Formulation of a Dynamic Material Point Method (MPM) for Geomechanical Problems

Von der Fakultät für Bau– und Umweltingenieurwissenschaften
der Universität Stuttgart
zur Erlangung der Würde eines Doktors der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Abhandlung,

vorgelegt von

ISSAM K. J. AL-KAFAJI

aus Bagdad, Irak

Hauptberichter:	Prof. Dr.-Ing. Pieter A. Vermeer
Mitberichter:	Prof. Dieter F. Stolle, P. Eng.
Mitberichter:	Prof. Dr.-Ing. habil. Christian Moormann

Tag der mündlichen Prüfung: 13. März 2013

Institut für Geotechnik der Universität Stuttgart

2013

Keywords: Dynamics, large deformation, material point method, two-phase media, pile driving

Printed by Ridderprint BV, The Netherlands, 2013

ISBN 978-90-5335-705-7
(D93 - Dissertation, Universität Stuttgart)

Preface

The application of numerical analyses in geotechnical engineering is substantial, but most usually quasi-static, small-deformation problems are analysed. Following pioneers in structural engineering, the Lagrangian finite element method (FEM) has been introduced into geotechnical engineering, rather than the Eulerian FEM, or resembling methods, as typically used in fluid mechanics for the analyses of flow problems with corresponding large-displacements. In recent years coupled Eulerian Lagrangian FEM has been applied successfully for solving dynamical large-deformation problems, e.g. for simulating pile driving, but it is difficult to extend this method to soil-fluid interaction problems as typical in branches of geotechnical engineering. For this reason, Issam K.J. Al-Kafaji has chosen to focus his research on the so-called Material Point Method (MPM), which may be conceived as an advancement of FEM.

At the beginning of this study Issam has set himself very ambitious goals, i.e. the extension of MPM for application to large-deformation problems in geomechanics, including pile driving in sand using an advanced highly non-linear constitutive model. Secondly, the modeling of two-phase soil with full consideration of dynamic generation and dissipation of pore pressures has been carried out. These goals have been reached in a convincing way. In addition the candidate has performed an extensive survey of relevant literature and is nicely putting his study in a wider context of leading edge, worldwide research in geomechanics. As a consequence, his research has already attracted good attention in the international research community. To my judgment this well-written study is of finest quality and finally Issam passed his PhD examination with the highest possible grade, i.e. 'with Honor'. The study was enabled with support of the 'German Academic Exchange Service'.

On arriving at the University of Stuttgart Issam had to learn basics of both MPM and German. So the first thing he did was to follow language courses. For getting familiar with MPM he went to the University of Stellenbosch in South-Africa, where Dr. Corné Coetzee taught him the essentials of this method. I am also indebted to Corné for his continuing support of this PhD study. This support was embedded in the Geo-Install IAPP project, being funded by the European Commission within the framework of Marie Curie FP7. Within the framework of the Geo-Install project Issam was enabled to spend a year at 'Deltares' in Delft, The Netherlands. This gave him the experience of working in an industry and of course in again another country. No doubt, Issam had to adjust himself to different ways of working and together with his family he had to get used to different places of living. This cannot always be easy, but to me it would seem that he mostly enjoyed it.

During many years it was a great pleasure to me to work with Issam. This was not only because of his fast progress in science and his abilities as a team player, but also because of his fine personality. I know that I share these feelings towards him with others

who advised him, i.e. Dr. Corné Coetzee and Professor Dieter Stolle from McMaster's University in Hamilton, Canada.

Pieter A. Vermeer
Delft, March 2013

Acknowledgments

I would like to express my greatest gratitude to my supervisor Professor Pieter Vermeer. I am proud that he gave me the opportunity to be one of his PhD students. I really appreciate his patience, continuous support, guidance and encouragement to conduct high quality research. Without Professor Vermeer this thesis would not have been completed. Likewise, I would like to acknowledge with much appreciation the ultimate help of my co-supervisor Professor Dieter Stolle. I consider myself lucky that I had the opportunity to cooperate with him during his sabbatical in Stuttgart. I feel indebted to Dr. Corné Coetzee and Professor Zdzisław Więckowski for the continuous support, advise and feedback. I also wish to thank Professor Bernhard Westrich for the daily discussions and his high interest in my research.

I would like to convey thanks for the German Academic Exchange Service for funding the research and completeness of this thesis. I am also indebted to Deltares for the expert advise throughout my research period. Here, I would like to thank Dr. John van Esch for his support and assistance in the research during my stay in Deltares.

Special thanks to my colleagues in the institute of geotechnical engineering, Stuttgart University for the wonderful time and ultimate support. Here, I thank Thomas Benz, Maximilian Huber, Annette Lächler, Martino Leoni, Axel Möllmann, Ruth Rose, Nadja Springer, Sven Möller, Bernd Zweschper, Ayman Abed, Lars Beuth, Syawal Satibi, Lia Daniela Zaman, Markus Wehnert, Marcus Schneider, Peter Strolle, Reinhold Mößner, Fursan Hamad, Josef Hintner and Stephan Ries. My thanks extend to Professor Minna Karstunen, Dr. Paul Hölscher, Dr. Peter Fokker, Dr. Alexander Rohe, Dr. Sascha Henke, Shuhong Tan, Phuong Nguyen, Wiep Hellinga and Oday Ibrahim.

I would like to thank all my colleagues in Baghdad University. Special thanks to Professor Somer Nacy and Professor Adnan Nagy for their continuous encouragement.

Thanks to my parents, my wife, my brothers, my sisters and my daughters. Without them this thesis would not have been completed.

Issam Al-Kafaji
Stuttgart, March 2013

To the memory of my father...

Declaration

I hereby declare that this thesis is the result of my own research effort except where otherwise indicated. I have only used the resources that are listed in the list of references.

Issam Al-Kafaji
Stuttgart, March 2013

Contents

Abstract

In geomechanics one often encounters large deformations, soil-structure interaction and dynamical problems, e.g., in pile driving and installation of anchors. Moreover, geomechanical phenomena that include excessive movement of soil masses like landslides can pose a danger to human life and property. The numerical simulation of the physics is challenging, particularly if a saturated soil is subjected to dynamic loading, leading to propagation of different waves in the soil.

Because of the reliance of Lagrangian finite element methods (FEM) on a mesh, they are not well suited for the treatment of extremely large deformations of solids. The need for overcoming this limitation urged researchers throughout the last decades to devote considerable effort to the development of more advanced computational techniques. Such techniques include the combination of Lagrangian and Eulerian finite element methods, meshless methods and mesh-based particle methods.

The intent of this thesis is to further develop and extend the material point method (MPM), which is a mesh-based particle method, for use in geomechanics. MPM can be conceived as an extension of FEM, in which soil and structural bodies are represented by Lagrangian particles that move through an Eulerian fixed mesh. The physical properties of the continuum reside with particles throughout the computations (deformations), whereas the Eulerian mesh and its Gauss points carry no permanent information. Hence, MPM combines the best aspects of both Lagrangian and Eulerian formulations and avoids as much as possible the shortcomings of them.

Three novel MPM development are described in this thesis. In the analysis of geomechanical problems that involve dynamics, absorbing boundaries are introduced to prevent the reflection of waves at the selected boundary of the domain. The well-known viscous boundaries, which will continuously creep under load, are modified to viscoelastic boundaries by introducing Kelvin-Voigt elements to limit such non-physical displacements.

The novel extension of MPM to model the behavior of saturated soils under dynamic loading is formulated. Enhancement of volumetric strains is adopted to mitigate the spurious pressure oscillations which plague low-order finite element implementations. The algorithm is applied to predict the generation and dissipation of pore pressures in a sea dike under heavy dynamic loading by wave attack.

Numerical simulation of pile driving is investigated. Results of shallow and deep penetration are presented. Due to the complex behavior of sand in pile driving, a highly non-linear advanced hypoplastic model is to be used for sand. Explicit Euler forward scheme with sub-stepping technique is used in the integration of this model.

MPM is applied to analyze different geomechanical problems, including the collapse of a tunnel face, the instability of a slope and the deep installation of a dynamic anchor. The dynamic MPM can be applied to quasi-static problems. To this end, a local damping

procedure for single and two-phase materials is discussed, being applied to reach fast convergence to quasi-static equilibrium. Such convergence is detected by two proposed criteria on force and energy. Mass scaling is presented as a procedure that allows the use of large time step size for problems, in which inertia effect can be disregarded.

Zusammenfassung

In der Geotechnik ergeben sich häufig Problemstellungen bei denen große Verformungen auftreten, wie etwa Hangrutschungen oder das Einrammen von Pfählen. Die Analyze solcher Fragestellungen stellt eine erhebliche Herausforderung für heutige numerische Methoden dar. Die klassische Finite-Elemente-Methode (FEM) liefert infolge großer Verzerrungen des FE-Netzes oft keine zuverlässigen Ergebnisse. In den vergangenen Jahrzehnten wurden zahlreiche numerische Methoden mit dem Ziel entwickelt, große Verformungen akkurat zu modellieren. Im Rahmen dieser Dissertation wurde der Ansatz der sogenannten Material-Punkt-Methode (MPM) verfolgt und für die Anwendung in der Geotechnik weiterentwickelt.

Kapitel 2 In diesem Kapitel wird eine umfassende Literaturrecherche zur Entwicklung numerischer Methoden geboten. Zu Beginn wird die Entwicklung der Finite-Elemente-Methode beschrieben, angefangen mit der Lösung von Randwertproblemen mittels Variationsrechnung durch Rayleigh und Ritz und der Formulierung des Courant' schen Konzeptes teilweise kontinuierlicher Funktionen. Hieraus entstanden später die Lagrange'sche und Euler'sche FEM, deren Vor- und Nachteile im Weiteren diskutiert werden. Anschließend werden netzfreie Methoden, die *Smoothed-Particle-Hydrodynamics*-Methode, die *Element-Free-Galerkin*-Methode und die *Particle-Finite-Element*-Methode behandelt. Vor diesem Hintergrund wird die Entwicklung netzbasierter Partikel-Methoden dargestellt: Aus der in der Mitte des letzten Jahrhunderts entstandenen *Particle-In-Cell*-Methode entwickelte sich die *Fluid-Implicit-Particle*-Methode und schließlich die in dieser Dissertation behandelte *Material-Punkt*-Methode. Neben einer ausführlichen Darstellung der Entwicklung der *Material-Punkt*-Methode werden in diesem Kapitel auch verschiedene Ingenieuranwendungen dieser Methode vorgestellt. Hierzu zählen die Modellierung granularer Materialien, Rissweitensimulationen, die Simulation von Membranen und der Interaktion zwischen Flüssigkeiten und Festkörpern.

Kapitel 3 In diesem Kapitel wird die klassische Lagrange'sche FEM behandelt, da diese Methode die Basis der Formulierung der MPM darstellt.

Hierzu wird das mathematische Modell eines Festkörperkontinuums hinsichtlich der Erhaltung von Masse, Moment und Energie erläutert. Anschliessend wird das Prinzip der virtuellen Arbeit erläutert, welches zur Herleitung des FEM-Gleichungssystems führt. Integrationsschemen, die zur Zeitintegration gewöhnlicher Differentialgleichungen 2. Ordnung Anwendung finden, werden ausführlich behandelt. Hierbei wird zuerst ein Feder-Masse-System mit einem Freiheitsgrad betrachtet. Anhand dieses einfachen mechanischen Problems wird die Stabilität verschiedener Integrationsschemen untersucht. Anschließend wird das in dieser Dissertation gewählte Schema auf die FEM erweitert, wobei inbesondere auf die Wahl der optimalen Größe eines Zeitschrittes eingegangen

wird. Es folgt die Validierung der durchgeführten FEM-Implementierung anhand numerischer Beispiele.

Kapitel 4 Inhalt dieses Kapitels ist die Erweiterung der FEM zur MPM. Die grundlegenden Konzepte der MPM sowie die Vorteile der MPM gegenüber der klassischen FEM werden erläutert. Nach der Beschreibung der Diskretisierung eines Kontinuums anhand einer Materialpunktmenge wird auf die Initialisierung der Materialpunkte eingegangen. Hierbei wird detailliert die Zuweisung von Masse, Kräften, Zustandsvariablen und anderer Eigenschaften an die Materialpunkte beschrieben. Im Anschluß hieran wird die Initialisierung und Lösung der Bewegungsgleichungen behandelt. Die Randbedingungen des MPM-Gleichungssystems werden ausführlich behandelt. Hierzu zählt die Erweiterung des Verfahrens von Lysmer and Kuhlemeyer [110], welches zur Formulierung eines Kelvin-Voigt-Elementes führt. In diesem Kapitel werden ferner Algorithmen zur Vermeidung von Locking niedrigwertiger Finite-Elemente, die Konvergenz von Simulationen zu quasi-statischem Gleichgewicht sowie der Einsatz künstlicher Dämpfung behandelt. Darüberhinaus wird ein Überblick über in der MPM eingesetzte Kontaktalgorithmen geboten, einschließlich der Validierung eines in dieser Dissertation eingesetzten Algorithmus.

Kapitel 5 In diesem Kapitel wird die Anwendung der MPM auf geomechanische Problemstellungen untersucht, bei denen große Verformungen und Bewegungen von Boden eintreten. Möglichkeiten der Anwendung der MPM im Ingenieurwesen werden aufgezeigt: das dynamische Versagen einer Tunnelortsbrust, das Versagen einer Böschung und das dynamische Eindringen eines Ankers in Boden zeigen deutlich die Möglichkeiten der MPM auf.

Kapitel 6 Viele Problemstellungen der Geotechnik erfordern nebst Modellierung des Bodens auch die Berücksichtigung von Flüssigkeiten im Boden. Die Erweiterung der MPM um die Modellierung von Flüssigkeiten in einem porösen Medium, gekoppelte 2-Phasen-Materialien wie etwa Grundwasser im Boden, wird in diesem Kapitel behandelt. Das Konzept der effektiven Spannungen als auch das mögliche Auftreten von Kompressionswellen in einem gesättigten Boden werden erläutert und die Gleichungen des Zwei-Phasen-Modells abgeleitet. Hierbei werden die Geschwindigkeiten der festen und flüssigen Phase als primäre Variablen verwendet, was zur sogenannten v-w-Formulierung führt. Anschließend wird eine Literaturstudie zu numerischen Algorithmen für die Modellierung von Zwei-Phasen-Materialien präsentiert. In Abschnitt 6.4 wird die FE-Lösung der v-w-Formulierung dargestellt. Diese Ausführungen werden auf die MPM erweitert. Die durchgeführte Implementierung wird validiert und mit analytischen Lösungen verglichen. Abschließend wird die Implementierung auf das Problem periodischen Wellenschlags gegen einen Deich angewendet.

Kapitel 7 Das Rammen von Pfählen stellt eine häufig eingesetzte Gründungsmethode dar. Für die Planung dieser Gründungsmethode ist es erforderlich, Art und Größe

des Rammhammers zu bestimmen, der für die Erreichung der erforderlichen Rammtiefe notwendig ist. Die numerische Simulation des Herstellungsvorganges kann einen wichtigen Beitrag zur Wirtschaftlichkeit dieser Technik leisten. Darüberhinaus ist es auch erforderlich, die Möglichkeit eines Versagens des Pfahls vorherzusagen, sollte dieser überbeansprucht werden. In diesem Kapitel wird eine dynamische, dreidimensionale Simulation des Rammens eines Pfahls in Sand mittels der MPM vorgestellt. Das Verhalten des Bodens unter zyklischer Belastung wird mit einem adaptierten hypoplastischen Sand-Modell simuliert, welches beispielhaft an Laborversuchen an Schlaberndorfer Sand validiert wird. Abschließend werden die Ergebnisse verschiedener Eindringtiefen miteinander verglichen und diskutiert.

Kapitel 8 In diesem abschließenden Kapitel werden die Ergebnisse zusammengefaßt sowie Empfehlungen für zukünftige Forschungen behandelt.

Chapter 1

Introduction

In geomechanics one often encounters problems involving large deformations and large movement of soil masses. As examples of such problems, one may consider landslides, tunnel collapse and pile driving. Figure 1.1 shows a disastrous landslide, in which the soil experienced considerable movement. The collapse of a tunnel is shown in Figure 1.2, with a crater being formed on the ground.

The analysis of such problems is challenging. At times, it may not be possible for classical finite element method (FEM) formulations to provide reliable solutions, owing to severe mesh distortions that accompany updated Lagrangian approaches. Although Lagrangian-Eulerian (ALE) methods largely overcome issues associated with mesh distortion, re-meshing may still be required for some applications. Furthermore, the issue of the need to remap state variables associated with material points is a drawback, which is of particular concern when the history of the material must be taken into account. This has led to the development of meshless methods and mesh-based particle methods such as the material point method (MPM), which is favored in this thesis. An attractive feature of MPM is that properties and state variables reside with material points that are not fixed to a mesh, but move through the mesh allowing possibilities of modeling large deformations.

The dynamic nature of the considered problems make the corresponding numerical simulation even more challenging. Predictions of the deformation pattern of the tunnel face and the ground settlement in real time require a fully dynamic formulation. In addition to the high dynamics in the pile driving, attention must be paid to the formulation of soil-structure interaction.

It is also common in geomechanical engineering to encounter problems that include the coupling of the solid and fluid phases. Such coupling adds considerable complexities to the mechanical behavior of the material and the corresponding numerical simulations. Algorithm that is capable of capturing the physics of saturated soils under dynamic loading, including the propagation of different waves, is to be formulated and tested.

1.1 Basic objectives

The ultimate objective of this thesis is to further develop, extend and exploit the potentials of the material point method to provide a new horizon for the modeling of problems in geomechanics. Examples to be analyzed include installation of anchors, slope slide and the collapse of a tunnel face. Special attention is to be paid to the MPM simula-

Figure 1.1: Landslide in Hongkong (1972) [179]

Figure 1.2: Collapse of tunnel face in Munich (1994) [180]

tion of pile driving in sand, being modeled by a highly advanced non-linear constitutive model. To this end, MPM will have to be extended to include absorbing boundaries to avoid waves reflection.

This study will also focus on another extension of the material point method to analyze coupled dynamic, two-phase boundary-value problems via a velocity formulation, in which solid and fluid phase velocities are the variables. This extension is applied to the simulation of the dynamic generation and dissipation of pore pressures in a sea dike being attacked by cyclic waves.

1.2 Thesis layout

Including this introduction, this thesis is arranged in eight chapters. An extensive literature review for numerical methods is presented in **Chapter 2**. Lagrangian and Eulerian finite element methods and other methods resulting from the combination of them are considered with the pros and cons of each method. Also reviewed, are the mesh-based particle methods with special attention being paid to the original development and application of the *material point method*.

The formulation of the classical Lagrangian finite element method is presented in **Chapter 3**. First of all, the mathematical model of a solid continuum is described. It includes conservation of mass, conservation of momentum and energy, kinematics, stress-strain relation, as well as boundary and initial conditions. A detailed study of one-step integration scheme for a second-order ordinary differential equation (ODE) is presented. The conservation properties and other attributes of the scheme are investigated and attention is paid to the selection of the time step size for discrete systems when adopting a conditionally-stable time integration scheme. Validation of the FEM model with some numerical examples is given with some concluding remarks.

The extension of FEM to MPM is formulated in **Chapter 4**, in which the basic concept of MPM is presented with the advantages of the method over the classical FEM. The application of the boundary conditions and the solution procedures in the framework of MPM are presented. The treatment of spatially unbounded domains using absorbing boundaries is discussed. An approach to deal with locking in low-order element is explained. The convergence of the solution to a quasi-static equilibrium and the application of artificial damping are discussed. A penetration problem is considered to validate the implementation of MPM.

The application of the material point method to geomechanical problems involving large deformation and movement of soil masses is presented in **Chapter 5**. Examples of this chapter include the dynamic collapse of a tunnel face, the instability of a slope consisting of frictional material and the installation of a dynamic anchor.

The extension of MPM to analyze problems involving the coupling of solid and fluid phase is formulated in **Chapter 6**. Numerical issues in solving two-phase problems are discussed, including the modeling of unbounded saturated domains, spurious spatial pressures, damping for water and solid phase as well as mass scaling. The implementation is validated with problems having analytical solutions. An application example of wave attack on sea dike is analyzed at the end of this chapter.

Simulation of pile driving in sand is discussed in **Chapter 7**. The pile is assumed to behave according to the linear elastic constitutive relation so that the probability of its damage is excluded from the present study. The sand is to be modeled as advanced as possible at the present state of soil constitutive models. The attention focused on the drivability of the pile considering sand with different initial densities. The results of shallow and deep penetration with varying skin friction, pile damping and the driving pressure are presented in this chapter.

The concluding remarks of the present study with outlook to future work are presented in **Chapter 8**.

Chapter 2

Large deformation: A review

For complex problems in continuum mechanics, analytical (closed-form) solutions are not possible because of the restrictions concerning geometries, loads and boundary conditions. Therefore, it became essential in engineering that alternative approximate solution procedures were adopted. During the last decades, several numerical methods were developed and became attractive for many fields of engineering. In this chapter, we present an overview of some numerical methods that are capable of modeling large deformations. We consider three main methods, namely the finite element methods, meshless methods and mesh-based particle methods.

2.1 The finite element methods

The finite element method (FEM) is a numerical technique that approximates the solutions of boundary-value problems. FEM has been successfully used for decades to analyze a wide range of problems in both solid and fluid mechanics. In this section, we consider a brief historical review of the main line of FEM development.

Approximate solutions of boundary-value problems using calculus of variations were established by Rayleigh [141] and Ritz [142]. In this approach, the differential equation is approximated by means of a test or trial function that is selected to be continuous over the domain and satisfies the boundary conditions. The ansatz or the approximation of the test function is a combination of a finite number of coordinate functions, each being weighted by a constant coefficient. The test function is substituted in the variational approximation of the problem. The extremum of the variational expression leads to a set of linear equations that are solved for the constant coefficients of the test function. Rayleigh [141] was the first to use the calculus of variation in 1877 in the calculation of the fundamental natural frequency for different systems, e.g., bars and plates. In 1909, Ritz [142] applied the variational approach to vibrating plates.

Another method to approximate the solutions of differential equations is the method of weighted residuals, in which the test function is used together with an integral form of the differential equation to minimize the error in an average sense over the domain of the problem. The method of weighted residuals includes many approximation techniques, e.g., Galerkin method [69], subdomain method [28] and collocation method [68]. An excellent review of the method of weighted residuals and more detailed bibliography are presented in reference [65]. At present, the Galerkin method is the most popular method for finite element formulation.

Although the earlier variational and weighted residuals methods provided the basis

for approximating the solution of boundary-value problems, they had a drawback. The trial function that is valid for the whole domain may require an excessive number of terms, making it cumbersome to use for problems with complex geometry and boundary conditions. In 1943, Courant [49] introduced the concept of piecewise continuous functions in the context of variational methods. He applied the concept to triangular subdomains to solve a plane torsion problem. Courant's concept might be considered as the real start of the finite element method. Subsequently, a matrix structural analysis method was developed by Argyris and Kelsey [4] in 1954 and the stiffness matrix of two-dimensional elements was first derived by Turner et al. [175] in 1956. The name finite element was first introduced by Clough [44] in 1960 when using triangular and rectangular finite elements for plane-stress analysis.

Turner et al. [176] further developed the method in 1960 for large deformation and thermal-mechanical coupling. Zienkiewicz and Cheung [204] wrote the first finite element textbook in 1967. A year later, Zienkiewicz et al. [206] published an article about the extension of the method to visco-elastic stress analysis. Costantino [48] was the first to use the finite element method for dynamic analysis in 1967. Belytschko and Hsieh [18] presented the use of the method for transient analysis of large-displacement, small-strain problems with material non-linearities. They presented the formulation of the constant strain triangular elements and Euler-Bernoulli beam elements. They compared the results with the available experimental data for several examples.

In the framework of the finite element method, there are two basic formulations, namely *Lagrangian FEM* and *Eulerian FEM*. The former is widely used in modeling solid mechanics, whereas the latter is mainly used in modeling fluid mechanics. Other finite element formulations were developed from the combination of Lagrangian and Eulerian FEM and named as *arbitrary Lagrangian-Eulerian* (ALE) formulations. The underlying mechanics of different formulations are discussed in the next sections with emphasis on the advantages and drawbacks of each one.

2.1.1 Lagrangian versus Eulerian FEM

In continuum mechanics, there are two different view points describing the deformation of a continuum, namely Lagrangian description and Eulerian description. In Lagrangian description, the time and material coordinates are used as independent variables to describe the motion and other physical properties of the material with time. The frame of reference in Lagrangian description is the initial (undeformed) configuration of the continuum and sometimes it is referred to as material description. On the other hand, Eulerian description uses the spatial coordinates and time to track the motion and other physical properties at points fixed in space while the material is passing through with time. It uses the spatial configuration as a frame of reference.

In the framework of Lagrangian finite element, two formulations can be developed depending on the frame of reference used to calculate strains and stresses. If the initial material coordinates are used to determine stresses, the formulation is then called a total Lagrangian FEM. An updated Lagrangian FEM arises if the updated material coordinates are used to calculate stresses. In 1975, Bathe et al. [15] presented the total and

updated Lagrangian FEM for large deformation. The formulation of both methods for a three-dimensional beam element were derived for large displacement and large rotation analysis by Bathe and Bolourchi [14] in 1979. They showed that both formulations yield identical element stiffness matrix and nodal force vectors, but that the updated Lagrangian FEM is more effective from computational view point. In their textbook, Belytschko et al. [21] showed in more detail that both total and updated Lagrangian FEM lead to identical results and the choice between them is just a matter of convenience.

In Lagrangian FEM, the nodes are always coincident with the material points, see Figure 2.1. Hence, the nodes that are initially located at the boundary of the continuum will always remain on the boundary throughout the computations, i.e., the free surface of the continuum is well defined in the framework of Lagrangian FEM, allowing easy track of the interface between different materials and simple imposition of the boundary conditions. Another advantage of Lagrangian FEM is that, by definition, it does not allow the material to flow between elements and hence history dependent material behavior can be easily handled as the quadrature points remain coincident with the material points. However, the mesh distortion problem makes the method cumbersome in modeling very large deformation.

Unlike Lagrangian FEM, in Eulerian FEM the computational mesh is kept spatially fixed while the material is deforming in time, see Figure 2.1. This allows the capability of handling large deformation without the problem of mesh distortion that appears in the updated Lagrangian FEM. As the computational mesh is completely decoupled from the material, convective terms appear in Eulerian FEM, introducing numerical difficulties because of their non-symmetrical properties [60]. In contrast to Lagrangian FEM,

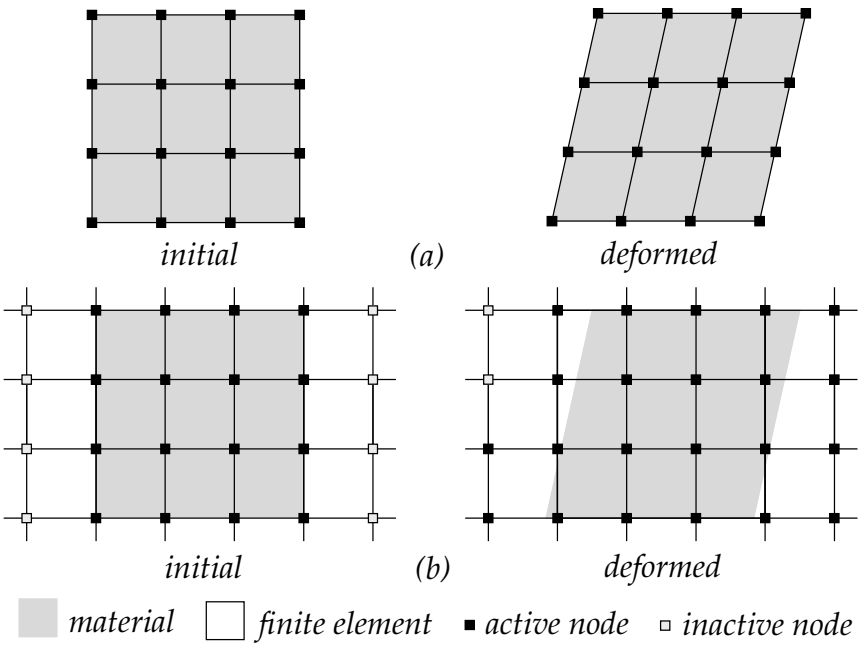


Figure 2.1: Initial and deformed configurations of a deforming body with (a) Lagrangian mesh and (b) Eulerian mesh

difficulties arise in modeling interface between materials as well as the application of the boundary conditions. Using coarse meshes with Eulerian FEM leads to smearing of the state variables. High quality results need fine meshes and are computationally expensive [147]. Moreover, the convection of materials between elements yields elements with different materials, leading to a change of the real material properties in these elements and reducing the quality of predictions.

2.1.2 Combined Lagrangian-Eulerian methods

The combination of Lagrangian and Eulerian descriptions results the so-called *arbitrary Lagrangian-Eulerian* (ALE) methods. In ALE, the relation between the computational mesh and the continuum can be arbitrary and selected by the user, allowing possibilities of modeling large deformation by combining the best aspects of Lagrangian and Eulerian descriptions and avoiding as much as possible the drawbacks of each of them. The state variables are mapped from the old (distorted) mesh to a new one. The mesh updating methods have great influence on the success of ALE. The mesh can be completely regenerated or adapted. Mesh adaptation keeps the same nodes and topology, but the nodes are relocated to avoid element distortion. For more details about ALE and the mesh updating methods, the reader is referred to reference [60] and references therein.

The so-called *coupled Eulerian-Lagrangian* (CEL) method is one of the most effective ALE methods in modeling penetration problems. In CEL, the target is discretized with Eulerian mesh, whereas the penetrometer is discretized with Lagrangian mesh. The interaction between the two meshes is modeled using contact algorithm selected by the user, see reference [89] for CEL applications in geotechnical engineering.

2.2 Meshless methods

In the meshless methods the nodes (points) connectivity is continuously adjusted as the continuum deforms, avoiding the problem of mesh distortion. The approximation in the meshless methods can be based on moving least squares, kernels or partitions of unity [20]. The *smoothed particle hydrodynamics* (SPH) is the oldest meshless method [109]. The SPH is a particle method, in which the closed form of the partial differential equation (PDE) is discretized using collocation methods. The SPH does not require a pre-defined mesh to calculate the spatial derivatives, which is a clear distinction from the material point method, see Section 2.3. For more details about SPH, the reader is referred to the publications of Monaghan and coworkers (e.g. [123]).

The *element-free Galerkin* (EFG) method is a relatively younger meshless method, in which the trial functions for the weak form are constructed using moving least squares interpolants, see reference [19].

Oñate et al. [98, 134] also classify the *particle finite element method* (PFEM) as a meshless method as it involves the meshless finite element interpolation. In PFEM, the nodal points represent the particles and the computational mesh is constructed by connecting these points. The mesh is then used to solve the governing equations in Lagrangian fashion. Large deformation requires frequent remeshing (adjustment of the connectivity).

The research effort on the meshless methods is increasing and many of these methods were developed recently. An extensive literature review is out of the scope of this thesis and the reader is referred to the book of Liu and Liu [105] for more abroad review.

As mentioned by Belytschko et al. [20], the meshless methods are suited to problems involving large deformation, but they still require considerable improvement in the computational efficiency. It was also shown that the computational cost of the element-free Galerkin (EFG) method is much higher than that of FEM.

2.3 Mesh-based particle methods

Unlike the meshless methods, the mesh-based particle methods involve a combination of a pre-defined background mesh and material points moving within this mesh. Only those methods most relevant to this thesis are presented in this section

2.3.1 Particle-in-cell method

The original development of the *particle-in-cell* (PIC) method can be tracked back to 1955. It was developed at Los Alamos by Harlow to simulate problems in fluid dynamics. Although the objective at that time was to develop a method that is capable of handling problems in two and three dimensions, the early development of PIC was made for one-dimensional problems [80].

However, it was shown later that the original PIC was suffering from energy dissipation [36, 37, 61, 102, 121, 132]. The reason for such dissipation was attributed to the fact that Harlow's original formulation was *partially Lagrangian*, in the sense that only a mass and position were assigned to each particle. Other physical properties such as velocity, momentum and energy were only temporarily mapped to the particles to convect information between cells at the end of the time step. The properties assigned to particles at the end of the time step were weighted by the mass of each particle. In other words, if a particle moves from a cell to another the momentum and energy associated with it are subtracted from the old cell and added to the new cell, but not stored permanently at particles. In this respect, the method might be considered more Eulerian [132]. In spite of this drawback, PIC was successfully used in modeling problems in fluid dynamics involving large distortion [63, 80–84].

Since the early development of PIC, many studies have been carried out to further improve the method and increase its numerical accuracy. In 1983, Nishiguchi and Yabe [132] developed a more accurate scheme for the convection between particles and grid point. Although they used the partially Lagrangian representation of particles as in the original PIC, they improved the accuracy of the advection of momentum and energy to a second order accuracy in space. During the seventies and eighties, considerable improvements in the PIC were achieved, see references [38, 61, 102, 120, 121, 168]. The main effort concentrated on modifying the original PIC from *partially Lagrangian* representation of particles to *fully Lagrangian*. In this modification, all physical quantities such as momentum and energy were assigned to particles. The grid was then used only to

get the solution of the governing equations. As a consequence of using full Lagrangian approach, a considerable reduction in the numerical diffusion was achieved. This reduction was however compromised by the need for more storage capacity than required by the partially Lagrangian, which was an issue when solving complicated problems at that time [120, 132].

2.3.2 Fluid-implicit particle method

In 1986, Brackbill and Ruppel [36] used the full Lagrangian PIC as a basis to extend the applicability of the method by using adaptively zoned mesh. They developed what is so-called *fluid-implicit particle* (FLIP), which is a PIC formulation, in which the particles carry all physical properties of the continuum, e.g., mass, momentum, energy and constitutive properties. As a consequence of using adaptive mesh, complex geometries could be modeled with FLIP and better accuracy could be achieved. Furthermore, they clearly separated the computational cycle of PIC to Lagrangian and convection or Eulerian phase, which was very helpful in understanding the method. In addition, this separation allowed the use of existing finite difference algorithms to solve the discrete equations in the Lagrangian phase, as particles play no permanent role in this phase. Further understanding and more stability analysis of FLIP and its applications in hydrodynamics are given in references [34, 35, 37].

In 1991, FLIP was extended by Sulsky and Brackbill [158] to model problems involving interaction between elastic bodies and fluids. According to the fact that in PIC particle velocities are single valued in the mapping and re-mapping procedure between grid points and particles, the method is naturally capable of handling non-slip contact between different bodies without any special algorithm [165].

In 1992, Burgess et al. [41] showed that the mapping of velocities from particles to nodes can be made conservative on the basis of a mass-weighted least square procedure. This procedure required inverting the mass matrix to calculate velocities from momentums. They also showed that the conservation of kinetic energy could be fully achieved when using a consistent-mass matrix. However, inverting such matrix was found to be computationally expensive. Thus the lumped-mass matrix, which is a diagonal matrix with entries corresponding to the row sum of the consistent-mass matrix, was used as an alternative solution having computational and storage advantages. It should be realized that the cost of using a lumped-mass matrix is some numerical dissipation in the kinetic energy [41]. However, it is widely used in literature (e.g. [47, 162]). The FLIP method is fairly well developed for fluid flow problems. The discrete equations of motion are solved using standard finite difference methods [34–37, 41].

2.3.3 Material point method

The *material point method* (MPM) is a sophisticated numerical technique suited to model large deformation [47, 126, 162, 163, 192]. In MPM, the continuum is represented by *Lagrangian points*, called *material points* or *particles*. Large deformation is modeled by particles moving through an Eulerian fixed mesh. The particles carry all physical properties

of the continuum such as mass, momentum, material parameters, strains, stresses as well as external loads, whereas the Eulerian mesh and its Gauss points carry no permanent information. At the beginning of the time step, information is transferred from particles to the computational mesh. The mesh is then used to determine the incremental solution of the governing equations in a Lagrangian fashion. At the end of the time step, the solution is mapped from the mesh back to particles to update their information.

Through this approach, MPM combines the best aspects of both Lagrangian and Eulerian formulations while avoiding some shortcomings of them. Indeed, numerical diffusion associated with the convective terms in Eulerian approach does not appear in MPM solution. Furthermore, the problem of mesh distortion, which an updated Lagrangian solution shows for large deformation, is avoided in the framework of MPM [161, 165]. The material point method can eventually be viewed as an extension of the particle-in-cell method. The following literature study on MPM is divided into paragraphs for the sake of easy track by the reader.

Early development on MPM In 1993, Sulsky et al. [164][a] considerably extended the FLIP method to the application for solid mechanics. The weak formulation and the discrete equations were casted in a form that is consistent with the traditional finite element method. Furthermore, they applied the constitutive equation at each single particle, avoiding the interpolation of the history-dependent variables, as the particles are tracked throughout the computations. Given this considerable extension, the method was then made capable of handling history-dependent material behavior. Elements having material with different parameters or different constitutive equations were automatically treated which is a clear advantage over Eulerian FEM. They considered numerical examples with large rigid body rotation and showed that the energy dissipation which tends to occur in Eulerian approach did not occur in their approach. This extension was then applied to different normal and oblique impact of objects in plane-strain condition with elastic and strain hardening plastic material behaviors [161]. In the same year (1993), Sulsky and Schreyer [160] extended the application of PIC to incorporate constitutive laws expressed in terms of Jaumann rate of stress. Further applications of PIC method to solid mechanics are given in reference [166].

In 1996, Sulsky and Schreyer [162] named the method as the *material point method* (MPM) and presented its axisymmetric formulation. They applied MPM to upsetting of billets and Taylor impact problems. They also incorporated the thermal effect in the constitutive equation.

MPM in granular materials In 1999, Więckowski et al. [194] applied the MPM to simulate the problem of silo discharge, which involved flow of granular material. The formulation of a frictional contact algorithm and the solution procedure of contact problems were presented in the framework of MPM. They used triangular elements with linear shape functions to discretize the problem. Furthermore, a formula for calculating the

[a] Their work was first documented as a technical report in 1993 [164] and was later published as a journal article in 1994 [165].

critical time step for MPM with one-dimensional linear element was suggested and derived from stability condition. It was shown that, among the mesh size and the wave speed, the critical time step size in MPM is significantly affected by the number and position of particles in the elements. Moreover, they showed that MPM requires, in general, a smaller time step size than FEM. Other publications of Więckowski on the material point method are references [190, 191]. The macroscopic stress-strain response of dry granular materials subjected to compression was investigated by Sulsky [157]. She showed that MPM can reproduce the experimental observations of stiffening under compression. Bardenhagen and Brackbill [7] used MPM to model stress bridging and localization in granular materials. Observations for both quasi-static and dynamic loading were discussed. Coetzee [45] applied the material point method to model the flow of granular material in front of flat bulldozer blades and into dragline type buckets. His formulation was based on Cosserat continuum. He compared MPM results to experimental measurements as well as results obtained by the discrete element method.

Quasi-static MPM All previous development of the material point method were based on dynamics. Guilkey et al. [74] were among the first to formulate a quasi-static MPM to simulate slow rate of loading. They applied their formulation to the modeling of multicellular constructs. An implicit integration scheme was used to integrate the equilibrium equation. They deviated a little from the traditional MPM by not resetting the mesh after each time step. The reason for this was to mitigate the error associated with particle crossing between elements, which is known to be troublesome for quasi-static analysis as there are no inertia forces [201]. However, it is felt that this approach should be used with great care as it might lead to severe mesh distortion. Other work on quasi-static MPM was carried out by Beuth et al. ([25–27, 182]).

Contact algorithms in MPM It was previously mentioned that, a natural consequence of MPM is the capability of simulating non-slip contact between different bodies without a special algorithm. However, in many engineering problems a contact algorithm is required to model the relative motion at the interface between the contacting bodies. In 2000, Bardenhagen et al. [9] developed a frictional contact algorithm to model interaction between grains in granular materials. The algorithm allows sliding and rolling with friction as well as separation between grains, but does not allow interpenetration. The strength of the algorithm is the automatic detection of the contact nodes, i.e., a predefinition of the contact surface is not required. In addition, special interface elements to model the contact are not required. The algorithm is applicable to model contact between solid bodies. It was further improved by Bardenhagen et al. [12] and applied to simulate stress propagation in granular materials. Although there exist other contact algorithms (e.g. [95, 194]), Bardenhagen's algorithm is the one most used in MPM literature [3, 10, 11, 45].

Fracture and crack growth in MPM Considerable research effort was put in exploiting the capability of the material point method to model dynamic fracture and crack

growth. In 2002, Tan and Nairn [169] developed a method to calculate fracture parameters, e.g., the dynamic energy release rate. They combined the method with adaptive meshing technique to refine the mesh around the crack tip, which is necessary to capture the process of crack growth. Furthermore, they used a consistent-mass matrix to accurately calculate the energy release rate. More development and applications of the material point method in modeling fracture and crack growth are presented in references [55, 76, 77, 126, 128, 129].

Membranes in MPM In 1997, York II [195] developed modifications to the material point method to allow the simulation of thin membranes. He used the algorithm to study the interaction between the membrane and compressible fluids. His work was then used as a basis to further development of the membrane formulation within the material point method. Ionescu et al. [99] used the membrane algorithm to simulate biological soft tissues subjected to penetrating trauma. The response of geomembranes to quasi-static loading condition due to the settlement in landfills was studied by Zhou et al. [202]. The MPM membrane theory was also applied to solve several complex problems, e.g., the pinching of a membrane and a pressurized membrane, see references [196, 197].

Implicit integration in MPM It is usual in the dynamic material point method to advance the solution in time using explicit integration schemes [6, 43, 146, 163, 200]. However, in problems involving slow rate of loading, i.e., quasi-static problems, the flow of the material is much slower than the speed of the wave propagation in the material. Hence, employing an implicit time integration scheme reduces the computational time considerably. MPM was used together with implicit schemes in several studies. Cummins and Brackbill [52] considered an implicit formulation of the material point method and applied it to the simulation of quasi-static loading of granular materials. They adopted the matrix-free Newton-Krylov algorithm to reduce the computational time and avoid the construction of the tangent stiffness matrix. A similar approach with a slightly different time discretization was used by Sulsky and Kaul [159]. In the latter, it was shown that the use of implicit solvers with large time step size results in some energy dissipation and loss of accuracy when applied to problems of high frequency content. In contrast to references [52] and [159], Guilkey and Weiss [72] explicitly formed the tangent stiffness matrix and used Newton's method together with Newmark integration scheme to solve the equilibrium equations in time.

Other examples of using implicit integration with MPM are found in the work of Love and Sulsky [107, 108]. They showed that an implicit formulation of MPM can be unconditionally stable. They used a consistent-mass matrix together with the implicit scheme, adding considerable complexity to the algorithm [155]. Apart from dynamic MPM, there exist quasi-static formulation with implicit solvers (e.g. [26, 27, 74]).

Generalized interpolation MPM In 2004, Bardenhagen and Kober [8] generalized the MPM solution procedure by applying a Petrov-Galerkin discretization scheme. They used the shape functions together with particle characteristic functions in the variational formulation. Different combinations of the shape functions and particle characteristic

functions resulted in a family of methods named, by Bardenhagen and Kober, as the *generalized interpolation material point method* that is abbreviated as GIMP. This generalization allowed the material point method to be derived as a special case GIMP. Their main motivation to investigate GIMP was to eliminate the numerical noise associated with MPM when particles cross element boundary. Indeed, they showed in one-dimensional examples that, GIMP is capable of eliminating the noise in stresses observed in MPM solution. However, the use of Petrov-Galerkin discretization scheme deviated the method more towards meshless methods [8]. GIMP was used in several studies, see references [2, 55, 111, 187].

MPM in geomechanics The studies on using the material point method in modeling geotechnical problems are limited. Andersen [3] used MPM to model the dynamic collapse of slopes. Coetzee et al. [47] used it for studying the large deformation problem of anchor pull-out. The ultimate capacity of the anchor pulled in sand with 45° was predicted with 10% difference from the measured data. Furthermore, it was shown that MPM can accurately capture the movement of the free surface. The group of Vermeer applied the material point method mainly for modeling problems in geomechanics [25–27, 182]. Examples of geotechnical problems solved using a quasi-static MPM are elastic-plastic slope problem and retaining wall problem. Modeling the behavior of saturated soils under rapid loading, e.g., pile driving in saturated soils is a challenge. It requires a method that is capable of modeling large deformation and accounting for dynamic generation and dissipation of pore water pressures, a phenomenon that is well-known and of great importance in geotechnical community and called consolidation. The work of Zhang et al. [199] is the only publication, we are aware of, on the application of the material point method on the modeling of dynamic analysis of saturated porous media.

MPM in fluid-structure interaction Guilkey et al. [75] developed a numerical scheme for fluid-structure interaction. In their scheme, they used the material point method to model solid field in a Lagrangian frame, while an Eulerian frame was used for the fluid. The fact that MPM uses a temporary mesh to calculate incremental solution of the governing equations, allowed them to use the same Eulerian mesh for the solid and fluid. Moreover, they combined their approach with adaptive mesh refinement procedure in order to refine the mesh in parts of the domain and hence improved the computational efficiency. They applied the scheme to simulate explosions of energetic devices, which are metal containers filled with high explosive materials such as bombs and rocket motors. References [73, 85, 136, 152, 198] contain more details and applications of fluid-structure interaction and multiphase schemes using MPM. The use of the material point method in modeling multiphase drag interactions with several modeling strategies was investigated by Mackenzie-Helnwein et al. [118].

Improvement of MPM accuracy Fundamental studies were carried out on the improvement of MPM accuracy and reduction of some numerical errors, e.g., grid crossing error and energy conservation error. Steffen et al. [153] studied the reduction of quadrature error and grid crossing error. They concluded that the use of a smooth basis func-

tion such as quadratic and cubic B-splines reduces these errors considerably. Further investigations of different types of interpolation in the material point method were carried out by Andersen and Andersen [1]. Love and Sulsky [107] investigated the energy consistency of the material point method when used together with thermodynamically consistent hyperelastic-plastic materials. They formulated the material point method in a way that can dissipate energy and conserve momentum to achieve the desired properties of energy-consistent method. They considered a mixed discretization procedure to avoid the volumetric locking, in which they split the governing equations into volumetric and deviatoric parts, allowing the use of different interpolation for each part. A similar mixed approach was already used by Moresi et al. [125] in the framework of another PIC formulation. Methods of updating strain and stress rates in MPM and energy conservation issues associated with each method were subjects to several studies.

Bardenhagen [6] showed that the original MPM algorithm by Sulsky et al. [165] does not conserve energy although it conserves mass and momentum by construction. In the original MPM, the nodal velocities that are calculated directly from the integration of nodal accelerations, are used to update particle strain and stress rates. This procedure is referred to as *update stress last* (USL), see references [6, 126]. It is known to be numerically unstable having serious problem when a node gets almost zero mass. This situation occurs in some scenarios like, when a single particle just entered a previously empty element and located very close to the boundary of this element. A solution to this problem was suggested by Sulsky et al. [166]. They modified the algorithm such that it has better numerical properties and much better energy conservation features. In their modification, they used the updated particle momentums to calculate the nodal velocities, which are then used to update particle strain and stress rates. Nairn [126] named this approach as *modified update stress last* (MUSL).

An alternative algorithm to improve the energy conservation properties of the original MPM was proposed by Bardenhagen [6]. In his algorithm, he updated strain and stress rates at the beginning of the time step from the velocities of the previous time step. This approach is referred to as *update stress first* (USF). The main difference between MUSL and USF is that the former calculates the nodal velocities from particle momentums at the end of the time step, i.e, before updating particles position, whereas USF calculates the nodal velocities from particle momentums at the beginning of the next time step, i.e., after updating particles position [126]. The two approaches were compared and scrutinized by Nairn [126]. He observed that MUSL slowly dissipates energy while USF slowly increases energy. However, MUSL might be more logical choice as the algorithm is almost consistent, except for some details, with the well-established FEM solution procedure. Moreover, the slow dissipation observed in MUSL might be required in some cases, acting as an artificial damping in the system. Issues like MPM accuracy, energy conservation, implementation choices and extension of its applicability are further discussed in other publications (e.g. [42, 143, 153–155, 185–188]).

MPM computational cost The material point method consumes less computational time compared to the meshless methods. Unlike the meshless methods, the incremental solution in MPM is obtained on a pre-defined background mesh and the CFL (Courant,

Friedrichs and Lewy) condition depends on the mesh spacing rather than the material point spacing [146]. The time step size may decrease significantly during computations in the meshless methods. Ma et al. [112] showed in particular example of hypervelocity impact that the SPH time step size was reduced by a factor of 3.8 during computations due to the considerable reduction in the material point spacing resulting from the severe compression of the material, whereas MPM kept the time step size constant throughout the computations. In addition, the time consuming neighbor searching, which is not required in MPM, is essential in most meshless methods like SPH and EFG.

Other applications of MPM The material point method finds itself applicable in a wide spectrum of engineering problems. For instance, Nairn [127] applied MPM to the modeling of transverse compression and densification of wood. He showed that MPM has the potential to reproduce some physical phenomena in wood that are beyond the capability of the finite element method, e.g., wood anatomy and variation in wood structure upon loading. Other studies on using MPM in modeling wood are found in references [128, 130]. Banerjee [5] exploited the capability of the material point method to model fragmentation of objects. He applied it to simulate cylinders that fragment due to explosively expanding gases generated by reactions in a high energy material contained inside. Examples of using MPM to model fragmentation on concrete walls are found in reference [93]. Furthermore, the method was used to model the physics of foam densification (e.g. [13, 40]). It was also applied by Sulsky et al. [167] in the simulation of ice transport. However, other formulations of PIC were already used to model sea ice transport (e.g. [67, 96]).

Schreyer et al. [146] applied the material point method to the modeling of delamination or decohesion, which is a failure mode that occurs in layered composite materials. They exploited the capability of MPM of tracking history variables such as decohesion without the need for remeshing and mapping history variables. Other applications of using the material point method together with a decohesion constitutive models are available in reference [163].

It is of interest to point out that there exists a recent mesh-based particle method with almost the same underlying mechanics as that of MPM. The method is called *Lagrangian integration point finite element method* [125]. It differs from MPM mainly in the way of calculating the integration weights. In MPM, they are calculated from the initial configuration and updated during computations, whereas in the other method, they are calculated by defining the field quantity that needs to be integrated as a polynomial of a certain order. The coefficients of the assumed polynomial are then equated to the coefficients of the usual quadrature expansion of the field quantity. Although this method is not widely used, it was applied to some engineering problems, e.g., simulation of landslides [140], modeling of geomaterials [125] and viscous flow [124].

Summary of literature review After this literature review, one can conclude that the material point method has been proven as a robust numerical tool, having the capability of modeling large deformation. It has been successfully employed in modeling a wide range of problems in different engineering fields. MPM avoids most of the shortcom-

ings that appear in both Lagrangian and Eulerian FEM. Another strong advantage is that, MPM can be viewed as an extension of the well-developed finite element method. Hence, the long tradition and experiences of FEM can be easily exploited and used with MPM. Furthermore, when compared to meshless methods, it is less complex and computationally more efficient.

Considering these *"beautiful"* features of MPM, we find a logical justification to our strong motivation to put our effort into further development of the method and make it applicable to practical problems in geomechanical engineering, especially for those involving two-phase dynamics.

Chapter 3

Mathematical and numerical models: FEM

We already mentioned in Chapter 2 that the material point method can be viewed as an extension of the finite element method and that the solution procedure of the equilibrium equations within a time step in MPM is consistent with that of Lagrangian FEM. For this reason, we find it appropriate to devote this chapter to the classical Lagrangian finite element method. This way we will have better understanding of the MPM basics and it will be easier to highlight the differences between FEM and MPM, which is considered in the next chapter. Section 3.1 considers the mathematical model of a solid continuum, including the conservation of mass, momentum and energy; kinematics; stress-strain relation; as well as boundary and initial conditions.

In Section 3.2, the weak form of momentum balance is considered, which leads to the well-known virtual work equation that is used to develop FEM equations in Section 3.3. The assemblage procedure of the global matrices from the element matrices is discussed as well in this section. Special attention is paid in this section to the lumping procedure of the mass matrix.

Time discretization follows in Section 3.4, in which a detailed study of one-step integration scheme for a second-order ordinary differential equation (ODE) is presented. The scheme is applied first to a single degree-of-freedom spring-mass system and stability conditions, conservation properties and other attributes of the scheme are investigated. The application of the scheme is then extended to FEM equilibrium equations. Attention is paid to the selection of the time step size for discrete systems when adopting a conditionally-stable time integration scheme. Validation of the FEM model with some numerical examples is given in Section 3.5. Some concluding remarks for this chapter are discussed in Section 3.6.

3.1 Mathematical model

Let us consider a continuum that occupies a domain $\Omega_0 \subseteq \Re^3$ at time t_0 and a domain $\Omega \subseteq \Re^3$ at time t, where $t \in [t_0, t_f]$. The domain Ω_0 represents the initial state of the continuum and is referred to as the *initial configuration* or the *undeformed configuration*, whereas the domain Ω represents the state of the continuum after it has experienced deformation. This domain is referred to as the *current configuration* or the *deformed configuration*, see Figure 3.1. The coordinates of a material point in the current configuration at time t are denoted by the vector $\boldsymbol{x}$, whereas the vector $\boldsymbol{x}_0$ denotes the coordinates of this material point in the initial configuration at time t_0. In Cartesian coordinate system, the vectors $\boldsymbol{x}$ and $\boldsymbol{x}_0$ are defined as

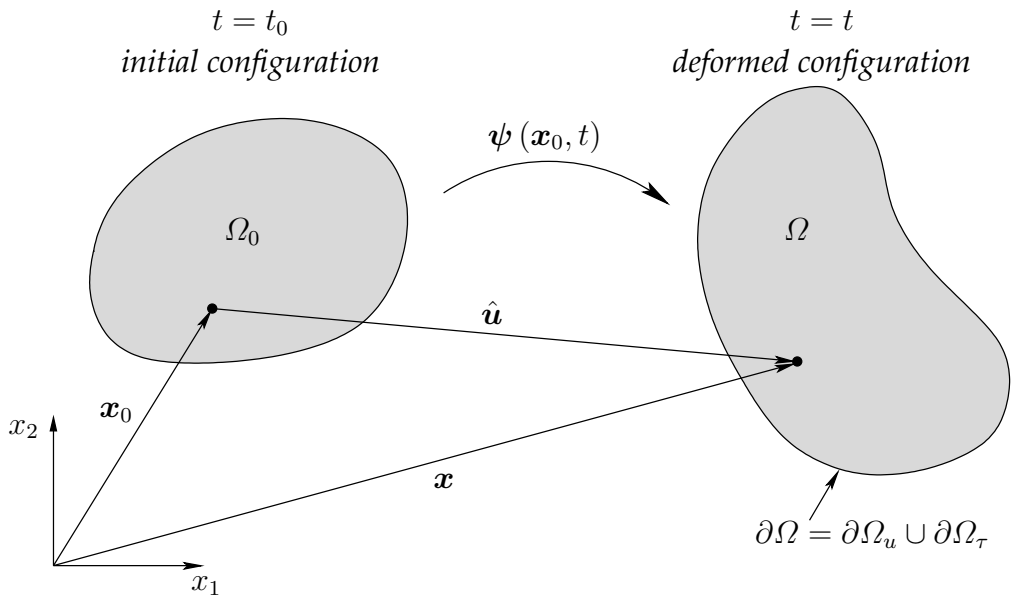


Figure 3.1: Initial and deformed configurations of a continuum

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T \qquad \text{and} \qquad \boldsymbol{x}_0 = \begin{bmatrix} x_{01} & x_{02} & x_{03} \end{bmatrix}^T, \tag{3.1}$$

respectively, with T indicating the transpose. The position $\boldsymbol{x}$ in the current configuration can always be linked to the position $\boldsymbol{x}_0$ in the initial configuration as

$$\boldsymbol{x} = \boldsymbol{\psi}(\boldsymbol{x}_0, t), \tag{3.2}$$

which implies that at $t = t_0$, $\boldsymbol{x} = \boldsymbol{x}_0$. The function $\boldsymbol{\psi}$ is used in the transformation from the initial to the current configuration. This function is invertible, allowing the identification of the initial position $\boldsymbol{x}_0$ of a material point currently located at $\boldsymbol{x}$, i.e.,

$$\boldsymbol{x}_0 = \boldsymbol{\psi}^{-1}(\boldsymbol{x}, t) \tag{3.3}$$

with superscript -1 indicating the inverse.

3.1.1 Motion and kinematics

Let the displacement, velocity and acceleration vectors of a material point be

$$\begin{aligned} \hat{\boldsymbol{u}}(\boldsymbol{x}, t) &= \begin{bmatrix} \hat{u}_1 & \hat{u}_2 & \hat{u}_3 \end{bmatrix}^T, \\ \hat{\boldsymbol{v}}(\boldsymbol{x}, t) &= \begin{bmatrix} \hat{v}_1 & \hat{v}_2 & \hat{v}_3 \end{bmatrix}^T, \\ \hat{\boldsymbol{a}}(\boldsymbol{x}, t) &= \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \hat{a}_3 \end{bmatrix}^T, \end{aligned} \tag{3.4}$$

respectively. The displacement $\hat{\boldsymbol{u}}$ can be defined in terms of the positions of the material point in the initial and current configurations as indicated in Figure 3.1, i.e.,

$$\hat{\boldsymbol{u}}(\boldsymbol{x}, t) = \boldsymbol{x} - \boldsymbol{x}_0 \tag{3.5}$$

and the velocity $\hat{\boldsymbol{v}}$ is defined as the material time derivative of the displacement $\hat{\boldsymbol{u}}$,

$$\hat{\boldsymbol{v}}\left(\boldsymbol{x},t\right)=\frac{d}{dt}\hat{\boldsymbol{u}}\left(\boldsymbol{x},t\right), \tag{3.6}$$

while the acceleration $\hat{\boldsymbol{a}}$ is related to the velocity $\hat{\boldsymbol{v}}$ by

$$\hat{\boldsymbol{a}}\left(\boldsymbol{x},t\right)=\frac{d}{dt}\hat{\boldsymbol{v}}\left(\boldsymbol{x},t\right) \tag{3.7}$$

with

$$\frac{d}{dt}=\frac{\partial}{\partial t}+\hat{v}_i\frac{\partial}{\partial x_i} \tag{3.8}$$

being the *material time derivative* of a quantity. We should emphasize here that the second term of the right-hand side of Equation 3.8, which represents the convective term, vanishes in Lagrangian description and hence the material time derivative is simply expressed by the partial derivative with respect to time [60]. Hence, the following condition applies throughout this thesis

$$\frac{d}{dt}=\frac{\partial}{\partial t}. \tag{3.9}$$

The strain rate tensor $\dot{\varepsilon}_{ij}$, which is the symmetric part of the velocity gradient tensor is defined as

$$\dot{\varepsilon}_{ij}=\frac{d\varepsilon_{ij}}{dt}=\frac{1}{2}\left(\frac{\partial\hat{v}_i}{\partial x_j}+\frac{\partial\hat{v}_j}{\partial x_i}\right). \tag{3.10}$$

The components of ε_{ij} are

$$\varepsilon_{ij}=\begin{pmatrix}\varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13}\\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23}\\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33}\end{pmatrix}. \tag{3.11}$$

3.1.2 Constitutive relation

Having the strain tensor and its rate defined, a relation that relates the rate of stress to the rate of strain and the state of the material is required. This is what is so-called *constitutive relation*. Let us denote Cauchy stress tensor as σ_{ij}, its rate as $\dot{\sigma}_{ij}$ and Jaumann rate of stress as $\overset{\nabla\mathrm{J}}{\sigma}_{ij}$. Following Prager [139], these tensors are related as

$$\dot{\sigma}_{ij}=\overset{\nabla\mathrm{J}}{\sigma}_{ij}+\omega_{ik}\sigma_{kj}-\sigma_{ik}\omega_{kj}, \tag{3.12}$$

in which ω_{ij} denotes the spin tensor, which is the antisymmetric part of the velocity gradient tensor and is written as

$$\omega_{ij}=\frac{1}{2}\left(\frac{\partial\hat{v}_i}{\partial x_j}-\frac{\partial\hat{v}_j}{\partial x_i}\right) \tag{3.13}$$

and σ_{ij} has the following components

$$\sigma_{ij} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}. \tag{3.14}$$

The Jaumann rate of stress can now be defined in terms of the co-rotational rate of Kirchhoff stress $\overset{\nabla \mathrm{H}}{\sigma}_{ij}$, also referred to as Hill stress rate [91], i.e.,

$$\overset{\nabla \mathrm{J}}{\sigma}_{ij} = \overset{\nabla \mathrm{H}}{\sigma}_{ij} - \dot{\varepsilon}_{kk}\sigma_{ij}, \tag{3.15}$$

where $\dot{\varepsilon}_{kk}$ is the volumetric strain rate, which is defined as

$$\dot{\varepsilon}_{kk} = \frac{\partial \hat{v}_k}{\partial x_k}. \tag{3.16}$$

Substituting Equation 3.15 into Equation 3.12 yields

$$\dot{\sigma}_{ij} = \overset{\nabla \mathrm{H}}{\sigma}_{ij} - \dot{\varepsilon}_{kk}\sigma_{ij} + \omega_{ik}\sigma_{kj} - \sigma_{ik}\omega_{kj}. \tag{3.17}$$

An incrementally linear constitutive relation can be written as

$$\overset{\nabla \mathrm{H}}{\sigma}_{ij} = D_{ijkl}\dot{\varepsilon}_{kl}, \tag{3.18}$$

where D_{ijkl} denotes the constitutive tensor. In the special case of an isotropic linear elastic material Hooke's law applies and

$$D_{ijkl} = (K - \frac{2}{3}G)\delta_{ij}\delta_{kl} + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{3.19}$$

with δ_{ij} taking the usual definition of Kronecker delta, K and G being the bulk modulus and the shear modulus of the material, respectively. They are related to Young's modulus E and Poisson's ratio ν by

$$K = \frac{E}{3(1-2\nu)} \qquad \text{and} \qquad G = \frac{E}{2(1+\nu)}. \tag{3.20}$$

For small deformations, Equation 3.19 corresponds to (isotropic) elasticity, but more general constitutive models including plasticity are to be considered elsewhere in this thesis. We emphasize that throughout this thesis, tensile stresses are considered to be positive while compressive stresses are negative.

3.1.3 Conservation laws

We consider now the fundamental conservation laws of thermodynamics, which a continuum should satisfy. These laws are *conservation of mass*, *conservation of momentum* and *conservation of energy*.

Conservation of mass When *sources* and *sinks* are neglected, i.e., no mass entering or leaving the domain Ω occupied by the material, the change of mass with time should be zero. The mathematical form of the *conservation of mass* can be written as

$$\frac{d\varrho}{dt} + \varrho\frac{\partial \hat{v}_i}{\partial x_i} = 0 \tag{3.21}$$

with ϱ being the mass density of the material. Equation 3.21 is also referred to as the *continuity equation*.

Conservation of momentum Conservation of momentum implies both, conservation of linear and angular momentums. The conservation of linear momentum represents the equation of motion of a continuum, i.e., Newton's second law of motion. It relates the motion or the kinetic of a continuum to the internal and external forces acting upon it. Hence, it is a key equation in the formulation and consequently the discretization of the continuum. Without going into details of development and derivation of the equation, we give the mathematical form of the *conservation of linear momentum* as

$$\varrho\frac{d\hat{v}_i}{dt} = \frac{\partial \sigma_{ij}}{\partial x_j} + \varrho g_i, \tag{3.22}$$

in which the term ϱg_i represents the forces due to gravity or the self weight of the continuum, which are the only body forces considered in this thesis and g_i stands for the components of gravitational acceleration, i.e.,

$$g_i = \begin{bmatrix} g_1 & g_2 & g_3 \end{bmatrix}^T. \tag{3.23}$$

The value of the gravitational acceleration is assumed to be $10\,[m/s^2]$ unless stated otherwise. The *conservation of angular momentum* implies that the stress tensor is symmetric, i.e.,

$$\sigma_{ij} = \sigma_{ji}. \tag{3.24}$$

As the conservation of angular momentum adds no new equation, most literature calls the conservation of linear momentum as the *momentum equation*.

Conservation of energy As we disregard heat effects and any source of thermal energy in this thesis and consider the mechanical work as the only source of energy, the *conservation of energy* takes the form

$$\varrho\frac{dr}{dt} = \dot{\varepsilon}_{ij}\sigma_{ij} \tag{3.25}$$

with r being the internal energy per unit mass.

3.1.4 Boundary and initial conditions

Let $\partial\Omega$ represents the boundary of the domain Ω, which consists of two parts, namely $\partial\Omega_u$ and $\partial\Omega_\tau$. The boundary $\partial\Omega_u$ denotes the *prescribed displacement boundary*, whereas

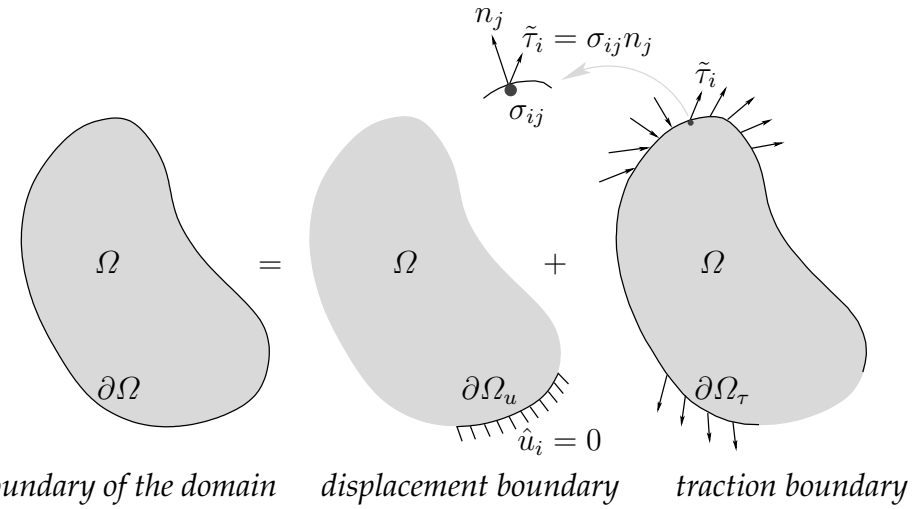


Figure 3.2: Displacement and traction boundary conditions

$\partial\Omega_\tau$ denotes the *prescribed stress boundary* or *prescribed traction boundary* as indicated in Figure 3.2. The displacement boundary conditions can be written as

$$\hat{u}_i(\boldsymbol{x}, t) = \hat{U}_i(t) \qquad \text{on} \quad \partial\Omega_u(t) \tag{3.26}$$

and they are also referred to as *essential boundary conditions* or *Dirichlet boundary conditions*. The surface traction vector $\tau_i(\boldsymbol{x}, t)$ can be written as a multiplication of a space vector of traction $\tilde{\tau}_i(\boldsymbol{x})$ and a time function $\mathcal{T}(t)$, i.e.,

$$\tau_i(\boldsymbol{x}, t) = \tilde{\tau}_i(\boldsymbol{x})\, \mathcal{T}(t). \tag{3.27}$$

Hence, the traction boundary conditions are defined as

$$\sigma_{ij}(\boldsymbol{x}, t)\, n_j = \tilde{\tau}_i\, \mathcal{T}(t) \qquad \text{on} \quad \partial\Omega_\tau(t) \tag{3.28}$$

with n_j indicating the unit vector normal to the boundary $\partial\Omega$ and pointing outward. The vector $\tilde{\tau}_i$ represents the prescribed traction at a boundary material point located at $\boldsymbol{x}$. The traction boundary conditions are also called *natural boundary conditions* or *Neumann boundary conditions* with Equation 3.28 being the *Cauchy's formula*. At each material point located at the boundary $\partial\Omega$, either displacement or traction must be prescribed, but not both of them, i.e.,

$$\partial\Omega_u \cap \partial\Omega_\tau = \varnothing \quad \text{and} \quad \partial\Omega_u \cup \partial\Omega_\tau = \partial\Omega. \tag{3.29}$$

It should be realized that the above conditions should be satisfied for each of the three directions. The *initial conditions* are considered as

$$\hat{u}_i(\boldsymbol{x}, t_0) = \hat{U}_{0i}, \quad \hat{v}_i(\boldsymbol{x}, t_0) = \hat{V}_{0i} \quad \text{and} \quad \sigma_{ij}(\boldsymbol{x}, t_0) = \sigma_{0ij}. \tag{3.30}$$

More details about the development and derivation of the basic equations of solid continuum can be found in books dealing with continuum mechanics, e.g., Malvern [119].

