solid mechanics applications. However, the lack of a consistent framework for this kind of problems has lead to different implementations of the method presented in the literature, each with its own special features. The main objective of this thesis is to implement a variation of the Particle Finite Element Method and investigate the efficiency of the different features available in literature.

Initially, the meshing procedure of the method was developed, which consists of a Delaunay triangulation for assessing the connectivity of the particles and the α-shape method for detection of the boundaries of the different domains. This was followed by an investigation on the influence of the related parameter α_{lim} on the outcome of the analysis; it appears that this choice has an impact on the results, in terms of the recovered domain volumes and the simulation response; this parameter has to be selected with care, with respect to the nature of the examined problem. Volume variations are also observed, caused by element deletion and/or addition during remeshing, which, eventually, lead to mass oscillations. These effects can be mitigated by either adjusting the value of the α_{lim} parameter, refining the particle distribution or prescribing the boundary surface during remeshing, by using a constrained Delaunay triangulation.

Another important feature of the PFEM is the treatment of contact, which is, typically, done in literature via employment of an interface mesh. This mesh is generated during remeshing, using the same scheme as for the regular domain meshes, i.e. a Delaunay triangulation and the α-shape method, and the generated contact elements are then used to enforce the contact constraints, with a variety of methods. In this work, a simple algorithm that disallows inter-penetration and allows free separation and free movement perpendicular to the contact surfaces was formulated and validated against benchmark solid mechanics problems. The automatic contact detection and interface mesh generation allows for the incorporation of more advanced contact treatment schemes.

Transference of information between successive meshes is important in PFEM for solid mechanics, especially when the history of elemental variables, e.g. stresses, is required for capturing the solid material behavior accurately. The most popular technique is the nodal smoothing technique, where the values are mapped back and forth between the integration points and the particles at each time step; other schemes have been also

presented in literature. This scheme has been shown to introduce some smoothing of information, which can be reduced by refining the particle distribution and, in general, does not seem to affect the overall system response significantly.

The developed method was, finally, compared with the available in-house implicit Material Point Method code, which shares the same formulation, on some benchmark quasi-static and dynamic solid mechanics problems. The PFEM demonstrates a more stable behavior in terms of capturing the evolution of stresses and kinematic variables, despite some inaccuracies caused by the smoothing of information and the use of simple, constant-strain triangles. On the other hand, the MPM -in its standard form- exhibits some instabilities in the assembly of equations and stress recovery, which is intensified when cell-crossing occurs, i.e. jumping of material points between elements. Regarding the computational cost of the two methods, the MPM seems to be faster and require less computer memory for the same number of information points, i.e. particles, with the simulation times, however, increasing exponentially with the number of degrees of freedom.

CONTENTS

LIST OF FIGURES

1

INTRODUCTION

1.1. PROBLEM CONTEXT

The Finite Element Method (FEM) is being used in most engineering fields for the numerical simulation of any kind of physical problem. However, there are inherent limitations in this powerful tool, which emerge when dealing with large deformations, as the high geometric non-linearity tends to compromise the accuracy and efficiency of the method. The main issues occur from the change in the problem geometry. The primary cause of concern is mesh tangling, which leads to loss of accuracy or numerical breakdown of the solution, due to a possibly infinite rate of deformation. A second geometry related issue is the constant change of the domain boundaries, even from one computational step to the next. This issue is usually disregarded in practice, with the same boundaries being maintained throughout the whole simulation, even for large deformations.

Although a great deal of geotechnical structures is subjected to deformations within the "small deformation range", there are several situations where the deformations are large enough to alter the problem geometry significantly. Some examples are slope failures, landslides, debris flows, and object penetration in the ground. Especially in material flow situations, knowledge of the rapid velocities and long run-out distances involved is essential for performing a risk assessment, especially in cases where the earth structure is situated close to an inhabited area – a typical scenario in the Netherlands. In common practice, most slope stability designs are limited in predicting the initiation of failure (e.g. limit equilibrium methods or strength reduction factor method in conventional FEM), ignoring the post-failure behavior of the structure, which is also crucial in evaluating the outcome of such a catastrophic event.

In this context, a need arises for the development of a tool capable of operating beyond the small deformation domain and efficiently predicting both failure initiation and capturing post-failure behavior of any given geotechnical structure, particularly slopes. To this end, several numerical simulation methods have been employed, including mesh-based (adaptations of the conventional FEM; Arbitrary Lagrangian Eulerian - [39], Cou-

pled Eulerian-Lagrangian - [20]) and particle techniques (Smoothed Particle Hydrodynamics - [33], Material Point Method - [54], Element Free Galerking methods - [4], Particle Finite Element Method - [47]), all demonstrating both advantages and shortcomings. Among these, the mesh-based particle methods, the MPM and the PFEM, recently developed, have gained popularity for the simulation of large deformation problems.

The latter, developed initially for fluid-structure interaction, has been recently used in solid mechanics applications [11, 43, 45], including slope failure simulations [65, 70], and is proving to be a promising tool for the future of numerical simulations of physical problems involving large deformations. The fundamental principles of the method, which are the storage of information on particles that are allowed to move freely and the constant update of their connectivity by means of frequent remeshing, have been the basis for several implementations in literature, each with different specifications, customized to the needs of the simulated problem. The last part refers to the various approaches used by researchers to tackle the special features of the method, which include, among others, the treatment of contact and the handling of internal information [11, 35]. The promising results published so far and the increasing popularity of the method, combined with its proved versatility, make the Particle Finite Element Method an attractive tool for further consideration.

1.2. PROBLEM STATEMENT

The Particle Finite Element Method described previously is a new tool, lately adopted for solid mechanics by various research teams. It is based on the Finite Element Method, with certain additional features that allow it to deal with large deformation problems. Despite several research teams exploring the capabilities of the method for solid (and soil) mechanics applications, there is inadequate documentation and published investigation on the specific features of the method, that differentiate it from the variation for fluid-structure interaction applications.

In this context, the main objective of this thesis is to *develop an implementation of the Particle Finite Element Method and investigate its special characteristics for simulation of, initially, solid and, potentially, soil mechanics applications.*

Based on the above, the research questions of this project are formulated as follows:

- How is the domain discretization, including particle connectivity and domain boundaries, updated to handle large deformation problems in PFEM and how is the outcome of the analysis affected?
- How is contact treated in the Particle Finite Element Method and how can such an algorithm be implemented?
- How is internal information handled in the Particle Finite Element Method and what is the impact on the results?
- How is mass conservation affected by the frequent remeshing employed in PFEM?
- How does the Particle Finite Element Method compare with other recently developed numerical tools, like the Material Point Method, on benchmark solid mechanics problems?

1.3. REPORT STRUCTURE

The focus of the present thesis project is to implement the Particle Finite Element Method for solid mechanics and provide a preliminary evaluation of its various features.

In chapter 2, an overview of the background of numerical tools for large deformation problems is given, with emphasis on the Particle Finite Element Method. The first part includes a brief literature review on the various numerical tools in literature. In the second part, the basic principles of the PFEM are presented, along with some published applications. Finally, the derivation of the governing equations of the tool is given, based on standard FEM procedures and similar to the MPM code already available in the Geo-engineering section of TU Delft [27, 57].

Chapter 3 involves a thorough presentation of the tools used to arrive at a discretization of high quality, on which the governing equations are solved, at each time step. Emphasis is put on the schemes used for the triangulation of the set of points, the identification of the boundaries and the dynamic evolution of the number and position of particles in the domain, in order to improve accuracy. The investigation on those aspects is based partially on literature review and simulations with the developed code.

The main features of the method, besides the remeshing procedure, are examined in chapter 4. In the first part, different ways of imposing contact treatment in literature are presented, followed by the derivation and a brief validation of the implementation adopted in this work. Moving on, the various schemes used for transferring information between old and new discretizations are presented, with some of them being further examined. The chapter closes with a discussion on mass conservation issues caused by the frequent remeshing in PFEM and proposed mitigation measures.

Chapter 5 involves a comparison between the Particle Finite Element Method with the Material Point Method. After a brief description of the standard Material Point Method procedures and the in-house MPM code, the two methods are compared on benchmark solid mechanics problems. The comparison is carried out in terms of the yielded results, e.g. developed stress state, energy conservation and the required computational resources, e.g. memory and time.

A summary of the work, conclusions on the main findings of the project and recommendations for future research are included in chapter 6.

2

BACKGROUND AND FORMULATION

In this chapter, a general overview of numerical tools for dealing with large deformation problems in engineering is presented. The first part encloses a brief literature review on the various schemes developed throughout the years, through classification of the methods and presentation of the basic principles behind the most popular ones. In the second part of the chapter, emphasis is put on the Particle Finite Element Method, which is the tool developed and studied in the current thesis. The fundamental principles of the method are initially presented and the various applications found in the literature are mentioned. The chapter closes with the derivation of the implicit formulation of the method, which is similar to the available in-house Material Point Method code.

2.1. LARGE DEFORMATION NUMERICAL SIMULATION TOOLS

The numerical simulation of large deformation problems in engineering has been a topic of research for many years, giving birth to a number of different approaches. The various techniques can be categorized based on their different features, such as the representation of the problem (discrete, continuum), the selected description of motion (Lagrangian, Eulerian) and the adopted connectivity (mesh-based, meshless).

The problem under examination can be modelled using either a discrete or a continuous approach. In the former, the problem is represented by a number of particles, e.g. grains, powder, etc., each one of which moves according to its own mass and external/internal forces applied to it, according to Newton's law; external forces are the result of interaction with neighboring particles. The most typical example of discontinuous methods is the Discrete Element Method (DEM). On the other hand, in the continuum approach, the medium under consideration is viewed as a continuum body, the motion of which is formulated based on conservation laws (e.g. momentum, mass, energy) and a suitable stress-strain constitutive relation. Generally, continuum approaches are more suitable compared to discrete approaches for modelling practical problems, including soil mechanics problems, because of the easier material parameter calibration and faster simulation times [69].

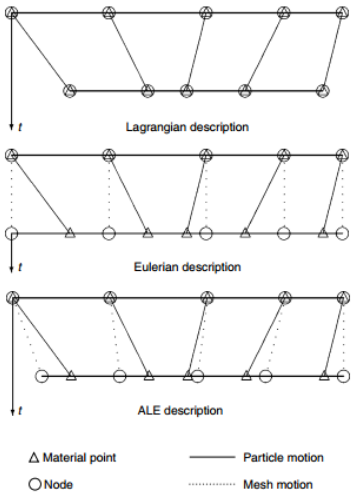

Figure 2.1: Lagrangian, Eulerian and ALE descriptions of motion for mesh-based techniques [21].

Further on, continuum approaches are classified into mesh-based and mesh-free (or meshless), depending on the use of a computational mesh.

2.1.1. MESH-BASED METHODS

The most common mesh-based tool is the standard Finite Element Method (FEM), widely employed for simulations of geotechnical applications. When using FEM, a very important feature for the formulation of any problem, especially ones involving large deformations, is the selection of a suitable kinematic description for the continuum. The two basic descriptions are the Eulerian and the Lagrangian.

In the Eulerian (or spatial) approach, the movement of a material through a fixed space is observed. This is particularly attractive when there is no free boundary and is most commonly used to model flow problems. However, the convective terms present in the equations cause instabilities, while difficulties also occur when trying to model changing boundaries and history-dependent materials [53].

On the other hand, the Lagrangian approach is used -either in its Total or Updated form- when there are large displacements, rotations and strains of solids. In the former, the reference configuration is the initial one, while, in the latter, the configuration in the previous time step is used as a reference, which means that it is being constantly updated. The Total Lagrangian (TL) is considered useful only when the displacements are large but the strains are small. This is why the Updated Lagrangian (UL), which is suitable for large strains, is more popular nowadays. Despite their suitability for capturing history-dependent material behavior and free surface evolution, the Lagrangian formulations are dysfunctional when large deformations result in severe mesh distortion [40], unless frequent remeshing is employed (adaptive techniques).

In an attempt to overcome the limitations of each description and exploit their merits, combinations of the two have been proposed. One approach is the Arbitrary Lagrangian-Eulerian. In the ALE technique, the computational (background) mesh from

the Eulerian approach is employed. The difference is that this grid is not fixed, instead it is now free to move, decoupled from the material motion (Fig. 2.1); however, if this mesh was made to follow the movement of the material particles, then a typical Lagrangian formulation would be obtained. This is why the velocity fields of the two grids are separated, minimizing the possibility of severe mesh distortion. Nevertheless, the convective term is still present, new surfaces cannot be created, mesh tangling is still encountered and the method is limited to the case where material deformations are relatively predictable [53, 67].

Another technique, which was originally introduced [28] as an ALE variation, is the Remeshing and interpolation technique with small strain (RITSS). The two features that were added by the creators was the refinement in areas of high gradients, based on a Delaunay triangulation, and the linear interpolation of the stress values at the Gauss points of the new mesh from values at the nodes of the old mesh. The RITSS technique has been mostly used in offshore geotechnical applications with large deformations, mainly ones involving penetration [61, 63]. Despite the encouraging results, the RITSS method still suffers from some of the ALE issues that were already mentioned, e.g. when the original boundaries change, in case a one-piece solid gets split into more than one parts [67].

Another combination of the two descriptions is the Coupled Eulerian Lagrangian (CEL), which is, however, more suitable for soil-structure interaction. In this technique, the mesh is fixed (Eulerian) and the soil is modelled as an Eulerian material that flows through the mesh. Rigid bodies, such as penetrating piles or rock blocks, are considered as a Lagrangian domain. The Eulerian Volume Fraction (EVF) concept is employed, where every (fixed) Eulerian element is characterized by a percentage indicating the portion of the element that is filled with actual material. An empty (void) element has EVF = 0 and a cell completely filled with material has EVF = 1. The Lagrangian elements can move through the Eulerian mesh freely, until they encounter an element with EVF > 0. Then, the contact between the Lagrangian and Eulerian material is controlled via a contact algorithm, e.g. a penalty or kinematic contact algorithm. The technique can be incorporated in commercial FEM software, such as ABAQUS, and was used [20] to model progressive clay slope failure. The main drawback of the method is located in the great computational times, the mesh-related variable mapping and the mesh refinement dependence [53], with the last one, however, which was overcome with the use of an element size scaling rule in the constitutive relations [20].

2.1.2. MESHLESS PARTICLE METHODS

In meshless particle methods, information is carried by nodes (or particles) and node connectivity is evaluated through a certain domain of influence defined for each node (particle). Every particle interacts with the particles that are within its domain of influence, which is controlled by a smoothing or weight function. This eliminates the need for a mesh and, subsequently, the problem of mesh distortion vanishes. The different approximations between mesh-based methods and meshless particle methods are shown in Fig. 2.2.

The Smoothed Particle Hydrodynamics (SPH) method is the oldest meshfree particle method, first introduced [25, 33] to solve gas dynamics problems in astrophysics and is a truly meshless Lagrangian numerical technique. The domain is discretized into particles

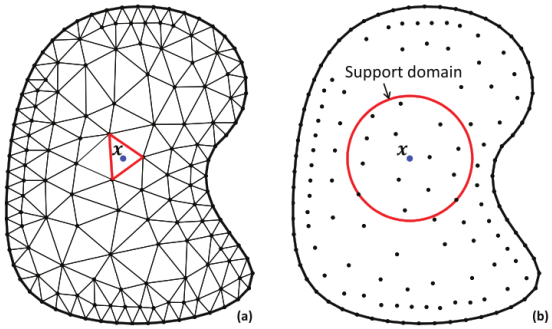

Figure 2.2: Approximations in the (a) FEM and (b) Meshfree Particle methods [66].

that have material properties and a spatial distance, called the smoothing length, over which the material properties are “smoothed” by a kernel function, which is the integral (continuous) approximation of a function or a derivative. These integrals are discretized in terms of contributions from other particles within the domain.

SPH can model large deformation problems without mesh distortion and is able to handle governing equations and constitutive models of geomaterials, since it is based on continuum approximation. This has led to its use for several soil mechanics applications, including slope stability [5, 6, 42]. However, among other drawbacks, SPH exhibits spatial instabilities, as a consequence of pointwise integration, and the fact that insufficient neighboring particles can cause inconsistencies [53]. Moreover, most existing SPH simulations of granular flows suffer from significant stress oscillation during the post-failure process [41].

Additionally, a general disadvantage of meshless methods is that node connectivity depends on the particle positions, which, often, are updated at each time step, leading to a high demand for CPU usage. Other meshless particle methods, such as Moving Least Squares (MLS) and Element Free Galerkin (EFG), exhibit the same, fundamental, drawback.

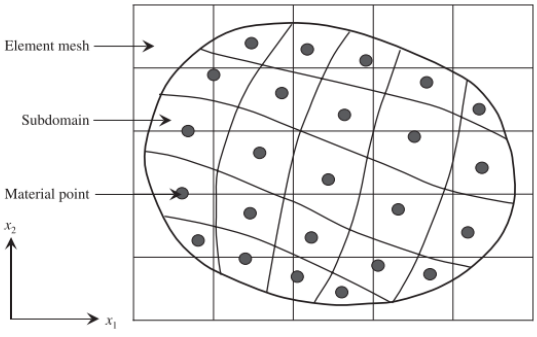

Figure 2.3: Background grid, material points and domain for the Material Point Method [15].

2.1.3. MESH-BASED PARTICLE METHODS

A new category of numerical techniques has been developed in the last decades, labeled Mesh-based Particle methods. These methods enjoy the flexibility of the particle methods in handling large deformations and the robust mathematical foundation of the traditional FEM.

The most popular one is the Material Point Method (MPM), which was born as a simple ALE method and was initially presented [54] as an expansion of the Particle-in-cell (PIC) method, for flow modelling initially, and, later on, for application in solid mechanics [55]. In PIC, materials are represented by particles, to which a mass and position are assigned. The differentiation of MPM and the base idea is that the particles are also given momentum and energy (Fig. 2.3). A fixed Eulerian grid is used for computations based on an Updated Lagrangian description; after the variable mapping and the calculation step (Updated Lagrangian phase), the material points are moved to their new positions (convective phase) and the mesh is reset to its initial state. The Material Point Method code developed in the geoengineering section of TU Delft has been proven to be a useful tool for capturing pre- and post-failure slope behavior [58–60], despite some existing limitations, e.g. cell crossing noise, lowered accuracy due to constant forward and backward mapping etc. While effort is being devoted on optimizing the current in-house MPM code [26, 27], attention is also set on exploring the capabilities of other available numerical techniques proposed in literature, through comparison with the already available MPM code.

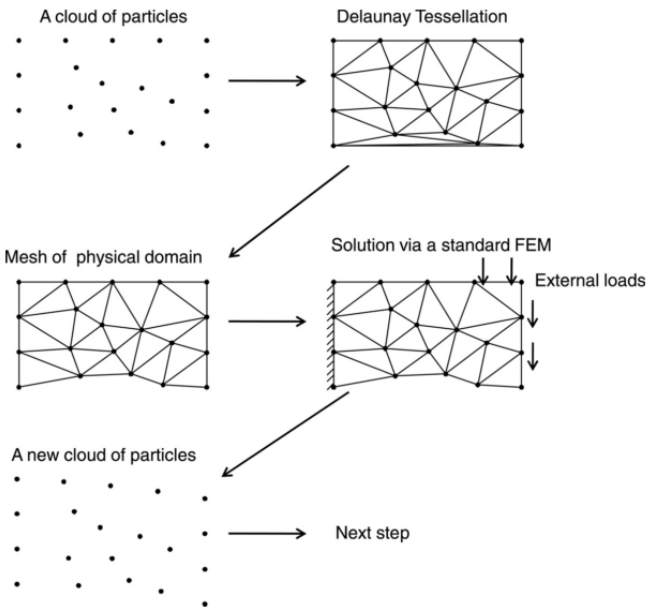

Figure 2.4: Typical solution scheme in the Particle Finite Element Method [65].

Another promising tool of this category is the Particle Finite Element Method (PFEM) [47], which uses constant mesh regeneration for dealing with large deformations. The basic features of the method [12] are transmission of information by particles, which are

used as the FEM mesh nodes, the constant update of the analysis domain, the instant, implicit identification of domain boundaries and contact treatment during mesh generation. A typical solution scheme for the method is shown in Fig. 2.4. Although the method was initially developed for Fluid-structure interaction (FSI), adaptations have been made for use in solid mechanics problems and the encouraging results already published, along with the relatively straightforward implementation, make this an attractive alternative for numerical analysis of large deformation problems in soil mechanics, including slope stability assessment.

2.2. THE PARTICLE FINITE ELEMENT METHOD

2.2.1. INTRODUCTION

The Particle Finite Element Method started out as an attempt to overcome the limitations of existing Fluid-Structure Interaction (FSI) numerical modelling methods, especially the Arbitrary Lagrangian-Eulerian (ALE), which is considered as the most popular one. The authors [47] managed to bypass most of the issues posed by ALE, by using a Lagrangian description for the motion of both the solid and the fluid, able to capture the constantly changing domain geometry and contact interfaces in problems of such nature. The resulting formulation allows for tracking of the motion of all individual particles, which, in the work of the authors, are used as the nodes in a finite element mesh. These "nodes" are allowed to move freely and even separate from the analysis domain, resembling, e.g., water drops. The motion of separated particles is determined by their initial velocity and the body force they are subjected to. This means that the analysis domain and its boundaries are allowed to evolve, which is easily captured via frequent remeshing.

A typical solution scheme in the PFEM, as presented in Fig. 2.4, involves the following steps:

1. A cloud of particles, labeled C^n, is given at time n.
2. The analysis domain, V^n, is identified.
3. The domain is discretized into a finite element mesh, M^n.
4. The state variables from the previous mesh M^{n-1} are mapped to the new mesh, M^n.
5. The discrete governing equations are solved in a standard FEM manner.
6. The positions of the particles are updated, based on (5), and a new cloud C^{n+1} is obtained.
7. The procedure is repeated until the end of analysis time.

The main differences from classical FEM are the update of the analysis domain and the boundary identification procedure at each time step; the rest of the computation steps are identical to FEM.

The main advantages of the method are its ability to capture large deformations with accuracy, while automatically imposing boundary conditions and contact laws, enabling the simulation of multiple body interaction. The primary drawback of the method is the need for increased computational resources [53, 57], compared to other alternatives, due to the frequent remeshing. However, the CPU time required during the remeshing process is lower than the times needed for assembly and equation solving [46], and this downside is expected to diminish as computer hardware becomes more powerful.

PFEM FOR SOLID MECHANICS

Despite being a young numerical method, the Particle Finite Element Method has already been used in many different applications. Although PFEM was initially developed for tackling fluid-structure interaction problems, its features made it attractive for solid mechanics applications, soon after it was first introduced [10, 43, 45]. The main difference between fluid and solid modelling in PFEM is the need for mapping of internal variables between the old and new mesh in the latter, due to the use of history-dependent materials. Constitutive laws that are evaluated at the integration points of elements would be hard to incorporate into PFEM [17]; the constant mapping of stresses (and other related variables) between Gauss points and particles, given that the latter are the information carriers, would compromise the method's accuracy. Other differences between fluid and solid modelling are found in the way that the contact between various domains and rigid boundaries is modelled and the enforcement of mass conservation.

Having employed various solutions for tackling the arising concerns, researchers have been able to model a number of solid mechanics problems, including some of geotechnical nature. Only a few years after being introduced, PFEM has already been employed to simulate problems involving material flows, e.g. granular flows and landslides [3, 16, 68], slope stability [65, 70], consolidation [36, 37], undrained object insertion, e.g. CPT [35–37], tunneling and excavation [11, 13], etc. More particularly, on the topic of slope stability, a variation of the method has been employed [70] to simulate progressive failures of horizontally-based, sensitive clay slopes; they used a rate dependent combination of viscous and elastoplastic constitutive models, to capture both the undisturbed and remoulded phases of the material, and transformed the mixed (displacement -stress) formulation into a min-max optimization problem. The results, including failure initiation and post-failure behavior, enhance the reliability of the method for modelling such problems in soil mechanics, with use of advanced constitutive laws.

2.2.2. FORMULATION

The scope of this thesis is to compare the efficiency of the already built implicit MPM code available in the Geo-Section of TU Delft with a similar PFEM formulation. Therefore, the formulation of the PFEM will follow the derivation of the implicit material point method (IMPM) [57], which is the basis for the aforementioned standard MPM code. Moreover, the derivation of the implicit MPM code has been carried out using standard FEM procedures [2], and because, as already discussed, PFEM essentially uses an FEM technique for solution of the governing equations, the correspondence between the formulation for the two methods should be straightforward.

The derivation [57] starts from the conservation of momentum

$$\rho a = \nabla\sigma + \rho b \tag{2.1}$$

where ρ is the material density, a is the acceleration, σ is the Cauchy stress tensor and b are body forces (e.g. gravity forces).

For convenience, in the derivations provided below, the dynamic term is initially omitted, meaning that the quasi-static formulation of the conservation of momentum is used.

$$\nabla\sigma + \rho b = 0 \tag{2.2}$$

By application of the virtual displacement principle, the equilibrium at the next time step for a given element of the spatial discretization can be expressed as

$$\int_{\Omega_e^t} S^{t+\Delta t}\delta\epsilon^{t+\Delta t}\mathrm{d}\Omega = R_{ext}^{t+\Delta t} \tag{2.3}$$

Where S is the second Piola-Kirchoff stress tensor, $\delta\epsilon$ is the small strain tensor, Ω is the domain under investigation, and R_{ext} is the virtual work equal to the external forces applied multiplied by the corresponding virtual displacement. Superscript $(\cdot)^{t+\Delta t}$ refers to the end of the time step and subscript e refers to the element being examined.

For large strain analysis, an objective strain tensor, such as the Green-Lagrange, could be used and the higher order terms should be considered in the derivation of the governing equations. This is also the case for the original MPM formulation [57]. However, in the MPM code under comparison the extra terms are neglected, under the assumption of operating in small deformations, which is ensured by the constant mesh reset. The same assumption is followed in this work, as, additionally, infinitesimal strains in the elements in PFEM can be ensured by the frequent mesh regeneration employed.

The stress at time $t+\Delta t$ can be calculated, considering an incremental solution procedure, as the stress at time t plus a stress increment.

$$S^{t+\Delta t} = S^t + \Delta\sigma \tag{2.4}$$

Where, the stress at the current state is the Cauchy stress,

$$S^t = \sigma^t \tag{2.5}$$

The strain at time $t+\Delta t$ is the incremental strain

$$\delta\epsilon^{t+\Delta t} = \Delta\epsilon \tag{2.6}$$

Where, the incremental strain for infinitesimal strains is given by

$$\Delta\epsilon = \frac{1}{2}(\nabla\overline{u} + (\nabla\overline{u})^T) \tag{2.7}$$

Where $\overline{u}$ is the displacement increment.

Substituting Eqs. 2.4 and 2.6 in Eq. 2.3, and expressing the product of the external force with the virtual displacement at the current configuration as $R_{ext}^{t+\Delta t}$, the following expression is derived

$$\int_{\Omega_e^t} \Delta\sigma\delta\Delta\epsilon d\Omega = F_{ext}^{t+\Delta t} - \int_{\Omega_e^t} \sigma^t \delta\Delta\epsilon d\Omega \tag{2.8}$$

SPATIAL DISCRETIZATION

The virtual displacement terms on both sides of Eq. 2.8 are eliminated, so a simple equation of force equilibrium is obtained. In order to solve this equation, it must be spatially discretized. In PFEM, this is done in the same way as in traditional FEM, i.e. with use of shape functions to approximate displacement within the element (continuum field) with the nodal displacements.

$$[K]^t\{\overline{u}\} = F_{ext}^{t+\Delta t} - F_{int}^t \tag{2.9}$$

where vector F_{int}^t holds the internal forces, calculated from the stresses at the beginning of the time step,

$$F_{int}^t = \sum [B]\sigma det(J)w \tag{2.10}$$

For simple, constant strain triangles (CST), with one integration point, the stiffness matrix of each element is calculated as

$$[k] = [B]^T[D][B]det(J)w \tag{2.11}$$

where [B] is the standard strain-displacement transformation matrix, holding the derivatives of the shape functions, which are constant for the CST, [D] is the constitutive matrix, [J] is the Jacobian matrix and w is the weighting of the integration point. All the quantities are evaluated at the position of the single integration point within each element.

DYNAMIC FORMULATION

The dynamic governing equation can be obtained by introducing an intertial term in Eq. 2.9

$$[K]^t\overline{u} + [M]^t a^{t+\Delta t} = F_{ext}^{t+\Delta t} - F_{int}^t \tag{2.12}$$

where $[M]^t$ is the mass matrix at time t. Using Newmark's time integration scheme, the velocity and displacement at the end of the time step can be computed by

$$v^{t+\Delta t} = v^t + [(1-\gamma)a^t + \gamma a^{t+\Delta t}]\Delta t \tag{2.13}$$

$$\begin{aligned} u^{t+\Delta t} &= u^t + v^t\Delta t + [(\frac{1}{2} - \beta)a^t + \beta a^{t+\Delta t}]\Delta t^2 \\ \overline{u} &= v^t\Delta t + [(\frac{1}{2} - \beta)a^t + \beta a^{t+\Delta t}]\Delta t^2 \end{aligned} \tag{2.14}$$

The solution of Eq. 2.14 for $a^{t+\Delta t}$, after some manipulations, yields

$$a^{t+\Delta t} = \frac{\overline{u}}{\beta\Delta t^2} - \frac{v^t}{\beta\Delta t} - (\frac{1}{2\beta} - 1)a^t \tag{2.15}$$

Eq. 2.15 is substituted in Eq. 2.12, which becomes

$$\begin{aligned}
[K]^t\overline{u} + [M]^t(\frac{\overline{u}}{\beta\Delta t^2} - \frac{v^t}{\beta\Delta t} - (\frac{1}{2\beta} - 1)a^t) &= F_{ext}^{t+\Delta t} - F_{int}^t \\
[K]^t\overline{u} + [M]^t(\frac{\overline{u}}{\beta\Delta t^2}) &= F_{ext}^{t+\Delta t} - F_{int}^t - [M]^t(-\frac{v^t}{\beta\Delta t} - (\frac{1}{2\beta} - 1)a^t) \\
[K]^t\overline{u} + [M]^t(\frac{\overline{u}}{\beta\Delta t^2}) &= F_{ext}^{t+\Delta t} - F_{int}^t + [M]^t(\overline{a}^t) \\
([K]^t + \frac{[M]^t}{\beta\Delta t^2})\overline{u} &= F_{ext}^{t+\Delta t} - F_{int}^t + [M]^t(\overline{a}^t)
\end{aligned} \tag{2.16}$$

where, $\overline{a}^t$ has been defined as

$$\overline{a}^t = \frac{v^t}{\beta\Delta t} + (\frac{1}{2\beta} - 1)a^t \tag{2.17}$$

Finally, the final expressions for the stiffness matrix and the external force vector are taken by

$$[\overline{K}]^t = [K]^t + \frac{[M]^t}{\beta\Delta t^2} \tag{2.18}$$

$$\overline{F}_{ext}^{t+\Delta t} = F_{ext}^{t+\Delta t} + [M]^t(\overline{a}^t) \tag{2.19}$$

Eventually, the final dynamic equilibrium of the system can be written as

$$[\overline{K}]^t\overline{u} = \overline{F}_{ext}^{t+\Delta t} - F_{int}^t \tag{2.20}$$

Eq. 2.20 is solved in every time step on the mesh nodes and the displacement of each particle is obtained.

If the displacements are known, then, using Newmark's scheme, the velocities and accelerations of the particles can be also calculated. The new acceleration can be found by Eq. 2.15. Then, the velocity can be found by Eq. 2.13.

The selected values for the Newmark's scheme's parameters in the analyses are β= 0.25 and γ= 0.5.

The mass matrix initially introduced in Eq. 2.12 is in a diagonal form:

$$[M]^t = \begin{bmatrix} M_{11} & 0 & \dots & \dots & 0 \\ 0 & M_{22} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \dots & \dots & 0 & M_{nn} \end{bmatrix} \tag{2.21}$$

where n is the total number of equations. A lumped mass matrix is used, which is straightforward for a low order element, such as the 3-noded triangle used in this work. The contribution of each Gauss point g (hence, element, for CST) in the formation of

the mass matrix for every degree of freedom k of node i associated with the element el is given by [52]

$$m_g^k = \frac{(\rho Area)_g}{number\ of\ nodes_{el}} \tag{2.22}$$

The density ρ of the integration point is calculated from the density of the particles, using standard shape functions.

3

DOMAIN DISCRETIZATION

The Particle Finite Element Method can be crudely described as a conventional Finite Element Analysis scheme, where the quality of the computational mesh is ensured by performing constant remeshing. In this context, the analysis domain is of utmost importance and is inextricably related with the efficiency of the method. The remeshing procedure serves not only as a means for assessing the particle connectivity, but also for identifying the domain boundaries, in order to impose the necessary boundary conditions. Considering the need for frequent remeshing, sometimes performed in each time step or even between iterations within the same time step, a fast and efficient meshing procedure is required. Moreover, the importance of correct boundary definition for the outcome of the analysis highlights the sensitivity of the method on the selected algorithm. In this chapter, first an introduction on mesh classification and quality is performed, followed by the presentation of the Delaunay triangulation, which is the tool used for generating the mesh in all PFEM variations in literature. Then, the remeshing procedure, which involves first the triangulation of the domain and then the boundary identification via the α-shape method, are discussed. The influence of the latter procedure on the simulation is also demonstrated. Finally, the chapter closes with a presentation of the different schemes for improving the quality of the analysis domain, by dynamically altering the number of particles throughout the course of the simulation.

3.1. BACKGROUND

In simple terms, the Particle Finite Element Method can be described as a conventional FEM tool where the requirement for a quality mesh is satisfied by frequent remeshing. Depending on the mesh type, different quality metrics are being used. Although the use of simplicial meshes (triangles and tetrahedra) is predominant in PFEM in literature and is also adopted in this work, a brief presentation of all the different types of FEM meshes is done at this point, followed by a discussion on the criteria and the metrics used for assessing their quality.

3.1.1. MESH CLASSIFICATION

ELEMENT TYPES

A set of points in a 2D space can be joined either by simplicial (triangles) or non-simplicial (quadrilaterals) elements; the simplicial element in three dimensions is the tetrahedron and the non-simplicial the hexahedron. Non-simplicial elements are usually preferred when more accurate interpolations and approximations are needed. Considering the simplest 2D elements in each case, a bi-linear quadrilateral (4 nodes) can produce a linear strain/stress distribution within an element, which is, in general, preferable than the constant strain field given by the 3-noded triangle. On the other hand, simplicial meshes are easier -and faster- to generate; the generation of triangular meshes from a given set of points is a much simpler procedure and a fine discretization can be employed to overcome their inherent disadvantage for handling areas of high stress/strain gradients.

STRUCTURE

The structure of a mesh concerns the way that the element connectivity information is stored. In a *structured* mesh, such as a regular orthogonal grid in 2D, the nodes are numbered in such a way that their connectivity can be recovered using simple operations. Typically, every node is appointed corresponding i and j index values which are unique. Then, the neighbours of node (i, j) consist of all the combinations of $(i \pm 1, j \pm 1)$. This approach reduces the required computer storage significantly, shows better convergence and can achieve higher resolution, but limits the versatility of the mesh topology. On the other hand, *unstructured* meshes are more popular, owing to their ability to model more complex shapes and provide better element shapes and the possibility to create quality graded meshes. Usually, quadrilaterals are used for structured meshes, while triangles are the elements of choice for unstructured grids.

GRADING

The element size in a computational mesh must be small enough to ensure accuracy in the solution. The areas of interest in the numerical simulation of a physical phenomenon, however, are usually limited to certain regions of the total domain. However, in order to improve the accuracy at regions of interest and still maintain a relatively low amount of unknowns, the distribution of points can be adjusted accordingly. This points towards the use of smaller elements only where high accuracy is desired, reducing, in this way, the computational cost of the simulation. Based on this, meshes are categorized as *graded* or *ungraded*.

CONFORMALITY

Grid conformality comes into play when a graded mesh is selected. In such cases, transition zones are present between regions of different refinement and if all the nodes are connected with each other, i.e. there are no “hanging” nodes, then the mesh is said to be *conforming*. Zones of intense transition from coarse to finer parts of the mesh may compromise the accuracy of the solution, especially when wave propagation is examined [31]. On the other hand, a *non-conforming* mesh may decrease the sparseness of large linear systems and allow the use of higher-order elements in areas of interest; it

requires, however, a different formulation and special treatment compared to the conventional Finite Element approach. Typically, conforming meshes are used in practice, as they are easier to generate and more general formulations are applicable.

3.1.2. MESH QUALITY

In general, a FEM mesh must cover some basic requirements; it has to conform as strictly as possible with the object or simulation domain under consideration; ideally the size of elements should adapt regarding the local response of the system, i.e. smaller elements in places of interest and larger elements where no much "action" is expected; and the shapes of the elements should be "acceptable", e.g. skinny triangles are usually not preferred. This subsection presents some of the general conditions that a good Finite Element mesh should meet, along with indexes that can be employed for the assessment of the quality of the generated mesh.

REQUIREMENTS

The elements that comprise a mesh used in the Finite Element Method should fulfill some requirements, regarding their geometrical properties, which are directly or indirectly associated with the accuracy of the outcome of the analysis and the computational cost. Focus will be put on two-dimensional elements.

The first two constraints regard the angles of the element. In general, too small or too large angles should be avoided. Large angles lead to increase in the discretization error, which is typically associated with the error in the gradient approximation. On the other hand, small angles may lead to discrepancies in the calculation of the stiffness matrix of an element; as an angle approaches zero, the largest eigenvalue of the stiffness matrix approaches infinity.

The third condition comes into play when the computational cost of the simulation is of high importance. In general, a finer discretization, i.e. smaller elements, is required at points of interest, e.g. where high stresses are anticipated, while larger elements can be used in other areas. Using small elements everywhere will increase the need for computational resources, while using large elements universally will compromise the accuracy of the solution. The obvious way to meet this condition is using a graded mesh, placing large and small elements accordingly.

The last requirement solely concerns dynamic simulations where an explicit time integration scheme is employed. The stability of such a scheme is governed by the Currant-Friedrichs-Lewy (CFL) condition, according to which the time step used in the analysis must be small enough, so that a disturbance of any kind, e.g.a wave, can not travel across more than one element at one time step. This implies that, for explicit time integration, short-edged elements should be avoided, otherwise very small time steps would be needed, resulting in large computational times, or spurious energy will accumulate, compromising the performance of the numerical scheme. Apparently, this condition is deemed irrelevant when an implicit time integration scheme is used, as in the present thesis (see 2.2.2).

ELEMENT QUALITY METRICS

Different metrics are used for evaluating the quality of elements with different geometries. Elements can be categorized as simplicial and non-simplicial; the former include

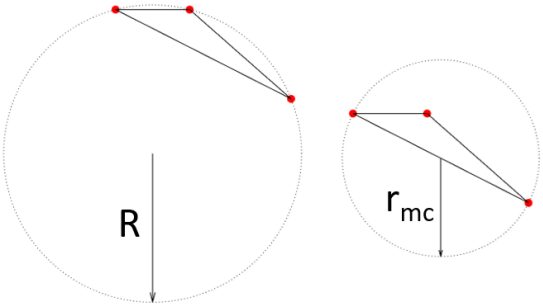

Figure 3.1: Characteristic circles of a triangle. Left: Circumcircle. Right: min-containment circle.

triangles in two and tetrahedra in three dimensions and the latter consist of quadrilaterals and hexahedra, correspondingly.

Regarding triangles, the two circles associated with a triangle are depicted in Fig. 3.1 and are unique for each triangle. The *circumcircle* is the circle that passes through all of its vertices and the *min-containment circle* is the smallest circle passing through two of the vertices and including the third and always has an edge as the diameter.

A simple, common metric for triangles is the *radius-edge ratio*, which is the ratio of the radius of the circumcircle, i.e. the circumradius, over the length of the shortest edge.

$$\alpha_{2D} = \frac{R_c}{l_{min}} \tag{3.1}$$

The lower bound of α value is $\frac{1}{\sqrt{3}}$ and corresponds to an equilateral triangle. Smaller values of α indicate better element quality.

Another representative index of a triangle's shape (slenderness) is the ratio between the radii of the circumscribed and the inscribed circles, R_c and R_i, respectively;

$$f = \frac{R_c}{R_i} \tag{3.2}$$

This ratio is 2 for equilateral triangles and greater than 2 for all other triangles. If any of the angles of the triangle is greater than the other two, the ratio f increases, showing that it comprises an efficient indicator of element distortion.

In three dimensions, the equivalent of the circumcircle for a tetrahedron is the *circumsphere*. However, the criterion in Eq. 3.1 is not suitable for tetrahedra, as the computation of the circumradius in 3D is quite expensive and its value is highly sensitive to even minor vertex perturbations. Instead, a different metric has been proposed [48], the *volume-length ratio*

$$\alpha_{3D} = \frac{V}{l_{rms}^3} \tag{3.3}$$

where l_{rms} is the root-mean-squared length of the six edges of the element. The maximum and optimal value of α_{3D}, computed for an equilateral tetrahedron, is $\frac{1}{6\sqrt{2}}$, while

a value of 0 corresponds to a degenerate tetrahedron; this means that higher values are generally preferred.

3.1.3. DELAUNAY TRIANGULATION

BACKGROUND

Given a set of points $\mathcal{N} = \{n_1, n_2, n_3, \ldots, n_n\}$ in $\mathcal{R}^3$ (or $\mathcal{R}^2$ in 2D), the following definitions can be given.

The *Voronoi diagram* is the partition of the space ($\mathcal{R}^3$ or $\mathcal{R}^2$) into regions V_i (*Voronoi cells*), which may be either closed and convex or unbounded, with each region V_i being associated with point n_i and having the property that every point inside it is closer to n_i than any other point n_k in the distribution. For each set of points $\mathcal{N}$ there is a unique Voronoi diagram.

The *Voronoi circle* (or *sphere* in 3D) or simply *empty circle* is a circle inside which there is no point belonging to $\mathcal{N}$.

The *convex hull* of $\mathcal{N}$ is the minimum convex area that encloses all the points of $\mathcal{N}$.

The *dual graph* of a plane graph, as the Voronoi diagram, is the graph that has a vertex for each face of the plane graph. The different ways of connecting those vertices define various types of dual graphs.

Considering the set of points $\mathcal{N}$, a triangle is considered to be a *Delaunay triangle*, if its vertices belong in $\mathcal{N}$ and no other points of $\mathcal{N}$ is *inside* the circumcircle of the triangle; there is no limitation on the number of points *on* the circumcircle. If the only points on the circumcircle are the vertices of the triangle, then the triangle is considered to be *strongly Delaunay*.

The *Delaunay Triangulation* (DT) is the convex hull of the points of $\mathcal{N}$ in the Voronoi diagram, in which every triangle is a Delaunay triangle, i.e. having an empty circumcircle. The Delaunay Triangulation is the *straight line dual graph* of the Voronoi diagram, i.e. the points on which the Voronoi diagram is constructed are connected with straight lines. In other words, the correspondent of a Voronoi polygon is a Delaunay vertex, a Voronoi edge corresponds to a Delaunay edge and a Voronoi vertex to a Delaunay triangle. This duality is exploited for improvement of the initial Delaunay tessellation for mesh generation.

PROPERTIES

Optimality The Delaunay Triangulation, in its original form, ensures some qualities of the generated triangulation. More particularly, the minimum angle of all the triangles in the tessellation is maximized. This property is related with the requirement for avoiding small angles in the domain, as discussed in 3.1.2.1. Secondly, the largest circumcircle of all the triangles is minimized. This means that the radius-edge ratio (Eq. 3.1) of the triangles is decreased, approaching the value of the equilateral triangle; this leads to better quality meshes. Finally, the largest mid-containment circle is minimized, which also improves the triangle quality of the generated grid.

Uniqueness A Delaunay Triangulation is unique if and only if there are no more than three points on a common empty circle. If there are, in example, four points on a circumcircle, then a degeneracy is generated and different combinations can be made, as

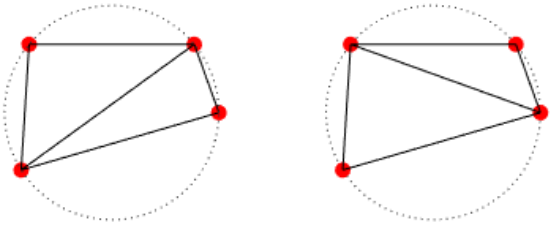

Figure 3.2: Four points in an empty circle leading to two different triangulations.

seen in Fig. 3.2. This phenomenon can have an implication during remeshing, as minor relative movement of adjacent vertices may lead to what is called a "flip", i.e. the change of the common edge of two adjacent triangles, with repercussions in the course of the analysis (see 4.2).

CONSTRAINED DELAUNAY TRIANGULATION

In some cases, given a set of points for triangulation, there may be a requirement for a prescribed connection between some of those points, which the original DT might miss. This can be, for example, the boundary of a non-convex object. A constrained Delaunay triangulation is used to impose the existence of those specified edges, which would not be constructed by the original triangulation, without introducing new vertices. The effect of performing a constrained Delaunay triangulation is that the "Delaunay property" of triangles, as described in 3.1.3, is relaxed, meaning that there may be points of the given data set inside one of the circumcircles of the newly specified triangles. However, the quality of the generated triangulation still remains high, enjoying similar optimality properties to that of the original Delaunay Triangulation, in the sense that, among all triangulations of the given data point set and the specified edges, the constrained Delaunay triangulation will maximize the minimum angle and minimize the largest circumcircle and the largest mid-containment circle of the triangulation (see 3.1.3.2).

3.2. MESHING PROCEDURE

The first step for (re)generating the computational mesh involves an initial assessment of nodal connectivity with a Delaunay Triangulation (3.1.3). During the second step, any irregular triangles that may be created from the DT have to be eliminated, in order to enhance the quality of the mesh and determine the boundaries; this is done with the α-shape method [22]. After the mesh is created and the incorrect elements are discarded, separated particles and particles lying on the boundaries of the domain can be identified, so the boundary conditions and contact constraints can be imposed. In order to showcase the meshing procedure, which is, typically, performed at each time step, a demonstration is performed on the cloud of points C shown in Fig. 3.3.

3.2.1. INITIAL PARTITION

In the original form of the PFEM [47], the authors used a so-called Extended Delaunay Tessellation (EDT) [9], which allows the combination of elements of different shapes, i.e. polygons (or polyhedra in 3D) of different number of vertices. Such a choice increases

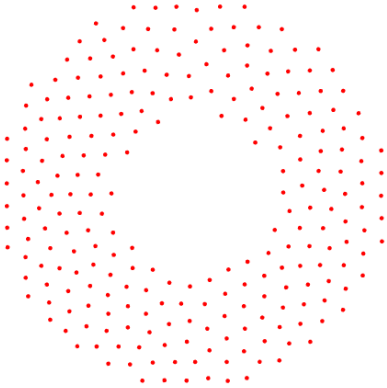

Figure 3.3: Cloud of points C.

the implementation difficulty of the meshing procedure and the complexity of the numerical method, as the EDT algorithm is not readily available online and special Meshless Finite Element (MFEM) shape functions are required. This alternative did not enjoy much popularity among researchers in the following years, who showed a preference for the conventional DT. Thanks to its popularity, the algorithm for the simple DT used in the thesis was found available online [7], with some manipulation needed to make the code compatible with modern FORTRAN 90 compilers.

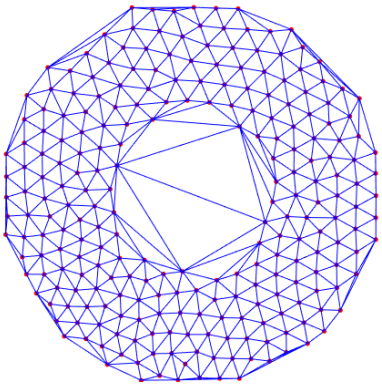

Figure 3.4: Initial Delaunay Triangulation.

As already mentioned, the Delaunay Triangulation is a popular technique for generating simple, two-dimensional, triangular meshes. The result is a conforming, unstructured, ungraded, triangular mesh (3.1.1). The generated mesh consists of low-order three-node triangles, with one integration point (Constant Strain Triangles – CST). This is expected to introduce numerical instabilities, i.e. volumetric locking in incompressible materials, therefore, either a stabilization treatment for FSI problems [17, 29] or mixed formulations for solid mechanics problems [35, 67] have been proposed. However, such issues were neglected in the present work. Moreover, the stress discontinuities between elements caused by the low-order interpolation can be mitigated by reducing their size in areas of interest, which, however, will unavoidably increase the times for meshing, as-

sembly and solution. In any case, the drawbacks of using CST are counterbalanced by the reduced computational times required for performing DT. This allows for generating adequately fine meshes, on which the performance of CST approaches that of higher order, six-noded Linear Strain Triangles (LST).

In its simplest form, the algorithm creates all the possible connections between the points of the cloud. Fig. 3.4 shows the results of the initial Delaunay Triangulation on the initial cloud of points C in Fig. 3.3, where it is apparent that some unwanted elements are also present. These include both the slender elements on the outer boundary on the domain and the large – compared to the average element size - elements formed in the internal cavity of the domain. These irregular triangles have to be discarded, for the boundaries to be identified and for a final quality mesh, representative of the problem domain, to be obtained. The next subsection discusses the required procedure.

3.2.2. α - SHAPE METHOD

In a numerical simulation, an important aspect for the consistency and the accuracy of the solution is the correct determination of the analysis domain. This matter becomes progressively important in fluid or flow-like simulations, e.g. a landslide; in this kind of simulations, the boundary conditions are critical for the physical problem under consideration, e.g. free surface (atmospheric) pressure in fluids, frictional (or any other contact behavior) in solids, and the domain boundaries are prone to frequent change, even from one step to the next. Moreover, the boundary identification must take into account both the boundary of the domain and any particles separating or re-entering the domain. The correct -and consistent - definition of the boundary surface of the domain is also an important step towards mass conservation throughout the analysis.

Sometimes, the external boundaries of a domain are explicitly defined, usually at the beginning of the simulation, and the boundary nodes – or particles – are distinguished from the internal nodes. However, in other situations, the only available information is the cloud of points, based on which the domain boundaries must be determined. This is also the case in the Particle Finite Element Method, where the boundary identification is performed every time the mesh is “destroyed” and rebuilt, i.e. during remeshing.