3.2 Weak form of momentum and traction

As we mentioned previously, the conservation of the momentum (Equation 3.22) is a key equation in continuum mechanics, representing the equation of motion of the continuum. To discretize this equation, its *strong form* has to be transformed into the *weak form* or the well-known *virtual work equation*. The momentum equation is now multiplied by a *test function* or a *virtual velocity* $\delta\hat{v}_i$ and is integrated over the current domain Ω occupied by the continuum, i.e.,

$$\int_\Omega \delta\hat{v}_i \varrho \frac{d\hat{v}_i}{dt}\, d\Omega = \int_\Omega \delta\hat{v}_i \frac{\partial \sigma_{ij}}{\partial x_j}\, d\Omega + \int_\Omega \delta\hat{v}_i \varrho g_i\, d\Omega \tag{3.31}$$

with

$$\delta\hat{v}_i = 0 \ \text{ on } \ \partial\Omega_u. \tag{3.32}$$

The virtual velocity $\delta\hat{v}_i$ must be kinematically admissible, that is, it satisfies the essential boundary conditions and is continuous over the domain. The first term on the right-hand side of Equation 3.31 can be expressed as

$$\int_\Omega \delta\hat{v}_i \frac{\partial \sigma_{ij}}{\partial x_j}\, d\Omega = \int_\Omega \frac{\partial}{\partial x_j}\left(\delta\hat{v}_i \sigma_{ij}\right)\, d\Omega - \int_\Omega \frac{\partial(\delta\hat{v}_i)}{\partial x_j}\sigma_{ij}\, d\Omega. \tag{3.33}$$

Substituting Equation 3.33 into Equation 3.31 yields

$$\int_\Omega \delta\hat{v}_i \varrho \frac{d\hat{v}_i}{dt}\, d\Omega = \int_\Omega \frac{\partial}{\partial x_j}\left(\delta\hat{v}_i \sigma_{ij}\right)\, d\Omega - \int_\Omega \frac{\partial(\delta\hat{v}_i)}{\partial x_j}\sigma_{ij}\, d\Omega + \int_\Omega \delta\hat{v}_i \varrho g_i\, d\Omega. \tag{3.34}$$

Using *Gauss' theorem*, also called as *divergence theorem*, the first term on the right-hand side of Equation 3.34 can be written as

$$\int_\Omega \frac{\partial}{\partial x_j}\left(\delta\hat{v}_i \sigma_{ij}\right)\, d\Omega = \int_{\partial\Omega} \delta\hat{v}_i n_j \sigma_{ij}\, dS \tag{3.35}$$

with dS denoting an integral over a surface. One must keep in mind that the only integral which does not vanish in the weak form is the integral over the traction boundary $\partial\Omega_\tau$. The integral over the displacement boundary $\partial\Omega_u$ vanishes in the weak form, see Equation 3.32. This fact together with Cauchy's formula (Equation 3.28) lead to

$$\int_\Omega \frac{\partial}{\partial x_j}\left(\delta\hat{v}_i \sigma_{ij}\right)\, d\Omega = \int_{\partial\Omega_\tau} \delta\hat{v}_i \tau_i\, dS, \tag{3.36}$$

which after substitution into Equation 3.34 yields the virtual work equation, that represents the weak form of the momentum equation, i.e.,

$$\int_\Omega \delta\hat{v}_i \varrho \frac{d\hat{v}_i}{dt}\, d\Omega = \int_{\partial\Omega_\tau} \delta\hat{v}_i \tau_i\, dS + \int_\Omega \delta\hat{v}_i \varrho g_i\, d\Omega - \int_\Omega \frac{\partial(\delta\hat{v}_i)}{\partial x_j}\sigma_{ij}\, d\Omega, \tag{3.37}$$

which will be used then in the formulation of the finite element discrete equations.

3.3 Space discretization of the virtual work

In finite element method, the domain Ω is decomposed into finite subdomains Ω_e called *finite elements*, see Figure 3.3. The union of these subdomains comprises the total domain, $\Omega = \cup_{e=1}^{n_{elm}} \Omega_e$, where n_{elm} denotes the total number of finite elements in the mesh. Each element is jointed with its surrounding elements by number of points called *nodes*. The state variable is assumed to have pre-defined interpolation functions within the element and the solution is then obtained at the nodes. Hence, the equilibrium is satisfied at the nodes. It is usual in the finite element method to use matrix notations in the discretization of the virtual work equation, see for instance, reference [150]. Throughout this thesis, bold symbols are used to denote matrices and vectors. Taking advantage of the symmetry of the strain tensor, the strain and its rate can be represented in a vector form as

$$\dot{\boldsymbol{\varepsilon}}(\boldsymbol{x}, t) = \frac{d}{dt}\boldsymbol{\varepsilon}(\boldsymbol{x}, t), \tag{3.38}$$

where

$$\boldsymbol{\varepsilon}(\boldsymbol{x}, t) = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{22} & \varepsilon_{33} & \gamma_{12} & \gamma_{23} & \gamma_{31} \end{bmatrix}^T \tag{3.39}$$

with ε_{ij}, $i = j$ being the normal strain in the direction of x_i coordinate and $\gamma_{ij} = 2\varepsilon_{ij}$, $i \neq j$ being the shear strain in $x_i x_j$ plane. Similarly, the stress rate vector is

$$\dot{\boldsymbol{\sigma}}(\boldsymbol{x}, t) = \frac{d}{dt}\boldsymbol{\sigma}(\boldsymbol{x}, t) \tag{3.40}$$

with

$$\boldsymbol{\sigma}(\boldsymbol{x}, t) = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & \sigma_{23} & \sigma_{31} \end{bmatrix}^T, \tag{3.41}$$

where σ_{ij}, $i = j$ being the normal stress in the direction of x_i coordinate and σ_{ij}, $i \neq j$ being the shear stress in $x_i x_j$ plane. The above representation of stress and strain is known in the literature as *Voigt notation* [21]. Such representation has storage and computational

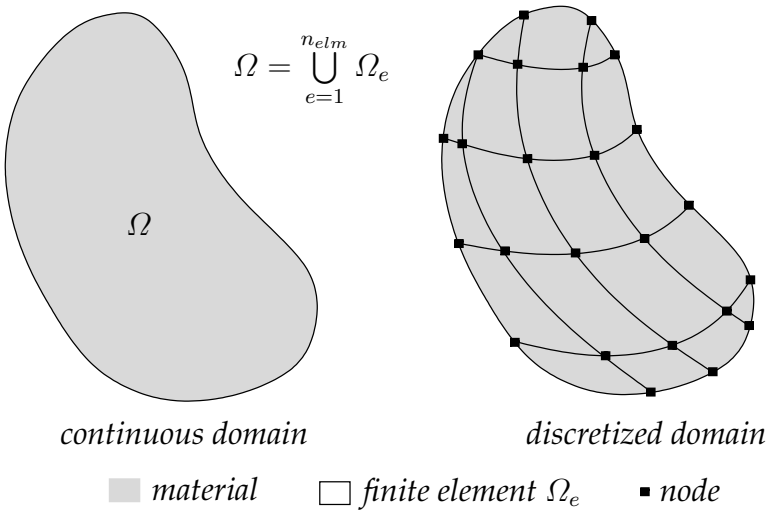


Figure 3.3: Finite element discretization of a domain

advantage in numerical calculations. However, one should realize that the representation of the strain is different from stress in Voigt notation. The last three terms of the strain vector are represented by $\gamma_{ij} = 2\varepsilon_{ij}$. The reason for this is to ensure that energy is preserved and that different expressions of energy using tensors or vectors are equal, i.e.,

$$\varepsilon_{ij}\sigma_{ij} = \boldsymbol{\varepsilon}^T\boldsymbol{\sigma}. \tag{3.42}$$

In finite element method, the discrete form is obtained by approximating the displacement and eventually the velocity and acceleration as

$$\begin{aligned}
\hat{\boldsymbol{u}}(\boldsymbol{x},t) &\approx \bar{\boldsymbol{N}}(\boldsymbol{x})\,\boldsymbol{u}(t),\\
\hat{\boldsymbol{v}}(\boldsymbol{x},t) &\approx \bar{\boldsymbol{N}}(\boldsymbol{x})\,\boldsymbol{v}(t),\\
\hat{\boldsymbol{a}}(\boldsymbol{x},t) &\approx \bar{\boldsymbol{N}}(\boldsymbol{x})\,\boldsymbol{a}(t),
\end{aligned} \tag{3.43}$$

respectively. The corresponding virtual quantities are approximated the same way, for instance, $\delta\hat{\boldsymbol{v}} \approx \bar{\boldsymbol{N}}\delta\boldsymbol{v}$. The *interpolation function* or the *shape function* matrix $\bar{\boldsymbol{N}}$ has the following form

$$\bar{\boldsymbol{N}}(\boldsymbol{x}) = \begin{bmatrix}\bar{\boldsymbol{N}}_1(\boldsymbol{x}) & \bar{\boldsymbol{N}}_2(\boldsymbol{x}) & \bar{\boldsymbol{N}}_3(\boldsymbol{x}) & \cdots & \bar{\boldsymbol{N}}_{n_T}(\boldsymbol{x})\end{bmatrix} \tag{3.44}$$

with

$$\bar{\boldsymbol{N}}_i(\boldsymbol{x}) = \begin{bmatrix}\bar{N}_i(\boldsymbol{x}) & 0 & 0\\ 0 & \bar{N}_i(\boldsymbol{x}) & 0\\ 0 & 0 & \bar{N}_i(\boldsymbol{x})\end{bmatrix} \tag{3.45}$$

and n_T denoting the total number of nodes in the mesh. The bar superscript implies that the shape functions are written in terms of the global coordinate system $\boldsymbol{x}$. The vectors of nodal displacements, velocities and accelerations are denoted as $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{a}$, respectively. Using the notation of reference [27], these vectors can be written as

$$\begin{aligned}
\boldsymbol{u}(t) &= \begin{bmatrix}u_{11} & u_{12} & u_{13} & \dots & u_{n_T1} & u_{n_T2} & u_{n_T3}\end{bmatrix}^T,\\
\boldsymbol{v}(t) &= \begin{bmatrix}v_{11} & v_{12} & v_{13} & \dots & v_{n_T1} & v_{n_T2} & v_{n_T3}\end{bmatrix}^T,\\
\boldsymbol{a}(t) &= \begin{bmatrix}a_{11} & a_{12} & a_{13} & \dots & a_{n_T1} & a_{n_T2} & a_{n_T3}\end{bmatrix}^T
\end{aligned} \tag{3.46}$$

with, for instance, v_{12} indicating the velocity at node 1 in the direction of coordinate x_2. The kinematic relation can be written using matrix notations as

$$\dot{\boldsymbol{\varepsilon}}(\boldsymbol{x},t) = \boldsymbol{L}\hat{\boldsymbol{v}}(\boldsymbol{x},t) \tag{3.47}$$

with $\boldsymbol{L}$ being a linear differential operator, that has the following form

$$\boldsymbol{L} = \begin{bmatrix} \dfrac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \dfrac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \dfrac{\partial}{\partial x_3} \\ \dfrac{\partial}{\partial x_2} & \dfrac{\partial}{\partial x_1} & 0 \\ 0 & \dfrac{\partial}{\partial x_3} & \dfrac{\partial}{\partial x_2} \\ \dfrac{\partial}{\partial x_3} & 0 & \dfrac{\partial}{\partial x_1} \end{bmatrix}. \tag{3.48}$$

Substituting the second of Equation 3.43 into Equation 3.47 yields

$$\dot{\boldsymbol{\varepsilon}}\left(\boldsymbol{x}, t\right) = \boldsymbol{B}\left(\boldsymbol{x}\right)\boldsymbol{v}\left(t\right), \tag{3.49}$$

in which $\boldsymbol{B}$ is the *strain-displacement matrix* that can be written as

$$\boldsymbol{B}\left(\boldsymbol{x}\right) = \begin{bmatrix} \boldsymbol{B}_1\left(\boldsymbol{x}\right) & \boldsymbol{B}_2\left(\boldsymbol{x}\right) & \boldsymbol{B}_3\left(\boldsymbol{x}\right) & \ldots & \boldsymbol{B}_{n_T}\left(\boldsymbol{x}\right) \end{bmatrix} \tag{3.50}$$

with

$$\boldsymbol{B}_i\left(\boldsymbol{x}\right) = \boldsymbol{L}\bar{\boldsymbol{N}}_i\left(\boldsymbol{x}\right). \tag{3.51}$$

Substituting Equation 3.45 into Equation 3.51 yields

$$\boldsymbol{B}_i\left(\boldsymbol{x}\right) = \begin{bmatrix} \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_1} & 0 & 0 \\ 0 & \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_2} & 0 \\ 0 & 0 & \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_3} \\ \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_2} & \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_1} & 0 \\ 0 & \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_3} & \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_2} \\ \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_3} & 0 & \dfrac{\partial \bar{N}_i\left(\boldsymbol{x}\right)}{\partial x_1} \end{bmatrix}. \tag{3.52}$$

Now the virtual work equation (Equation 3.37) can be written in matrix form as

$$\delta\boldsymbol{v}^T \int_\Omega \bar{\boldsymbol{N}}^T \varrho\, \bar{\boldsymbol{N}}\boldsymbol{a}\, d\Omega = \delta\boldsymbol{v}^T \int_{\partial\Omega_\tau} \bar{\boldsymbol{N}}^T \boldsymbol{\tau}\, dS + \delta\boldsymbol{v}^T \int_\Omega \bar{\boldsymbol{N}}^T \varrho\, \boldsymbol{g}\, d\Omega - \delta\boldsymbol{v}^T \int_\Omega \boldsymbol{B}^T \boldsymbol{\sigma}\, d\Omega, \tag{3.53}$$

where $\boldsymbol{\tau}$ and $\boldsymbol{g}$ are vectors containing the components of tractions and gravitational acceleration, respectively. The vector $\delta\boldsymbol{v}$ contains the virtual nodal velocities that are arbitrary except where the velocities are prescribed. Hence, Equation 3.53 can be written as

$$\int_{\Omega} \bar{\boldsymbol{N}}^T \varrho\, \bar{\boldsymbol{N}} \boldsymbol{a}\, d\Omega = \int_{\partial\Omega_\tau} \bar{\boldsymbol{N}}^T \boldsymbol{\tau}\, dS + \int_{\Omega} \bar{\boldsymbol{N}}^T \varrho\, \boldsymbol{g}\, d\Omega - \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma}\, d\Omega. \tag{3.54}$$

It is of interest to emphasize here that in the computer implementation of the finite element method, the integrals in Equation 3.54 are performed on each finite element when sweeping over the elements in the mesh. Element matrices resulting from the integration are gathered in global matrices through the *assemblage* procedure, which is explained in Appendix D. Equation 3.54 can now take the following form

$$\begin{aligned}\left(\mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \bar{\boldsymbol{N}}^T \varrho_e\, \bar{\boldsymbol{N}}\, d\Omega\right) \boldsymbol{a} = & \mathop{\mathrm{A}}_{e=1}^{n_{\tau elm}} \int_{\partial\Omega_{\tau e}} \bar{\boldsymbol{N}}^T \boldsymbol{\tau}_e\, dS \\ & + \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \bar{\boldsymbol{N}}^T \varrho_e\, \boldsymbol{g}\, d\Omega - \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{\sigma}_e\, d\Omega\end{aligned} \tag{3.55}$$

with A indicating the assemblage. The first term on the right-hand side of the above equation is the only term that consists of integrations over the surface and hence the assemblage of this term includes only surface elements where the traction is prescribed. The number of these elements is denoted as $n_{\tau elm}$. The matrices $\bar{\boldsymbol{N}}$ and $\boldsymbol{B}$ in Equation 3.55 take the forms that are defined in Equations 3.44 and 3.50, respectively. The subscript n_T is replaced by n_{en}, which denotes the number of nodes in each finite element.

3.3.1 Numerical space integration

Closed-form integrations in Equation 3.55 are difficult, or even impossible in some cases. Instead, *numerical integration* is adopted to evaluate these integrals. We consider, for instance, the integral in the last term of Equation 3.55 as an example to explain the numerical quadrature. On denoting the number of *quadrature points* in element e as n_{eq}, the numerical integration of this term in element e is evaluated as

$$\int_{\Omega_e} \boldsymbol{B}^T(\boldsymbol{x})\, \boldsymbol{\sigma}_e(\boldsymbol{x}, t)\; d\Omega \approx \sum_{q=1}^{n_{eq}} W_q\, \boldsymbol{B}^T(\boldsymbol{x}_q)\, \boldsymbol{\sigma}_e(\boldsymbol{x}_q, t) \tag{3.56}$$

with W_q being the global integration weight associated with the quadrature point q. It represents a portion of the element volume, i.e., $\sum_{q=1}^{n_{eq}} W_q = \Omega_e$. After applying the numerical quadrature on the terms of Equations 3.55, it takes the form

$$\begin{aligned}\boldsymbol{M}^c \boldsymbol{a} &= \boldsymbol{F}^{trac} + \boldsymbol{F}^{grav} - \boldsymbol{F}^{int} \\ &= \boldsymbol{F}^{ext} - \boldsymbol{F}^{int},\end{aligned} \tag{3.57}$$

where $\boldsymbol{M}^c$ denotes the consistent-mass matrix, $\boldsymbol{F}^{ext}$ is the vector of external forces, which consists of both the forces due to surface tractions $\boldsymbol{F}^{trac}$ and due to gravity $\boldsymbol{F}^{grav}$, while the vector $\boldsymbol{F}^{int}$ denotes the internal forces due to stresses. The terms in Equation 3.57 correspond to those of Equation 3.55.

It is usual to perform the numerical integration in finite element method in the domain of the *parent element*. Hence, the procedure requires the shape functions to be written in terms of parent element coordinate system, i.e., *natural coordinates* or *local coordinates*. The geometry of the element in the current domain is always related to the geometry of the parent element via interpolation functions. This thesis is restricted to the use of *isoparametric element* and hence the same functions that are used to approximate the state variables are used to relate the current and the parent domains. For more details the reader is referred to Appendix C.

3.3.2 Lumped-mass matrix

We already pointed out in Chapter 2 that the use of a *consistent-mass* matrix adds considerable complexities to the algorithm. The main reason for such complexities is that the solution of large system of algebraic equations can consume considerable computational time. Hence, the consistent-mass matrix is not widely used in practice. The practical alternative is the *lumped-mass* matrix, which has computational and storage advantages over the consistent-mass matrix. However, the lumped-mass matrix is associated with some numerical dissipation of the kinetic energy [41].

There are several methods to construct the lumped-mass matrix, but they are *"ad hoc"* as described by Belytschko et al. [21] as there is little mathematical theory supporting these methods. The reader is referred to the book by Hughes [97] for details about different methods of constructing the lumped-mass matrix. The most common method however is the one in which, the lumped-mass matrix is constructed from the consistent one so that the entries of the lumped matrix correspond to the row sum of the consistent matrix. Let $\boldsymbol{M}^L$ denotes the global lumped-mass matrix and $\boldsymbol{M}_e^L$ is the element mass matrix. The matrix $\boldsymbol{M}^L$ is then assembled as

$$\boldsymbol{M}^L = \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \boldsymbol{M}_e^L \tag{3.58}$$

with

$$\boldsymbol{M}_e^L = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{m}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{m}_{en} \end{bmatrix}. \tag{3.59}$$

A sub-matrix $\boldsymbol{m}_i$ corresponds to the mass of node i, and the matrix $\boldsymbol{0}$ is a null matrix. For three-dimensional finite element, they have the following forms

$$\boldsymbol{m}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \quad \text{and} \quad \boldsymbol{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.60}$$

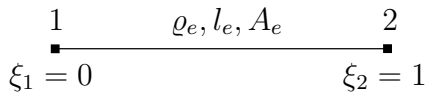


Figure 3.4: Two-noded rod element

with

$$m_i \approx \sum_{q=1}^{n_{eq}} w_q \, N_i \left(\boldsymbol{\xi}_q \right) \varrho_e \left(\boldsymbol{x} \left(\boldsymbol{\xi}_q \right) \right) \left| \boldsymbol{J} \left(\boldsymbol{\xi}_q \right) \right| , \tag{3.61}$$

in which w_q is the local integration weight of the integration point q, N_i is the shape function of node i written in terms of the local coordinates system $\boldsymbol{\xi}$, and $\boldsymbol{J}$ is the Jacobian matrix, see Appendix B. With a close scrutiny of the above procedure of constructing the lumped-mass matrix, one should realize that we never explicitly construct it from the row sum of the consistent one, but we rather use a *"direct lumping"* procedure as described above.

The direct lumping procedure gives identical lumped-mass matrix as the one constructed by the row sum of the consistent matrix and it requires much less computational time and is easy to implement. However, when considering Lagrangian FEM, this is normally a minor priority as the mass matrix is constructed only once throughout the calculations, but when considering MPM, this procedure is very efficient because the mass matrix is constructed at each time step as particles move through the mesh and hence the total number of degrees-of-freedom changes during the computations. The above procedure can be proven to be identical to the row sum of the consistent-mass matrix. To do so, we consider the 2-noded rod element of Figure 3.4. The two shape functions of such element are

$$\begin{aligned} N_1 \left(\xi \right) &= 1 - \xi, \\ N_2 \left(\xi \right) &= \xi \end{aligned} \tag{3.62}$$

with $\xi \in [0, 1]$ being the local coordinate of the corresponding parent element. The shape function matrix of this element is written as

$$\boldsymbol{N} \left(\xi \right) = \begin{bmatrix} N_1 \left(\xi \right) & N_2 \left(\xi \right) \end{bmatrix} . \tag{3.63}$$

The integration of the consistent-mass matrix of element e can be written as

$$\boldsymbol{M}_e^c = \int_{\Omega_e} \boldsymbol{N}^T \varrho_e \, \boldsymbol{N} \, d\Omega. \tag{3.64}$$

On considering a single Gauss point q located at ξ_q, the numerical integration of the above equation gives

$$\boldsymbol{M}_e^c \approx \varrho_e \, l_e \, A_e \begin{bmatrix} N_1^2 \left(\xi_q \right) & N_1 \left(\xi_q \right) N_2 \left(\xi_q \right) \\ N_1 \left(\xi_q \right) N_2 \left(\xi_q \right) & N_2^2 \left(\xi_q \right) \end{bmatrix} \tag{3.65}$$

with l_e and A_e being the length and the cross-sectional area of element e, respectively. Let us now sum the rows of the matrix $\boldsymbol{M}_e^c$

$$\begin{aligned}\sum row_1 &= N_1^2(\xi_q) + N_1(\xi_q) N_2(\xi_q) \\ &= (1-\xi_q)^2 + (1-\xi_q)\xi_q \\ &= 1-\xi_q \\ &= N_1(\xi_q)\end{aligned}$$

and

$$\begin{aligned}\sum row_2 &= N_1(\xi_q) N_2(\xi_q) + N_2^2(\xi_q) \\ &= (1-\xi_q)\xi_q + \xi_q^2 \\ &= \xi_q \\ &= N_2(\xi_q).\end{aligned}$$

Hence, the lumped-mass matrix which results from the row sum of the consistent one is

$$\boldsymbol{M}_e^L \approx \varrho_e\, l_e\, A_e \begin{bmatrix} N_1(\xi_q) & 0 \\ 0 & N_2(\xi_q) \end{bmatrix}, \tag{3.66}$$

which confirms the *"direct lumping"* procedure explained in Equations 3.58-3.61. Having the lumped-mass matrix defined, Equation 3.57 can now be written as

$$\begin{aligned}\boldsymbol{M}\boldsymbol{a} &= \boldsymbol{F}^{trac} + \boldsymbol{F}^{grav} - \boldsymbol{F}^{int} \\ &= \boldsymbol{F}^{ext} - \boldsymbol{F}^{int}.\end{aligned} \tag{3.67}$$

The superscript L is dropped from $\boldsymbol{M}^L$ to simplify the presentation of the equations. Hence, $\boldsymbol{M}$ is used to denote the lumped-mass matrix.

3.4 Time discretization

The system in Equation 3.67 is discretized in space, but still continuous in time. It represents a series of second-order ordinary differential equations (ODE) in time. To get a full discrete system of algebraic equations, Equation 3.67 has to be discretized in time. A common procedure to do that is by replacing the differentials in the ODE by finite difference quotients on a discretized time domain, which is the independent variable in our ODE. The time domain is discretized using sampling points and the solution is then advanced in steps starting from the initial time until the desired final time is reached. The discretization of the time domain is shown in Figure 3.5, where uniformly spaced sample points of a time step size Δt are considered. The solution at a sample point $(t + \Delta t)$ can be approximated using *one-step* schemes, in which the solution at $(t + \Delta t)$ depends only on the state at point t and hence such schemes are also referred to as *two-point* schemes. *Two-step* schemes can also be used to approximate the solution at the

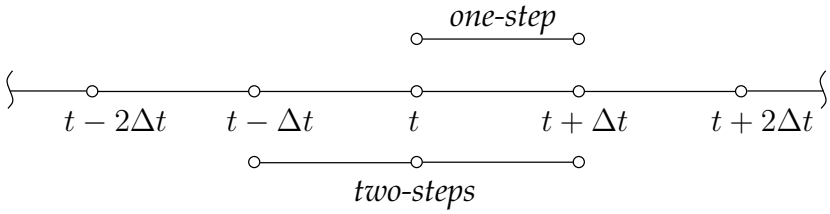


Figure 3.5: One and two-steps approximation of ODE

sample point $(t + \Delta t)$. In such schemes, the solution at $(t + \Delta t)$ depends on the state at t and $(t - \Delta t)$ and hence they are also known as *three-point* schemes.

The integration schemes of ODE and their applications in computational mechanics are discussed in many well-known finite element books (e.g. [16, 97, 203]). The book by Hughes [97] might be the most comprehensive one in this regard. Hughes in his book arrived to the conclusion that the attributes of multi-step schemes can be achieved within one-step schemes and that in one-step schemes, the accelerations, velocities and displacements at the beginning of the time step together with the applied forces are sufficient to advance the solution to the next time step. The one-step schemes require less storage and computational effort when compared to multi-step schemes. Therefore, we focus our attention in this thesis on the one-step schemes and their application to our ODE. The reader is referred to the aforementioned books and the references therein for details about the multi-steps schemes.

For easy understanding of the time integration and the discretization of an ODE, a scalar function f is considered to be integrated in a time interval $[t, t + \Delta t]$. The integration of such function can be numerically approximated as

$$\int_t^{t+\Delta t} f dt \approx \left[\lambda f^{t+\Delta t} + (1-\lambda)\, f^t\right] \Delta t, \tag{3.68}$$

in which the parameter $\lambda \in [0, 1]$ is used to adjust the integration between fully explicit ($\lambda = 0$) and fully implicit ($\lambda = 1$). The numerical integration in Equation 3.68 is illustrated in Figure 3.6. A field variable y is considered to be related to the function f by the

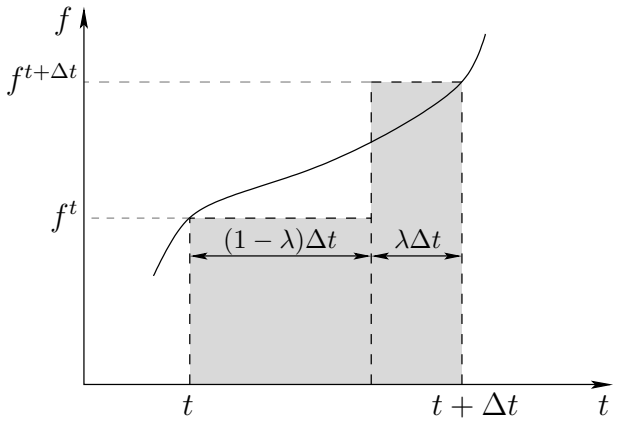


Figure 3.6: Illustration of a one-step integration of a first-order ODE [122]

following first-order ordinary differential equation

$$\dot{y} = \frac{dy}{dt} = f. \tag{3.69}$$

The time integration of this equation in the time interval $[t, t + \Delta t]$ reads

$$\int_t^{t+\Delta t} dy = \int_t^{t+\Delta t} f dt. \tag{3.70}$$

Substituting Equation 3.68 into Equation 3.70 yields

$$y^{t+\Delta t} - y^t \approx \left[\lambda f^{t+\Delta t} + (1-\lambda) f^t\right] \Delta t. \tag{3.71}$$

Depending on the value of the parameter λ, different integration schemes can be defined.

3.4.1 Single degree-of-freedom spring-mass system

To keep matters as simple as possible, the one-step integration scheme of Equation 3.71 is applied to the second-order ODE of the single degree-of-freedom spring-mass system of Figure 3.7. The equation of motion of this system is

$$m\frac{d\hat{v}}{dt} = f = f^{ext} - f^{int} \tag{3.72}$$

with

$$f^{ext} = mg \quad \text{and} \quad f^{int} = k\hat{u}, \tag{3.73}$$

where m and k are the mass and stiffness of the system, respectively and g stands for the gravitational acceleration. This example is restricted to the case of linear spring, i.e., k is constant. Integrating Equation 3.72 in the time interval $[t, t + \Delta t]$ gives

$$\int_t^{t+\Delta t} m d\hat{v} = \int_t^{t+\Delta t} f dt, \tag{3.74}$$

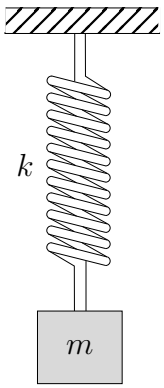


Figure 3.7: Single degree-of-freedom spring-mass system

which after using Equation 3.71 takes the form

$$m\left(\hat{v}^{t+\Delta t}-\hat{v}^{t}\right)\approx\left[\lambda_f f^{t+\Delta t}+\left(1-\lambda_f\right)f^{t}\right]\Delta t. \tag{3.75}$$

The subscript f indicates the integration of the forces. Substituting $f = f^{ext} - f^{int}$ into Equation 3.75 yields

$$m\left(\hat{v}^{t+\Delta t}-\hat{v}^{t}\right)\approx\left[mg-\lambda_f k\hat{u}^{t+\Delta t}-\left(1-\lambda_f\right)k\hat{u}^{t}\right]\Delta t. \tag{3.76}$$

The gravitational acceleration g is assumed to be applied instantaneously and kept constant and hence the external force $f^{ext} = mg$ is a step function in time. The velocity $\hat{v}$ is the time derivative of the displacement $\hat{u}$, i.e., $\hat{v} = \frac{d\hat{u}}{dt}$. Integrating this equation in the time interval $[t, t + \Delta t]$ yields

$$\int_t^{t+\Delta t} d\hat{u} = \int_t^{t+\Delta t} \hat{v}dt, \tag{3.77}$$

which is approximated as

$$\hat{u}^{t+\Delta t}-\hat{u}^{t}\approx\left[\lambda_v\hat{v}^{t+\Delta t}+\left(1-\lambda_v\right)\hat{v}^{t}\right]\Delta t. \tag{3.78}$$

The subscript v indicates the integration of the velocity. Substituting Equation 3.78 into Equation 3.76 for $\hat{u}^{t+\Delta t}$ gives

$$m\left(\hat{v}^{t+\Delta t}-\hat{v}^{t}\right)\approx\Delta tmg-\lambda_f\lambda_v\Delta t^2k\hat{v}^{t+\Delta t}-\Delta tk\hat{u}^{t}-\lambda_f\left(1-\lambda_v\right)\Delta t^2k\hat{v}^{t}, \tag{3.79}$$

which is written for $\hat{v}^{t+\Delta t}$ as

$$\hat{v}^{t+\Delta t}\approx\hat{v}^{t}+\Delta tg-\lambda_f\lambda_v\Delta t^2\frac{k}{m}\hat{v}^{t+\Delta t}-\Delta t\frac{k}{m}\hat{u}^{t}-\lambda_f\left(1-\lambda_v\right)\Delta t^2\frac{k}{m}\hat{v}^{t}. \tag{3.80}$$

Hence, the velocity recursion equation of the single degree-of-freedom spring-mass system can be written for short, with an iteration index i, as

$$\hat{v}_{i+1}^{t+\Delta t}\approx A^{t}+B\hat{v}_{i}^{t+\Delta t}\qquad i=1,i_{max} \tag{3.81}$$

with

$$A^{t}=\hat{v}^{t}\left[1-\lambda_f\left(1-\lambda_v\right)\Delta t^2\omega_n^2\right]+\Delta t\left[g-\omega_n^2\hat{u}^{t}\right]\quad\text{and}\quad B=-\lambda_f\lambda_v\Delta t^2\omega_n^2, \tag{3.82}$$

where $\omega_n = \sqrt{k/m}$ is the angular natural frequency of the system, which will be shortly denoted as natural frequency. To solve for the velocity at the end of the time step, Equation 3.81 is solved iteratively. However, for $\lambda_f = 0$ or $\lambda_v = 0$, the solution is straight forward and requires no iteration. The iteration process starts with a predicted velocity $\hat{v}_i^{t+\Delta t}$ and then a couple of correction steps follow to correct the predicted value and bring it to the converged solution. Hence, the considered scheme is a predictor-corrector procedure. The solution is assumed to be converged and the iteration process is terminated

when the relative error R_{error} between $\hat{v}_{i+1}^{t+\Delta t}$ and $\hat{v}_{i}^{t+\Delta t}$ reached a pre-defined tolerance tol, i.e.,

$$R_{error} = \frac{\left|\hat{v}_{i+1}^{t+\Delta t} - \hat{v}_{i}^{t+\Delta t}\right|}{\left|\hat{v}_{i+1}^{t+\Delta t}\right|} \leqslant tol. \tag{3.83}$$

The coefficient B in Equation 3.81 is constant as long as Δt is constant throughout the calculations and that the system is linear, i.e., the stiffness k has no dependency on the displacement $\hat{u}$, which is the case in our example. On the other hand, it is obvious that the coefficient A changes from time step to another as, among other parameters, it depends on the displacement and velocity at the beginning of the time step, i.e., $\hat{u}^t$ and $\hat{v}^t$, respectively. However, A is constant during the iteration process of the time step.

3.4.1.1 Effect of frequency content on the iteration procedure

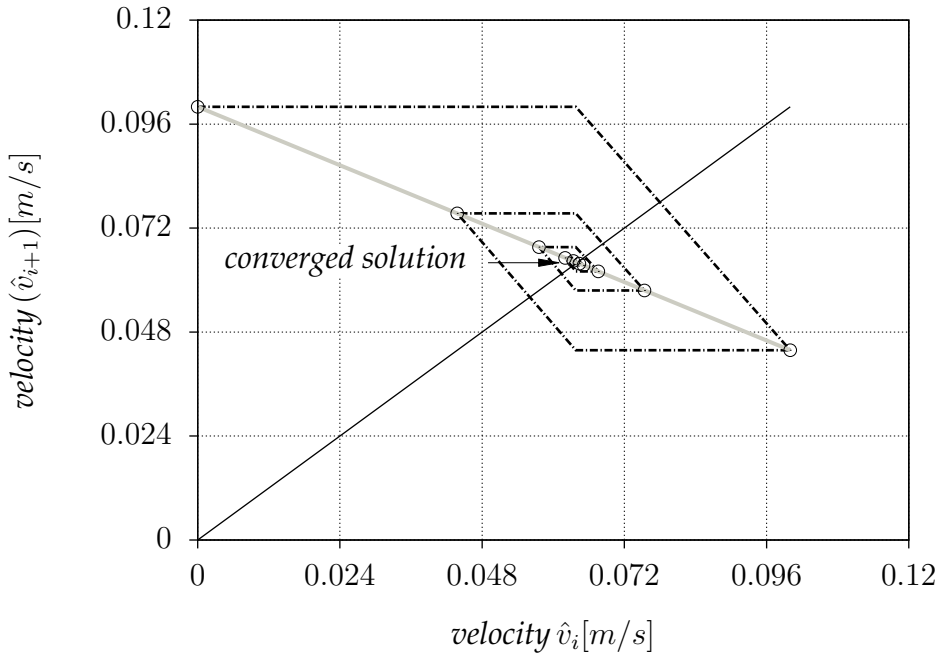
For easy understanding of the effect of the frequency content of the dynamic system on the iteration process, two systems with different frequency contents are considered for comparison. The first system is denoted with the subscript 1 and the second with 2. The first is assumed to have higher frequency than the second, i.e., $\omega_{n1} > \omega_{n2}$. To only include the effect of the frequency in the comparison of coefficient B, we assume that $\lambda_{f1} = \lambda_{f2}$, $\lambda_{v1} = \lambda_{v2}$ and $\Delta t_1 = \Delta t_2$.

For a proper comparison, the first time step is considered, in which the effect of the coefficient A is excluded and hence only the effect of the coefficient B and eventually the natural frequency ω_n is investigated. In the first time step, coefficient A equals to $\Delta t g$ for both systems as the initial conditions of the problem are $\hat{u}\,(0) = \hat{v}\,(0) = 0$. The numerical values of the considered parameters for both systems are listed in Table 3.1 and the gravitational acceleration is set to $10\,[m/s^2]$.

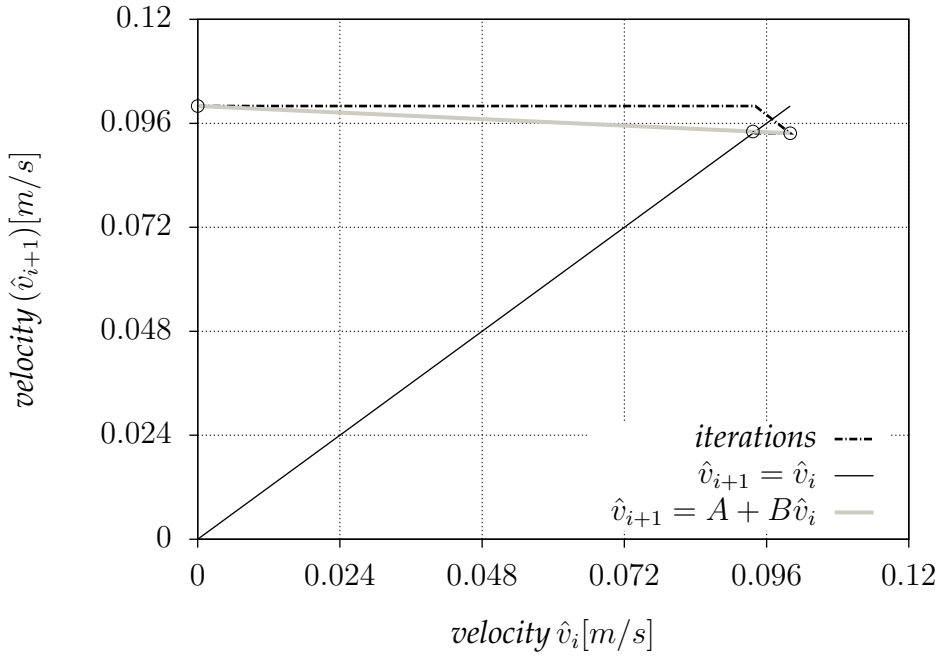
The numerical results of the iteration procedure in the first time step are given in Table 3.2 and their graphical interpretations are illustrated in Figure 3.8. We see that system 1, which has higher frequency than system 2, requires more iterations than what system 2 requires to reach a solution with the desired accuracy. It must be realized that the value of the velocity corresponding to $i = 1$ in Table 3.2 represents the predicted value to start the iterations and the following values are the subsequent corrected values of the velocity until convergence is reached. In this example, the velocity is predicted so that $\hat{v}_1 = A$ at the end of the first iteration (predictor). This is a result of setting the velocity at the beginning of the first iteration to zero, i.e., the second term of the right-hand side of Equation 3.81 is zero at the beginning of the first iteration. The time superscript is dropped from the velocity $\hat{v}$ in Table 3.2 and Figure 3.8 as only a particular time step is considered for investigations. It remains to emphasize here that Δt on one

Table 3.1: The numerical values of the parameters for the considered systems

system	$\omega_n\,[rad/s]$	$\Delta t\,[s]$	$\lambda_f\,[-]$	$\lambda_v\,[-]$	$tol\,[\%]$
1	150	0.01	0.5	0.5	1
2	50	0.01	0.5	0.5	1



(a)



(b)

Figure 3.8: Illustration of the iteration procedure of the first time step (a) high frequency system (b) low frequency system[a]

[a]The legends apply to both figures (a) and (b).

Table 3.2: Iteration of the first time step

system	$i\,[-]$	$\hat{v}_i\,[m/s]$	$R^i_{error}\,[\%]$
1	1	0.10000	100.000
high frequency	2	0.04375	128.572
	3	0.07539	41.9685
	4	0.05759	30.9081
	5	0.06760	14.8076
	6	0.06197	9.08504
	7	0.06514	4.86644
	8	0.06336	2.80934
	9	0.06436	1.55376
	10	0.06379	0.89355
2	1	0.10000	100.000
low frequency	2	0.09375	6.66667
	3	0.09414	0.41427

hand and $\lambda_f \lambda_v$ on the other hand can have the same effect on the number of iterations as the effect of ω_n. This fact can be realized if one scrutinizes Equations 3.81 and 3.82.

3.4.1.2 Conservation properties of the scheme

A detailed study of the energy conservation properties of the integration scheme is presented here. A simple and direct understanding of the conditions under which the scheme conserves, dissipates or increases energy can be achieved by applying Liouville's theorem [79]. For a dynamic system to be conservative, it should preserve the phase-space area in the single degree-of-freedom system, and the phase-space volume in general. Following the approach presented by Timberlake and Hasbun [173], one can investigate the conservation properties of an integration scheme by comparing the infinitesimal phase-space area for two subsequent time steps. For this task, we compare the phase-space areas at time t and $(t + \Delta t)$, i.e.,

$$d\hat{u}^{t+\Delta t} d\hat{v}^{t+\Delta t} = |\boldsymbol{\mathcal{H}}|\, d\hat{u}^t d\hat{v}^t, \tag{3.84}$$

where $|\boldsymbol{\mathcal{H}}|$ denotes the determinant of the Jacobian matrix $\boldsymbol{\mathcal{H}}$ of the scheme, which is defined as

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \dfrac{\partial \hat{u}^{t+\Delta t}}{\partial \hat{u}^t} & \dfrac{\partial \hat{u}^{t+\Delta t}}{\partial \hat{v}^t} \\ \dfrac{\partial \hat{v}^{t+\Delta t}}{\partial \hat{u}^t} & \dfrac{\partial \hat{v}^{t+\Delta t}}{\partial \hat{v}^t} \end{bmatrix}. \tag{3.85}$$

The scheme conserves energy only if the infinitesimal phase-space area is preserved, i.e., $|\boldsymbol{\mathcal{H}}| = 1$. It dissipates energy and produces an artificial numerical damping if $|\boldsymbol{\mathcal{H}}| < 1$, whereas energy is increased when $|\boldsymbol{\mathcal{H}}| > 1$. To investigate the effect of the parameters λ_f and λ_v on $|\boldsymbol{\mathcal{H}}|$ and eventually on the conservation properties of the scheme, we re-write

Equations 3.76 and 3.78 as

$$\hat{v}^{t+\Delta t} \approx \hat{v}^t + \Delta t g - \lambda_f \Delta t \omega_n^2 \hat{u}^{t+\Delta t} - (1-\lambda_f)\,\Delta t \omega_n^2 \hat{u}^t \tag{3.86}$$

and

$$\hat{u}^{t+\Delta t} \approx \hat{u}^t + \lambda_v \Delta t \hat{v}^{t+\Delta t} + (1-\lambda_v)\,\Delta t \hat{v}^t, \tag{3.87}$$

respectively. Hence, Equation 3.85 can now be written as

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \dfrac{1-(1-\lambda_f)\,\lambda_v \Delta t^2 \omega_n^2}{1+\lambda_f \lambda_v \Delta t^2 \omega_n^2} & \dfrac{\Delta t}{1+\lambda_f \lambda_v \Delta t^2 \omega_n^2} \\ \dfrac{-\Delta t \omega_n^2}{1+\lambda_f \lambda_v \Delta t^2 \omega_n^2} & \dfrac{1-\lambda_f\,(1-\lambda_v)\,\Delta t^2 \omega_n^2}{1+\lambda_f \lambda_v \Delta t^2 \omega_n^2} \end{bmatrix} \tag{3.88}$$

and hence

$$|\boldsymbol{\mathcal{H}}| = \frac{1+(1-\lambda_f-\lambda_v+2\lambda_f\lambda_v)\,\Delta t^2\omega_n^2 + \left(\lambda_f\lambda_v - \lambda_f\lambda_v^2 - \lambda_f^2\lambda_v + \lambda_f^2\lambda_v^2\right)\Delta t^4 \omega_n^4}{1+2\lambda_f\lambda_v\Delta t^2\omega_n^2 + \lambda_f^2\lambda_v^2\Delta t^4\omega_n^4}. \tag{3.89}$$

Scrutiny of Equation 3.89 shows that $|\boldsymbol{\mathcal{H}}|$ is a symmetric polynomial of the parameters λ_f and λ_v. It can be easily shown that $\forall\,(\lambda_f+\lambda_v) = 1$, $|\boldsymbol{\mathcal{H}}| = 1$ and hence the scheme is conservative. The scheme is dissipative and produces numerical damping $1 < \forall\,(\lambda_f+\lambda_v) \leqslant 2$, which results in $|\boldsymbol{\mathcal{H}}| < 1$, whereas energy is increased with time $0 \leqslant \forall\,(\lambda_f+\lambda_v) < 1$ which gives $|\boldsymbol{\mathcal{H}}| > 1$. One must keep in mind that $(\lambda_f, \lambda_v) \in [0,1]$.

For numerical investigation, the spring-mass system with frequency $\omega_n = 150\,[rad/s]$ is considered again. A series of calculations with different sets of λ_f and λ_v were performed. All calculations for these investigations were carried out with $\Delta t = 0.001\,[s]$ and $tol = 1\,[\%]$. The numerical results confirm the approach presented by Timberlake and Hasbun [173]. Figure 3.9 shows results for phase-space area for different values of $(\lambda_f+\lambda_v)$. One can obviously see that for $(\lambda_f = 0.5, \lambda_v = 0.5)$, the scheme preserves the phase-space area of the considered system and hence its energy. For $(\lambda_f = 0, \lambda_v = 0)$, the area increases, whereas it decreases for $(\lambda_f = 1, \lambda_v = 1)$.

It should be pointed out here that both features, numerical dissipation and increase of energy, inherent in the scheme do not resolve the physical properties of the system when dynamics and wave propagation are of interest. However, energy dissipation could be an attractive feature when considering quasi-static problems, in which the static equilibrium solution is of interest rather than the propagation of waves. Hence, an advantage of the numerical artificial damping introduced by the scheme is that the static equilibrium solution of a quasi-static problem can be reached much faster. When the scheme is used as dissipative, i.e., $1 < \forall\,(\lambda_f+\lambda_v) \leqslant 2$, the amount of numerical damping is proportional to the value of $(\lambda_f+\lambda_v)$. This fact is depicted in Figure 3.10, in which a comparison of different values of $(\lambda_f+\lambda_v)$ is shown. In this figure, it can be seen that the higher the value of $(\lambda_f+\lambda_v)$ the faster the decay of the amplitude. The results are compared to the analytical solution which is given in reference [172] as

$$\hat{u}\,(t) = \frac{g}{\omega_n^2}\left[1-\cos\left(\omega_n t\right)\right] \tag{3.90}$$

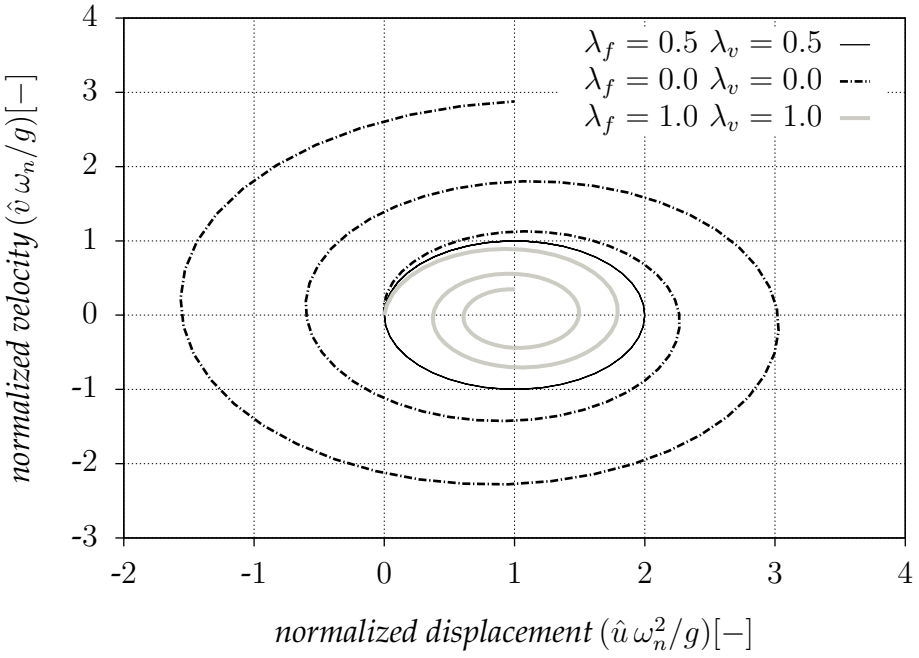


Figure 3.9: Phase-space area for the single degree-of-freedom spring-mass system

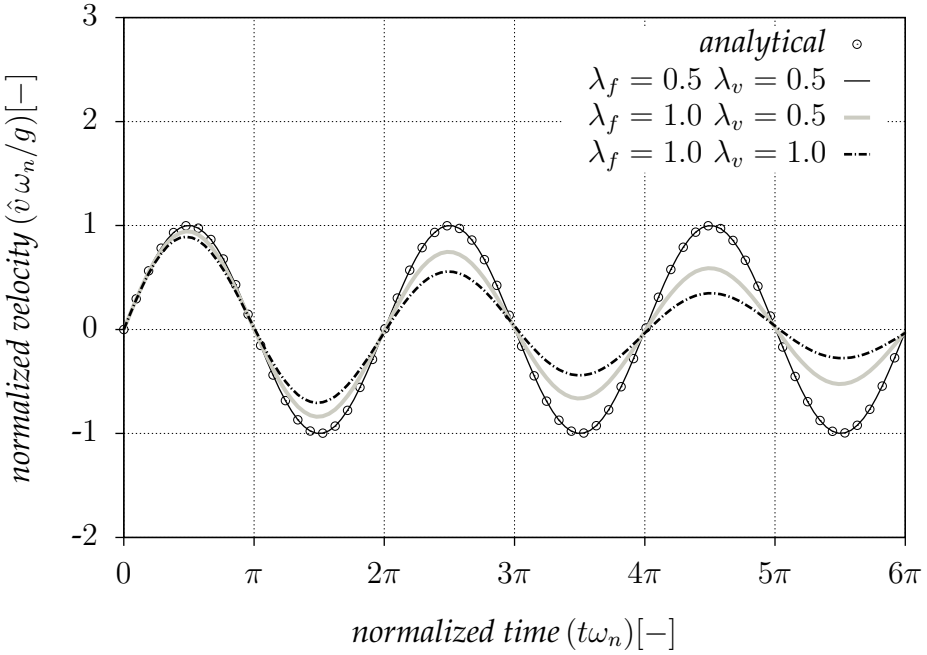


Figure 3.10: Results of different values of λ_f and λ_v compared to the analytical solution

and

$$\hat{v}\left(t\right) = \frac{g}{\omega_n} \sin\left(\omega_n t\right). \tag{3.91}$$

3.4.1.3 Stability requirements and critical time step

To analyze the stability of the scheme, we follow the procedure presented in reference [39]. The stability of a numerical integration scheme requires the following condition

$$\text{a scheme is stable} \iff \mathcal{J}\left(\boldsymbol{\mathcal{H}}\right) \leqslant 1, \tag{3.92}$$

in which $\mathcal{J}\left(\boldsymbol{\mathcal{H}}\right)$ is the spectral radius of the matrix $\boldsymbol{\mathcal{H}}$. It is defined as

$$\mathcal{J}\left(\boldsymbol{\mathcal{H}}\right) = max\left|\mathcal{I}_i\right| \qquad i = 1, 2 \tag{3.93}$$

with $\mathcal{I}_i$ being the eigenvalues of the matrix $\boldsymbol{\mathcal{H}}$. For simplicity, we consider here the case in which $(\lambda_f = 0, \lambda_v = 1)$. Substituting these values in Equation 3.88, the matrix $\boldsymbol{\mathcal{H}}$ is then reduced to

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} 1 - \Delta t^2 \omega_n^2 & \Delta t \\ \\ -\Delta t \omega_n^2 & 1 \end{bmatrix} \tag{3.94}$$

and hence the two eigenvalues of $\boldsymbol{\mathcal{H}}$ are

$$\begin{aligned} \mathcal{I}_1 &= \left(1 - \frac{1}{2}\Delta t^2 \omega_n^2\right) + \sqrt{\frac{1}{4}\Delta t^4 \omega_n^4 - \Delta t^2 \omega_n^2} \\ \mathcal{I}_2 &= \left(1 - \frac{1}{2}\Delta t^2 \omega_n^2\right) - \sqrt{\frac{1}{4}\Delta t^4 \omega_n^4 - \Delta t^2 \omega_n^2}. \end{aligned} \tag{3.95}$$

Applying the stability condition of Equation 3.92, one can show that

$$\left(1 - \frac{1}{2}\Delta t^2 \omega_n^2 + \sqrt{\frac{1}{4}\Delta t^4 \omega_n^4 - \Delta t^2 \omega_n^2}\right) \leqslant -1$$

and hence

$$\Delta t^2 \omega_n^2 \leqslant 4$$

which leads to the following critical time step

$$\Delta t_{crit} = \frac{2}{\omega_n}. \tag{3.96}$$

The same stability analysis can be applied to other cases with different values of λ_f and λ_v. Table 3.3 lists some integration schemes depending on the values of the parameters λ_f and λ_v. Regarding the accuracy of the one-step integration schemes, the analysis done by Dahlquist [54] showed that all one-step schemes are first-order accurate except Crank-Nicolson scheme is second-order accurate. It is worth mentioning here that Euler-Cromer scheme results from a minor modification to the standard Euler scheme. The

Table 3.3: Some one-step integration schemes for a second-order ODE

λ_f	λ_v	scheme	implicitness	$\lvert\boldsymbol{\mathcal{H}}\rvert$	stability[a]	accuracy
0	0	forward Euler	fully explicit	$1+\Delta t^2\omega_n^2$	un-us	$O\left(\Delta t^1\right)$
0	1	Euler-Cromer	semi implicit	1	$\Delta t_{crit}=\frac{2}{\omega_n}$	$O\left(\Delta t^1\right)$
$\frac{1}{2}$	$\frac{1}{2}$	Crank-Nicolson	fully implicit	1	un-s	$O\left(\Delta t^2\right)$
1	1	backward Euler	fully implicit	$\left(1+\Delta t^2\omega_n^2\right)^{-1}$	un-s	$O\left(\Delta t^1\right)$
1	0	-	semi implicit	1	$\Delta t_{crit}=\frac{2}{\omega_n}$	$O\left(\Delta t^1\right)$

[a] un (unconditionally), us (unstable), s (stable).

parameter λ_v has a value of one in Euler-Cromer scheme instead of zero in the standard Euler scheme [51]. This scheme is also known in literature as a modified Euler method (MEM) [78] and is widely used in modeling dynamic problems, see reference [62] and references therein.

It should be noted that the viscous damping decreases the size of the critical time step [21]. On denoting the damping ratio as ζ_d, the critical time step for a damped system is given as

$$\Delta t_{crit}=\frac{2}{\omega_n}\left(\sqrt{\zeta_d^2+1}-\zeta_d\right). \tag{3.97}$$

Hence, for $\zeta_d>0$, the time step size decreases by a factor of $\left(\sqrt{\zeta_d^2+1}-\zeta_d\right)$.

3.4.1.4 A predictor for the first iteration

Attention should be paid to the possibilities of predicting the velocity for the first iteration which is corrected in a couple of correction steps during the iteration procedure within the time step. The predicted value of velocity plays an important role in the speed of convergence and eventually on the efficiency of calculations. For this task, the three trivial possible predictors that follow are discussed:

- *predictor 1*: The velocity is predicted as $\hat{v}_{i=1}^{t+\Delta t}=\hat{v}^t+\Delta t\left[g-\omega_n^2\hat{u}^t\right]$, which corresponds to the velocity at $(t+\Delta t)$ when setting $\lambda_f=0$.
- *predictor 2*: Using the velocity of the previous time step, i.e., $\hat{v}_{i=1}^{t+\Delta t}=\hat{v}^t$.
- *predictor 3*: The predictor is $\hat{v}_{i=1}^{t+\Delta t}=\hat{v}^t\left[1-\lambda_f\left(1-\lambda_v\right)\Delta t^2\omega_n^2\right]+\Delta t\left[g-\omega_n^2\hat{u}^t\right]$, which results when setting the second term of the right-hand side of Equation 3.81 to zero.

A series of calculations was performed using the three proposed predictors with different time steps. The system of $\omega_n=150\,[rad/s]$ is considered again in these calculations. Integration parameters $\lambda_f=\lambda_v=0.5$ were adopted with a tolerance of $tol=1\,[\%]$. The results of using a time step $\Delta t=1.5/\omega_n$ were chosen to be shown in Figures 3.11 and 3.12. Figure 3.11 compares the results of using the three predictors to the analytical solution. It can be seen that the curves corresponding to the three different predictors

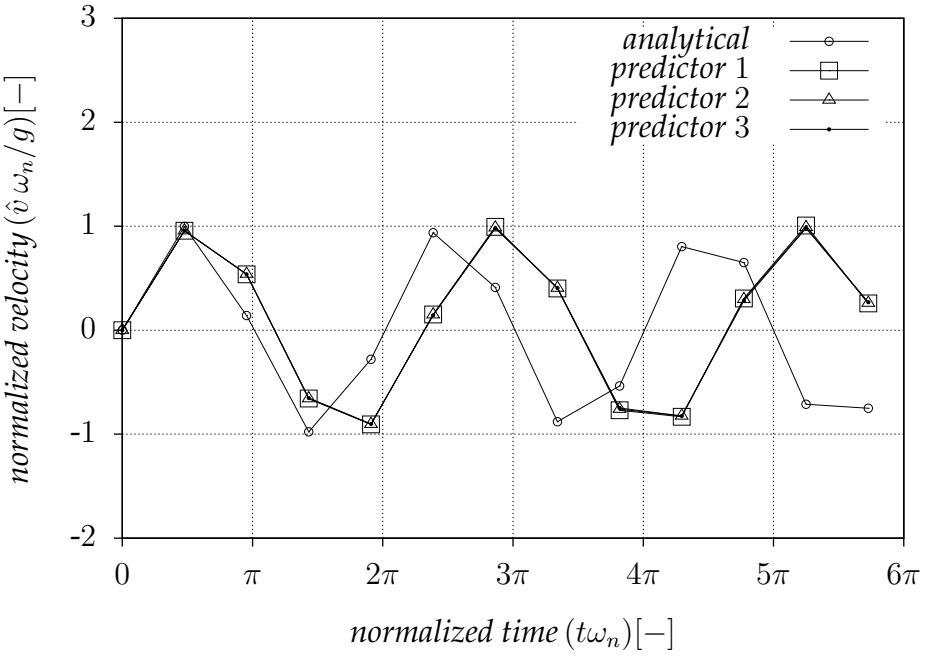


Figure 3.11: Normalized velocity using a time step size of $\Delta t = 1.5/\omega_n$

coincide each other while they deviate considerably from the analytical solution. This is expected as the time step is too large to accurately capture the analytical solution of the problem.

When considering the computational effort and the speed of convergence, it is obvious from Figure 3.12 that the solution corresponding to predictor 3 requires less iterations than those for the other two predictors. However, predictor 3 has an explicit dependency on the natural frequency ω_n and hence it is difficult to generalize for FEM as the natural frequency is costly to compute and sometimes even impossible when considering non-linear systems. Therefore, using predictors 1 and 2 would seem to be most appropriate choice. This would, of course, need more investigation when considering FEM problems later in this chapter. It is worth mentioning that in this example, the maximum time step size that could be adopted is $\Delta t \approx 2.0/\omega_n$. The calculations diverged when using a time step size that is just beyond this size.

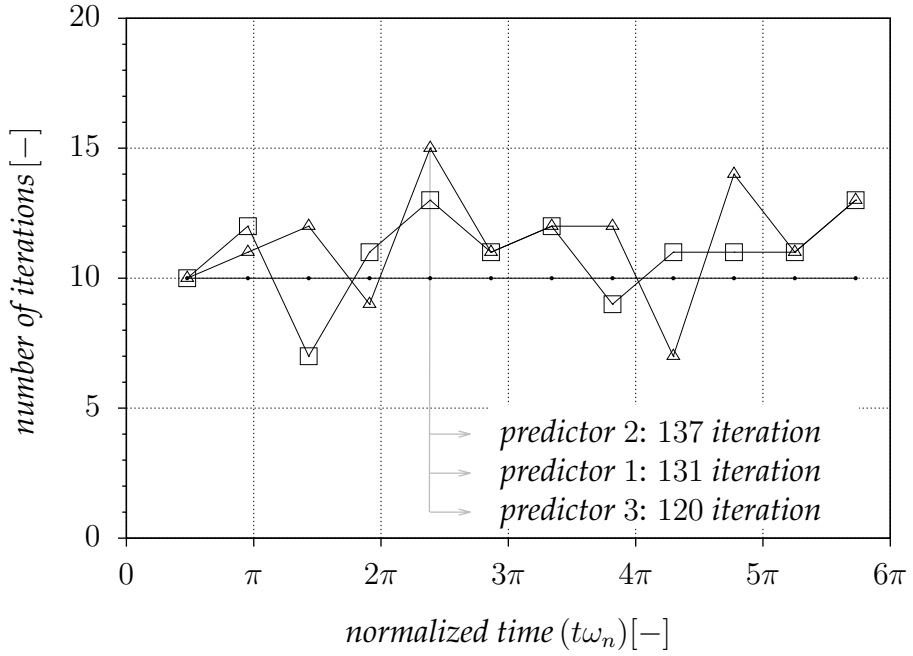


Figure 3.12: Number of iterations for different predictors using a time step size of $\Delta t = 1.5/\omega_n$

3.4.2 Time integration of FEM equilibrium

In Section 3.4.1, the time integration of the second-order ODE of the single degree-of-freedom spring-mass system was presented. After the detailed discussion of the integration scheme, its stability conditions and conservation properties, we focus in this section on the time stepping of FEM equilibrium equations. For this we re-consider Equation 3.67, which is conveniently re-written here as

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{F} \tag{3.98}$$

with $\boldsymbol{F} = \boldsymbol{F}^{ext} - \boldsymbol{F}^{int}$. The above equation represents a system of second-order ordinary differential equations in time, but discrete in space. Let us apply the time integration scheme of Section 3.4.1 to Equation 3.98

$$\boldsymbol{M}\frac{d\boldsymbol{v}}{dt} = \boldsymbol{F}, \tag{3.99}$$

where the term $\frac{d\boldsymbol{v}}{dt}$ replaces $\boldsymbol{a}$. Integrating this equation in the time interval $[t, t+\Delta t]$ yields

$$\int_t^{t+\Delta t} \boldsymbol{M} d\boldsymbol{v} = \int_t^{t+\Delta t} \boldsymbol{F} dt. \tag{3.100}$$

Hence,

$$\boldsymbol{M}\left(\boldsymbol{v}^{t+\Delta t} - \boldsymbol{v}^t\right) \approx \left[\lambda_f \boldsymbol{F}^{t+\Delta t} + (1-\lambda_f)\,\boldsymbol{F}^t\right]\Delta t, \tag{3.101}$$

which gives

$$\boldsymbol{v}^{t+\Delta t} \approx \boldsymbol{v}^t + \boldsymbol{M}^{-1}\left[\lambda_f \boldsymbol{F}^{t+\Delta t} + (1-\lambda_f)\,\boldsymbol{F}^t\right]\Delta t. \tag{3.102}$$

The time superscript is dropped from the mass matrix $\boldsymbol{M}$ as in Lagrangian FEM the mass matrix is constant throughout the computations. The displacement $\boldsymbol{u}$ is the time integral of the velocity $\boldsymbol{v}$, i.e., $\boldsymbol{v} = \frac{d\boldsymbol{u}}{dt}$. Hence,

$$\int_t^{t+\Delta t} d\boldsymbol{u} = \int_t^{t+\Delta t} \boldsymbol{v} dt, \tag{3.103}$$

which is approximated as

$$\boldsymbol{u}^{t+\Delta t} - \boldsymbol{u}^t \approx \left[\lambda_v \boldsymbol{v}^{t+\Delta t} + (1-\lambda_v)\,\boldsymbol{v}^t\right] \Delta t \tag{3.104}$$

or

$$\Delta \boldsymbol{u}^{t+\Delta t} \approx \left[\lambda_v \boldsymbol{v}^{t+\Delta t} + (1-\lambda_v)\,\boldsymbol{v}^t\right] \Delta t, \tag{3.105}$$

where $\Delta \boldsymbol{u}^{t+\Delta t} = \boldsymbol{u}^{t+\Delta t} - \boldsymbol{u}^t$ represents the vector of incremental displacement at the end of the time step. Unlike the single degree-of-freedom dynamic system, it is impossible to find a recursion equation, similar to Equation 3.81, for the iteration within a time step for a finite element system. Therefore, Equations 3.102 and 3.105 together with the constitutive relation must be solved sequentially during iterations. The strain rate at a Gauss point q of element e is calculated as

$$\dot{\boldsymbol{\varepsilon}}_q^{t+\Delta t} = \boldsymbol{B}_q \boldsymbol{v}_e^{t+\Delta t} \tag{3.106}$$

and the strain increment follows as

$$\Delta \boldsymbol{\varepsilon}_q^{t+\Delta t} = \boldsymbol{B}_q \Delta \boldsymbol{u}_e^{t+\Delta t}. \tag{3.107}$$

The vectors $\boldsymbol{v}_e$ and $\Delta \boldsymbol{u}_e$ are the velocity and the incremental displacement vectors of element e, respectively. They are extracted from the global vectors $\boldsymbol{v}$ and $\Delta \boldsymbol{u}$ using the connectivity matrix, see Appendix D. However, they have the following components

$$\begin{aligned} \boldsymbol{v}_e &= \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{n_{en}1} & u_{n_{en}2} & u_{n_{en}3} \end{bmatrix}^T, \\ \Delta \boldsymbol{u}_e &= \begin{bmatrix} \Delta u_{11} & \Delta u_{12} & \Delta u_{13} & \dots & \Delta u_{n_{en}1} & \Delta u_{n_{en}2} & \Delta u_{n_{en}3} \end{bmatrix}^T \end{aligned} \tag{3.108}$$

with, for instance, v_{12} indicating the velocity at node 1 of element e in the direction of coordinate x_2. The strain increment vector has the following components

$$\Delta \boldsymbol{\varepsilon} = \begin{bmatrix} \Delta\varepsilon_{11} & \Delta\varepsilon_{22} & \Delta\varepsilon_{33} & \Delta\gamma_{12} & \Delta\gamma_{23} & \Delta\gamma_{31} \end{bmatrix}^T, \tag{3.109}$$

which is then used to update stresses of the Gauss point using the constitutive relation, i.e.,

$$\{\boldsymbol{\sigma}, \text{material state}\}^t \xrightarrow[\Delta\boldsymbol{\varepsilon}^{t+\Delta t}]{\text{constitutive relation}} \{\boldsymbol{\sigma}, \text{material state}\}^{t+\Delta t}. \tag{3.110}$$

The flowchart of Figure 3.13 summarizes the iteration procedure and the way of advancing the solution from time t to $t + \Delta t$ for a finite element system. In this flowchart, i denotes the iteration index, n_{elm} is the total number of elements, n_{eq} is the number of Gauss points per element, R_{error} is the relative error, tol is the error tolerance and $\| \ \|$ indicates the Euclidean norm of a vector.