In most of the implementations of the Particle Finite Element Method in literature, the alpha-shape method is employed for defining the boundaries of the domain. According to a simple variation of the method [17] which is also employed in this thesis, for every element generated by the initial Delaunay Triangulation, a geometrical index similar to the one in Eq. 3.1, is calculated, based on

$$\alpha = \frac{R_c}{h_{avg}} \tag{3.4}$$

where R_c is the circumradius of the element and h_{avg} the characteristic spacing of the particle distribution. The procedure can be sped up if all the generated empty (Voronoi) circles from the Delaunay Triangulation algorithm are stored. Then, the computed α value is compared with a threshold value α_{lim} and, if the value is exceeded, the element is discarded. After the incorrect triangles are removed, the domain boundary can be determined. For a point in the cloud, if an empty circle (or sphere in 3D) of radius R = $\alpha_{lim} h_{avg}$ can be drawn that passes through it, then the point is considered to be on

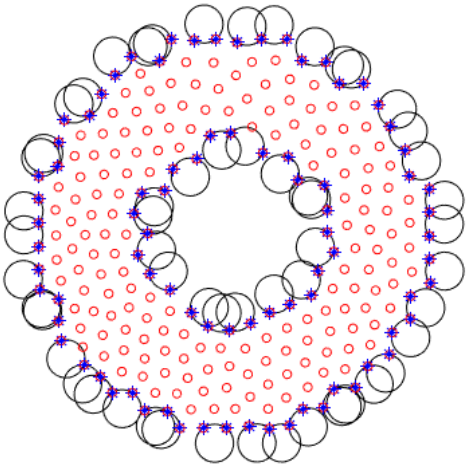

Figure 3.5: Boundary recognition based on cloud of points.

the boundary of the domain. The α_{lim} is the one used in the α-shape method. The recovered boundary particles from the cloud of points C in Fig. 3.3 and $\alpha_{lim} = 1.2$ are shown in Fig. 3.5, indicated by blue stars.

The boundary surface in 2D is, then, defined by all the polygons (or polyhedral surfaces in 3D) having all their nodes on the boundary and belonging to just one polygon. The error in the identification of the boundary surface using the α concept is proportional to h [47], which is considered acceptable. The boundary surface determination can only be made more precise by decreasing h, i.e. increasing the particle distribution refinement.

The choice of α_{lim} is up to the user, a fact which introduces a subjectivity in the final form of the domain and has a direct impact on mass conservation and, sometimes, the overall result of the analysis, as will be shown later on (3.2.2.1, 4.3). Fig. 3.6 shows the outcome of the alpha-shape method for different values of the α_{lim} value. It should be noted that for the "ideal", equilateral triangle, the value of α_{2D} is $1/\sqrt{3}$. For α_{lim} values closer to $1/\sqrt{3}$, more elements will be deleted, while as α_{lim} tends to infinity, the initial DT is recovered. Typically, a value slightly higher than 1 is expected to return a representative domain. However, a "correct" value of the parameter cannot be determined a priori, as it the exact choice is case-specific, i.e. dependent on the problem being modelled. As an example, for static and dynamic simulations on purely cohesive or purely frictional solids, a value of α_lim in the range of 1.4-1.6 is suggested [67]; a value around 1.3 is proposed [23] for fluid modelling.

EFFECT OF SELECTED α_{lim}

The effect of the choice of the α_{lim} parameter in the outcome of a simulation is showcased in a simple example. A specimen with dimensions 10 cm by 20 cm and a notch on its left side, shown in Fig. 3.7, is subjected to an upward displacement on its top side. The bottom particles are completely fixed and the horizontal movement of the top side particles is also restricted. An upward prescribed displacement is imposed at the top in steps of 0.75 mm, with remeshing taking place at the end of each loading step. The same

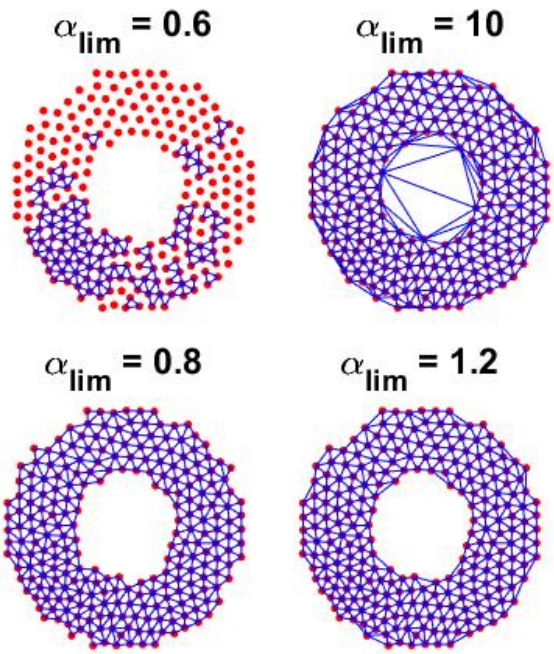

Figure 3.6: Outcome of α-shape method for different values of α_{lim}.

problem is analyzed for three different values of α_{lim}, namely 1.1, 1.2 and 1.4, and the developed reaction force at the bottom is plotted in Fig. 3.10.

As shown in the top row of Fig. 3.8, the impact of the choice of α_{lim} begins from the initial mesh, as it appears that the initial triangulation at the notch is different; for $\alpha = 1.1$, the elements on the inside of the notch are deleted during the $\alpha-$shape step, while for larger values, they are preserved. The saw-like edges of the notch angle are a result of the $\alpha-$shape method as well. As the simulation proceeds, a new mesh is constructed at every step, with the Delaunay triangulation, followed by the $\alpha-$method for deletion of incorrect elements. In the top row of Fig. 3.8, the result of the remeshing procedure after 48 mm of displacement at the top is shown, where it appears that the $\alpha-$method for $\alpha = 1.1$ has "decided" to delete the outer element at the notch, while for higher values these elements are preserved. This is typical for the rest of the simulation. The particle positions at different times in the analysis for the different values of the parameter are shown in Fig. 3.9.

In general, the value of α_{lim} can be seen as an index of the maximum distortion allowed in the elements, with lower values allowing less distortion. This governs the node topology connecting the top and bottom side of the specimen, i.e. above and below the notch, i.e. higher values of α_{lim} allow the connecting elements to be more stretched. In Fig. 3.10, it appears that the higher values of α_{lim} result in the connection of the two parts lasting longer, resulting in a higher reaction force at the base. This trend is in qualitative agreement with publications investigating a similar application [51]. From this example, it becomes apparent that the choice of α_{lim} can be crucial in the overall response of the system.

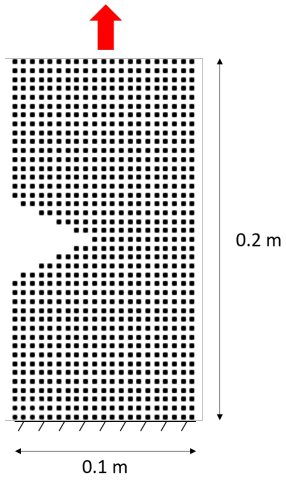

Figure 3.7: Specimen with notch on one side subjected to upward extension.

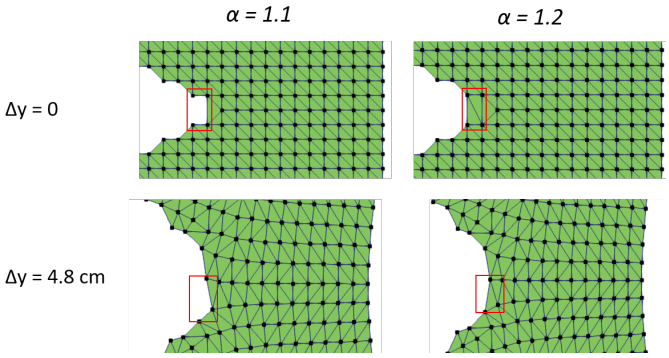

Figure 3.8: Closeup of the mesh at the notch of the generated mesh at the beginning of the analysis (top) and at an intermediate step (bottom), for two values of α_{lim}.

The α-shape method described is directly applicable to ungraded meshes and can be adapted for graded meshes as well. This can be done by introducing a local h_{avg} to different areas of the domain, even to each particle separately. In this way, for each element, the ratio α is computed using the averaged h_{min} of all the particles/nodes of the element and compared to the limit value α_{lim} defined by the user. The extension to graded meshes is, however, beyond the scope of the project.

Fluctuations of the domain boundary between time steps may introduce volume and mass errors in the system. Researchers have acknowledged the negative impact that the dynamic boundary surface during remeshing has on the conservation of the system. To this end, the use of a constrained DT (3.1.3.3) to mitigate volume oscillations during boundary reconstruction has been proposed [19, 49]. This can be done using the boundary surface defined at the initial stage or at previous time steps. However, enforcing a constrained DT is not always possible, especially in cases where complex material

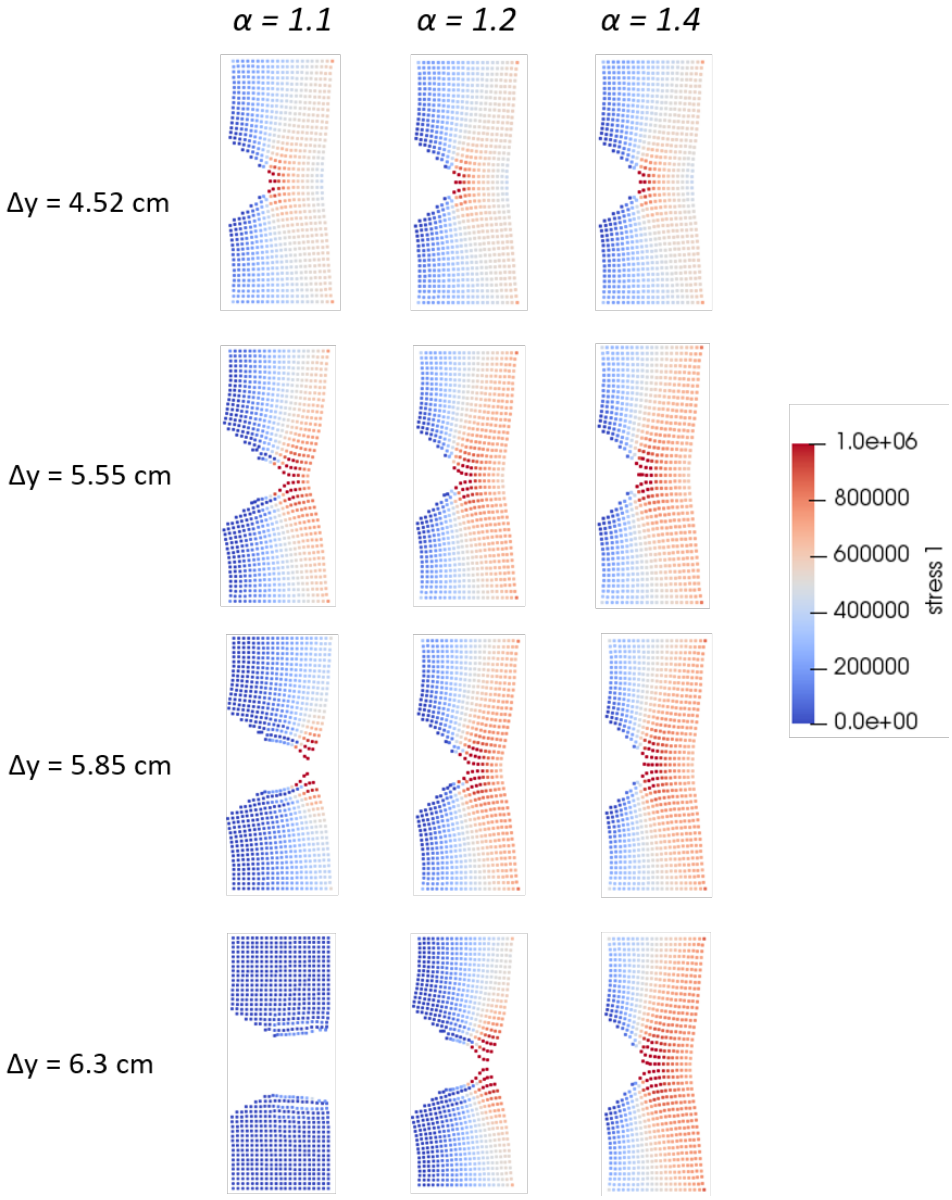

Figure 3.9: Snapshots during stretching of notched specimen for different values of α_{lim}.

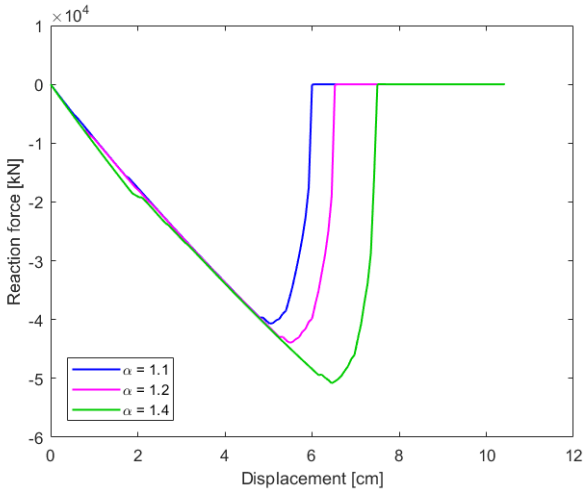

Figure 3.10: Basal reaction force during stretching of notched specimen for different values of α_{lim}.

flow takes place and in some 3D applications. To overcome this, a combination of a non-constrained DT with the alpha shape technique and other techniques to preserve the boundary contour has also been proposed [49].

3.2.3. SEPARATED PARTICLES

During the α-shape method (3.2.2), the particles separating from the domain are also identified. If all the elements to which a particle belonged to where destroyed during the α-shape algorithm, then the particle is considered to be "separated". The particle maintains its properties and state and is, also, assigned the total mass of the deleted elements to which it belonged, in order to preserve the total system mass [17]. After separation, the particle moves based on the body force and the initial velocity it had at the end of previous step. At each subsequent mesh regeneration, the separated particle will become a vertex during the initial Delaunay Triangulation and, if the generated elements are not eliminated during the α-shape algorithm, then the particle will re-enter the domain. The separation and re-incorporation of particles in the domain introduce some oscillations in the system volume and, thus, mass, which will be discussed later on section 4.3.

For thermal diffusion and convection modelling in fluids with PFEM, two possible ways for treating a particle that has separated from the main computational domain have been recovered [1]; one choice is to conserve the same temperature as in the end of the previous step, while the second is to assign to it the temperature of the external medium on which it now lies. In both cases, it is suggested that the particle should move according to gravity forces, maintaining a constant temperature.

3.2.4. REMESHING CRITERION

In its original form [47], the authors of PFEM suggested that the remeshing procedure should be performed at every time step. The rapid motion of fluids, which was the primary material type under consideration in the early implementations of the method,

leads to faster and more severe mesh distortion, compared to solids. Later on, however, in order to save on computational time and limit the inaccuracies caused by information mapping (see 4.2), the idea of using remeshing criteria was proposed. However, such criteria have not been defined explicitly in literature, with one exception [17]. A criterion based on the global mesh distortion has been proposed, making use of the quality metric for triangles of Eq. 3.1. According to this criterion, which is, however, proposed for fluid simulations, at the end of each time step, after the particle positions are updated, the index of distortion is computed for each element and, then, a global average is calculated for the whole mesh, based on

$$\beta = \frac{1}{N_{el}} \sum_{e=1}^{N_{el}} \beta_e \tag{3.5}$$

This is compared to a threshold value, β_{lim}, and if the limit is exceeded, the remeshing procedure is performed. The choice of β_{lim} is up to the user and, should be selected with respect to the adopted α_{lim} value. The use of an averaged global index may not be very efficient in solid (and, specifically, soil) mechanics applications, where local mesh distortion, usually due to shearing, may appear. To overcome this limitation, without resorting to constant remeshing at every time step, different remeshing criteria can be useful.

3.3. DYNAMIC PARTICLE DISCRETIZATION

In PFEM, particles are allowed to move freely, increasing the possibility of two particles occupying the same space or the development of regions of sparsely located particles. The numerical instabilities arising from these phenomena, e.g. high gradients, have been observed by researchers working on both fluid and solid mechanics problems. Some of the introduced criteria, either geometrical or mechanical, for the addition, removal or reposition of "problematic" particles and proposed ways of handling their information. Nonetheless, this phenomenon of particle superposition is more likely to happen in a closed domain, where the particles are not allowed to move completely freely and cover a different, larger area.

3.3.1. PARTICLE ADDITION AND REMOVAL

For simulation of fluid-structure interaction, it has been suggested [30] that every body (domain) under consideration could be macro-discretized in areas of variable h = h(x), where h is an index of the magnitude of the generated gradients of the continuum field. This concept is similar to the concept of graded meshes used in conventional FEM, and can be introduced in particle methods via "joining", i.e. turning two particles into one, or "breaking", e.g. splitting one particle into two, based on their spacing and the required gradient. More particularly, considering a variable h(x) distribution inside the analysis domain, during every remeshing procedure, the following check is performed:

1. If there is an empty sphere (circle in 2D) of radius r>Λh(x), a new particle is introduced at the center of the sphere (circle) and its variables and properties are interpolated from surrounding particles. Λ is a parameter depending on the dimensions of the analysis, and is given by $\Lambda = \frac{1.1\sqrt{n_{dim}}}{2}$. It is also noted that the

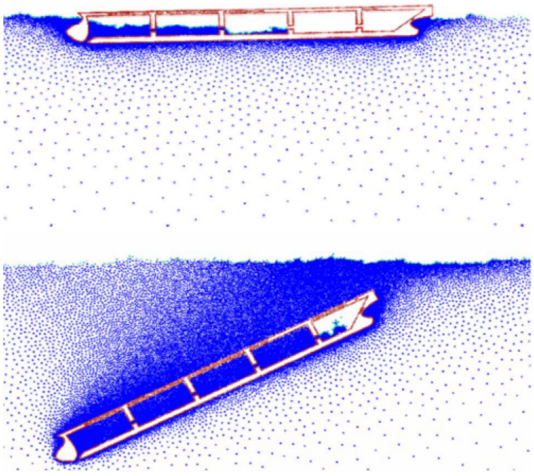

Figure 3.11: Particle distributions at two different times during a fluid-structure interaction simulation, using the joining and breaking particles concept for the fluid domain (after [30]).

parameter Λ should be selected with respect with the α parameter from the alpha shape method, in order to avoid implications.

2. If there is a point x in the domain with h(x) with at least a particle closer than d < λh(x), where $\lambda = 0.5$, then there is no need to insert a particle in position x.

An example of the application of this technique is shown in Fig. 3.11, where the particle distribution during two different times of the simulation of a sinking tanker are shown. At the beginning of the analysis, particles are accumulated close to the surface of the fluid, where the focus is. As the ship sinks, particles are inserted deeper in the fluid to capture the response of the system.

Regarding solid mechanics, the first mention of the possibility of dynamically changing the number of particles in the domain in order to increase the stability of the solution, through mitigation of the instabilities caused by perturbations in the generated boundaries during remeshing was done in [18]; the suggestions were based on geometrical criteria introduced in [32].

The insertion of particles is determined in elemental level and uses different criteria for interior and boundary elements. Initially, for all elements, a reference (triangular) element of area w is used. Based on that, for each element i of area α_i a *size index* is calculated by

$$\tau^{size} = \frac{\alpha}{w} \tag{3.6}$$

Specifying a related tolerance τ^{ins}_{lim}, if $\tau^{size}_i > \tau^{ins}_{lim}$ then a particle must be added in element i. For interior elements, the new particle is inserted at the barycenter of the element, as shown in Fig. 3.12. For boundary elements, an additional criterion is used for the position of the new particle; after deciding if a particle must be inserted inside

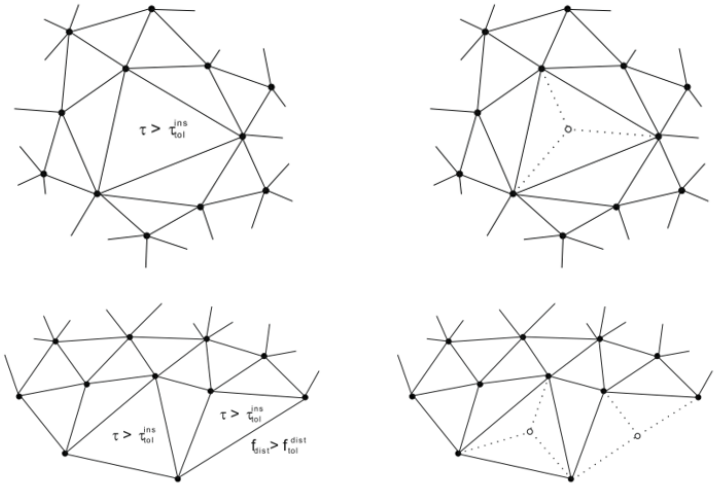

Figure 3.12: Addition of particles in internal (top) and boundary (bottom) elements, after [18].

a boundary element j, based on the size criterion presented above, the exact position of insertion must be decided. This decision is made based on the geometry of the element, taking into account that elements with large aspect ratios are undesirable, based on the criterion of Eq. 3.2. Again, a relative tolerance f_{tol} must be determined; boundary elements that pass the first criterion are subjected to the second criterion, to decide on the final position of the newly inserted particle. If the calculated f_j is smaller than the tolerance, then a particle is added in the barycenter of the boundary element, similarly to the interior elements; otherwise, the particle is inserted in the middle of the longest side of the triangle, i.e. opposite of the largest angle (Fig. 3.12).

In contrast to the criteria for particle addition, which were element-related, the check for particle removal is nodal-based. Again, a discrimination is done between internal and boundary particles. The two criteria are shown in Fig. 3.13.

For interior particles, after the size indexes α_i for all elements have been calculated, according to Eq. 3.6, an average value is assigned to each particle, based on its neighbouring elements, per

$$\tau_{nodal}^{size} = \frac{1}{n_{nb}} \sum_{i=1}^{n_{nb}} \frac{\alpha_i}{w_i} \tag{3.7}$$

Where n_{nb} is the number of neighboring particles of each examined point. This value is compared for each particle with a user-determined tolerance value, i.e. f_{tol}^{rem}, and, if greater, then the particle is removed.

On the other hand, if the distance between two boundary particles is less than a specified tolerance, i.e. $l < l_{tol}^{rem}$, then the two particles collapse into one particle, which inherits their averaged information, including among other their coordinates.

Another approach is proposed [50] for simulation of metal chip formation in steel cutting, a process associated with plastic shear bands. The authors realized adding or

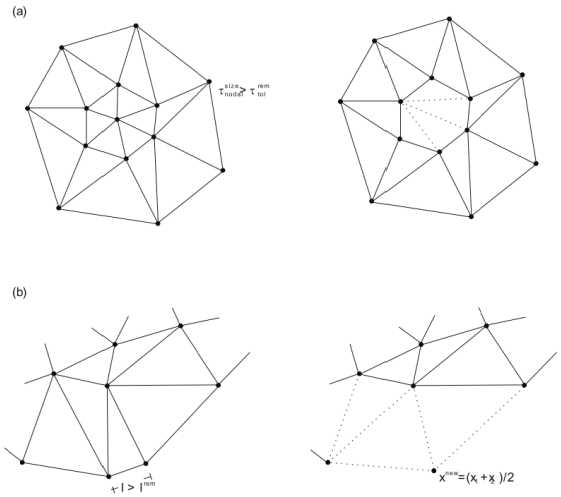

Figure 3.13: Criteria for internal (a) and boundary (b) particle removal, after [18].

removing particles and, therefore, information points inside the domain based solely on geometric criteria could introduce instabilities in the numerical scheme, especially when the history of the domain is important. To overcome problems with mesh distortion, they are the only researchers, to the author's knowledge, who introduced mechanical criteria, along with geometrical, for addition and removal of particles form the domain.

The geometrical criteria are related to the characteristic distance h of the particles, with a similar principle as the solutions presented in [18], except for the fact that all of them are nodal-based. Regarding particle insertion, a particle is added at the center of an internal element's circumsphere or at its sides and on the middle of a boundary segment that is larger than a prescribed tolerance. The deletion of elements is based on similar criteria, i.e. removing closely spaced internal or boundary particles.

The mechanical criteria apply on both particle insertion and removal. For the first case, the distribution of mechanical power is used as a metric. Based on this, the mechanical power over an element is computed and compared with its neighbours. If it exceeds a certain threshold, then a new particle is inserted at the center of the element, so that the power is more evenly distributed (see left of Fig. 3.14). Newly inserted particles acquire their information through an interpolation procedure. On the other hand, the authors based their mechanical criterion for particle removal on error estimators of plastic strain values or the norm of isochoric stress; low values of error indicate that there is not something "interesting" taking place at a certain region, meaning that the mesh can be coarsened, as seen on the right part of Fig. 3.14. When a particle is removed from the domain, its information is lost for the rest of the analysis, but not before the required interpolations during remeshing have taken place, so as to maintain a certain degree of accuracy. The authors also suggest that the refinement procedure can either be applied to the whole domain or in certain areas of interest, which would be the vicinity of the cutting tool in a cutting simulation, or the failure plane in a slope instability analysis. An example of the outcome of the dynamic discretization of the domain is shown

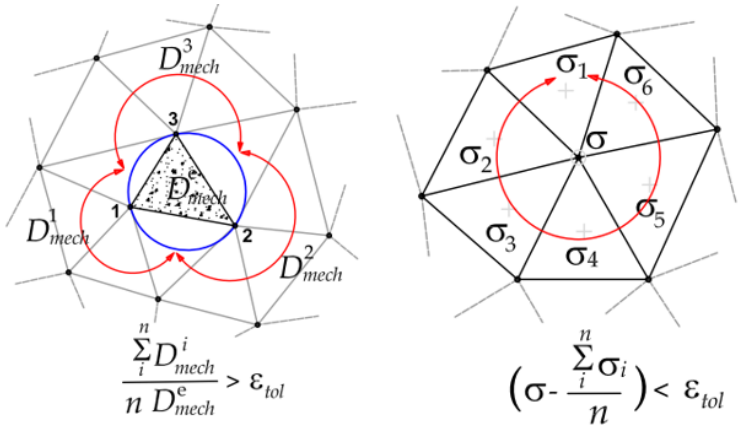

Figure 3.14: Mechanical criteria for the addition (left) and removal (right) of particles during remeshing, after [50].