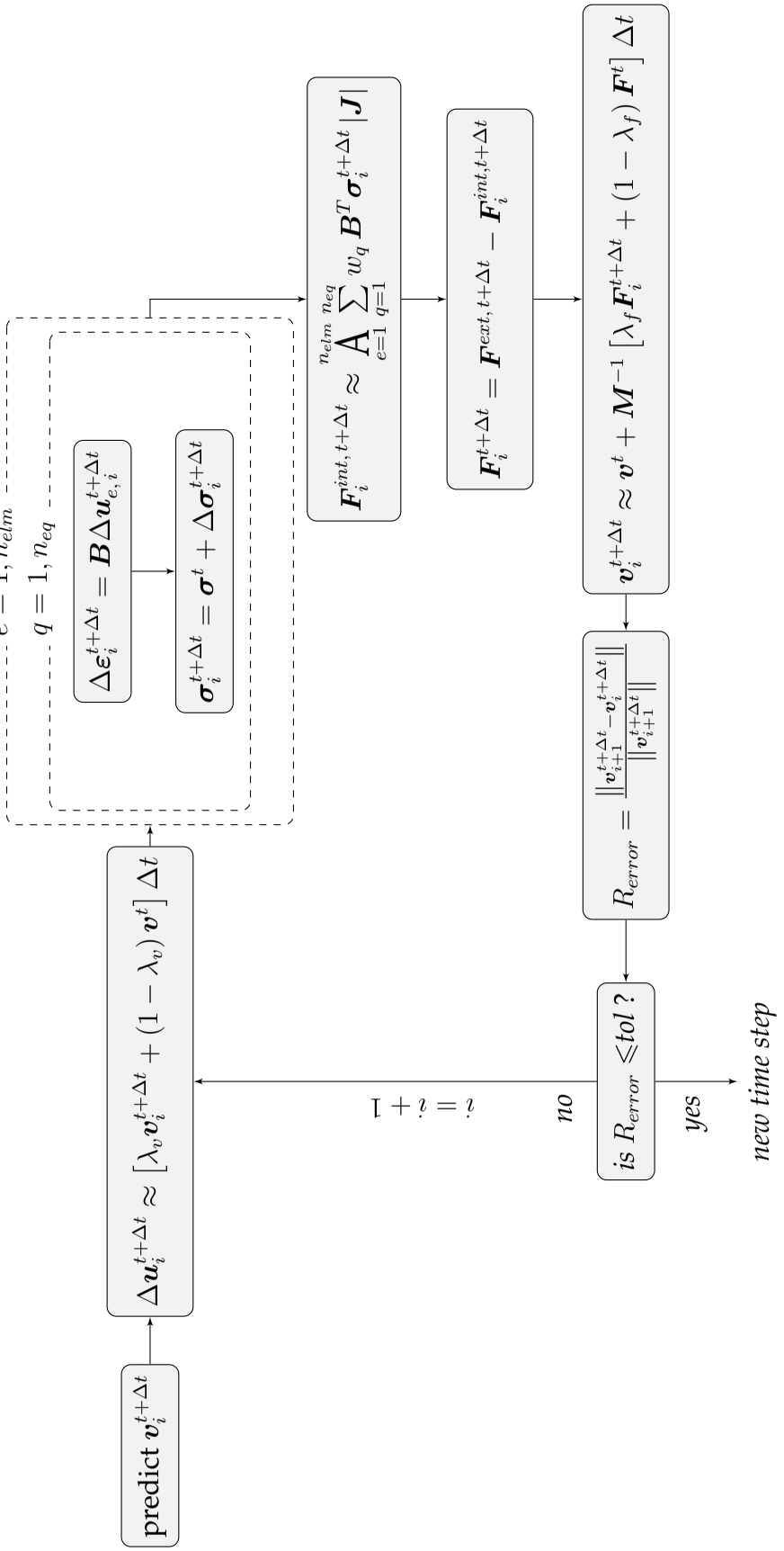


Figure 3.13: Flowchart of the iteration procedure within a time step

3.4.2.1 Critical time step for discrete systems

The stability analysis of Section 3.4.1 shows that the critical time step of a dynamic system, when adopting a conditionally-stable integration scheme, depends directly on the natural frequency of the system. For the single degree-of-freedom spring-mass system, the natural frequency is trivial and explicitly available. However, for complex systems, the calculation of the natural frequency requires considerable computational effort and some times it is not possible to calculate for non-linear systems [62]. The alternative is to use the CFL (Courant, Friedrichs and Lewy) condition [50] where the critical time step is bounded by the characteristic length of the element and the wave speed of the material. We shall consider first the critical time step of a rod element for better understanding. For convenience, we re-call Figure 3.4, which represents a two-noded rod element. It can be easily shown that the element lumped-mass matrix $\boldsymbol{M}_e$ and the element stiffness matrix $\boldsymbol{K}_e$ are

$$\boldsymbol{M}_e \approx \varrho_e \, l_e \, A_e \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{K}_e \approx \frac{E_e A_e}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.111}$$

with ϱ_e being the mass density of the material of element e, E_e is Young's modulus, l_e and A_e are the length and cross-sectional area of element e, respectively. The natural frequency ω_n can be determined solving the following equation

$$\left| \boldsymbol{K}_e - \omega_n^2 \boldsymbol{M}_e \right| = \boldsymbol{0}, \tag{3.112}$$

which results in

$$\omega_n = \frac{2}{l_e} \sqrt{\frac{E_e}{\varrho_e}}, \tag{3.113}$$

in which the term $\sqrt{\frac{E_e}{\varrho_e}}$ represents the speed of wave propagation, which is denoted in this thesis as c_p. Hence,

$$\omega_n = \frac{2 c_p}{l_e}. \tag{3.114}$$

Comparing the above equation to Equation 3.96, one can obviously see that

$$\Delta t_{crit} = \frac{l_e}{c_p}, \tag{3.115}$$

which tells us that the wave is limited to travel no farther than the element length within a time step when adopting a conditionally-stable time integration scheme. In our idealized example there is no Poisson's effect as we assume one-dimensional deformation and hence lateral strains do not take place. For this reason the speed of the wave is proportional to Young's modulus. For a general use of Equation 3.115, Young's modulus E should be replaced by E^c which is the constrained modulus of the material. It is related to Young's modulus E and Poisson's ratio ν by

$$E^c = \frac{(1-\nu)}{(1+\nu)\,(1-2\nu)} E. \tag{3.116}$$

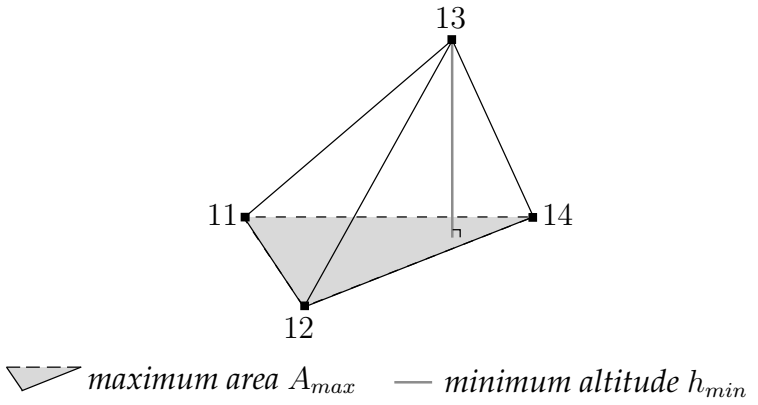


Figure 3.14: Illustration of the minimum altitude h_{min} for a tetrahedral

Furthermore, l_e should be replaced by the characteristic length of the element, which depends on the shape of the element. For the tetrahedral element adopted in this thesis, the characteristic length is the minimum altitude of the tetrahedral, which is the shortest distance from the side of maximum area to the opposite vertex, see Figure 3.14. In this thesis, the minimum altitude of a tetrahedral is denoted by h_{min}. Hence, the critical time step is

$$\Delta t_{crit} = \frac{h_{min}}{\sqrt{\frac{E^c}{\varrho}}}. \tag{3.117}$$

The above equation must be applied for all elements in the mesh and the smallest time step must be used when adopting a conditionally-stable time integration scheme, see Table 3.3.

3.5 Numerical examples

In this section, we consider two problems with existing analytical solutions to validate the implementation of the FEM model and to investigate the applicability, advantages and limitations of the time integration scheme with FEM. The first problem is the spring-mass system. The motivation to consider this problem was that, its single degree-of-freedom form was considered and analyzed comprehensively in Section 3.4. Hence, the reader may easily link the single degree-of-freedom problem to its FEM solution. The second problem is a tension bar problem, in which an elastic bar being stretched is analyzed. Both problems are modeled as linear elastic material without any geometry update.

3.5.1 Spring-mass system

In this example, the solution of the spring-mass system problem using FEM is presented. Figure 3.15 shows how the spring-mass system can be modeled in three-dimensional space as a rigid block attached to a light rod. The light rod acts here as a weightless

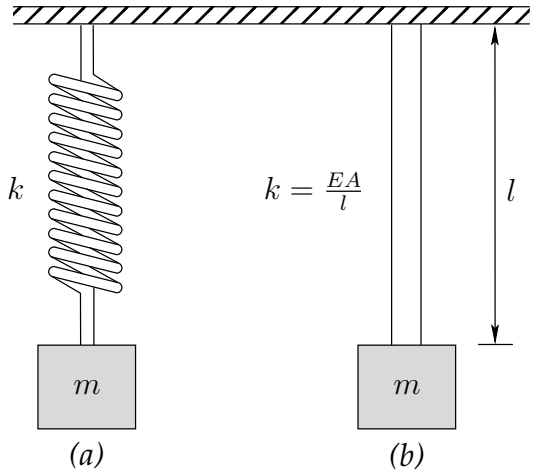


Figure 3.15: Spring-mass system (a) with equivalent rod-block system (b)

spring with a spring coefficient k related to Young's modulus E, the cross-sectional area A and the length l of the rod via

$$k = \frac{EA}{l}. \tag{3.118}$$

Using the above equation together with the mass m of the attached block, one can find a natural frequency that is equivalent to the one of the single degree-of-freedom spring-mass system. Hence, the natural frequency can be written as

$$w_n = \sqrt{\frac{EA}{lm}}. \tag{3.119}$$

For a FEM simulation of the problem, we analyzed a rod of $1\,[m]$ length with a square cross-section of $0.01\,[m^2]$ area. The attached block is a cube of $0.1\,[m]$ edge length. The properties of the block and the rod materials are listed in Table 3.4. With these properties, the natural frequency of the problem is $\omega_n = 10\,[rad/s]$. The finite element mesh of the problem is shown in Figure 3.16. It was constructed of 66 tetrahedral elements with 48 nodes. The top surface of the mesh was fully fixed and roller boundaries were prescribed along the sides so that only displacement in the vertical direction was possible. Considering the adopted mesh and the material properties, the critical time step was found to be $\Delta t_{crit} = 5.7735 \times 10^{-4}[s]$. For all computations of this problem, an error tolerance of $tol = 1\,[\%]$ was used.

Table 3.4: Material properties of the spring-mass system problem

property	unit	rod	block
E	$[kN/m^2]$	10	100
ν	$[-]$	0	0
ϱ	$[kg/m^3]$	1	1000
c_p	$[m/s]$	100	10

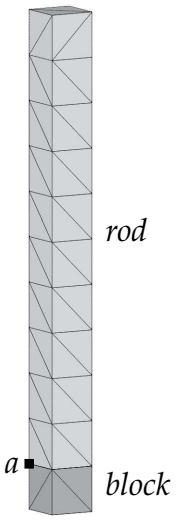


Figure 3.16: Mesh of the spring-mass system problem

Validation using critical time step We would like first to validate the implementation of FEM and the time integration scheme. For this purpose, some calculations using the critical time step, $\Delta t_{crit} = 5.7735 \times 10^{-4}\ [s]$, were performed. These calculations were carried out adopting predictor 1. Node a in Figure 3.16 was chosen as a sample point to plot the results. Figure 3.17 shows the results of the vertical velocity v_2 of node a for two sets of integration parameters. We observe that the numerical predictions are almost

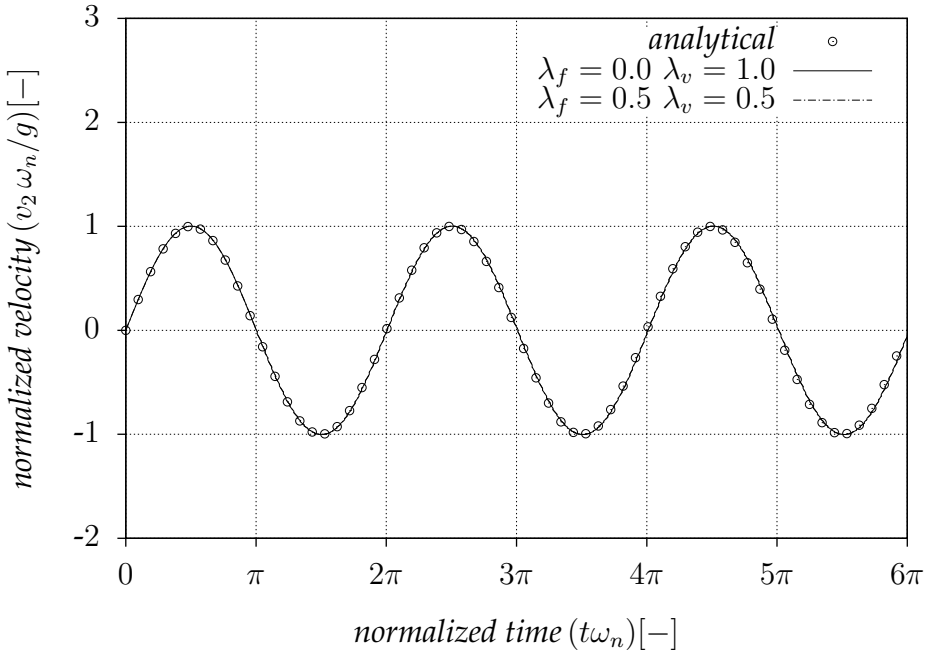
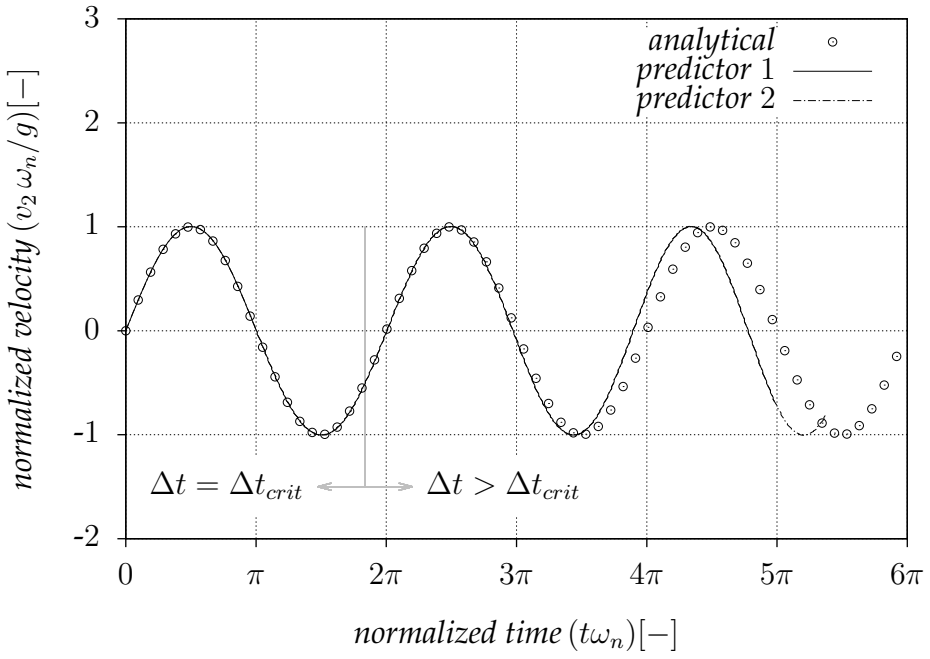


Figure 3.17: FEM solution of the spring-mass system problem using critical time step

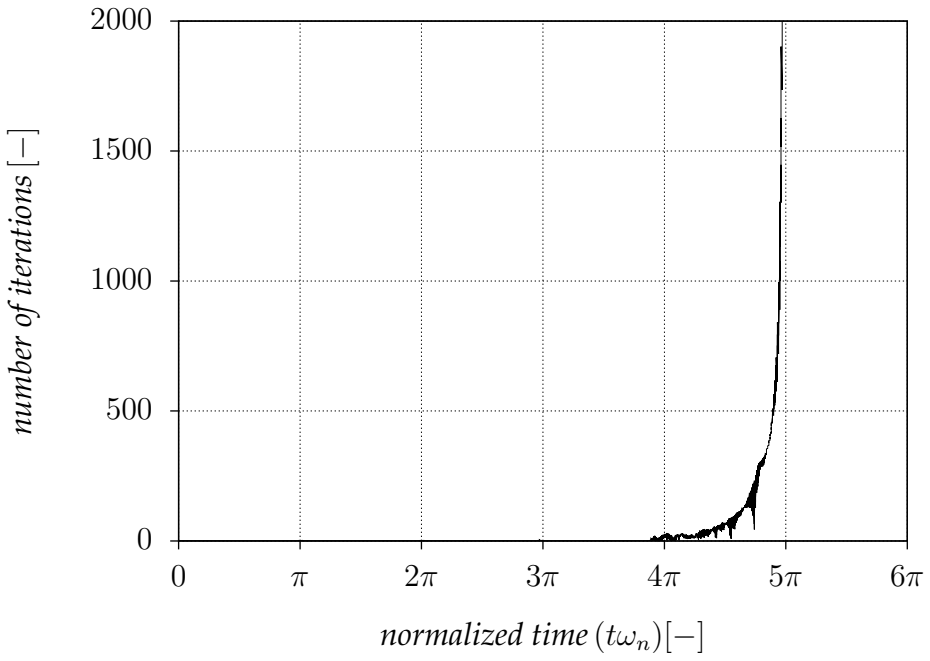
identical to the analytical solution.

Beyond the critical time step Having validated the FEM solution using the critical time step, investigations of the scheme performance with time step size beyond critical one were also carried out. A comparison between predictors 1 and 2 was included in the investigations as well. The computations this time were restricted to the case of $\lambda_f = \lambda_v = 0.5$. Figure 3.18(a) shows the results of two computations compared to the analytical solution. The results in Figure 3.18(a) were computed using the critical time step to $t = 0.57735\ [s]$ which corresponds to a normalized time $t\omega_n \approx 1.837\pi$. After this point, the time step was increased linearly with an increment of $\Delta t_{crit} \times 10^{-4}\ [s]$, i.e., $5.7735 \times 10^{-8}\ [s]$.

Considering the results in which predictor 1 was adopted, the computations stopped when a time step size of $\Delta t = 1.17\Delta t_{crit}$ was reached. These computations required 142536 iterations before diverging. The required number of iterations per time step increased with increasing the time step size, see Figure 3.18(b). Up to a normalized time $t\omega_n \approx 3.91\pi$, the average number of iterations per time step was 1.033. Although not shown, the number of iterations then increased up to a couple of thousand per time step. The results look much better when adopting predictor 2. The computations continued until a time step size of $\Delta t = 1.19\Delta t_{crit}$ was reached with a corresponding total number of iterations of 5529 which is much less than what is required when adopting predictor 1. The average number of iterations per single time step when using predictor 2 was found to be 1.89 with a maximum of 3 iterations, whereas with predictor 1 a couple of hundreds of iterations were required when going beyond the critical time step size, see Figure 3.18(b). We can see from Figure 3.18(a) that the FEM solution diverges more from the analytical solution with increasing the time step size.



(a)



(b)

Figure 3.18: FEM solution of the spring-mass system problem (a)[a] normalized velocity using predictors 1 and 2 (b) number of iterations required when using predictor 1

[a] The curves of predictor 1 and 2 are virtually identical

3.5.2 Tension bar

In this example, a simple problem involving an elastic bar under tension is presented. The bar has $0.3\,[m]$ length, $0.3\,[m]$ height and $0.1\,[m]$ thickness. The material of the bar has a density of $\varrho = 10\,[kg/m^3]$, Young's modulus of $E = 1\,[kN/m^2]$ and a Poisson's ratio of $\nu = 0$. The boundary conditions were prescribed so that the right end of the bar was fully fixed, the left end and all sides were free from constraints. The left end of the bar was given a sudden velocity of $\bar{v} = 0.001\,[m/s]$ to the left. The velocity at this end was kept constant throughout the computations. At steady-state, the displacement along the bar must be linearly distributed under the considered loading and boundary conditions starting from zero at the fixed end to a value of $\bar{v}t$ at the left end, where t indicates the time. The strain is constant through the bar and increases as time increases. It should equal

$$\varepsilon\,(t) = \frac{\bar{v}t}{l}, \tag{3.120}$$

in which $\varepsilon\,(t)$ is the strain in the direction of stretching at any time t and l is the length of the bar. This uniform strain results in a uniform stress σ along the bar, which equals

$$\sigma\,(t) = E\,\varepsilon\,(t)\,. \tag{3.121}$$

The mesh of the problem is shown in Figure 3.19. It consists of 54 tetrahedral elements with 32 nodes. The minimum characteristic length in the mesh was $h_{min} = 0.057735\,[m]$, which resulted in a critical time step of $\Delta t_{crit} = 5.7735 \times 10^{-3}\,[s]$ as the wave speed of the material is $c_p = 10\,[m/s]$. The first computation was carried out using the critical time step and predictor 2. The problem was run to a time of $t = 5\,[s]$ with $\lambda_f = 0$ and $\lambda_v = 1$. The left end displaced $5 \times 10^{-3}[m]$ to the left. The stress was found to be perfectly uniform along the bar with a value of $16.67 \times 10^{-3}\,[kN/m^2]$. The same results were obtained when adopting $\lambda_f = \lambda_v = 0.5$ with the critical time step and predictor 2.

Other computations were carried out with $\lambda_f = \lambda_v = 0.5$ and predictor 2 which was shown in the spring-mass example to perform much better when going beyond the critical time step size than predictor 1. The critical time step was adopted to a time of $t = 5\,[s]$

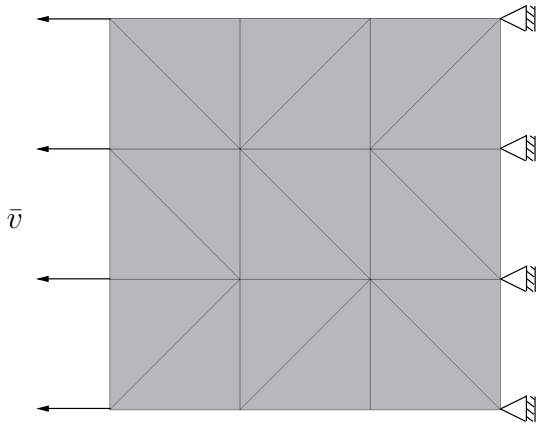


Figure 3.19: Mesh of the tension bar problem

and then the time step size was increased linearly with an increment of $\Delta t_{crit} \times 10^{-4}$, i.e., 5.7735×10^{-7} $[s]$. The calculations diverged when a time step size of $\Delta t = 1.1171 \Delta t_{crit}$ was reached, i.e., at a time of $t \approx 12.1683532\,[s]$. Table 3.5 lists the time step size with the corresponding required number of iterations before the calculations diverged. Before time step number 2019 was reached, the average number of iterations per step was found to be 3.72.

At step 2019, the number of iterations increased considerably to reach 13. The time step size adopted at step 2019 was $\Delta t = 1.11510 \Delta t_{crit}$. The relative error after the first iteration was $R_{error} \approx 35.2\,[\%]$. In spite of the high number of iterations, the results were found to be as expected. The stress was uniformly distributed along the bar with a value of 40.13157×10^{-3} $[kN/m^2]$. At step 2021 the number of iterations was 1072, which is considered very high. However, the relative error after the first iteration of this time step was $R_{error} \approx 16.7\,[\%]$, which then decreased and converged to the prescribed error tolerance $tol = 1\,[\%]$. The converged solution of this time step showed expected results as well. For the diverging point, i.e., step 2039, the relative error is plotted versus number of iterations, see Figure 3.20. We can see that the relative error reached a value of $R_{error} \approx 1.7\,[\%]$ and then increased slowly and continuously.

Table 3.5: Time step size with the corresponding required number of iterations for some time steps before the calculations diverged

time step $[-]$	$t\,[s]$	$\Delta t / \Delta t_{crit}\,[-]$	iterations $[-]$
2018	12.0330333	1.11500	4
2019	12.0394713	1.11510	13
2020	12.0459099	1.11520	4
2021	12.0523491	1.11530	1072
2022	12.0587889	1.11540	4
2023	12.0652292	1.11550	4
2024	12.0716701	1.11560	3
2025	12.0781116	1.11570	4
2026	12.0845537	1.11580	4
2027	12.0909964	1.11590	4
2028	12.0974396	1.11600	4
2029	12.1038834	1.11610	4
2030	12.1103278	1.11620	4
2031	12.1167727	1.11630	4
2032	12.1232183	1.11640	4
2033	12.1296644	1.11650	4
2034	12.1361111	1.11660	4
2035	12.1425584	1.11670	4
2036	12.1490062	1.11680	4
2037	12.1554546	1.11690	3
2038	12.1619036	1.11700	4
2039	12.1683532	1.11710	diverged

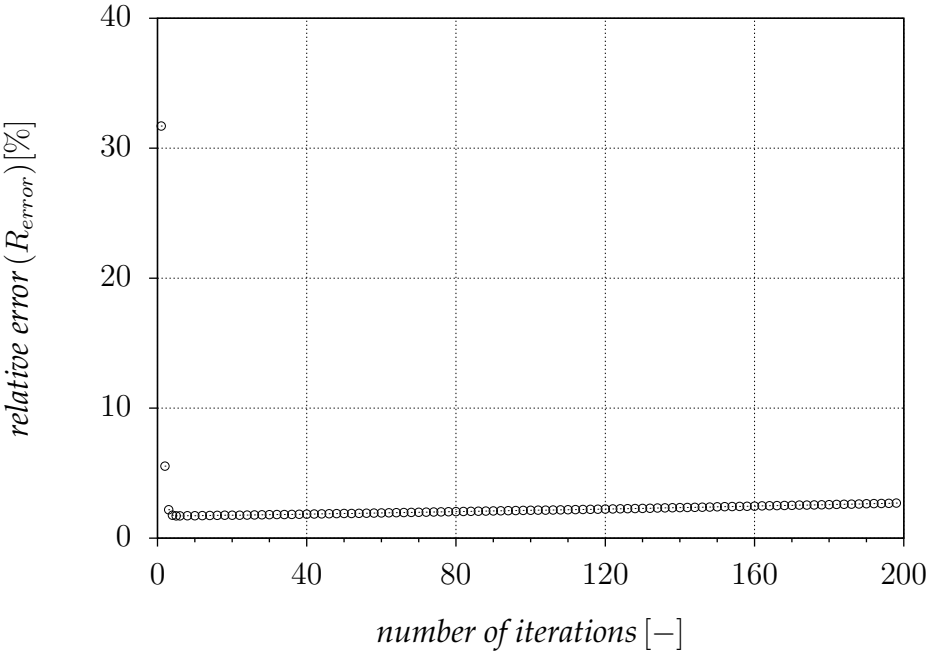


Figure 3.20: Number of iterations versus the relative error for time step 2039

3.6 Concluding remarks

Unfortunately, the algorithm in its current form is unable to circumvent the time step size restriction associated with the conditionally-stable integration schemes. This is realized even when adopting the integration scheme under conditions that are supposed to be unconditionally-stable by definition, e.g., Crank-Nicolson integration. This is consistent with the experience of others [46, 177, 193]. This might be attributed to the fact that we do not construct a stiffness matrix explicitly. Guilkey and Weiss [72] adopted an incremental-iterative solution algorithm, in which the stiffness matrix was constructed explicitly and the equations of motion were solved with Newmark integration. They reported a considerable increase in the time step size over the explicit time step size in some numerical examples. However, our intention in this thesis is not to construct a stiffness matrix, but rather to keep the algorithm simple and matrix-free to avoid complexity, particularly when considering MPM.

It should be pointed out here that a scheme similar to that of Section 3.4 was employed by Sulsky and Kaul [159]. In some numerical examples, a considerable increase of the time step size beyond the CFL restriction was possible. However, this result should not be very disappointing as long as the main goal of this thesis is concerned. The focus in this thesis is to simulate geomechanical problems involving high dynamics and high speed wave propagation, e.g., pile driving. Such problems need anyway small time step size to resolve the physical properties of the system and to capture most frequencies. On the other hand, some quasi-static problems would also be considered in this thesis, for which a damping technique would be employed to get a fast convergence to a quasi-static equilibrium. Hence, Euler-Cromer integration scheme [51] will be adopted in this

thesis. In the rest of this thesis, a fraction of the critical time step size would be used, i.e.,

$$\Delta t = \alpha_{Cour}\, \Delta t_{crit}, \tag{3.122}$$

in which α_{Cour} is a reduction factor called *Courant number*. Belytschko et al. [21] recommend the following range for α_{Cour}

$$0.8 \leqslant \alpha_{Cour} \leqslant 0.98. \tag{3.123}$$

Chapter 4

The dynamic material point method

In Chapter 3, the formulation of Lagrangian FEM was considered. The weak form, space and time discretization of the governing equations were discussed as well. The focus in this chapter is on the extension of FEM to MPM. In Section 4.1, we present the basic concept of MPM and the advantages of the method over the classical FEM. The discretization of the continuum to a cloud of material points is explained in Section 4.2. The initialization of the material points is given in Section 4.3. This includes assigning mass, forces and other properties and state variables to each material point in the mesh. The initialization of the equations of motion is discussed in Section 4.4. This section also includes the solution procedure of the equations of motion in the framework of MPM. Two solution algorithms are discussed with the advantages and disadvantages of each of them. An overview of the overall MPM solution procedure for a single time step is also given in this section.

Special attention is paid in this chapter to the applications of the boundary conditions in MPM. Both zero and nonzero kinematic and traction boundary conditions are discussed in Section 4.5. Mitigating the well-known grid-crossing error in MPM is considered in Section 4.6. In Section 4.7, the treatment of spatially unbounded domains using absorbing boundaries is discussed. The well-known procedure by Lysmer and Kuhlemeyer [110] is extended in this section to Kelvin-Voigt kind of response to limit the continuous creep of the boundary induced by the dashpots. An approach to deal with locking in low-order element is presented in Section 4.8. The convergence of the solution to a quasi-static equilibrium and the application of artificial damping are discussed in Sections 4.9 and 4.10, respectively. In Section 4.11, we present an overview of MPM contact algorithms and validate the implementation of the adopted algorithm. A penetration problem is considered in Section 4.12 to validate the current implementation of MPM.

4.1 Basic concept of MPM

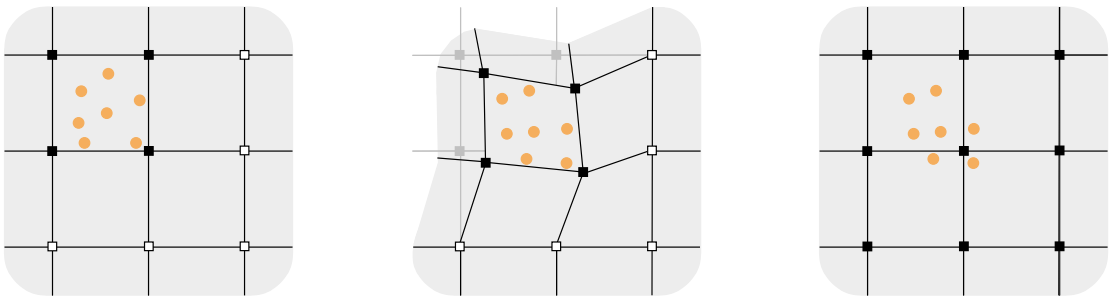
The material point method can be viewed as an extension of a finite element procedure, in which the continuum is represented by *Lagrangian points*, called *material points* or *particles* [47, 126, 162, 163, 192]. Deformations of the continuum are modeled by tracking particles that move through an Eulerian fixed mesh, requiring special book-keeping for mapping data from particles to grid points and vice versa. The particles carry all physical properties of the continuum such as mass, momentum, material parameters, strains, stresses, constitutive properties as well as external loads, whereas the Eulerian mesh

and its Gauss points carry no permanent information. At the beginning of the time step, information is transferred from particles to the computational mesh to initialize the discrete equations of motion. The computational mesh is used to determine the incremental solution of the field equations in Lagrangian fashion. At the end of the time step, the solution is mapped from the computational mesh back to particles to update their information. The computational mesh is then redefined or reset to its initial configuration and book-keeping is updated. We can observe in Figure 4.1 that as particles move through the mesh, elements and eventually nodes might become active or inactive. Hence, the number of active degrees-of-freedom changes with time during computations.

Through this approach, MPM combines the best aspects of both Lagrangian and Eulerian formulation while avoiding as much as possible the shortcomings of them. Indeed, numerical diffusion associated with the convective terms in Eulerian approach does not appear in MPM solution. On the other hand, the problem of mesh distortion, which an updated Lagrangian solution shows in large deformation, is avoided as well in the framework of MPM [161, 165]. Hence, the method is well suited to model engineering problems involving large deformation. Examples of such problems in geomechanics are slope instability, tunnel collapse and pile driving.

The weak formulation and discrete equations of motion in MPM are consistent with those of the traditional finite element method. Furthermore, the constitutive equation is applied at each single particle, avoiding the interpolation of the history-dependent variables as particles are tracked throughout the computations. Given this, the method is then capable of handling history-dependent material behavior. Elements having material with different parameters or different constitutive equations are automatically treated which is a clear advantage over Eulerian FEM [165].

MPM is naturally capable of handling non-slip contact between different bodies without a special algorithm. This is a consequence of the fact that, in MPM particle velocities are single valued in the mapping procedure between grid points and particles [165]. For



(a) beginning of time step *(b) incremental deformation* *(c) end of time step after resetting the mesh*

■ *active node* ▫ *inactive node* • *particle*

Figure 4.1: Basic concept of MPM: Lagrangian particles moving through Eulerian fixed mesh

more details about MPM, its potentials and features, the reader is referred to Chapter 2 of this thesis, in which an extensive literature review is presented. Throughout this thesis, we will often switch between both names, *material point* and *particle*. This should not cause any confusion to the reader. The same applies for *grid point* and *node*.

4.2 Discretization in MPM

In Lagrangian FEM, the nodes (and integration points) correspond to material points of the continuum and this is retained during the deformation of the continuum. One of the essential differences between Lagrangian FEM and MPM is that, in the latter, the mesh generally does not reflect any physical property of the continuum, but particles do so. As particles move through a fixed mesh during deformations (computations), meshing the whole region where particles are expected to move is necessary.

In this section, we discuss the discretization of a continuum in MPM. For this task, we consider a simple problem for illustration purposes. This problem involves a block being pushed, see Figure 4.2(a). Figure 4.2(b) shows the initially active and inactive finite elements of this problem. The initially active elements represent the undeformed geometry of the block. In the next step, material points are initialized in the active elements representing the continuum in its undeformed state. Figure 4.2(c) shows that each active element is filled with four particles in this particular example. Particles marked (dark) red are identified as *boundary particles*. They carry surface tractions throughout the computations. However, we have the option of filling the initially active elements with different number of particles, see Appendix E.

Most of the existing dynamic MPM codes are based on a regular grid. For complex structures, being represented by a cloud of material points, this would require a special CAD-type preprocessor. On the other hand, existing user-friendly preprocessors can be used in combination with non-regular and non-structured meshes. This is the main reason for the use of non-structured meshes in this thesis. Another consequence of using a non-structured mesh is that it allows for local mesh-refinement. This is applied in this thesis for reason of achieving good accuracy in zones of intense deformations.

4.3 Initialization of particles

In this section, we discuss the initialization of particles within the background mesh. This includes association of mass, body forces, tractions and other properties of the continuum to particles. Let us consider a single tetrahedral element to explain the full procedure of initialization of particles information. Each particle is initially positioned at a predefined local position inside the parent element, see Appendix E, and hence the local position vector $\boldsymbol{\xi}_p$ of particle p is initialized. The global position vector $\boldsymbol{x}_p$ is then obtained as

$$\boldsymbol{x}\left(\boldsymbol{\xi}_p\right) \approx \sum_{i=1}^{n_{en}} N_i\left(\boldsymbol{\xi}_p\right) \boldsymbol{x}_i, \tag{4.1}$$

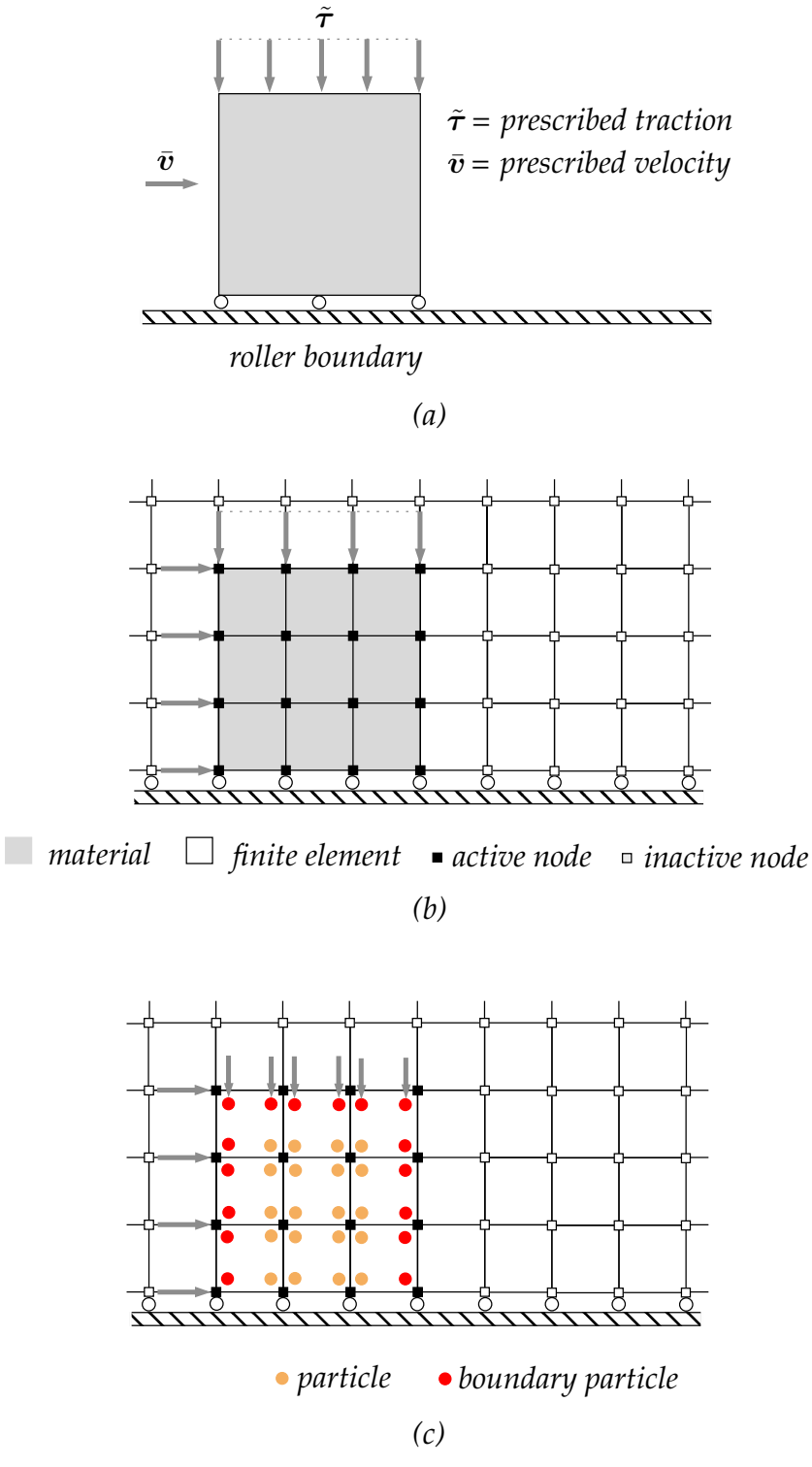


Figure 4.2: MPM discretization (a) continuum (b) Eulerian mesh with active and inactive elements (c) particles are initialized in the initially active elements

in which n_{en} denotes the number of nodes per element and $N_i\left(\boldsymbol{\xi}_p\right)$ is the shape function of node i evaluated at the local position of particle p. Volumes associated with particles are calculated so that all particles inside the element have initially the same portion of the element volume, i.e.,

$$\Omega_p = \frac{1}{n_{ep}} \int_{\Omega_e} d\Omega \approx \frac{1}{n_{ep}} \sum_{q=1}^{n_{eq}} w_q \left|\boldsymbol{J}\left(\boldsymbol{\xi}_q\right)\right|, \tag{4.2}$$

where Ω_p is the volume associated with particle p, n_{ep} denotes the number of particles in the element, n_{eq} is the number of Gauss points in the element, w_q is the local integration weight associated with Gauss point q, and $\boldsymbol{J}$ is the Jacobian matrix. The mass m_p is then calculated as

$$m_p = \Omega_p \, \varrho_p \tag{4.3}$$

with ϱ_p being the mass density of the material to which particle p belongs. The gravity force $\boldsymbol{f}_p^{grav}$ is simply calculated using the mass of the particle and the vector of gravitational acceleration $\boldsymbol{g}$ as

$$\boldsymbol{f}_p^{grav} = m_p \, \boldsymbol{g}. \tag{4.4}$$

The prescribed traction of the element is integrated first in the classical way of FEM. The following equation explains the integration of a prescribed surface traction $\tilde{\boldsymbol{\tau}}_e$ over the triangular surface element e shown in Figure 4.3

$$\begin{aligned} \tilde{\boldsymbol{F}}_e^{trac}\left(\boldsymbol{x}\right) &= \int_{S_e} \boldsymbol{N}^T\left(\boldsymbol{\xi}\right) \tilde{\boldsymbol{\tau}}_e\left(\boldsymbol{x}\right) dS \\ &\approx \sum_{q=1}^{n_{seq}} w_q \, \boldsymbol{N}^T\left(\boldsymbol{\xi}_q\right) \left(\sum_{i=1}^{n_{tri}} N_i\left(\boldsymbol{\xi}_q\right) \tilde{\boldsymbol{\tau}}_e\left(\boldsymbol{x}_i\right) \right) S_e. \end{aligned} \tag{4.5}$$

One should realize that $\boldsymbol{N}$ in the above equation is a matrix containing the shape functions of the surface element where traction is prescribed. For the 4-noded tetrahedral element, the surface traction is prescribed on a 3-noded triangular element, (see also Appendix C). $\tilde{\boldsymbol{\tau}}_e\left(\boldsymbol{x}_i\right)$ denotes the traction vector at node i of the surface element e, S_e is the surface area of the triangular element, n_{seq} indicates the number of Gauss points per triangular element, n_{tri} represents the number of nodes in the triangular element, w_q and $\boldsymbol{\xi}_q$ are the local integration weight and the local coordinate vector of Gauss point q in two-dimensions, respectively. The resulting traction forces vector $\tilde{\boldsymbol{F}}_e^{trac}$ from the integration of Equation 4.5 has 9 components, each 3 corresponding to one node of the 3-noded triangular element. On denoting the traction force vector of each node of the triangular element as $\tilde{\boldsymbol{f}}^{trac}$, one can write

$$\tilde{\boldsymbol{F}}_e^{trac} = \begin{bmatrix} \tilde{\boldsymbol{f}}_1^{trac} & \tilde{\boldsymbol{f}}_2^{trac} & \tilde{\boldsymbol{f}}_3^{trac} \end{bmatrix}^T, \tag{4.6}$$

in which each vector $\tilde{\boldsymbol{f}}$ consists of three components.

The traction vector $\tilde{\boldsymbol{\tau}}_e$ applied at the triangular surface e is interpolated from the nodes

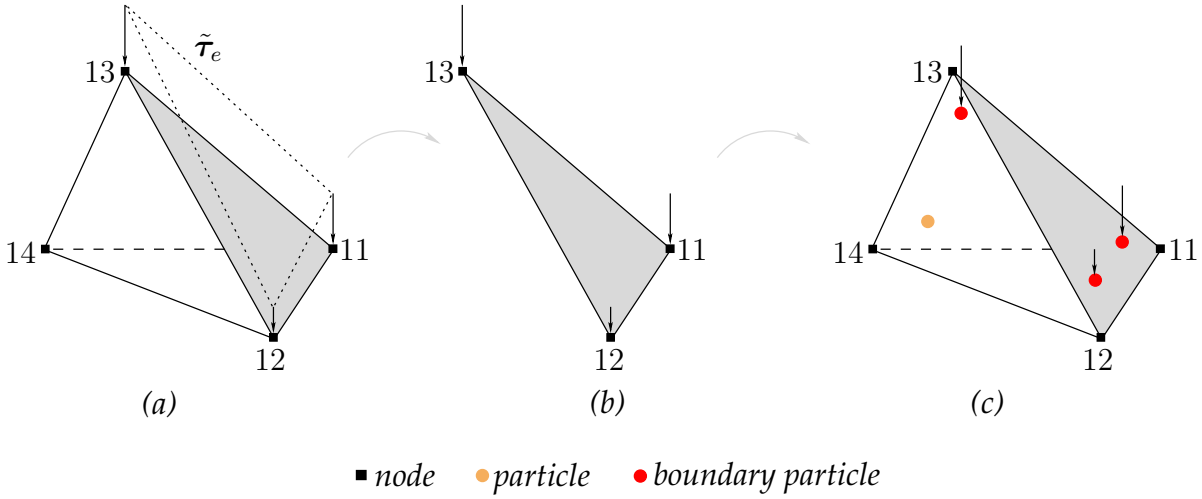


Figure 4.3: Initialization of surface traction in MPM (a) tetrahedral element (b) triangular element (c) traction is mapped to the boundary particles

of this surface to the boundary particles located next to it. Hence, the traction at boundary particle p is

$$\tilde{\boldsymbol{\tau}}_e\left(\boldsymbol{x}_p\right) \approx \sum_{i=1}^{n_{tri}} N_i\left(\boldsymbol{\xi}_p\right) \tilde{\boldsymbol{\tau}}_e\left(\boldsymbol{x}_i\right), \tag{4.7}$$

where N_i is the shape function of node i of the triangular surface element and $\boldsymbol{\xi}_p$ are the coordinates of the boundary particle p inside the parent triangular element. These coordinates simply represent the projection of particle p on the triangular surface element. The traction force vector $\tilde{\boldsymbol{f}}_p^{trac}$ of boundary particle p is then

$$\begin{aligned} \tilde{\boldsymbol{f}}_p^{trac} &= \tilde{\boldsymbol{\tau}}_e\left(\boldsymbol{x}_p\right) \frac{S_e}{n_{ebp}} \\ &= \frac{S_e}{n_{ebp}} \sum_{i=1}^{n_{tri}} N_i\left(\boldsymbol{\xi}_p\right) \tilde{\boldsymbol{\tau}}_e\left(\boldsymbol{x}_i\right), \end{aligned} \tag{4.8}$$

in which n_{ebp} denotes the number of boundary particles located next to the loaded surface.

In this thesis, only uniform surface tractions are considered and the boundary particles are initially positioned so that they are evenly distributed with respect to the loaded surface. This implies that

$$\tilde{\boldsymbol{f}}_p^{trac} = \tilde{\boldsymbol{\tau}}_e \frac{S_e}{n_{ebp}}, \tag{4.9}$$

which is equivalent to

$$\tilde{\boldsymbol{f}}_p^{trac} = \frac{1}{n_{ebp}} \sum_{i=1}^{n_{tri}} \tilde{\boldsymbol{f}}_i^{trac}. \tag{4.10}$$

In the initialization of particles, initial conditions, material parameters and constitutive variables are assigned to them as well. Furthermore, book-keeping is initialized at this step, including information such as to which element each particle initially belongs and the initial number of particles per each active finite element.

4.4 MPM solution procedure

Chapter 3 details the momentum equation including the space and time discretization of the weak form. The gathering and assemblage procedure of the discrete algebraic equations of motion in the framework of Lagrangian FEM is also presented in this chapter. The weak form and the discrete equations of motion in the context of MPM are consistent with those of the finite element method [165, 166]. The main difference is in the way the weak form of the momentum equation is integrated in space. In FEM, standard Gauss integration is commonly adopted in the space integration as explained in Chapter 3 and Appendix C, whereas in classical MPM, the space integration is performed using particles as integration points instead of Gauss points. This is, of course, a natural consequence of the fact that in the context of MPM, the continuum is discretized using a finite number of particles. We will not re-consider the full procedure of the weak formulation and space discretization in this chapter, but rather focus on the procedure of integrating and constructing the equations of motion in the framework of MPM and the algorithm of solving thses equations within a time step.

4.4.1 Initialization of equations of motion

As mentioned previously, in MPM particles carry all physical information of the continuum throughout the computations. Hence, in order to solve the discrete equations of motion in Lagrangian frame, these equations must be initialized and solved at the grid points. Thus information must be mapped from particles to the grid points temporarily. The lumped-mass matrix is formed by the direct lumping procedure of Chapter 3. The lumping is carried out using particle masses and the shape functions to construct the elements mass matrices, which are gathered in the global mass matrix via the assemblage procedure. On considering the state of the continuum at time t, the lumped-mass matrix is written as

$$\boldsymbol{M}^t = \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \boldsymbol{M}_e^t, \tag{4.11}$$

where $\bar{n}_{elm}$ represents the number of active finite elements in the mesh at time t and $\boldsymbol{M}_e^t$ is defined as

$$\boldsymbol{M}_e^t = \begin{bmatrix} \boldsymbol{m}_1^t & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{m}_2^t & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{m}_{en}^t \end{bmatrix}. \tag{4.12}$$

The matrix $\boldsymbol{m}_i^t$ corresponds to the mass of node i and $\mathbf{0}$ is a null matrix. In three-dimensions, they are defined as follows

$$\boldsymbol{m}_i^t = \begin{bmatrix} m_i^t & 0 & 0 \\ 0 & m_i^t & 0 \\ 0 & 0 & m_i^t \end{bmatrix} \quad \text{and} \quad \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.13}$$

with

$$m_i^t \approx \sum_{p=1}^{n_{ep}} m_p \, N_i \left(\boldsymbol{\xi}_p^t \right). \tag{4.14}$$

In the initialization procedure, velocities are mapped from particles to the nodes of the background mesh. A weighted mapping is used to determine nodal velocities from particle velocities. The weighting is the mass of particles. Hence, the procedure is equivalent to mapping the momentum of particles to the nodes and then calculating the nodal velocities from the mapped momentum using the nodal masses. On denoting the global vector (which contains the mapped momentum from particles to nodes) as $\boldsymbol{P}$, the nodal velocities vector $\boldsymbol{v}$ is calculated from

$$\boldsymbol{M}^t \boldsymbol{v}^t = \boldsymbol{P}^t, \tag{4.15}$$

in which

$$\boldsymbol{P}^t \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} m_p \, \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t \right) \hat{\boldsymbol{v}}_p^t, \tag{4.16}$$

where $\hat{\boldsymbol{v}}_p^t$ is the velocity of particle p at time t. It was shown by Burgess et al. [41] that this procedure of mapping velocities from particles to nodes is conservative when a consistent-mass matrix is used. A lumped-mass matrix is however associated with slight dissipation of the kinetic energy [41].

The nodal traction vector is constructed by mapping the tractions from boundary particles to the nodes as

$$\tilde{\boldsymbol{F}}^{trac,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{\tau elm}} \sum_{p=1}^{n_{ebp}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t \right) \tilde{\boldsymbol{f}}_p^{trac}, \tag{4.17}$$

then

$$\boldsymbol{F}^{trac,t} = \tilde{\boldsymbol{F}}^{trac,t} \, \mathcal{T}\left(t\right) \tag{4.18}$$

with $\mathcal{T}\left(t\right)$ being the time multiplier of the traction and $\bar{n}_{\tau elm}$ is the number of elements that contain traction-carrying particles. This number will generally vary in time during computations. The nodal vector of body forces due to the weight is integrated as follows

$$\boldsymbol{F}^{grav,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t \right) \boldsymbol{f}_p^{grav} \tag{4.19}$$

and the internal forces due to stresses are following as

$$\boldsymbol{F}^{int,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{B}^T \left(\boldsymbol{\xi}_p^t \right) \boldsymbol{\sigma}_p^t \, \Omega_p^t. \tag{4.20}$$

Hence, the resulting discrete equations of motion are

$$\begin{aligned} \boldsymbol{M}^t \boldsymbol{a}^t &= \boldsymbol{F}^{trac,t} + \boldsymbol{F}^{grav,t} - \boldsymbol{F}^{int,t} \\ &= \boldsymbol{F}^t, \end{aligned} \tag{4.21}$$

in which the symbol $\boldsymbol{a}$ denotes the unknown vector of nodal accelerations. Equation 4.21 has a form similar to that obtained when considering Lagrangian FEM, see Equation 3.67. However, the difference is as explained before in the way of constructing those equations. One should keep in mind that, unlike Lagrangian FEM, the number of the active degrees-of-freedom is changing in MPM as particles move through the mesh and hence the mass matrix changes with time, requiring computation of new matrix at the beginning of each time step.

4.4.2 Solving the equations of motion

The algorithm of solving Equation 4.21 in the context of MPM to advance the state of the continuum from time t to $t+\Delta t$ is presented in this section. As MPM equations of motion are casted in a way that is consistent to that of Lagrangian FEM, the early MPM solution algorithm within the time step is identical to that of Lagrangian FEM [165]. In this thesis, this algorithm is called *identical Lagrangian algorithm*. However, due to some shortcomings of this algorithm, a slightly modified algorithm was proposed by Sulsky et al. [166] and has subsequently been used in most MPM literature. We refer to this algorithm as *modified Lagrangian algorithm*. In the following both algorithms are presented.

4.4.2.1 Identical Lagrangian algorithm

In this algorithm, Equation 4.21 is solved for the unknown nodal acceleration vector $\boldsymbol{a}$ as

$$\boldsymbol{a}^t = \boldsymbol{M}^{t,-1} \boldsymbol{F}^t. \tag{4.22}$$

The nodal velocity vector then follows from the time integration of the acceleration vector

$$\boldsymbol{v}^{t+\Delta t} = \boldsymbol{v}^t + \Delta t\, \boldsymbol{a}^t. \tag{4.23}$$

The vector $\boldsymbol{v}^t$ is already calculated in the initialization of the equations of motion, see Equation 4.15. The velocity vector $\boldsymbol{v}^{t+\Delta t}$ is used to determine the vector of incremental displacement at the nodes

$$\Delta \boldsymbol{u}^{t+\Delta t} = \Delta t\, \boldsymbol{v}^{t+\Delta t}. \tag{4.24}$$

Nodal velocities and displacements are used to update particles velocities, displacements and positions as follows

$$\begin{aligned}
\hat{\boldsymbol{v}}_p^{t+\Delta t} &= \hat{\boldsymbol{v}}_p^t + \sum_{i=1}^{n_{en}} \Delta t \, N_i \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{a}_i^t, \\
\hat{\boldsymbol{u}}_p^{t+\Delta t} &= \hat{\boldsymbol{u}}_p^t + \sum_{i=1}^{n_{en}} N_i \left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_i^{t+\Delta t}, \\
\boldsymbol{x}_p^{t+\Delta t} &= \boldsymbol{x}_p^t + \sum_{i=1}^{n_{en}} N_i \left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_i^{t+\Delta t}.
\end{aligned} \tag{4.25}$$

The summation in the above equations runs over the nodes of the element where particle p is located at time t. Strains and stresses at particles are updated using the same procedure as for Gauss points with the standard FEM, see Section 3.4.2. However, the mesh in most cases is not updated as usually done with the updated Lagrangian FEM. Hence, only the location of particles is updated according to Equation 4.25. Particles eventually cross element boundaries, which entails that the new element of a crossing particle has to be detected.

With particles moving through the mesh, special attention must be paid to the case that a particle hardly crosses a previously empty element. Thus this particle is located very close to the element boundary, a scenario illustrated in Figure 4.4. Consider, for example, the single particle in Figure 4.4 and following Equation 4.14, the mass of node 2 is $m_2 = m_p \, N_2 \left(\xi_p\right)$, in which m_p is the mass of particle p and $N_2 \left(\xi_p\right)$ is the shape function of node 2 evaluated at the position of particle p. The value of this shape function will

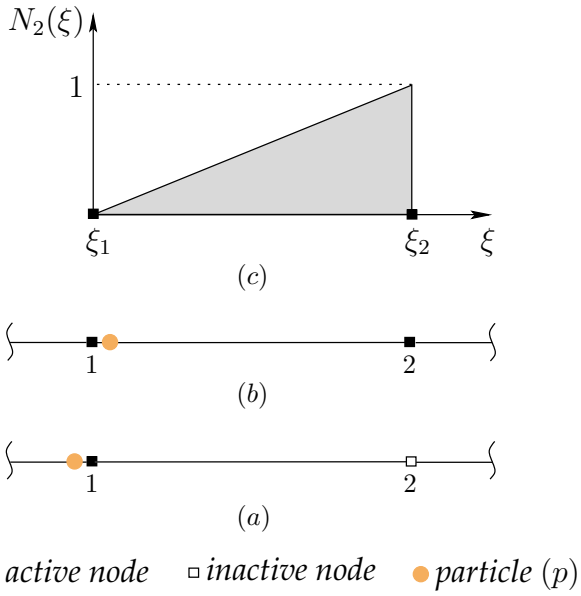


Figure 4.4: Particle hardly crossing element boundary (a) before crossing (b) after crossing (c) the shape function of node 2 across the element length

thus approach zero as the particle just crosses the element boundary. As a consequence, the mass m_2 approaches zero, yielding a nearly singular, ill-conditioned mass matrix.

The internal force involves the gradient of the shape function via the matrix $\boldsymbol{B}$, see Equation 4.20. The gradient of a linear shape function is constant in the element and hence the internal force of node 2 does not approach zero. The consequence of this case is that the acceleration at node 2 is non-physical when computed by dividing the force by the mass of this node, which ultimately leads to particle separating from the rest of the body [166].

A cutoff value for the small mass can be introduced to detect small nodal masses. If a nodal mass is less than a predefined cutoff value, the internal forces corresponding to this node are then set to zero. However, this procedure does not solve the problem totally as it is difficult to choose a cutoff value [166]. Furthermore, from the author's experience, a consequence of such a procedure is that the algorithm inadvertently treats the small mass node as a fixed node, introducing a constraint that does not exist in the system. The problem of numerical instability and the non-physical accelerations can also be mitigated by using tiny time step, but this is still not a practical solution as the computational efficiency is of considerable importance.

4.4.2.2 Modified Lagrangian algorithm

As a remedy of the small mass dilemma in MPM, Sulsky et al. [166] introduced a slightly modified algorithm. In this algorithm, Equation 4.21 is solved for the unknown nodal acceleration vector $\boldsymbol{a}$ as

$$\boldsymbol{a}^t = \boldsymbol{M}^{t,-1}\boldsymbol{F}^t. \tag{4.26}$$

The above step is identical to the first step of the identical Lagrangian algorithm. The computed vector of nodal accelerations can still have some non-physical component if a small nodal mass occurs. However, the next step eliminates the problem on the level of velocity. This step consists of the calculation of particle velocities directly from the nodal accelerations as

$$\hat{\boldsymbol{v}}_p^{t+\Delta t} = \hat{\boldsymbol{v}}_p^t + \sum_{i=1}^{n_{en}} \Delta t\, N_i\left(\boldsymbol{\xi}_p^t\right) \boldsymbol{a}_i^t. \tag{4.27}$$

The key point in the modified Lagrangian algorithm is that the nodal velocities are not directly integrated from the nodal accelerations as in the identical Lagrangian algorithm, but they are calculated from the updated particles velocities as

$$\boldsymbol{M}^t \boldsymbol{v}^{t+\Delta t} = \boldsymbol{P}^{t+\Delta t}, \tag{4.28}$$

in which

$$\boldsymbol{P}^{t+\Delta t} \approx \mathop{\mathbb{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} m_p\, \boldsymbol{N}^T\left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{v}}_p^{t+\Delta t}. \tag{4.29}$$

Nodal incremental displacements follow by integrating the nodal velocities as

$$\Delta \boldsymbol{u}^{t+\Delta t} = \Delta t\, \boldsymbol{v}^{t+\Delta t}. \tag{4.30}$$

Particles displacements and positions are updated by interpolating nodal displacements using the shape functions as

$$\begin{aligned}\hat{\boldsymbol{u}}_p^{t+\Delta t} &= \hat{\boldsymbol{u}}_p^t + \sum_{i=1}^{n_{en}} N_i\left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_i^{t+\Delta t},\\ \boldsymbol{x}_p^{t+\Delta t} &= \boldsymbol{x}_p^t + \sum_{i=1}^{n_{en}} N_i\left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_i^{t+\Delta t}.\end{aligned} \tag{4.31}$$

Let us for the moment apply this algorithm to the one-dimensional case of Figure 4.4. The accelerations at node 1 and 2 are calculated according to Equation 4.26 as

$$a_1^t = \frac{f_1^t}{m_1^t} \qquad \text{and} \qquad a_2^t = \frac{f_2^t}{m_2^t}, \tag{4.32}$$

where

$$m_1^t = m_p\, N_1\left(\xi_p^t\right) \qquad \text{and} \qquad m_2^t = m_p\, N_2\left(\xi_p^t\right) \tag{4.33}$$

according to Equation 4.14. These nodal accelerations are used to update the velocity of particle p according to Equation 4.27. Hence,

$$\begin{aligned}\hat{v}_p^{t+\Delta t} &= \hat{v}_p^t + \Delta t \left(N_1\left(\xi_p^t\right) a_1^t + N_2\left(\xi_p^t\right) a_2^t\right)\\ &= \hat{v}_p^t + \Delta t \left(\frac{N_1\left(\xi_p^t\right) f_1^t}{m_1^t} + \frac{N_2\left(\xi_p^t\right) f_2^t}{m_2^t}\right)\\ &= \hat{v}_p^t + \Delta t \left(\frac{N_1\left(\xi_p^t\right) f_1^t}{m_p\, N_1\left(\xi_p^t\right)} + \frac{N_2\left(\xi_p^t\right) f_2^t}{m_p\, N_2\left(\xi_p^t\right)}\right)\\ &= \hat{v}_p^t + \Delta t \left(\frac{f_1^t}{m_p} + \frac{f_2^t}{m_p}\right),\end{aligned} \tag{4.34}$$

in which the effect of a small mass is eliminated from the denominator by the corresponding shape function in the nominator. This obviously solves the problem of small masses.

Let us continue the solution by assuming $\hat{v}_p^t = 0$ for simplicity. Now the velocity of particle p is used to calculate the nodal velocities according to Equation 4.28. Hence,

$$\begin{bmatrix} m_p\, N_1\left(\xi_p^t\right) & 0 \\ 0 & m_p\, N_2\left(\xi_p^t\right) \end{bmatrix} \begin{bmatrix} v_1^{t+\Delta t} \\ v_2^{t+\Delta t} \end{bmatrix} = \begin{bmatrix} \Delta t \left(f_1^t + f_2^t\right) N_1\left(\xi_p^t\right) \\ \Delta t \left(f_1^t + f_2^t\right) N_2\left(\xi_p^t\right) \end{bmatrix}, \tag{4.35}$$

which gives

$$\begin{bmatrix} v_1^{t+\Delta t} \\ v_2^{t+\Delta t} \end{bmatrix} = \Delta t \begin{bmatrix} \frac{f_1^t + f_2^t}{m_p} \\ \frac{f_1^t + f_2^t}{m_p} \end{bmatrix}. \tag{4.36}$$

However, solving the same problem using the identical Lagrangian algorithm yields the following equation of motion

$$\begin{bmatrix} m_p\, N_1\left(\xi_p^t\right) & 0 \\ 0 & m_p\, N_2\left(\xi_p^t\right) \end{bmatrix} \begin{bmatrix} a_1^t \\ a_2^t \end{bmatrix} = \begin{bmatrix} f_1^t \\ f_2^t \end{bmatrix} \tag{4.37}$$

and the acceleration is then

$$\begin{bmatrix} a_1^t \\ a_2^t \end{bmatrix} = \begin{bmatrix} \frac{f_1^t}{m_p N_1\left(\xi_p^t\right)} \\ \frac{f_2^t}{m_p N_2\left(\xi_p^t\right)} \end{bmatrix}, \tag{4.38}$$

which results in the following velocity after integration and assuming that $v_1^t = v_2^t = 0$ as in the previous case

$$\begin{bmatrix} v_1^{t+\Delta t} \\ v_2^{t+\Delta t} \end{bmatrix} = \Delta t \begin{bmatrix} \frac{f_1^t}{m_p N_1\left(\xi_p^t\right)} \\ \frac{f_2^t}{m_p N_2\left(\xi_p^t\right)} \end{bmatrix}. \tag{4.39}$$

Comparing the right-hand side of Equations 4.36 to that of Equations 4.39, we can see that the effect of the shape functions is eliminated in the solution corresponding to the modified Lagrangian algorithm and hence the problem of small nodal masses is solved. It should be clarified that the accelerations computed by the modified Lagrangian algorithm are smoothed, see the right-hand side of Equations 4.36. A consequence of such smoothening is that the algorithm slowly dissipates energy with time [126]. Nevertheless, this algorithm improves the numerical stability and eliminates the problem of small mass in MPM.

We can find in literature alternatives to improve the numerical stability of MPM. For instance, Bardenhagen [6] proposed the procedure, in which strains and stresses are updated before updating the momentum, i.e., at the beginning of the time step. However, Nairn [126] reports that the modified Lagrangian algorithm and Bardenhagen's algorithm are almost identical. The only mathematical difference between them is that the modified Lagrangian algorithm uses the positions of particles at the end of the time step, i.e., before updating particles positions, to calculate the shape functions that in turn will be used to extrapolate the momentum from particles to nodes and eventually to calculate the nodal velocities. Bardenhagen's algorithm uses the new position of particles, i.e., after resetting the mesh, to evaluate the shape functions and extrapolate the same momentum from particles to nodes. It is also observed that Bardenhagen's algorithm tends to slowly increase energy with time [126], a feature that is not desired in numerical procedures. Nairn [126] proposed a procedure, in which the strain increment is averaged from the one calculated by modified Lagrangian algorithm and that of Bardenhagen's algorithm. He reported in his paper that this approach conserves energy nearly exactly.

Another approach to mitigate the small mass problem was proposed by Ma et al. [113]. In their approach, instead of setting the forces of small mass nodes to zero as in the early development of MPM, they distribute these forces to surrounding nodes. The reader is however referred to the aforementioned literatures and the references therein for further information about MPM solution algorithms.

In the summary of this review, we would like to point out that the most common solution algorithm in MPM literature is the one of Sulsky et al. [166], i.e., the modified Lagrangian algorithm, which is also adopted in this thesis.

4.4.2.3 Overall solution algorithm for a single time step

We summarize here the solution procedure of the modified Lagrangian algorithm. Let us consider the state of a continuum at time t and describe the procedure of advancing the solution to time $t + \Delta t$. It consists of the following steps

1. The lumped-mass matrix $\boldsymbol{M}^t$ is computed at the beginning of the time step. This is done as
$$\boldsymbol{M}^t = \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \boldsymbol{M}_e^t,$$
where $\boldsymbol{M}_e^t$ is integrated following the procedure described in Equations 4.12-4.14.

2. The momentum is mapped from particles to nodes using the shape functions and is then used to calculate the nodal velocity vector $\boldsymbol{v}^t$ solving the following equation
$$\boldsymbol{M}^t \boldsymbol{v}^t \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} m_p \, \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{v}}_p^t.$$

3. The traction force vector $\boldsymbol{F}^{trac,t}$ follows as
$$\boldsymbol{F}^{trac,t} = \tilde{\boldsymbol{F}}^{trac,t} \, \mathcal{T}\left(t\right), \qquad \text{in which} \qquad \tilde{\boldsymbol{F}}^{trac,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{\tau elm}} \sum_{p=1}^{n_{ebp}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \tilde{\boldsymbol{f}}_p^{trac}.$$

4. The gravity forces $\boldsymbol{F}^{grav,t}$ and the internal forces $\boldsymbol{F}^{int,t}$ are integrated as
$$\boldsymbol{F}^{grav,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{f}_p^{grav} \qquad \text{and} \qquad \boldsymbol{F}^{int,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{B}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{\sigma}_p^t \, \Omega_p^t.$$

5. At this stage, the discrete system of equations is complete
$$\begin{aligned} \boldsymbol{M}^t \boldsymbol{a}^t &= \boldsymbol{F}^{trac,t} + \boldsymbol{F}^{grav,t} - \boldsymbol{F}^{int,t} \\ &= \boldsymbol{F}^t. \end{aligned}$$

6. The system is now solved for the nodal accelerations as
$$\boldsymbol{a}^t = \boldsymbol{M}^{t,-1} \boldsymbol{F}^t.$$

7. In this step, the velocities of particles are updated using the nodal accelerations and the shape functions
$$\hat{\boldsymbol{v}}_p^{t+\Delta t} = \hat{\boldsymbol{v}}_p^t + \sum_{i=1}^{n_{en}} \Delta t \, N_i \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{a}_i^t.$$

8. The nodal velocities $\boldsymbol{v}^{t+\Delta t}$ are then calculated from the updated particles velocities solving the following equation

$$\boldsymbol{M}^t \boldsymbol{v}^{t+\Delta t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} m_p \, \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{v}}_p^{t+\Delta t}.$$

9. Nodal velocities are integrated to get nodal incremental displacements

$$\Delta \boldsymbol{u}^{t+\Delta t} = \Delta t \, \boldsymbol{v}^{t+\Delta t}.$$

10. Strains and stresses at particles are calculated as

$$\Delta \boldsymbol{\varepsilon}_p^{t+\Delta t} = \boldsymbol{B}\left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_e^{t+\Delta t},$$

$$\{\boldsymbol{\sigma}_p, \text{material state}\}^t \xrightarrow[\Delta \boldsymbol{\varepsilon}_p^{t+\Delta t}]{\text{constitutive relation}} \{\boldsymbol{\sigma}_p, \text{material state}\}^{t+\Delta t}.$$

See Equation 3.108 for the definition of the nodal incremental displacement vector $\Delta \boldsymbol{u}_e$.

11. Volumes associated with particles are updated using the volumetric strain increment $\Delta \varepsilon_{vol}$

$$\Omega_p^{t+\Delta t} = \left(1 + \Delta \varepsilon_{vol,p}^{t+\Delta t}\right) \Omega_p^t \qquad \text{with} \qquad \Delta \varepsilon_{vol} = \Delta \varepsilon_{11} + \Delta \varepsilon_{22} + \Delta \varepsilon_{33}.$$

Consequently, the mass density is updated as follows

$$\varrho_p^{t+\Delta t} = \frac{\varrho_p^t}{\left(1 + \Delta \varepsilon_{vol,p}^{t+\Delta t}\right)} \tag{4.40}$$

12. Displacements and positions of particles are updated according to

$$\hat{\boldsymbol{u}}_p^{t+\Delta t} = \hat{\boldsymbol{u}}_p^t + \sum_{i=1}^{n_{en}} N_i\left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_i^{t+\Delta t},$$

$$\boldsymbol{x}_p^{t+\Delta t} = \boldsymbol{x}_p^t + \sum_{i=1}^{n_{en}} N_i\left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_i^{t+\Delta t}.$$

13. The book-keeping is updated at this step, including the detection of the new elements for those particles that have left the old elements. Consequently the new number of particles per each finite element must be determined. The active and inactive elements must be identified and the new local position of each particle inside the element must be obtained.