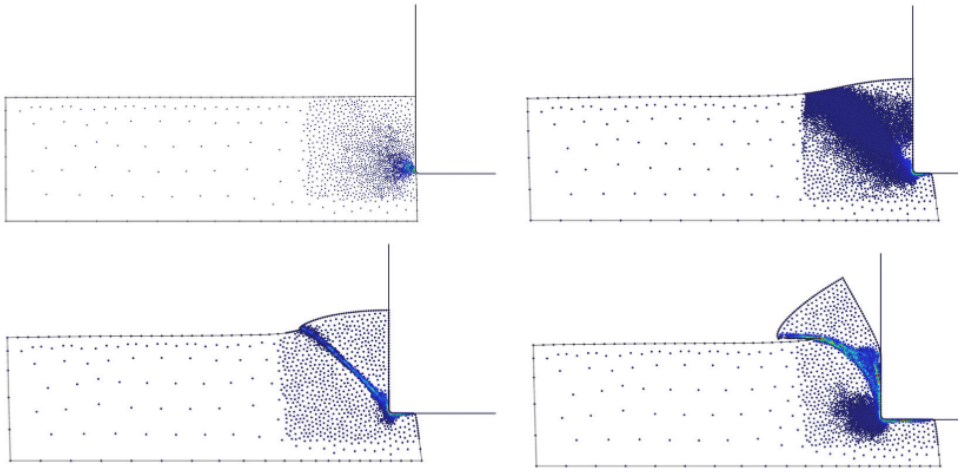

Figure 3.15: Example of particle refinement during a steel cutting simulation, taken from [50].

in Fig. 3.15, during a simulation of metal cutting.

3.3.2. PARTICLE REPOSITION

The particle reposition was addressed in [1] for modelling thermal convection-diffusion in a fluid, and was suggested that when two particles get too close to each other, at the point of superposition, then their information may be lost or smoothed and the mesh quality will deteriorate. The proposed solution was to keep track of the relative positions between all particles using a backround grid, which would increase computational cost, and if the distance between a pair of them falls below a certain threshold, then one of the two should be repositioned in another area of the domain with lower particle density, where the problem variables will be interpolated based on a standard linear projection. Supposedly, this reintroduction -instead of deletion- of the particle is proposed in order to maintain the same number of particles.

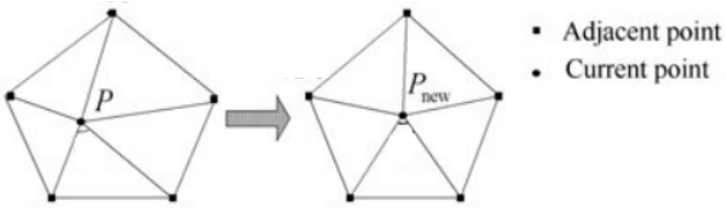

Figure 3.16: Laplacial smoothing procedure (after [34]).

Once the domain is reconstructed through remeshing, the quality of the new mesh is examined in a second phase, in order to avoid remaining distortion of the new elements [18]. The check is performed on the basis of an algebraic quality metric given in [32] and takes into account both the size and the shape of each element. This metric is computed for each element and, then, transferred to the nodes in an averaged sense. If the computed nodal value is lower than a specified tolerance, then the particle is repositioned, based on Laplacian smoothing (Fig. 3.16).

Correct application of dynamic particle discretization can be crucial in reducing the computational requirements of the simulation significantly and increasing the accuracy of the solution. The study of this sophisticated, albeit rarely documented, feature of the Particle Finite Element Method is not extended in this thesis beyond a literature investigation. However, it is highly recommended that it is examined in more depth by future research on PFEM.

4

FEATURES

As already mentioned, the PFEM was initially developed for modelling fluid-structure interaction problems and its early-stage features were tailor-made for this kind of problems. After the encouraging results in this field, it was not long before the method caught the attention of researchers from other disciplines, mainly solid mechanics. However, this meant that the method had to be adapted in order to be able to handle the different nature of each problem. These adjustments are mainly associated with the way that body-body and body-boundary interaction is treated, the handling of information throughout the simulation and the special treatment needed for conserving mass in the analysis.

Despite being a quite young numerical scheme, interestingly, even a brief literature investigation will reveal the numerous different implementations adopted by researchers, depending on the problem to be simulated and the focus of the analysis. From the above it becomes apparent that there is not one single variation of the method which can be studied and, subsequently, implemented and investigated further. In this chapter, the various approaches used in literature for different types of problems are investigated. For each one, first, a literature review is performed to highlight the various approaches, with special remarks on the differences between fluid and solid modelling. Then, the ones deemed most suitable and applicable on this project are implemented and assessed.

4.1. CONTACT TREATMENT

4.1.1. BACKGROUND

A main differentiation of PFEM from conventional FEM, which is also considered an attractive feature that adds versatility, is its ability to automatically track contact interfaces between domains during each remeshing step and, thus, apply contact laws in a straightforward manner. Among the techniques used in the literature for the different possible contact scenarios, the most popular one is the generation of contact elements. These contact elements are used either as special elements that participate in the system

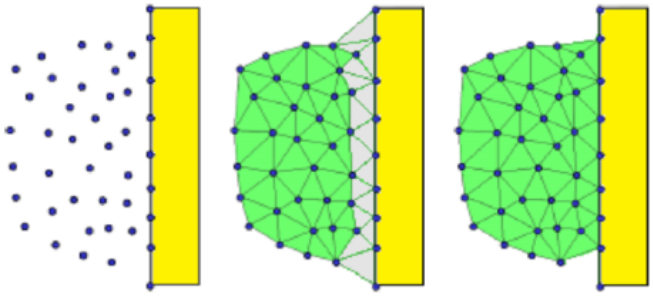

Figure 4.1: Generation of contact elements for fluid modelling (after [17]).

of equations, e.g. through contributions in the stiffness matrix, or as indicators of which particles might interact with another domain. The enforced boundary restrictions typically involve denying penetration and allowing separation in the normal direction and friction, slip or stick behavior in the tangential direction.

The idea of using interface elements is common for both fluid and solid modelling. However, they are being employed in a different fashion. When fluids are modelled, any generated contact elements automatically become fluid elements, as shown in Fig. 4.1, with the same properties and behavior as the elements of the main domain. The incompressibility condition in the governing equations of the fluid ensures no-penetration in the normal direction and a stick or slip condition is applied in the tangential direction. However, the creation or deletion of fluid elements at the interface introduce volume and, thus, mass variations in the system, as will be discussed later on, in section 4.3.

On the other hand, when solids are modelled, the contact elements are used in a different way. The interface between bodies or between a body and a rigid boundary is treated as a continuum itself, with elements of special properties contributing in the system of equations. As an example, elements with density $\rho = 0$ and Poisson's ratio $\nu = 0$, that are only activated when compressed have been used [12]. The interface has the same dimensions as the domains under consideration; this contradicts the conventional approaches, where the interface is of one dimension lower than the domains, e.g. for a 2D analysis, the contact interface is treated as one-dimensional (line).

4.1.2. IMPLEMENTATION

In this project, a simple adaptation of a contact algorithm based on different implementations available in literature is developed, for a number of reasons. First of all, the main scope of the project is to compare the basic features of PFEM and MPM and contact treatment is not of outmost importance. Moreover, at this stage, contact is considered only between the solid domain and the rigid boundaries, which remain constant throughout the analysis. Additionally, only normal contact, i.e. no penetration and free separation, is considered adequate. Finally, the lack of extensive documentation on the matter available in literature and time limitations for completion of the project suggested that a simple, yet effective contact algorithm should suffice at this point.

The adopted implementation for contact treatment is presented in the next subsections, starting with the interface mesh generation, followed by the formulation and a brief validation on some simple benchmark problems.

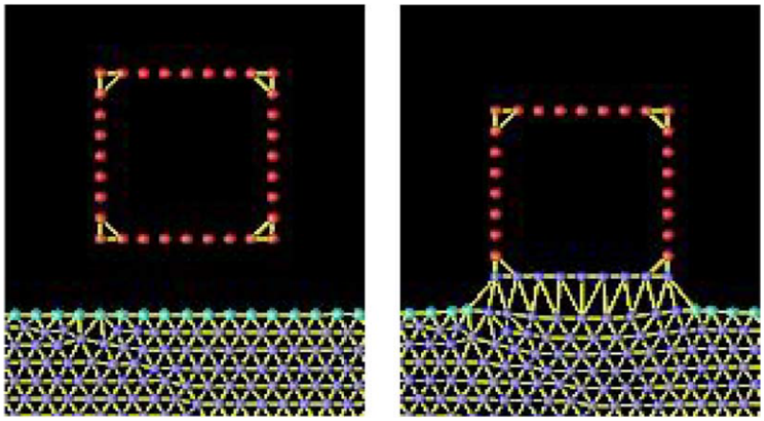

Figure 4.2: Interface mesh generation and contact detection at each time step [30].

INTERFACE MESH GENERATION

In order to treat contact, the contact elements are automatically generated during the remeshing procedure. After the steps of section 3.2 are completed and the boundary particles of each domain are identified, another meshing procedure is carried out, using these boundary particles and any nodes on the rigid boundaries of the model. Identically to what is the case for the analysis domains, first a Delaunay Triangulation and then the alpha-shape method for deletion of distorted elements are performed; any elements that are preserved after this sequence constitute the "interface mesh". An example of contact anticipation is shown in Fig. 4.2. The created contact elements can be used either for contact detection or be incorporated in the solution domain.

FORMULATION

The main concept of the developed contact technique is to disallow movement towards the rigid boundary in the normal direction of contact. This is done by increasing the stiffness in said direction. As shown in Fig. 4.3, the contact face of a contact element is defined as the segment between the two nodes of the element belonging to the same "body", i.e. the domain or the rigid boundary surface at this case. Then, the normal vector is defined at the projection of the third node on the face. Essentially, the method imposes a "zero" prescribed displacement in the direction normal to the interface, which, in the cases examined herein, coincides with the rigid boundaries.

In order to calculate the contact element stiffness, the displacement field u for each contact element is projected on the direction of the normal to the contact faces. This can be written as

$$u^n = (u \cdot \overline{n}^1) \cdot \overline{n}^1 = Pu \tag{4.1}$$

where $\overline{n}^1$ is the unit normal vector to the contact surface. Following the notation proposed in [62] and adopted in [11], the superscript $(\cdot)^1$ denotes the body on which the unit normal vector is found and the overline $\overline{(\cdot)}$ indicates that it is calculated at the "minimum distance point". This last term is more applicable to curved interfaces; however,

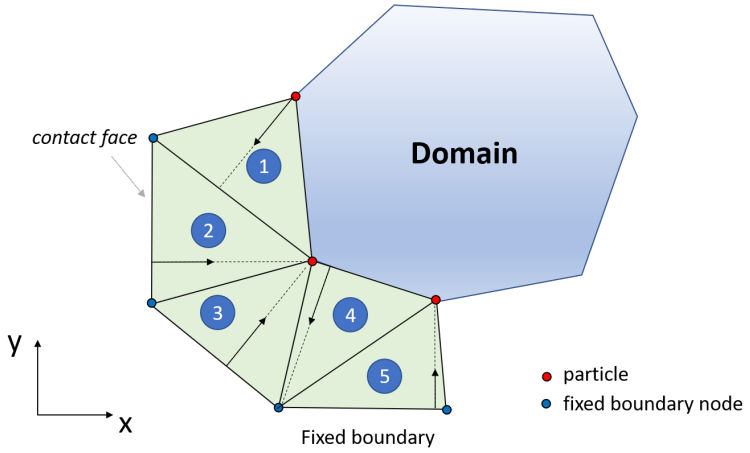

Figure 4.3: Sketch of contact elements in the interface between the domain and a rigid boundary surface.

here, the contact surfaces are divided into linear segments. For the 2D case, the unit normal vector is given by

$$\overline{n}^1 = \{n_x \; n_y\}^T \tag{4.2}$$

and

$$P = \overline{n}^1 \otimes \overline{n}^1 = \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} \{n_x \; n_y\} = \begin{bmatrix} n_x^2 & n_x n_y \\ n_y n_x & n_y^2 \end{bmatrix} \tag{4.3}$$

The projected displacement can be written as

$$\{u^n\} = \begin{Bmatrix} u_x^n \\ u_y^n \end{Bmatrix} = P\{u\} = \begin{bmatrix} n_x^2 & n_x n_y \\ n_y n_x & n_y^2 \end{bmatrix} \begin{Bmatrix} \boldsymbol{u_x} \\ \boldsymbol{u_y} \end{Bmatrix} = \begin{Bmatrix} n_x^2 \boldsymbol{u_x} + n_x n_y \boldsymbol{u_y} \\ n_y n_x \boldsymbol{u_x} + n_y^2 \boldsymbol{u_y} \end{Bmatrix} \tag{4.4}$$

The 2D displacement field for a point inside an element with m nodes, using linear shape functions, is approximated as

$$\{u\} = \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & \dots & N_m & 0 \\ 0 & N_1 & \dots & 0 & N_m \end{bmatrix} \begin{Bmatrix} u_x^1 \\ u_y^1 \\ \vdots \\ u_x^m \\ u_y^m \end{Bmatrix} = [S]\{e\} \tag{4.5}$$

where vector e holds the nodal displacements. The projection of the approximated displacement field can be written as

$$
\begin{aligned}
\{u^n\} = P\{u\} &= \begin{bmatrix} n_x^2 & n_x n_y \\ n_y n_x & n_y^2 \end{bmatrix} \begin{Bmatrix} \boldsymbol{u_x} \\ \boldsymbol{u_y} \end{Bmatrix} = \\
&= \begin{bmatrix} n_x^2 & n_x n_y \\ n_y n_x & n_y^2 \end{bmatrix} \begin{bmatrix} N_1 & 0 & \dots & N_m & 0 \\ 0 & N_1 & \dots & 0 & N_m \end{bmatrix} \begin{Bmatrix} u_x^1 \\ u_y^1 \\ \vdots \\ u_x^m \\ u_y^m \end{Bmatrix} = \\
&= \begin{bmatrix} n_x^2 & n_x n_y \\ n_y n_x & n_y^2 \end{bmatrix} \begin{bmatrix} N_1 & 0 & \dots & N_m & 0 \\ 0 & N_1 & \dots & 0 & N_m \end{bmatrix} \{e\} = \\
&= \begin{bmatrix} n_x^2 \boldsymbol{N_1} & n_y n_x \boldsymbol{N_1} & \dots & n_x^2 \boldsymbol{N_m} & n_y n_x \boldsymbol{N_m} \\ n_x n_y \boldsymbol{N_1} & n_y^2 \boldsymbol{N_1} & \dots & n_x n_y \boldsymbol{N_m} & n_y^2 \boldsymbol{N_m} \end{bmatrix} \{e\} = \\
&= [S^n]\{e\}
\end{aligned}
\tag{4.6}
$$

The stiffness matrix of the contact element is constructed in the same way as the stiffness matrix for a usual, domain element (Eq. 2.11). The only difference is that a modified strain-displacement matrix constructed with the derivatives of the terms in $[S^n]$ and a special constitutive matrix D_{cont}, which is constructed with v = 0 and the Young's modulus of the contact element, are used.

$$
[k]_{cont} = [B^n]^T [D_{cont}][B^n] det(J) w \tag{4.7}
$$

The above relationships "distribute" the stiffness of the contact element to the x and y components accordingly. As an example, a contact element with a completely horizontal contact face, e.g. element 4 in Fig. 4.3, will contribute only in the stiffness along the y degree of freedom of the associated domain particle; similarly, element 2 will contribute only in the x direction.

The stiffness of the contact elements needs to be significantly higher than that of the domain elements, in order to ensure no-penetration. Several approaches were tested, but

$$
E_{contact} = (E_{domain})^{\nu} \tag{4.8}
$$

with $\nu \geq 2$, proved to be the most efficient.

Despite the limitation of implicit time integration schemes for dealing with impact problems, the contact treatment algorithm can be adjusted to address these phenomena. More specifically, an adaptive time step may be introduced, which will take into account the velocities in the domain and a characteristic contact dimension, e.g. the "contact detection clearance". Then, during the simulation, an iterative procedure may be utilized for examining whether impact is about to happen and decrease the time step accordingly for as long as collision takes place. Furthermore, in order to take into account the relative velocity and the different stiffness of the colliding bodies, a different relationship for the stiffness of the contact elements can be used [11]. This adaptive stiffness modulus can also help as a means of damping for the dynamic oscillations that may occur during the highly non-linear event of body impact.

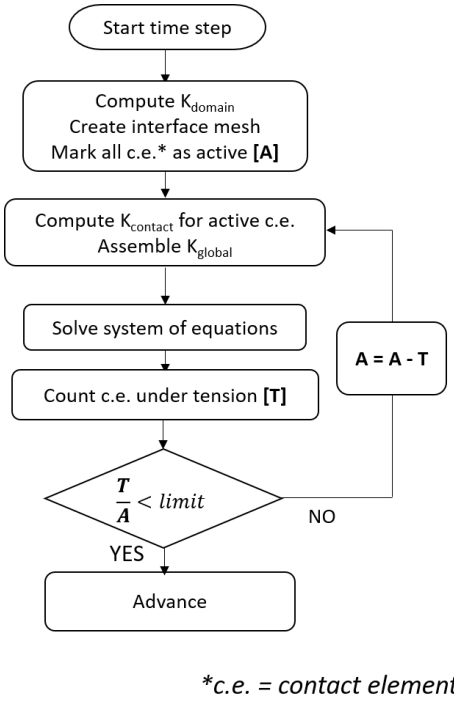

Figure 4.4: Flowchart for Active Contact Loop.

ACTIVE CONTACT LOOP

In order to utilize the contact elements and enforce the contact constraints, a so-called "active contact loop" is performed, based on the concept presented in [11]. The flowchart of the scheme is shown in Fig. 4.4 and a sketch of the concept in Fig. 4.5. The two subsets of elements will be explicitly referred to as "domain" and "contact" elements.

Initially, at the beginning of each time step, the global stiffness matrix is assembled for the domain elements and the interface mesh is generated, as already described. At first, all contact elements are marked as "active", meaning that they contribute in the global stiffness matrix and are included in the **A** set. At the beginning of each contact iteration, the stiffness contribution of all active contact elements from **A** is calculated and added to the global stiffness matrix. Then, the system of equations can be solved, in a standard manner. After the solution is obtained, the stresses at all the contact elements are retrieved and the ones under tension are flagged and sent to the subset **T** (red in Fig. 4.5). As the contact elements used in this method are only allowed to work under compression, i.e. to block penetration, any contact elements belonging to **T** are discarded for the next contact iteration and they do not contribute in the global stiffness; only the green elements will be considered. The procedure is repeated until the compressed contact elements are above a limit percentage, here taken as 95%.

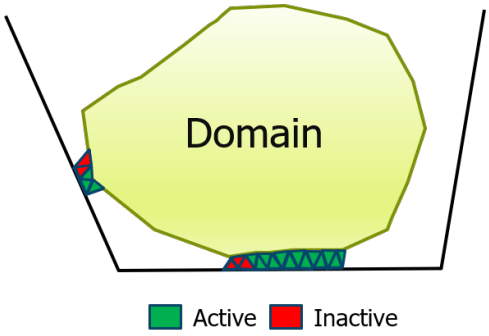

Figure 4.5: Sketch for Active Contact Loop

VALIDATION

At this point, the proposed contact implementation is validated against some simple benchmark problems, for which either analytical or numerical solutions are available. The scope of this step is to prove that the presented formulation for contact treatment is suitable for normal and tangential movement with respect to the rigid boundaries. These examples will be also used for validating the implicit formulation of the governing equations presented in section 2.2.2.

1D Column Compression The first problem concerns the one-dimensional compression of a column and compared to a UL-FEM solution [57], that is considered capable of providing good results for large deformations, albeit small strains. Replicating the simulation parameters of the benchmark solution, so that a direct comparison may be possible, the column is 10 meters high and 1 m wide, plane strain conditions are assumed and the material of the column is taken to be linear elastic, with a Young's modulus of 10 MPa and a Poisson's ratio v = 0.3. Moreover, a total load of 10 MPa is applied at the top, in 40 increments of equal load, i.e. 0.25 MPa per step.

The initial configuration of the particles is shown on the left in Fig. 4.6. A total of 800 particles are used in the analysis, a discretization coarse enough to facilitate the demonstration of the movement of the domain boundaries and interface and fine enough to not lead to severe distortion in the contact elements. In the center and right of the same figure, snapshots of the column at different times in the simulation are given, with close-ups of the interface elements. It should be noted that the results are depicted at the end of the respective load step, which is why the contact elements appear distorted. In general, the interface mesh is reconstructed at the end of each step. The configuration of the column at the end of the analysis is shown on the right, where the column height has been reduced by more than half and the particles are densely positioned close to the fixed end.

The agreement of the load-displacement curve in Fig. 4.7 with the UL-FEM results proves the ability of contact elements to allow particles to move freely perpendicular to their face, while disallowing any normal movement.

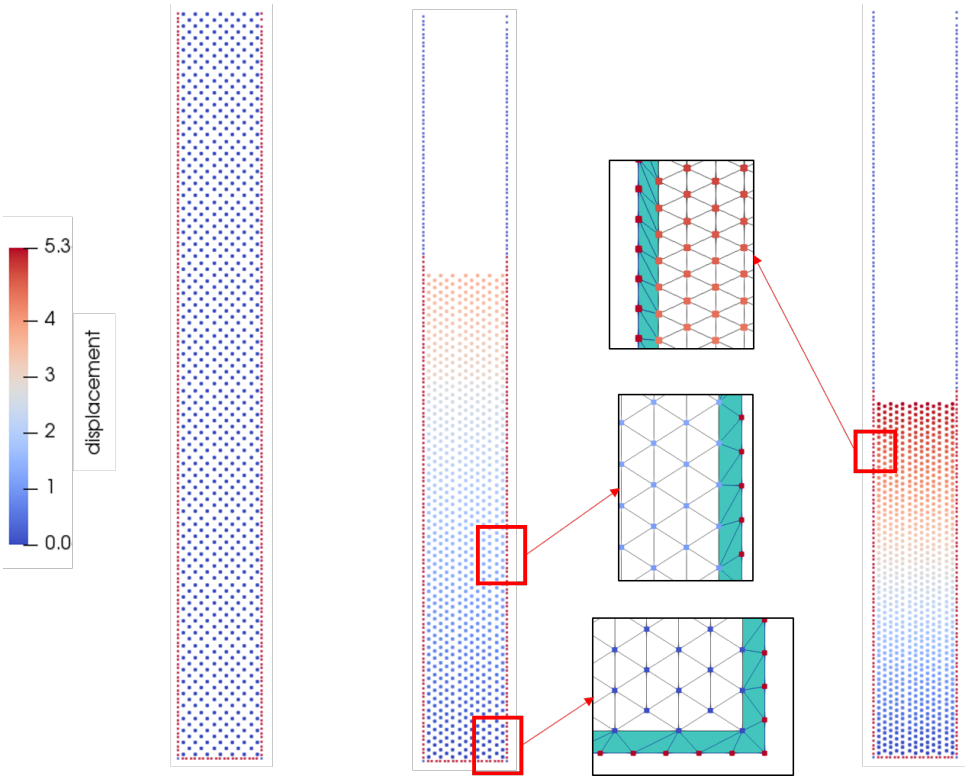

Figure 4.6: Different snapshots during 1D compression of a column, using the implemented contact formulation of PFEM. Closeups of the contact elements that allow free tangential movement are also shown.

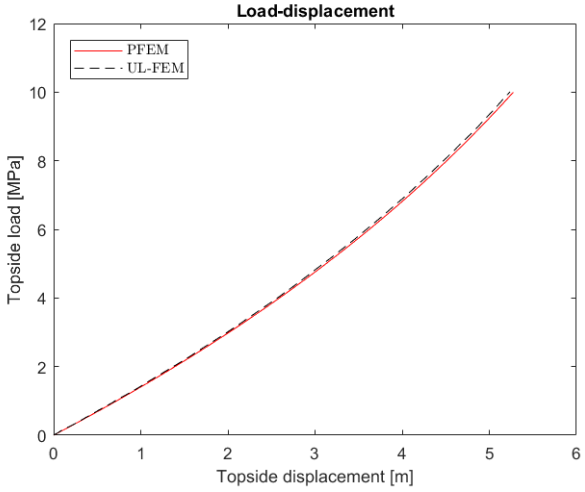

Figure 4.7: Load displacement curve for 1D compression of a column, using the implemented contact formulation of PFEM.

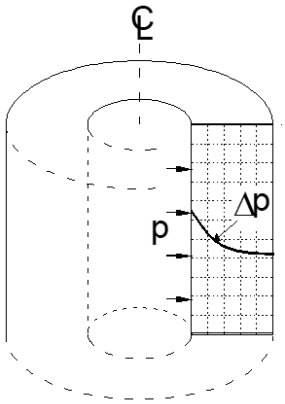

Figure 4.8: Sketch of the axisymmetric cylinder submitted to outward displacement at its inner wall.

Axisymmetric Cylinder The second quasi-static problem examined is the axisymmetric cylinder with a fixed outer wall subjected to outward displacement at its inner wall (Fig. 4.8). The analytical solution is available for this problem; for a given displacement

$$u = C_1 r + C_2 \frac{1}{r} \tag{4.9}$$

The radial stresses can be recovered by

$$\sigma_{rr} = +A\frac{1}{r^2} + 2C \tag{4.10}$$

where

$$A = -\frac{EC_2}{1+\nu} \tag{4.11}$$

$$C = -\frac{EC_1}{2(1+\nu)(1-2\nu)} \tag{4.12}$$

An outward prescribed displacement is applied at the inner wall of the cylinder in 20 steps of 0.0226 m, summing up to a total of 0.47 m. The initial particle distribution inside the cylinder is shown on the top part of Fig. 4.9; half of the total displacement is applied at the middle snapshot and the bottom one shows the accumulation of particles towards the outer wall, at the end of the analysis. In all three cases, the red points indicate the fixed boundary nodes, on the basis of which the interface elements (green) are constructed. The stress evolution inside the cylinder for this two moments in the simulation is plotted in Fig. 4.16, in terms of the radial stress across a cross section at the middle height of the body. The retrieved stresses show a very good agreement with the analytical solution, indicating that the contact elements do not interfere with the outcome of the analysis at all.