It can be observed that the nodal velocity vector $\boldsymbol{v}^t$ calculated in Step 2 is not explicitly used in the algorithm. However, we need this vector in the calculation of the local damping, which will be explained later in this chapter.

4.5 Boundary conditions in MPM

Within the framework of traditional Lagrangian finite element, the application of prescribed boundary conditions is trivial as they are specified more or less directly for the degree-of-freedom on the boundary nodes and that the boundary nodes are well defined throughout the computations. In the material point method, the computational mesh does not align with the boundary, making the application of the prescribed boundary conditions not straightforward. The difficulty arises, in particular, when dealing with nonzero boundary conditions, including nonzero tractions and nonzero kinematic boundary conditions. The discussion is split into two sub-sections, the first is dealing with zero prescribed boundary conditions, whereas the second is treating the application of nonzero boundary conditions in the framework of MPM.

4.5.1 Zero kinematic and traction boundary conditions

Such boundary conditions are easy to apply in MPM. Zero kinematic boundary conditions are applied in a way that is identical to that of standard Lagrangian FEM. Special attention should however be paid when applying those conditions in the framework of MPM, that is they also have to be applied on those boundary nodes that might become active during computations. This ensures that, when a previously inactive node becomes active during computations, zero prescribed kinematic boundary conditions at this node must be applied (activated). Figure 4.5 illustrates such a situation. Regarding zero traction boundary conditions, they are automatically enforced to be satisfied by the solution of equations of motion.

4.5.2 Nonzero kinematic and traction boundary conditions

Questions arise on how to deal with nonzero traction boundary conditions. The treatment of surface tractions in the context of MPM deviates strongly from that of Lagrangian FEM. A way to treat such tractions in MPM is to map them from boundary particles to all nodes of the element where a boundary particle is located, see Figure 4.6. However,

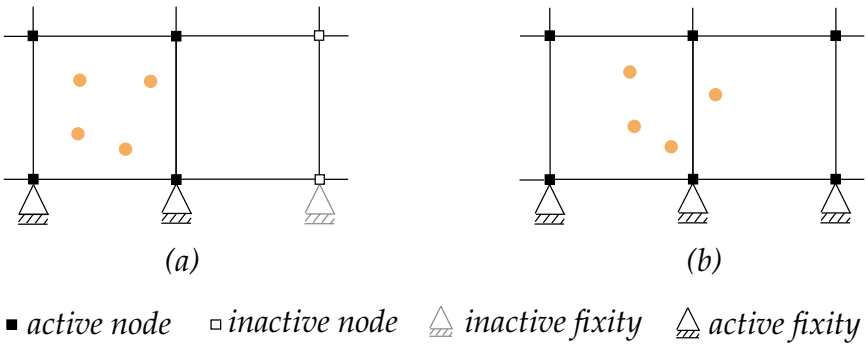


Figure 4.5: Kinematic boundary conditions are prescribed at the active and inactive nodes (a) inactive fixity (b) active fixity

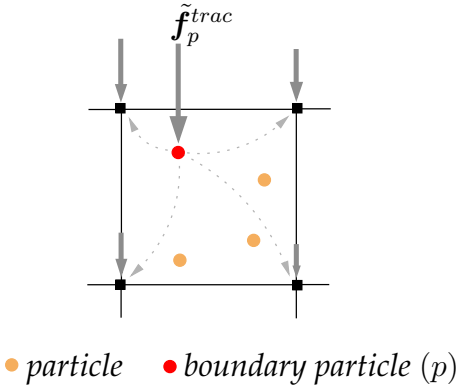


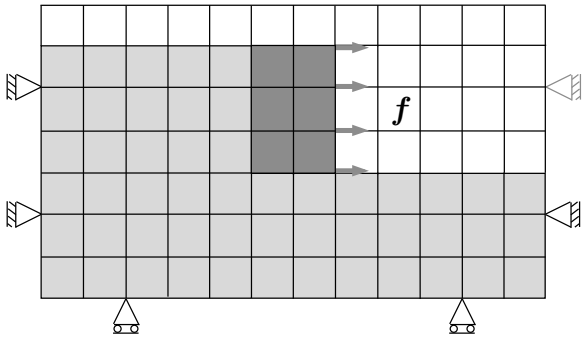
Figure 4.6: Traction is mapped from a boundary particle to all nodes of the element

such mapping procedure is carried out according to Equation 4.17. One should remember here that in the initialization of particles, traction forces are mapped to the boundary particles, see Section 4.3. The disadvantage of this way of dealing with surface tractions is that the effect of the surface force is distributed over the layer of elements that borders the boundary. To reduce such smearing error, the thickness of the elements along the boundary should be small. However, this can be helpful in some cases, but it has to be admitted that it is not a practical solution to the problem. To apply the nonzero boundary conditions in a more consistent way with that of Lagrangian FEM, we introduce the concept of *moving mesh*.

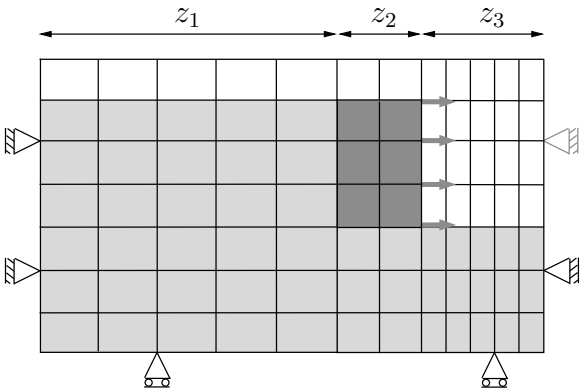
4.5.2.1 Concept of moving mesh

It is a procedure, in which the surface tractions and kinematic boundary conditions are applied in a way that is consistent with that of Lagrangian FEM and hence considered to be accurate. In the *moving mesh* procedure, we always ensure that the computational mesh aligns with the surface where tractions are prescribed. To illustrate this procedure, let us consider a rigid block being displaced by the action of surface tractions, see Figure 4.7. We deviate, in this procedure, from the standard MPM by using a mesh which is fixed to the rigid block, indicated as moving mesh in Figure 4.7. Thereby, the traction is kept at the same boundary nodes and never mapped between particles and nodes. This way of treating the surface tractions is identical to that of Lagrangian FEM. A consequence of such procedure, as shown in Figure 4.7, is that the mesh in front of the block (right) gets compressed and that behind the block (left) gets stretched with time by the same displacement of the rigid block.

For extremely large deformations the mesh can get distorted, a problem that could be solved in this case by meshing a wider regions in the stretched and compressed zones or if necessary re-meshing these zones to avoid mesh distortion, making use of the MPM feature that the mesh could be arbitrarily modified after each time step. An advantage of the moving mesh procedure is that the moving mesh zone could be made fine and hence we ensure that this fine mesh will always remain around the block. With standard MPM, the block would move through a fixed mesh, requiring mesh refinement over the



(a)



(b)

z_1 *stretched mesh* z_2 *moving mesh* z_3 *compressed mesh*

Figure 4.7: Illustration of the moving mesh procedure (a) initial configuration (b) deformed configuration with stretched, compressed and moving zones

complete region where the block is expected to displace. It remains to mention here that the moving mesh concept could also be used to apply nonzero kinematic boundary conditions. One can also find in literature that the nonzero boundary conditions could be applied using what is so-called boundary layer method. The reader is however referred to the work of Hu and Chen [92] and references therein for more details about this method.

4.6 Mitigating error associated with grid-crossing for quasi-static problems

When dealing with quasi-static problems, in which the rate of loading is very slow and hence the effect of inertia is insignificant, a numerical dilemma could occur when particles cross element boundaries. The early description of such numerical difficulty or phenomenon in MPM can be tracked back to the work by Zhou [201]. In this section, we consider such numerical difficulty and discuss possible solutions for it in the framework of the adopted MPM algorithm. In quasi-static problems, equilibrium could break down and eventually computations fail, in some severe scenarios when particles cross elements boundaries. The reason of this problem is that, such particle crossing leads to significant unbalance forces at the nodes that are shared between previous and new elements of crossing particles. Eventually, such numerical difficulties are imposed by the linear shape functions. The gradient of a linear shape function is constant within the element, but discontinuous at the element edges. Hence, artifact unbalance forces are induced at elements boundaries when particles cross those boundaries. To explain the problem of grid-crossing in a mathematical way, let us consider the one-dimensional mesh of Figure 4.8. The mesh shows two elements, each having two particles initially. The two shape functions of the 2-noded element are

$$N_1(\xi) = 1 - \xi,$$
$$N_2(\xi) = \xi.$$

In Figure 4.8(c) and (d), the shape functions of node 2 and the corresponding gradient are plotted across elements length, respectively. However, node 2 represents the shared node between the two elements. The internal forces at this node are integrated according to Equation 4.20 as follows

$$f_2^{int} \approx \sum_{p=1}^{n_{ep1}} \sigma_p \Omega_p - \sum_{p=1}^{n_{ep2}} \sigma_p \Omega_p, \tag{4.41}$$

in which n_{ep1} and n_{ep2} are the number of particles in elements e_1 and e_2, respectively. Assuming a static equilibrium and uniform mesh, Equation 4.41 is then reduced to

$$f_2^{int} \approx \sigma \Omega \left(n_{ep1} - n_{ep2} \right), \tag{4.42}$$

where σ is the constant stress in both elements resulting from equilibrium and Ω is a constant volume associated with each particle resulting from the uniform mesh. Under these conditions, the internal force of node 2 is zero as long as $n_{ep1} = n_{ep2}$ and no crossing occurs, the case of Figure 4.8(a). Once a particle crosses from one element to the other, the equilibrium at node 2 breaks down as an artifact unbalance force is induced by the integration of the internal forces, a situation of Figure 4.8(b), in which $n_{ep1} = 1$ and $n_{ep2} = 3$ after a particle moved from element e_1 to element e_2. Hence, the internal force of node 2 is then $f_2^{int} = -2\sigma\Omega$ instead of zero, see Equation 4.42.

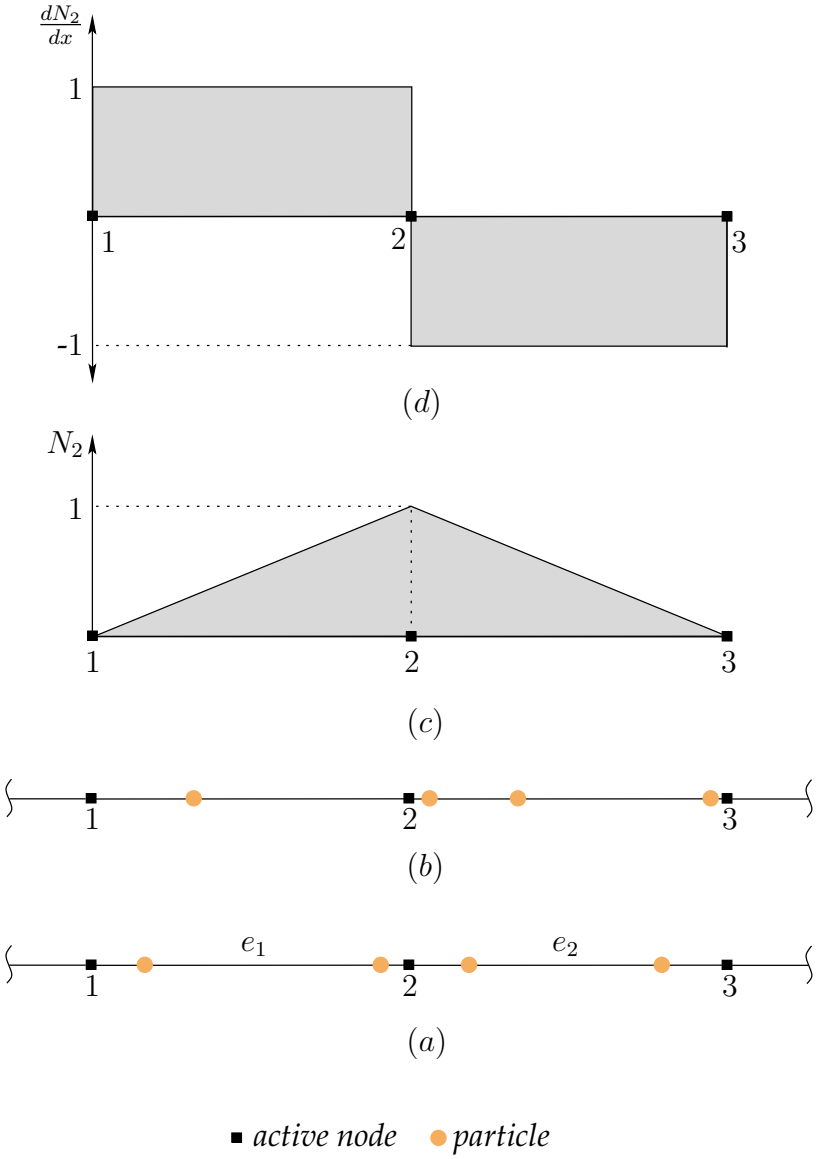


Figure 4.8: Grid-crossing in MPM (a) before crossing (b) after crossing (c) shape functions of node 2 (d) gradient of shape functions of node 2

4.6.1 Attempts to reduce the grid-crossing error

The problem is more severe when there are few particles in each finite element. On the other hand, using a fine mesh may cause the migration of particles between elements to be quicker and the problems is more pronounced then [74]. However, increasing the initial number of particle per element might be a partial solution to the problem, but again the computational cost can be very high for high number of particles. Guilkey et al. [74] were among the first to formulate a quasi-static material point method to simulate slow rate of loading. They deviated a little from the traditional MPM by not resetting the mesh after each time step. The reason for this was to avoid the error associated with particle crossing between elements. The locations of particles, in this approach, with respect to the nodes do not change as long as the mesh is not reset. However, this approach should be used with a great care as it might lead to a severe mesh distortion. A "by-product" of not resetting the mesh for a considerable deformation could be non-physical unbalance forces induced at the nodes after resetting the mesh.

Recently, Beuth [24] introduced what is so-called *zero load step* to eliminate these unbalance forces in the framework of quasi-static MPM. A zero load step is applied after resetting the mesh without applying any additional load increment. Only the nodal unbalance forces are reduced as in classical Lagrangian FEM. Particle positions are not updated after a zero load step and stresses obtained are assumed to be artifact and hence will be overwritten in the next load step [24].

Fernandes [64] investigated the performance of quadratic shape functions that are continuous across the element edge. Among other issues, he concluded that the negative regions of the quadratic shape functions can induce negative entries when integrating the mass matrix, resulting in a singular-mass matrix. This observation is however reported for both lumped and consistent matrices.

Bardenhagen and Kober [8] generalized the MPM discrete solution by applying a Petrov-Galerkin discretization scheme. They used the computational grid shape functions together with particle characteristic functions in the variational formulations. Different combinations of grid shape functions and particle characteristic functions result in a family of methods named, after Bardenhagen and Kober, as the *generalized interpolation material point method* and abbreviated as GIMP. This generalization allows the material point method to be derived as a special case from GIMP.

Unlike the traditional MPM [165], in which the mass of the particle represents the whole mass of the subregion surrounding this particle, in GIMP each particle only contributes to nodes whose shape functions overlap its characteristic function [8]. Hence, the interpolation could span multi elements, providing a smooth crossing of particles between elements. Their main motivation to investigate GIMP was to eliminate the numerical noise associated with MPM when particles cross elements boundaries. Indeed, they showed, in one-dimensional examples, that GIMP is capable of eliminating the noise in stresses obtained by MPM. The use of Petrov-Galerkin discretization scheme deviates the method more towards some meshless methods [8]. Furthermore, the method adds significant complexities to the MPM algorithm and its computational cost is considerably higher than that of MPM. Fernandes [64] reported a slow down factor of 2.68 in a particular numerical example. However, GIMP was used in several studies, see references

[2, 55, 111, 187]. Other attempts to reduce the problem of grid-crossing are proposed by other researchers see, for instance, [153, 201] and references therein.

4.6.2 Gauss integration in MPM

As means to solve the numerical difficulty of grid-crossing in quasi-static problems, we adopt Gauss integration in a way that is identical to that of Lagrangian FEM. In the latter, the integration points have fixed local positions inside the element. Such points move according to the deformation of the element, but never leave the element. Let us assume having a single Gauss point located anywhere inside each element of Figure 4.8. Considering the internal force of node 2, it is again

$$f_2^{int} \approx \sum_{p=1}^{n_{ep1}} \sigma_p \Omega_p - \sum_{p=1}^{n_{ep2}} \sigma_p \Omega_p.$$

If we replace the summation (integration) over particles by the summation over Gauss points, we get

$$f_2^{int} \approx \sum_{q=1}^{n_{eq}} w_q \sigma_q \left| J\left(\xi_q\right) \right| - \sum_{q=1}^{n_{eq}} w_q \sigma_q \left| J\left(\xi_q\right) \right|, \tag{4.43}$$

in which

$$\sigma_q = \frac{\sum_{p=1}^{n_{ep}} \sigma_p \Omega_p}{\sum_{p=1}^{n_{ep}} \Omega_p}. \tag{4.44}$$

Considering the assumed conditions that σ and Ω are the same for all particles, resulting from the equilibrium and the uniform mesh assumption, it yields that the internal force of node 2 is $f_2^{int} = 0$ for any number of particles inside each element. Hence, when particles cross elements boundaries, the unbalance forces induced from such crossing is mitigated by adopting a classical finite element integration. However, one could argue that we are integrating using the element volume which, in MPM, does not represent the true volume of particles occupying the considered element. This is true, but it should be clear that we adopt such integration only for quasi-static problems, in which the equilibrium solution is of interest rather than the wave propagation. For dynamic problems, such integration is not used and the integration is done using particles as it is clear that using Gauss integration with full element volume does not conserve energy. A second reason that this integration procedure does not conserve energy is that, stresses are mapped from particles to Gauss points using a weighted least square approach, which smoothens stresses inside the element.

An important distinction must be made in the treatment of *fully filled elements* versus *partially filled elements*, shown in Figure 4.9. For elements that are partially filled, regular particle integration is adopted, whereas Gauss integration is used in the integration of fully filled elements. Elements that are located in the interior of the considered body are identified to be alway fully filled. An element located on the boundary of the body is considered to be partially filled when the volume sum of all particles inside this element is less than a prescribed percentage of the element volume, i.e.,

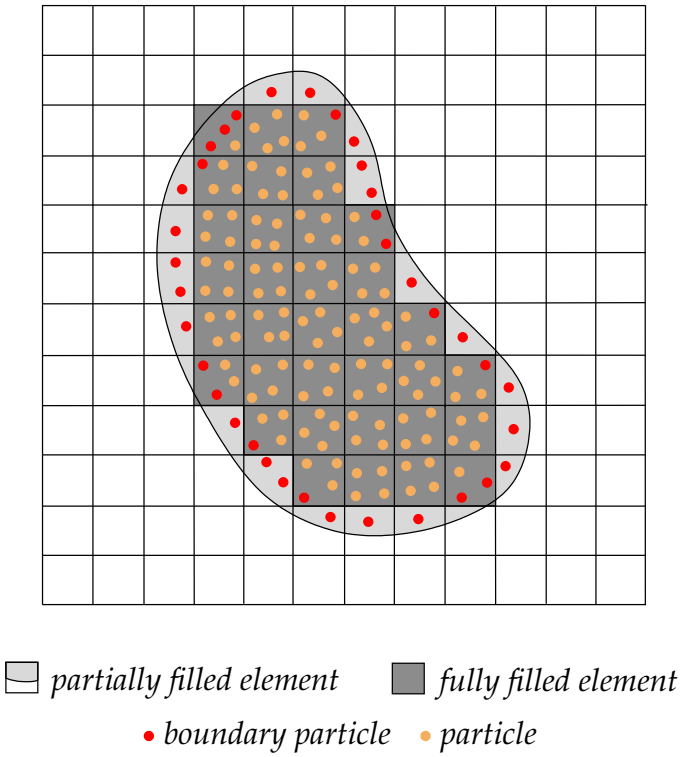


Figure 4.9: Illustration of partially and fully filled elements in MPM

$$\sum_{p=1}^{n_{ep}} \Omega_p < \beta\, \Omega_e, \tag{4.45}$$

in which the factor β is set to 0.9 as the sensitivity study by Beuth [24] showed very little influence when using a value of β around 0.9. It remains to mention here that elements having at least a boundary particle are identified as boundary elements. Furthermore, when using Gauss integration, the state variables of the constitutive model are averaged using the same way of Equation 4.44, i.e., by a weighted least square. Elements that have particles following different constitutive models are identified as *mixed elements* and are integrated using particle integration. One should also keep in mind that Gauss integration for fully filled elements is applied only for the integration of the internal forces as these forces are the main reason for the problem of particle crossing. We would like to emphasize here that throughout this thesis wherever *MPM integration* is mentioned, it means classical particle integration for all elements and no distinction is made between fully and partially filled elements, whereas *mixed integration* means that we distinguish between fully and partially filled elements.

An advantage of adopting the mixed integration is some gain in the computational efficiency as the summation when integrating the internal forces runs over the number of Gauss points in the element. In the adopted low-order tetrahedral element, we use a single Gauss point per each element. Moreover, the stress increment and other state variables are evaluated only for Gauss point and assigned then to all particles in the element. The procedure adopted for *mixed integration* can be summarized as follows

1. At the beginning of the time step, fully filled elements are identified. The fully filled element e satisfies the following criterion

$$\sum_{p=1}^{n_{ep}} \Omega_p \geqslant \beta\, \Omega_e.$$

2. Stresses of the fully filled element e are averaged using the volume associated with each particle as a weighting

$$\boldsymbol{\sigma} = \frac{\sum_{p=1}^{n_{ep}} \boldsymbol{\sigma}_p \Omega_p}{\sum_{p=1}^{n_{ep}} \Omega_p}.$$

   This stress is then assigned to all particles and Gauss points of the fully filled element. All state variables of the constitutive model are treated with the same procedure as for stresses.

3. The internal force vector is integrated using Gauss integration as follows

$$\boldsymbol{F}_e^{int} \approx \sum_{q=1}^{n_{eq}} w_q\, \boldsymbol{B}^T\left(\boldsymbol{\xi}_q\right) \boldsymbol{\sigma}_q \left|\boldsymbol{J}\left(\boldsymbol{\xi}_q\right)\right|,$$

   in which $\boldsymbol{\sigma}_q$ is the average stress of the element.

4. The constitutive equations are applied to Gauss points of the fully filled elements. After updating the stresses and other state variables of Gauss points, they are assigned to all particles of the element.

Partially filled elements are integrated using particle integration and hence the above steps are not applied for them. It is worth mentioning here that the above mapping procedure of stresses and other state variables between particles and Gauss points is trivial as we are considering low-order element, in which the stress is constant by definition. However, for higher-order elements, the mapping procedure requires more complicated least squares, in which higher-order polynomials have to be solved. These polynomials have to be consistent with the stress field distribution inside the element, e.g., for a second-order element, the stress is linear and hence the mapping polynomial has to be linear as well.

4.6.3 Numerical example on MPM and mixed integration

In this example, the loading of a column by its self weight is analyzed. The column has a height of $1\,[m]$ and a square cross-section of $0.01\,[m^2]$ area. The bottom of the column was fully fixed and roller boundaries were prescribed along the sides so that displacement was only possible in vertical direction. The material of the column was assumed to be linear elastic with a density of $\varrho = 2000\,[kg/m^3]$, Young's modulus of $E = 40\,[kN/m^2]$ and a Poisson's ratio of $\nu = 0$. The mesh of the problem was constructed of 504 tetrahedral elements with 198 nodes. Each active element in the initial configuration had 10

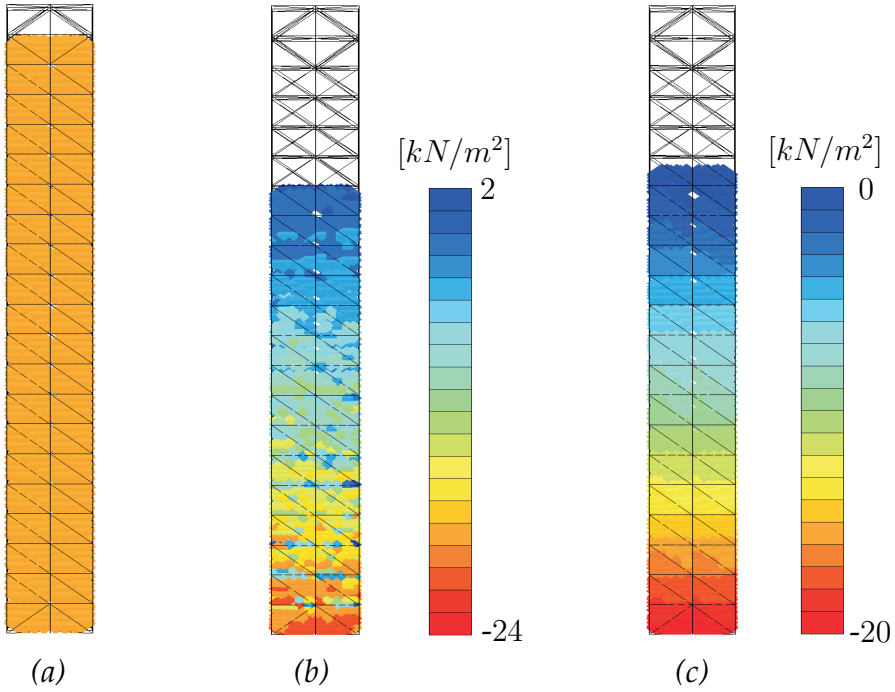


Figure 4.10: Discretization and results of the column problem (a) initial configuration (b) vertical stresses using MPM integration (c) vertical stresses using mixed integration

particles, see Figure 4.10(a). The local damping procedure of Section 4.10 was used with $\alpha = 0.75$ to get faster convergence to quasi-static equilibrium. The convergence to quasi-static equilibrium was detected by the convergence procedure of Section 4.9. A Courant number of $\alpha_{Cour} = 0.98$ was adopted for this problem.

The gravity load was applied instantaneously and kept constant throughout the computations. A comparison of the mixed and MPM integration is shown in Figure 4.10. We can see in Figure 4.10(b) that MPM integration produces a pronounced noise in stresses because of the problem of particle crossing between elements. It is also shown that some points have tensile (positive) stress and that there is considerable overestimation of stresses. When adopting mixed integration, the problem of the spacial oscillations in stresses is mitigated, see Figure 4.10(c). This figure shows a smooth and linear distribution of vertical stresses along the column with the expected values.

4.7 Treatment of spatially unbounded domains using absorbing boundaries

In numerical simulations of problems involving wave propagation, the use of finite boundaries leads to reflection of waves upon reaching the boundaries of the mesh. In geomechanics, rigid boundary is mostly numerical artifact and reflecting waves are not physical. Hence, they affect the solution considerably. Therefore, the attenuation of waves reflection is necessary in problems where there are artificial boundaries. This problem might be overcome by choosing the finite boundaries of the mesh far enough so that no reflection occurs. This is however not always a practical solution as it makes the mesh unnecessarily large leading to a substantial increase in the computational effort. The literature on solving the reflections problem is quite extensive, see references [101, 106] and references therein. An excellent review of treating wave reflection and more detailed bibliography is given by Givoli [70]. We however do not intend to cover such extensive literature, but rather briefly discuss the most common ones.

A solution to the problem was introduced by Lysmer and Kuhlemeyer [110]. They proposed the procedure where the boundary is supported on a dashpot. They showed that the absorption of the waves is nearly perfect for a wide range of incidence angles. Smith [151] proposed a technique to eliminate the reflections. It is based on superimposing solutions that satisfy both Dirichlet and Neumann boundary conditions. Although the technique is based on analytical formulation and was shown to be efficient in wave absorption, it is computationally very expensive. In general, if n boundary faces are required to be absorbing, 2^n solutions need to be superimposed to get the full solution [151]. The so-called infinite elements introduced by Bettess [22], Bettess and Zienkiewicz [23] are also used to solve infinite domain problems. Infinite elements are defined as radiating strips in the exterior regions. The shape functions of such elements include, for instance, an exponential decay term so that they mimic the asymptotic behavior at infinity [22, 23, 70]. However, Bettess and Zienkiewicz [23] pointed out that infinite elements involve more computations than the finite elements as they use more parameters. Furthermore, the choice of the decay length is somewhat arbitrary.

In this thesis, we intend to adopt the method of the boundary dashpots by Lysmer and Kuhlemeyer [110] with some modification to the basic approach developed in this study. Our motivation to adopt this kind of absorbing boundaries is that they are simple, effective and ultimately based on analytical justification.

4.7.1 Formulation of boundary dashpots

A simple way to formulate the boundary dashpots is to consider the tractions corresponding to them in the virtual work equation. The boundary of the continuum is then defined as

$$\partial\Omega_u \cup \partial\Omega_\tau \cup \partial\Omega_{\tau^{vb}} = \partial\Omega \tag{4.46}$$

and

$$\partial\Omega_i \cap \partial\Omega_j = \varnothing \qquad i,j = u,\tau,\tau^{vb} \tag{4.47}$$

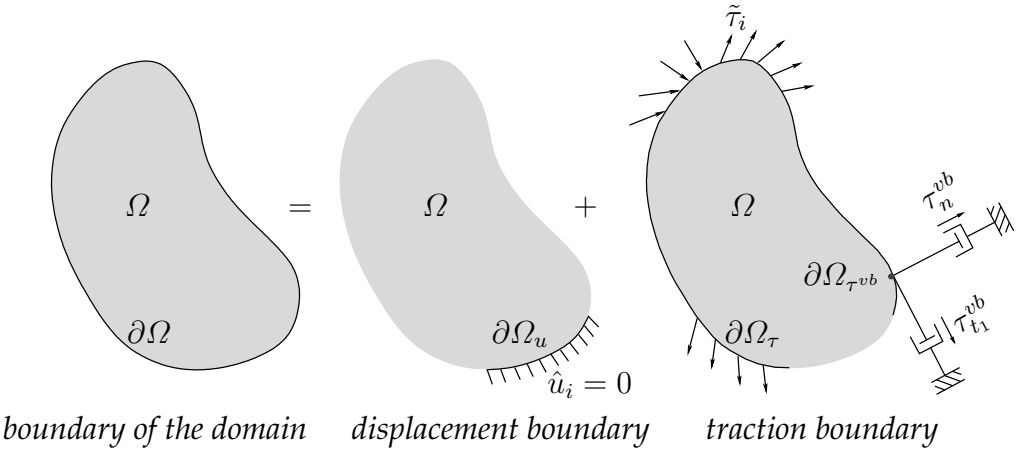


Figure 4.11: Absorbing dashpots at the boundary of a continuum

with $\partial\Omega_{\tau^{vb}}$ being the boundary where the dashpots are prescribed. On denoting the traction vector corresponding to boundary dashpots at a material point located at the boundary of the domain as τ_i^{vb}, see Figure 4.11, one can write the following

$$\tau_i^{vb} = \begin{bmatrix} \tau_n^{vb} & \tau_{t_1}^{vb} & \tau_{t_2}^{vb} \end{bmatrix}^T. \tag{4.48}$$

It contains one normal component τ_n^{vb} and two tangential components, $\tau_{t_1}^{vb}$ and $\tau_{t_2}^{vb}$. However, in Figure 4.11, we only see τ_n^{vb} and $\tau_{t_1}^{vb}$ as we consider a two-dimensional plot for simplicity. Lysmer and Kuhlemeyer [110] investigated many possible analytical expressions to control the tractions induced by the dashpots for best absorbing properties. They however found that an effective control could be by the following expressions

$$\begin{aligned} \tau_n^{vb} &= -a\varrho c_p \hat{v}_n, \\ \tau_{t_1}^{vb} &= -b\varrho c_s \hat{v}_{t_1}, \\ \tau_{t_2}^{vb} &= -b\varrho c_s \hat{v}_{t_2}, \end{aligned} \tag{4.49}$$

in which a and b are dimensionless parameters, $\hat{v}_n$, $\hat{v}_{t_1}$ and $\hat{v}_{t_2}$ are the velocities of the boundary point where the dashpots are applied. These velocities correspond to the directions of τ_n^{vb}, $\tau_{t_1}^{vb}$ and $\tau_{t_2}^{vb}$, respectively, c_p and c_s are the propagation speeds of the primary and secondary waves, respectively. They are defined in terms of the density ϱ, constrained modulus E^c and the shear modulus G as follows

$$c_p = \sqrt{\frac{E^c}{\varrho}} \qquad \text{and} \qquad c_s = \sqrt{\frac{G}{\varrho}}. \tag{4.50}$$

Let us re-consider the virtual work equation (Equation 3.37) again

$$\int_\Omega \delta\hat{v}_i \varrho \frac{d\hat{v}_i}{dt}\, d\Omega = \int_{\partial\Omega_\tau} \delta\hat{v}_i \tau_i\, dS + \int_\Omega \delta\hat{v}_i \varrho g_i\, d\Omega - \int_\Omega \frac{\partial(\delta\hat{v}_i)}{\partial x_j}\sigma_{ij}\, d\Omega. \tag{4.51}$$

We account now for the boundary dashpots as an additional traction boundaries as shown in Figure 4.11. Hence, we simply add them to the virtual work equation as

$$\int_{\Omega} \delta\hat{v}_i \varrho \frac{d\hat{v}_i}{dt}\, d\Omega = \int_{\partial\Omega_\tau} \delta\hat{v}_i \tau_i\, dS + \int_{\partial\Omega_{\tau^{vb}}} \delta\hat{v}_i \tau_i^{vb}\, dS + \int_{\Omega} \delta\hat{v}_i \varrho g_i\, d\Omega - \int_{\Omega} \frac{\partial(\delta\hat{v}_i)}{\partial x_j}\sigma_{ij}\, d\Omega. \tag{4.52}$$

Let us re-write Equation 4.49 in the following compact form

$$\tau_i^{vb} = -\eta_{ij}\hat{v}_j, \tag{4.53}$$

in which

$$\eta_{ij} = \begin{bmatrix} a\varrho c_p & 0 & 0 \\ 0 & b\varrho c_s & 0 \\ 0 & 0 & b\varrho c_s \end{bmatrix} \quad \text{and} \quad \hat{v}_j = \begin{bmatrix} \hat{v}_n & \hat{v}_{t_1} & \hat{v}_{t_2} \end{bmatrix}^T. \tag{4.54}$$

Substituting Equation 4.53 into Equation 4.52 yields

$$\int_{\Omega} \delta\hat{v}_i \varrho \frac{d\hat{v}_i}{dt}\, d\Omega = \int_{\partial\Omega_\tau} \delta\hat{v}_i \tau_i\, dS - \int_{\partial\Omega_{\tau^{vb}}} \delta\hat{v}_i \eta_{ij}\hat{v}_j\, dS + \int_{\Omega} \delta\hat{v}_i \varrho g_i\, d\Omega - \int_{\Omega} \frac{\partial(\delta\hat{v}_i)}{\partial x_j}\sigma_{ij}\, d\Omega. \tag{4.55}$$

Without repeating the space discretization of the above equation, as it is already detailed in Section 3.3, we directly write the discrete form of the virtual work equation using matrix notations as

$$\int_{\Omega} \bar{\boldsymbol{N}}^T \varrho \bar{\boldsymbol{N}} \boldsymbol{a}\, d\Omega = \int_{\partial\Omega_\tau} \bar{\boldsymbol{N}}^T \boldsymbol{\tau}\, dS - \int_{\partial\Omega_{\tau^{vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta} \bar{\boldsymbol{N}} \boldsymbol{v}\, dS + \int_{\Omega} \bar{\boldsymbol{N}}^T \varrho \boldsymbol{g}\, d\Omega - \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma}\, d\Omega, \tag{4.56}$$

where $\boldsymbol{\eta} = \eta_{ij}$, see Equation 4.54. All other symbols and terms are previously defined. Hence, one can write the equation of motion including the term corresponding to the boundary dashpots as

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{F}^{trac} - \boldsymbol{F}^{vb} + \boldsymbol{F}^{grav} - \boldsymbol{F}^{int}, \tag{4.57}$$

in which all terms are well defined previously except the second term of the right-hand side which is defined as

$$\boldsymbol{F}^{vb} = \boldsymbol{C}^c \boldsymbol{v}, \tag{4.58}$$

where

$$\boldsymbol{C}^c = \int_{\partial\Omega_{\tau^{vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta} \bar{\boldsymbol{N}}\, dS \tag{4.59}$$

is a consistent matrix containing the dashpot coefficients and called in this thesis *dashpots matrix*.

4.7.1.1 Integration of the dashpots matrix

In this thesis, our intention is not to construct consistent matrices, but rather use lumped matrices for computational efficiency. Hence, the consistent dashpots matrix $\boldsymbol{C}^c$ will be lumped in a way similar to that explained in Chapter 3. On denoting the lumped dashpots matrix of the volume element e as $\boldsymbol{C}_e$, we can define it in the following way

$$\boldsymbol{C}_e = \begin{bmatrix} \boldsymbol{c}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{c}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{c}_{en} \end{bmatrix}. \tag{4.60}$$

The sub-matrix $\boldsymbol{c}_i$ corresponds to the dashpots coefficients of node i and the matrix $\mathbf{0}$ is a null matrix. They have the following forms

$$\boldsymbol{c}_i = \begin{bmatrix} c_n & 0 & 0 \\ 0 & c_{t_1} & 0 \\ 0 & 0 & c_{t_2} \end{bmatrix} \quad \text{and} \quad \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.61}$$

One should realize here that when defining the element dashpot matrix $\boldsymbol{C}_e$, all nodes of the volume elements are included, but it is clear that the dashpots coefficients of the interior nodes will be zero as the dashpots are only applied at the boundaries. We now consider the integration of the dashpots matrix of node i, i.e., $\boldsymbol{c}_i$. It is integrated as

$$\boldsymbol{c}_i \approx \sum_{q=1}^{n_{seq}} w_q \, N_i \left(\boldsymbol{\xi}_q\right) \boldsymbol{\eta} \, S_e, \tag{4.62}$$

in which S_e is the area of the triangular surface element where the dashpots are applied, see Figure 4.12, and N_i is the shape function of node i of the 3-noded triangular surface

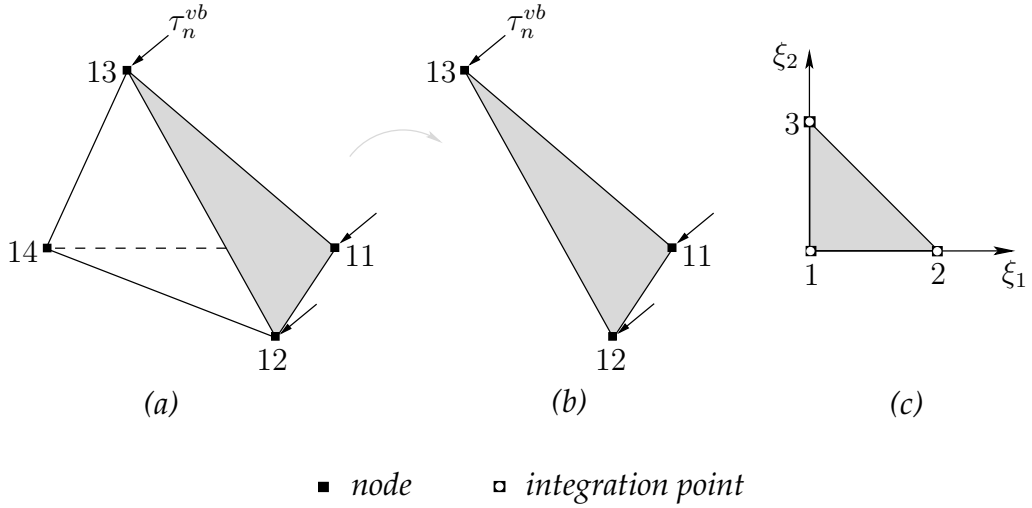


Figure 4.12: Tractions induced by the boundary dashpots are integrated using Newton-Cotes scheme (a) tetrahedral element (b) surface element and (c) parent triangular element with three Newton-Cotes points located at the nodes

element. Adopting Newton-Cotes integration simplifies Equation 4.62. The integration points in Newton-Cotes scheme are located at the same position of the nodes as shown in Figure 4.12. Hence, Equation 4.62 reduces to

$$\boldsymbol{c}_i \approx \frac{1}{3}\boldsymbol{\eta}\, S_e, \tag{4.63}$$

in which $\frac{1}{3}$ represents the integration weight of each Newton-Cotes point in the 3-noded triangular element. It is trivial that the shape function is eliminated in Equation 4.63 as it has a value of 1 at its corresponding node and zero at other nodes. This is the simplicity of Newton-Cotes scheme in that context. The element matrices are then constructed from the nodal ones, see Equation 4.60. The global dashpots matrix is assembled as

$$\boldsymbol{C} = \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \boldsymbol{C}_e. \tag{4.64}$$

The superscript c is eliminated from the dashpots matrix indicating that it is lumped matrix. It must be clarified here that the initially inactive elements are included in the initialization of the dashpots matrix as well. The dashpots coefficients of an initially inactive boundary node are activated once it becomes active. The same illustration of Figure 4.5 applies here.

4.7.2 Extending the boundary dashpots to Kelvin-Voigt elements

Scrutiny of Equation 4.49 shows that the boundary that is supported by dashpots tends to creep continuously, particularly under sustained loading. This is however not always required and the creep of this boundary must be controlled. To do so, the technique of the boundary dashpots is modified in the current study by replacing these dashpots with Kelvin-Voigt elements, see Figure 4.13. The response of these elements is given by the following equations

$$\begin{aligned}
\tau_n^{vb} &= -a\varrho c_p \hat{v}_n - \kappa_p \hat{u}_n, \\
\tau_{t_1}^{vb} &= -b\varrho c_s \hat{v}_{t_1} - \kappa_s \hat{u}_{t_1}, \\
\tau_{t_2}^{vb} &= -b\varrho c_s \hat{v}_{t_2} - \kappa_s \hat{u}_{t_2},
\end{aligned} \tag{4.65}$$

in which κ represents a stiffness per unit area and $\hat{u}$ refers to a displacement component of the boundary point where the Kelvin-Voigt element is prescribed. Equation 4.65 can also be written in a compact form as

$$\tau_i^{vb} = -\eta_{ij}\hat{v}_j - \kappa_{ij}\hat{u}_j, \tag{4.66}$$

in which

$$\kappa_{ij} = \begin{bmatrix} \kappa_p & 0 & 0 \\ 0 & \kappa_s & 0 \\ 0 & 0 & \kappa_s \end{bmatrix} \qquad \text{and} \qquad \hat{u}_j = \begin{bmatrix} \hat{u}_n & \hat{u}_{t_1} & \hat{u}_{t_2} \end{bmatrix}^T. \tag{4.67}$$

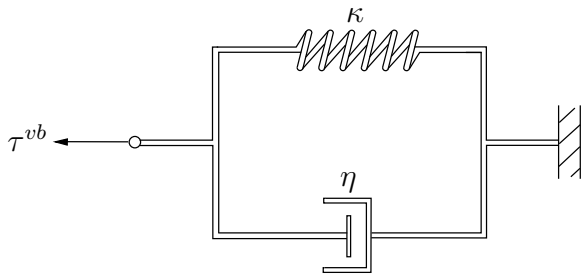


Figure 4.13: Kelvin-Voigt element

Now we simply add the tractions induced by the springs to Equation 4.55 as

$$\int_{\Omega} \delta\hat{v}_i \varrho \frac{d\hat{v}_i}{dt}\, d\Omega = \int_{\partial\Omega_\tau} \delta\hat{v}_i \tau_i\, dS - \int_{\partial\Omega_{\tau^{vb}}} \delta\hat{v}_i \eta_{ij} \hat{v}_j\, dS - \int_{\partial\Omega_{\tau^{vb}}} \delta\hat{v}_i \kappa_{ij} \hat{u}_j\, dS + \int_{\Omega} \delta\hat{v}_i \varrho g_i\, d\Omega - \int_{\Omega} \frac{\partial(\delta\hat{v}_i)}{\partial x_j} \sigma_{ij}\, d\Omega. \tag{4.68}$$

After the space discretization, the above equation can be written in matrix notations as

$$\int_{\Omega} \bar{\boldsymbol{N}}^T \varrho \bar{\boldsymbol{N}} \boldsymbol{a}\, d\Omega = \int_{\partial\Omega_\tau} \bar{\boldsymbol{N}}^T \boldsymbol{\tau}\, dS - \int_{\partial\Omega_{\tau^{vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta} \bar{\boldsymbol{N}} \boldsymbol{v}\, dS - \int_{\partial\Omega_{\tau^{vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\kappa} \bar{\boldsymbol{N}} \boldsymbol{u}\, dS + \int_{\Omega} \bar{\boldsymbol{N}}^T \varrho \boldsymbol{g}\, d\Omega - \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma}\, d\Omega \tag{4.69}$$

with $\boldsymbol{\kappa} = \kappa_{ij}$, see Equation 4.67. Hence, the term corresponding to the viscous boundary in Equation 4.57 is now modified to

$$\boldsymbol{F}^{vb} = \boldsymbol{C}\boldsymbol{v} + \boldsymbol{K}^c \boldsymbol{u}, \tag{4.70}$$

where

$$\boldsymbol{K}^c = \int_{\partial\Omega_{\tau^{vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\kappa} \bar{\boldsymbol{N}}\, dS \tag{4.71}$$

is a consistent matrix containing the springs coefficients and called in this thesis *springs matrix*. A lumped springs matrix will be used in this thesis and it is integrated following the procedure for the dashpots matrix, i.e.,

$$\boldsymbol{K}_e = \begin{bmatrix} \boldsymbol{k}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{k}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{k}_{en} \end{bmatrix}, \tag{4.72}$$

where

$$\boldsymbol{k}_i = \begin{bmatrix} k_n & 0 & 0 \\ 0 & k_{t_1} & 0 \\ 0 & 0 & k_{t_2} \end{bmatrix} \quad \text{and} \quad \boldsymbol{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.73}$$

The springs matrix of node i is then integrated using Newton-Cotes scheme as

$$\boldsymbol{k}_i \approx \frac{1}{3}\boldsymbol{\kappa}\, S_e. \tag{4.74}$$

The global lumped springs matrix $\boldsymbol{K}$ then follows from the assemblage of the element matrices as with the dashpots matrix. Hence,

$$\boldsymbol{K} = \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \boldsymbol{K}_e. \tag{4.75}$$

4.7.2.1 Stability of Kelvin-Voigt element

An investigation of the restrictions of Kelvin-Voigt element on the critical time step size is necessary. In order not to lose generality, we consider a general Kelvin-Voigt element which is subjected to a primary or secondary wave. Hence,

$$\tau^{vb} = -\eta\hat{v} - \kappa\hat{u}, \tag{4.76}$$

which can also be written as

$$-\eta\frac{d\hat{u}}{dt} = \tau^{vb} + \kappa\hat{u}. \tag{4.77}$$

Integrating the above equation in the time interval $[t, t+\Delta t]$ gives

$$-\int_t^{t+\Delta t} \eta d\hat{u} = \int_t^{t+\Delta t} \left(\tau^{vb} + \kappa\hat{u}\right) dt, \tag{4.78}$$

which results after explicit time integration in

$$-\eta\left(\hat{u}^{t+\Delta t} - \hat{u}^t\right) = \Delta t\left(\tau^{vb,t} + \kappa\hat{u}^t\right) \tag{4.79}$$

or

$$\hat{u}^{t+\Delta t} = \hat{u}^t - \Delta t\eta^{-1}\left(\tau^{vb,t} + \kappa\hat{u}^t\right). \tag{4.80}$$

Following the stability analysis of Section 3.4, one can write

$$\mathcal{H} = \frac{d\hat{u}^{t+\Delta t}}{d\hat{u}^t}, \tag{4.81}$$

which gives

$$\mathcal{H} = 1 - \Delta t\eta^{-1}\kappa. \tag{4.82}$$

As explained in Section 3.4, the stability requires the following condition to be satisfied

$$|\mathcal{H}| \leqslant 1 \tag{4.83}$$

and hence

$$\Delta t \leqslant \frac{2\eta}{\kappa}. \tag{4.84}$$

4.7.2.2 Choosing the spring coefficients

The coefficients κ_p and κ_s are chosen so that they provide a virtual viscous layer that goes beyond the boundaries of the mesh. The stiffness of this virtual layer is adopted from the material that is adjacent to the boundary where the Kelvin-Voigt elements are prescribed. Hence,

$$\kappa_p = \frac{E^c}{\delta} \qquad \text{and} \qquad \kappa_s = \frac{G}{\delta}, \tag{4.85}$$

in which δ is the thickness of the virtual viscous layer. It has the following range

$$0 \leqslant \delta \leqslant \infty, \tag{4.86}$$

which implies the following conditions

$$\begin{aligned} \delta = 0 &\Longrightarrow \textit{the element reduces to a rigid boundary} \\ \delta = \infty &\Longrightarrow \textit{the element reduces to a dashpot boundary.} \end{aligned} \tag{4.87}$$

On introducing c_p, the speed of the primary wave, and c_s, the speed of the secondary wave, one can write

$$E^c = \varrho c_p^2 \qquad \text{and} \qquad G = \varrho c_s^2. \tag{4.88}$$

Substituting the above equation into Equation 4.85 yields

$$\kappa_p = \frac{\varrho c_p^2}{\delta} \qquad \text{and} \qquad \kappa_s = \frac{\varrho c_s^2}{\delta}. \tag{4.89}$$

If the Kelvin-Voigt element is subjected to a primary wave, i.e., the dashpot coefficient $\eta = a\varrho c_p$ and the spring coefficient $\kappa = \frac{\varrho c_p^2}{\delta}$, the critical time step is

$$\Delta t_{crit} = \frac{2a\delta}{c_p} \tag{4.90}$$

and for a secondary wave

$$\Delta t_{crit} = \frac{2b\delta}{c_s}. \tag{4.91}$$

However, for given c_p and c_s, one should make sure that

$$\delta \geqslant \frac{h_{min}}{2a} \quad \text{and} \quad \delta \geqslant \frac{h_{min}}{2b} \tag{4.92}$$

to exclude the restriction of δ on the critical time step size. In the above equation h_{min} is the characteristic length of the finite element mesh, see Equation 3.117.

4.7.2.3 Remark on the implementation of Kelvin-Voigt element

The values of the dimensionless parameters a and b are chosen depending on the wave incidence angle so that they provide maximum absorption. However, Lysmer and Kuhlemeyer [110] showed that the absorption cannot be perfect over the whole range of the

incidence angle by any choice of a and b. They performed series of tests for the values of a and b for a wide range of the incidence angle. They noticed that the choice $a = b = 1$ is effective for both primary and secondary waves. Hence, we will use these values in this thesis. As particles move through the mesh, the scheme must be implemented in an incremental form so that displacement and velocity increments of a particle are only accounted for when this particle enters the considered element. Hence,

$$\boldsymbol{F}^{vb,t} = \boldsymbol{F}^{vb,t-\Delta t} + \Delta \boldsymbol{F}^{vb,t}, \tag{4.93}$$

where

$$\Delta \boldsymbol{F}^{vb,t} = \boldsymbol{C}^t \Delta \boldsymbol{v}^t + \boldsymbol{K}^t \Delta \boldsymbol{u}^t. \tag{4.94}$$

Another advantage of such implementation is to allow flexibility in changing the parameter δ during computations.

4.7.3 Numerical example on viscous boundaries

As a numerical example, a material column is considered to be loaded instantaneously by a compression surface traction of $1\,[kN/m^2]$. The column has a height of $1\,[m]$. The material of the column was assumed to be linear elastic with a density of $\varrho = 1800\,[kg/m^3]$, Young's modulus of $E = 100\,[kN/m^2]$ and a Poisson's ratio of $\nu = 0$. The problem was discretized with 1000 rows of elements along the height, each containing 6 tetrahedral elements. The total number of particles was 6000 and each element had a single particle. A fine grid was required to properly capture the propagating front for the idealized loading. Roller boundaries were prescribed along the sides so that only displacement along the column was possible.

Three different boundary conditions were considered at the bottom: fully fixed (rigid), dashpot and Kelvin-Voigt boundary. With the rigid boundary, the entire energy reflected back when the wave reached the bottom and hence the stress was doubled after reflection, see Figure 4.14. Considering the plots of the rigid boundary case, the wave front was traveling downwards at time $t = 0.107\,[s]$ and upwards with double amplitude at time $t = 0.16\,[s]$. It happened to be at the same spatial point of the column, but with different travel directions, at both times. When replacing the rigid boundary by a dashpot, only a very small portion of the energy reflected which is hardly seen in Figure 4.14. For this case the wave front was plotted at time $t = 0.16\,[s]$, showing almost no reflections. The high frequency oscillations observed in Figure 4.14 are attributed to the numerical process and the instantaneous application of the full load in a single time step. They can be suppressed by applying the load gradually over a couple of time steps. Further refinement of the mesh can also mitigate them.

However, the drawback of using only dashpot is a continuous creep of the boundary as shown in Figure 4.15. On the other hand, Kelvin-Voigt element limits the creep of the boundary as shown in Figure 4.16. The final displacement of the bottom is calculated as

$$u = \frac{\sigma \delta}{E^c} \tag{4.95}$$

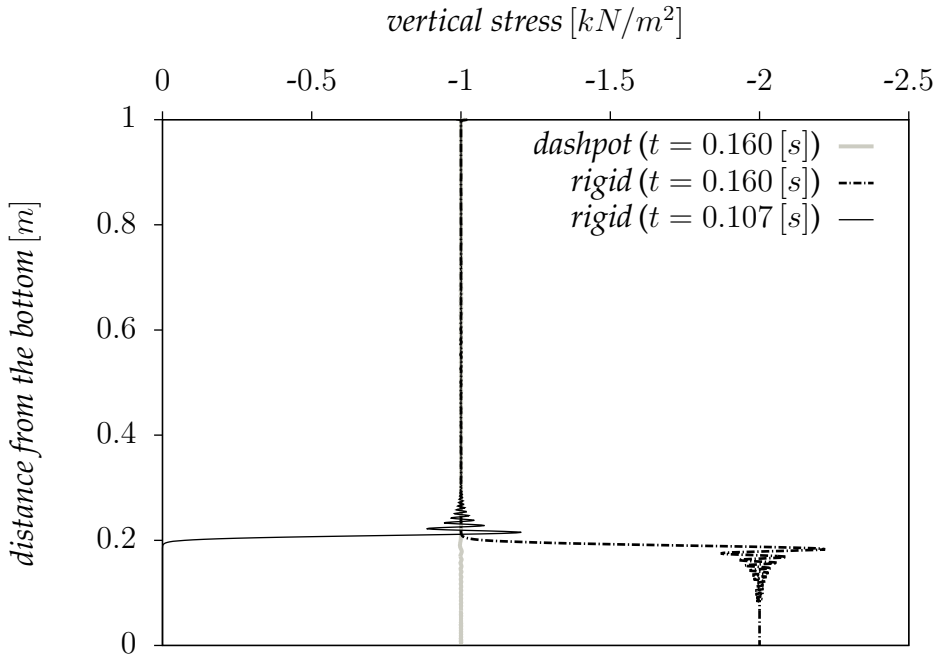


Figure 4.14: Wave propagation along the column with rigid and dashpots boundaries prescribed at the bottom

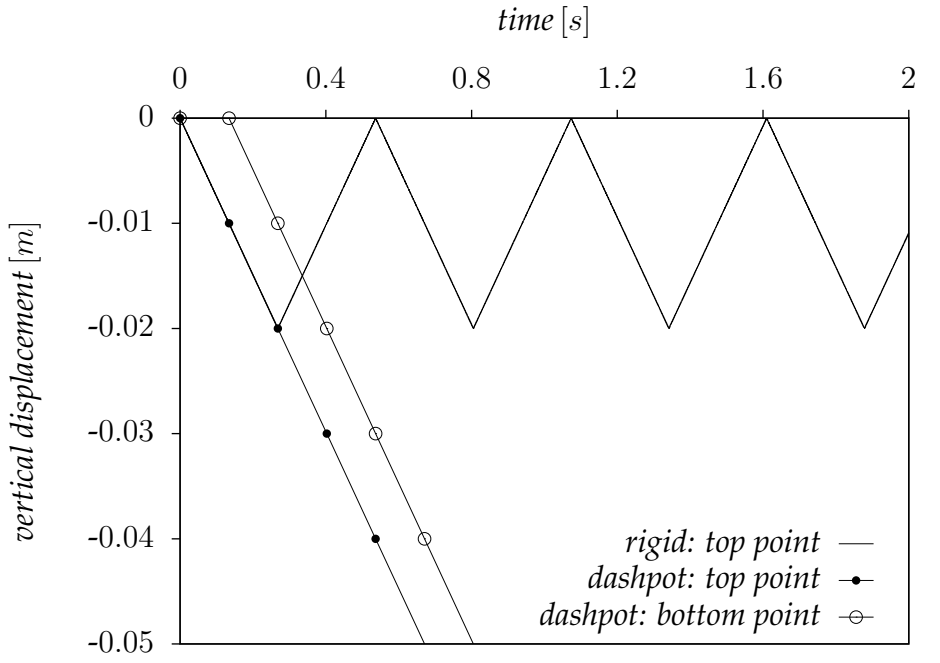


Figure 4.15: Displacement with time for the top and bottom points of the column with rigid and dashpots boundaries prescribed at the bottom

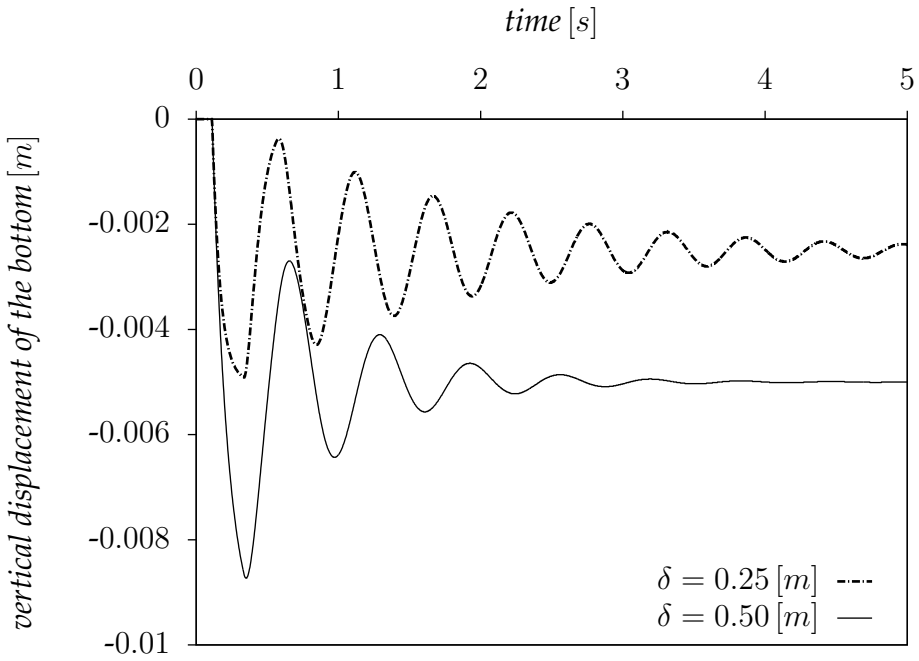


Figure 4.16: Displacement of the bottom with time when Kelvin-Voigt boundary is used with different values of δ

with E^c and σ being the constrained modulus and the applied stress, respectively. The final displacement for the case of $\delta = 0.5\,[m]$ was found to be $0.005\,[m]$ as shown in Figure 4.16 and that for the case of $\delta = 0.25\,[m]$ was $0.0025\,[m]$.

4.8 Pathological locking in low-order elements

It should be clearly stated that the material point method suffers from the same numerical problems that are encountered when using low-order finite element method for incompressible problems. Therefore, this section reviews the problems associated with low-order element and possible procedures to mitigate them. Difficulties arise when determining the displacement field for a solid that is nearly incompressible. For such a material, the bulk modulus is very large and small errors in strain will yield large errors in stress. Furthermore, when dealing with low-order elements, the mesh may lock when incompressibility constraints from neighboring elements are imposed. For a simple demonstration of the locking phenomenon, a two-dimensional domain discretized with 3-noded triangular elements is shown in Figure 4.17. Elements e_1 and e_2 share the same free node. Taking into account the incompressibility constraint, the allowable displacements at the free node are $\boldsymbol{u}_{e_1}$ and $\boldsymbol{u}_{e_2}$, respectively. Since, the two directions are not the same, this node locks. As the material is incompressible, constraining of this node also leads to the locking of the free node attached to element e_3. Hence, such locking usually propagates throughout the entire mesh yielding unrealistic stiff response.

A common approach to deal with locking in low-order elements is through the use

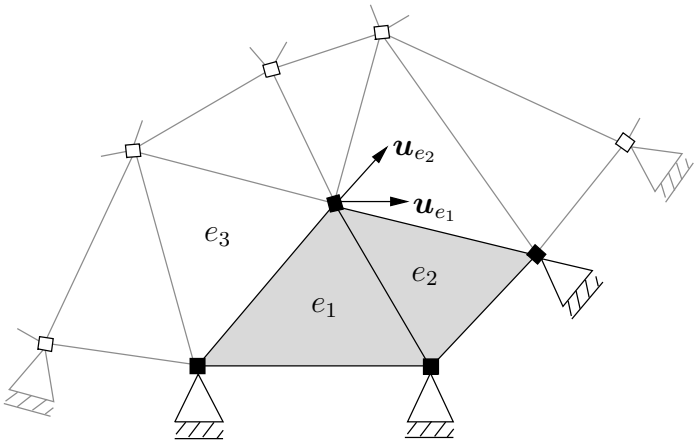


Figure 4.17: Illustration of locking with 3-noded triangular elements

of enhanced strain fields such as bubble functions [170]. Bonet et al. [31] compare two approaches to mitigate mesh locking, both taking advantage of explicit time-stepping algorithms: average nodal pressure (ANP) formulation versus fractional step or velocity correction procedure. A mixed discretization technique, which is similar to the ANP and is useful for mitigating locking associated with incompressible plastic deformations, is proposed by Detournay and Dzik [59]. The technique is referred to as *nodal mixed discretization* (NMD). In this thesis, we will adopt the NMD procedure, referred to here as the *enhanced volumetric strain* approach, as it showed effectiveness in mitigating the locking associated with (near) incompressible deformations that are encountered in engineering problems.

4.8.1 Attenuation of locking by the enhanced volumetric strain procedure

To formulate the procedure, let us decompose the stress rate tensor $\dot{\sigma}_{ij}$ to a tensor of deviatoric stress rate $\dot{s}_{ij}$ and a tensor containing the rate of the mean stress $\dot{p}\delta_{ij}$ as

$$\dot{\sigma}_{ij} = \dot{s}_{ij} + \dot{p}\delta_{ij}, \tag{4.96}$$

in which δ_{ij} takes the usual definition of Kronecker delta and $\dot{p}$ is the rate of pressure or mean stress. It is related to the stress rate tensor $\dot{\sigma}_{ij}$ as

$$\dot{p} = \frac{\dot{\sigma}_{kk}}{3} = \frac{\dot{\sigma}_{11} + \dot{\sigma}_{22} + \dot{\sigma}_{33}}{3}. \tag{4.97}$$

The strain rate tensor $\dot{\varepsilon}_{ij}$ can also be partitioned into a tensor of deviatoric strain rate $\dot{\varepsilon}^d_{ij}$ and a tensor containing the rate of the volumetric strain $\dot{\varepsilon}_{vol}\delta_{ij}$ as

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}^d_{ij} + \frac{1}{3}\dot{\varepsilon}_{vol}\delta_{ij}, \tag{4.98}$$

where the volumetric strain rate $\dot{\varepsilon}_{vol}$ can be written as

$$\dot{\varepsilon}_{vol} = \dot{\varepsilon}_{kk} = \dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}. \tag{4.99}$$

For elastic material, the rate of pressure $\dot{p}$ is related to the volumetric strain rate $\dot{\varepsilon}_{vol}$ by the bulk modulus of the material K as

$$\dot{p} = K\dot{\varepsilon}_{vol}. \tag{4.100}$$

We now multiply Equation 4.100 by a virtual pressure rate $\delta\dot{p}$ and integrate over the domain Ω as

$$\int_{\Omega} \delta\dot{p}\,\dot{p}\,d\Omega = \int_{\Omega} \delta\dot{p}\,K\dot{\varepsilon}_{vol}\,d\Omega. \tag{4.101}$$

The pressure rate can be discretized as

$$\dot{p} \approx \bar{\boldsymbol{N}}_p\,\dot{\boldsymbol{p}} \tag{4.102}$$

with $\bar{\boldsymbol{N}}_p$ being a matrix containing the interpolation functions for the pressure rate. It is defined as

$$\bar{\boldsymbol{N}}_p = \begin{bmatrix} \bar{N}_1 & \bar{N}_2 & \bar{N}_3 & \cdots & \bar{N}_{n_T} \end{bmatrix} \tag{4.103}$$

and $\dot{\boldsymbol{p}}$ is a vector containing the nodal values of pressure rate and defined as

$$\dot{\boldsymbol{p}} = \begin{bmatrix} \dot{p}_1 & \dot{p}_2 & \dot{p}_3 & \ldots & \dot{p}_{n_T} \end{bmatrix}^T, \tag{4.104}$$

where n_T denotes the total number of nodes. Substituting Equation 4.102 and its corresponding virtual form into Equation 4.101 gives

$$\delta\dot{\boldsymbol{p}}^T \int_{\Omega} \bar{\boldsymbol{N}}_p^T \bar{\boldsymbol{N}}_p\,\dot{\boldsymbol{p}}\,d\Omega = \delta\dot{\boldsymbol{p}}^T \int_{\Omega} \bar{\boldsymbol{N}}_p^T\,K\dot{\varepsilon}_{vol}\,d\Omega, \tag{4.105}$$

in which $\delta\dot{\boldsymbol{p}}$ contains virtual values of pressure rate and hence can be eliminated so that

$$\int_{\Omega} \bar{\boldsymbol{N}}_p^T \bar{\boldsymbol{N}}_p\,\dot{\boldsymbol{p}}\,d\Omega = \int_{\Omega} \bar{\boldsymbol{N}}_p^T\,K\dot{\varepsilon}_{vol}\,d\Omega. \tag{4.106}$$

For a homogeneous material, Equation 4.106 can be written as

$$\int_{\Omega} \bar{\boldsymbol{N}}_p^T \bar{\boldsymbol{N}}_p\,\dot{\boldsymbol{\varepsilon}}_{vol}\,d\Omega = \int_{\Omega} \bar{\boldsymbol{N}}_p^T \dot{\varepsilon}_{vol}\,d\Omega, \tag{4.107}$$

where $\dot{\boldsymbol{\varepsilon}}_{vol}$ is a vector containing the smoothed values of the nodal volumetric strain rate. This vector represents the unknown for which Equation 4.107 is solved. It is defined as

$$\dot{\boldsymbol{\varepsilon}}_{vol} = \begin{bmatrix} \dot{\varepsilon}_{vol1} & \dot{\varepsilon}_{vol2} & \dot{\varepsilon}_{vol3} & \ldots & \dot{\varepsilon}_{voln_T} \end{bmatrix}^T. \tag{4.108}$$

Referring to Chapter 3 and Appendices D and C for the assemblage procedure of global arrays from element arrays and for the space integration using the parent element domain, one can write the following

$$\boldsymbol{V}^c\dot{\boldsymbol{\varepsilon}}_{vol} = \tilde{\dot{\boldsymbol{\varepsilon}}}_{vol}, \tag{4.109}$$

in which

$$\boldsymbol{V}^c \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{N}_p^T \left(\boldsymbol{\xi}_q\right) \boldsymbol{N}_p \left(\boldsymbol{\xi}_q\right) \left|\boldsymbol{J} \left(\boldsymbol{\xi}_q\right)\right| \tag{4.110}$$

and

$$\tilde{\dot{\boldsymbol{\varepsilon}}}_{vol} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{N}_p^T \left(\boldsymbol{\xi}_q\right) \dot{\varepsilon}_{vol} \left(\boldsymbol{\xi}_q\right) \left|\boldsymbol{J} \left(\boldsymbol{\xi}_q\right)\right| . \tag{4.111}$$

We however adopt a lumped form of the matrix $\boldsymbol{V}^c$ and denote it as $\boldsymbol{V}$. The lumped matrix is then assembled as

$$\boldsymbol{V} = \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \boldsymbol{V}_e \tag{4.112}$$

with

$$\boldsymbol{V}_e = \begin{bmatrix} V_1 & 0 & \cdots & 0 \\ 0 & V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_{en} \end{bmatrix}, \tag{4.113}$$

where V_i is the volume associated with node i. It is integrated as

$$V_i \approx \sum_{q=1}^{n_{eq}} w_q \, N_i \left(\boldsymbol{\xi}_q\right) \left|\boldsymbol{J} \left(\boldsymbol{\xi}_q\right)\right| . \tag{4.114}$$

Hence, Equation 4.109 takes its final form as

$$\boldsymbol{V} \dot{\boldsymbol{\varepsilon}}_{vol} = \tilde{\dot{\boldsymbol{\varepsilon}}}_{vol}, \tag{4.115}$$

which is solved for $\dot{\boldsymbol{\varepsilon}}_{vol}$. Having calculated the vector $\dot{\boldsymbol{\varepsilon}}_{vol}$, the field of the smoothed volumetric strain rate $\bar{\dot{\varepsilon}}_{vol}$ inside an element is calculated as

$$\bar{\dot{\varepsilon}}_{vol} \left(\boldsymbol{\xi}_q\right) = \sum_{i=1}^{n_{en}} N_i \left(\boldsymbol{\xi}_q\right) \dot{\varepsilon}_{voli}. \tag{4.116}$$

The modified strain rate tensor is then calculated as follows

$$\bar{\dot{\varepsilon}}_{ij} = \dot{\varepsilon}_{ij} - \frac{1}{3} \left(\dot{\varepsilon}_{vol} - \bar{\dot{\varepsilon}}_{vol}\right) \delta_{ij}. \tag{4.117}$$

The idea of the procedure as explained above, also stated by Detournay and Dzik [59], is to extend the volumetric behavior over which the incompressibility constraint applies to include neighboring elements via a least squares smoothening process. Figure 4.18 illustrates the idea of such smoothening procedure.