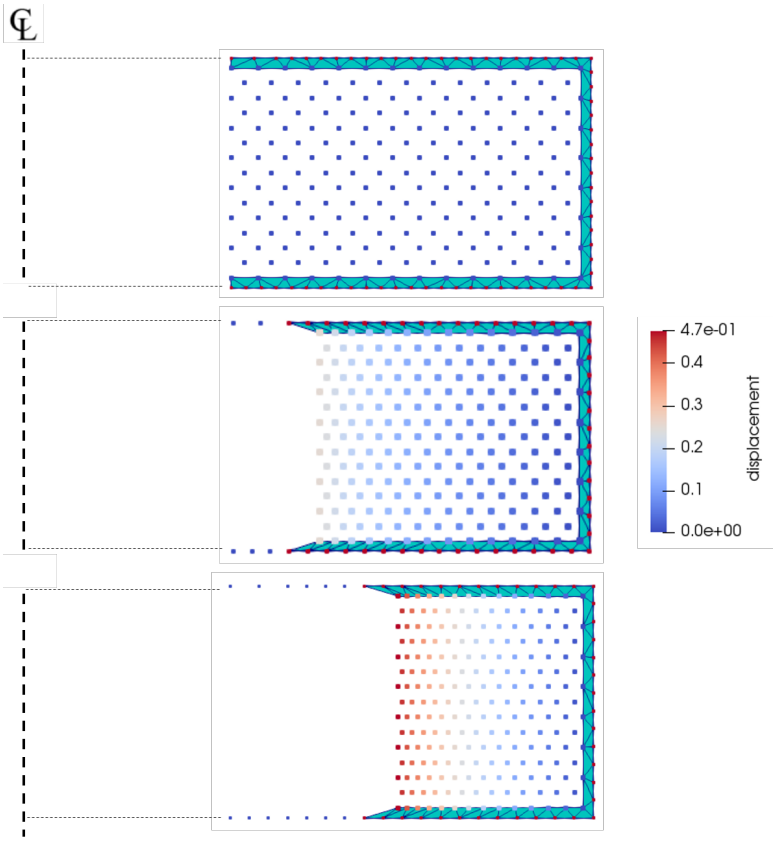

Figure 4.9: Snapshots during the compression of a thick wall axisymmetric cylinder.

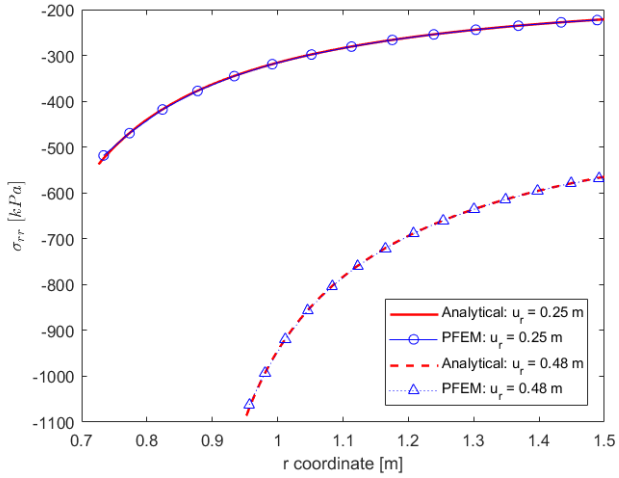

Figure 4.10: Comparison of numerical and analytical radial stresses at a cross section across the cylinder, for two different displacements at the inner wall.

Axial Vibration of a Continuum Bar The first two cases concerned quasi-static applications. The free vibration of a continuous bar is considered at this point. The bottom end of the bar (x = 0) is fixed, the top end (x = L) is free and an initial velocity is prescribed in the bar. The eigenvalue for normal modes of oscillation is given by

$$\beta_n = \frac{(2n-1)\pi}{2L}, \quad n = 1,2,3,\ldots \tag{4.13}$$

The frequency of oscillation is related to the eigenvalue and is given by

$$f_n = \frac{\beta_n}{2\pi} c_v \tag{4.14}$$

where c_v is the wave speed and is a function of the material stiffness E and density ρ

$$c_v = \sqrt{\frac{E}{\rho}} \tag{4.15}$$

The initial velocity along the bar is given by

$$v(x,0) = v_0 sin(\beta_n x) \tag{4.16}$$

Then, the analytical solutions for the displacement and the velocity are given in the next two equations, respectively

$$u(x,t) = \frac{v_0}{\omega_n} sin(\omega_n t) sin(\beta_n x) \tag{4.17}$$

$$v(x,t) = v_0 cos(\omega_n t) sin(\beta_n x) \tag{4.18}$$

The first mode of oscillation is examined here, for which the eigenvalue and frequency are given (for n = 1) by

$$\beta_1 = \frac{\pi}{2L} \tag{4.19}$$

$$f_1 = \frac{c}{2\rho L} \tag{4.20}$$

A sketch of the examined problem and the initial velocity distribution for the 1st mode are depicted in Fig. 4.11.

The height of the bar is L = 1 m and its width is b = 0.1 m. The bar has a Young's modulus E = 100 kPa and a density $\rho = 1\ tn/m^3$. An initial downward velocity of $v_0 = 0.25\ m/sec$ is prescribed at the top. The simulation is run for three different time steps, $\Delta t = 0.1\ sec$, $\Delta t = 0.01\ sec$ and $\Delta t = 0.001\ sec$. Regarding contact elements, the bottom elements are not considered in the active contact loop (4.1.2.3), and, are, instead, taken to be active throughout the simulation.

A total of 100 particles are distributed in the domain. Focus is put on the velocity at particle A (free end) of the bar, which is compared with the available analytical solution, and at the stress evolution at an integration point B (fixed end) of the bar, with the two points shown in the sketch of Fig. 4.11.

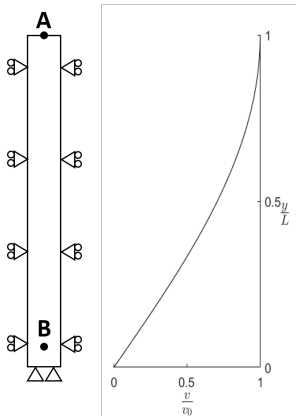

Figure 4.11: Sketch of the bar under axial vibration and distribution of the initial velocity for the 1st mode of vibration.

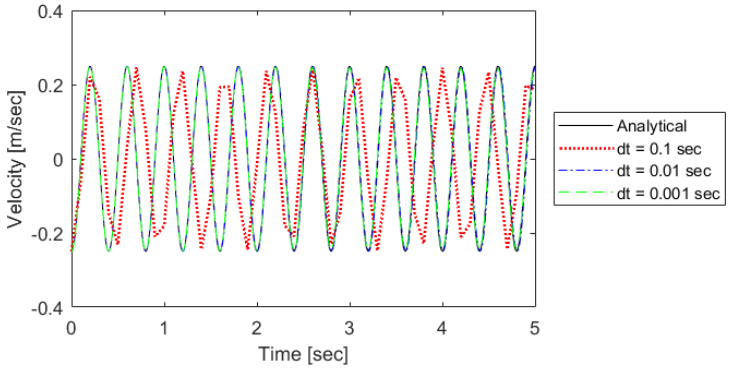

Figure 4.12: Velocity over time at point A.

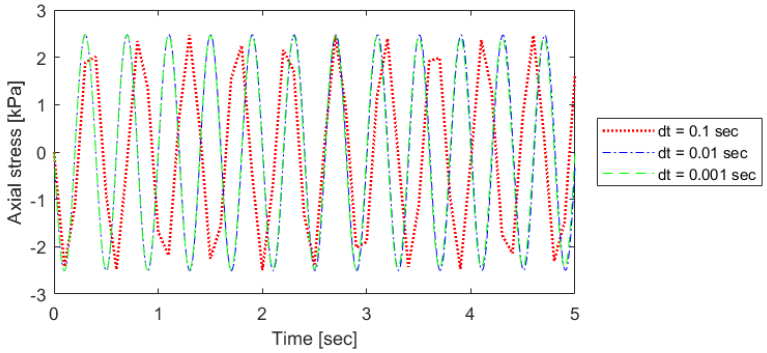

Figure 4.13: Axial stress evolution at point B.

From Fig. 4.12, it appears that the velocity for $\Delta t = 0.01$ *sec* and $\Delta t = 0.001$ *sec* agrees with the analytical solution perfectly. For the largest time step, the velocity at the free end of the bar shows a completely different frequency from the analytical solution. Similar behavior is observed for the stresses at point B, shown in Fig. 4.13. Even though there is not an analytical solution available for the stresses, the results for $\Delta t = 0.01$ *sec* and $\Delta t = 0.001$ *sec* demonstrate an agreement with each other, while for $\Delta t = 0.1$ *sec* the frequency and amplitudes of the oscillation are in disagreement. Nonetheless, it appears from the above results that the developed contact algorithm is suitable for dealing with simple dynamic problems.

The results from the three problems examined in this subsection indicate that the implicit formulation presented in 2.2.2 and the adopted contact implementation are capable for simulating linear elastic system responses for simple applications, encouraging their use for other applications in the report. More advanced contact models can also be formulated to capture more complex interaction scenarios and be used on the basis of the constructed interface mesh; additionally, the contact algorithm can be manipulated to consider body-body interaction as well, which is not, however, in the scope of this work.

4.2. INFORMATION MAPPING

4.2.1. INTRODUCTION

Another main differentiation between modelling of fluids and solids in the Particle Finite Element Method can be found in the way that the information is handled. In the early implementations of the method, where fluid motions were mainly simulated, the interest was on nodal variables, such as velocity, pressure, etc. These are computed at the nodes of the FEM mesh, which coincide with the particles, therefore, storing them and transferring between meshes is straightforward.

For solid mechanics problems, however, when usually non-linear material behavior is modelled, the handling of elemental variables becomes also relevant. This is because the history of stresses, strains, constitutive variables and material properties, which are computed inside the element, hence, termed internal, is important for correctly capturing the behavior of solid materials. Therefore, a proper technique for storage and transference of this information that minimizes the errors in the accuracy of the method is required. This is a crucial and controversial matter for the PFEM, judging by the lack of a unique operation in literature. At this point, a study on the most popular information handling schemes used by researchers is performed, along with a brief investigation on the efficiency of some of them.

4.2.2. NODAL SMOOTHING

The first [10] approach, which stills remains the most popular [11, 13, 44, 45, 51], was to stay in the spirit of PFEM and follow the convention used typically in particle methods, where all the information is stored in the particles. According to this technique, termed *nodal smoothing*, the stresses (and all other internal variables) are calculated at every step in the Gauss points and the values are mapped back to the particles, in an averaged sense, in order to generate a continuous stress field. The scheme is shown left in Fig. 4.14.

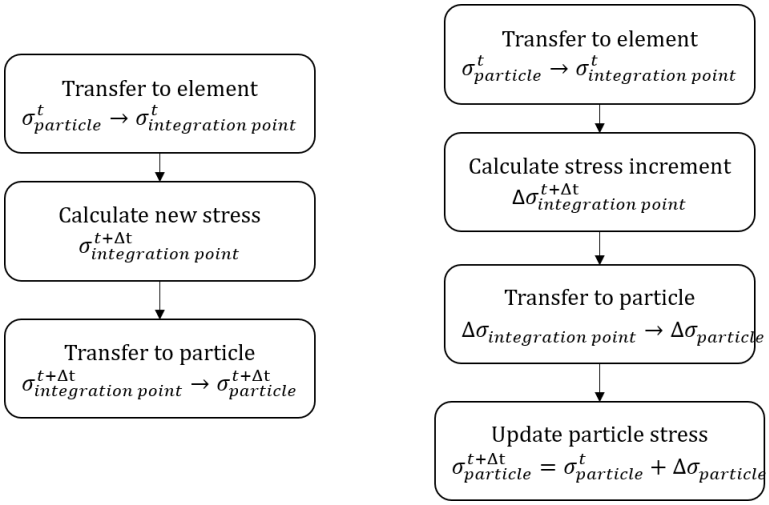

Figure 4.14: Flowchart of the nodal smoothing operation (particles - elements - particles). Left: original scheme. Right: modified scheme.

This is expected to introduce some smoothing in the information, trimming extremely low or high values.

Researchers identified that this high degree of smoothing can be reduced by modifying the initial scheme. After the initial mapping of the nodal stresses to the integration points, assembly and solution of equations, instead of transferring the newly computed stresses at the elements to the particles, only the averaged increments of the stresses (or any other relative variable) are mapped back to the particles. The flowchart for this modified transfer scheme is shown on the right of Fig. 4.14. The error that accumulates through the analysis is limited to the smoothing of the stress (or any other variable) increment. Another drawback of the method, besides smoothing, is that the mapping is performed everywhere, even in regions where the mesh does not change, leading to the equilibrium previously achieved being disturbed.

In order to showcase the smoothing of information, the axisymmetric cylinder compression problem is examined. A total displacement of 0.0226 m is applied at the free side of the cylinder, either in one step or in 10 equal increments. The problem is simulated with two different discretizations, one with 39 particles and one with 137, shown in Fig. 4.15, on the left and right, respectively. In each case, focus is put on the evolution of the radial stress σ_{rr} on a cross section in the center of the cylinder and on two Gauss points, the one closest to the fixed wall and the one closest to the inner wall, labeled A and B.

The radial stresses at the mid-height cross section for the 1-step scenario are plotted in Fig. 4.16, for both discretizations. The blue curve shows the calculated stress right after solution and the purple curve corresponds to the recovered stresses after two-way mapping, i.e. the increments are mapped to adjoint particles and then mapped back to the Gauss points.

The stresses at the free end (point A), which are the highest in the domain, show a significant decrease, as a result of the smoothing taking place during the mapping pro-

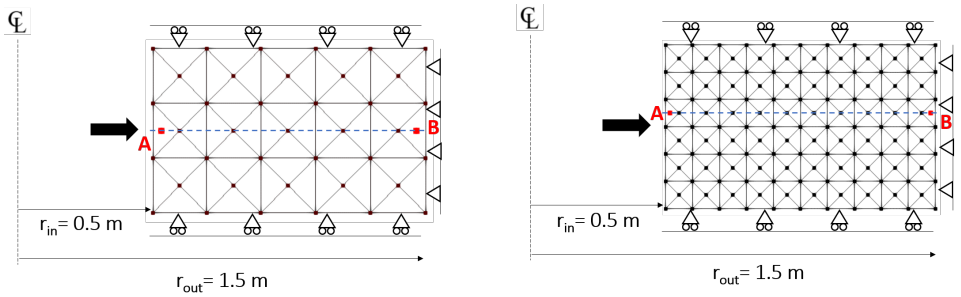

Figure 4.15: Two different discretizations for the axisymmetric cylinder compression.

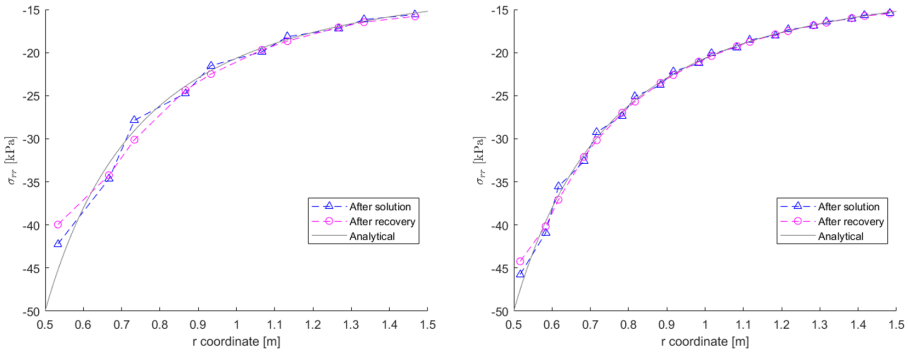

Figure 4.16: Radial stresses at the integration points after solution and after nodal smoothing, for two different discretizations for the axisymmetric cylinder compression. Left: 39 particles. Right: 137 particles.

cedure, increasing the error compared to the analytical solution. More specifically, for the coarser mesh, the recovered stress shows a decrease of -5.3% from the computed and a difference from the analytical of -11.8%. Close to the fixed end, the respective values are 1.3% and 2.5%, indicating an increase in the minimum stress.

For the finer discretization, at the right graph, the smoothing effect is partly reduced, but still evident, especially where stresses are higher. After the displacement is applied in one step, the recovered radial stress at the free end is 3.3% lower than the initially computed value and -6.9% from the analytical solution. Close to the fixed end (point B), the recovered stress is increased by 0.6% compared to the initial one and is 1.1% higher than the analytical. However, in both cases, the smoothed line shows a better approximation to the analytical solution towards the middle of the cylinder

The plot in Fig. 4.17 shows the development of the deviation of the computed stresses from the analytical solution close to the free end (point A), for the case of constant mapping and the case where no mapping takes place, i.e. the historical stresses are stored at the Gauss points. In the latter case, the accumulated radial stress is underestimated with a constant rate around 3.5%, which can be attributed to the use of constant strain triangles and a rather coarse mesh; the former case shows a greater error in the built-up stresses, as a result of the smoothing of extreme values during nodal mapping. It should be noted that although the percentage remains constant for higher values of stress, the difference in absolute terms increases.

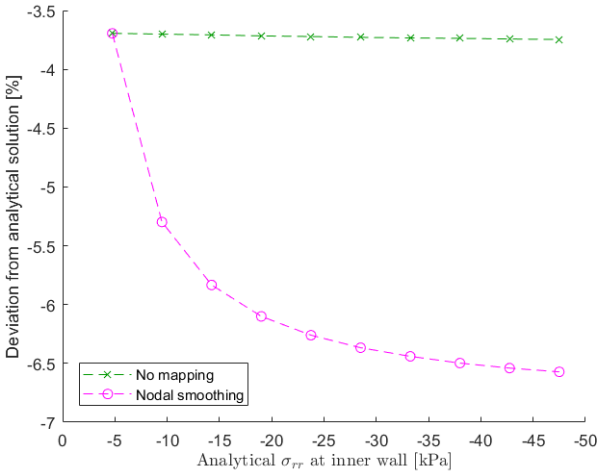

Figure 4.17: Deviation of the computed radial stresses from the analytical solution with and without nodal smoothing.

From the example presented above, it becomes apparent that a certain amount of information smoothing takes place when the stresses are stored in the particles, which is expected to have further implications in the solution when the recovered stresses are used to compute the internal forces. However, the smoothing effect is mitigated for finer discretizations, making the "nodal smoothing" scheme the most preferable in solid mechanics applications with PFEM. Nonetheless, the impact of this operation will be further studied in this work.

4.2.3. MESH PROJECTION

The "nodal smoothing" scheme presented in the previous subsection has been the most popular in the implementations of PFEM for solid mechanics. However, in recent publications [19, 24, 35, 36, 49], a different approach has emerged, aiming in reducing the unavoidable perturbation of equilibrium and the computational cost; the latter refers to the fact that more computer memory space is required for storing the additional particle information.

According to this mapping rule, termed either *mesh projection* or *centroid transfer*, the computed information at the integration points is not related with the particles at all, but is, instead, transferred between integration points of the old and the new mesh directly. Every integration point of the new mesh inherits the information from the element of the old mesh with the nearest centroid to it. In contrast with the particle storage technique, the information in this approach is preserved for elements that have not been altered during remeshing, minimizing the introduced disturbance of equilibrium, as the transfer of information is limited only in areas of large deformation.

The computational cost of this method comes from the tool used for finding the closest centroid to a new integration point. Optimized algorithms, instead of a *brute force* approach, can reduce the required times significantly. In any case, if the "mesh projec-

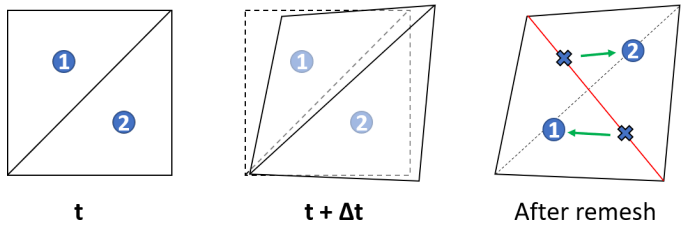

Figure 4.18: Example of edge flip during remeshing, causing perturbation in the previously achieved equilibrium.

tion" technique is used, the transference needs to take place every time a new mesh is created, which may not occur at every time step, making this scheme more attractive in computational resources terms.

An issue with the method is the handling of information for deleted or newly inserted integration points. As an example, in the notch problem examined in 3.2.2.1, when the elements close to the notch are deleted during remeshing because of the α-shape method, the stresses (and any additional information) at the integration points of these elements will be wiped from the analysis. On the opposite case, if the specimen was under compression instead of extension, which would lead to the notch closing, then new elements would be inserted in the domain. These elements would inherit information based on their closest existing elements of the previous mesh, which would lead to disturbance in the equilibrium.

Another common occurrence observed in the "mesh projection" scheme is the edge flip, depicted in Fig. 4.18, where a two-triangle configuration is examined. After the solution of the equations, the derived deformed state is seen in the middle of the figure, with the node connectivity defined at the beginning of the time step. If a remeshing procedure takes place, the Delaunay triangulation will flip the internal edge of the triangles, in order to maximize the minimum angles of the mesh (3.1.3.2). Then, the history of the variables in the new integration points is inherited from the close centroid of the two deformed, old elements. Subsequently, this flip of edges and information will disturb the equilibrium that was achieved at the end of the previous time step and will be manifested in the calculation of internal forces. This phenomenon cannot be avoided, but its effect is strictly local and can be reduced, if the remeshing frequency is moderated.

4.2.4. OTHER METHODS

LEAST INTERPOLATION TECHNIQUE

Besides the "mesh projection" technique presented above, [35] proposed another transfer operator, which makes use of a least interpolation procedure. According to this approach, the value of a certain variable at a new integration point in the new mesh is computed based on the values at elements of the previous mesh which overlapped the same area. In other words, the new value is the average of the elements that overlapped its position, averaged in terms of the overlapping area. This method has two drawbacks; firstly, the newly calculated stresses may be outside the yield surface, which would require a yield correction algorithm, increasing the complexity and computational cost of the scheme; the second disadvantage is, as in "mesh projection", that the equilib-

rium reached at the previous time step is violated, which, however, the authors explicitly choose to ignore.

UNIQUE ELEMENT METHOD

The *Unique Element Method* (UEM), initially proposed in [28], has been employed for the transfer operation in [66]. For this method, first, for each new integration point the host element of the old mesh is found. Then, the information is transferred from the Gauss points of that element in terms of distance-based weighted contributions. This method is reported to have quite good accuracy, but requires more than 1 integration points to be effective. For the single-integration point elements used in the PFEM formulation of this work, the UEM essentially degenerates to the "mesh projection" operation (4.2.3).

SMOOTHED PARTICLE FINITE ELEMENT METHOD

The principles of the PFEM and the Smoothed Finite Element Method (SFEM) [64], were combined in [65] to develop the *Smoothed Particle Finite Element Method* (SPFEM). In SFEM, the strain (and, consequently, the stress) field is recovered by splitting the domain into so-called smoothing domains and applying a strain smoothing technique, i.e. after partition of the domain into "smoothing" subdomains and with introduction of the divergence theorem, a (constant) strain is computed for each smoothing domain based on the values of the shape functions (not their derivatives or gradients) and the displacement on points on the boundaries of the smoothing domains. This means that Gauss integration and stress recovery are redundant and that lower quality mesh and severe mesh distortion are allowed. Thanks to the use of smoothing sub-domains in this technique, all the information is calculated in the particles, meaning that there is no need for any mapping of information that reduces the accuracy of the results. However, given that it is a very young implementation, there is not enough evidence further supporting the efficiency of the method in large deformation problems.

4.3. MASS CONSERVATION

4.3.1. INTRODUCTION

As in every numerical tool for simulation of physical problems, the conservation of mass throughout the analysis is a fundamental issue and must be considered with care. As opposed to other particle methods, such as the Material Point Method, were a fixed mass is assigned to the particles at the beginning of the simulation and, thus, conserved naturally, in the Particle Finite Element Method, the particles are given a density at the start of the analysis. Then, the total mass is evaluated at each time by integrating the density over the domain

$$M = \int \rho \mathrm{d}V \tag{4.21}$$

This allows for constructing areas with different particle refinement, in contrast with MPM, where the material points must be positioned with care in order for the mass to be distributed evenly inside the domain. On the downside, in its original form, the PFEM scheme does not guarantee mass conservation exactly, as mass variations are introduced by unwanted oscillations in the computed density or changes in the domain volume,

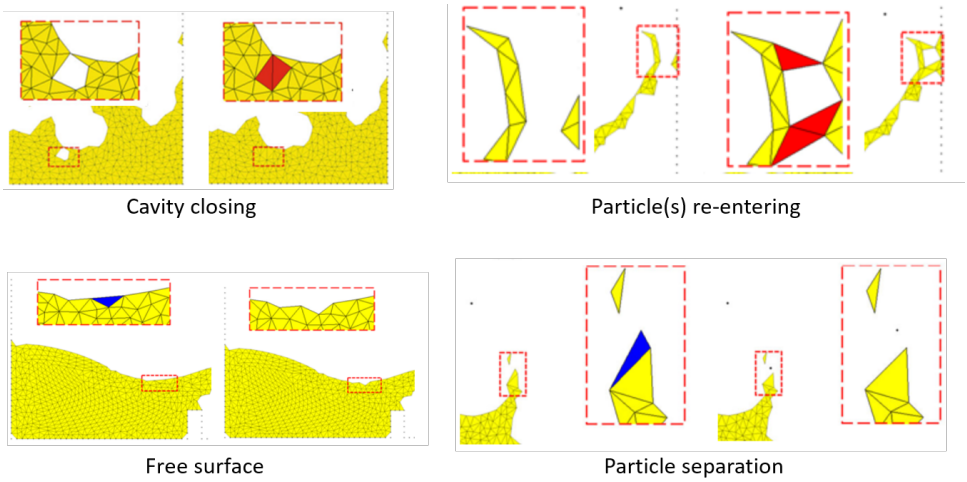

Figure 4.19: Causes of mass oscillation during re-meshing (after [23]).

either by numerical reasons or generated during remeshing. At this point, focus is put on the latter.

When all the elements that are associated with a certain particle are deleted during a remeshing step, then all the mass of these elements is assigned to the particle, which is let to move based on its current velocity and subject to body forces. When the particle rejoins the domain, the mass that is added to the main domain is not precisely the mass of the particle, but is, instead, the mass of the newly created elements that connect the particle with the domain; these two phenomena compensate each other in such a way that the total mass variation is insignificant [17]. However, these variations have a toll on the conservation of momentum as well, as the momentum that is added to the system when the particle re-attaches is not the same the particle had when moving on its own.

In the following, the differences between FSI and solid mechanics problems regarding mass conservation are presented.

4.3.2. FLUID MECHANICS

In fluid mechanics modelling with PFEM, there are three main causes of mass conservation violations. Firstly, as usually incompressible fluids are modelled, ρ is kept constant. However, the way of enforcing the incompressibility constraint for fluid flows may contribute to mass conservation violations, usually mass reduction [14]. In view of overcoming this discrepancy, stabilization terms may be introduced in the equation of mass conservation [71].

Moreover, the main reasons that introduce volume variations in PFEM during remeshing have been listed in [23], some of which are displayed in Fig. 4.19. The volume variations practically occur due to element creation and deletion and are directly related with the choice of the α_{lim} value for the alpha-shape method (3.2.2), as was mentioned. As can be seen in Fig. 4.19, volume and, hence, mass is increased when new elements are created as cavities close and particles re-enter the domain. On the other hand, mass is lost when particles separate from the domain or from local distortion of the free sur-

face. The average of these fluctuations results in volume increase most of the times, as reported by the authors.

Finally, in fluids, as already discussed, contact is treated by introducing fluid elements at the contact interface. These elements behave as regular fluid elements, with mass, which means that, when a fluid and a solid body come close enough to generate contact elements, mass will be added to the system; on the opposite, when the bodies drift away and the contact elements are discarded during interface mesh generation, the system mass will be decreased. These oscillations in the system mass can affect the solution, but monitoring and filtering them throughout the simulation is quite straightforward. However, the authors conclude that these problems tend to decrease with finer meshes and a correct selection of the α_{lim} parameter.

4.3.3. SOLID MECHANICS

In solid mechanics, the density is updated based on the mass conservation law, which in Lagrangian description is given by

$$\rho = \frac{\rho_0}{det(F)} \tag{4.22}$$

where

$$F = I + (\nabla u)^T \tag{4.23}$$

is the deformation gradient. After the equations are solved and the displacements are obtained, the deformation gradient at each integration point can be computed and used to update the Gauss point density. In this context, the choice of the transfer process of Gauss points variables, including density, (see 4.2) could play an important role in mass conservation [38]. This factor, however, becomes less crucial for soil mechanics applications, since the highest degree of mesh distortion occurs when shearing takes place and the material reaches incompressible critical state conditions, thus, changes in density are less evident.

Overlooking the effects of density, the emphasis here is put on volume variations. Although the examples in Fig. 4.19 refer to fluid modelling, similar phenomena may occur in solid modelling. However, these phenomena are more intense in fluid modelling, which is logical as it is more possible for "water drops" to separate and re-enter in the domain, and cavities to be created. In solid mechanics, volume variations are mainly caused by changes in the domain boundaries during remeshing, as in the example at the bottom left of Fig. 4.19. As already mentioned in 3.2, a constrained Delaunay triangulation has been proposed [19, 49] in order to eradicate such changes in the domain boundaries. Overall, the mass conservation issue is merely addressed by the authors that use PFEM for solid mechanics. As an exception, a reference is made in [66], where the author suggests that the volume oscillations due to changes in the free boundaries can be mitigated by updating the α_{lim} accordingly at each remeshing step. More particularly, the proposed relationship is

$$\alpha_{lim}^n = \frac{m^0}{m^n}\alpha_{lim}^0 \tag{4.24}$$

where, m is the total mass in the domain and the superscripts $(\cdot)^0$ and $(\cdot)^n$ denote the values are assessed at times t_0 and t_n, respectively. The author employed this scheme for the numerical simulation of a real-case landslide and reports that the total mass did not demonstrate a variation greater than 2% throughout the analysis.

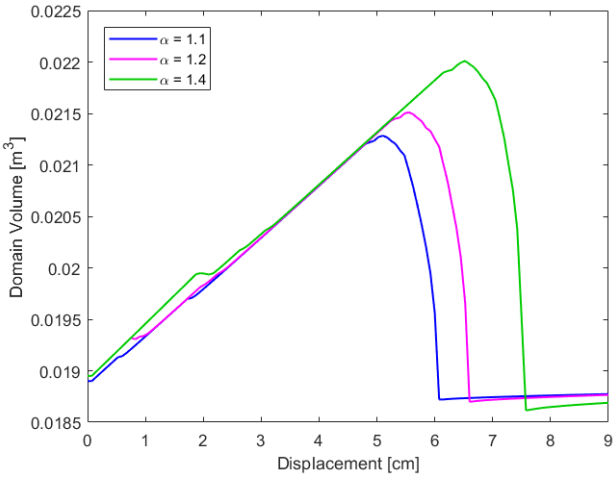

Figure 4.20: Volume variations during stretching of the notched specimen, for different α_{lim} values.

The effect of the α_{lim} parameter in the system volume during remeshing can be witnessed in the notched specimen extension examined in 3.2.2.1. The plot in Fig. 4.20 indicates the different evolution of the domain volume throughout the analysis, based on the selection of the mesh distortion threshold value.

The evolution of the domain volume follows a similar trend with the reaction force (Fig. 3.10). Initially, for alpha $\alpha_{lim} = 1.2$ and $\alpha_{lim} = 1.4$, the volume is slightly higher, due to the elements on the notch, which are deleted for $\alpha_{lim} = 1.1$ at the initial triangulation at step 0. The higher values of α_{lim} lead to a prolonged extension of the specimen, which results in a higher degree of volume increase, for a longer period. When the specimen starts splitting, the domain volume starts to decrease, due to the deletion of those highly distorted elements. When all of the connecting elements are gone, the volume suddenly drops and is stabilized at a value slightly lower than the initial; this is due to the volume of the elements at the notch being removed from the domain during the simulation.

5

COMPARISON WITH THE MATERIAL POINT METHOD

The Material Point Method and the Particle Finite Element Method are the two most popular mesh-based particle methods, going through their 3rd and 2nd decade, respectively, since they were first introduced. Ever since, they have been employed for various applications, with MPM, however, being more widely used. The two methods share two common, fundamental features. First and foremost, the information is assigned, stored and tracked in certain points, labeled particles or material points, the motion of which is followed throughout the analysis. The second feature is that a typical Finite Element formulation is employed and a standard FE mesh is used to solve the equations of motion and update the positions and other, relevant variables of these particles. On the other hand, there are some crucial differences between them, mainly concerning the computational cycle at each time step, the computation of derivatives and the treatment of information. These special characteristics are the main topic of discussion in this chapter.

The first part of the chapter includes a brief presentation of the Material Point Method, with some brief comments on its advantages and drawbacks. Then, the implementation of MPM that is available in the Geo-Section of TU Delft and was used in this work is presented. This is based on a standard MPM formulation, but some modifications are also examined. In the main part of the chapter, a comparison between the two methods is performed. This concerns a brief discussion on the fundamental processes of the two methods, followed by a comparison on the results yielded by the two methods for basic solid mechanics applications. Finally, an investigation on the computational cost of each method is performed and some suggestions are made.

5.1. MATERIAL POINT METHOD

5.1.1. GENERAL

A brief presentation of the Material Point Method was done in 2.1. The method can be described as a hybrid Eulerian–Lagrangian approach, which uses moving material points to represent the material under investigation and the nodes on a background mesh for computations. At the beginning of the analysis, initial values of position, velocity, mass, volume and stress (and any other required internal variable) are assigned to the material points and are updated throughout the analysis. Moreover, depending on the nature of the simulated problem and material, additional parameters, such as temperature, pressure and others, can be also appointed to the particles. The background Eulerian grid covers the solid in its initial configuration and must be dimensioned to sufficiently include the whole region where the solid is expected to move. Moreover, the background grid also allows interaction of the body with other solid bodies or fluids.

Typically, every time step of the method involves the following processes, as shown in Fig. 5.1.

1. The variables are mapped from the material points to the background mesh grid.
2. The equations are solved in an Updated-Lagrangian fashion.
3. The material points are moved to their new positions and the mesh is reset back to its original form.

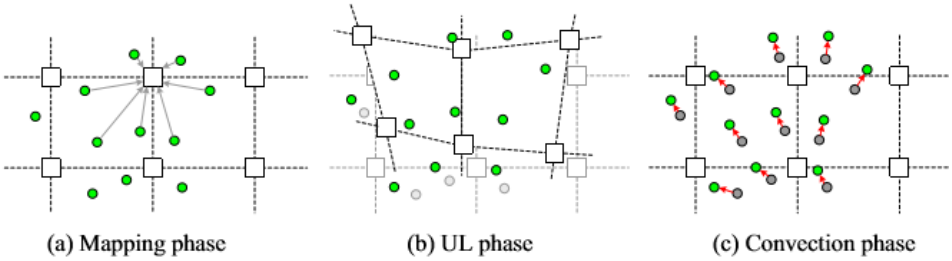

Figure 5.1: Computational cycle of MPM [56].

The Material Point Method has gained a lot of popularity in recent years, which can be easily justified by its significant merits. The primary advantage of MPM is that it can easily handle large deformation problems without any mesh tangling or distortion issues, thanks to the continuous reset of the background grid. Another important aspect is the implicit satisfaction of mass conservation, as the mass is appointed to the material points and remains constant throughout the analysis. Additionally, different material properties and constitutive models can be assigned to each material point, allowing for investigation of problems involving heterogeneity. Regarding contact treatment, interpenetration between different domains is excluded in the standard MPM formulation, since the velocity of each material point is single-valued. Moreover, during analysis, the discretization of the domain can be adjusted by adding or removing material points, as

there is no coupling between the particles and the Eulerian grid. Finally, the implementation of the method is not very burdensome, owing to the adoption of features from conventional FEM.

On the downside, the MPM lacks in efficiency compared to the FEM, due to the continuous backward and forward mapping between particles and nodes and the decreased accuracy of particle quadrature, i.e. evaluation of the integrals at the positions of the material points, compared to the Gauss quadrature in FEM. Additionally, stress oscillations are introduced during cell crossing of material points. This issue is partially mitigated by employing the Generalized Interpolation Material Point (GIMP) or the Composite MPM (CMPM) [27], which use special shape functions for assembly and stress recovery.

5.1.2. ADOPTED IMPLEMENTATION

The available implementation of the Material Point Method, which will be used for comparison with the PFEM, uses an implicit integration scheme [60]. In the available implementation, the initial geometry is generated based on the input from the user. In each element of the background grid, four material points are inserted in the positions of the Gauss integration points. Moreover, at the beginning of the analysis, a second set of elements is generated adjacent to the active mesh; this is done for covering the area on which the material points may migrate throughout the simulation.

FORMULATION

The formulation used for the general MPM is based on the Implicit FEM, presented in 2.2.2. The difference between the conventional FEM and the MPM is the way that the stiffness and mass matrices are assembled, where in the latter, the material points contribute in terms of a particle quadrature, instead of the standard Gaussian quadrature.

More specifically, a consistent mass matrix is assembled using the material points masses and standard FEM shape functions. Regarding the stiffness matrix, instead of Eq. 2.11, the contribution for each element in MPM is given by

$$[k] = \sum_{p=1}^{S_p} [B(x_p)]^T [D]_p [B(x_p)] W_p V_p \tag{5.1}$$

where x_p is the particle position, at which the strain-displacement matrix B is evaluated, W_p is the integration weighting of the particle, V_p the particle volume and S_p the total number of material points within the element. As the stiffness matrix is obviously dependent on S_n, overly stiff or soft elements may appear in cases of element crossing. In order to avoid such phenomena, a modification in the weighting coefficient [26] of the material points, based on the number of particles inside each element, is examined. The modified coefficients are given as

$$W_p = \frac{V}{\sum_{p=1}^{e_p} V_p} \tag{5.2}$$

where, V is the element volume and e_p is the number of material points inside the element. In the following analyses, both the standard MPM with unmodified weighting coefficients and the modified version will be considered.

STRESS UPDATE

Based on the calculated velocity, the material points are moved to their new positions, where the new stresses are calculated. This is called the Update Stress Last (USL) [8] approach. The alternative is the Update Stress First (USF) technique, in which the nodal velocities are first used to update the particle stresses, which are then used to compute the internal force vector and solve the equations of motion, before updating the particle positions. In general, USL shows a loss of energy in the system over time, while USF leads to energy increase.

5.2. COMPARISON

5.2.1. PRINCIPLES

The two mesh-based particle methods, MPM and PFEM, share a common trait, which differentiates them from the conventional FEM; the material properties are not attached to the elements, but are, instead, associated with the material points in the former and the particles in the latter. Another difference with the classical FEM is the way the mesh is treated; in MPM, the mesh is reset at the end of each calculation cycle, while in PFEM the nodal connectivity is reset and reassembled during each remeshing step.

A main differentiation between the Material Point Method and the conventional FEM is the location at which the (standard FEM) shape functions and their derivatives are computed, for calculations of the stiffness matrix and the internal load vector and recovery of the stresses. In the latter, this is done in the optimized Gauss point locations, while in the standard MPM formulation the sub-optimal current positions of the material points are used, using the same weighting coefficients for all points. This leads potentially to severe stress oscillations and discontinuities, and becomes crucial especially in cases where advanced constitutive models are used, related with variables sensitive to this kind of phenomena. These effects are intensified when loading is applied in increments and material points cross between cells.

Additionally, in dynamic applications with MPM, variables such as mass, velocity and momentum are mapped from the material points to the grid nodes at every time step and the computed solutions are transferred back to the material points, using the shape functions. This two-way mapping taking place in each time step is expected to cause some problems in the accuracy of the method.

On the other hand, the Particle Finite Element Method adopts the standard FEM procedures on this matter, so any inaccuracies in the assembly of the stiffness matrix and the stress recovery are avoided. However, the accuracy of the solution might be compromised during stress mapping between different meshes, where a perturbation of equilibrium is caused and may be manifested during the assembly of the internal loads vector. Considering the frequency of this mapping, which takes place either every time step or every a few time steps, depending on the chosen method (see 4.2), the error in the solution of the system may accumulate in a higher or lower degree during the analysis. Moreover, the recovered stress distribution in the domain is deteriorated by the stress discontinuities caused between elements, due to the use of simple, linear triangles, but can be improved by refining the mesh, especially in areas of high stress gradients.

5.2.2. NUMERICAL RESULTS

At this point, the matters discussed in the previous section are put to the test, as the two methods are compared on simple quasi-static and dynamic problems. Two benchmark problems, in which the internal forces play an important role in the outcome, are the 1D column compression and the axial bar vibration, that were already examined in section 4.1.2. These problems are simulated with the PFEM and the MPM. For the Particle Finite Element Method scheme, remeshing is performed at every step and the nodal smoothing (see 4.2.2) scheme is used for storing the stresses. As already discussed (4.2.2), this mapping operation is expected to introduce some smoothing on the recovered stresses in the system. On the other hand, both the standard and modified MPM 5.2 are used.

1D COLUMN COMPRESSION

In a 1D compression problem, a uniform stress field is expected, i.e. the axial stresses in the whole column must be equal to the applied load on top. The stress development at a particle on the bottom and one at the top of the column are shown in Fig. 5.2. For the PFEM, the recovered stresses are in perfect agreement with the applied pressure at the top. On the other hand, the results with the original MPM are problematic. More specifically, the stresses at the top of the column are constantly overestimated, while the opposite is the case for the material points close to the fixed end. As soon as the second from top row of material points crosses over the element below, the stiffness in the domain is distributed disproportionally, leading to the formation of overly stiff or soft elements, which results in erroneous calculations of stresses (left at Fig. 5.4). As a result, the load-displacement response for the standard MPM does not agree with the UL-FEM results (red dashed line in Fig. 5.3). This issue can be overcome for this simple type of problem by employing the modified MPM, through adjustment of the weighting coefficients of the material points according to Eq. 5.2. The results are drastically improved in this case, both in terms of stress evolution (Fig. 5.2) and load-displacement response (yellow line in Fig. 5.3). The difference in the computed stiffness and the recovered stresses for the two MPM schemes is shown in Fig. 5.4, while Fig. 5.5 depicts the final deposition of particles at the end of loading for the three methods.

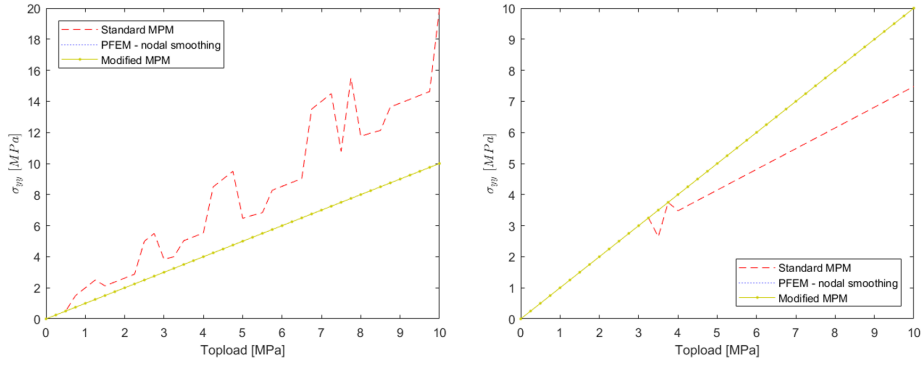

Figure 5.2: Axial stress development for the 1D column compression problem, with different numerical methods. Left: topside (free end). Right: bottom (fixed end).

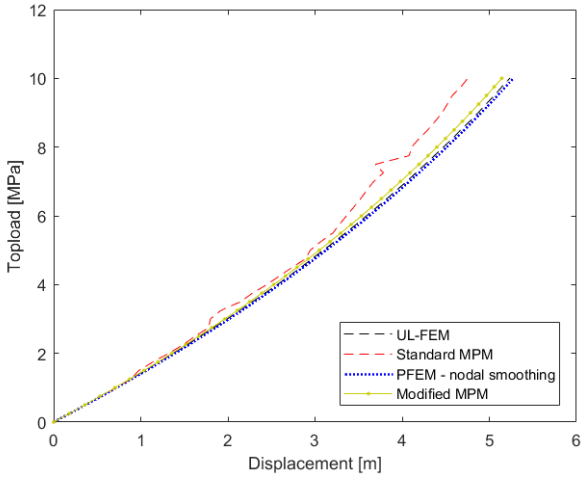

Figure 5.3: Load-displacement curves for the 1D column compression problem, with different numerical methods.

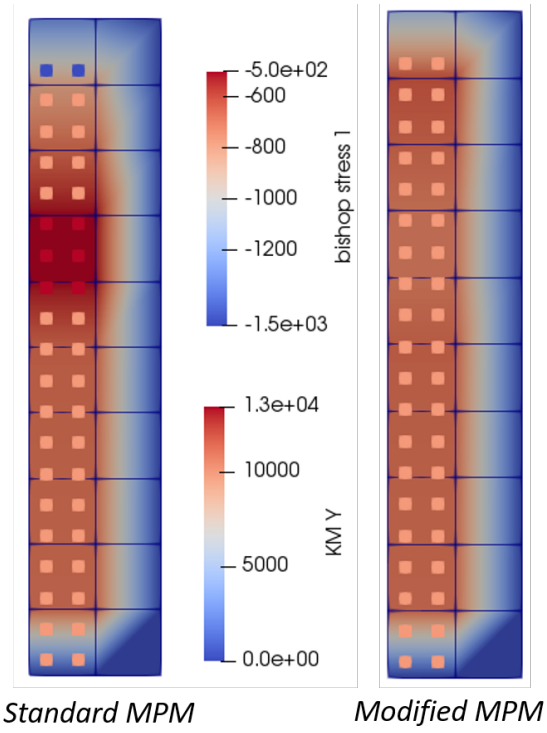

Figure 5.4: Impact of cell crossing on the stiffness and recovered stresses for MPM. Left: standard MPM. Right: modified MPM.

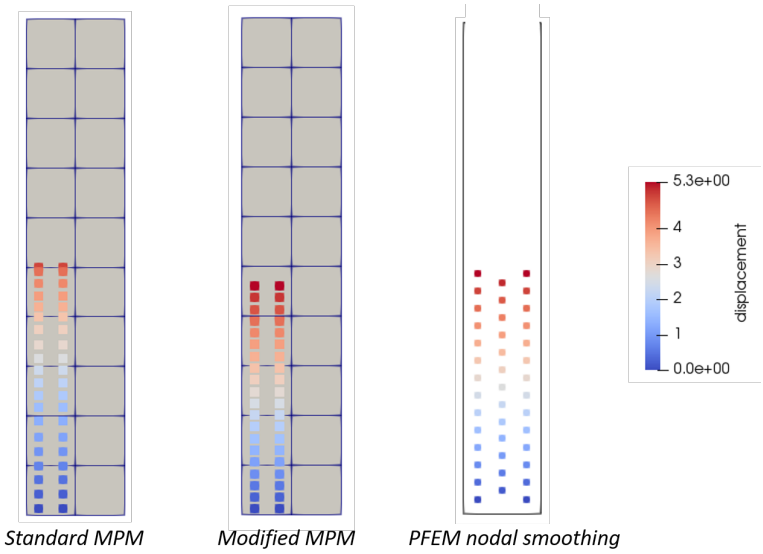

Figure 5.5: Final deposit of particles for column compression.

AXIAL VIBRATION OF CONTINUUM BAR

At this point, the dynamic problem of a bar vibrating freely along its axial direction is examined. The bar is given an initial velocity, i.e. kinetic energy, based on which is deformed; the kinetic energy, then, is transformed into strain energy, which forces the bar to return to its original state. In this kind of problem, the calculation of internal forces plays an important role in capturing the response of the system. Both the 1st and 2nd mode of vibration are considered.

Emphasis is put on the developed velocities and stresses both at the free end and close to the fixed end of the bar, and on the total energy conservation in the system. The last one is computed at each time as

$$E_{total}^{t+\Delta t} = E_{kin}^{t+\Delta t} + E_{strain}^{t+\Delta t} \tag{5.3}$$

Where the kinetic energy of the system is given by

$$E_{kin}^{t+\Delta t} = \frac{1}{2}\sum_{i=1}^{np} m_p^{t+\Delta t} v_p^{t+\Delta t} \tag{5.4}$$

Where p refers to the particle; the strain energy of the system at any point in the simulation is calculated at the Gauss points, according to

$$E_{strain}^{t+\Delta t} = E_{strain}^{t} + \sum_{i=1}^{total\ i.p.} V_i \frac{\sigma_i^{t+\Delta t} + \sigma_i^t}{2} : \delta\epsilon_i \tag{5.5}$$

After the shortcomings of standard MPM where demonstrated in the 1D compression problem, in the following, only the modified MPM will be considered and compared with the PFEM results.

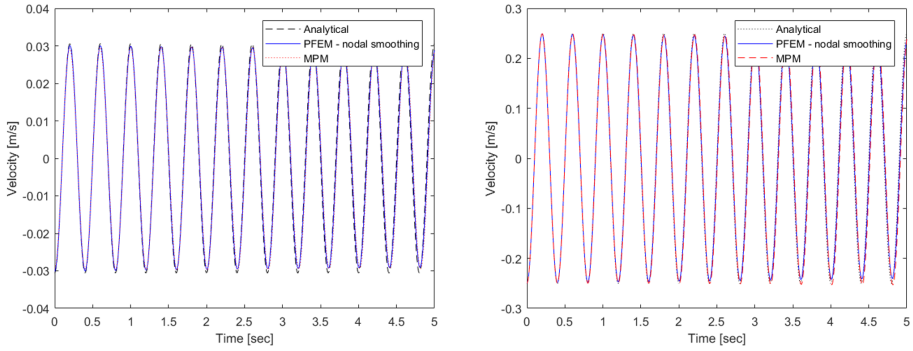

Figure 5.6: Comparison of velocity evolution between modified MPM and PFEM and the analytical solution, for the 1st mode of oscillation and $v_0 = 0.25 m/s$. Left: close to the fixed end. Right: free end.

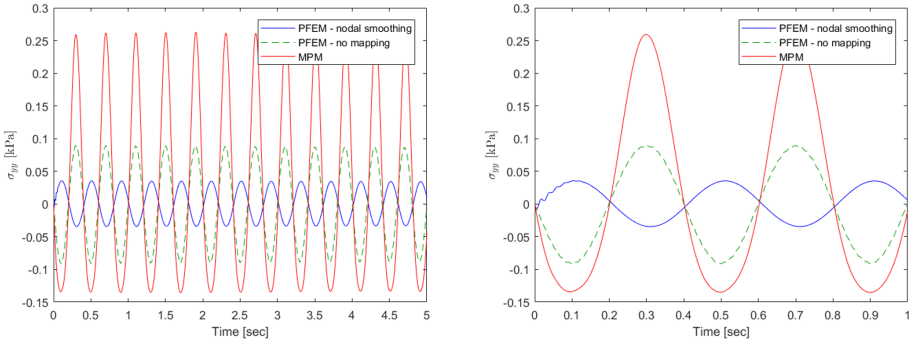

Figure 5.7: Axial stress evolution with modified MPM and PFEM, for the 1st mode of oscillation and $v_0 = 0.25 m/s$. Left: full time scale. Right: beginning of vibration.