We would like to remind the reader that the above procedure is adopted for low-order elements, in which the strain is constant and usually a single Gauss point is used within such elements. Hence, the summation over Gauss points in the above equations can be replaced by evaluating the quantities at the position of the single point. We would also like to emphasize here that in the computer implementation of the enhanced volumetric strain procedure, we never explicitly construct the matrix $\boldsymbol{V}$, but rather adopt a procedure which is as simple as prescribed in the following steps

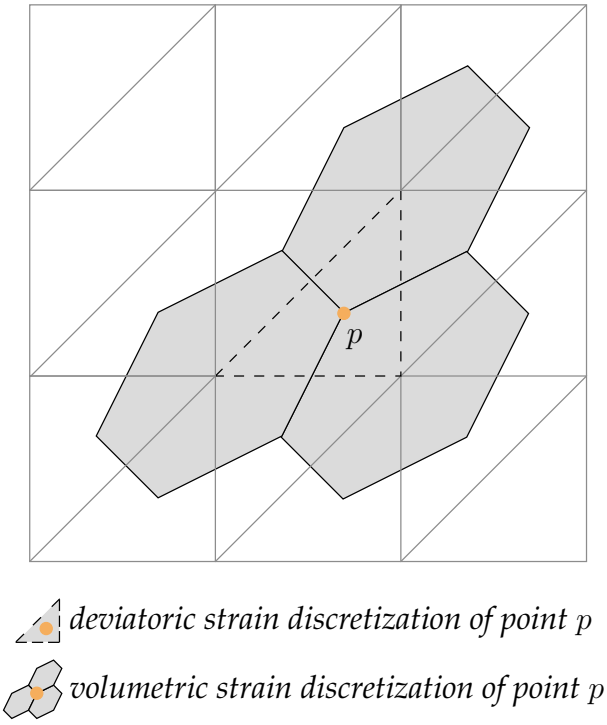


Figure 4.18: Discretization of the enhanced volumetric strain procedure [59]

1. After calculating the strain rate tensor $\dot{\varepsilon}_{ij}$ of each element, the nodal values of the smoothed strain rate are calculated. On considering node i as an example, it yields

$$\bar{\dot{\varepsilon}}_{voli} = \frac{\sum \dot{\varepsilon}_{vole} \Omega_e}{\sum \Omega_e}, \tag{4.118}$$

   where the sum runs over all elements attached to node i.

2. Once the smoothed volumetric strain rate associated with each node is calculated, the average volumetric strain rate for the element is determined by averaging the nodal values of the considered element. Consider element e as an example, one can write

$$\bar{\dot{\varepsilon}}_{vole} = \frac{1}{n_{en}} \sum_{i=1}^{n_{en}} \bar{\dot{\varepsilon}}_{voli}, \tag{4.119}$$

   where $\bar{\dot{\varepsilon}}_{vole}$ is the enhanced volumetric strain of element e.

3. As explained before, the deviatoric strain rate tensor needs not to be enhanced, only the volumetric strain rates. As a result, the final strain rate tensor within an element is redefined as

$$\begin{aligned} \bar{\dot{\varepsilon}}_{ij} &= \dot{\varepsilon}_{ij} - \frac{1}{3} \left(\dot{\varepsilon}_{vol} - \bar{\dot{\varepsilon}}_{vol} \right) \delta_{ij} \\ &= \dot{\varepsilon}^d_{ij} + \frac{1}{3} \bar{\dot{\varepsilon}}_{vol} \delta_{ij}. \end{aligned} \tag{4.120}$$

The new strains are then used to determine the stresses of particles using the constitutive relation. It remains to clarify here that the enhanced volumetric strain should be applied separately to each material in the domain of the problem.

4.8.2 Numerical example on the enhanced volumetric strain procedure

To illustrate the performance of the enhanced volumetric strain procedure, the numerical prediction of the bearing capacity of a strip footing was analyzed. The problem domain has a height of $10\,[m]$ and a width of $40\,[m]$. The footing width is $6\,[m]$. In the numerical simulation of the problem, plane-strain conditions were assumed. Half symmetry of the domain was simulated and hence the mesh shown in Figure 4.19 has a width of $20\,[m]$ and footing width of $3\,[m]$. This mesh consists of 4800 tetrahedral element with 1722 node and 4800 particle, each element has one particle. The bottom of the domain was fully fixed and roller boundaries were prescribed at the sides. Nodes that are located underneath the footing were constrained in the horizontal direction to simulate a rough contact. Small deformation solution was assumed, i.e., no geometry update. Hence, particles were kept in their initial positions throughout the computations.

The soil on which the footing is located has a density of $\varrho = 1800\,[kg/m^3]$, a cohesion of $c = 100\,[kN/m^2]$, Young's modulus of $E = 200000\,[kN/m^2]$, Poisson's ratio of $\nu = 0.33$, a friction angle of $\phi = 0°$ and a dilation angle of $\psi = 0°$. Tresca model was adopted in the simulation of the problem. The footing pressure was increased incrementally from zero, each increment consisting of a $10\,[kN/m^2]$ increase. The convergence procedure of Section 4.9 was adopted to detect the quasi-static equilibrium at the end of each load step. A local damping of $\alpha = 0.75$ was used in the simulation. The application of the local damping is explained in Section 4.10. A Courant number of $\alpha_{Cour} = 0.98$ was adopted in this problem. The analytical solution of the problem is given according to Hill [90] as

$$q = (2 + \pi)\, c, \tag{4.121}$$

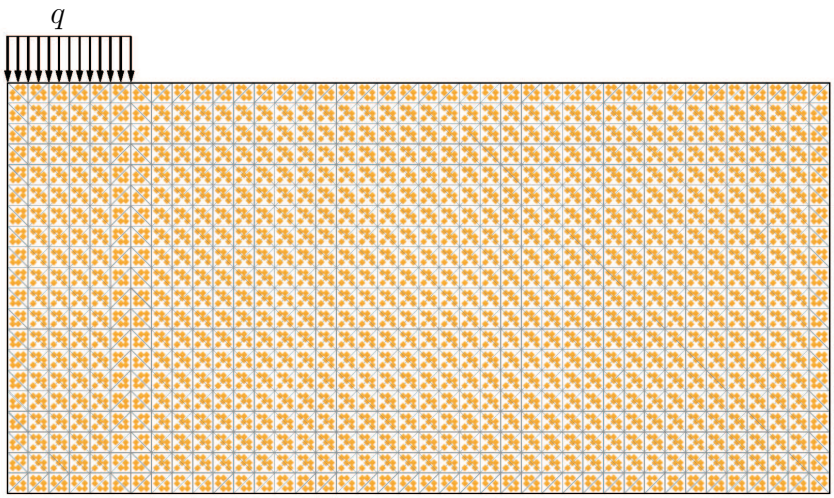


Figure 4.19: Mesh and particles distribution of the footing problem

in which q represents the footing pressure at collapse (bearing capacity) and c is the cohesion. Figure 4.20 compares the predicted load displacement curves to the analytical solution of the problem. The numerical predictions consist of a case with enhancement and another without enhancement of the volumetric strain. We can see the locking effect predicted when strain enhancement was not used. In this case, the numerical solution over predicts the stiffness because of the incompressibility constraints induced by the low-order elements. When adopting the enhanced volumetric strain procedure, the numerical solution predicts a collapse pressure of $520\,[kN/m^2]$ approximately. This value is however close to the pressure calculated from Equation 4.121, that is $514.16\,[kN/m^2]$. Figure 4.21(a) and (b) show the vectors of the incremental displacement and the contours of the accumulated horizontal displacements at the end of computations for the case of $\nu = 0.33$ when the enhancement procedure was adopted. The predicted failure mechanism is however consistent with the expectation.

To assess the performance of the enhanced volumetric strain procedure with nearly incompressible material, the previous computations were repeated with a Poisson's ratio of $\nu = 0.495$. The results are shown in Figure 4.22. It is obvious that the predicted bearing capacity is virtually identical to the analytical solution of the problem. It remains to mention here that this problem required a smaller load increment to converge. Hence, the load was increased by an increment of $2.5\,[kN/m^2]$ each load step. Therefore, the results of this case are closer to the analytical solution than the previous case.

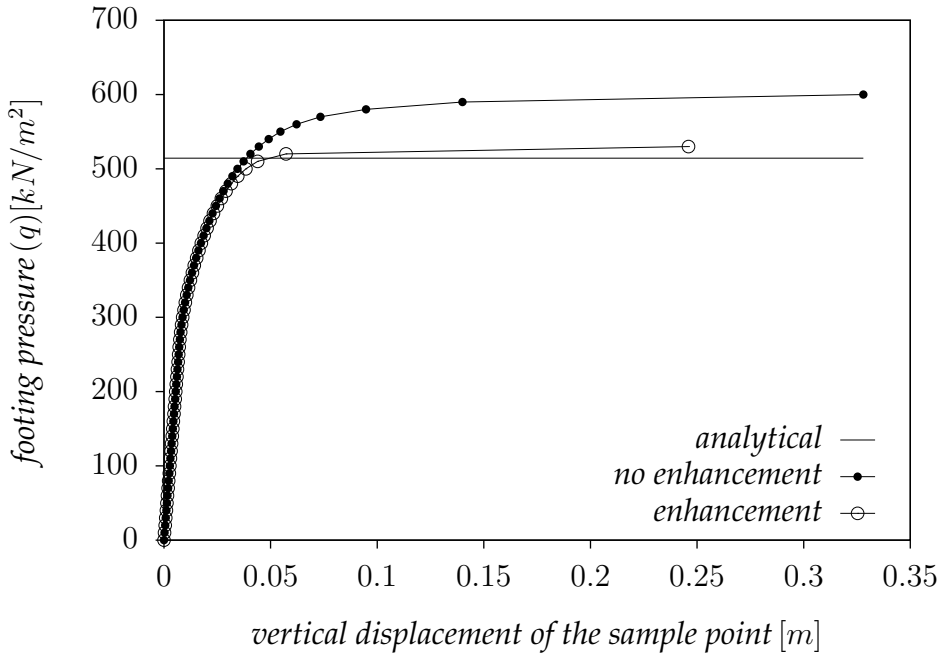
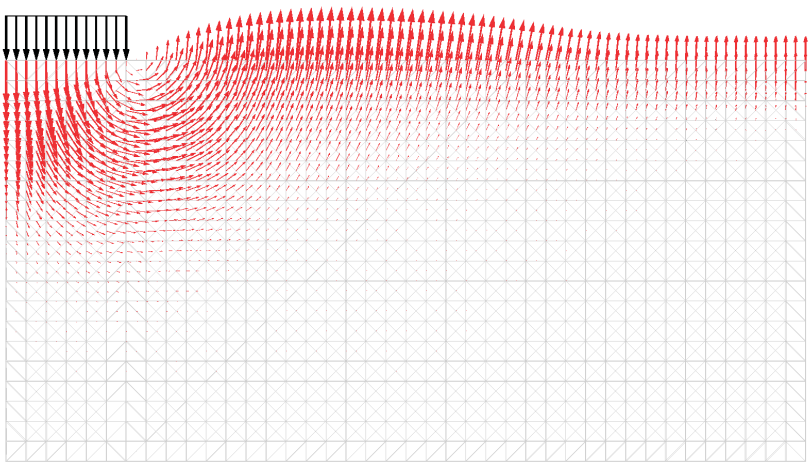
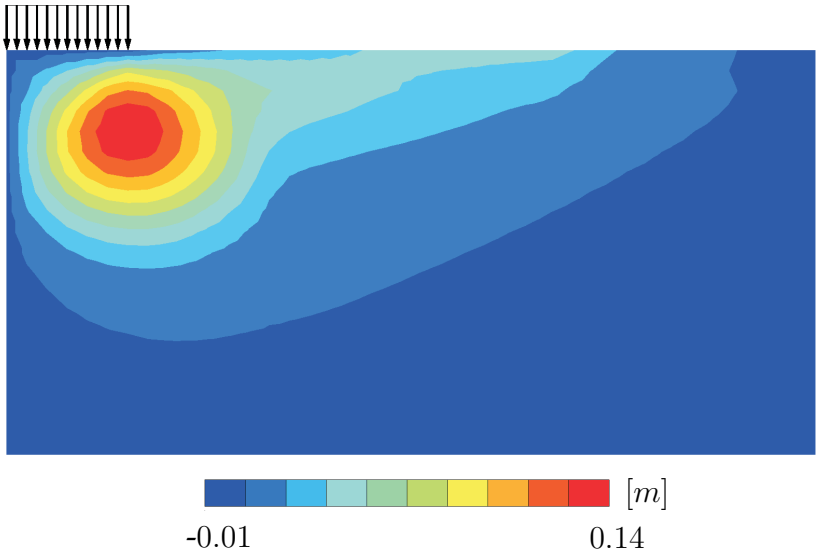


Figure 4.20: Comparison of the numerical predictions with the analytical solution of the bearing capacity of the strip footing problem with $\nu = 0.33$[a]

[a]The left top corner point of Figure 4.19 is selected as a sample point to plot the displacement.



(a)



(b)

Figure 4.21: Failure mechanism at the end of simulation for the case with strain enhancement and $\nu = 0.33$ (a) vectors of incremental displacements (b) contours of accumulated horizontal displacements

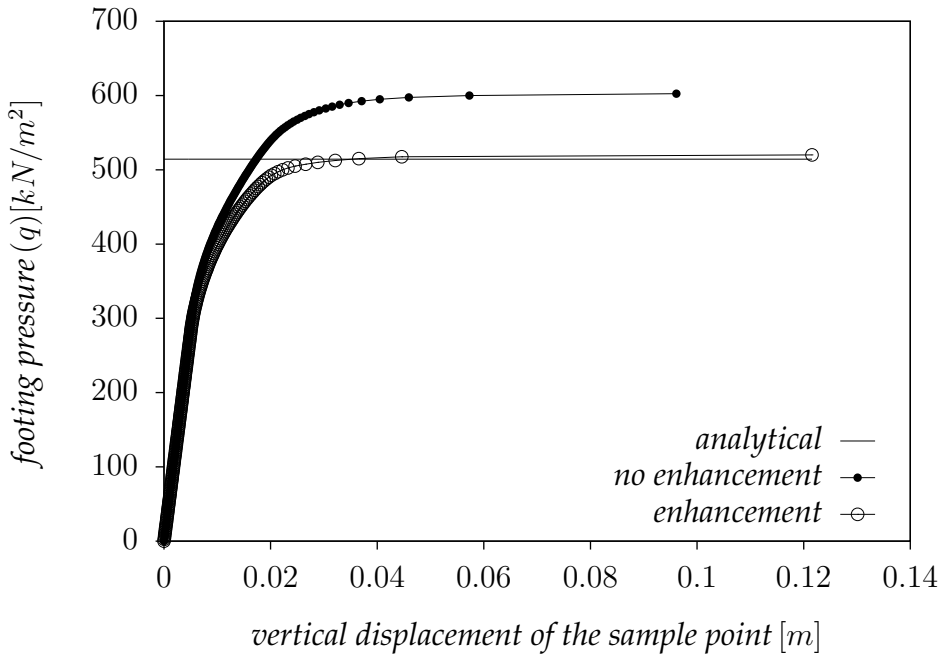


Figure 4.22: Comparison of the numerical predictions with the analytical solution of the bearing capacity of the strip footing problem with $\nu = 0.495$

4.9 Convergence to a quasi-static equilibrium

The dynamic MPM can also be adopted to model quasi-static problems, as demonstrated in the previous example. Therefore, a detection of the quasi-static equilibrium is required. Such quasi-static equilibrium is reached when both the out-of-balance forces and the kinetic energy of the system vanish. The out-of-balance convergence criterion is usually adopted in quasi-static formulation to detect equilibrium. Let us define a dimensionless force ratio $\mathcal{F}$ as

$$\mathcal{F} = \frac{\|\boldsymbol{F}^{ext} - \boldsymbol{F}^{int}\|}{\|\boldsymbol{F}^{ext}\|}. \tag{4.122}$$

Together with the out-of-balance criterion, we adopt another criterion to ensure that the kinetic energy of the system vanishes. Let us define $\mathcal{E}$, a dimensionless energy ratio, as

$$\mathcal{E} = \frac{KE}{W^{ext}}, \tag{4.123}$$

where KE denotes the kinetic energy of the system and W^{ext} is the work induced by the external forces. The kinetic energy is calculated as

$$KE = \frac{1}{2}\sum_{i=1}^{n_T} m_i\, \boldsymbol{v}_i^T \boldsymbol{v}_i \tag{4.124}$$

with m_i and $\boldsymbol{v}_i$ being the mass and the velocity vector of node i, respectively. The external work is calculated as

$$W^{ext} = W_0^{ext} + \Delta W^{ext}, \tag{4.125}$$

in which

$$\Delta W^{ext} = \Delta \boldsymbol{u}^T \boldsymbol{F}^{ext} \tag{4.126}$$

with $\Delta \boldsymbol{u}$ and $\boldsymbol{F}^{ext}$ being the global vectors of incremental displacements and external forces, respectively. The system is assumed to have reached quasi-static equilibrium when $\mathcal{F}$ and $\mathcal{E}$ reach a pre-defined tolerance. The experience of the author shows that a tolerance of 0.01 is sufficient for both criteria. This value will be adopted throughout this thesis.

Numerical example on convergence This example consists of a block of material subjected to external pressure at the top surface. The block has a shape of a cube with $1\,[m]$ edge length. The material of the block was assumed to be linear elastic with a density of $\varrho = 1800\,[kg/m^3]$, Young's modulus of $E = 10000\,[kN/m^2]$ and a Poisson's ratio of $\nu = 0$. The mesh of the problem was constructed of 6 tetrahedral elements with 8 nodes and 6 particles. The bottom of the mesh was fully fixed and the sides had roller boundaries. A pressure of $10\,[kN/m^2]$ was applied on the top surface of the block. A local damping of $\alpha = 0.75$ and a Courant number of $\alpha_{Cour} = 0.98$ were adopted in the simulation of this problem. The load was applied instantaneously and kept constant throughout the computations. The convergence to quasi-static equilibrium was detected by the force and energy criteria. Figure 4.23(a), (b) and (c) show forces, energy, work, force ratio and energy ratio with time. It is seen that, at convergence the force and energy ratio are bounded. However, the converged solution had a uniform distribution of vertical stresses with a value of $10\,[kN/m^2]$ compression. The displacement had a linear distribution with a value of $0.001\,[m]$ at the top surface in the direction of the applied pressure.

4.10 Local damping

For quasi-static problems, in which the steady state solution is of interest rather than the propagation of waves, the application of damping to the equation of motion plays an important role in the convergence to the steady state solution in a faster way, allowing a considerable reduction in the computational time. Viscous damping cab be used to achieve such convergence. However, this kind of damping introduces body forces that retard the steady state collapse and might influence the mode of failure [86]. Cundall [53] describes a local non-viscous damping to overcome the difficulty associated with the viscous damping. For any degree-of-freedom in the considered system, the local damping can be described as follows

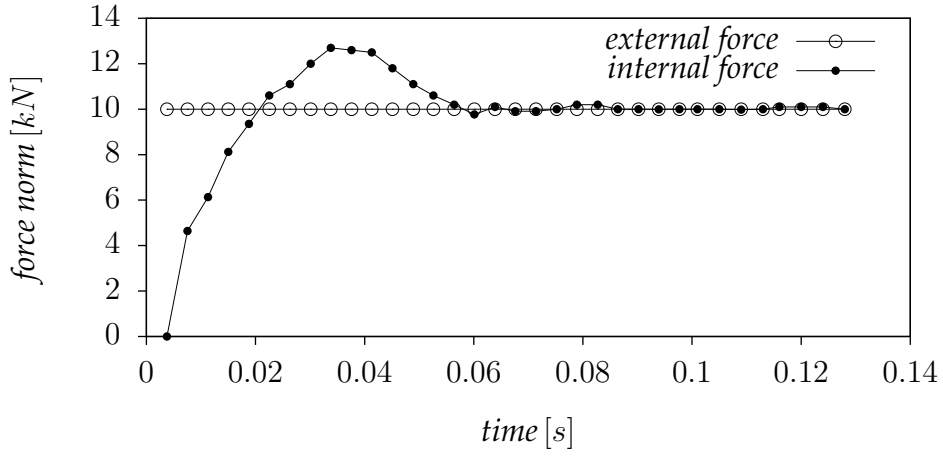
$$ma = f + f^{damp} \tag{4.127}$$

with

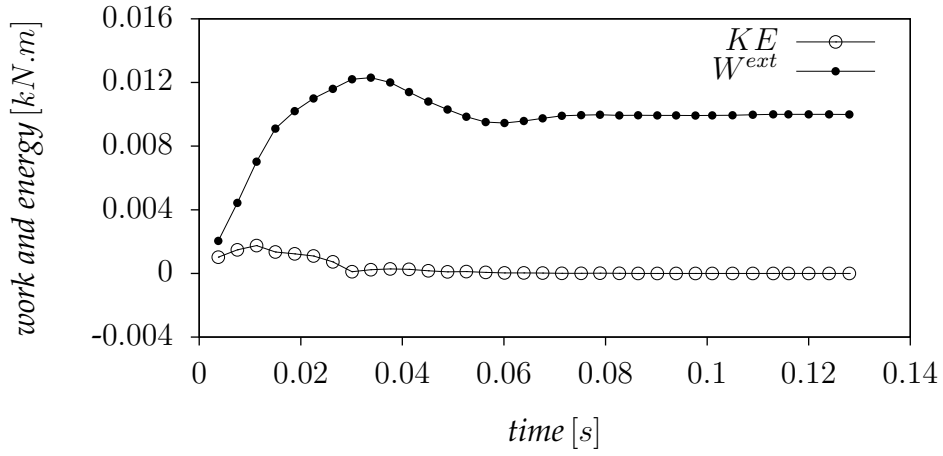
$$f = f^{ext} - f^{int}$$

being the out-of-balance force and

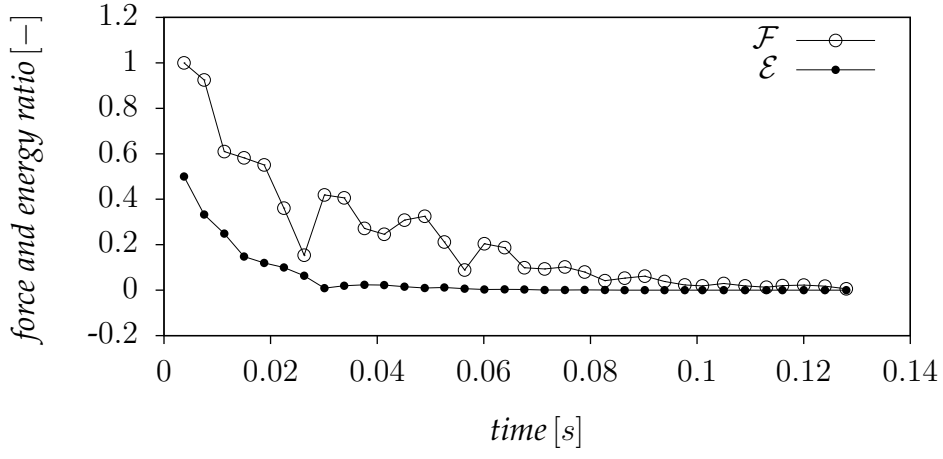
$$f^{damp} = -\alpha\,|f|\,sign\,(v) \tag{4.128}$$



(a)



(b)



(c)

Figure 4.23: Convergence to quasi-static equilibrium (a) norms of external and internal forces with time (b) external work and kinetic energy with time (c) force and energy ratio with time

is the damping force at the considered degree-of-freedom. The parameter α is a dimensionless damping factor and $sign\,(v)$ is defined for nonzero values of v as

$$sign\,(v) = \frac{v}{|v|}. \tag{4.129}$$

It is clear from Equation 4.128 that the damping force f^{damp} is proportional to the out-of-balance force and acts opposite to the direction of the velocity at the considered degree-of-freedom. This ensures that only accelerating motion, i.e., when $f \neq 0$ is damped and hence energy is dissipated effectively [138]. One can also observe that, unlike Rayleigh damping [137], the local damping is frequency independent as the damping factor α is a dimensionless parameter selected without any knowledge about the frequency of the system. When having a close look at Equation 4.127, one can observe that this equation may be put in the following form

$$\bar{m}a = f \tag{4.130}$$

with $\bar{m}$ is a scaled mass that is related to m and α as

$$\bar{m} = \begin{cases} \dfrac{m}{1-\alpha} & \text{if } sign\,(v) = sign\,(f) \\ \\ \dfrac{m}{1+\alpha} & \text{if } sign\,(v) \neq sign\,(f)\,. \end{cases} \tag{4.131}$$

The application of the local damping can alternatively be viewed as a procedure that scales the mass / density adaptively up or down depending on the signs of the out-of-balance force and the velocity of the considered degree-of-freedom.

The analysis of the last part of Section 3.4.2 is re-considered taking into account Equation 4.131. Now Equation 3.117 takes the following form

$$\Delta t_{crit} = \frac{h_{min}}{\sqrt{\frac{E^c}{\bar{\varrho}}}} \tag{4.132}$$

with $\bar{\varrho}$ being a scaled density,

$$\bar{\varrho} = \begin{cases} \dfrac{\varrho}{1-\alpha} & \text{if } sign\,(v) = sign\,(f) \\ \\ \dfrac{\varrho}{1+\alpha} & \text{if } sign\,(v) \neq sign\,(f)\,. \end{cases} \tag{4.133}$$

Reference [66] reports an important observation that the damping factor α can be directly related to the damping ratio ζ_d by the following equation

$$\alpha = \pi\zeta_d. \tag{4.134}$$

The reader is referred to reference [66] for a detailed analysis that leads to Equation 4.134.

4.10.1 Numerical example on damping

In this example we consider the spring-mass system of Section 3.5 with the same dimensions, mesh and material properties. The only difference is that in this example we used MPM with single particle per each active element in the initial configuration. The analytical solution of the single degree-of-freedom spring-mass-dashpot system is available in many books, e.g., [137]. It is given as

$$\hat{v}\left(t\right) = \frac{g}{\omega_n} \frac{e^{-\zeta_d \omega_n t} \sin\left(\omega_n t \sqrt{1-\zeta_d^2}\right)}{\sqrt{1-\zeta_d^2}}. \tag{4.135}$$

In this example, the natural frequency was $\omega_n = 10\,[rad/s]$, see the properties in Section 3.5. For comparison, a damping ratio of $\zeta_d = 0.05$ was used which corresponds to a local damping factor of $\alpha \approx 0.157$, see Equation 4.134. Figure 4.24 compares the numerical solution of the problem to its analytical solution. Scrutiny of the numerical and analytical solutions indicates that they are comparable, illustrating the performance of the local damping procedure.

4.10.2 Numerical example on damping layers

The idea was that the local damping can be used as a substitute for the viscous boundary. For this, we analyzed a column of $1\,[m]$ height to study the effectiveness of different damping layers. A compression surface traction of $1\,[kN/m^2]$ was applied on the top

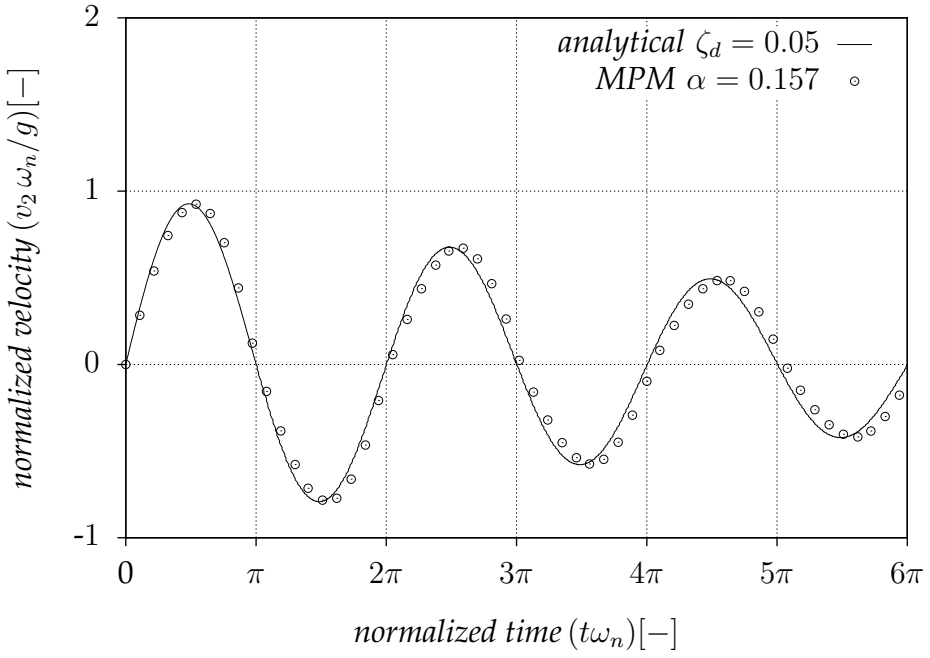


Figure 4.24: Comparison of the numerical and analytical solutions of the spring-mass-dashpot system

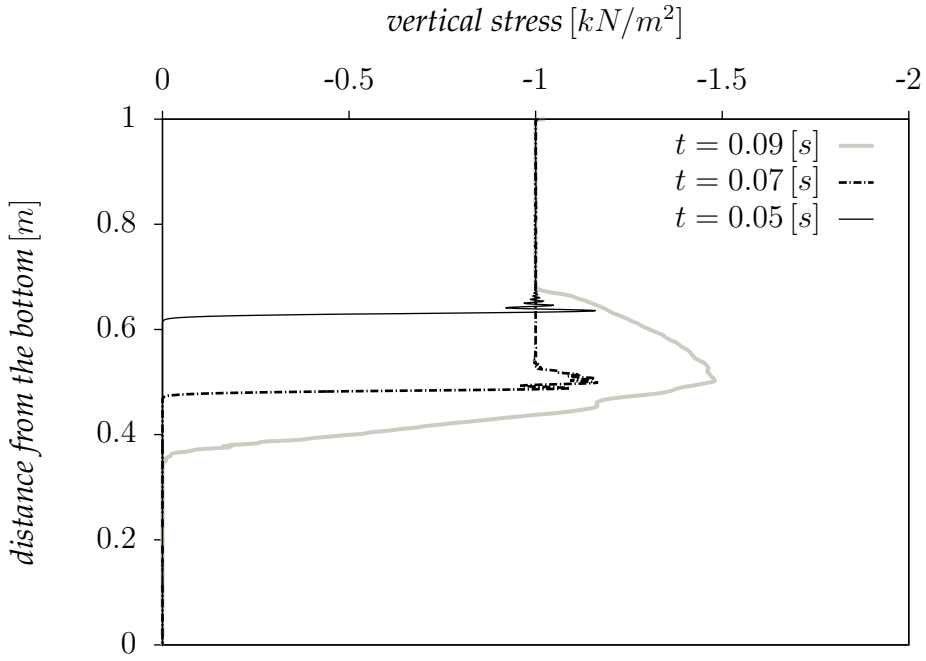


Figure 4.25: Early reflection of the wave in the two-layers column

surface of the column. The column has a density of $\varrho = 1800\,[kg/m^3]$, Young's modulus of $E = 100\,[kN/m^2]$ and a Poisson's ratio of $\nu = 0$. The column was discretized with 1000 row of elements along the height, in which each row contains 6 tetrahedral elements. The total number of particles was 6000, each element having a single particle. Roller boundaries were prescribed along the sides and the bottom was fully fixed. The upper half of the column was assumed to have no damping, whereas the lower half was assigned a local damping factor of $\alpha = 0.25$. We can see from Figure 4.25 that, once the wave reaches the interface between the upper and lower layers, it starts to reflect. This means that this interface acts as an artificial constraints that cause the early reflection. Indeed, we can see from Equations 4.127 and 4.128 that the local damping introduces forces that act in an opposite direction of the momentum. Hence, we do not advocate the use of local damping to replace the viscous boundary.

4.11 Interaction of deformable solid continuums

In geomechanics, it is common to deal with problems involving soil-structure interaction, e.g., pile driving in soil. We already indicated, that MPM is naturally capable of handling non-slip contact between different bodies without any special algorithm and interpenetration between different bodies cannot occur [165]. However, when solid continuums come into contact, in most cases frictional sliding occurs at the contact surface. To model such sliding and interaction, a contact algorithm that allows relative motion at the interface between the contacting bodies is required.

In this section, we provide an overview of different contact algorithms and validate

the implementation of the algorithm adopted in this thesis. The formulation and the solution procedure of the algorithm are detailed in Appendix A

4.11.1 Contact algorithms in MPM

Recently, Beuth [24] extended the implementation of interface elements from Lagrangian FEM to a quasi-static MPM. Such interface elements are placed at the surface where different bodies are expected to be in contact during deformation. An interface element has pairs of surface nodes that allow sliding and relative motion between the contacting bodies. Such sliding is modeled by relating the relative displacements between the surface nodes to the contact stresses through a constitutive equation [24]. Although the procedure is proven to be capable of handling large deformation and large sliding at the contact surface, it needs full implementation of surface interface elements adding considerable complexity to the MPM algorithm. Furthermore, a consequence of using a contact algorithm that requires a special contact element is that the contact surface has to be pre-defined and the contact elements have to be placed at that surface. In addition, using contact elements in an explicit dynamic code might effect the size of the time step as the contacting bodies exchange stress waves through the contact surface, which has virtually no thickness. For these reasons, we do not advocate the concept for dynamic MPM. In the following we consider a brief overview of the development of some MPM contact algorithms that do not require a special contact element.

York II et al. [196] proposed a slight modification to the non-slip MPM algorithm allowing for free separation of the contacting bodies. They proposed that when different bodies approach each other at contact points, the non-slip MPM algorithm is used at these points so that each contact point will move with a single valued velocity and hence interpenetration between the contacting bodies is prevented. When the bodies are separating from each other, no constraint is applied at the contact surface. Hence, the algorithm does not relate the relative movement between the contacting bodies to a constitutive equation, i.e., it is unable to model frictional contact between different bodies.

A more appropriate contact algorithm in the framework of dynamic MPM was developed by Bardenhagen et al. [9]. They developed a frictional contact algorithm to model interaction and frictional sliding between different bodies. The algorithm also allows for separation between bodies, but does not allow for interpenetration. The strength of the algorithm is that it detects the contact surface automatically, i.e., a pre-definition of the contact surface is not required and a special element to model the contact is not required as well. Furthermore, the algorithm is shown to be efficient in modeling interaction between solid bodies as well as shearing in granular materials [12].

Hu and Chen [92] adopted another algorithm that allows free separation and prevents interpenetration. They call their scheme a multi-mesh mapping scheme. In this scheme, each continuum lies in its own background mesh. Interpenetration is prevented by calculating the normal velocity of a contact point using the momentum of the coupled bodies, whereas the tangential velocity is calculated from the momentum of the single body. Although this algorithm is efficient in modeling separation between contacting bodies, it does not consider frictional contact between different bodies. Pan et al.

[135] improved the multi-mesh scheme by applying Coulomb friction in the tangential direction. Huang et al. [95] proposed a general formulation of the contact problem in the framework of MPM. They presented two methods to implement the contact in Lagrange multiplier form. The methods are applied to simulate the penetration of a steel ball and the perforation of a thick plate.

Although there exist other contact algorithms developed for MPM (e.g. [194]), Bardenhagen's algorithm [9] is the one most used in MPM literature [3, 10, 11, 45]. This algorithm is adopted in this thesis to model contact problems. The reader is referred to Appendix A for the formulation of this algorithm and its extension to adhesive contact.

4.11.2 Sliding block

To validate the implementation of the contact algorithm, we analyzed the simple example of the sliding block shown in Figure 4.26. The block starts sliding once

$$w \sin(\vartheta) = r, \tag{4.136}$$

in which w is the weight of the block and r is the resistance tangential force that is written as

$$r = \mu w \cos(\vartheta) + f_{adh} \tag{4.137}$$

with $f_{adh} = aA$ being the force induced by the adhesion a, A is the area of the contact surface and μ is the coefficient of friction. Substituting Equation 4.137 into Equation 4.136 and solving for ϑ yields

$$\vartheta = \tan^{-1}\left(\frac{\mu\sqrt{\mu^2 w^2 + w^2 - f_{adh}^2} + f_{adh}}{-\mu f_{adh} + \sqrt{\mu^2 w^2 + w^2 - f_{adh}^2}}\right). \tag{4.138}$$

Let us consider a block of $0.25\,[m]$ length, $0.25\,[m]$ height and $0.05\,[m]$ thickness. The material of the block is linear elastic with a density of $\varrho = 100\,[kg/m^3]$, Young's modulus of $E = 1000\,[kN/m^2]$ and a Poisson's ratio of $\nu = 0$. The mesh of the problem is shown in Figure 4.27. Computations were carried out to find initial stresses due to gravity in the initial configuration, i.e., $\vartheta = 0$. Then ϑ was increased in steps, and at the end of each step equilibrium was found due to the new rotation. For the case of frictional

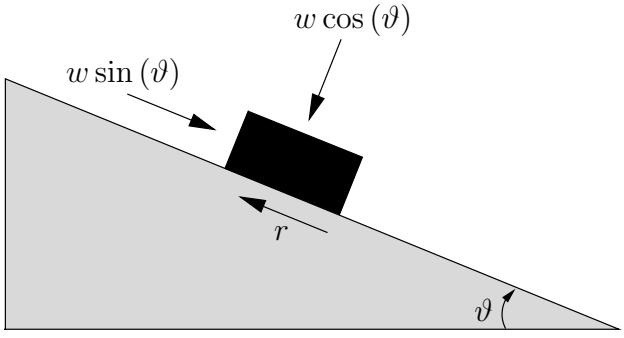


Figure 4.26: Forces acting on a sliding block

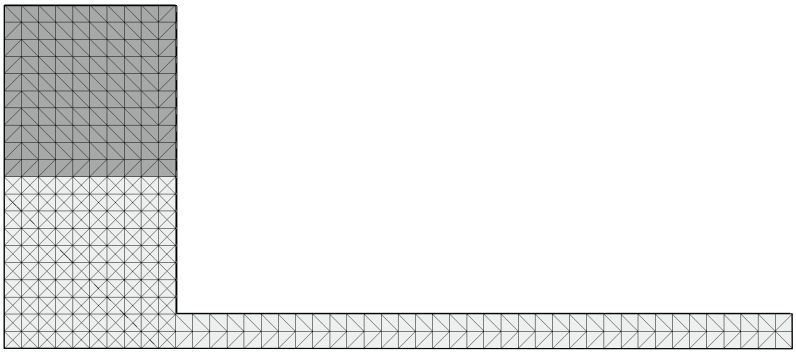


Figure 4.27: Mesh of the sliding block problem

contact, i.e., $a = 0$ the block starts sliding once $\vartheta = \tan^{-1}(\mu)$. Adopting $\mu = 0.5$ results in $\vartheta = 26.56°$ approximately. The MPM simulation resulted almost the same angle of sliding, see Figure 4.28. Figure 4.29 shows the block after sliding. We can clearly see the ability of the algorithm to simulate separation between different bodies.

Other computations were carried out with $\mu = 0$ and $a = 2\,[kN/m^2]$. We used a density of $\varrho = 1800\,[kg/m^3]$ to get approximately the same sliding angle as for the frictional contact case. According to Equation 4.138, the block starts sliding once $w \sin(\vartheta) = f_{adh}$. Although not shown here, the results are almost identical to those of Figure 4.28.

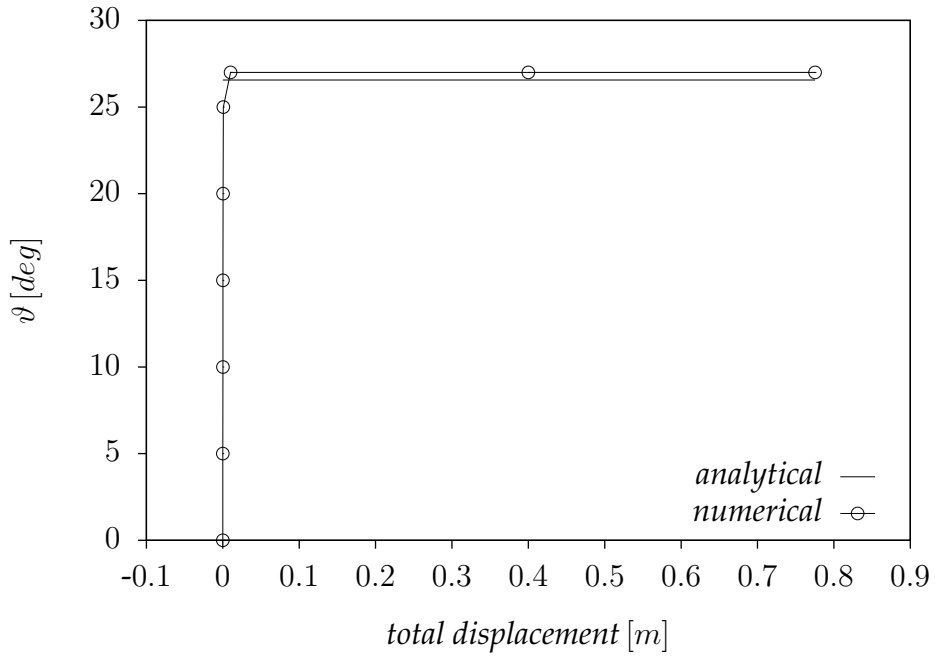


Figure 4.28: Analytical and MPM results for the block problem

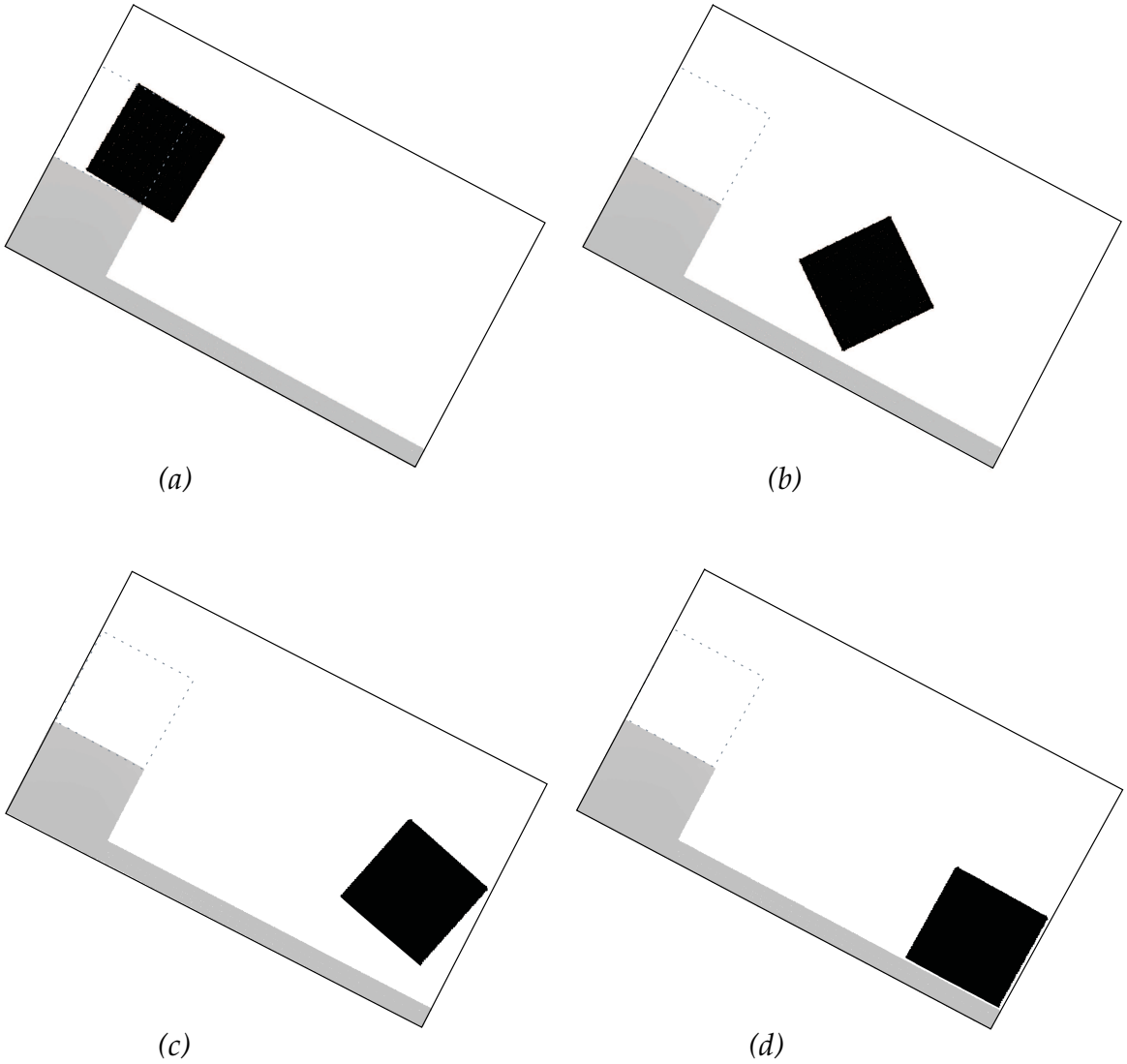


Figure 4.29: Particles at different times for the case of frictional contact

4.12 MPM validation: penetration example

This section consists of a validation example of the current MPM implementation. In this example, we simulated the penetration of a rigid body into a sand layer. The measurement data of the problem is available in reference [58]. In the test, the penetration of a concrete hemisphere into a layer of Hokksund sand was measured with time. The radius of the concrete hemisphere is $0.1525\,[m]$ and it has a total mass of $27\,[kg]$ including the mass of the steel casting attached to it. The hemisphere was dropped from a height of $5\,[m]$ above the surface of the sand layer. In the simulation, the hemisphere was assumed just touching the surface of the sand layer and assigned an initial velocity of $10\,[m/s]$. This value corresponds to the velocity of a free falling object from a height of $h = 5\,[m]$, calculated as $v = \sqrt{2gh}$.

Considering the symmetry of the problem, a sector of 20° was discretized in the three-dimensional MPM simulation. A domain of $5\,[m]$ height and $5\,[m]$ width was considered for the sand layer. The fine mesh region is shown in Figure 4.30. The Mohr-Coulomb elastic-perfectly plastic model was used in the simulation of the sand layer of a density of $\varrho = 1631\,[kg/m^3]$, a cohesion of $c = 0\,[kN/m^2]$, a Young's modulus of $E = 55000\,[kN/m^2]$, a Poisson's ratio of $\nu = 0.2$, a friction angle of $\phi = 58.7^\circ$ and a dilation angle of $\psi = 0^\circ$. These properties of the Hokksund sand were taken from reference [58], in which results of numerical simulation of the problem using Drucker-Prager model with isotropic hardening and a dilation angle of $\psi = 6^\circ$ are reported. The initial stresses due to the self

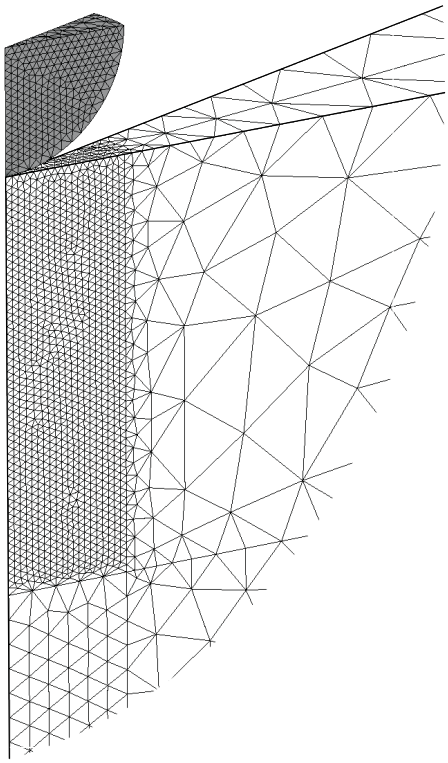


Figure 4.30: A close up of the fine mesh region of the hemisphere penetration problem

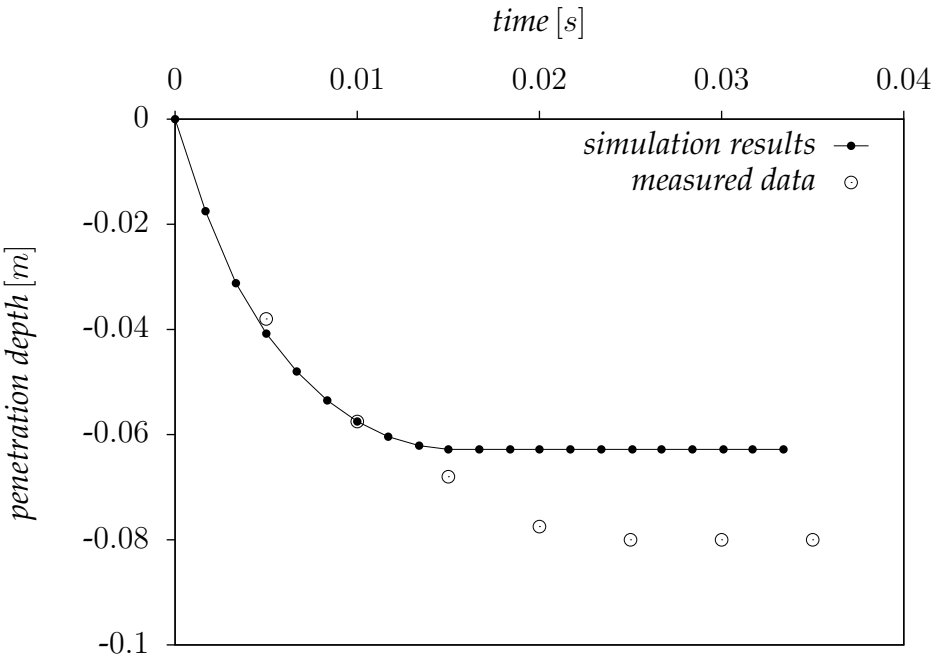


Figure 4.31: Comparison of the simulation results and the measured data

weight of the sand layer were calculated using K_0-procedure.

The frictional contact algorithm of Appendix A was used to simulate the interaction between the concrete hemisphere and the sand. A friction coefficient of $\mu = 0.25$ was used as proposed in reference [58]. The hemisphere was simulated as a stiff elastic body with a Young's modulus of $E = 500000\,[kN/m^2]$ and a Poisson's ratio of $\nu = 0$. A density of $\varrho = 3650\,[kg/m^3]$ was assigned to the hemisphere to resolve its total mass. The moving mesh concept of Section 4.5 was adopted in the simulation, with the zone from the bottom to a height of $4\,[m]$ being the compressed mesh zone and the upper zone being the moving mesh zone. Of course, no stretched mesh zone was needed because the upper boundary was open. Roller boundaries were prescribed along the sides of the domain and the bottom was fully fixed. The domain was set to be large enough to ensure that reflections from boundaries during computations do not disturb the results. The computations were run for $0.035\,[s]$, which is reasonably short time that one can disregard the viscous boundaries.

Figure 4.31 compares the results of the numerical simulation to the measured data and Figure 4.32 shows particles distribution after reaching the final penetration, with principal stresses directions around the hemisphere being shown in Figure 4.33. The numerical predictions are seen to be reasonable. The differences between the numerical predictions and the measured data can most likely be attributed to the use of the simple elastic-perfectly plastic Mohr-Coulomb model.

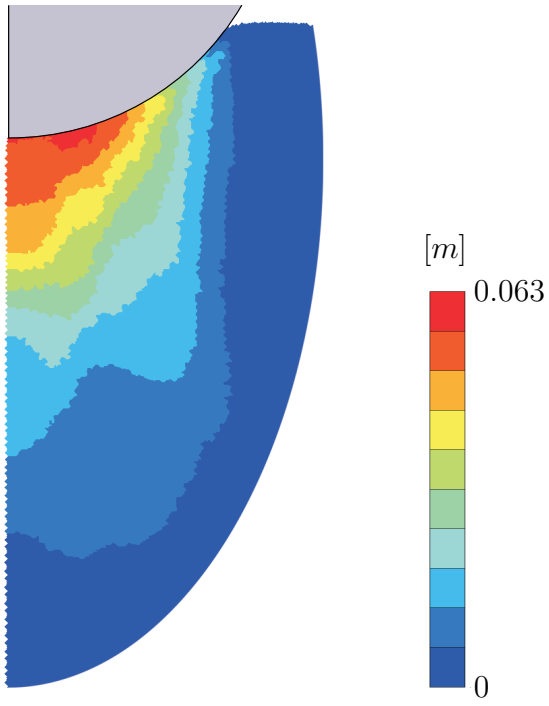


Figure 4.32: Total displacement of particles in the final configuration

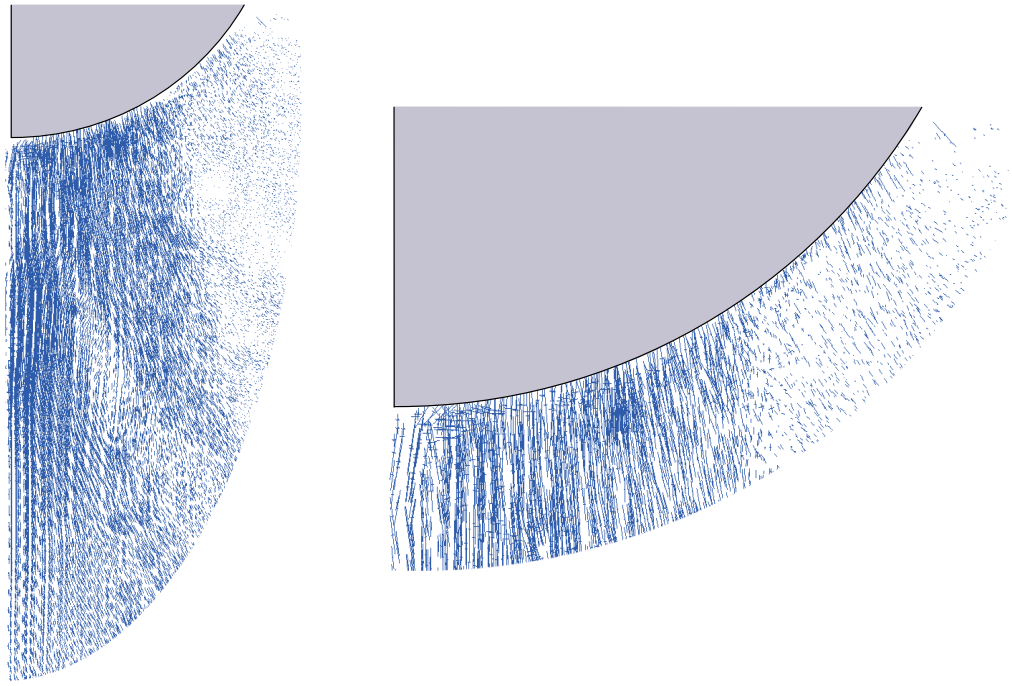


Figure 4.33: Principal stresses directions around the hemisphere (left) with a close up (right)

Chapter 5

Application of MPM to practical problems in geomechanics

This chapter consists of the application of the material point method to geomechanical problems involving large deformation and movement of soil mass. The examples of this chapter show the capability of the material point method in modeling practical engineering problems. In Section 5.1, the dynamic collapse of a tunnel face is analyzed. The instability of a slope consisting of frictional material is discussed in Section 5.2 and the problem of a dynamic anchor penetration is presented in Section 5.3.

5.1 Dynamic collapse of tunnel face

Tunnel collapse can pose a danger to life and property. Both the life of the workers standing in front of the tunnel face and individuals at the ground surface might be in danger. An example of tunnel collapse in Munich is shown in Figure 1.2. The aim of this analysis was to estimate the real collapse time of a tunnel and the collapsing of the ground. The pattern of the tunnel face collapse was also investigated. The dimensions of the computational domain were $5\,[m]$ height, $10\,[m]$ width and $7.5\,[m]$ depth. The tunnel was assumed to have a circular profile with a radius of $2.5\,[m]$. The depth of the tunnel was $5\,[m]$. The mesh of the problem is shown in Figure 5.1. It consists of 17035 tetrahedral elements with 11816 of them being initially active. Each active element in the initial configuration had 10 particles and hence the domain had in total 118160 particles.

A fine mesh was created around the tunnel opening where the soil is expected to flow. Roller boundaries were prescribed to the sides of the domain and the upper surface was free from constraints. Kelvin-Voigt elements were prescribed normal to the base with $\delta = 3.0\,[m]$. The curved surface in front of the tunnel face was considered to be fully fixed which automatically provided rough contact between the soil and the lining (fixed surface). The Mohr-Coulomb model was used in the simulation of the soil. The soil had a density of $\varrho = 16\,[kg/m^3]$, a cohesion of $c = 1\,[kN/m^2]$, a Young's modulus of $E = 10000\,[kN/m^2]$, a Poisson's ratio of $\nu = 0.3$ and a friction angle of $\phi = 25^\circ$. Considering the symmetry of the problem, only a half was considered in the computations.

For the first stage of computation, stresses due to the self weight of the soil were computed. A local damping of $\alpha = 0.75$ was used in this stage to get fast convergence to quasi-static equilibrium. The converged solution was detected by the convergence criteria of Section 4.9. During this stage equilibrium was maintained assuming roller boundaries at the face of the tunnel. In the second stage, these supports were removed

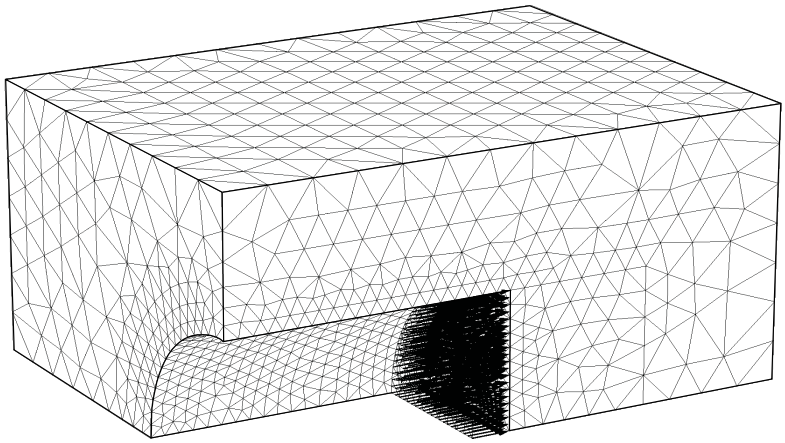
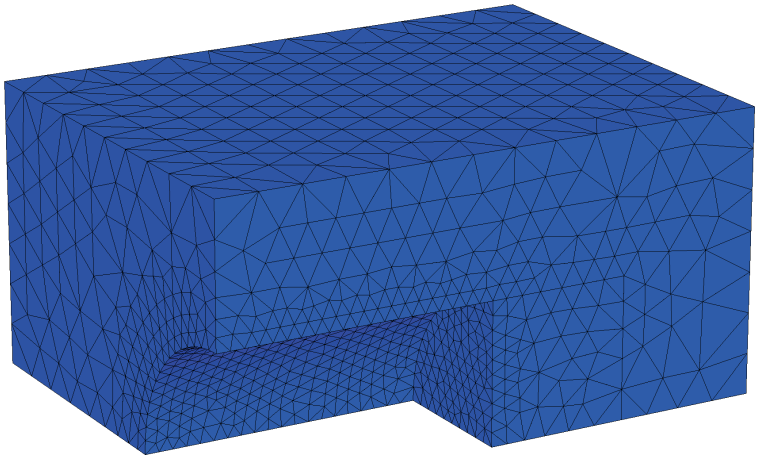


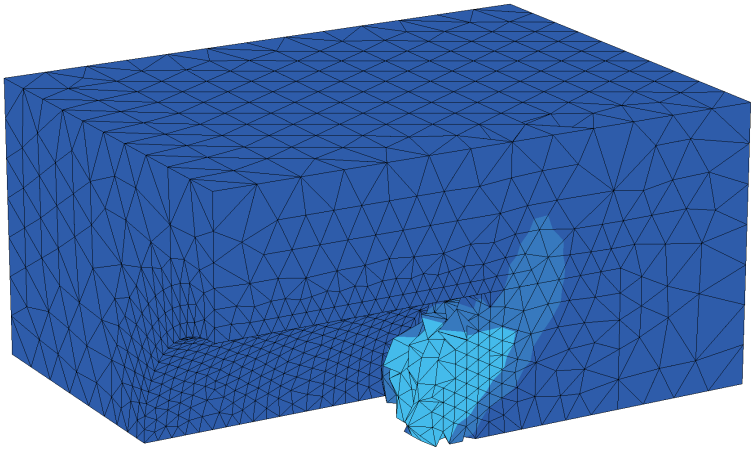
Figure 5.1: Mesh of the tunnel problem

suddenly to trigger instability and hence the collapse of the tunnel face. A Courant number of $\alpha_{Cour} = 0.98$ was adopted in the computations.

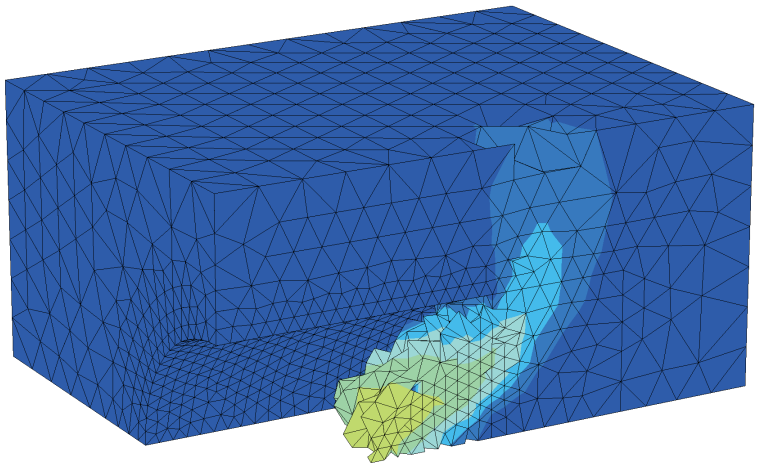
The deformation sequences of Figure 5.2 show the progressive collapse of the tunnel face and the flow mechanism of the soil with time. In this figure, we see the shading of the total displacements. This shading was plotted using the nodal displacements that were extrapolated from the particles displacements. We can also see the relatively fine mesh region where the soil is expected to flow. The time required to reach the final static equilibrium was found to be $t = 9.21\,[s]$. The final pattern of the collapse is shown in the last picture of Figure 5.2. In this picture, one can also see that the disturbance reached the ground surface forming a crater. Figure 5.3 shows particle displacements after reaching the final collapse pattern. The resulting slope at the final configuration was found to be 26°, which is a little higher than the friction angle of the considered soil due to the cohesion. Figure 5.4 shows the displacement history of a sample particle that was initially located at the top corner of the tunnel face.



$t = 0\,[s]$



$t = 0.6\,[s]$



$t = 1.0\,[s]$

Figure 5.2: Progressive collapse of the tunnel with time

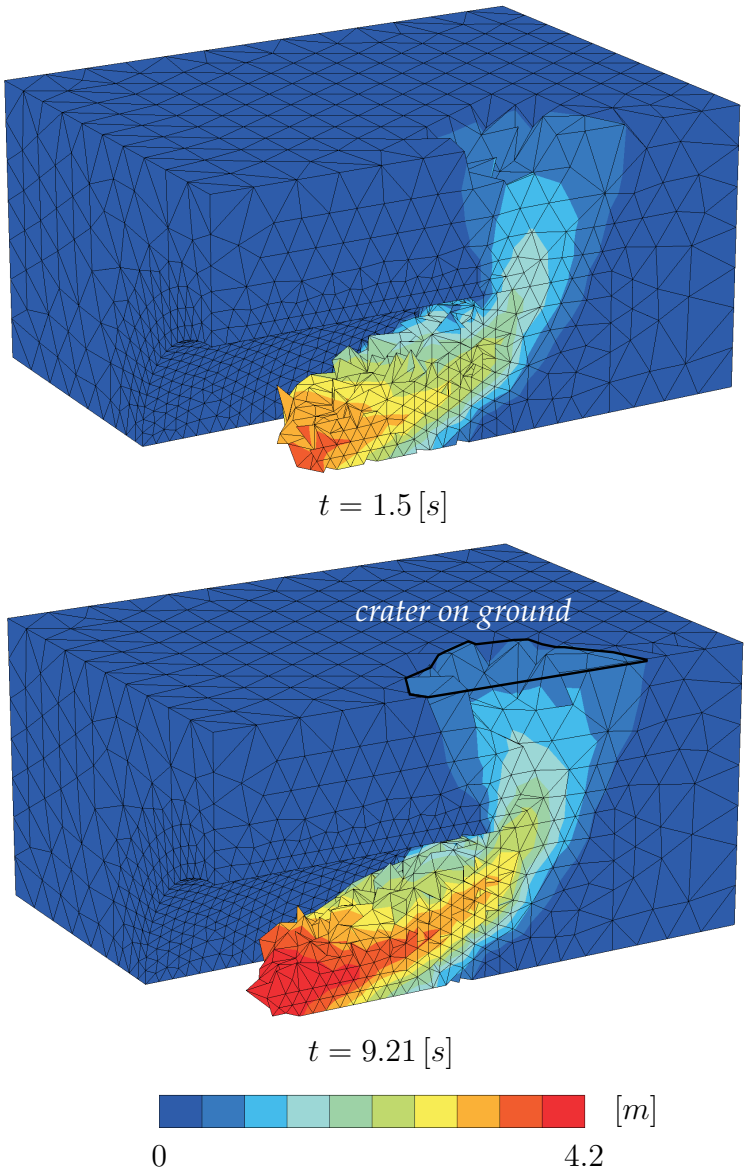


Figure 5.2: Continued

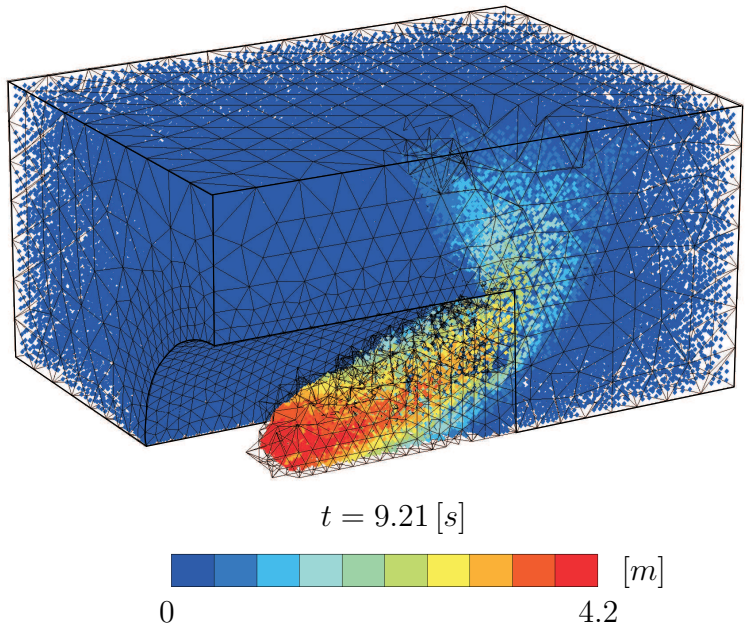


Figure 5.3: Particles displacements in the final stage of the collapse

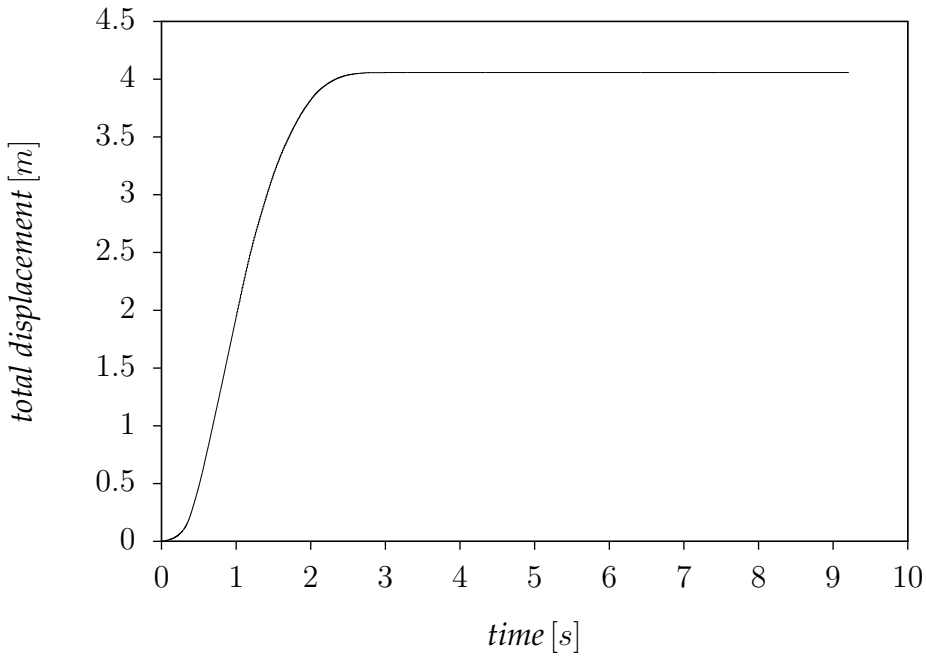


Figure 5.4: Total displacement of a particle initially located at the top corner of the tunnel face

5.2 Slope instability

The instability of a slope consisting of frictional material is analyzed in this example. The soil of the slope was assumed to behave according to the elastic-perfectly plastic Mohr-Coulomb model, having a modulus of elasticity of $E = 200\,[kN/m^2]$, a Poisson's ratio of $\nu = 0.33$, a cohesion of $c = 1\,[kN/m^2]$ and a friction angle of $\phi = 25°$. The slope had a height of $1\,[m]$, a base length of $2.1\,[m]$, a thickness of $0.1\,[m]$ and an inclination angle of $45°$. The mesh of the problem is shown in Figure 5.5. A relatively fine mesh was adopted along the base of the mesh where contact between the soil and the rigid boundary occurred. The number of tetrahedral elements in the initial configuration was 11091 with 10 particles per element. Plane-strain condition was assumed with roller boundaries along the left side of the mesh.

A local damping of $\alpha = 0.75$ was adopted. The frictional contact algorithm of Appendix A was used to model the interaction between the soil and the base boundary of the mesh. A friction coefficient of $\mu = 0.5$ was assumed at the contact surface. A Courant number of $\alpha_{Cour} = 0.98$ was adopted in the computations. The deformation of the slope was obtained by increasing the unit weight of the soil to a value of $\gamma = 160\,[kN/m^3]$ in 40 steps.

Figure 5.6 shows the deformed configuration of soil particles for various values of the unit weight. The scale ranges from 0 to $0.6\,[m]$, with extremes corresponding to blue and red, respectively. We can see the movement of the soil on a curved slip surface. Figure 5.7 shows the accumulated shear strains at the final stage of deformation. We can see the developed shear band along the slip surface. Vertical effective stresses at the last stage of deformation are shown in Figure 5.8. The load-displacement curve corresponding to the crest point is shown in Figure 5.9. More or less, elastic behavior is observed up to a unit weight of approximately $28\,[kN/m^3]$.

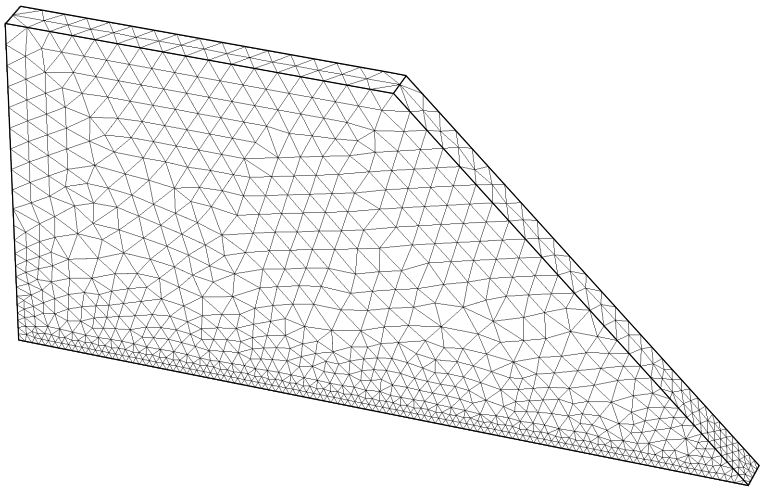


Figure 5.5: Mesh of the slope problem

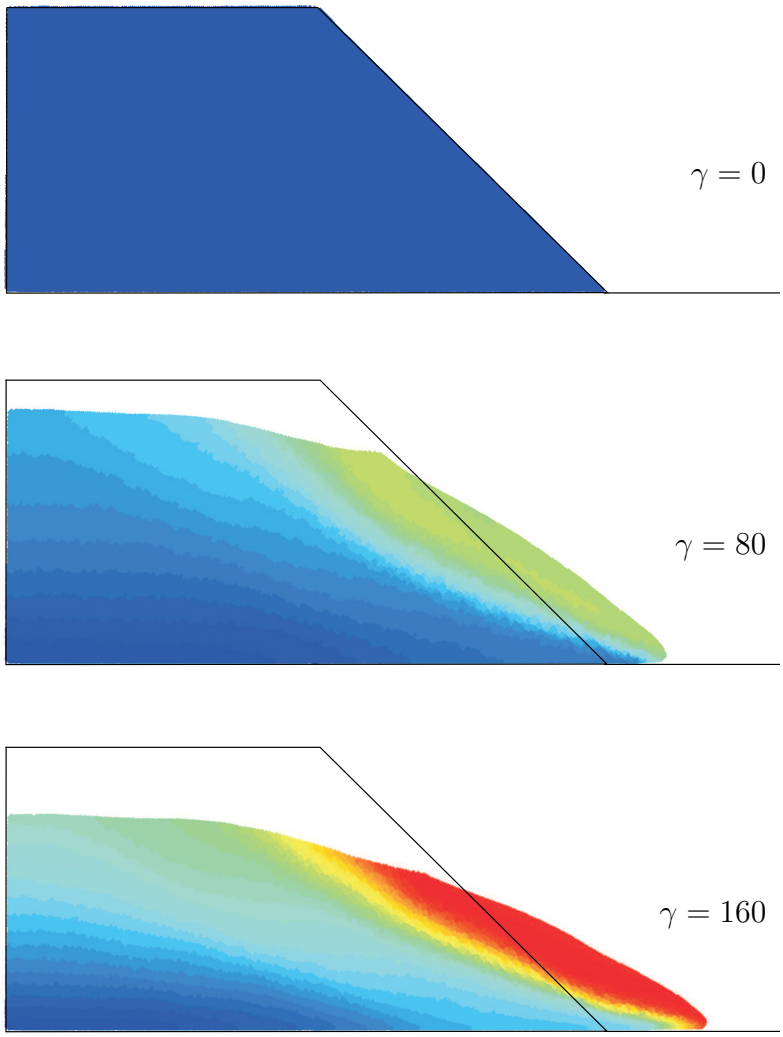


Figure 5.6: Total displacement of particles (0 to $0.6\,[m]$) for various values of soil unit weight $[kN/m^3]$

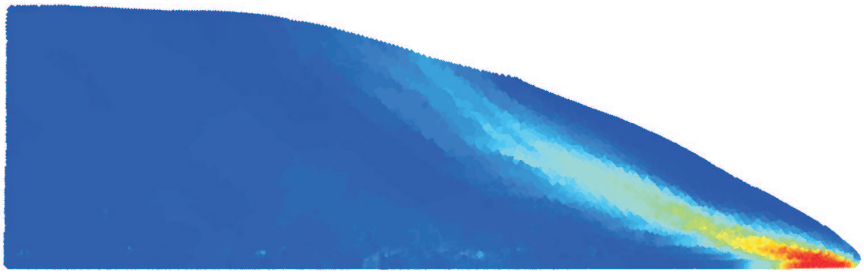


Figure 5.7: Accumulated shear strains at $\gamma = 160\,[kN/m^3]$

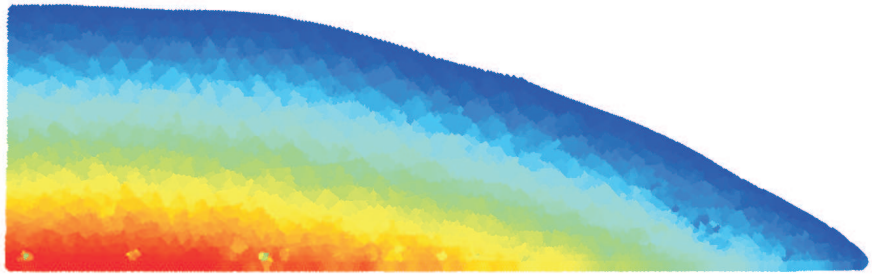


Figure 5.8: Vertical effective stresses (-145 to $0\,[kN/m^2]$) at $\gamma = 160\,[kN/m^3]$

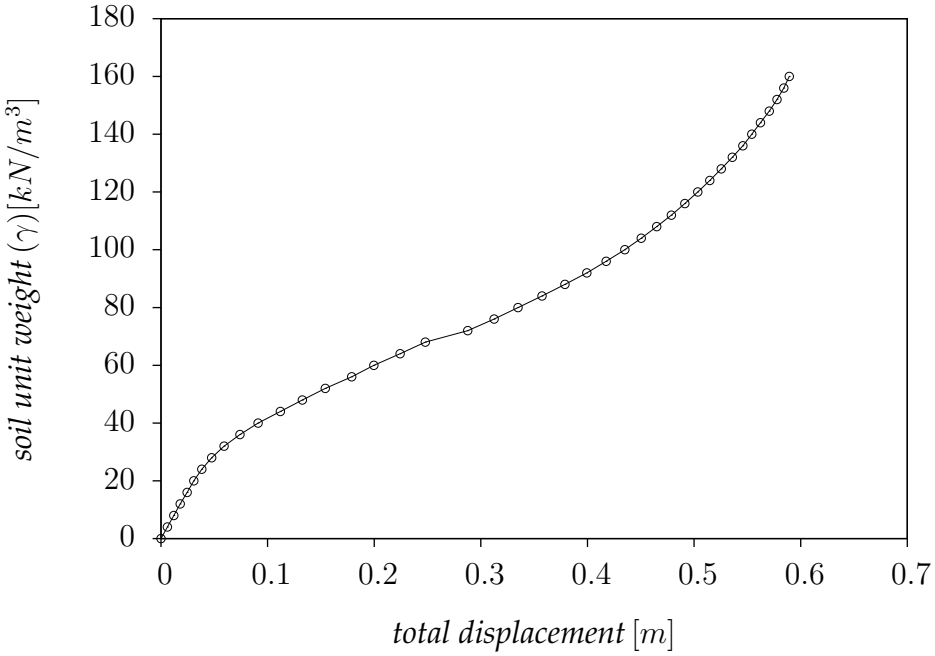


Figure 5.9: Load-displacement curve for a particle initially located at the crest of the slope

5.3 Installation of dynamic anchor

The (drained) analysis of the dynamic anchor of Figure 5.10 is presented in this section. These torpedo-shaped anchors are used in the offshore oil and gas industry as a cost effective anchoring solution [156]. The anchor is dropped from a particular height and penetrates the seafloor by the kinetic energy gained during the free fall. A similar problem was modeled by Sturm and Andresen [156] using a quasi-static procedure. In the present study, a fully dynamic penetration process is presented.

The material of the stiff elastic anchor was assumed to have a Young's modulus of $E = 500000\,[kN/m^2]$, a Poisson's ratio of $\nu = 0$ and a density of $\varrho = 2500\,[kg/m^3]$. The Mohr-Coulomb elastic-perfectly plastic model was used in the simulation of the soil, which was assumed to have a density of $\varrho = 1600\,[kg/m^3]$, a cohesion of $c = 5\,[kN/m^2]$, a Young's modulus of $E = 10000\,[kN/m^2]$, a Poisson's ratio of $\nu = 0.33$ and a friction angle of $\phi = 30°$. The mesh of the soil layer had a $5\,[m]$ width and a $17\,[m]$ height, see Figure 5.11. The moving mesh concept of Section 4.5 was adopted in the simulation. The domain from the bottom of the mesh to a height of $16\,[m]$ represented the compressed mesh zone, whereas the moving mesh zone was represented by the upper domain with relatively fine mesh around the tip of the anchor. Although not shown, the domain from the soil surface to the anchor head initially had inactive elements, representing particle storage domain. The anchor had a diameter of $0.3\,[m]$ and a length of $10\,[m]$. Taking advantage of the symmetry of the problem, only a sector of $20°$ was discretized. The mesh was constructed from 23031 tetrahedral elements with 21557 of them being initially active with 4 particles per each.

The drop height of the anchor was assumed to be $h = 97\,[m]$ with a corresponding free falling velocity of $v = \sqrt{2gh} = 44\,[m/s]$. In the simulation, this value was used as initial velocity of the anchor. The initial stresses due to the weight of the soil layer were calculated using K_0-procedure with $K_0 = 0.5$. To avoid wave reflection, Kelvin-Voigt elements with $\delta = 5\,[m]$ were prescribed on the bottom and the radial boundary of the mesh.

Computations were carried out with different frictional coefficients of the contact surface between the anchor and the soil. Figure 5.12 shows the computed penetration depth for different values of μ plotted over time. For fully rough contact ($\mu = 1$), the final penetration depth was found to be $9.5D$, with D being the diameter of the anchor. In the very beginning, the anchor resistance was more or less linear with penetration, but later the penetration slowed down because of the loss of the kinetic energy of the anchor. As the anchor was penetrating deeper into the soil, the resistance was getting higher because the contact between the anchor and soil was increasing and because the shear resistance of the soil increased with depth. The ultimate penetration depth obviously increased with smaller friction coefficients. The largest penetration depth ($18.5D$) corresponds to the case of smooth contact ($\mu = 0$), as one might expect. The principal stresses directions are shown in Figure 5.13 with a close up around the anchor tip being shown in Figure 5.14. The right half represents the case of smooth contact ($\mu = 0$) and the left is the case of fully rough contact ($\mu = 1$) after a penetration of $9.5D$. We can see that the case of smooth contact shows the expected result, with the principal stresses being more

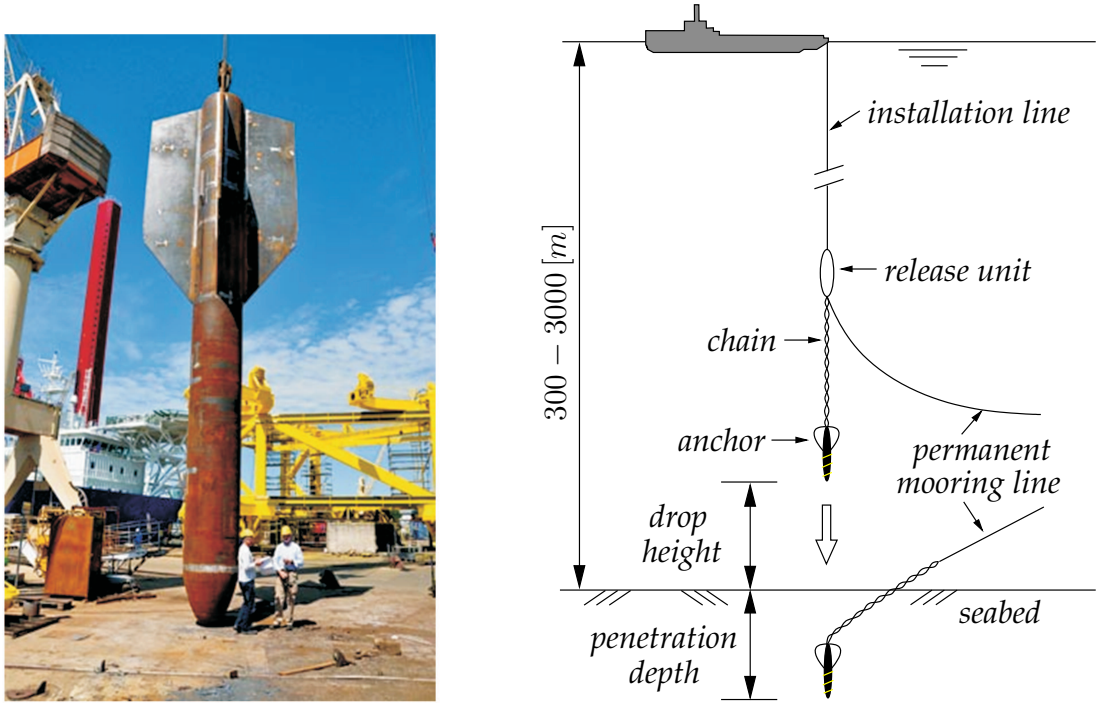


Figure 5.10: A torpedo-shaped dynamic anchor [156]

or less perpendicular to the anchor. Figure 5.15 shows the horizontal stresses along the depth of the soil layer for various offsets from the anchor. These results correspond to the case of smooth contact after a penetration of $18.5D$. As expected, high stresses developed around the tip of the anchor. In sections that are relatively far from the anchor the stresses hardly deviated from those corresponding to the initial stresses calculated by K_0-procedure. Figure 5.16 shows the accumulated shear strains around the anchor for the cases of the smooth and rough contact at their corresponding final configurations. During the penetration of the anchor, considerable shearing of the soil was developed. Figure 5.17 shows the particles distribution after a penetration of $18.5D$ for the case of smooth contact ($\mu = 0$). Soil heave around the anchor can also be seen in this figure.

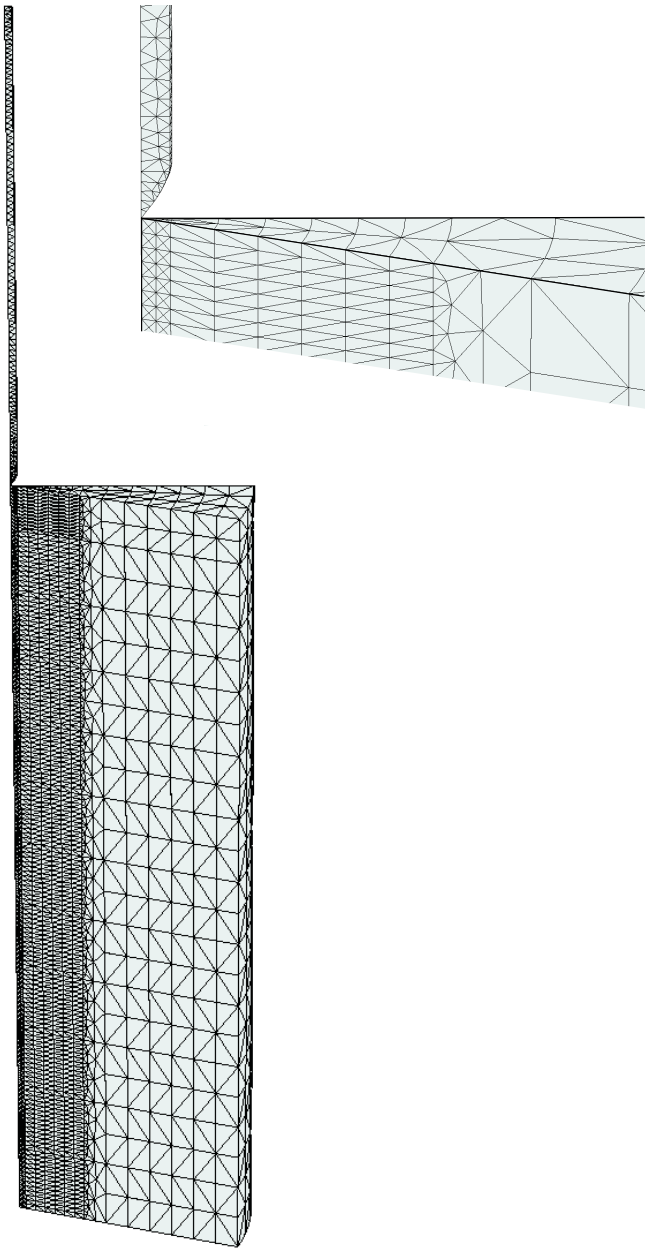


Figure 5.11: The mesh of the dynamic anchor problem

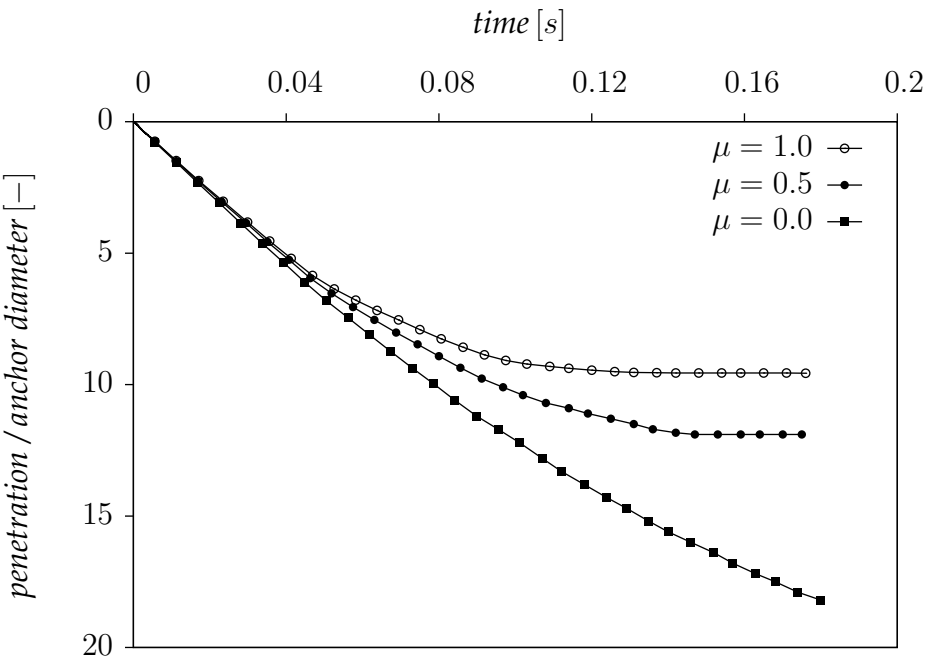


Figure 5.12: Penetration with time for different values of the friction coefficient

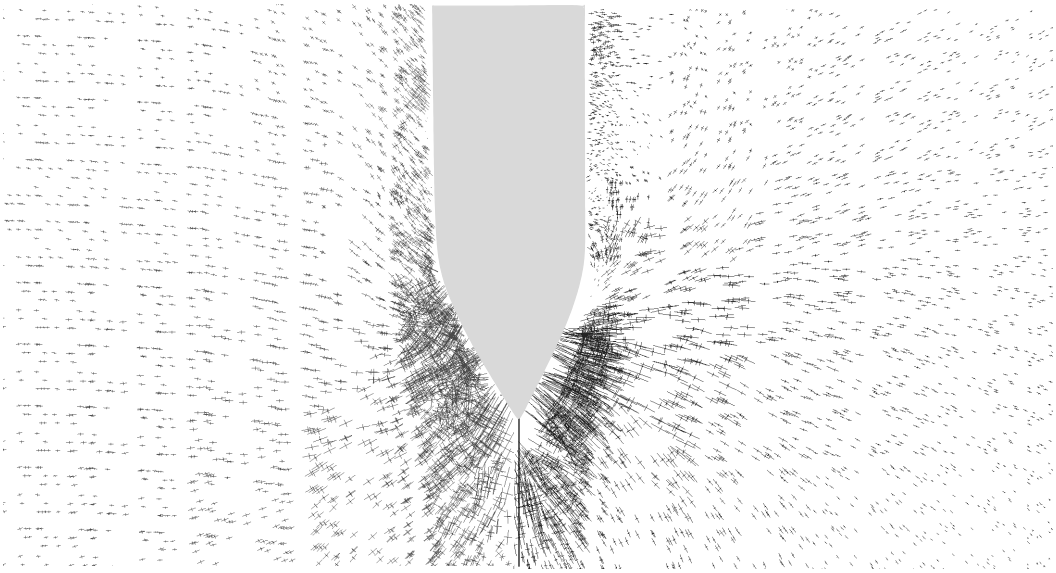


Figure 5.13: Principal stresses directions around the anchor after a penetration of $9.5D$ for the smooth contact (right) and the rough contact (left)

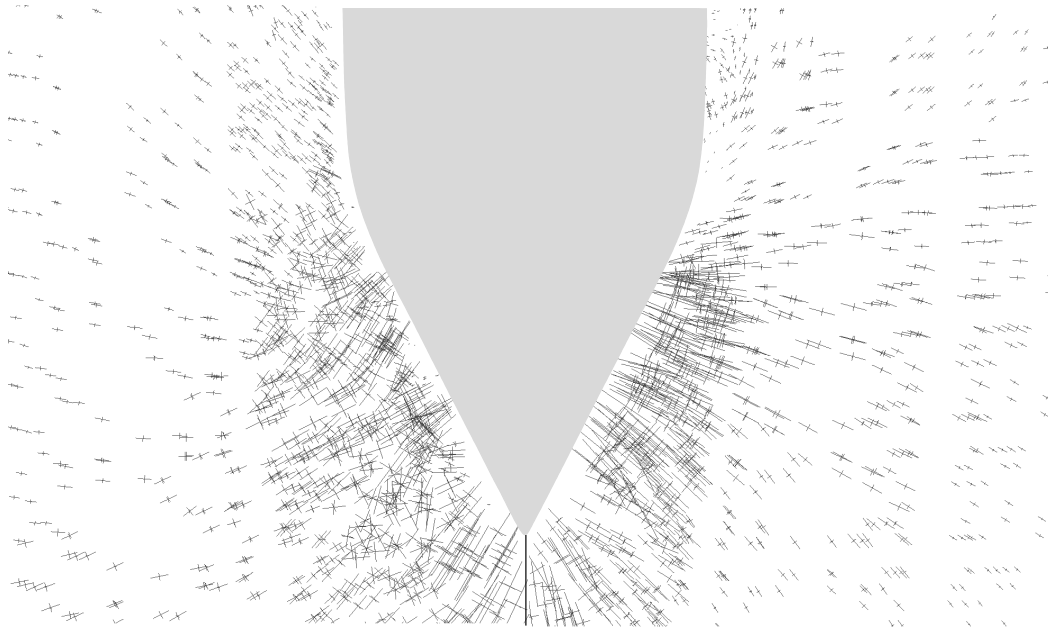


Figure 5.14: A close up of principal stresses directions around the anchor after a penetration of $9.5D$ for the smooth contact (right) and the rough contact (left)

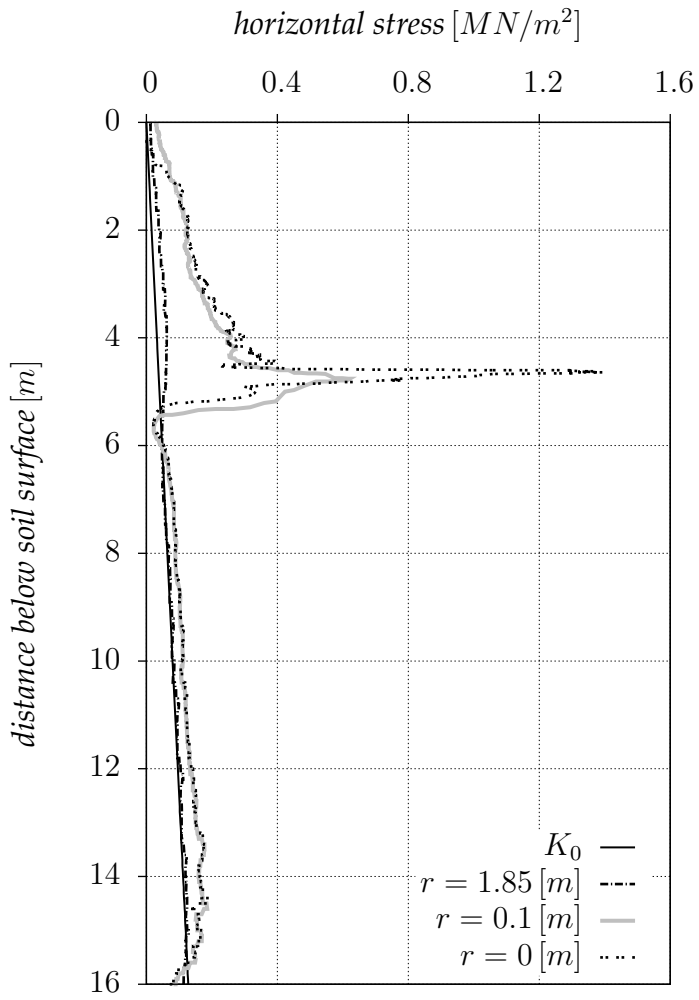
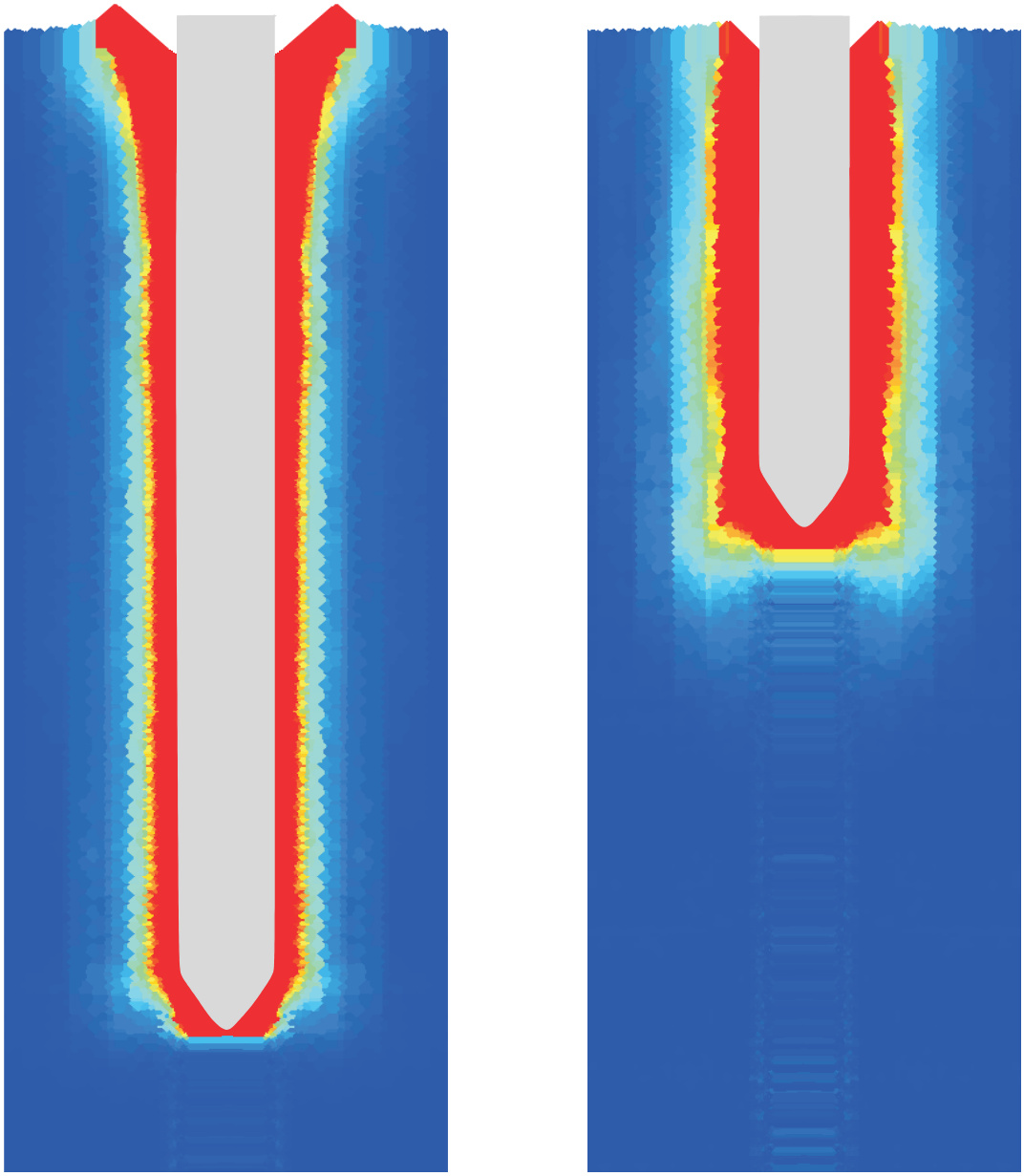
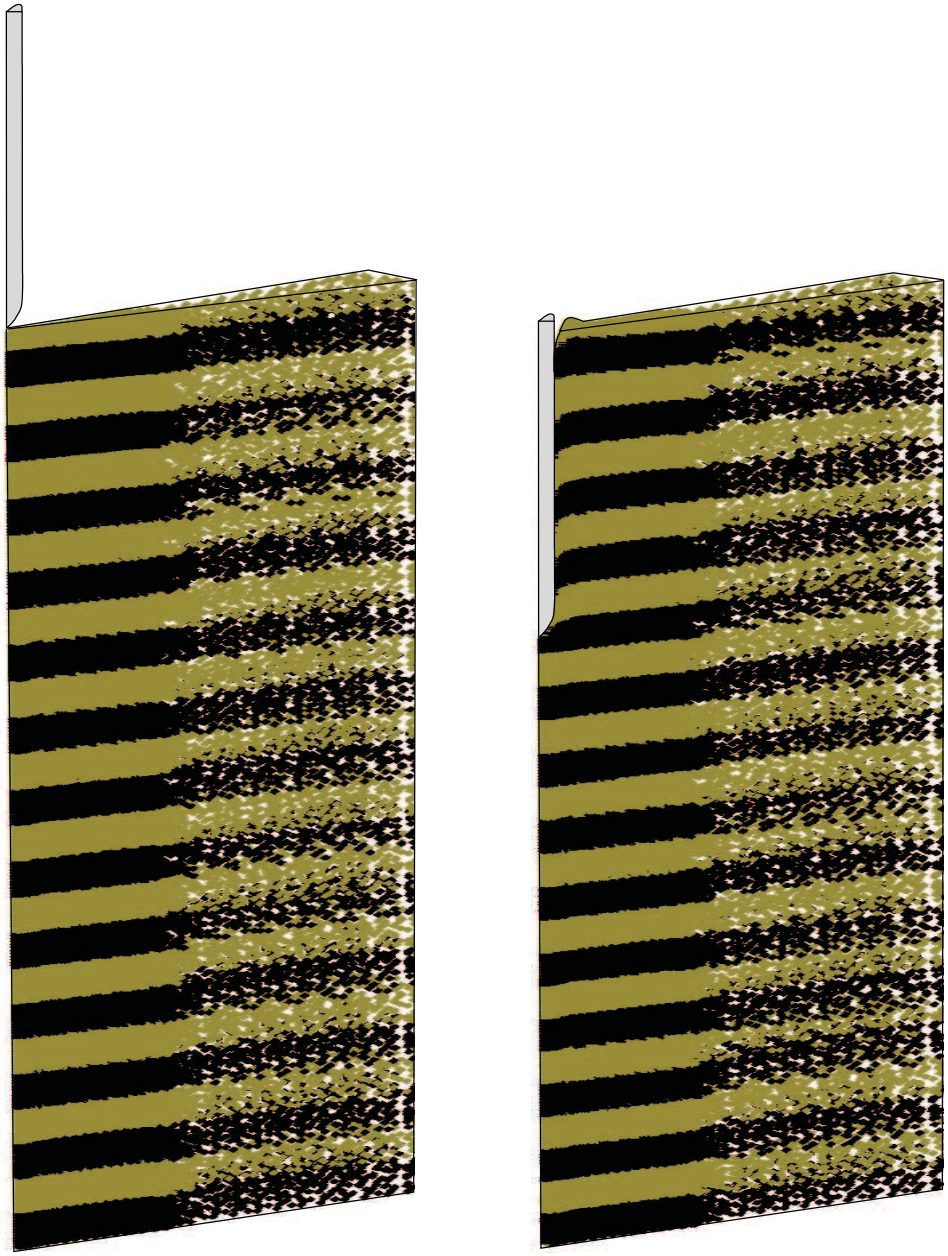


Figure 5.15: Horizontal stresses along the depth of the soil layer for various offsets (r) form the anchor for the case of smooth contact after a penetration of $18.5D$



(a) (b)

Figure 5.16: Accumulated shear strains (a) smooth contact at a penetration of $18.5D$ and (b) rough contact at a penetration of $9.5D$



(a) (b)

Figure 5.17: Particles distribution for the case of smooth contact (a) initial configuration (b) after a penetration of $18.5D$[a]

[a]The domain was stretched radially by a factor of 2 for better visualization.

Chapter 6

Dynamic generation and dissipation of pore pressures

It is not uncommon in geotechnical engineering to encounter problems that include the coupling of solid and fluid phase. A fluid in a porous material, e.g., groundwater in soil, introduces considerable complexities to the mechanical behavior of the material and its numerical simulation. The attention in this chapter focuses on the extension of the material point method to analyze coupled dynamic, two-phase problems by a velocity formulation, in which solid and fluid phase velocities are the primary variables.

In Section 6.1, we present the phase relations of porous media and define the important variables. The concept of effective stress is also presented in this section as well as the possible compression waves in saturated soil. Section 6.2 contains the definitions of the governing equations for two-phase problems, namely the conservation of mass, conservation of momentum for fluid and solid phase as well as the constitutive relation. In Section 6.3, we provide a literature review on the two-phase numerical algorithms. Section 6.4 presents the FEM solution procedure for the $v - w-$formulation. In this section, we also discuss the boundary and initial conditions, weak form as well as the space and time discretization. The algorithm is then extended to MPM in Section 6.5. Section 6.6 presents some important numerical issues with two-phase problems. This includes the modeling of unbounded saturated domains, spurious spatial pressures, damping for two-phase problems as well as mass scaling. Section 6.7 includes validation examples and comparison with problems with existing analytical solutions. An application example of wave attack on sea dike is analyzed in Section 6.8.

6.1 Preliminaries and definitions

In this section, we consider the definition of important quantities that are required in the mathematical model of porous media. The first subsection consists of the phase relations of porous media. The concept of effective stress is considered then in the second subsection. We would like to point out that in the definitions and the mathematical formulation, we consider the soil as a representative example for porous media.

6.1.1 Phase relations of soil

In general, soil is composed of three phases, namely solid, fluid and gas phase. The fluid and gas phase of soil are mostly represented by water and air, respectively. Let us

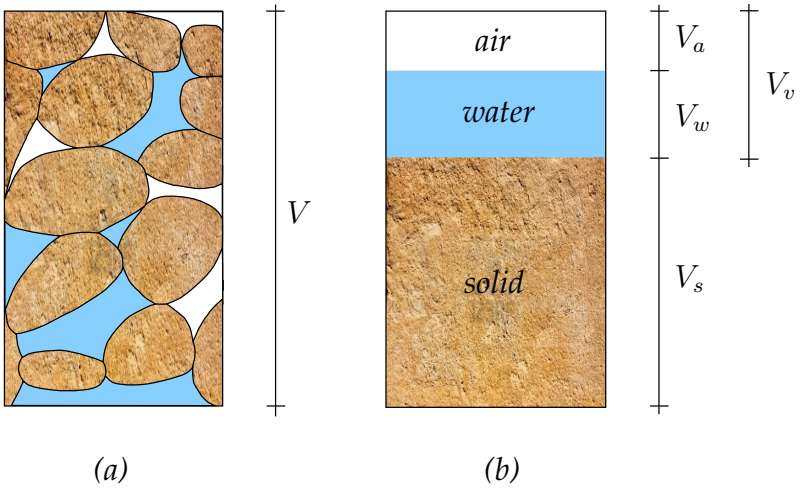


Figure 6.1: Soil element (a) with its representative volume diagram (b)

consider the soil element of Figure 6.1. This element has a volume V, which is composed of void volume V_v and solid volume V_s. The void volume in turn is composed of water volume V_w and air volume V_a. Hence,

$$V = V_v + V_s = V_w + V_a + V_s. \tag{6.1}$$

The mechanical properties of soil are considerably affected by the proportion of the three phases. From the phase relation, the following soil properties can be defined

$$e = \frac{V_v}{V_s} \qquad \text{and} \qquad n = \frac{V_v}{V}, \tag{6.2}$$

where e is the *void ratio* and n is the *porosity*. They are related as

$$e = \frac{n}{1-n}. \tag{6.3}$$

The *degree of saturation* S_r represents the ratio of the water volume to the void volume. It is expressed as

$$S_r = \frac{V_w}{V_v} \times 100\%. \tag{6.4}$$

The soil is said to be dry if $S_r = 0$ and fully saturated if $S_r = 100\%$ otherwise it is partially saturated [183]. This thesis restricts itself to the cases of fully saturated soil ($S_r = 100\%$), and hence the soil is considered to be two-phase material containing solid grains and pore water. When considering saturated soils, different densities can be defined. The *dry density* ϱ_d is defined as

$$\varrho_d = (1-n)\,\varrho_s, \tag{6.5}$$

in which ϱ_s represents the *grain density* of the solid phase, which is the ratio of the mass of solid grains to the volume of solid grains (solid volume). The *saturated density* ϱ_{sat} is expressed as

$$\varrho_{sat} = \varrho_d + n\varrho_w = (1-n)\,\varrho_s + n\varrho_w, \tag{6.6}$$

in which ϱ_w is the *water density*. The *submerged density* ϱ' is related to ϱ_{sat} and ϱ_w via

$$\varrho' = \varrho_{sat} - \varrho_w \tag{6.7}$$

6.1.2 Effective stress concept

The concept of effective stress might be the earliest theory that governs the deformation of soil. This concept accounts for the effect of pore pressures in the deformation of soil. Hence, it plays a very important role in soil mechanics. The concept was first introduced by Terzaghi [171]. Based on this concept, Terzaghi proposed a solution to the one-dimensional consolidation problem in a homogeneous fully saturated soil. The concept was then generalized by Biot [29] to a linear theory of three-dimensional poroelastic material. The theory was further developed by other authors. However, it is not our intention to give a wide overview in this regard. For abroad review of porous media theories, the reader is referred to reference [56] and the references therein.

According to the effective stress concept for saturated soils, the *total stress* σ_{ij} is decomposed into *effective stress* σ'_{ij} and *pore pressure* p as

$$\sigma_{ij} = \sigma'_{ij} + p\delta_{ij}, \tag{6.8}$$

where σ and p are considered to be positive in tension and suction, respectively.

6.1.3 Waves in saturated soil

According to the study of Verruijt [183], there are two characteristic compression waves in saturated soil. In the first wave, water and solid particles move practically with the same velocity. This wave is called the *undrained wave*. It propagates with the following speed

$$c_1 = \sqrt{\frac{E_u^c}{\varrho_{sat}}} \tag{6.9}$$

with E_u^c being the undrained constrained modulus of the saturated soil. It can be expressed as

$$E_u^c = E^c + \frac{K_w}{n}, \tag{6.10}$$

in which K_w represents the bulk modulus of water. In the second wave, water and solid particles move in opposite directions. Unlike the first wave, this wave is strongly damped due to the interaction between water and solid particles. Hence, it can only be observed in the vicinity of the loading source. Such waves can occur in very stiff porous material, e.g., in porous rock. It propagates with the following speed

$$c_2 = \beta_s c_w \tag{6.11}$$

with c_w being the speed of the compression wave in water. It is given as

$$c_w = \sqrt{\frac{K_w}{\varrho_w}}. \tag{6.12}$$

The dimensionless factor β_s, which is written as

$$\beta_s = \sqrt{\frac{n\frac{E^c}{K_w}}{1-n+n\frac{E^c}{K_w}}} \tag{6.13}$$

is smaller than 1 and hence c_2 is smaller than c_w. We refer to the second wave, in this thesis, as the *damped wave*.

6.2 Hydromechanical model for saturated soil

This section is devoted to the presentation of the governing equations that are required for the solution of coupled dynamic, two-phase problems. The equations that describe the two-phase physics are well-known (e.g. [183, 205]); conservation of mass; conservation of momentum; and the constitutive relation. It is not our intention to consider the detailed derivations that lead to those equations, but rather define the terms of the equations and clarify the assumptions of such equations.

6.2.1 Conservation of mass

The conservation of mass of the solid phase is expressed as

$$\frac{d}{dt}\left[(1-n)\,\varrho_s\right] + \frac{\partial}{\partial x_j}\left[(1-n)\,\varrho_s \hat{v}_j\right] = 0, \tag{6.14}$$

in which $\hat{v}_j$ is a component of the velocity vector of the solid phase. On denoting the components of the vector of the (true) velocity of the water phase as $\hat{w}_j$, the conservation of mass of this phase can be written as

$$\frac{d}{dt}\left(n\varrho_w\right) + \frac{\partial}{\partial x_j}\left(n\varrho_w \hat{w}_j\right) = 0. \tag{6.15}$$

When considering incompressible solid grains and disregarding the spatial variations in densities and porosity, one can reduce the expression for the conservation of mass of the solid and water phases to

$$-\frac{dn}{dt} + (1-n)\,\frac{\partial \hat{v}_j}{\partial x_j} = 0 \tag{6.16}$$

and

$$n\frac{d\varrho_w}{dt} + \varrho_w\frac{dn}{dt} + n\varrho_w\frac{\partial \hat{w}_j}{\partial x_j} = 0, \tag{6.17}$$

respectively. Substituting Equation 6.16 into Equation 6.17 eliminates the term $\frac{dn}{dt}$. Hence,

$$n\frac{d\varrho_w}{dt} + \varrho_w\,(1-n)\,\frac{\partial \hat{v}_j}{\partial x_j} + n\varrho_w\frac{\partial \hat{w}_j}{\partial x_j} = 0. \tag{6.18}$$

The water is assumed to be linearly compressible via the relation

$$\frac{d\varrho_w}{dp} = -\frac{\varrho_w}{K_w}. \tag{6.19}$$

Substituting Equation 6.19 into Equation 6.18 and re-arranging terms yields

$$\frac{dp}{dt} = \frac{K_w}{n}\left[(1-n)\frac{\partial \hat{v}_j}{\partial x_j} + n\frac{\partial \hat{w}_j}{\partial x_j}\right]. \tag{6.20}$$

The above equation represents the conservation of mass of the saturated soil. It is also known as the *storage equation*.

6.2.2 Conservation of momentum

The conservation of momentum of the solid phase can be expressed as

$$(1-n)\,\varrho_s\frac{d\hat{v}_j}{dt} - \frac{\partial \sigma'_{ij}}{\partial x_i} - (1-n)\frac{\partial p}{\partial x_j} - (1-n)\,\varrho_s g_j - \frac{n^2\varrho_w g}{k}\left(\hat{w}_j - \hat{v}_j\right) = 0, \tag{6.21}$$

where k is the hydraulic conductivity of soil, also referred to as the Darcy permeability. It can be expressed in terms of the intrinsic permeability κ and the dynamic viscosity of water μ_d as

$$k = \kappa\frac{\varrho_w g}{\mu_d}. \tag{6.22}$$

The conservation of momentum of water phase is written as

$$n\varrho_w\frac{d\hat{w}_j}{dt} - n\frac{\partial p}{\partial x_j} - n\varrho_w g_j + \frac{n^2\varrho_w g}{k}\left(\hat{w}_j - \hat{v}_j\right) = 0. \tag{6.23}$$

The term $n\left(\hat{w}_j - \hat{v}_j\right)$ represents the velocity of water with respect to the solid. It is also known as the specific discharge, which is often denoted as q_j,

$$q_j = n\left(\hat{w}_j - \hat{v}_j\right). \tag{6.24}$$

Adding the momentum of solid phase (Equation 6.21), to the momentum of the water phase (Equation 6.23), yields the momentum of the mixture (saturated soil), which can be expressed as

$$(1-n)\,\varrho_s\frac{d\hat{v}_j}{dt} + n\varrho_w\frac{d\hat{w}_j}{dt} = \frac{\partial \sigma_{ij}}{\partial x_i} + \varrho_{sat}\,g_j. \tag{6.25}$$

6.2.3 Constitutive relation

When considering saturated soils, the constitutive relation relates the effective stress of the soil skeleton to the deformations. The constitutive relation is already considered in Section 3.1. The difference here is that it is written in terms of effective stresses as

$$\dot{\sigma}'_{ij} = \overset{\nabla \mathrm{H}'}{\sigma}_{ij} - \dot{\varepsilon}_{kk}\sigma'_{ij} + \omega_{ik}\sigma'_{kj} - \sigma'_{ik}\omega_{kj} \tag{6.26}$$

with

$$\overset{\nabla \mathrm{H}'}{\sigma}_{ij} = D_{ijkl}\dot{\varepsilon}_{kl}. \tag{6.27}$$

We presume that the reader is already familiar with nomenclature of the above equations from Section 3.1. Full description of the boundary and initial conditions is given in Section 6.4.

6.3 Literature review on numerical modeling of two-phase problems

The literature on the finite element implementation of two-phase problems is quite extensive (e.g. [209, 211]). Therefore, only references that are most relevant to the development of numerical procedures of two-phase problems are included here, as the references themselves refer to large amount of literature on the topic.

Early studies showed a preference for implicit or semi-implicit schemes together with high-order elements, see reference [208]. In more recent years emphasis has been placed on adopting low-order elements that incorporate stabilizing techniques to circumvent the Babuška-Brezzi restriction imposed on elements with equal order interpolation for displacement and pore pressure. References [94, 100, 104, 189, 210] provided an excellent insight into the numerical analysis developed in this area.

Although the $v-p-$formulation, i.e., solid velocity-water pressure formulation, can capture dynamic response for various scenarios, it was shown by van Esch et al. [178] that such a formulation cannot accurately capture two-phase dynamic behavior that involves, for instance, the propagation of a compression wave (undrained wave) followed by a second wave that is associated with the consolidation process (damped wave). The full set of equations including all acceleration terms such as the $v-w-$formulation, i.e., solid velocity-water velocity formulation, is required to capture both waves. An extensive discussion on the applicability of various two-phase formulations can be found in earlier publications by Zienkiewicz and coworkers (e.g. [205, 207]).

Many possibilities exist with regard to the selection of interpolation functions for velocity and pressure (e.g. [97]). Hughes [97] presented various possibilities outlining the strengths and weaknesses within the context of incompressible elasticity. Van Esch et al. [178] presented a detailed comparison on the finite element algorithms of $v-p-$formulation and $v-w-$formulation. They showed that both formulations can capture the undrained wave, but the $v-p-$formulation is not able to capture the damped wave. The capability of $v-w-$formulation to precisely capture the physical response of saturated soil under dynamic loading is attributed to the fact that in this formulation all acceleration terms are considered. From a physical view point, pressure should be interpolated one order lower than that of velocity for consistency between variations in pressure and stress. The $v-w-$formulation automatically ensures the consistency between pressure and stress.

For the $v-w-$formulation, the size of the critical time step is limited by

$$\Delta t_{crit} = min\left(\frac{h_{min}}{c_1}, \frac{h_{min}}{c_2}\right). \tag{6.28}$$

In addition to the above criterion, another criterion must be satisfied when selecting the size of the critical time step for $v-p-$formulation. For one-dimensional case, this criterion is expressed, following Verruijt [183], as

$$\Delta t_{crit}^{vp} = \frac{h_{min}^2}{2c_v} \tag{6.29}$$

with c_v being the consolidation coefficient, which is defined as

$$c_v = \frac{k}{\varrho_w g\left(\frac{1}{E^c} + \frac{n}{K_w}\right)}. \tag{6.30}$$

The $v-w-$formulation provides a physical based mapping of water momentum (extensive thermodynamic property) between particles and grid points when considering MPM. As in this formulation, water velocity is a primary variable calculated at the nodes. The mapping is done in the same way as for the solid phase, see Chapter 4. When adopting $v-p-$formulation, in which pressure is calculated at the nodes, one should map the pressure (intensive thermodynamic property) between particles and grid points. For these reasons, the $v-w-$formulation is favored in this study.

In the $v-w-$formulation, one would have four unknown variables, these are the velocity of solid phase $\hat{v}_i$, the velocity of water phase $\hat{w}_i$, water pressure p and the effective stress σ'_{ij}. The velocities are calculated at the nodes as primary variables, whereas the pressure and stress are calculated at Gauss points (considering FEM) or particles (considering MPM). Following Verruijt [183], the sequence of the solution for the four unknowns starts with calculating the velocity of the water phase from the momentum of water. Hence,

$$\varrho_w \frac{d\hat{w}_j}{dt} = \frac{\partial p}{\partial x_j} + \varrho_w g_j - \frac{n\varrho_w g}{k}\left(\hat{w}_j - \hat{v}_j\right). \tag{6.31}$$

The velocity of the solid phase follows from the momentum of the mixture as

$$\left(1-n\right)\varrho_s \frac{d\hat{v}_j}{dt} = -n\varrho_w \frac{d\hat{w}_j}{dt} + \frac{\partial \sigma_{ij}}{\partial x_i} + \varrho_{sat}\, g_j. \tag{6.32}$$

The storage equation is then used to calculate the pressure rate as

$$\frac{dp}{dt} = \frac{K_w}{n}\left[\left(1-n\right)\frac{\partial \hat{v}_j}{\partial x_j} + n\frac{\partial \hat{w}_j}{\partial x_j}\right]. \tag{6.33}$$

Eventually the constitutive relation is used to calculate the rate of the effective stress as

$$\overset{\nabla \mathrm{H}'}{\sigma}_{ij} = D_{ijkl}\dot{\varepsilon}_{kl}. \tag{6.34}$$

It is important to realize that the order of the solution of equations is important to develop good predictions.

Challenges associated with the use of low-order elements in two-phase problems that involves incompressible water are mesh locking and spurious pressures due to incompressibility. The enhanced volumetric strain procedure of Chapter 4 was found to be useful for accommodating incompressible plastic strains. When dealing with incompressible fluids, the fractional step method could mitigate the pathological locking and spurious spatial pressure oscillations [133].

It is of interest to briefly refer here to the fractional step method. For single-phase solids, the momentum equation is partitioned into two equations, one involves the deviatoric stress and the other involves the isotropic stress, with the sum of them yielding the expression for the total momentum. The basic idea is to accommodate the influence of pressure gradient in the pressure equation and hence relaxing the incompressibility constraint and spurious pore pressure variations are suppressed. Within the context of the considered two-phase $v-w-$formulation, such a partitioning occurs automatically according to the mixture and water momentum equations, i.e., $\frac{d\hat{w}_i}{dt}$ is separate from $\frac{d\hat{v}_i}{dt}$. When considering the enhanced volumetric strain procedure for the solid and water, it plays an effective role in mitigating the mesh locking and spurious pressures due to incompressibility.

One more remark is that the considered two-phase formulation has an additional stabilizing factor that is related to the interaction between solid and water. These forces are represented by the last term of the right-hand side of Equation 6.31.

6.4 Finite element model for v-w-formulation

In this section, we consider the numerical algorithm of the $v-w-$formulation in the framework of Lagrangian FEM. Boundary and initial conditions of two-phase problems are discussed first followed by the weak and space discretization of the momentum equations. Last part of this section focused on the time discretization of the equations.

6.4.1 Boundary and initial conditions

The $v-w-$formulation requires that the boundary $\partial\Omega$ of the domain Ω is the union of the following components

$$\partial\Omega = \partial\Omega_u \cup \partial\Omega_\tau = \partial\Omega_w \cup \partial\Omega_p, \tag{6.35}$$

where $\partial\Omega_w$ and $\partial\Omega_p$ are the prescribed velocity and prescribed pressure boundaries of the water phase, respectively, whereas $\partial\Omega_u$ is the prescribed displacement (velocity) boundary of the solid phase and $\partial\Omega_\tau$ is the prescribed total stress boundary. The following conditions should also be satisfied at the boundary

$$\partial\Omega_u \cap \partial\Omega_\tau = \varnothing \quad \text{and} \quad \partial\Omega_w \cap \partial\Omega_p \varnothing. \tag{6.36}$$

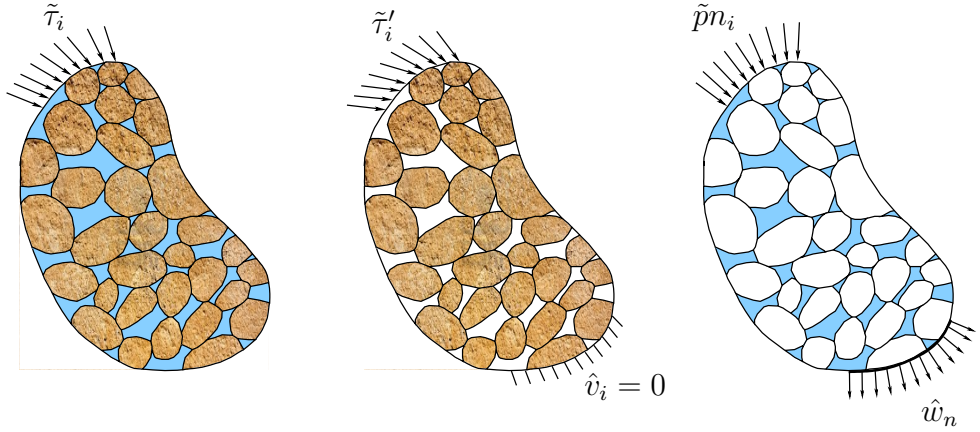


Figure 6.2: Boundary conditions for two-phase problems[a]

[a]The effective traction $\tilde{\tau}'_i$ is included for completeness and $\hat{w}_n$ represents the normal component of water velocity.

The displacement (velocity) boundary conditions for the solid and water phases can be written as

$$\hat{v}_i(\boldsymbol{x}, t) = \hat{V}_i(t) \qquad \text{on} \quad \partial\Omega_u(t) \tag{6.37}$$

and

$$\hat{w}_i(\boldsymbol{x}, t) = \hat{W}_i(t) \qquad \text{on} \quad \partial\Omega_w(t), \tag{6.38}$$

respectively. On defining the prescribed total traction as $\tau_i(\boldsymbol{x}, t) = \tilde{\tau}_i(\boldsymbol{x})\mathcal{T}(t)$ and the prescribed pressure as $\bar{p}_i(\boldsymbol{x}, t) = \tilde{p}(\boldsymbol{x}) n_i \mathcal{T}(t)$, the total traction and pressure boundary conditions can be written as

$$\sigma_{ij}(\boldsymbol{x}, t) n_j = \tau_i(\boldsymbol{x}, t) \qquad \text{on} \quad \partial\Omega_\tau(t) \tag{6.39}$$

and

$$p(\boldsymbol{x}, t) n_i = \bar{p}_i(\boldsymbol{x}, t) \qquad \text{on} \quad \partial\Omega_p(t), \tag{6.40}$$

respectively. One should distinguish between the porosity n and the vector n_i, which represents the unit vector normal to the boundary $\partial\Omega$. Figure 6.2 shows the boundary conditions for a two-phase problem. The initial conditions for solid and water velocities are considered as

$$\hat{v}_i(\boldsymbol{x}, t_0) = \hat{V}_{0i} \qquad \text{and} \qquad \hat{w}_i(\boldsymbol{x}, t_0) = \hat{W}_{0i}, \tag{6.41}$$

respectively. Initial total stress and pressure are

$$\sigma_{ij}(\boldsymbol{x}, t_0) = \sigma_{0ij} \qquad \text{and} \qquad p(\boldsymbol{x}, t_0) = p_0, \tag{6.42}$$

respectively.

6.4.2 Weak formulations

Section 3.3 considered the detailed procedure of the weak form and the space discretization of the momentum equation of a single-phase material. The same procedure is applied to the two-phase formulation here. For the fine points, the reader is referred to Section 3.3. The strong forms of the momentum equations have to be transfered to the weak forms before the discretization. This is achieved by multiplying Equations 6.31 and 6.32 by weighting functions $\hat{t}_j$ and integrating them over the current domain Ω occupied by the continuum. Hence,

$$\int_\Omega \hat{t}_j \varrho_w \frac{d\hat{w}_j}{dt}\, d\Omega = \int_\Omega \hat{t}_j \frac{\partial p}{\partial x_j}\, d\Omega + \int_\Omega \hat{t}_j \varrho_w g_j\, d\Omega - \int_\Omega \hat{t}_j \frac{n\varrho_w g}{k}\left(\hat{w}_j - \hat{v}_j\right)\, d\Omega \tag{6.43}$$

and

$$\int_\Omega \hat{t}_j \left(1-n\right) \varrho_s \frac{d\hat{v}_j}{dt}\, d\Omega = -\int_\Omega \hat{t}_j n \varrho_w \frac{d\hat{w}_j}{dt}\, d\Omega + \int_\Omega \hat{t}_j \frac{\partial \sigma_{ij}}{\partial x_i}\, d\Omega + \int_\Omega \hat{t}_j \varrho_{sat}\, g_j\, d\Omega. \tag{6.44}$$

Applying the divergence theorem and subsequently the traction boundary conditions yield the final respective weak forms

$$\begin{aligned}\int_\Omega \hat{t}_j \varrho_w \frac{d\hat{w}_j}{dt}\, d\Omega = &\int_{\partial\Omega_p} \hat{t}_j \bar{p}_j\, dS - \int_\Omega \frac{\partial \hat{t}_j}{\partial x_j} p\, d\Omega \\ &+ \int_\Omega \hat{t}_j \varrho_w g_j\, d\Omega - \int_\Omega \hat{t}_j \frac{n\varrho_w g}{k}\left(\hat{w}_j - \hat{v}_j\right)\, d\Omega\end{aligned} \tag{6.45}$$

and

$$\begin{aligned}\int_\Omega \hat{t}_j \left(1-n\right) \varrho_s \frac{d\hat{v}_j}{dt}\, d\Omega = &-\int_\Omega \hat{t}_j n \varrho_w \frac{d\hat{w}_j}{dt}\, d\Omega + \int_{\partial\Omega_\tau} \hat{t}_j \tau_j\, dS \\ &- \int_\Omega \frac{\partial \hat{t}_j}{\partial x_i} \sigma_{ij}\, d\Omega + \int_\Omega \hat{t}_j \varrho_{sat}\, g_j\, d\Omega.\end{aligned} \tag{6.46}$$

6.4.3 Space discretization

To proceed with the space discretization of the momentum equations, we follow the same procedure of Section 3.3, in which matrix notation is used. The same shape functions are used to approximate the velocities of water and solid phases as well as the weighting function. Such approximation can be written as

$$\hat{\boldsymbol{v}}\left(\boldsymbol{x},t\right) \approx \bar{\boldsymbol{N}}\left(\boldsymbol{x}\right)\boldsymbol{v}\left(t\right), \qquad \hat{\boldsymbol{w}}\left(\boldsymbol{x},t\right) \approx \bar{\boldsymbol{N}}\left(\boldsymbol{x}\right)\boldsymbol{w}\left(t\right) \quad \text{and} \quad \hat{\boldsymbol{t}}\left(\boldsymbol{x},t\right) \approx \bar{\boldsymbol{N}}\left(\boldsymbol{x}\right)\boldsymbol{t}\left(t\right). \tag{6.47}$$

We presume here that the reader is already familiar with nomenclature and dimensions of arrays from Chapter 3 and hence there is no need to repeat the definitions again unless there are new quantities. Substituting the above approximation into Equations 6.45 and 6.46 yields

$$\begin{aligned}\int_\Omega \bar{\boldsymbol{N}}^T \varrho_w \bar{\boldsymbol{N}} \boldsymbol{a}_w\, d\Omega = &\int_{\partial\Omega_p} \bar{\boldsymbol{N}}^T \bar{\boldsymbol{p}}\, dS - \int_\Omega \boldsymbol{B}^T \boldsymbol{\delta} p\, d\Omega \\ &+ \int_\Omega \bar{\boldsymbol{N}}^T \varrho_w \boldsymbol{g}\, d\Omega - \int_\Omega \bar{\boldsymbol{N}}^T \frac{n\varrho_w g}{k} \bar{\boldsymbol{N}}\left(\boldsymbol{w}-\boldsymbol{v}\right)\, d\Omega\end{aligned} \tag{6.48}$$

and

$$\int_\Omega \bar{\boldsymbol{N}}^T (1-n)\, \varrho_s\, \bar{\boldsymbol{N}} \boldsymbol{a}_s \, d\Omega = -\int_\Omega \bar{\boldsymbol{N}}^T n \varrho_w\, \bar{\boldsymbol{N}} \boldsymbol{a}_w \, d\Omega + \int_{\partial\Omega_\tau} \bar{\boldsymbol{N}}^T \boldsymbol{\tau} \, dS - \int_\Omega \boldsymbol{B}^T \boldsymbol{\sigma} \, d\Omega + \int_\Omega \bar{\boldsymbol{N}}^T \varrho_{sat}\, \boldsymbol{g} \, d\Omega. \tag{6.49}$$

The vector $\boldsymbol{a}_w$ contains the nodal accelerations of the water phase, $\bar{\boldsymbol{p}}$ is a vector of prescribed water pressure, which is defined in Section 6.4.1 and $\boldsymbol{\delta}$ is expressed as

$$\boldsymbol{\delta} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T. \tag{6.50}$$

Let us now write the equations in terms of the local shape functions $\boldsymbol{N}$, which are written in terms of the parent element coordinates, to get

$$\left(\mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T \varrho_w\, \boldsymbol{N} \, d\Omega \right) \boldsymbol{a}_w = \mathop{\mathrm{A}}_{e=1}^{n_{pelm}} \int_{\partial\Omega_{pe}} \boldsymbol{N}^T \bar{\boldsymbol{p}}_e \, dS + \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T \varrho_w\, \boldsymbol{g} \, d\Omega - \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{\delta} p \, d\Omega - \left(\mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T \frac{n \varrho_w g}{k} \boldsymbol{N} \, d\Omega \right) (\boldsymbol{w} - \boldsymbol{v}) \tag{6.51}$$

and

$$\left(\mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T (1-n)\, \varrho_s\, \boldsymbol{N} \, d\Omega \right) \boldsymbol{a}_s = -\left(\mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T n \varrho_w\, \boldsymbol{N} \, d\Omega \right) \boldsymbol{a}_w + \mathop{\mathrm{A}}_{e=1}^{n_{\tau elm}} \int_{\partial\Omega_{\tau e}} \boldsymbol{N}^T \boldsymbol{\tau}_e \, dS + \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T \varrho_{sat}\, \boldsymbol{g} \, d\Omega - \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{\sigma} \, d\Omega, \tag{6.52}$$

in which n_{pelm} represents the number of the surface elements where water pressure is prescribed. We presume at this point that the reader is familiar with the numerical integration of the above equations from Chapter 3 and Appendix C. The discrete system of equations can now be written as

$$\boldsymbol{M}_w \boldsymbol{a}_w = \boldsymbol{F}_w^{trac} + \boldsymbol{F}_w^{grav} - \boldsymbol{F}_w^{int} - \boldsymbol{F}_w^{drag} \tag{6.53}$$

and

$$\boldsymbol{M}_s \boldsymbol{a}_s = -\bar{\boldsymbol{M}}_w \boldsymbol{a}_w + \boldsymbol{F}^{trac} + \boldsymbol{F}^{grav} - \boldsymbol{F}^{int}. \tag{6.54}$$

The subscripts s and w indicate solid and water, respectively. No subscript indicates that the quantity belongs to the mixture. All matrices are lumped in the above equations and the direct lumping procedure of Chapter 3 is used again to lump them. The drag force $\boldsymbol{F}_w^{drag}$ is defined as

$$\boldsymbol{F}_w^{drag} = \boldsymbol{Q} (\boldsymbol{w} - \boldsymbol{v}) \tag{6.55}$$

with $\boldsymbol{Q}$ being a lumped matrix expressed as

$$\boldsymbol{Q} = \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \boldsymbol{Q}_e \tag{6.56}$$

with

$$\boldsymbol{Q}_e = \begin{bmatrix} \boldsymbol{q}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{q}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{q}_{en} \end{bmatrix}. \tag{6.57}$$

Each sub-matrix $\boldsymbol{q}_i$ corresponds to node i, and the matrix $\mathbf{0}$ is a null matrix, they are expressed as

$$\boldsymbol{q}_i = \begin{bmatrix} q_i & 0 & 0 \\ 0 & q_i & 0 \\ 0 & 0 & q_i \end{bmatrix} \quad \text{and} \quad \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.58}$$

with

$$q_i \approx \sum_{q=1}^{n_{eq}} w_q \, N_i \left(\boldsymbol{\xi}_q\right) \frac{n \varrho_w}{k} \left(\boldsymbol{x} \left(\boldsymbol{\xi}_q\right)\right) g \left|\boldsymbol{J} \left(\boldsymbol{\xi}_q\right)\right|. \tag{6.59}$$

6.4.4 Time integration and solution procedure

As indicated previously, we adopt Euler-Cromer scheme, in which the velocity is updated explicitly and the displacement increment is calculated implicitly. This also applies to the $v-w-$formulation. Let us re-write Equation 6.53 as

$$\boldsymbol{M}_w \frac{d\boldsymbol{w}}{dt} = \boldsymbol{F}_w^{trac} + \boldsymbol{F}_w^{grav} - \boldsymbol{F}_w^{int} - \boldsymbol{Q} \left(\boldsymbol{w} - \boldsymbol{v}\right), \tag{6.60}$$

in which $\frac{d\boldsymbol{w}}{dt}$ replaces $\boldsymbol{a}_w$. Integrating this equation in the time interval $[t, t+\Delta t]$ yields

$$\int_t^{t+\Delta t} \boldsymbol{M}_w d\boldsymbol{w} = \int_t^{t+\Delta t} \left[\boldsymbol{F}_w^{trac} + \boldsymbol{F}_w^{grav} - \boldsymbol{F}_w^{int} - \boldsymbol{Q} \left(\boldsymbol{w} - \boldsymbol{v}\right)\right] dt. \tag{6.61}$$

The explicit numerical integration of the above gives

$$\boldsymbol{M}_w \left(\boldsymbol{w}^{t+\Delta t} - \boldsymbol{w}^t\right) \approx \left[\boldsymbol{F}_w^{trac,t} + \boldsymbol{F}_w^{grav,t} - \boldsymbol{F}_w^{int,t} - \boldsymbol{Q}^t \left(\boldsymbol{w}^t - \boldsymbol{v}^t\right)\right] \Delta t \tag{6.62}$$

and hence the update of water velocity at time $t+\Delta t$ can be written as

$$\boldsymbol{w}^{t+\Delta t} \approx \boldsymbol{w}^t + \boldsymbol{M}_w^{-1} \left[\boldsymbol{F}_w^{trac,t} + \boldsymbol{F}_w^{grav,t} - \boldsymbol{F}_w^{int,t} - \boldsymbol{Q}^t \left(\boldsymbol{w}^t - \boldsymbol{v}^t\right)\right] \Delta t. \tag{6.63}$$

The same integration procedure is then applied to Equation 6.54 to get

$$\boldsymbol{v}^{t+\Delta t} \approx \boldsymbol{v}^t + \boldsymbol{M}_s^{-1} \left[-\bar{\boldsymbol{M}}_w^{-1} \boldsymbol{a}_w^t + \boldsymbol{F}^{trac,t} + \boldsymbol{F}^{grav,t} - \boldsymbol{F}^{int,t}\right] \Delta t, \tag{6.64}$$

in which

$$\boldsymbol{a}_w^t = \frac{\boldsymbol{w}^{t+\Delta t} - \boldsymbol{w}^t}{\Delta t}. \tag{6.65}$$

After calculating the velocity $\boldsymbol{v}^{t+\Delta t}$ of the solid phase, the strain increment of the solid phase at a Gauss point q of element e is calculated implicitly as

$$\Delta\boldsymbol{\varepsilon}_q^{t+\Delta t} = \boldsymbol{B}_q \boldsymbol{v}_e^{t+\Delta t} \Delta t, \tag{6.66}$$

in which the vector $\boldsymbol{v}_e$ contains velocities of element e. This vector is extracted from the global vector $\boldsymbol{v}$. Water pressure is updated by integrating the storage equation (Equation 6.33) implicitly. Hence,

$$p_q^{t+\Delta t} \approx p_q^t + \Delta t \frac{K_w}{n} \boldsymbol{\delta}^T \left[(1-n)\, \boldsymbol{B}_q \boldsymbol{v}_e^{t+\Delta t} + n \boldsymbol{B}_q \boldsymbol{w}_e^{t+\Delta t} \right]. \tag{6.67}$$

The effective stress is then updated using the constitutive relation, i.e.,

$$\{\boldsymbol{\sigma}', \text{material state}\}^t \xrightarrow[\Delta\boldsymbol{\varepsilon}^{t+\Delta t}]{\text{constitutive relation}} \{\boldsymbol{\sigma}', \text{material state}\}^{t+\Delta t}. \tag{6.68}$$

The total stress at the considered Gauss point is then calculated as

$$\boldsymbol{\sigma}_q^{t+\Delta t} = \boldsymbol{\sigma}_q'^{\,t+\Delta t} + \boldsymbol{\delta} p_q^{t+\Delta t}. \tag{6.69}$$

6.5 Two-phase dynamic material point method

In previous sections of this chapter, the differential equations of the hydromechanical model were presented. Some literature was reviewed to give an overview on different discretization possibilities with their pros and cons. The finite element discretization of the $v-w-$formulation was detailed as well in this chapter. This FEM discretization is extended to MPM in this section. We would like to remind the reader that in this section, we will repeat some initialization procedure and concepts from Chapter 4.