1st Mode of Vibration Two scenarios were tested, one with an initial velocity of -0.25 m/s and one with -0.5 m/s. A time step of $\Delta_t = 0.001$ *sec* is chosen for output visualization purposes, even though both methods have proven to be effective for larger time steps (see [57], 4.1.2.4).

Focus is put on the velocity at particle A (free end) of the bar, which is compared with the available analytical solution, and at the stress evolution at an integration point B (fixed end) of the bar, with the two points shown in the sketch of Fig. 4.11.

First, the bar is given an initial velocity of **-0.25** m/sec, with the minus sign indicating a downward initial velocity. The initial velocity distribution in the bar for the first mode is given in Fig. 4.11. The results in terms of velocity, plotted for both points in Fig. 5.6, are in perfect agreement between the two methods and the analytical solution.

Regarding stress evolution, as a comparison, the PFEM without information mapping is also plotted (Fig. 5.8). The two methods show a very good agreement for point B at the bottom of the bar. However, in point A, there is serious deviation in the calculated stresses with PFEM, shown in Fig. 5.7 with the blue curve, not only in the magnitude, but also in the direction, i.e. tension and compression; a phase drift is evident. To elaborate, it appears that for PFEM with “nodal smoothing”, as already discussed, the extreme val-

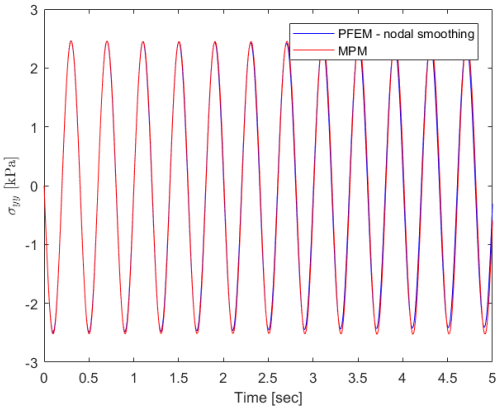

Figure 5.8: Axial stress evolution at the fixed end of the bar with modified MPM and PFEM, for the 1st mode of oscillation and $v_0 = 0.25 m/s$.

ues of internal variables are smoothed during the two-way mapping. This means that the equilibrium previously achieved at the top elements is disturbed and the internal forces are now off-balance. The internal forces acting on the top nodes are higher after mapping the historic stresses from the particles to the integration points, which results in the top row of particles moving slightly less than the particles immediately below; this results in the top element being extended, instead of compressed. This anomaly in the internal forces goes on until the end of the analysis, with the same period as in the vibration of the system. Despite this phenomenon not affecting the outcome of the analysis in terms of kinematic variables, as the top row of particles, i.e. A, shows perfect agreement with the analytical solution, it might influence the credibility of the method in applications where stress recovery is of utmost importance. On the other hand, MPM does not oscillate around 0, meaning that when the bar returns at its original shape, the stresses at the top row of elements do not go back to zero, indicating a residual tension.

Finally, the total energy oscillation curves are plotted in Fig. 5.9; the blue line, corresponding to PFEM, shows a noteworthy loss of energy, amounting up to 2.5% by the end of the analysis, while the MPM shows a very small decrease in energy, less than 0.3%.

When the initial velocity at the top is **-0.5** m/sec, the system response in MPM becomes unstable, as shown in Fig. 5.10. The difference between the two scenarios is that in the second one the initial velocity is enough to cause in MPM the second row of material points from the top to cross over to the element below. This occurrence seems to be causing an instability in the system, which is manifested in the spikes in the velocity at both the free and fixed end, with the former showing more intense deviations; however, the frequency of the vibration is captured. Moreover, in dynamic problems, information associated with the mass of the material points is also mapped to the grid nodes, besides the internal forces and the stiffness, inflicting further accumulation of error.

In terms of stresses at the free end of the bar (Fig. 5.11), the stresses predicted by the modified MPM become very unstable, with the frequency being hardly captured and the unrealistic extremes, especially in the first cycles. The PFEM with nodal smoothing

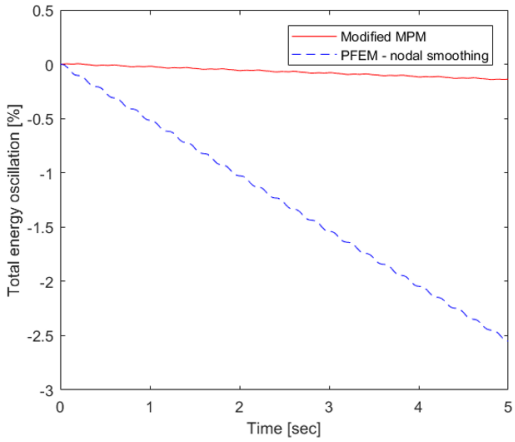

Figure 5.9: Total energy oscillations observed in the bar with the modified MPM and PFEM with nodal smoothing, for the 1st mode of oscillation and $v_0 = 0.25 m/s$.

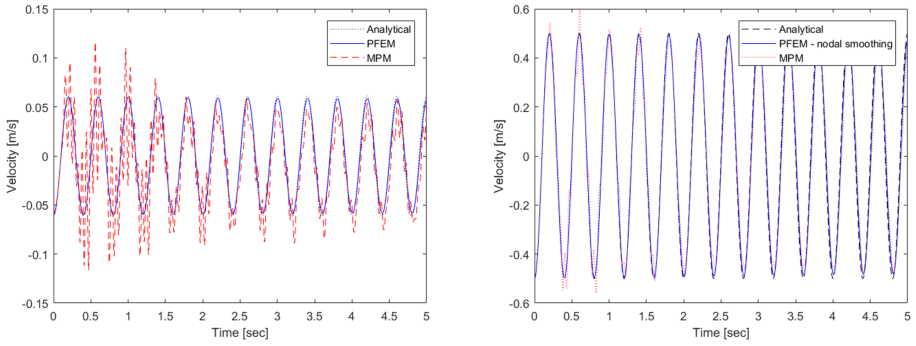

Figure 5.10: Comparison of velocity evolution between modified MPM and PFEM and the analytical solution, for the 1st mode of oscillation and $v_0 = 0.5 m/s$. Left: close to the fixed end. Right: free end.

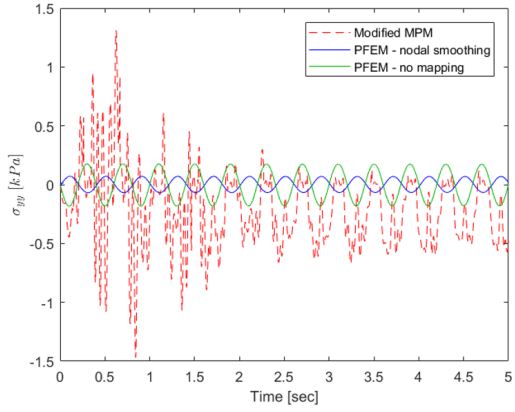

Figure 5.11: Axial stress evolution at the free end of the bar with modified MPM and PFEM, for the 1st mode of oscillation and $v_0 = 0.5 m/s$.

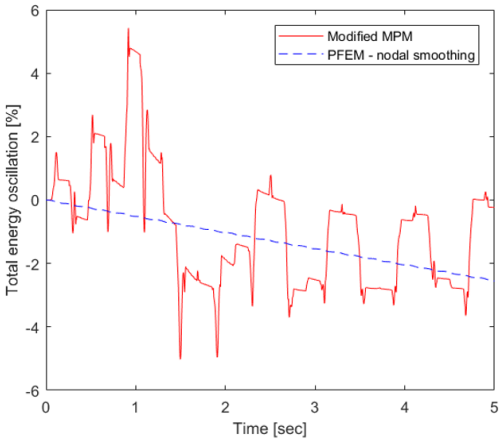

Figure 5.12: Total energy oscillations observed in the bar with the modified MPM and PFEM with nodal smoothing, for the 1st mode of oscillation and $v_0 = 0.5 m/s$.

exhibits a similar phase drift as was observed for v_0 = **-0.25** m/sec in Fig. 5.7.

The total energy loss percentage is plotted in Fig. 5.12. The PFEM response is almost identical with the results in Fig. 5.9, while the modified MPM curve (in red) demonstrates a quite unstable behavior, with unrealistic energy gains and losses in the system throughout the analysis. This comes to show that the PFEM is expected to demonstrate a rather stable response, regardless of the initial conditions of the problem, while the modified MPM is highly dependent on the initial state of the system, in this case, the initial velocity.

2nd Mode of Vibration For the second mode of vibration, the eigenvalue and eigenfrequency of the oscillation are computed by substituting $n = 2$ in Eq. 4.13. Again, two different scenarios were tested, one with v_0 = **-0.25** m/s and one with v_0 = **-0.75** m/s, with the latter being selected in such a way to cause cell-crossing in MPM. In Figs. 5.13 and 5.14, the energy loss in the system for the two scenarios is plotted. For v_0 = **-0.25** m/s, in Fig. 5.13, the dashed red curve shows that the energy loss in MPM is less intense compared to the PFEM solution with the same discretization (dashed blue curve). Refining the discretization (same colors, dotted lines) reduces the energy losses significantly, with MPM still showing a better behavior.

For higher initial velocity, i.e. v_0 = **-0.75** m/s, the general PFEM behavior does not change significantly (Fig. 5.14); the loss of energy observed for coarse discretization (blue dashed curve) is drastically improved by increasing the number of particles (blue dotted curve). On the other hand, the response of the system for the modified MPM is sub optimal. Although the system energy losses are reduced when the number of material points is increased in the domain, an increase in the total system is observed, which is evident by the spikes of the red dotted curve that at points exceed +5%. The total energy in the system for the coarser and finer MPM discretizations are shown in Figs. 5.15 and 5.16, respectively. Correspondingly, the energy for the coarse and fine PFEM cases are shown in Figs. 5.17 and 5.18.

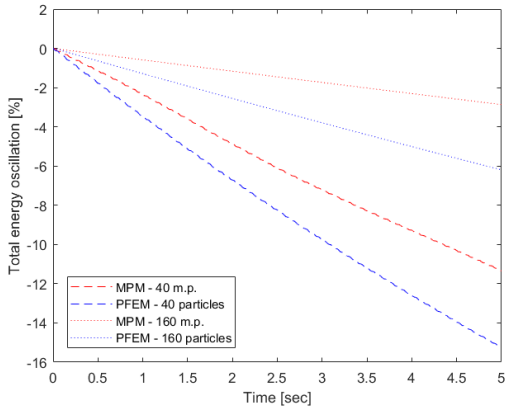

Figure 5.13: Total energy oscillation for the 2nd vibration mode, with $v_0 = 0.25 m/s$, for two different discretizations with PFEM and the modified MPM.

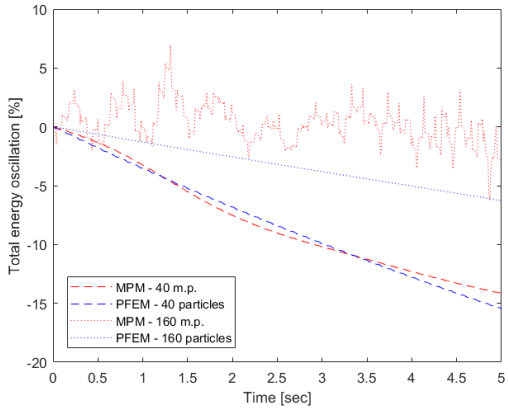

Figure 5.14: Total energy oscillation for the 2nd vibration mode, with $v_0 = 0.75 m/s$, for two different discretizations with PFEM and MPM.

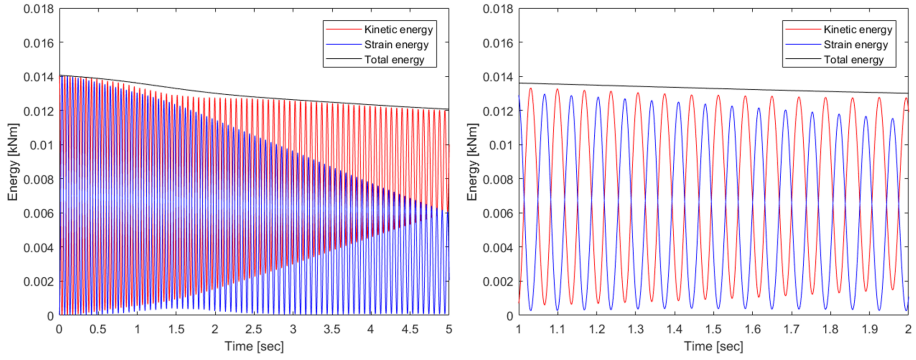

Figure 5.15: Energy in the system for the 2nd vibration mode, with $v_0 = 0.75 m/s$, for modified MPM for 40 material points.

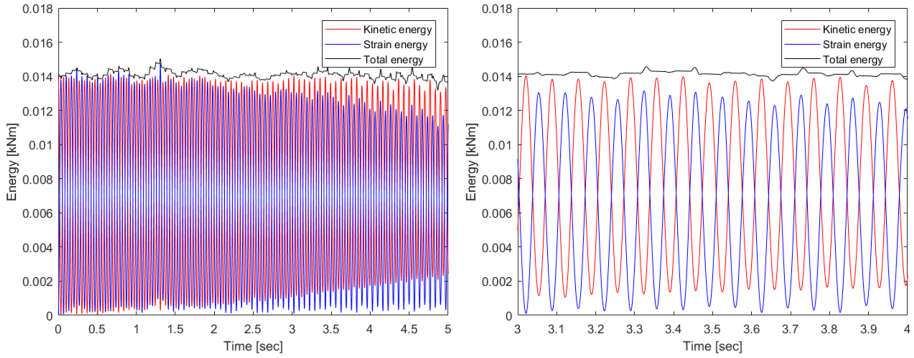

Figure 5.16: Energy in the system for the 2nd vibration mode, with $v_0 = 0.75 m/s$, for modified MPM for 160 material points.

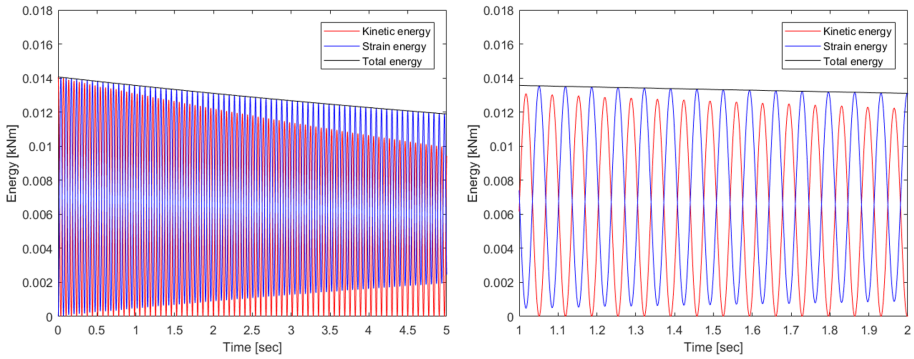

Figure 5.17: Energy in the system for the 2nd vibration mode, with $v_0 = 0.75 m/s$, for modified PFEM for 40 material points.

It is interesting to observe the cause of oscillation in the total system energy for the two methods, for which the focus will be put on the coarser configurations, i.e. in Figs. 5.15 and 5.17, and more specifically in the right, zoomed plots. In MPM, the kinetic energy in the system does not return to zero as it should when the vibration amplitudes are reached. This leads to residual kinetic energy in the system, which is not transformed to strain energy, that would force the bar to return to its exact original shape. The cause of this phenomenon can be located in the mapping of velocity and momentum between the material points and the grid nodes at every time step, which, unavoidable, causes trimming of extreme values. In contrast, for PFEM, the energy losses are triggered by the smoothing induced by the mapping of stresses; this is manifested by the residual strain energy always present in the system, which is not converted into kinetic energy to drive the bar to its initial state. The total energy loss is similar for both cases. In PFEM, however, this effect diminishes to a high degree when the discretization is refined (Fig. 5.18), which is not the case for the modified MPM (Fig. 5.16).

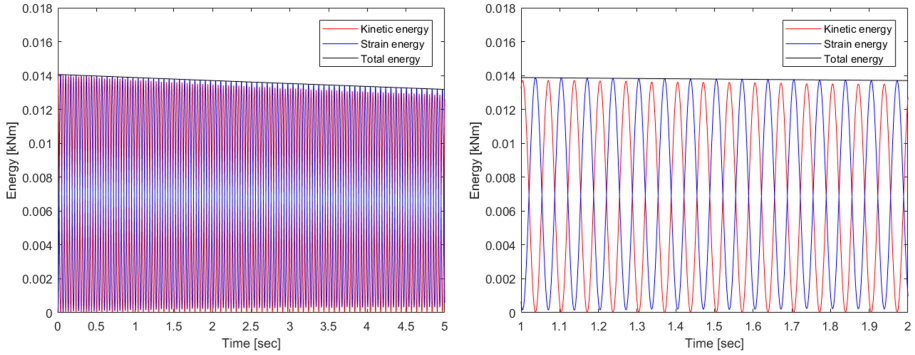

Figure 5.18: Energy in the system for the 2nd vibration mode, with $v_0 = 0.75 m/s$, for modified PFEM for 160 material points.

5.2.3. COMPUTATIONAL COST

After the evaluation of the performance of the two methods in terms of the accuracy on their results, the focus at this point is put on their need for computational resources. These are mainly associated with both computer memory space and the time needed for running a simulation. These factors can become more crucial when larger problems are simulated, both in terms of space (discretization) and time.

At this point, a brief investigation on the computational times of each method was conducted; analyses with both methods were run in the same computer system. More specifically, the bar vibration problem with 1000 time steps was simulated. For MPM, the modified version was used, which should not cause, however, any difference in the computational times, compared to the standard version. For PFEM, constant (at each time step) remeshing was performed, and the information mapping was performed with the "nodal smoothing" scheme. Given the differences between the two methods, a direct correspondence in the discretization of each one cannot be made. More specifically, if a one-to-one comparison between the two methods was required, this could be done either in terms of same number of information points (particles - material points), same number of degrees of freedom in the system (particles - computational grid nodes), same number of stress recovery points (elements - material points). To avoid such misconceptions, the results are presented and discussed for each method separately takes place.

The main procedures taking place during each time step for each method are discussed in the following. First, the tasks that are common in both methods are presented and, then, some commentary on the special processes of each one.

ASSEMBLY

This step involves assembling the stiffness and mass matrices and load vectors. In the MPM, the node connectivity is more or less constant throughout the simulation and can be predetermined in such a way that the bandwidth of the stiffness matrix remains relatively low. On the other hand, in PFEM, even if the particle distribution and numbering is done efficiently at preprocessing, the possibility of particles moving freely inside the solution domain can alter the nodal connectivity significantly along the course of the simulation. This can potentially trigger a severe increase in the matrix bandwidth, in-

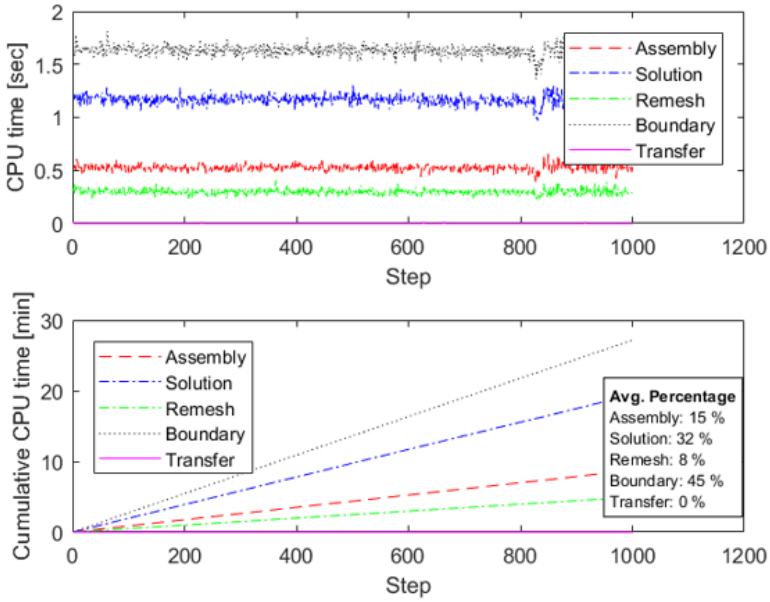

Figure 5.19: Computational time of every procedure in PFEM. Top: Average time for each cycle. Bottom: Cumulative cost throughout of the analysis.

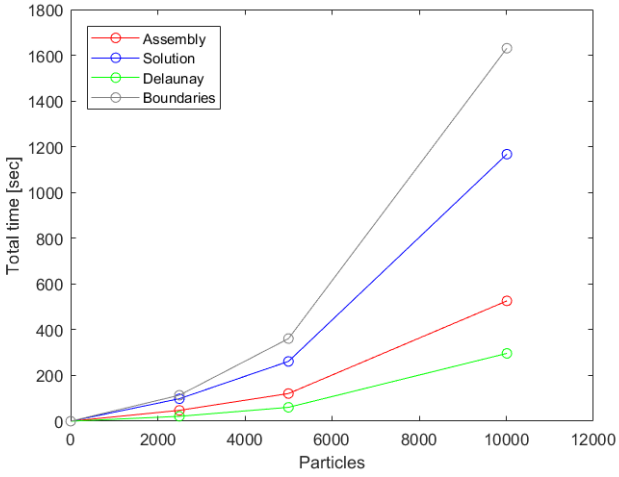

Figure 5.20: Cumulative computational time of every procedure in PFEM for 1000 cycles and different particle discretizations.

creasing the overall memory required for PFEM, for similar discretizations. However, this is the fastest process in MPM and the second fastest in PFEM.

SOLUTION

In both methods, an implicit solution scheme is employed (2.2.2, 5.1.2.1), which means that the stiffness matrix must be inverted in each time step for the solution of the system of equations. Using a modified Newton-Raphson iterative procedure instead of a regular Newton-Raphson for the convergence of the solution decreases significantly the required time at each step, because the stiffness matrix needs to be inverted only once. In both methods, the system of equations is solved for the displacements at the nodes and, then, the strains, stresses and other variables of interest are calculated, which is expected to require about the same running time.

However, if the active contact loop scheme (4.1.2.3) is used for treating the contact constraints in the PFEM, then the system of equations needs to be solved more than once. This means that the relative time of the solution step for the PFEM may be higher compared to the MPM. In any case, different contact implementations can result in faster simulation times.

REMESHING (PFEM)

One of the main arguments [53, 57] against the use of the Particle Finite Element Method for numerical simulation of complex physical problems is the great computational demand, especially time-wise, which is mainly attributed to the frequent remeshing required for maintaining a quality mesh. The remeshing procedure involves the mesh generation, boundary identification and interface mesh creation steps. In Fig. 5.19, the computational times for each procedure for a simulation of 10,000 particles with 1,000 steps are plotted, while, in Fig. 5.20, the total computational times of each process for different discretizations are shown. The Delaunay triangulation is the fastest process in each step, costing almost half of the assembly and about 25% of the solution step. The boundary identification process seems to be taking the longest in each simulation; this is due to the linear search (brute force) approach used for finding the neighbours of each particle in order to perform the check for boundary detection (3.2.2). This process can be made faster by employing a different *nearest neighbour search* algorithm, such as space partitioning. Nevertheless, this step may become redundant if a constrained DT (3.1.3.3) is used for construction of the domain at each remeshing step.

If additional features of the PFEM, such as the dynamic particle discretization (3.3) for improving the quality of the solution, are incorporated in the scheme, then the total remeshing times may be increased significantly.

MAPPING (PFEM)

This process takes place only in PFEM for solid mechanics applications where the history of the internal variables is essential information for the analysis. As discussed (4.2), there are various ways to perform this task and the choice can affect the computational needs. If the “nodal smoothing” technique is used, then the transfer time can be limited, but there will be need for extra arrays to store the information on the particles. On the other hand, if the “mesh projection” scheme is employed, there will be no need for additional computer memory, but the search for the closest centroid to each point will increase the

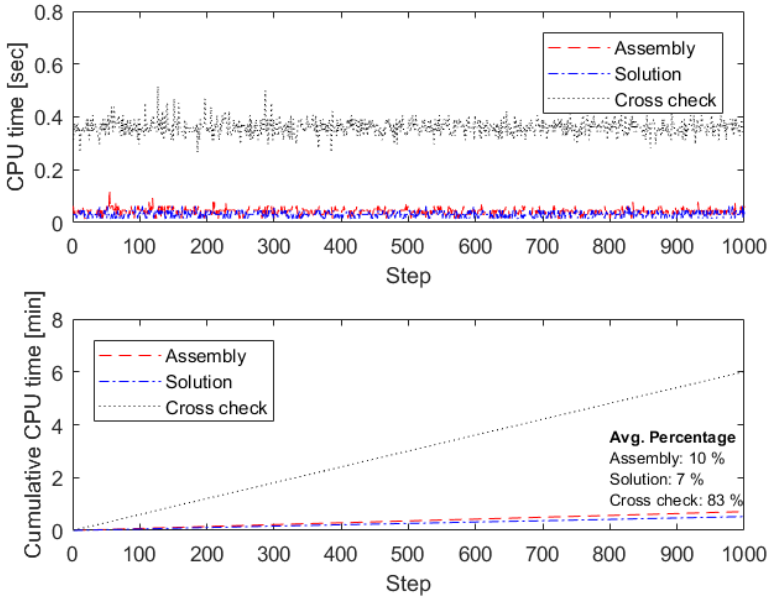

Figure 5.21: Computational time of every procedure in MPM. Top: Average time for each cycle. Bottom: Cumulative cost throughout of the analysis.

computational times heavily; nonetheless, as mentioned, more optimized algorithms for the “nearest neighbour search” can significantly reduce this cost.