6.5.1 MPM discretization and particles initialization

The basic assumption here is that the domain of the saturated soil is represented by a single set of particles. Hence, there is no distinction between solid and water particles. The domain is discretized following the same procedure of Section 4.2. Particles carry all properties of the solid and water phase and their positions are updated according to the displacement of the solid phase. We will consider in this section the initialization of particles including the association of mass, body forces, tractions and other properties of the saturated soil to particles. Knowing the local and global positions of particle p, the volume Ω_p associated with it is calculated using Equation 4.2. One should realize that this volume represents the total volume, which is composed of water and solid phase volumes. Densities, porosity, hydraulic conductivity and bulk modulus of water are then assigned to particle p. The following masses are then calculated and assigned to the particle

$$m_{s,p} = \varrho_{s,p}\, \Omega_p, \qquad m_{w,p} = \varrho_{w,p}\, \Omega_p \qquad \text{and} \qquad m_p = \varrho_{sat,p}\, \Omega_p. \tag{6.70}$$

These masses are used to integrate the mass matrices that are required in the momentum equations. The corresponding body forces are calculated from the masses associated with each particle as follows

$$\boldsymbol{f}_{s,p}^{grav} = m_{s,p}\,\boldsymbol{g}, \qquad \boldsymbol{f}_{w,p}^{grav} = m_{w,p}\,\boldsymbol{g} \qquad \text{and} \qquad \boldsymbol{f}_{p}^{grav} = m_{p}\,\boldsymbol{g}. \tag{6.71}$$

The total traction force vector $\tilde{\boldsymbol{f}}_p^{trac}$ of a boundary particle p is calculated following the procedure of Section 4.3. This procedure is also applied in the calculation of the forces corresponding to the prescribed pressure. For a boundary particle p these forces are calculated as

$$\tilde{\boldsymbol{f}}_{w,p}^{trac} = \tilde{\boldsymbol{p}}_e \frac{S_e}{n_{ebp}}, \tag{6.72}$$

in which S_e is the surface area of the triangular element where the pressure is prescribed. One should recall from Section 6.4 that

$$\tilde{\boldsymbol{p}} = \tilde{p}\boldsymbol{n} = \tilde{p}n_i, \tag{6.73}$$

where $\tilde{p}$ is the prescribed pressure on the boundary and $\boldsymbol{n} = n_i$ is the unit vector normal to that boundary.

6.5.2 Solution procedure of a single time step

This section focuses on the initialization of the momentum equations by mapping the state of the continuum from particles to nodes as these equations are eventually solved on the nodes. A full description of the overall solution procedure of two-phase problems in MPM is also considered in this section.

6.5.2.1 Initialization of momentum equations

As done in Chapter 4, we consider the state of the saturated soil at time t. First of all, the mass matrices $\boldsymbol{M}_w$, $\bar{\boldsymbol{M}}_w$ and $\boldsymbol{M}_s$ have to be integrated at the beginning of the time step. The reader is referred to Equations 4.11 to 4.14 for the detailed procedure of integration and construction of such matrices. We however concentrate here on the contribution of particles to the nodes of the element where these particles are located at time t. This means that Equation 4.14 is required to be written as

$$m_{w,i}^t \approx \sum_{p=1}^{n_{ep}} m_{w,p}\, N_i\left(\boldsymbol{\xi}_p^t\right), \tag{6.74}$$

$$\bar{m}_{w,i}^t \approx \sum_{p=1}^{n_{ep}} n_p^t\, m_{w,p}\, N_i\left(\boldsymbol{\xi}_p^t\right) \tag{6.75}$$

and

$$m_{s,i}^t \approx \sum_{p=1}^{n_{ep}} \left(1 - n_p^t\right) m_{s,p}\, N_i\left(\boldsymbol{\xi}_p^t\right) \tag{6.76}$$

when integrating the matrices $\boldsymbol{M}_w$, $\bar{\boldsymbol{M}}_w$ and $\boldsymbol{M}_s$, respectively. The next step in the initialization involves mapping the velocities from particles to nodes. This applies for the solid and water phase velocities. For the solid phase this is done via solving the following equation

$$\boldsymbol{M}_s^t \boldsymbol{v}^t = \boldsymbol{P}_s^t, \tag{6.77}$$

where the vector $\boldsymbol{P}_s^t$ contains the mapped momentum from particles to nodes. It is assembled as

$$\boldsymbol{P}_s^t \approx \bigA_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \left(1 - n_p^t\right) m_{s,p} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{v}}_p^t. \tag{6.78}$$

The same procedure is repeated to map the velocity of the water phase from particles to nodes. Hence, the following equation is solved

$$\bar{\boldsymbol{M}}_w^t \boldsymbol{w}^t = \boldsymbol{P}_w^t. \tag{6.79}$$

The water momentum vector is assembled as follows

$$\boldsymbol{P}_w^t \approx \bigA_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} n_p^t m_{w,p} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{w}}_p^t. \tag{6.80}$$

Nodal forces vector that corresponds to the prescribed total traction is assembled as follows

$$\tilde{\boldsymbol{F}}^{trac,t} \approx \bigA_{e=1}^{\bar{n}_{\tau elm}} \sum_{p=1}^{n_{ebp}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \tilde{\boldsymbol{f}}_p^{trac} \tag{6.81}$$

and the one that corresponds to the prescribed pressure is assembled as

$$\tilde{\boldsymbol{F}}_w^{trac,t} \approx \bigA_{e=1}^{\bar{n}_{pelm}} \sum_{p=1}^{n_{ebp}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \tilde{\boldsymbol{f}}_{w,p}^{trac}. \tag{6.82}$$

Then

$$\boldsymbol{F}^{trac,t} = \tilde{\boldsymbol{F}}^{trac,t} \mathcal{T}\left(t\right) \quad \text{and} \quad \boldsymbol{F}_w^{trac,t} = \tilde{\boldsymbol{F}}_w^{trac,t} \mathcal{T}\left(t\right). \tag{6.83}$$

The body force vectors are integrated as follows

$$\boldsymbol{F}^{grav,t} \approx \bigA_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{f}_p^{grav} \tag{6.84}$$

and

$$\boldsymbol{F}_w^{grav,t} \approx \bigA_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{f}_{w,p}^{grav}. \tag{6.85}$$

The internal forces due to total stresses and water pressure are

$$\boldsymbol{F}^{int,t} \approx \bigA_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{B}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{\sigma}_p^t \Omega_p^t. \tag{6.86}$$

and

$$\boldsymbol{F}_w^{int,t} \approx \bigwedge_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \boldsymbol{B}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{\delta} p_p^t \, \Omega_p^t, \tag{6.87}$$

respectively, with the vector of the drag force calculated as

$$\boldsymbol{F}_w^{drag,t} = \boldsymbol{Q}^t \left(\boldsymbol{w}^t - \boldsymbol{v}^t\right), \tag{6.88}$$

where the lumped matrix $\boldsymbol{Q}$ is constructed following the procedure described by Equations 6.56 to 6.59, but with the integration of q_i carried out as follows

$$q_i^t \approx \sum_{p=1}^{n_{ep}} \frac{n_p^t \, m_{w,p} \, g}{k_p^t} \, N_i \left(\boldsymbol{\xi}_p^t\right). \tag{6.89}$$

The discrete momentum equations are then written as

$$\boldsymbol{M}_w^t \boldsymbol{a}_w^t = \boldsymbol{F}_w^{trac,t} + \boldsymbol{F}_w^{grav,t} - \boldsymbol{F}_w^{int,t} - \boldsymbol{F}_w^{drag,t} \tag{6.90}$$

and

$$\boldsymbol{M}_s^t \boldsymbol{a}_s^t = -\bar{\boldsymbol{M}}_w^t \boldsymbol{a}_w^t + \boldsymbol{F}^{trac,t} + \boldsymbol{F}^{grav,t} - \boldsymbol{F}^{int,t}. \tag{6.91}$$

The above two equations are ready to be solved by the solution procedure that follows.

6.5.2.2 Solving the momentum equations using the modified Lagrangian algorithm

The momentum equations of two-phase problems are solved using the algorithm of Section 4.4, i.e., the modified Lagrangian algorithm. The solution sequence is described in the following steps

1. Equation 6.90 is solved for $\boldsymbol{a}_w^t$ as

$$\boldsymbol{a}_w^t = \boldsymbol{M}_w^{t,-1} \left[\boldsymbol{F}_w^{trac,t} + \boldsymbol{F}_w^{grav,t} - \boldsymbol{F}_w^{int,t} - \boldsymbol{F}_w^{drag,t}\right].$$

2. Once the vector $\boldsymbol{a}_w^t$ is calculated, the acceleration vector $\boldsymbol{a}_s^t$ is then calculated from Equation 6.91 as

$$\boldsymbol{a}_s^t = \boldsymbol{M}_s^{t,-1} \left[-\bar{\boldsymbol{M}}_w^t \boldsymbol{a}_w^t + \boldsymbol{F}^{trac,t} + \boldsymbol{F}^{grav,t} - \boldsymbol{F}^{int,t}\right].$$

3. The velocities of particles are updated using the nodal accelerations and the shape functions as

$$\hat{\boldsymbol{v}}_p^{t+\Delta t} = \hat{\boldsymbol{v}}_p^t + \sum_{i=1}^{n_{en}} \Delta t \, N_i \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{a}_{s,i}^t$$

and

$$\hat{\boldsymbol{w}}_p^{t+\Delta t} = \hat{\boldsymbol{w}}_p^t + \sum_{i=1}^{n_{en}} \Delta t \, N_i \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{a}_{w,i}^t.$$

4. Nodal velocities $\boldsymbol{v}^{t+\Delta t}$ and $\boldsymbol{w}^{t+\Delta t}$ are then calculated from the updated particles velocities solving

$$\boldsymbol{M}_s^t \boldsymbol{v}^{t+\Delta t} = \bigwedge_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} \left(1 - n_p^t\right) m_{s,p} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{v}}_p^{t+\Delta t}$$

and

$$\bar{\boldsymbol{M}}_w^t \boldsymbol{w}^{t+\Delta t} = \bigwedge_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep}} n_p^t \, m_{w,p} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{w}}_p^{t+\Delta t}.$$

5. Nodal incremental displacements of the solid phase are calculated as

$$\Delta \boldsymbol{u}^{t+\Delta t} = \Delta t \, \boldsymbol{v}^{t+\Delta t}.$$

6. Strains and effective stresses are determined using

$$\Delta \boldsymbol{\varepsilon}_p^{t+\Delta t} = \boldsymbol{B}\left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_e^{t+\Delta t}$$

and

$$\{\boldsymbol{\sigma}'_p, \text{material state}\}^t \xrightarrow[\Delta \boldsymbol{\varepsilon}_p^{t+\Delta t}]{\text{constitutive relation}} \{\boldsymbol{\sigma}'_p, \text{material state}\}^{t+\Delta t}.$$

The vector $\Delta \boldsymbol{u}_e$ is defined in Equation 3.108.

7. Water pressure at particle p is then updated via

$$p_p^{t+\Delta t} \approx p_p^t + \Delta t \frac{K_{w,p}}{n_p^t} \boldsymbol{\delta}^T \left[\left(1 - n_p^t\right) \boldsymbol{B}\left(\boldsymbol{\xi}_p^t\right) \boldsymbol{v}_e^{t+\Delta t} + n_p^t \boldsymbol{B}\left(\boldsymbol{\xi}_p^t\right) \boldsymbol{w}_e^{t+\Delta t}\right].$$

8. The total stress at each particle is subsequently calculated as

$$\boldsymbol{\sigma}_p^{t+\Delta t} = \boldsymbol{\sigma}'^{\,t+\Delta t}_p + \boldsymbol{\delta} p_p^{t+\Delta t}.$$

9. Volumes associated with particles are updated using the volumetric strain increment $\Delta \varepsilon_{vol}$ as

$$\Omega_p^{t+\Delta t} = \left(1 + \Delta \varepsilon_{vol,p}^{t+\Delta t}\right) \Omega_p^t \qquad \text{with} \qquad \Delta \varepsilon_{vol} = \Delta \varepsilon_{11} + \Delta \varepsilon_{22} + \Delta \varepsilon_{33}.$$

10. Positions of particles are updated using the displacement of the solid phase as

$$\boldsymbol{x}_p^{t+\Delta t} = \boldsymbol{x}_p^t + \sum_{i=1}^{n_{en}} N_i\left(\boldsymbol{\xi}_p^t\right) \Delta \boldsymbol{u}_i^{t+\Delta t}.$$

11. The book-keeping is updated using the new positions of particles.

6.6 Other numerical issues

This section addresses some issues that are of considerable importance in the numerical schemes. We consider viscous boundaries for two-phase problems, mitigation of spurious spatial pressures, damping for two-phase problems and mass scaling.

6.6.1 Unbounded saturated domain

The viscous boundaries of Section 4.7 are extended to saturated soil. We must define two sets of Kelvin-Voigt elements, one for water phase and another for the solid phase. The response of the Kelvin-Voigt element for the water phase is given as

$$\boldsymbol{p}^{vb} = -\boldsymbol{\eta}_w \hat{\boldsymbol{w}} - \boldsymbol{\kappa}_w \hat{\boldsymbol{u}}_w, \tag{6.92}$$

in which $\boldsymbol{p}^{vb}$ represents the traction corresponding to the Kelvin-Voigt element of the water phase at a material point located at the boundary. The matrices $\boldsymbol{\eta}_w$ and $\boldsymbol{\kappa}_w$ are written as

$$\boldsymbol{\eta}_w = \begin{bmatrix} a\varrho_w c_w & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\kappa}_w = \begin{bmatrix} \frac{\varrho_w c_w^2}{\delta} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{6.93}$$

The vectors $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{u}}_w$ are defined as

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \hat{w}_n & \hat{w}_{t_1} & \hat{w}_{t_2} \end{bmatrix}^T \quad \text{and} \quad \hat{\boldsymbol{u}}_w = \begin{bmatrix} \hat{u}_{wn} & \hat{u}_{wt_1} & \hat{u}_{wt_2} \end{bmatrix}^T, \tag{6.94}$$

where the first element in each vector represents the normal component and the other two are the tangential ones. If we account for the boundary term as an additional traction term in the momentum equation of the water phase (Equation 6.48), this equation can now be written as

$$\begin{aligned} \int_\Omega \bar{\boldsymbol{N}}^T \varrho_w \, \bar{\boldsymbol{N}} \boldsymbol{a}_w \, d\Omega = & \int_{\partial\Omega_p} \bar{\boldsymbol{N}}^T \bar{\boldsymbol{p}} \, dS - \int_{\partial\Omega_{pvb}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta}_w \bar{\boldsymbol{N}} \boldsymbol{w} \, dS - \int_{\partial\Omega_{pvb}} \bar{\boldsymbol{N}}^T \boldsymbol{\kappa}_w \bar{\boldsymbol{N}} \boldsymbol{u}_w \, dS \\ & - \int_\Omega \boldsymbol{B}^T \boldsymbol{\delta} p \, d\Omega + \int_\Omega \bar{\boldsymbol{N}}^T \varrho_w \, \boldsymbol{g} \, d\Omega - \int_\Omega \bar{\boldsymbol{N}}^T \frac{n \varrho_w g}{k} \bar{\boldsymbol{N}} \left(\boldsymbol{w} - \boldsymbol{v} \right) \, d\Omega. \end{aligned} \tag{6.95}$$

Now we shall consider the traction terms of the water and the solid phase in the momentum equation of the mixture. One can re-call from Section 4.7 that the response of Kelvin-Voigt element of the solid phase can be written as

$$\boldsymbol{\tau}^{vb} = -\boldsymbol{\eta}_s \hat{\boldsymbol{v}} - \boldsymbol{\kappa}_s \hat{\boldsymbol{u}}, \tag{6.96}$$

where all terms are defined in Section 4.7. Equation 6.49 can be written as

$$\begin{aligned}\int_{\Omega} \bar{\boldsymbol{N}}^T (1-n)\, \varrho_s\, \bar{\boldsymbol{N}} \boldsymbol{a}_s \, d\Omega = &- \int_{\Omega} \bar{\boldsymbol{N}}^T n \varrho_w\, \bar{\boldsymbol{N}} \boldsymbol{a}_w \, d\Omega + \int_{\partial\Omega_\tau} \bar{\boldsymbol{N}}^T \boldsymbol{\tau} \, dS \\ &- \int_{\partial\Omega_{_{pvb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta}_w \bar{\boldsymbol{N}} \boldsymbol{w} \, dS - \int_{\partial\Omega_{_{pvb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\kappa}_w \bar{\boldsymbol{N}} \boldsymbol{u}_w \, dS \\ &- \int_{\partial\Omega_{_{\tau vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta}_s \bar{\boldsymbol{N}} \boldsymbol{v} \, dS - \int_{\partial\Omega_{_{\tau vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\kappa}_s \bar{\boldsymbol{N}} \boldsymbol{u} \, dS \\ &- \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma} \, d\Omega + \int_{\Omega} \bar{\boldsymbol{N}}^T \varrho_{sat}\, \boldsymbol{g} \, d\Omega. \end{aligned} \tag{6.97}$$

The resulting momentum equations become

$$\boldsymbol{M}_w \boldsymbol{a}_w = \boldsymbol{F}_w^{trac} - \boldsymbol{F}_w^{vb} + \boldsymbol{F}_w^{grav} - \boldsymbol{F}_w^{int} - \boldsymbol{F}_w^{drag} \tag{6.98}$$

and

$$\boldsymbol{M}_s \boldsymbol{a}_s = -\bar{\boldsymbol{M}}_w \boldsymbol{a}_w + \boldsymbol{F}^{trac} - \boldsymbol{F}_w^{vb} - \boldsymbol{F}_s^{vb} + \boldsymbol{F}^{grav} - \boldsymbol{F}^{int}, \tag{6.99}$$

where

$$\boldsymbol{F}_w^{vb} = \boldsymbol{C}_w \boldsymbol{w} + \boldsymbol{K}_w \boldsymbol{u}_w \tag{6.100}$$

and

$$\boldsymbol{F}_s^{vb} = \boldsymbol{C}_s \boldsymbol{v} + \boldsymbol{K}_s \boldsymbol{u}. \tag{6.101}$$

The matrices for the dashpot and spring coefficients are defined as follows

$$\boldsymbol{C}_w = \int_{\partial\Omega_{_{pvb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta}_w \bar{\boldsymbol{N}} \, dS, \tag{6.102}$$

$$\boldsymbol{K}_w = \int_{\partial\Omega_{_{pvb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\kappa}_w \bar{\boldsymbol{N}} \, dS, \tag{6.103}$$

$$\boldsymbol{C}_s = \int_{\partial\Omega_{_{\tau vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\eta}_s \bar{\boldsymbol{N}} \, dS, \tag{6.104}$$

$$\boldsymbol{K}_s = \int_{\partial\Omega_{_{\tau vb}}} \bar{\boldsymbol{N}}^T \boldsymbol{\kappa}_s \bar{\boldsymbol{N}} \, dS. \tag{6.105}$$

The integration of these matrices and the lumping procedure is well explained in Section 4.7. We should re-call from Section 4.7 that the forces induced by the viscous boundaries are calculated in an incremental form and accumulated with time. Considering the water phase as an example, one can write the following

$$\boldsymbol{F}_w^{vb,t} = \boldsymbol{F}_w^{vb,t-\Delta t} + \Delta \boldsymbol{F}_w^{vb,t}, \tag{6.106}$$

where

$$\Delta \boldsymbol{F}_w^{vb,t} = \boldsymbol{C}_w^t \Delta \boldsymbol{w}^t + \boldsymbol{K}_w^t \Delta \boldsymbol{u}_w^t. \tag{6.107}$$

6.6.2 Spurious spatial pressures

The enhanced volumetric strain procedure of Section 4.8 is applied here to the solid and the water phase to mitigate the spurious spatial pressures. The water pressure rate at any point q is calculated from

$$\dot{p}\left(\boldsymbol{\xi}_q\right) = \frac{K_w}{n}\boldsymbol{\delta}^T\left[(1-n)\,\boldsymbol{B}\left(\boldsymbol{\xi}_q\right)\boldsymbol{v}_e + n\boldsymbol{B}\left(\boldsymbol{\xi}_q\right)\boldsymbol{w}_e\right], \tag{6.108}$$

which can be put as

$$\begin{aligned}\dot{p}\left(\boldsymbol{\xi}_q\right) &= \frac{K_w}{n}\left[(1-n)\,\boldsymbol{\delta}^T\boldsymbol{B}\left(\boldsymbol{\xi}_q\right)\boldsymbol{v}_e + n\boldsymbol{\delta}^T\boldsymbol{B}\left(\boldsymbol{\xi}_q\right)\boldsymbol{w}_e\right] \\ &= \frac{K_w}{n}\left[(1-n)\,\dot{\varepsilon}_{vols}\left(\boldsymbol{\xi}_q\right) + n\dot{\varepsilon}_{volw}\left(\boldsymbol{\xi}_q\right)\right],\end{aligned} \tag{6.109}$$

where $\dot{\varepsilon}_{vols}$ and $\dot{\varepsilon}_{volw}$ are the volumetric strain rates for the solid and water phase, respectively. Applying the enhanced strain procedure, Equation 6.109 takes the following form

$$\dot{p}\left(\boldsymbol{\xi}_q\right) = \frac{K_w}{n}\left[(1-n)\,\bar{\dot{\varepsilon}}_{vols}\left(\boldsymbol{\xi}_q\right) + n\bar{\dot{\varepsilon}}_{volw}\left(\boldsymbol{\xi}_q\right)\right], \tag{6.110}$$

where $\bar{\dot{\varepsilon}}_{vols}$ and $\bar{\dot{\varepsilon}}_{volw}$ are the enhanced volumetric strain rates for the solid and water phase, respectively. The calculation of the enhanced volumetric strain for the water phase is summarized by the following steps

1. After calculating the volumetric strain rate $\dot{\varepsilon}_{volwe}$ for each element, the nodal values of the smoothed volumetric strain rate are calculated as follows

   $$\bar{\dot{\varepsilon}}_{volwi} = \frac{\sum \dot{\varepsilon}_{volwe}\Omega_e}{\sum \Omega_e}, \tag{6.111}$$

   where the sum runs over all elements attached to node i.

2. Once the smoothed volumetric strain rate associated with each node is calculated, the average volumetric strain rate for the element is determined by averaging the nodal values of the considered element. Considering element e as an example, one can write

   $$\bar{\dot{\varepsilon}}_{volwe} = \frac{1}{n_{en}}\sum_{i=1}^{n_{en}}\bar{\dot{\varepsilon}}_{volwi}, \tag{6.112}$$

   where $\bar{\dot{\varepsilon}}_{volwe}$ is the enhanced volumetric strain of element e.

3. The enhanced volumetric strain of the element is then assigned to all points (particles) inside this element. Hence,

   $$\bar{\dot{\varepsilon}}_{volw}\left(\boldsymbol{\xi}_q\right) = \bar{\dot{\varepsilon}}_{volwe}, \tag{6.113}$$

   which is used then to calculate the water pressure rate at the considered point from Equation 6.110.

The new strain tensor which results from the enhancement of the volumetric strain of the solid phase is then used to calculate the effective stress tensor from the constitutive relation.

6.6.3 Local damping for two-phase problems

The same concept of local damping that is explained in Section 4.10 for a single-phase material is extended here to a two-phase material. Let us write the momentum equations of a two-phase material for any degree-of-freedom as

$$m_w a_w = f_w^{trac} + f_w^{grav} - f_w^{int} - f_w^{drag} \tag{6.114}$$

and

$$m_s a_s = -\bar{m}_w a_w + f^{trac} + f^{grav} - f^{int} \tag{6.115}$$

or for simplicity

$$m_w a_w = f_w - f_w^{drag} \tag{6.116}$$

and

$$m_s a_s = -\bar{m}_w a_w + f, \tag{6.117}$$

where

$$f_w = f_w^{trac} + f_w^{grav} - f_w^{int} \qquad \text{and} \qquad f = f^{trac} + f^{grav} - f^{int}. \tag{6.118}$$

The local damping is applied to the momentum of water as

$$m_w a_w = f_w + f_w^{damp} - f_w^{drag}, \tag{6.119}$$

with

$$f_w^{damp} = -\alpha_w \left| f_w \right| sign\left(w\right), \tag{6.120}$$

in which α_w is a dimensionless damping factor for the water phase and w is the water phase velocity at the considered degree-of-freedom. One should realize that the term f_w^{drag} is excluded when calculating the damping force. For two-phase interaction, this term represents the drag between the solid and water phase and hence it already provides a natural damping. On applying local damping to the momentum of mixture, Equation 6.117 can now be written as

$$m_s a_s = -\bar{m}_w a_w + f + f_w^{damp} + f_s^{damp}, \tag{6.121}$$

where

$$f_s^{damp} = -\alpha_s \left| f - f_w \right| sign\left(v\right), \tag{6.122}$$

which corresponds to the damping of the solid phase. The parameter α_s is a dimensionless damping factor for the solid phase and v is the solid phase velocity at the considered degree-of-freedom.

6.6.4 Mass scaling for quasi-static problems

The mass scaling is adopted in this thesis as a means to increase the time step size in problems involving slow rate of loading. The procedure consists of modification of Equations 6.90 and 6.91 to the forms

$$\beta_m \boldsymbol{M}_w^t \boldsymbol{a}_w^t = \boldsymbol{F}_w^{trac,t} + \boldsymbol{F}_w^{grav,t} - \boldsymbol{F}_w^{int,t} - \boldsymbol{F}_w^{drag,t} \tag{6.123}$$

and

$$\beta_m \boldsymbol{M}_s^t \boldsymbol{a}_s^t = -\beta_m \bar{\boldsymbol{M}}_w^t \boldsymbol{a}_w^t + \boldsymbol{F}^{trac,t} + \boldsymbol{F}^{grav,t} - \boldsymbol{F}^{int,t}, \tag{6.124}$$

respectively, with β_m being the dimensionless scale factor. It must be realized that we only scale the mass matrices, but not the densities and gravity loading. The new time step Δt_2 is related to the one without scaling Δt_1 as

$$\Delta t_2 = \sqrt{\beta_m}\, \Delta t_1. \tag{6.125}$$

This procedure can be used in case of dry soil as well. The assessment of the procedure is given in the numerical example of Section 6.7.2.

6.7 Validation examples

In this section, we analyze some boundary-value problems with existing analytical solutions to validate the implementation of the two-phase MPM. In the first example, we analyze the propagation of the wave resulting from instantaneous loading of a column of saturated porous material. The second example consists of the well-known one-dimensional consolidation problem of Terzaghi.

6.7.1 One-dimensional wave propagation

The extreme case of the propagation of a pressure wave in a $2.5\,[m]$ height column of saturated porous material, discretized with 1000 rows of elements along the height, each row containing 6 tetrahedral elements, is considered in this example. A fine grid was required to properly capture the propagating front for the idealized loading. Zero displacement and zero flux conditions were specified at the bottom, with roller boundaries prescribed along the sides so that only displacement along the column was possible. A compressive total stress and a pressure of $1\,[kN/m^2]$ were instantaneously applied and maintained at the top surface of the column. Such loading conditions can be experimentally achieved by increasing the ground water level from its initial position at the surface by $0.1\,[m]$.

The elastic porous material had a constrained modulus of $E^c = 5000\,[MN/m^2]$, a dry density of $\varrho_d = 1600\,[kg/m^3]$, and a porosity of $n = 0.4$. The bulk modulus of water was assumed to be $K_w = 2000\,[MN/m^2]$. The stiffness of the water phase reads $K_w/n = 5000\,[MN/m^2]$, which equals the constrained modulus of the porous material. As a consequence, the uniform traction of $1\,[kN/m^2]$ will initially be equally carried by the water and the solid phase.

Low permeability Computations were carried out with a hydraulic conductivity of $k = 10^{-5}\,[m/s]$ to observe the undrained wave. Given the selected properties, only the undrained compression wave could be noticed. From Equation 6.9, the speed of this wave was found to be $2236\,[m/s]$. Figure 6.3 shows the normalized increase of the initial pore pressure with time for a particle originally located at $0.675\,[m]$ from the top down.

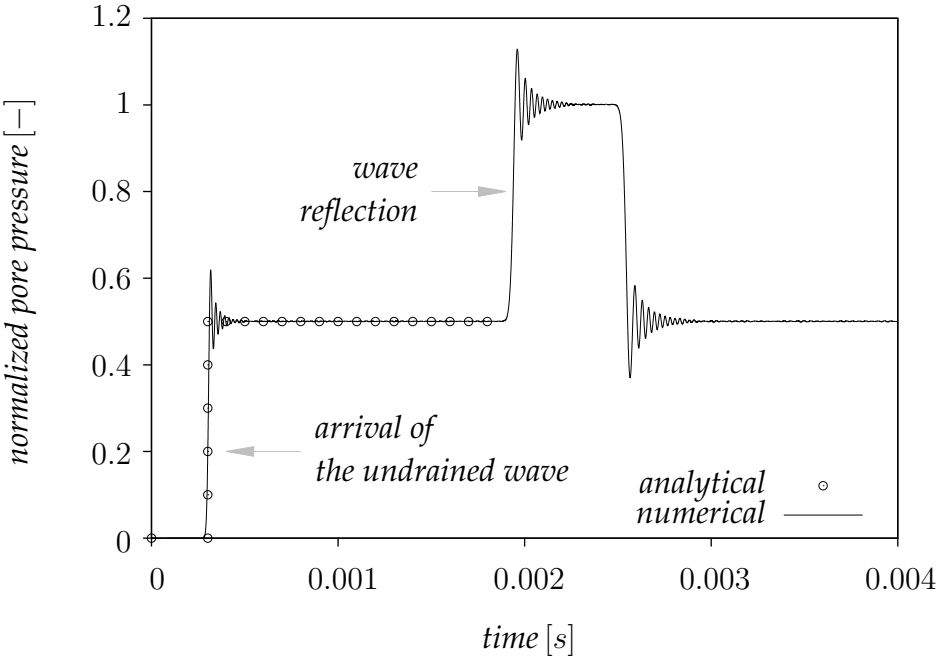


Figure 6.3: Normalized pore pressure history of a particle initially located at $0.675\,[m]$ depth for the case of rigid boundary and low conductivity

The increase of the initial pore pressure was normalized with respect to the total stress applied at the top surface ($1\,[kN/m^2]$). Also shown, is the theoretical solution given in reference [183]. The predicted time of arrival of the undrained wave ($3 \times 10^{-4}\,[s]$) agrees well with that calculated using Equation 6.9. The theoretical solution for the increase pore of pressure and effective stress from the initial values are given as

$$\Delta p_0 = \frac{\frac{1}{E^c}}{\frac{1}{E^c} + \frac{n}{K_w}} \sigma_s \tag{6.126}$$

and

$$\Delta \sigma'_0 = \frac{\frac{n}{K_w}}{\frac{1}{E^c} + \frac{n}{K_w}} \sigma_s, \tag{6.127}$$

respectively, with Δp_0 and $\Delta \sigma'_0$ being the increase of the initial pore pressure and the effective stress, respectively, and σ_s is the total stress applied at the top surface. With the selected properties, $\Delta p_0 = \Delta \sigma'_0 = 0.5\sigma_s$. Figure 6.3 shows excellent agreement between the analytical and the numerical predictions. Although not shown, the effective stress history is identical to Figure 6.3. The high frequency oscillations that appear are typical for numerical solutions, in which loads are applied instantaneously. They can be mitigated by applying the load gradually over a couple of time steps. Further refinement of the mesh also helps mitigating them. Scrutiny of the figure shows a doubling of the pressure due to the interaction of reflected waves at approximately $0.002\,[s]$, which is consistent with what is expected.

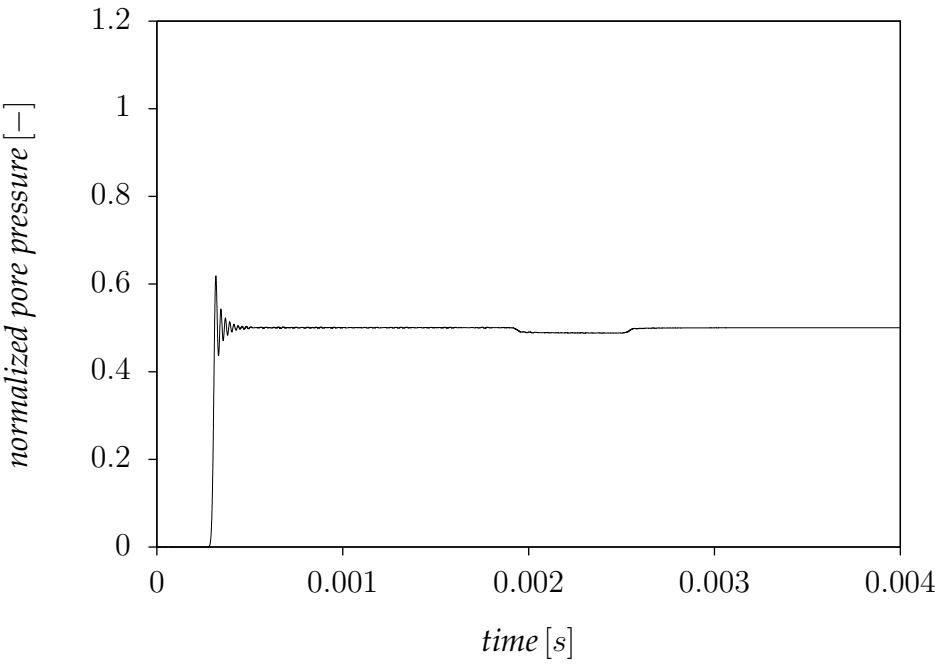


Figure 6.4: Normalized pore pressure history of a particle initially located at $0.675\,[m]$ depth for the case of viscous boundary and low conductivity

Application of viscous boundaries To assess the activity of the viscous boundaries, the problem was also solved applying Kelvin-Voigt elements with $\delta = 1.25\,[m]$. The results are shown in Figure 6.4. The simulation confirms that the viscous boundaries absorb the propagating waves. Nevertheless, the small reflection at time near $0.002\,[s]$ is attributed to the stiffness term in the Kelvin-Voigt element.

High permeability Additional computations were carried out with a hydraulic conductivity of $k = 10^{-3}\,[m/s]$ to observe the damped wave. With the adopted properties, the speed of the damped wave was found to be $1118\,[m/s]$, see Equation 6.11, whereas the speed of the undrained wave was $2236\,[m/s]$. Figure 6.5 shows that the predicted arrival time of the damped wave ($6 \times 10^{-4}\,[s]$) agrees with Equation 6.11. Although not shown, this wave was completely damped at a depth of approximately ($2.2\,[m]$). The increase of the pore pressure will asymptotically reach the prescribed value of $1\,[kN/m^2]$.

6.7.2 One-dimensional consolidation

Terzaghi's well-known problem of one-dimensional consolidation is considered for an elastic soil layer of $1\,[m]$ thickness. In contrast to the example of Section 6.7.1, the surface load in this problem was initially fully carried by the water. Hence, the initial pore water pressure was equal $1\,[kN/m^2]$ throughout the soil layer. This example demonstrates the capability of the dynamic material point method to capture the quasi-static equilibrium

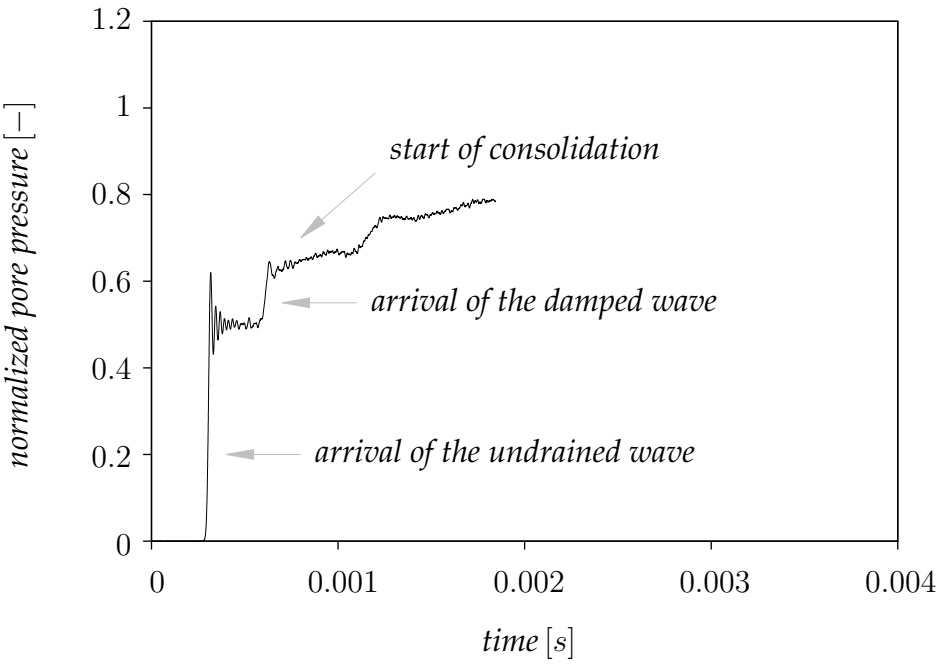


Figure 6.5: Normalized pore pressure history of a particle initially located at $0.675\,[m]$ depth for the case of rigid boundary and high conductivity

as the water is squeezed out of the pores of a saturated soil. A compression surface traction of $10\,[kN/m^2]$ was applied and maintained at the drainage upper surface. The hydraulic conductivity of the soil was $k = 10^{-3}\,[m/s]$, with a constrained modulus of $E^c = 10\,[MN/m^2]$, a dry density of $\varrho_d = 1600\,[kg/m^3]$ and a porosity of $n = 0.4$. The bulk modulus of water was assumed to be $K_w = 300\,[MN/m^2]$.

The layer was discretized using 40 rows, each row consisting of 6 tetrahedral elements with one particle per element. To capture the "closed-form" solution by Terzaghi, the undrained condition with a uniform pore pressure of $p_0 = 10\,[kN/m^2]$ was taken as an initial condition. Possible wave reflections were suppressed by introducing the Kelvin-Voigt boundary condition with $\delta = 0.5\,[m]$. Figure 6.6 compares the MPM and "closed-form" solutions of normalized pore pressure versus normalized depth. The pore pressure was normalized with respect to the initial value and the depth with respect to the soil layer thickness. The isochrones of Figure 6.6 are plotted for various time factor T, which is defined as

$$T = \frac{c_v t}{h^2} \tag{6.128}$$

with c_v being the consolidation coefficient as defined in Equation 6.30 and h is the thickness of the soil layer. We observe an excellent agreement between the two solutions. For small time-factors, this was only possible by preventing reflections at the lower boundary.

Application of mass scaling To assess the procedure of mass scaling, additional computations were carried out using the same time step ($\Delta t_1 = 5 \times 10^{-6}\,[s]$) of the previous

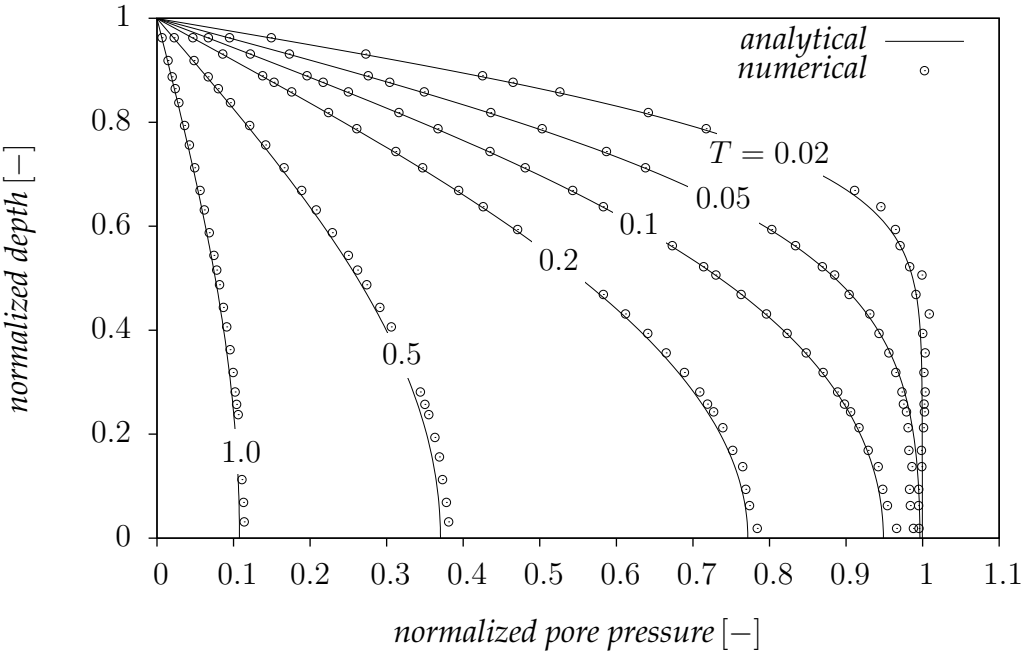


Figure 6.6: Comparison of analytical and MPM solution for one-dimensional consolidation

case until $T = 0.2$. After that a scale factor of $\beta_m = 100$ was used, allowing a time step of ($\Delta t_2 = 5 \times 10^{-5}\,[s]$). The results of Figure 6.7 show the effectiveness of the procedure.

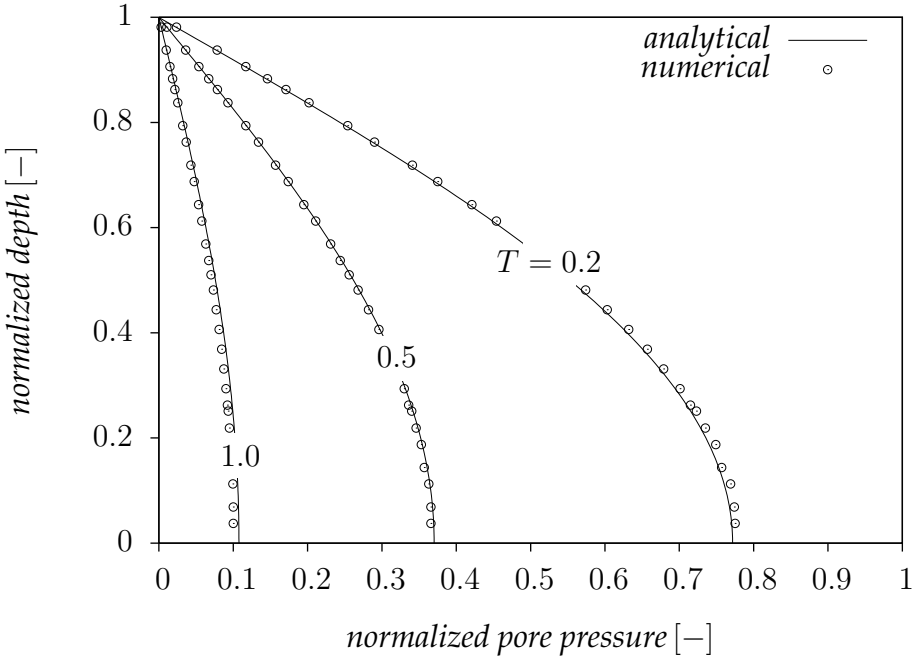


Figure 6.7: One-dimensional consolidation with mass scaling

6.8 Application example: Wave attack on sea dike

6.8.1 Problem description

More than $400\,[km]$ of the Dutch sea dikes have an asphalt revetment; mostly with a thickness of 0.2 to $0.3\,[m]$ in the wave attack zone. This asphalt revetment is typically applied above the usual tidal zone, so that it is not subjected to regular daily wave attack. During a storm, the so-called wind setup increases both the low and the high tide level; in the Netherlands typically by an additional $3.5\,[m]$. As a consequence, the revetment is severely and cyclically loaded during a storm. The duration of a Dutch design storm is $35\,[hr]$, as shown in Figure 6.8. During a storm the so-called still water level (SWL) rises, as it is the summation of the usual tidal SWL (with a period of $12\,[hr]$) and the wind setup by the storm.

The ground water pressure underneath the asphalt revetment is largely controlled by the SWL. For a very permeable soil, the phreatic ground water table (GWT) coincides more or less with the SWL, whereas it lags behind the SWL for a more impermeable soil. For a rising SWL after low tide, the GWT is below the SWL. After high tide, on the other hand, the SWL is falling and the GWT is above the SWL. The latter case creates uplift pore pressures below the revetment, also indicated in Figure 6.9. For the present analysis, the GWT is simply taken to be equal to the SWL, so that there are no uplift pressures at all.

The waves on the North Sea between England and the Netherlands have a period of 5 to $10\,[s]$, but the impact period of a braking wave on the dike revetment is of much

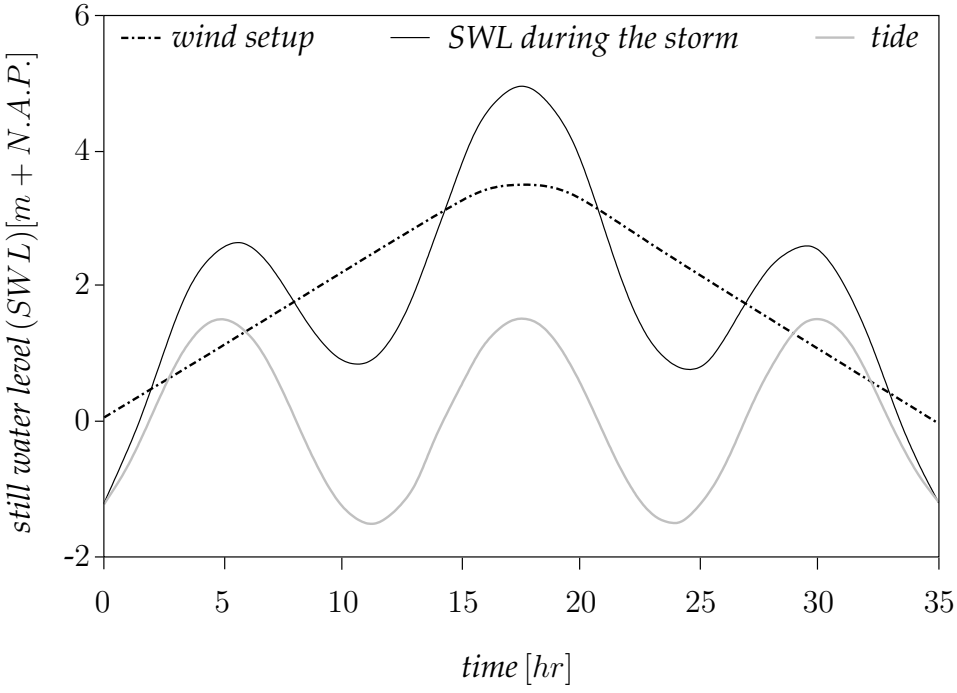


Figure 6.8: Still water level during a design storm [57]

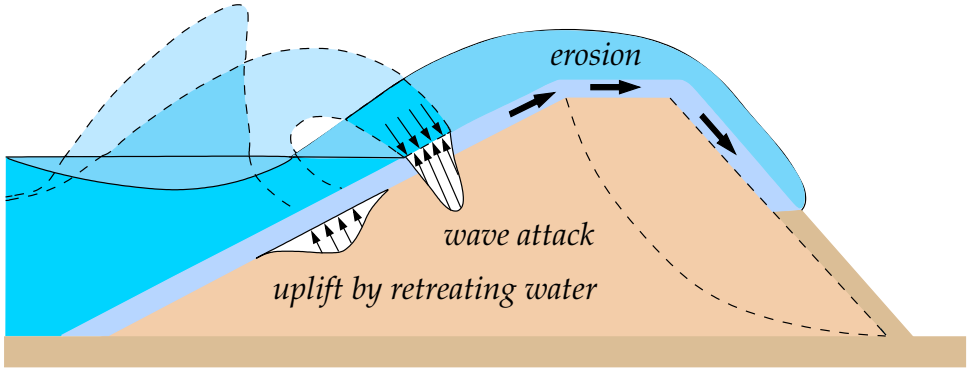


Figure 6.9: Wave attack and uplift pressure on a sea dike

shorter duration of about $0.1\,[s]$. In the following analyses, a wave period of $t_2 = 10\,[s]$ was assumed and the load impact duration was taken to be $t_1 = 0.1\,[s]$. During this very small period of loading a haversine time distribution was assumed, as indicated in the insert to Figure 6.10.

According to De Looff et al. [57], the load intensity of a braking wave is inversely proportional to the inclination of the slope, at least for a slope in the range between $1:3$ and $1:8$. Following De Looff et al. [57] and considering a slope inclination of $1:4$, a maximum wave impact pressure of $q = 240\,[kN/m^2]$ was assumed. This corresponds to a heavy wave attack during a storm with a wave height of $4\,[m]$. The spatial distribution was assumed to be triangular with a width of $2\,[m]$, as also indicated in Figure 6.10.

The maximum pressure of $240\,[kN/m^2]$ occurs at the so-called impact point, which varies in reality significantly per event. Measurements showed that this point lies most usually below the SWL. For a hydraulic conductivity of $k = 10^{-4}\,[m/s]$, as assumed in this study, the soil is very permeable and the ground water table (GWT) just below the revetment is consequently taken to be equal to the outside SWL. This implies that both above and below the revetment the quasi-static part of the (pore) water pressures is equal. The latter simplifies the analysis, as both the weight of the revetment and the saturated soil can be considered to be submerged. For the submerged unit weight, it yields

$$\gamma'_{asphalt} = \gamma_{asphalt} - \gamma_w = 24 - 10 = 14\,\left[kN/m^3\right]$$

and

$$\gamma'_{soil} = \gamma_{sat,soil} - \gamma_w = 20 - 10 = 10\,\left[kN/m^3\right].$$

All computations were carried out with submerged unit weights. Considering dynamic loading, the following mass densities were applied

$$\varrho_{asphalt} = 2400\,\left[kg/m^3\right] \quad \text{and} \quad \varrho_{s,soil} = 2667\,\left[kg/m^3\right]$$

with $\varrho_{s,soil}$ being the density of soil grains. The porosity of the soil was assumed to be $n = 0.4$, giving a dry soil density of $1600\,[kg/m^3]$ and a saturated soil density of $2000\,[kg/m^3]$. The bulk modulus of the pore water was estimated as $K_w = p/\left(1 - S_r\right)$, where p is roughly atmospheric pressure ($100\,[kN/m^2]$) and S_r is the degree of saturation

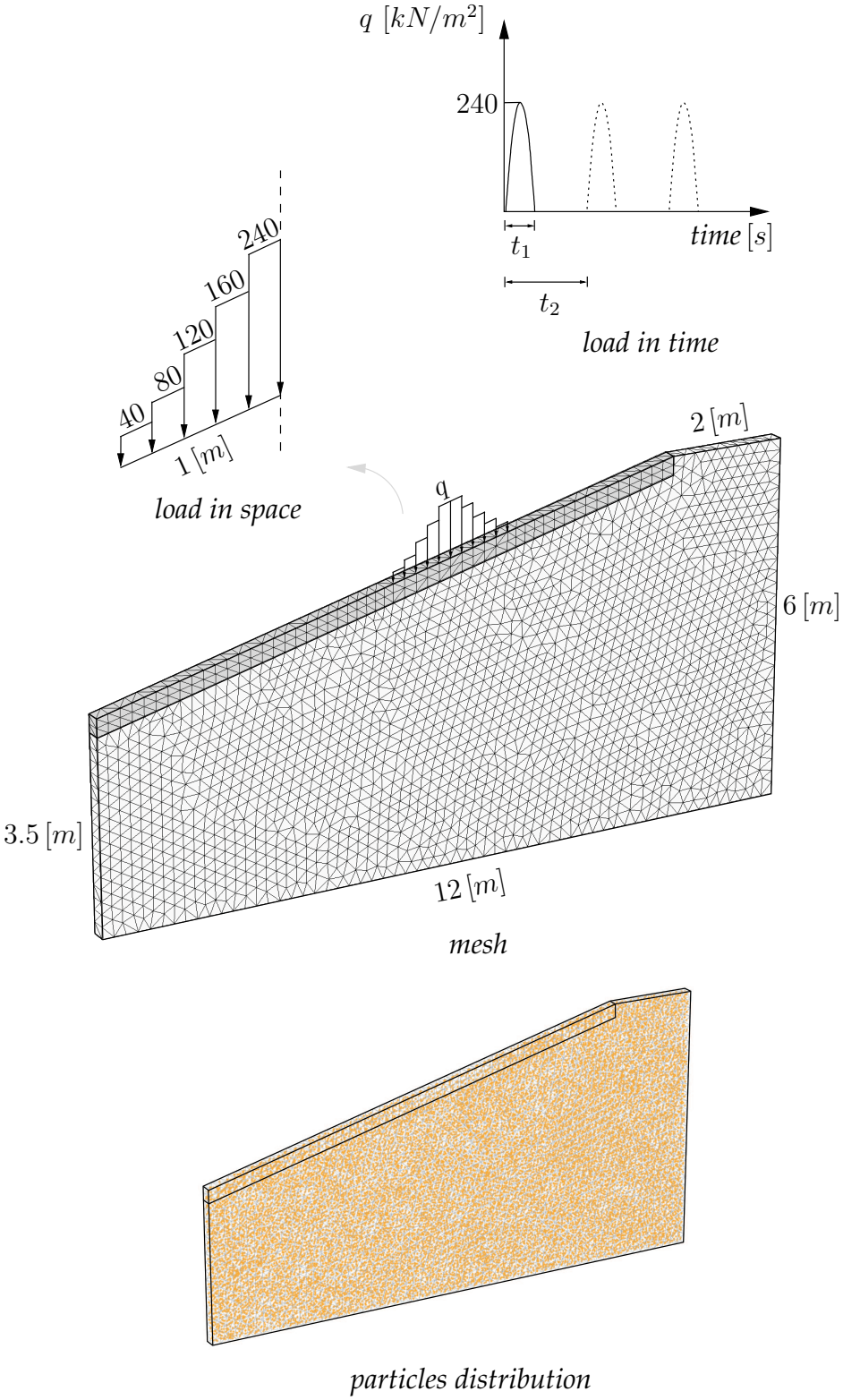


Figure 6.10: Discretization and loading conditions for the dike problem

of the soil. On assuming $S_r = 99\%$ for a soil that is being saturated by a rising ground water level during a storm, it follows that $K_w = 10\,[MN/m^2]$.

An asphalt revetment consists of bituminous concrete with a visco-plastic stress-strain behavior, as studied for instance in detail by Bonnier [32]. On using MPM, it is possible to use advanced constitutive models to model rupture (Figure 6.11) under extremely heavy wave loads, but for the present study, the simple linear elastic constitutive model was adopted. Realistic stiffnesses for bituminous concrete depend on the age and the quality of the revetment. For the present study, both $E_{asphalt} = 20\,[MN/m^2]$ and $1\,[GN/m^2]$ were used. Poisson's ratio was fixed to be 0.3.

The mesh of the problem is shown in Figure 6.10. It consists of 19786 tetrahedral elements and 19786 particles, each element having one particle. The impermeable revetment had a thickness of $0.3\,[m]$ and was divided into two layers of elements for proper simulation of bending during loading. Figure 6.10 also shows the distribution of the cyclic wave load in space and time. A repeated load pulse with a maximum intensity of $240\,[kN/m^2]$ was transferred via the asphalt to the Mohr-Coulomb soil that had Young's modulus of $10\,[MN/m^2]$, Poisson's ratio of 0.33, cohesion of $1\,[kN/m^2]$ and friction angle of 35°. These properties were fixed for all computations of the problem. Local damping was used to generate initial stresses due to gravity with a damping factor of 0.75. With regard to the dynamic analysis, no damping was applied in the domain. However, Kelvin-Voigt elements were prescribed on the boundaries with $\delta = 5.0\,[m]$ to minimize reflections. No geometry update was considered in this problem. Hence, particles were kept in their initial positions throughout the computations.

Figure 6.11: Rupture of a revetment by wave attack

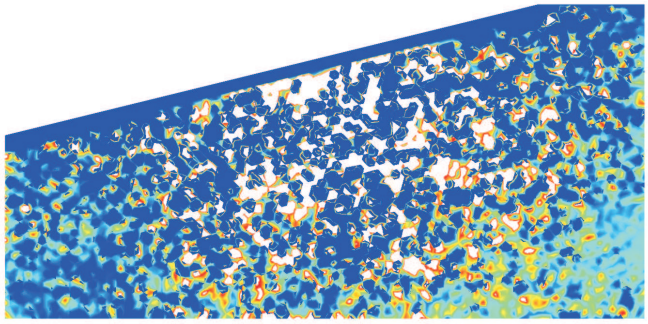
6.8.2 Simulation results

Before addressing the propagation of the pore pressure wave, the need for using the enhanced volumetric strain procedure is demonstrated in Figure 6.12. This figure compares the excess pore pressure solution for the case of no enhancement (a) with that making use of enhancement (b) corresponding to time $0.05\,[s]$. The contour scale ranges from -66 to $0\,[kN/m^2]$, with the extremes corresponding to red and blue, respectively. It is important to appreciate that the strain enhancement is part of the solution scheme, and not intended as a technique to make graphs look better. It is clear from the comparison that the enhancement mitigates the spurious spatial pressures. This example also assesses the applicability of the algorithm for a high ratio of water to soil stiffness. Water stiffness of $K_w = 2\,[GN/m^2]$ and concrete stiffness of $E_{asphalt} = 1\,[GN/m^2]$ were used in this computation.

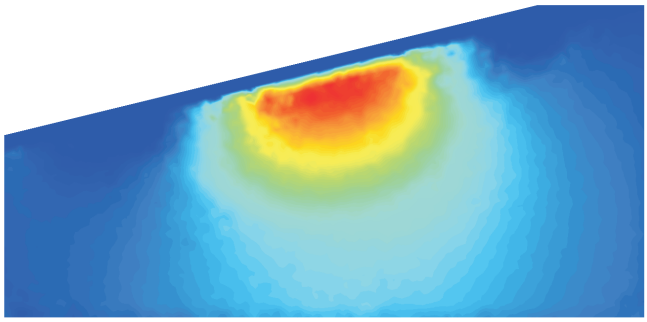
Figure 6.13 shows the evolution of excess pore pressure between 0.025 and $0.1\,[s]$, as well as that for vertical effective stress. The scale for the effective stress (from -218 to $0\,[kN/m^2]$) is different from that for pore pressure (-80 to $0\,[kN/m^2]$). An increase of excess pore pressure is observed within the soil from 0 and $0.05\,[s]$, which corresponds to the time of maximum impulse. Given that the asphalt concrete itself is impermeable, pore pressures do not develop. With regard to the effective stress plot, we have a situation where there is initially an increase in effective stress with the maximum intensity occurring in the asphalt concrete. The excess pore pressures and effective stresses decrease thereafter. This simulation corresponds to $K_w = 10\,[MN/m^2]$ and $E_{asphalt} = 20\,[MN/m^2]$.

The effect of the revetment stiffness is illustrated in Figure 6.14. This figure shows the effect of load spreading directly below the revetment. The distribution of the total stresses is shown at the peak load, i.e., at $0.05\,[s]$. For a low revetment stiffness of $E_{asphalt} = 20\,[MN/m^2]$, the soil is loaded much more locally (Figure 6.14(b)) than for the high revetment stiffness of $E_{asphalt} = 1\,[GN/m^2]$ (Figure 6.14(a)). For the low stiffness revetment, the total vertical stress directly under the impact point is $155\,[kN/m^2]$, whereas the stiffer revetment yields a much smaller load transmission ($85\,[kN/m^2]$), as might be expected. This means that the maximum total stress in the soil is approximately 65% and 35% of the maximum load intensity for the cases of low and high stiffness revetment, respectively. This is mainly due to the load spreading by the stiffer revetment. In addition a large effect of the revetment layer thickness is expected as well as a large effect of the soil stiffness.

Figure 6.15 shows the deformation pattern after 10 successive waves for a revetment with stiffness of $E_{asphalt} = 1\,[GN/m^2]$. A maximum residual displacement of about $4.4\,[cm]$ was computed for the impact point. Each wave resulted in a small contribution to the residual deformation, but one observes a so-called "shake-down". The latter is defined as an asymptotical convergence to a particular final deformation. Computational results for revetment stiffness of $E_{asphalt} = 20\,[MN/m^2]$ showed much larger deformation up to $26\,[cm]$, indicating the importance of high quality revetment.



(a)



(b)

Figure 6.12: Excess pore pressures at $t = 0.05\,[s]$ without enhancement (a) and with enhancement (b) for $K_w = 2\,[GN/m^2]$ and $E_{asphalt} = 1\,[GN/m^2]$, the scale ranges from -66 to $0\,[kN/m^2]$

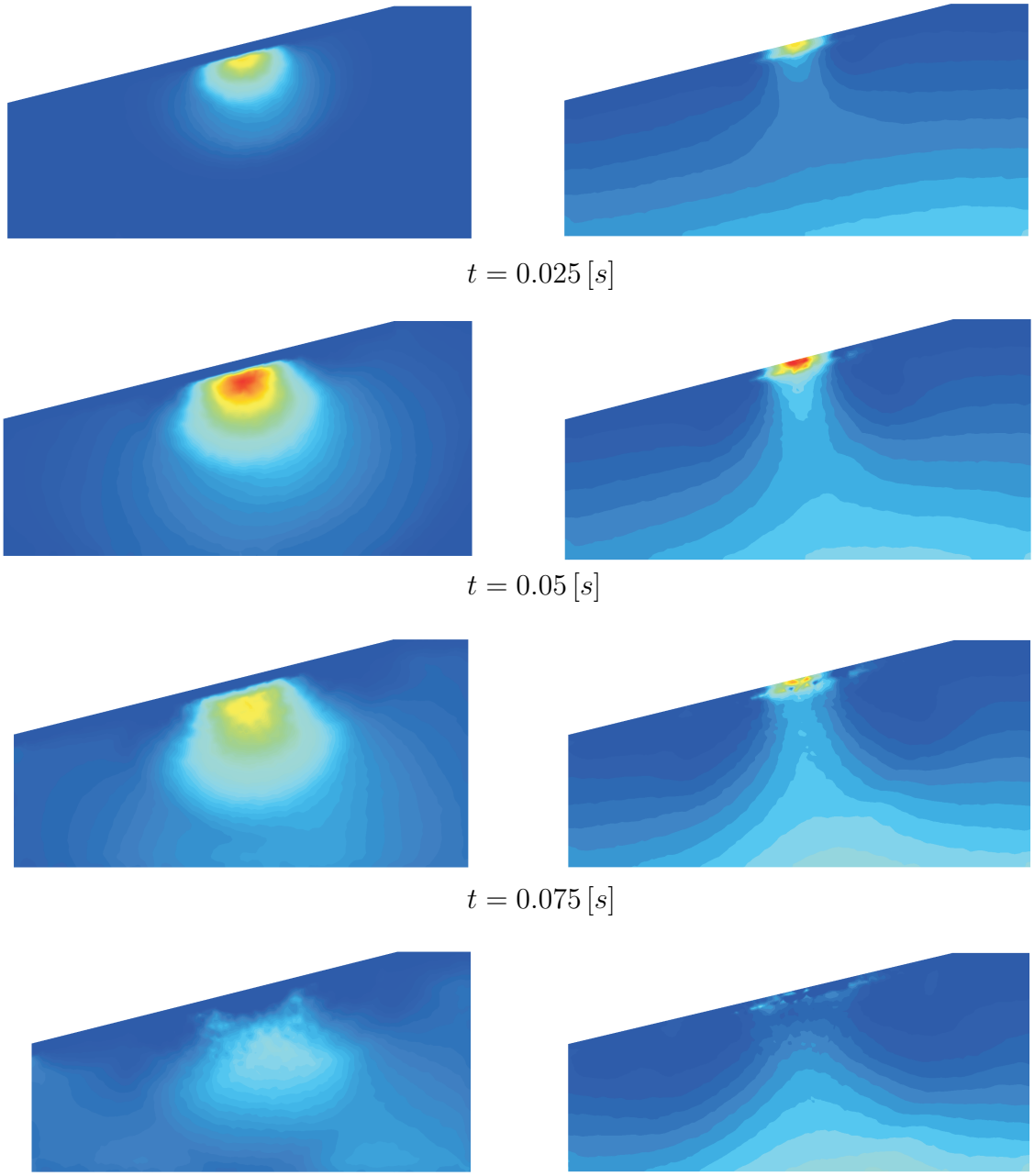
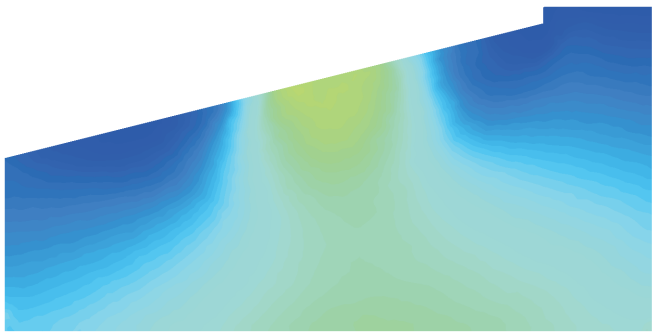
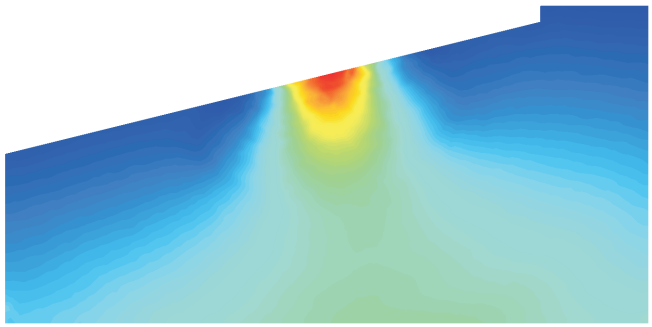


Figure 6.13: Evolution of excess pore pressure (-80 to $0\,[kN/m^2]$) and vertical effective stress (-218 to $0\,[kN/m^2]$) for the case with $K_w = 10\,[MN/m^2]$ and $E_{asphalt} = 20\,[MN/m^2]$



(a)



(b)

Figure 6.14: Vertical total stresses at $t = 0.05\,[s]$ for (a) $E_{asphalt} = 1\,[GN/m^2]$ and (b) $E_{asphalt} = 20\,[MN/m^2]$, the scale ranges from -155 to $0\,[kN/m^2]$, both cases with $K_w = 10\,[MN/m^2]$

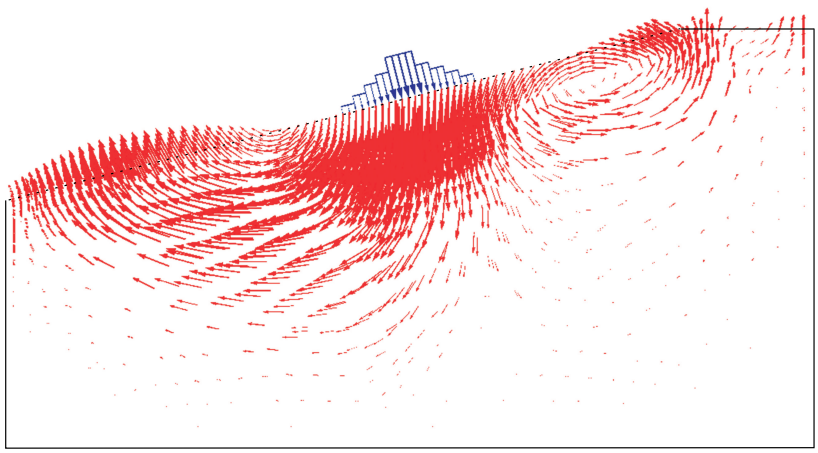


Figure 6.15: Deformation pattern after 10 successive waves

Chapter 7

Pile driving in dry sand

Pile driving is an essential installation method in foundation engineering. Engineers have to assess the size and the type of the hammer needed to drive the pile to the required depth. They also need to assess the probability of pile damage due to high driving stresses. In future, numerical simulations of pile driving may help engineers in the assessment and the investigation to design the best possible type of pile foundation.