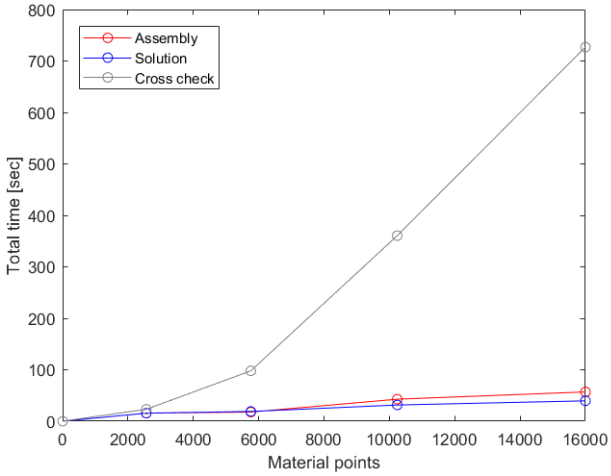

Figure 5.22: Cumulative computational time of every procedure in PFEM for 1000 cycles and different number of material points

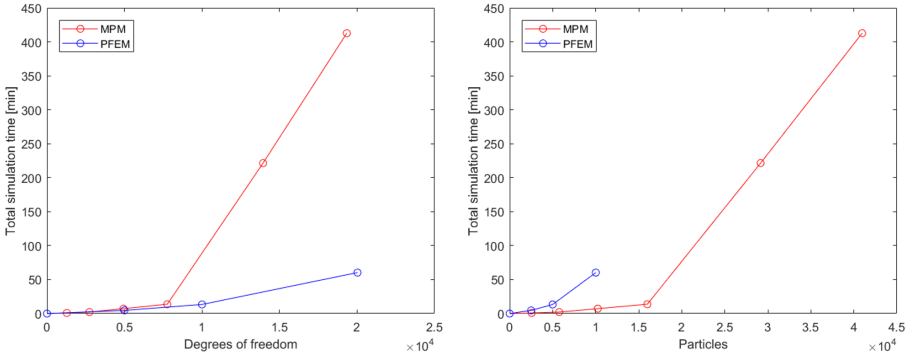

Figure 5.23: Total computation times with the MPM and the PFEM for different number of degrees of freedom (left) and information points (right).

ELEMENT CROSS CHECK (MPM)

Besides assembly and solution, the extra procedure employed in MPM compared to conventional FEM is to check if any material points have moved to new elements, update the local coordinates of the material points, activate the new elements and initialize -if needed- the new equation arrays. In Fig. 5.21, this procedure seems to be by far the most time-consuming, amounting for more than 80 % of the computational time for an analysis involving 10,000 material points. Also, in Fig. 5.22, it appears that the cost of this step increases exponentially as the number of material points in the system rises. As was commented for the PFEM procedures involving heavy duty spatial searches, the check for finding if any material points have crossed over from one element to another can also be sped up, by employing a more efficient algorithm.

In Fig. 5.23, a direct comparison between the computational times of the two methods is performed. As was already discussed, the comparison can be done on the basis of the same number of either degrees of freedom, information points (particles - material points) or stress recovery points (elements - material points). The comparison in terms of degrees of freedom is done in the left plot and the comparison based on information points in the right plot of the same figure. It is apparent that the Particle Finite Element Method is much faster for the same system dimensions, i.e. degrees of freedom. However, for the same system size, the Material Point Method offers many more information points, since - for the adopted implementation - there are, initially, four material points inside each element for the MPM, leading to a more refined solution. When increasing the refinement in PFEM, the computer storage requirements become significantly high, reaching a level that the used computer system cannot handle. This explains why, in the right graph, the most refined discretization in PFEM includes 10,000 particles, while for MPM the same number is greater than 40,000. Overall, the computational requirements for the Material Point Method are much lower.

6

SUMMARY AND DISCUSSION

6.1. CONCLUSIONS

The conclusions of this thesis are presented with respect to the research questions posed in chapter 1.

How is the domain discretization, including particle connectivity and domain boundaries, updated to handle large deformation problems in PFEM and how is the outcome of the analysis affected?

In PFEM, the solution domain is frequently disassembled and regenerated on basis of the new positions of the particles, in order to maintain a quality discretization. This procedure is done in two steps. First, a Delaunay triangulation is employed to provide an initial particle connectivity, which enjoys some optimal properties, such as the minimization of the smallest angle in all elements. However, the DT does not respect the internal and external boundaries of the domain, which means that an additional scheme is needed for identification of the boundaries. In most implementations of PFEM, this was done via the α-shape method, which deletes incorrect elements and offers a criterion for detecting particles lying on the boundary. However, the subjectivity introduced through the choice of the α_{lim} parameter seems to have an impact on the final form of the domain and, subsequently, on the outcome of the analysis; special care is needed when selecting an appropriate value. Finally, in some published variations of the PFEM, the quality of the discretization and the computational efficiency of the method are enhanced by utilizing a dynamic particle discretization, by adding, removing or repositioning particles.

How is contact treated in the Particle Finite Element Method and how can such an algorithm be implemented?

In most implementations of PFEM, contact detection and, subsequently, treatment is performed with the use of an interface mesh. The interface mesh is constructed at every step in a straightforward manner; given the particles lying on the boundary of each body, including rigid boundaries, the same meshing procedure as for the domain is used. First, as previously described, a Delaunay triangulation is performed, followed by the α-

shape method. Any elements that "survive" from the latter serve as indicators of possible contact. On the basis of this interface mesh, the contact constraints can be enforced by means of various conventional schemes, including Lagrange multipliers, the penalty method, Nitsche methods and others. In this project, the contact interface generation algorithm was implemented and a simple contact algorithm was formulated and validated against benchmark quasi-static and dynamic problems. The automatic contact detection and contact element generation allows for the use of more advanced contact techniques and adaptations for handling multi-body interaction.

How is internal information handled in the Particle Finite Element Method and what is the impact on the results?

In contrast with fluid mechanics, the simulation of solid mechanics problems with PFEM requires the storage and transfer of internal variables between meshes. There are several techniques to do this, each one with its one limitations and advantages; the selection is up to the user. The most popular so far are the *nodal smoothing* and the *mesh projection* techniques. The first one introduces some smoothing, which causes errors in the computed values of, e.g., stresses close to the boundaries of the problem by trimming of extreme values; however, the overall response of the system is not significantly impacted, as was seen in the dynamic applications. Moreover, with this technique, the computational cost of the analysis is increased in terms of required storage, as extra arrays are needed to store the projected elemental variables at the particles. On the other hand, the mesh projection operation is expected to cause some loss of information in case elements are deleted, but equilibrium is preserved in areas where elements are not altered; in this case, no extra storage is needed, but the increase of the computational time in the analysis from this method comes from the spatial search required for finding the nearest neighbours of each element.

How is mass conservation affected by frequent remeshing employed in PFEM?

Due to the remeshing and boundary identification taking place several times throughout an analysis with the Particle Finite Element Method, there may be variations in the domain volume which may lead to mass oscillations. These can be triggered by fluctuations in the domain boundaries, separation and re-entering of particles, closing and opening of internal cavities, and generation and deletion of contact elements in fluid simulations. All of these occurrences are directly associated with the choice of the α_{lim} value for the α-shape method. In solid mechanics, however, only the first phenomenon seems to be of importance and its effect can be mitigated by refining the particle distribution and calibrating the value of α_{lim} to the specific problem under investigation, even adjusting it during the course of the simulation. Despite this workaround, arguments are still being raised against the subjectivity of the method and the use of a constrained Delaunay triangulation, where the external boundaries of the domain are predetermined during remeshing, is being proposed as a countermeasure; this solution, however, is not always applicable and may lead to significantly distorted elements in the boundaries.

How does the Particle Finite Element Method compare with other recently developed numerical tools, like the Material Point Method, on benchmark solid mechanics problems?

Both the PFEM and the MPM use particles to store information and make use of a

Finite Element mesh on which the governing equations are solved. However, their main differences lie on the way the necessary mapping of this information takes place and the way the system of equations is assembled. In PFEM, information must be mapped between the old and new mesh and the assembly is done using standard FEM procedures; in MPM information is constantly mapped between the material points and the nodes of the computational grids and assembly is performed using the suboptimal positions of the material points. In both cases, some loss of accuracy is expected.

The standard Material Point Method fails to produce accurate results even for simple quasi-static problems like the 1D compression, due to cell-crossing taking place, indicating the need for modifications, either in the weighting coefficients of the material points or the shape functions. However, cell crossing effects are still unavoidable and become evident in dynamic applications, where erroneous estimations of the stresses and significant losses in system energy are observed. On the other hand, the smoothing of stresses caused by information mapping with the *nodal smoothing* tool in PFEM is not impactful for quasi-static applications, but can cause some anomalies in the stress field and energy loss in dynamic problems. These effects can, in any case, be mitigated by increasing the number of particles in the domain. In any case, the inaccuracies in the results for PFEM can be improved by refining the discretization, which, however, can not be said for MPM, where finer meshes result in more intense cell-crossing taking place.

In contrast to what is reported in literature regarding the computational cost of PFEM [53, 57], the method is not significantly limited by its computational speed. The remeshing step can be quite fast and optimization of certain procedures and introduction of remeshing criteria can further reduce simulation times; however, adding more features, such as particle addition, may increase it. On the other hand, the most time costly procedure in each MPM cycle is the search for any potential element crossing of material points; this, too, can be sped up by employing a more efficient searching algorithm. Nonetheless, MPM can offer better accuracy for much smaller computational cost, both time- and storage-wise, as for the same number of degrees of freedom more integration points are present in the domain.

6.2. RECOMMENDATIONS FOR FURTHER RESEARCH

- **Assess the developed PFEM code in more advanced problems, including large deformations.** For the needs of this thesis, a PFEM implementation for solid mechanics was developed and was tested on simple applications, where the main objective was to preliminary assess the efficiency of the method. Simulation of more complex problems that will involve large deformations and comparison with published results would be interesting for further evaluation of this new numerical tool.

- **Incorporate constitutive model for simulation of more complex problems, i.e. slope failure.** In this work, simple applications with linear elastic material models were examined. Although the PFEM did perform well in these cases, incorporation of more advanced constitutive models for capturing complex (soil) material behavior would be an interesting approach, given that the method has not been extensively used for soil mechanics in literature. In any case, since it is based on

the FEM framework, the method should be capable of handling any kind of constitutive models.

- **Compare the efficiency of "nodal smoothing" and "mesh projection" transfer operators, where applicable.** Given the nature of the examined applications, the efficiency of only the "nodal smoothing" technique could be investigated for the transference of elemental variables between successive meshes. The "mesh projection" operation, which has been lately introduced and is gaining popularity, is mostly applicable when high mesh distortion takes place, which is not the case for the applications in this work. For future considerations, if problems involving large deformation and extreme mesh tangling are examined, then a comparison between these two schemes should yield interesting results and lead towards better understanding and improved efficiency of the PFEM.

- **Investigate the mass conservation of PFEM in large deformation solid mechanics problems with respect to the remeshing procedure.** The conservation of mass during the analysis with PFEM was only examined in this work in the basis of a literature research; in any case, this is a topic not thoroughly addressed in bibliography, especially for solid mechanics applications. As a recommendation, future research on the efficiency of the method could also focus on how the conservation of mass is affected by the various procedures, especially the remeshing steps, and investigate potential remedies.

- **Implement a more advanced contact method, making use of the interface mesh generation tool.** For the needs of this thesis, an interface mesh generation algorithm was developed, on the basis of which a simple contact treatment scheme was formulated and implemented. Nevertheless, the proposed scheme is a rather simple one, focusing only on normal contact and validated only against some simple problems. By making use of the contact mesh generation scheme developed, more advanced contact treatment approaches can be incorporated in the method, that would facilitate modelling of frictional contact and multi-body interaction.

- **Investigate the potential and efficiency of dynamic particle discretization.** A feature of the Particle Finite Element Method that was merely addressed in this work based on a desk study is the possibility to dynamically alter the number and position of particles in the domain. Although this option is not examined extensively in literature, it could prove to be essential in improving the accuracy and minimizing the computational costs of the analysis. Considering the added complexity in the implementation of the method, this could still be a feature worth exploring in future work.

REFERENCES

[1] R. Aubry, S. R. Idelsohn, and E. Oñate. Particle finite element method in fluid-mechanics including thermal convection-diffusion. *Computers and Structures*, 83(17-18):1459–1475, 2005.

[2] K.-J. Bathe. *Finite element procedures*. K.-J. Bathe, Watertown, MA, second edition, 2006.

[3] P. A. Becker and S. R. Idelsohn. A multiresolution strategy for solving landslides using the particle finite element method. *Acta Geotechnica*, 11(3):643–657, 2016.

[4] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 1994.

[5] H. Bui, R. Fukagawa, K. Sako, and S. Ohno. Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic-plastic soil constitutive model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(12):1537–1570, 2008.

[6] H. Bui, R. Fukagawa, K. Sako, and J. Wells. Slope stability analysis and discontinuous slope failure simulation by elasto-plastic smoothed particle hydrodynamics (SPH). *Géotechnique*, 61(7):565–574, 2011.

[7] J. Burkardt. TABLE_DELAUNAY: FORTRAN code for 2D point triangulation, 2009.

[8] O. Buzzi, D. M. Pedroso, and A. Giacomlnl. Caveats on the implementation of the generalized material point method. *CMES - Computer Modeling in Engineering and Sciences*, 31(2):85–106, 2008.

[9] N. Calvo, S. R. Idelsohn, and E. Oñate. The extended Delaunay tessellation. *Engineering Computations*, 20(5/6):583–600, 2003.

[10] J. C. Cante, J. Oliver, and C. G. Ferrari. On the numerical modelling of forming processes using the particle finite element method. In *Computational Plasticity: Fundamentals and Applications - Proceedings of the 8th International Conference on Computational Plasticity, COMPLAS VIII*, pages 2–5, 2005.

[11] J. M. Carbonell, E. Oñate, and B. Suárez. *Modeling of ground excavation with the particle finite element method*. International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, first edition, 2010.

[12] J. M. Carbonell, E. Oñate, and B. Suárez. Modeling of ground excavation with the particle finite element method. *ASCE: Journal of Engineering Mechanics*, 136(April):455–463, 2010.

[13] J. M. Carbonell, E. Oñate, and B. Suárez. Modelling of tunnelling processes and rock cutting tool wear with the particle finite element method. *Computational Mechanics*, 52(3):607–629, 2013.

[14] M. L. Cerquaglia, G. Deliége, R. Boman, V. Terrapon, and J. P. Ponthot. Free-slip boundary conditions for simulating free-surface incompressible flows through the particle finite element method. *International Journal for Numerical Methods in Engineering*, 110(10):921–946, 2017.

[15] C. J. Coetzee, P. A. Vermeer, and A. H. Basson. The modelling of anchors using the material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(9):879–895, 2005.

[16] M. Cremonesi, F. Ferri, and U. Perego. A basal slip model for Lagrangian finite element simulations of 3D landslides. *International Journal for Numerical and Analytical Methods in Geomechanics*, 41(1):30–53, 2017.

[17] M. Cremonesi, A. Frangi, and U. Perego. A Lagrangian finite element approach for the analysis of fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 84(5):610–630, 2010.

[18] C. Dávalos, J. C. Cante, J. A. Hernández, and J. Oliver. *On the Modelling of Granular Flows in Industrial Applications via the Particle Finite Element Method*. International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, first edition, 2014.

[19] C. Dávalos, J. C. Cante, J. A. Hernández, and J. Oliver. On the numerical modeling of granular material flows via the Particle Finite Element Method (PFEM). *International Journal of Solids and Structures*, 71:99–125, 2015.

[20] R. Dey, B. Hawlader, R. Phillips, and K. Soga. Large deformation finite-element modelling of progressive failure leading to spread in sensitive clay slopes. *Géotechnique*, 65(8):657–668, 2015.

[21] J. Donea, A. Huerta, J. P. Ponthot, and A. Rodr. Arbitrary Lagrangian – Eulerian Methods. *Encyclopedia of Computational Mechanics*, pages 1–25, 1999.

[22] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[23] A. Franci and M. Cremonesi. On the effect of standard PFEM remeshing on volume conservation in free-surface fluid flow problems. *Computational Particle Mechanics*, 4(3):331–343, 2017.

[24] A. Franci, E. Oñate, and J. M. Carbonell. Unified Lagrangian formulation for solid and fluid mechanics and FSI problems. *Computer Methods in Applied Mechanics and Engineering*, 298:520–547, 2016.

[25] R. Gingold and J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 1977.

[26] L. Gonzalez Acosta, P. J. Vardon, and M. A. Hicks. An evaluation of MPM, GIMP and CMPM in geotechnical problems considering large deformations. In *Proceedings of the 15th International Conference of the International Association for Computer Methods and Advances in Geomechanics: 19-23 October 2017, Wuhan, China*, 2017.

[27] L. Gonzalez Acosta, P. J. Vardon, and M. A. Hicks. Composite Material Point Method (CMPM) to Improve Stress Recovery for Quasi-static Problems. In *Procedia Engineering*, volume 175, pages 324–331, 2017.

[28] Y. Hu and M. F. Randolph. A practical numerical approach for large deformation problems in soil. *International Journal for Numerical and Analytical Methods in Geomechanics*, 22(5):327–350, 1998.

[29] S. R. Idelsohn, E. Oñate, and F. Del Pin. The particle finite element method: A powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal for Numerical Methods in Engineering*, 61(7):964–989, 2004.

[30] S. R. Idelsohn, E. Oñate, F. Del Pin, and N. Calvo. Fluid-structure interaction using the particle finite element method. *Computer Methods in Applied Mechanics and Engineering*, 195(17-18):2100–2123, 2006.

[31] M. Kaltenbacher. Non-conforming Finite Elements for Flexible Discretization with Applications to Aeroacoustics. In *Computational Acoustics*, volume 579, pages 35–67. Springer, 2018.

[32] P. M. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3):217–241, 2003.

[33] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013, 1977.

[34] L. Meng, W. H. Zhang, J. H. Zhu, and L. Xia. A biarc-based shape optimization approach to reduce stress concentration effects. *Acta Mechanica Sinica/Lixue Xuebao*, 30(3):370–382, 2014.

[35] L. Monforte, M. Arroyo, J. M. Carbonell, and A. Gens. Numerical simulation of undrained insertion problems in geotechnical engineering with the Particle Finite Element Method (PFEM). *Computers and Geotechnics*, 82:144–156, 2017.

[36] L. Monforte, M. Arroyo, J. M. Carbonell, and A. Gens. Coupled effective stress analysis of insertion problems in geotechnics with the Particle Finite Element Method. *Computers and Geotechnics*, 101(February):114–129, 2018.

[37] L. Monforte, J. M. Carbonell, M. Arroyo, and A. Gens. Numerical Simulation of Penetration Problems in Geotechnical Engineering With the Particle Finite Element Method. In *IV International Conference on Particle-based Methods – Fundamentals and Applications*, pages 1–8, 2015.

[38] L. Monforte, J. M. Carbonell, M. Arroyo, and A. Gens. Performance of mixed formulations for the particle finite element method in soil mechanics problems. *Computational Particle Mechanics*, 4(3):269–284, 2017.

[39] M. Nazem, J. P. Carter, and D. W. Airey. Arbitrary Lagrangian-Eulerian method for non-linear problems of geomechanics. *IOP Conference Series: Materials Science and Engineering*, 10:012074, 2010.

[40] M. Nazem, D. C. Sheng, J. P. Carter, and S. W. Sloan. Arbitrary Lagrangian-Eulerian method for large-strain consolidation problems. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32:1023–1050, 2008.

[41] V. P. Nguyen. *Material point method: basics and applications*. PhD thesis, Cardiff University, 2014.

[42] H. Nonoyama, S. Moriguchi, K. Sawada, and A. Yashima. Slope stability analysis using smoothed particle hydrodynamics (SPH) method. *Soils and Foundations*, 55(2):458–470, 2015.

[43] J. Oliver, J. C. Cante, and C. Gonzalez. On particle finite element methods in solid mechanics problems. In *Computational Plasticity: Fundamentals and Applications - Proceedings of the 8th International Conference on Computational Plasticity, COMPLAS VIII*, number PART 1, pages 1–4, 2005.

[44] J. Oliver, J. C. Cante, R. Weyler, C. González, and J. Hernandez. Particle finite element methods in solid mechanics problems. *Computational Methods in Applied Sciences*, 7:87–103, 2007.

[45] J. Oliver, J. C. Cante, R. Weyler, and J. Hernandez. Possibilities of particle finite element methods in industrial forming processes. In *AIP Conference Proceedings*, volume 907, pages 1484–1489, 2007.

[46] E. Oñate, S. R. Idelsohn, M. A. Celigueta, and R. Rossi. Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1777–1800, 2008.

[47] E. Oñate, S. R. Idelsohn, F. Del Pin, and R. Aubry. The Particle Finite Element Method — an Overview. *International Journal of Computational Methods*, 01(02):267–307, 2004.

[48] V. N. Parthasarathy, C. M. Graichen, and A. F. Hathaway. A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design*, 15(3):255–261, 1994.

[49] J. M. Rodriguez, J. M. Carbonell, J. C. Cante, and J. Oliver. The particle finite element method (PFEM) in thermo-mechanical problems. *International Journal for Numerical Methods in Engineering*, 107(9):733–785, 2016.

[50] J. M. Rodriguez Prieto, J. M. Carbonell, J. C. Cante, J. Oliver, and P. Jonsén. Generation of segmental chips in metal cutting modeled with the PFEM. *Computational Mechanics*, pages 1–17, 2017.

[51] M. Sabel, C. Sator, and R. Müller. A particle finite element method for machining simulations. *Computational Mechanics*, 54(1):123–131, 2014.

[52] I. Smith, D. Griffiths, and L. Margetts. *Programming the Finite Element Method*. John Wiley & Sons Ltd, Chichester, West Sussex, United Kingdom, fifth edition, 2013.

[53] K. Soga, E. Alonso, A. Yerro, K. Kumar, and S. Bandara. Trends in large-deformation analysis of landslide mass movements with particular emphasis on the material point method. *Géotechnique*, 66(3):248–273, 2016.

[54] D. Sulsky, Z. Chen, and H. L. Schreyer. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118(1-2):179–196, 1994.

[55] D. Sulsky, S.-J. Zhou, and H. L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1-2):236–252, 1995.

[56] P. J. Vardon, B. Wang, and M. A. Hicks. Slope failure simulations with MPM. In *1st International Conference on the Material Point Method, MPM 2017*, volume 29, pages 445–451, 2017.

[57] B. Wang. *Slope failure analysis using the Material Point Method*. PhD thesis, Wuhan University, China & Technische Universiteit Delft, the Netherlands, 2017.

[58] B. Wang, M. A. Hicks, and P. J. Vardon. Slope failure analysis using the random material point method. *Géotechnique Letters*, 6(2):113–118, 2016.

[59] B. Wang, P. J. Vardon, and M. A. Hicks. Preliminary analysis of rainfall-induced slope failures using the material point method. In *Landslides and Engineered Slopes. Experience, Theory and Practice*, volume 3, pages 2037–2042, 2016.

[60] B. Wang, P. J. Vardon, M. A. Hicks, and Z. Chen. Development of an implicit material point method for geotechnical applications. *Computers and Geotechnics*, 71:159–167, 2016.

[61] D. Wang, B. Bienen, M. Nazem, Y. Tian, J. Zheng, T. Pucker, and M. F. Randolph. Large deformation finite element analyses in geotechnical engineering. *Computers and Geotechnics*, 65:104–114, 2015.

[62] P. Wriggers. *Computational contact mechanics*. Springer-Verlag Berlin Heidelberg, second edition, 2006.

[63] L. Yu, Y. Hu, J. Liu, M. F. Randolph, and X. Kong. Numerical study of spudcan penetration in loose sand overlying clay. *Computers and Geotechnics*, 46:1–12, 2012.

[64] W. Zeng and G. R. Liu. Smoothed Finite Element Methods (S-FEM): An Overview and Recent Developments. *Archives of Computational Methods in Engineering*, pages 1–39, 2016.

[65] W. Zhang, W. Yuan, and B. Dai. Smoothed particle finite-element method for large-deformation problems in geomechanics. *International Journal of Geomechanics*, 18(4):1–12, 2018.

[66] X. Zhang. *Particle Finite Element Method in Geomechanics*. PhD thesis, The University of Newcastle, Australia, 2016.

[67] X. Zhang, K. Krabbenhoft, D. M. Pedroso, A. V. Lyamin, D. Sheng, M. V. da Silva, and D. Wang. Particle finite element analysis of large deformation and granular flow problems. *Computers and Geotechnics*, 54:133–142, 2013.

[68] X. Zhang, K. Krabbenhoft, D. Sheng, and W. Li. Numerical simulation of a flow-like landslide using the particle finite element method. *Computational Mechanics*, 55(1):167–177, 2015.

[69] X. Zhang, K. Krabbenhoft, and D. C. Sheng. Particle Finite Element Simulation of Granular Media. *Applied Mechanics and Materials*, 553(November 2014):410–415, 2014.

[70] X. Zhang, D. Sheng, S. W. Sloan, and J. Bleyer. Lagrangian modelling of large deformation induced by progressive failure of sensitive clays with elastoviscoplasticity. *International Journal for Numerical Methods in Engineering*, 112(8):963–989, 2017.

[71] M. Zhu and M. H. Scott. Modeling fluid-structure interaction by the particle finite element method in OpenSees. *Computers and Structures*, 132:12–21, 2014.