A fully dynamic analysis of pile driving in sand using three-dimensional material point method is presented in this chapter. The pile is assumed to behave as linear elastic so that the assessment of its damage is excluded from the present study. To capture the behavior under cyclic loading, the advanced hypoplastic model [184] is adopted for sand. Attention is focused, in this study, on the drivability of a pile considering sand with different initial densities. Results of the parametric study, including the variation of the skin friction and the pile damping, are presented in this study.

Section 7.1 presents an overview of the previous studies on pile driving. The formulation of the hypoplastic model and the explicit time integration with sub-stepping technique are described in Section 7.2. Numerical simulations of laboratory tests on the so-called "Schlabendorfer" sand, which is considered in the pile driving problem of this chapter, are also shown. Section 7.3 presents the geometry and boundary conditions of the problem, including the application of the moving mesh concept to the pile driving problem, the approximation of the pile tip geometry, and the conversion of the hammer impact into surface traction applied on the pile head. The results of shallow and deep penetration are discussed in Sections 7.4 and 7.5, respectively.

7.1 Previous studies on pile driving

Saint-Venant's one-dimensional wave equation, which was derived for the wave propagation in an elastic rod, was widely used in the early analysis of pile driving problems. In 1962, Smith [148] applied this equation to a problem, in which the pile was discretized by a series of lumped masses connected by springs and the soil was represented by nonlinear springs and dashpots. Lee et al. [103] discretized the pile using one-dimensional finite elements, whereas the soil resistance to driving was represented by a series of springs and dashpots, with the springs being assumed to behave as elastic-perfectly plastic. There have been extensive studies on the analysis of stress wave propagation in driven piles. For abroad literature review about the use of the wave equation in pile driving problems, the reader is referred to the work of Goble and Rausche [71].

Although the wave problem was widely used in modeling pile driving, the predicted

results were not accurate enough, which can be attributed, among other reasons, to the one-dimensional approximation of the soil response. To predict realistic soil behavior, axisymmetric or three-dimensional modeling is required to capture all stress waves transfered from the pile to the soil and to properly model the interaction between the soil and the pile and hence the soil resistance [115].

Until the early eighties the attempts of three-dimensional modeling of pile driving problems were limited. In 1982, Smith and Chow [149] performed axisymmetric finite element analysis for driven piles in clay. They reported considerable difference in comparison to the one-dimensional analysis. This analysis was then extended by To and Smith [174] to different soil conditions and hammer characteristics. A landmark paper was published in 1994 by Mabsout and Tassoulas [115] on pile driving in clays. The problem was discretized using axisymmetric finite element technique. Bounding-surface plasticity was adopted to model the inelastic behavior of clays, with a frictional contact algorithm to characterize the interaction between the pile and the clay. Transmitting viscous elements were added to the far field boundaries to mitigate wave reflections. The impact of the hammer on the pile was simulated by a periodic force applied at the pile head. Other studies by Mabsout and coworkers on pile driving are found in references [114, 116, 117].

With the rapid development of numerical procedures and computational resources, three-dimensional problems could be modeled more easily. Henke [87, 88] analyzed different methods of pile installation using three-dimensional finite element. He analyzed the quasi-static pile jacking process and the vibratory pile driving. The effect of these different installation methods on the behavior of the soil surrounding the pile was investigated and the influence of driving a pile near existing piles was examined.

7.2 Hypoplastic model for sand

This section presents a brief overview of the hypoplastic model by von Wolffersdorff [184] and its extension to small strain stiffness by Niemunis and Herle [131]. The emphasis in this chapter is on pile driving in dry sand and therefore there is no distinction between effective and total stresses.

7.2.1 Von Wolffersdorff model

Following von Wolffersdorff [184], the hypoplastic constitutive equation can be written as

$$\overset{\nabla \mathrm{J}}{\sigma}_{ij} = G_{ij}\left(\sigma_{ij}, e, \dot{\varepsilon}_{ij}\right), \tag{7.1}$$

in which $\overset{\nabla \mathrm{J}}{\sigma}_{ij}$ is the Jaumann rate of stress, e is the current void ratio and $\dot{\varepsilon}_{ij}$ is the strain rate tensor. The rate of the void ratio with respect to the volumetric strain rate $\dot{\varepsilon}_{kk}$ is

$$\dot{e} = (1+e)\,\dot{\varepsilon}_{kk} \tag{7.2}$$

Following von Wolffersdorff [184], tensorial function G_{ij} is selected so that

$$\overset{\nabla \mathrm{J}}{\sigma}_{ij} = L_{ijkl}\dot{\varepsilon}_{kl} + N_{ij}\sqrt{\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}} \tag{7.3}$$

with L_{ijkl} and N_{ij} being the linear and non-linear constitutive tensors, respectively. The term $\sqrt{\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}}$ is the Euclidean norm of the strain rate tensor. The above decomposition of the function G_{ij} into a linear and a non-linear tensors is practical in achieving the inelastic behavior of sand without any need to decompose the strain into elastic and plastic parts [131]. The tensors L_{ijkl} and N_{ij} are functions of the stress and void ratio and are written as

$$L_{ijkl} = f_b f_e \frac{1}{\hat{\sigma}_{ij}\hat{\sigma}_{ij}} \left(f^2 \delta_{ik}\delta_{jl} + a^2 \hat{\sigma}_{ik}\hat{\sigma}_{jl} \right) \tag{7.4}$$

and

$$N_{ij} = f_d f_b f_e \frac{af}{\hat{\sigma}_{ij}\hat{\sigma}_{ij}} (\hat{\sigma}_{ij} + \hat{\hat{\sigma}}_{ij}). \tag{7.5}$$

The tensors $\hat{\sigma}_{ij}$ and $\hat{\hat{\sigma}}_{ij}$ are defined as

$$\hat{\sigma}_{ij} = \frac{\sigma_{ij}}{I_1} \qquad \text{and} \qquad \hat{\hat{\sigma}}_{ij} = \hat{\sigma}_{ij} - \frac{1}{3}\delta_{ij} \tag{7.6}$$

with

$$I_1 = \sigma_{11} + \sigma_{22} + \sigma_{33}. \tag{7.7}$$

Other parameters of Equations 7.4 and 7.5 are defined as

$$a = \frac{\sqrt{3}\,(3 - \sin\phi_c)}{2\sqrt{2}\sin\phi_c} \quad \text{and} \quad f = \sqrt{\frac{1}{8}\tan^2\hat{\hat{\psi}} + \frac{2 - \tan^2\hat{\hat{\psi}}}{2 + \sqrt{2}\tan\hat{\hat{\psi}}\cos 3\vartheta}} - \frac{1}{2\sqrt{2}}\tan\hat{\hat{\psi}}, \tag{7.8}$$

where ϕ_c is the critical state friction angle and the angles $\hat{\hat{\psi}}$ and 3ϑ are defined as

$$\tan\hat{\hat{\psi}} = \sqrt{3\,\hat{\hat{\sigma}}_{ij}\hat{\hat{\sigma}}_{ij}} \qquad \text{and} \qquad \cos 3\vartheta = -\sqrt{6}\,\frac{\delta_{ji}(\hat{\hat{\sigma}}_{ik}\hat{\hat{\sigma}}_{kl}\hat{\hat{\sigma}}_{lj})}{(\hat{\hat{\sigma}}_{ij}\hat{\hat{\sigma}}_{ij})^{\frac{3}{2}}}. \tag{7.9}$$

Let us define three characteristic void ratios, namely the maximum possible void ratio e_i corresponding to the minimum density, the minimum possible void ratio e_d corresponding to the maximum density and the critical void ratio e_c; see Figure 7.1. Also shown, are three void ratios e_{i0}, e_{d0} and e_{c0} corresponding to vanishing pressure represented by the first stress invariant I_1. In soil mechanics, e_{i0} and e_{d0} are usually denoted as e_{max} and e_{max}, respectively. Density dependency is taken into account by factors f_e and f_d, being defined as

$$f_e = \left(\frac{e_c}{e}\right)^{\beta} \qquad \text{and} \qquad f_d = \left(\frac{e - e_d}{e_c - e_d}\right)^{\bar{\alpha}} \tag{7.10}$$

with β and $\bar{\alpha}$ being input indices. The pressure dependency of the soil stiffness is represented by the factor f_b, being defined as

$$f_b = \frac{h_s}{\bar{n}} \left(\frac{1 + e_i}{e_i}\right) \left(\frac{e_{i0}}{e_{c0}}\right)^{\beta} \left(\frac{-I_1}{h_s}\right)^{1-\bar{n}} \left[3 + a^2 - \sqrt{3}\,a \left(\frac{e_{i0} - e_{d0}}{e_{c0} - e_{d0}}\right)^{\bar{\alpha}}\right]^{-1}, \tag{7.11}$$

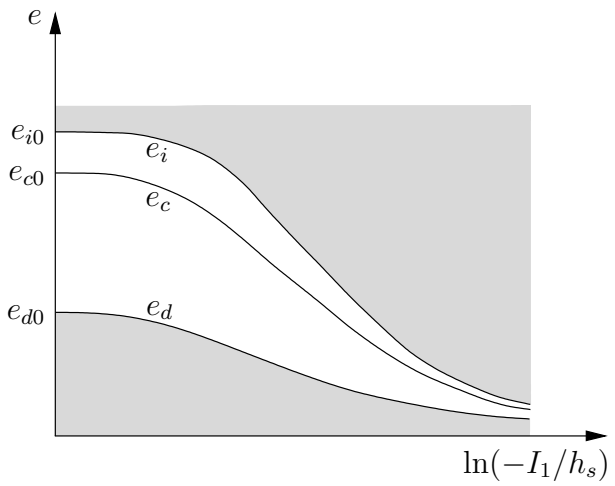


Figure 7.1: The characteristic void ratios [131]

where h_s is the pressure independent granular stiffness and $\bar{n}$ is an input index. The characteristic void ratios are updated according to Bauer's formula [17] as

$$\frac{e_i}{e_{i0}} = \frac{e_c}{e_{c0}} = \frac{e_d}{e_{d0}} = \exp\left[-\left(\frac{-I_1}{h_s}\right)^{\bar{n}}\right]. \tag{7.12}$$

Substituting Equations 7.4 and 7.5 into Equation 7.3 yields the final constitutive equation for von Wolffersdorff model

$$\overset{\nabla \mathrm{J}}{\sigma}_{ij} = f_b f_e \frac{1}{\hat{\sigma}_{ij}\hat{\sigma}_{ij}} \left[f^2 \dot{\varepsilon}_{ij} + a^2 \left(\hat{\sigma}_{ik}\hat{\sigma}_{jl}\right) \dot{\varepsilon}_{kl} + a f f_d (\hat{\sigma}_{ij} + \hat{\hat{\sigma}}_{ij}) \sqrt{\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}} \right]. \tag{7.13}$$

7.2.2 Extension to small strain stiffness after Niemunis and Herle

The major shortcoming of the von Wolffersdorff model is ratcheting in cyclic loading. Therefore, the model was improved by Niemunis and Herle [131], in which they extended the formulation to account for small strain stiffness, by introducing a new state variable called the intergranular strain, denoted as S_{ij} and calculated as

$$S_{ij} = \int \dot{S}_{ij} dt, \tag{7.14}$$

where

$$\dot{S}_{ij} = \begin{cases} (\delta_{ik}\delta_{jl} - \hat{S}_{ik}\hat{S}_{jl} r^{\bar{\beta}}) \dot{\varepsilon}_{kl} & \text{for } (\hat{S}_{ij}\dot{\varepsilon}_{ij}) > 0 \\ \\ \dot{\varepsilon}_{ij} & \text{for } (\hat{S}_{ij}\dot{\varepsilon}_{ij}) \leqslant 0 \end{cases} \tag{7.15}$$

with $\hat{S}_{ij}$ being the direction tensor of the intergranular strain, defined as

$$\hat{S}_{ij} = \begin{cases} S_{ij}/\sqrt{S_{ij}S_{ij}} & \text{for } S_{ij} \neq 0 \\ \\ 0 & \text{for } S_{ij} = 0 \end{cases} \tag{7.16}$$

with $\bar{\beta}$ being a material constant controlling the degradation of the stiffness. The normalized magnitude of S_{ij} is denoted by r and calculated as

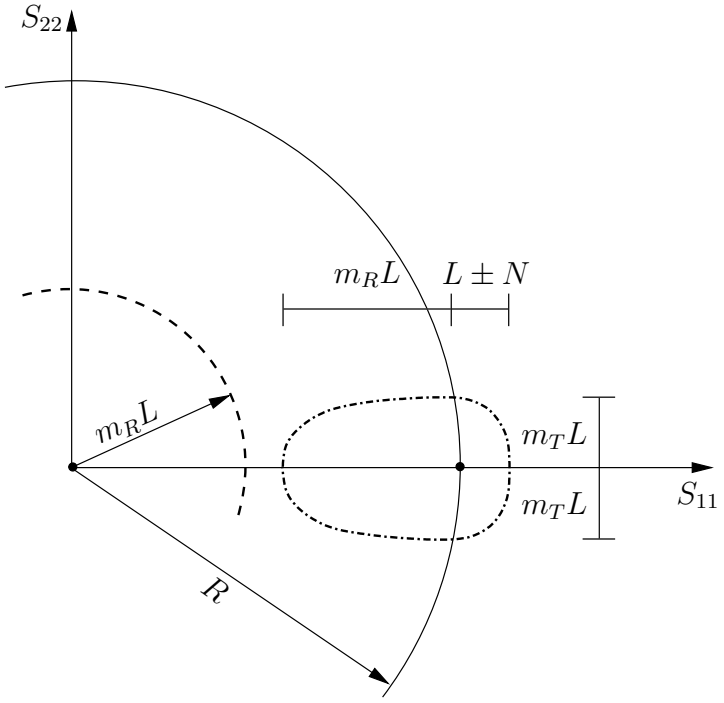
$$r = \frac{\sqrt{S_{ij}S_{ij}}}{R}, \tag{7.17}$$

where R is an input parameter for the size of the elastic range in strain space. In most papers on hypoplastic model, r is denoted as ρ, but we do not use ρ to avoid the confusion with the mass density. Following reference [131], the constitutive relation of the extended hypoplastic model is written as

$$\overset{\nabla \mathrm{J}}{\sigma}_{ij} = D_{ijkl}\dot{\varepsilon}_{kl} \tag{7.18}$$

with D_{ijkl} being the stiffness tensor constructed from the tensors L_{ijkl} and N_{ij}. It is modified by the two scalars m_T and m_R, depending on r and $(\hat{S}_{ij}\dot{\varepsilon}_{ij})$ as shown in Figure 7.2. This tensor is written as

$$\begin{aligned} D_{ijkl} = &\left[r^{\chi} m_T + (1 - r^{\chi})\, m_R\right] L_{ijkl} + \\ &\begin{cases} r^{\chi}\,(1 - m_T)\, L_{ijmn}(\hat{S}_{mk}\hat{S}_{nl}) + r^{\chi} N_{ik}\hat{S}_{jl} & \text{for } (\hat{S}_{ij}\dot{\varepsilon}_{ij}) > 0 \\ r^{\chi}\,(m_R - m_T)\, L_{ijmn}(\hat{S}_{mk}\hat{S}_{nl}) & \text{for } (\hat{S}_{ij}\dot{\varepsilon}_{ij}) \leqslant 0 \end{cases} \end{aligned} \tag{7.19}$$



- - - stress response for $r = 0$ *-·- stress response for* $r = 1$

Figure 7.2: Stiffness dependency on m_R and m_T for $r = 0$ and $r = 1$ [131]

with χ being an index controlling the stiffness degradation. In elastoplastic modeling, the case of $(\hat{S}_{ij}\dot{\varepsilon}_{ij}) > 0$ indicates loading and $(\hat{S}_{ij}\dot{\varepsilon}_{ij}) \leqslant 0$ indicates unloading.

To avoid numerical problems associated with vanishing mean stress for cohesionless soil, (artificial) cohesion can be introduced by shifting the mean stress by a value of p_t. The stress tensor thus becomes

$$\sigma_{ij} = \sigma_{ij} - p_t\delta_{ij}. \tag{7.20}$$

For all computations of this chapter we adopted $p_t = 1\,[kN/m^2]$.

7.2.3 Explicit time integration with sub-stepping

In all previous implementations, the Runge-Kutta method was adopted in the integration of the hypoplastic model. However, in this thesis we adopt the explicit Euler forward integration scheme with sub-stepping technique. This technique tracks the change in the mobilized friction angle ϕ_{mob}, which is calculated as

$$\sin^2\phi_{mob} = \frac{9 + (I_1 I_2/I_3)}{1 + (I_1 I_2/I_3)} \tag{7.21}$$

with I_1, I_2 and I_3 being the first, second and the third stress invariant, respectively. The change in the mobilized friction angle, denoted as $\Delta\phi_{mob}$, is calculated as

$$\Delta\phi_{mob} = \left|\phi^1_{mob} - \phi^0_{mob}\right| \tag{7.22}$$

with the superscripts 0 and 1 indicating the initial and the updated value for the time step, respectively. If $\Delta\phi_{mob} > tol$, the strain increment is divided into n_{inc} sub-increments with

$$n_{inc} = INT\left(\frac{\Delta\phi_{mob}}{tol}\right) + 2, \tag{7.23}$$

in which *INT* indicates the integer of the value. The tolerance tol was set to 1° for all computations of this chapter.

7.2.4 Calculation of the constrained modulus

For the present MPM formulation, the time integration of the momentum equation is restricted by the CFL condition as explained in Section 3.4.2. This condition requires the constrained modulus E^c to calculate the wave speed and eventually the size of the critical time step; see Equation 3.117. Let us consider the case of one-dimensional straining as in the odometer test. Hence, $\dot{\varepsilon}_{22} \neq 0$, but all other components of the strain rate tensor are zeros. For the sake of simplicity, stresses are assumed to be isotropic and it follows from Equation 7.6 that

$$\hat{\sigma}_{ij} = \frac{1}{3}\delta_{ij} \qquad \text{and} \qquad \hat{\dot{\sigma}}_{ij} = 0 \tag{7.24}$$

with $\hat{\sigma}_{ij}\hat{\sigma}_{ij} = \frac{1}{3}$. Given these conditions, Equations 7.9 and 7.8 result in $\hat{\psi} = 0$ and $f = 1$, respectively. It can be shown that

$$L = f_b f_e\left(3 + \frac{a^2}{3}\right) \qquad \text{and} \qquad N = a f_d f_b f_e \tag{7.25}$$

with L and N being the components of the tensors L_{ijkl} and N_{ij}, respectively, corresponding to $\dot{\varepsilon}_{22}$. Adopting the von Wolffersdorff model (Equation 7.3), the constrained modulus becomes

$$E^c = L \mp N \tag{7.26}$$

with the minus and the plus sign indicating the loading and unloading conditions, respectively. Applying the same conditions to the extended model after Niemunis and Herle yields the following constrained modulus

$$\begin{aligned} E^c = & \left[r^\chi m_T + (1 - r^\chi)\, m_R\right] L + \\ & \begin{cases} r^\chi\,(1 - m_T)\, L + r^\chi N & \text{for } (\hat{S}_{ij}\dot{\varepsilon}_{ij}) > 0 \\ \\ r^\chi\,(m_R - m_T)\, L & \text{for } (\hat{S}_{ij}\dot{\varepsilon}_{ij}) \leqslant 0. \end{cases} \end{aligned} \tag{7.27}$$

7.2.5 Numerical simulations of laboratory tests on "Schlabendorfer" sand

The properties of the Schlabendorfer sand for use in the pile driving problems of this chapter are listed in Tables 7.1 and 7.2. Simulations of the oedometer and triaxial tests were carried out with initial relative densities of $I_d = 30\%$, 63% and 80%, corresponding to initial void ratios of $e_0 = 0.83$, 0.645 and 0.55, respectively.

Oedometer test Simulations were performed by increasing the compressive vertical stress up to $100\,[kN/m^2]$ followed by unloading, in which the stress was decreased to $50\,[kN/m^2]$, and then a reloading up to $100\,[kN/m^2]$. Figure 7.3 shows the vertical stress versus the vertical strain for this case. The effect of small strain stiffness is observed up to a strain of approximately 7×10^{-4}. This is in-line with the prescribed small strain range of $R = 10^{-4}$. One notices some hysteresis corresponding to the case of $I_d = 30\%$,

Table 7.1: Hypoplastic model parameters of Schlabendorfer sand

parameter	ϕ_c	h_s	$\bar{n}$	e_{d0}	e_{c0}	e_{i0}	$\bar{\alpha}$	β
unit	$[deg]$	$[MN/m^2]$	$[-]$	$[-]$	$[-]$	$[-]$	$[-]$	$[-]$
value	33	1600	0.19	0.44	0.85	1.0	0.25	1.0

Table 7.2: Small strain stiffness parameters of Schlabendorfer sand

parameter	m_T	m_R	R	$\bar{\beta}$	χ
unit	$[-]$	$[-]$	$[-]$	$[-]$	$[-]$
value	2	5	0.0001	0.5	6

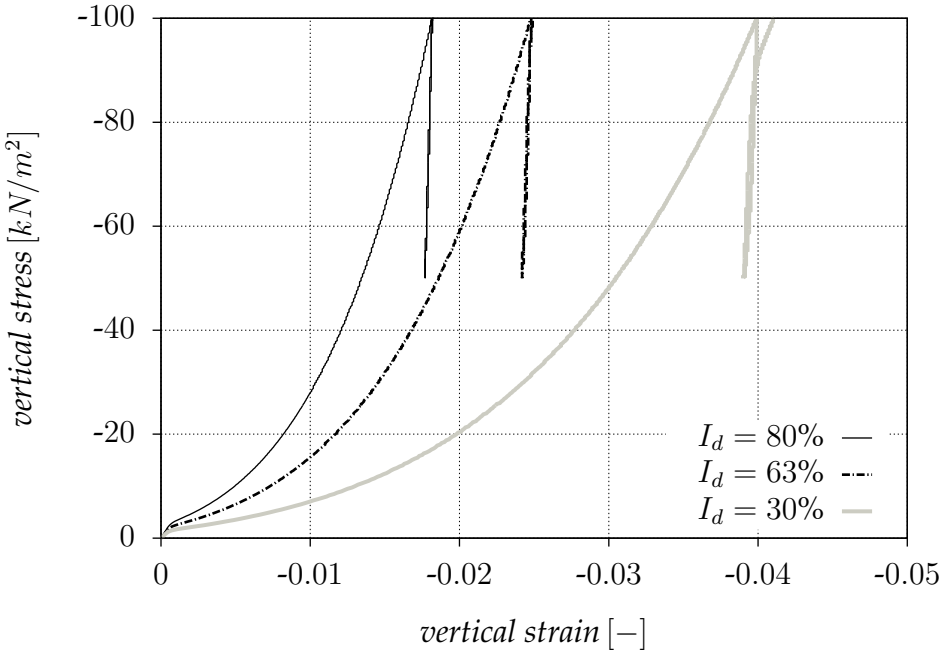


Figure 7.3: Stress-strain curves of the oedometer problem with loading-unloading-reloading

but it is not apparent for $I_d = 63\%$ and 80%. The constrained modulus E^c at a stress of $100\,[kN/m^2]$ (loading) was found to be 6870, 10400 and $13700\,[kN/m^2]$ for $I_d = 30\%$, $I_d = 63\%$ and $I_d = 80\%$, respectively. It was calculated as the tangent of the stress-strain curve at the considered value of stress, i.e.,

$$E^c = \frac{d\sigma_{22}}{d\varepsilon_{22}}, \tag{7.28}$$

in which σ_{22} is the vertical stress and ε_{22} is the corresponding vertical strain. On applying Equation 7.27, one gets $E^c = 8261$, 12100 and $15460\,[kN/m^2]$ for $I_d = 30\%$, $I_d = 63\%$ and $I_d = 80\%$, respectively. One notices that Equation 7.27 overpredicts the constrained modulus as it assumes an isotropic stress state.

Considering data on E^c-values from different sources and for different normally consolidated quartz sands, Vermeer [181] proposed the correlation $E^c = 60 I_d\,[MN/m^2]$ for the tangent modulus at a reference stress of $100\,[kN/m^2]$. For the range $30\% < I_d < 80\%$, this correlation yields the range $18 < E^c < 48\,[MN/m^2]$, which is well beyond the predicted values from the present numerical simulations. In fact, the present data would seem to be typical for a silty sand rather than a pure sand, but a silty sand would yield lower peak friction angles than the ones of the present "Schlabendorfer" sand, see discussion in the next paragraph. It is thus concluded that the input data being used for this material is not typical for a quartz sand and neither for a silty sand.

Triaxial compression test For the triaxial compression test, a confining pressure of $100\,[kN/m^2]$ was applied and then the vertical stress was increased until failure. Figure

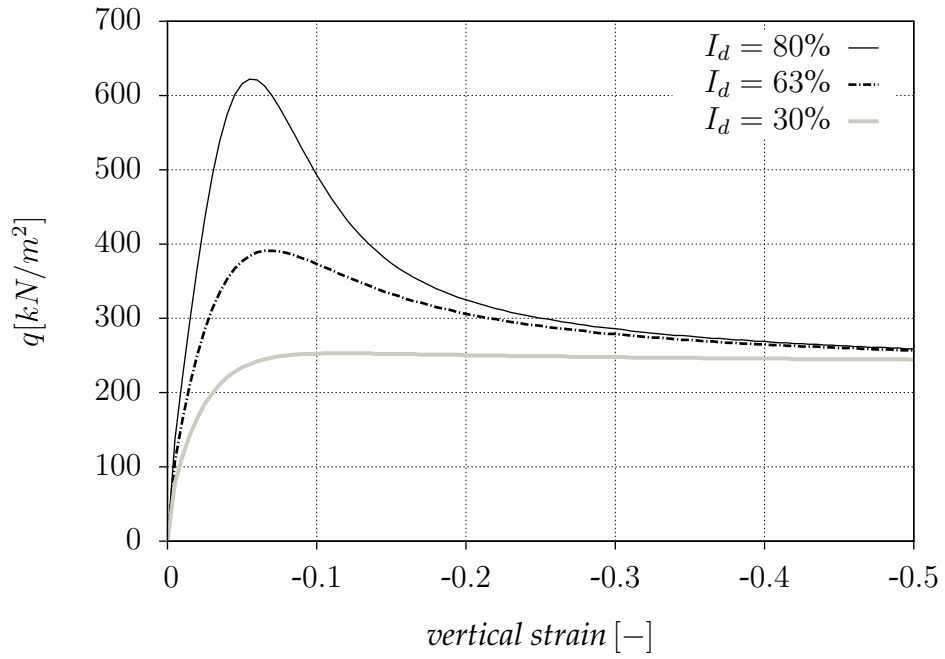


Figure 7.4: Stress-strain curves for the triaxial test

7.4 shows the stress-strain curves for different values of I_d, with q being the difference between the principal stresses σ_1 and σ_3, i.e., $q = |\sigma_1 - \sigma_3|$. The maximum value of q, denoted as q_{max}, was found to be 253, 391 and $623\,[kN/m^2]$ for $I_d = 30\%$, 63% and 80%, respectively, corresponding to peak friction angles of $33.8°$, $41.5°$ and $49°$. The curves for the volumetric strain versus the vertical strain are shown in Figures 7.5.

Bolton [30] published an experimental database on peak friction angles of quartz sands. Considering stress levels as reached in the present triaxial test simulations, this database indicates peak friction angles in the range $33° < \phi_{peak} < 40°$, at least for relative densities in the range $30\% < I_d < 80\%$. Hence, for low relative densities, the hypoplastic model would seem to yield realistic peak friction angles, but with $\phi_{peak} = 49°$ the model over-predicts the peak friction angle for $I_d = 80\%$.

Here it should be realized that the hypoplastic model does not include stress-level dependency of the peak friction angle, as clearly observed for dense sands. Considering low stress levels up to $20\,[kN/m^2]$, Bolton's data does show high peak friction angles approaching even the above value of $49°$. Unfortunately, stress levels are high in pile driving, which makes the model less attractive for the simulation of pile driving in dense sands. In such cases, resulting peak friction angles from hypoplastic input parameters would need special attention.

The stiffness modulus E_{50} Figure 7.6 illustrates the procedure of calculating the stiffness modulus E_{50} from the triaxial test. It corresponds to q at 50% of the strength. Applying this procedure to the curves of Figure 7.4, one finds that $E_{50} = 10500$, 14770 and $19540\,[kN/m^2]$ for $I_d = 30\%$, 63% and 80%, respectively, with the respective ratio E_{50}/E^c being 1.53, 1.42 and 1.43.

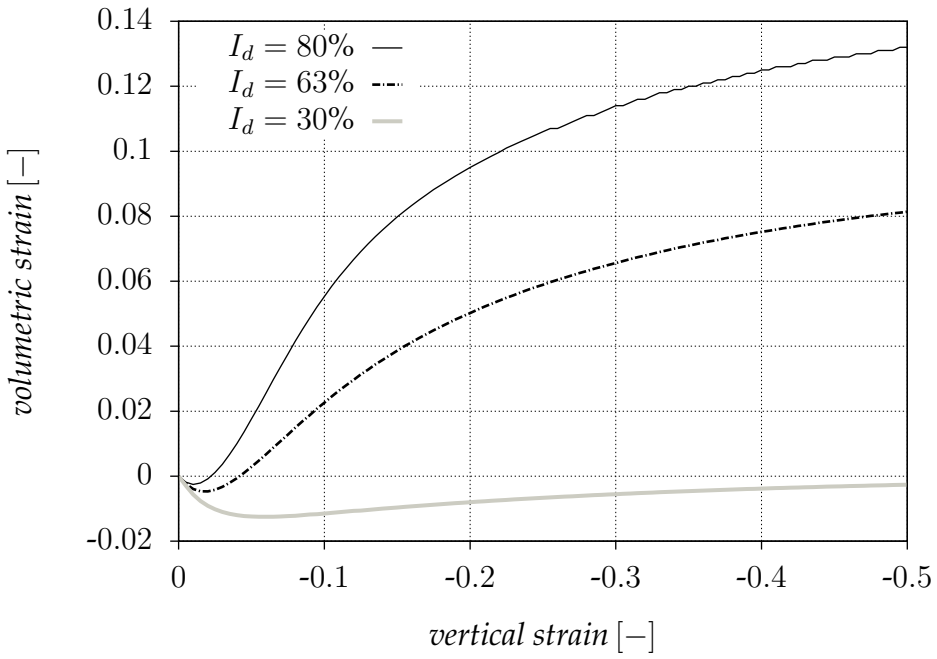


Figure 7.5: Volumetric strain versus vertical strain for the triaxial test

A database on sand stiffnesses in triaxial test was published by Schanz [144]. Considering data for many different quartz sands in the void ratio range $0.5 < e < 0.9$, stiffnesses in standard drained triaxial tests were found to be in the range $10 < E_{50} <$

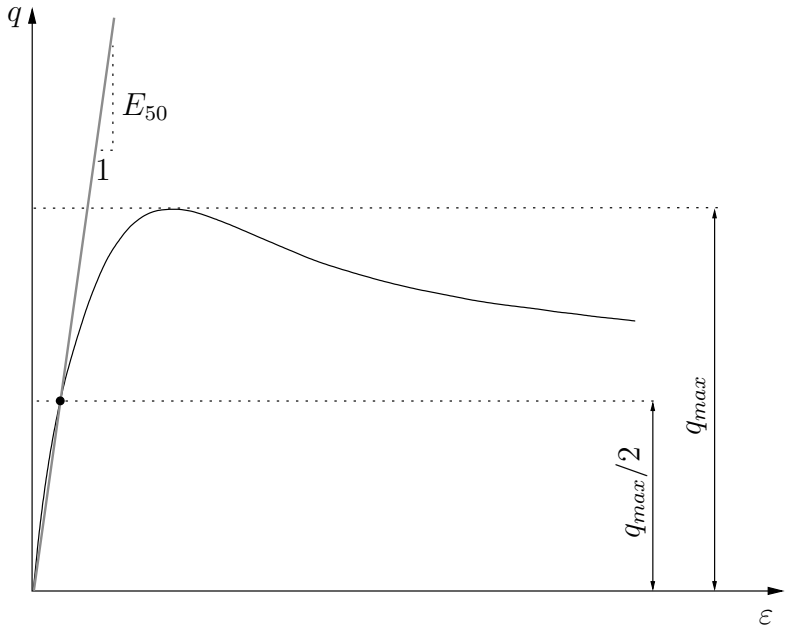


Figure 7.6: Illustration of the procedure of calculating the stiffness modulus E_{50} from triaxial test results

$60\,[MN/m^2]$, at least for tests with a confining pressure of $100\,[kN/m^2]$. Considering void ratios, the "Schlabendorfer" sand fits in, but its stiffness range of $10 < E_{50} < 20\,[MN/m^2]$ is out. Similarly, the "Schlabendorfer" stiffness ratio of about 1.5 indicates a very silty sand rather than a pure sand with a typical stiffness ratio of about one [145].

7.3 Geometry and boundary conditions

In this section, the geometry, boundary and loading conditions of the problem are presented.

7.3.1 Application of the moving mesh concept to the pile driving problem

The concept of moving mesh of Section 4.5 was adopted in the modeling of the pile driving problem. The computational domain was divided into a moving mesh and a compressed mesh zones as illustrated in Figure 7.7. The moving mesh zone is attached to the pile, i.e., as the pile penetrates into the soil, this zone moves with the same displacement as the pile. The elements of this zone keep the same shape throughout the computations. In contrast to the moving mesh zone, the elements of the compressed mesh zone are compressed during the computations. The compression of this zone is linearly distributed along its depth, with the nodes of its top surface getting the same displacement of the pile and those of the bottom surface having zero displacements.

Because of this procedure, the fine mesh is always kept around the pile and the occurrence of elements having pile and sand particles is avoided. Adopting this procedure, the pile is simulated in a way similar to that of Lagrangian finite elements and hence the load is applied at the nodes of the pile head and never mapped to particles. Moreover, the Lagrangian pile eliminates the need of identifying the new interface between the pile and the soil during computations as the interface nodes coincide with the geometry of the pile throughout the computations. As a consequence, the unit normal vectors, which are required in the contact algorithm do not change and hence the inaccuracy of recomputing them is eliminated.

We should emphasize that, because of the downwards movement of the moving mesh zone, a region with initially empty elements is required above the soil surface to accommodate the material during computations, see Figure 7.7. The elements of the compressed zone must have a reasonable aspect ratio (vertical to radial dimensions) in the initial configuration to avoid mesh distortion after considerable penetration as these elements are continuously compressed during the computations as explained above.

7.3.2 Pile tip geometry

For all computations, the diameter of the pile was fixed to $0.3\,[m]$ with a tip angle of 68° and a tip length of $0.34\,[m]$ as shown in Figure 7.8. In order to avoid numerical difficulties, the sharp transition from the shaft to the tip was made curved.

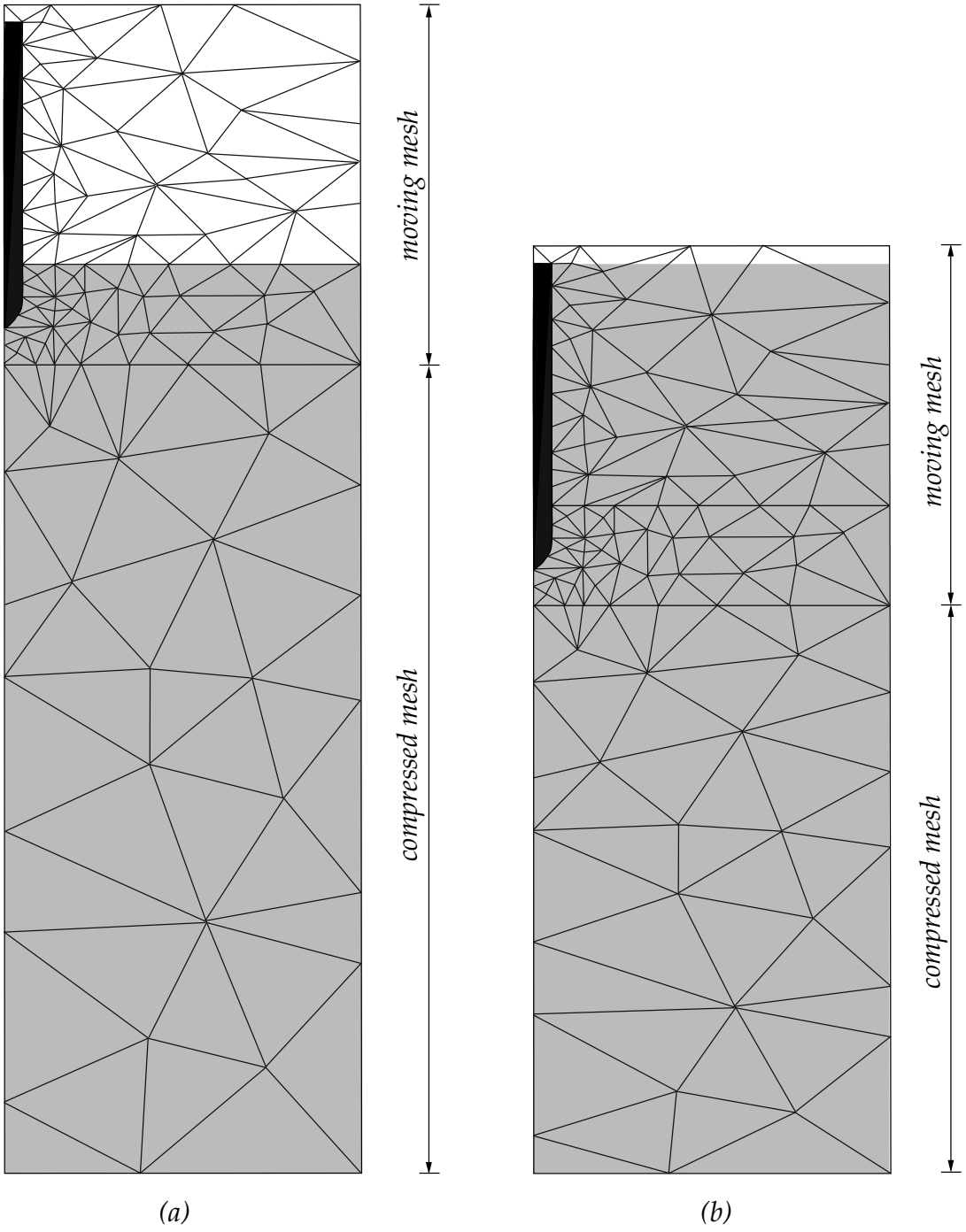


Figure 7.7: Concept of moving mesh for the pile driving problem: (a) initial configuration with embedded pile (b) after pile penetration

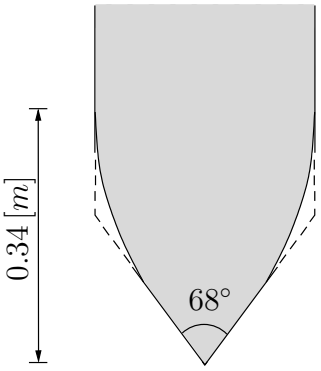


Figure 7.8: Pile tip geometry

As the problem is axisymmetric, only a sector was modeled and the line of symmetry which passes through the tip point had to have roller boundaries for proper modeling of the symmetry. The non-regular displacement of the tip point could not be modeled easily as also indicated in Figure 7.9(left half). We observe that the use of roller boundaries at the tip point resulted in a downward movement of the soil beneath the tip point. However, allowing the tip point to be free, improved the displacement considerably as indicated in Figure 7.9(right half). In this case, the displacement arrow at the tip point is normal to the pile, as one might expect.

7.3.3 Converting the hammer impact into surface traction

In practice the pile is driven by dropping a hammer from a certain height successively as indicated in Figure 7.10(a), in which m is the mass of the hammer, h is the drop height and $v = \sqrt{2gh}$ is the free falling velocity. The procedure of dropping the hammer can be modeled using MPM, but for simplicity, the momentum gained by the hammer during the free fall was converted into a traction acting at the pile head with a pre-defined function in time. Reference [71] provided measurements data for the loading function, with its time history being shown in Figure 7.11(left), where t_1 is the load duration and f_{max} is the peak load. The function was approximated as a so-called haversine as shown in Figure 7.11(right). For any hammer blow, the approximated loading function reads

$$f(t) = \begin{cases} f_{max} \sin\left(\dfrac{\pi t}{t_1}\right) & \text{for } t \leqslant t_1 \\ \\ 0 & \text{for } t > t_1. \end{cases} \tag{7.29}$$

The peak load f_{max} can be calculated from the following momentum balance

$$\int_0^{t_1} f_{max} \sin\left(\frac{\pi t}{t_1}\right) dt = \eta m \sqrt{2gh} \tag{7.30}$$

and hence

$$f_{max} = \frac{\pi \eta m \sqrt{2gh}}{2t_1} \tag{7.31}$$

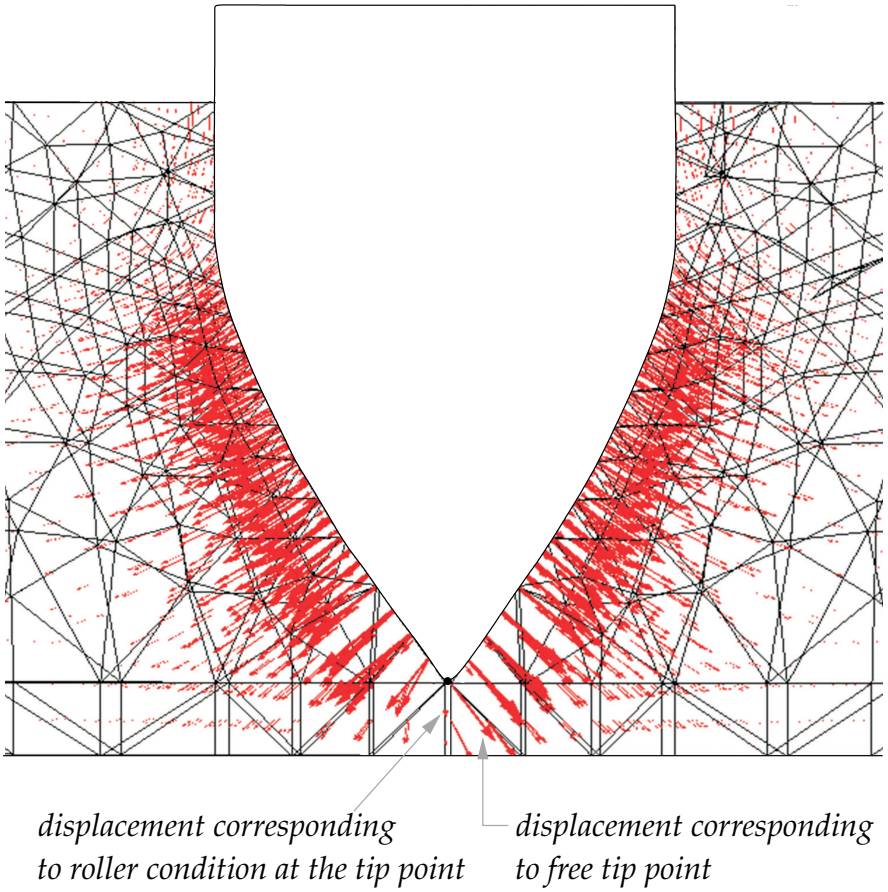


Figure 7.9: Boundary conditions of the pile tip point: (left half) roller boundaries (right half) free from constraints

with η representing the hammer efficiency, i.e, the fraction of load transfered to the pile. Following Borja [33], we adopted $\eta = 64\%$ for all computations of this chapter. The peak pressure p_{max} at the pile head can now be written as

$$p_{max} = \frac{\pi \eta m \sqrt{2gh}}{2t_1 A}, \tag{7.32}$$

in which A is the cross-sectional area of the pile.

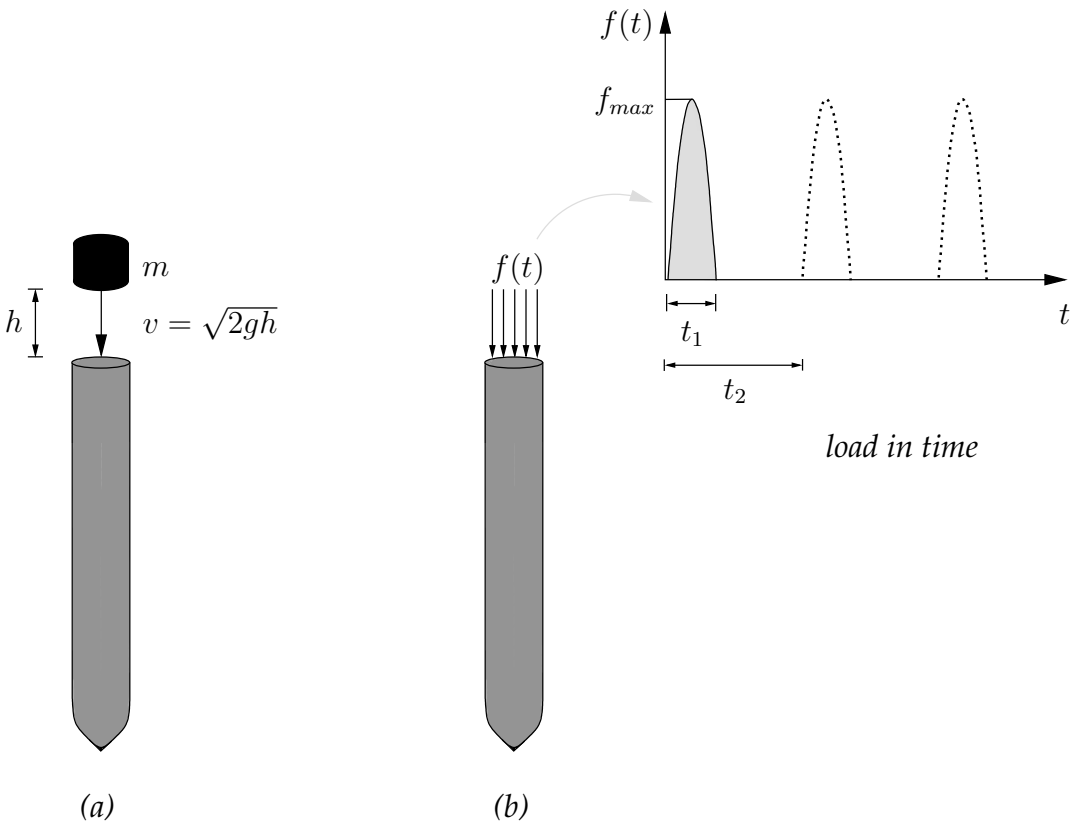


Figure 7.10: Illustration of the hammer drop (a) and the corresponding forcing function (b)

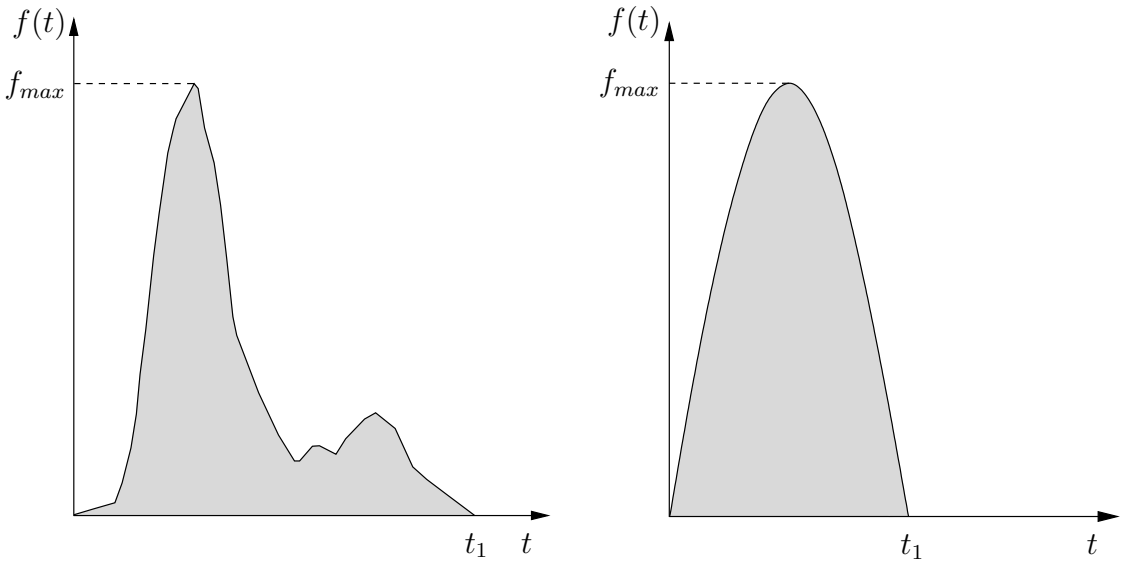


Figure 7.11: The forcing function for pile driving: (left) measured [71] and (right) the approximation to haversine time distribution

7.4 Shallow penetration

7.4.1 Problem description

In this section, a shallow pile driving problem is analyzed. The finite element mesh for the problem is shown in Figure 7.12, along with a moving and a compressed parts. Taking advantage of the symmetry of the problem, only a sector of 20° was discretized. The soil domain had a height of $1.6\,[m]$ and a width of $1.2\,[m]$. The compressed mesh had an initial height of $0.9\,[m]$, whereas the moving mesh zone had an initial height of $1.45\,[m]$ including the initially empty mesh, which was required to accommodate sand particles during computations, see Section 7.3.1.

A relatively fine mesh was used around the pile tip. The mesh had a total number of elements of 5256 including the initially inactive elements, with the total number of particles being 56704. Four particles per pile element were used, with each element in the compressed mesh zone having initially 20 particles, whereas each active element in the moving mesh zone had initially 10 particles. An aspect ratio of 2 to 1 (vertical to radial) was selected for the elements of the compressed zone to a radial dimension of $0.15\,[m]$.

The pile, which was initially embedded in the soil to a height of $0.4\,[m]$ from its tip point, was assumed to behave according to the linear elastic constitutive relation, with a Young's modulus of $E = 500\,[MN/m^2]$, a Poisson's ratio of $\nu = 0$ and a density of $\varrho = 2500\,[kg/m^3]$. The Schlabendorfer sand had the material parameters that are listed in Tables 7.1 and 7.2. The mass density of the sand grains was assumed to be $\varrho_s = 2667\,[kg/m^3]$. Although we consider sand with different initial relative densities, all computations of this chapter were preformed with a dry density of $\varrho_d = 1600\,[kg/m^3]$. Dry sand is rare in reality, but for the sake of simplicity, we considered the dry unit weight.

Considering a hammer having a mass of $m = 200\,[kg]$ dropped from a height of $h = 1\,[m]$ and a load duration of $t_1 = 0.012\,[s]$, the corresponding peak pressure at the pile head was found to be $p_{max} \approx 1000\,[kN/m^2]$, see Equation 7.32. The time between successive blows was $t_2 = 0.25\,[s]$, corresponding to a driving frequency of $4\,[Hz]$.

Roller boundaries were prescribed along the planes and lines of symmetry and the pile tip point was free from constraint. To simulate the infinite soil domain, Kelvin-Voigt elements with $\delta = 1\,[m]$ were prescribed along the bottom and the vertical boundaries of the mesh. The wave speeds that were necessary to compute the spring and the dashpot coefficients of the Kelvin-Voigt element were varied with depth.

The initial stresses due to the weight of the soil and the pile were calculated in two stages. In the first stage, the stresses due to the soil weight were calculated using K_0-procedure with $K_0 = 0.5$. The void ratio decreased with depth according to Bauer's formula. In the second stage, the pile weight was applied and the computations were carried out until quasi-static equilibrium was reached, according to the convergence criteria of Section 4.9.

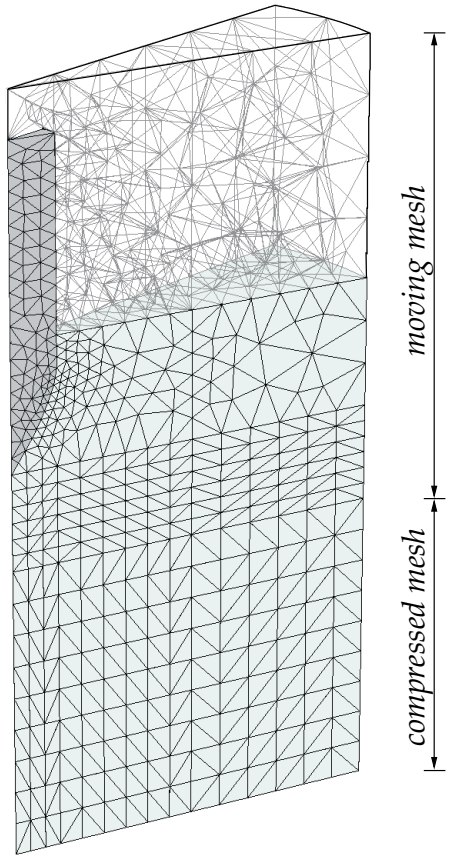


Figure 7.12: The mesh of the shallow driving problem with the moving and compressed mesh zones

7.4.2 Results for varying pile damping

As the hypoplastic model accounts for the material damping, no other type of damping was required for the sand. According to the work of Mabsout and Tassoulas [115], it is appropriate to use structural damping of the viscous type for the elastic pile. The frequency dependent Rayleigh damping was adopted in reference [115] with a damping ratio of $\zeta_d = 3\%$ estimated at the natural frequency of the pile. On the other hand, the use of the frequency independent local damping described in Section 4.10 is recommended instead of Rayleigh damping. As validated in Section 4.10, the local damping factor α is related to the damping ratio ζ_d as $\alpha = \pi\zeta_d$.

Computations were carried out with different values of the local damping factor α to study the effect of damping on the drivability of the pile. Three values were selected, $\alpha = 0\%$, 6% and 9%, corresponding to $\zeta_d = 0\%$, 2% and 2.8%, respectively. For all computations with varying damping factor, the coefficient of friction between the sand and the pile was fixed to $\mu = 0.5$ and an initial relative density of $I_d = 63\%$ corresponding to $e_0 = 0.645$ was used for the sand.

Figure 7.13 compares the penetration depth with time for the first blow of the three values of α. One observes that damping effects the penetration significantly. The final penetration depth for the first blow was found to be $8.3\,[cm]$, $6\,[cm]$ and $5.12\,[cm]$, for $\alpha = 0\%$, 6% and 9%, respectively. The time required to reach the final penetration was also effected by the damping. For the case of $\alpha = 9\%$, the final penetration was reached rapidly ($0.0875\,[s]$), whereas for cases of $\alpha = 6\%$ and $\alpha = 0\%$, $0.11\,[s]$ and $0.18\,[s]$, respectively, were required to reach the final penetration depth. Figure 7.14 shows the number

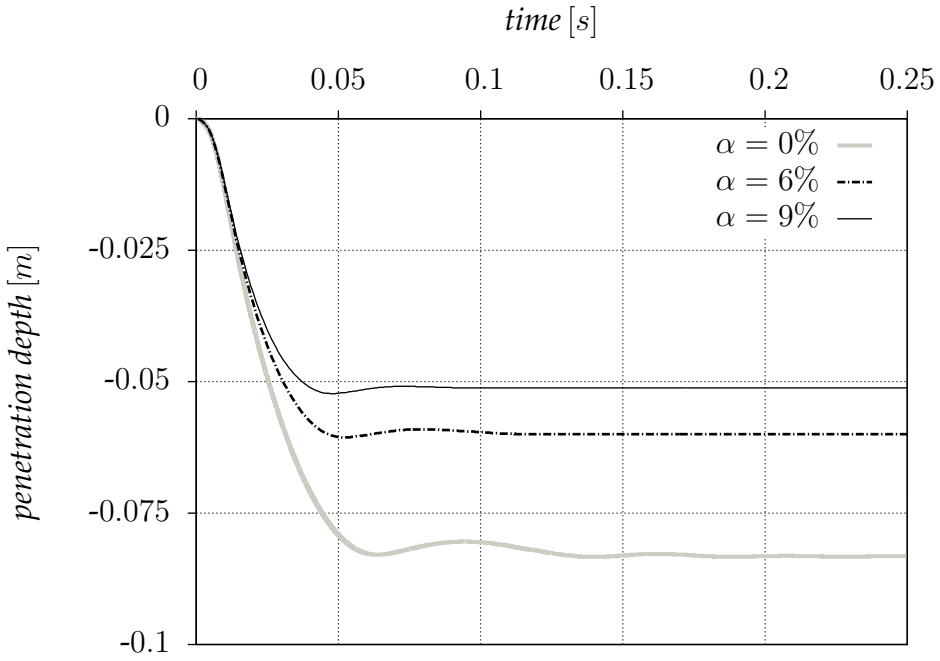


Figure 7.13: Penetration depth with time for the first blow with $\mu = 0.5$ and $I_d = 63\%$: comparison of different damping factors

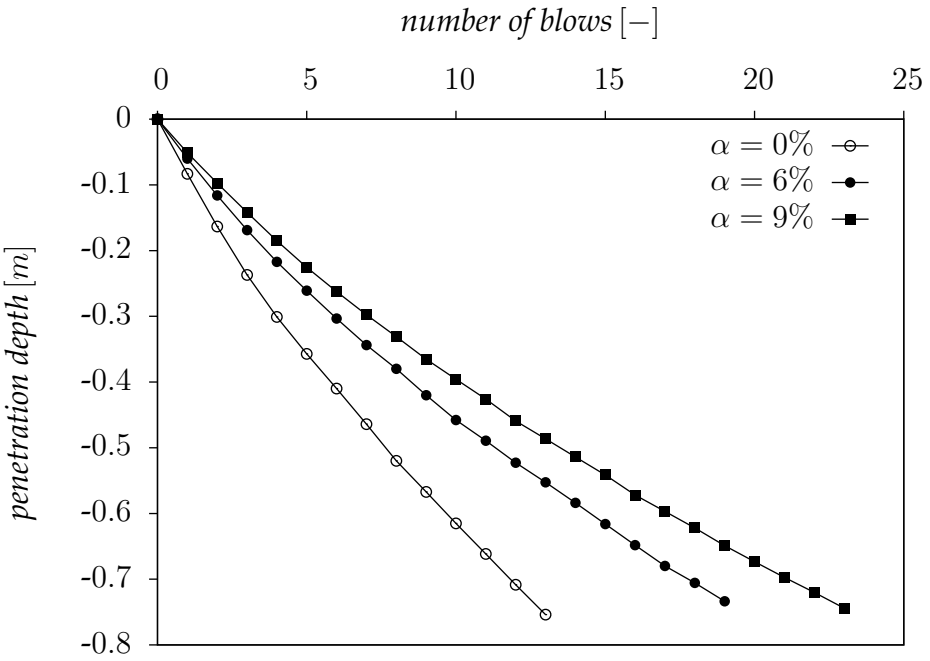


Figure 7.14: Number of blows versus penetration depth for the case of $\mu = 0.5$ and $I_d = 63\%$: comparison of different damping factors

of blows versus penetration depth for the considered cases. These results confirm the importance of good estimation of the damping factor for predicting realistic drivability of a pile. Figure 7.15 compares the finite element mesh of the problem in the initial and final configuration after a penetration of $0.75\,[m]$. One observes that the compressed mesh zone (z_2) compressed by the same displacement as the pile, whereas the moving mesh zone moved downwards with this displacement without any deformation of its elements.

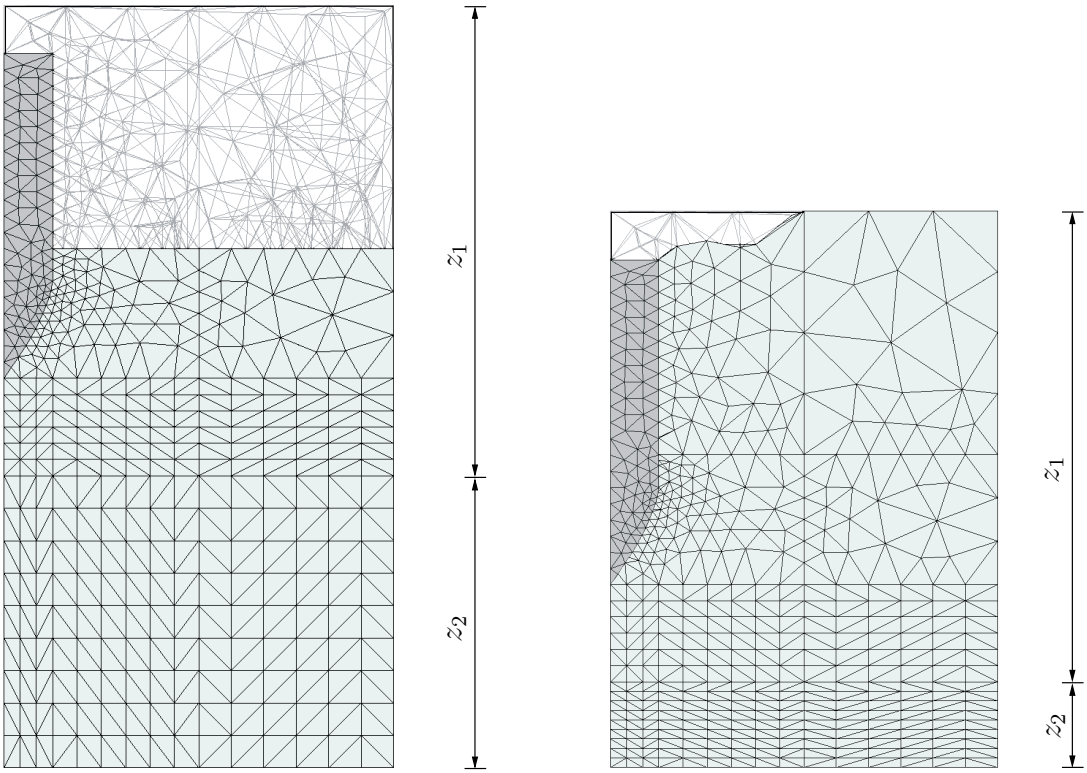


Figure 7.15: The finite element mesh of the shallow penetration problem with the moving mesh zone z_1 and the compressed mesh zone z_2: (a) initial configuration and (b) after a penetration of $0.75\,[m]$

7.4.3 Results for varying skin friction

Another important parameter in the study of pile driving is the friction between the soil and the pile, also called the skin friction. The skin friction influences the forces that are resisting the pile penetration and hence the pile drivability. For the study of the friction effect, we selected three values of the friction coefficient, $\mu = 0$ and 1, corresponding to the fully smooth and fully rough contact, respectively, which are considered to be theoretical cases. The third value was $\mu = 0.5$, which is realistic for concrete piles. All computations for studying the effect of μ were carried out with a damping factor of $\alpha = 6\%$ in the pile, and an initial relative density of $I_d = 63\%$ for the sand.

Figure 7.16 shows the results of penetration with time for the first blow for different values of μ. For the case of fully smooth contact ($\mu = 0$), the final penetration was found to be $12.3\,[cm]$, whereas in the case of fully rough contact, the pile penetrated $4.5\,[cm]$, indicating high skin resistance. The penetration depth for the case of $\mu = 0.5$ was found to be $6\,[cm]$. The results of the number of blows versus the penetration depth are shown in Figure 7.17, indicating considerable differences in the drivability of the pile with different skin friction. Figure 7.18 shows the particles distribution in the initial and

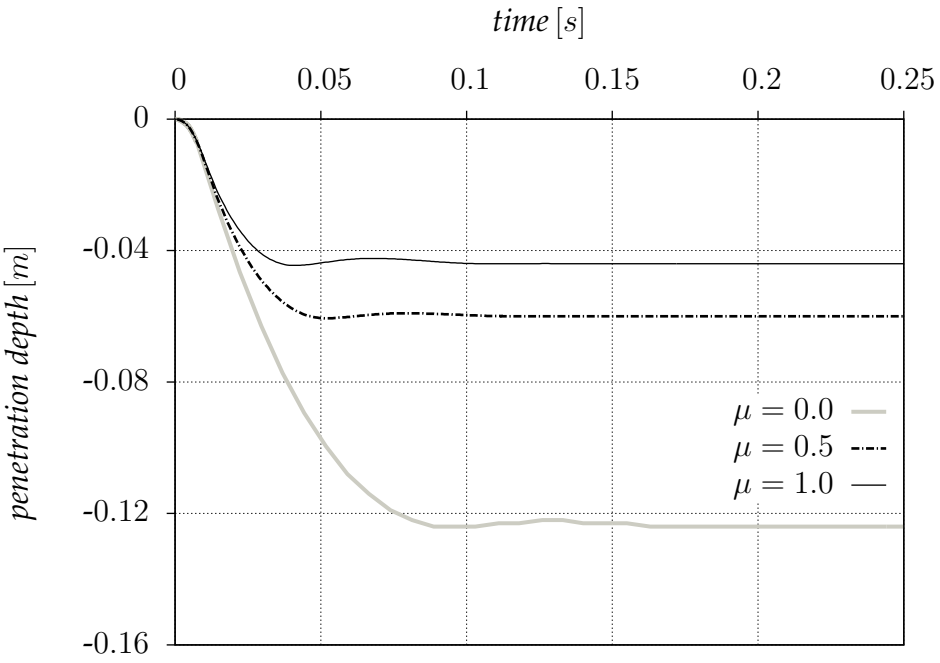


Figure 7.16: Penetration depth with time for the first blow with $\alpha = 6\%$ and $I_d = 63\%$: comparison of different values of the friction coefficient

the final configuration for the case of smooth contact ($\mu = 0$).

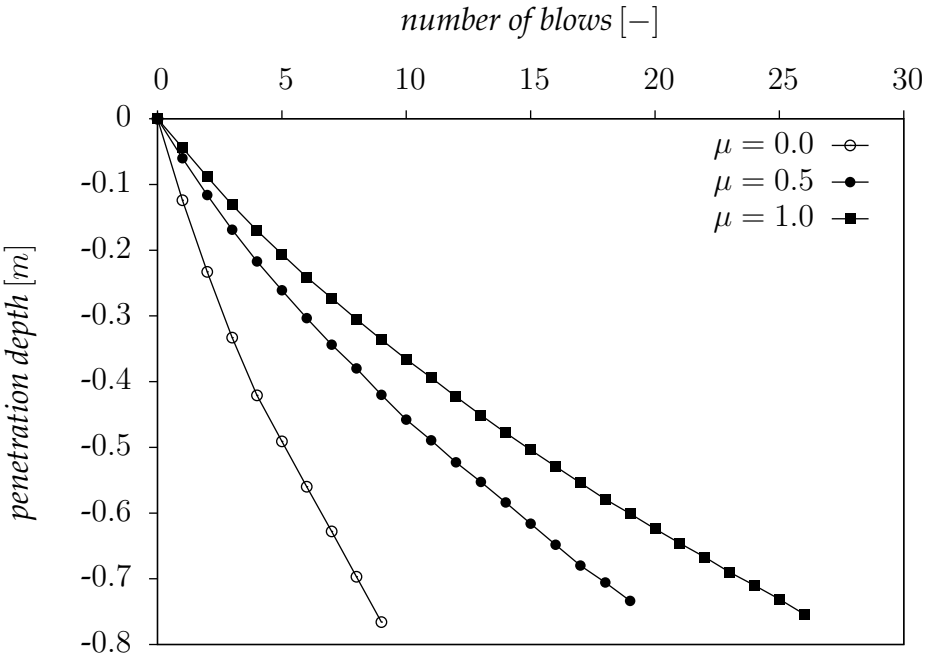
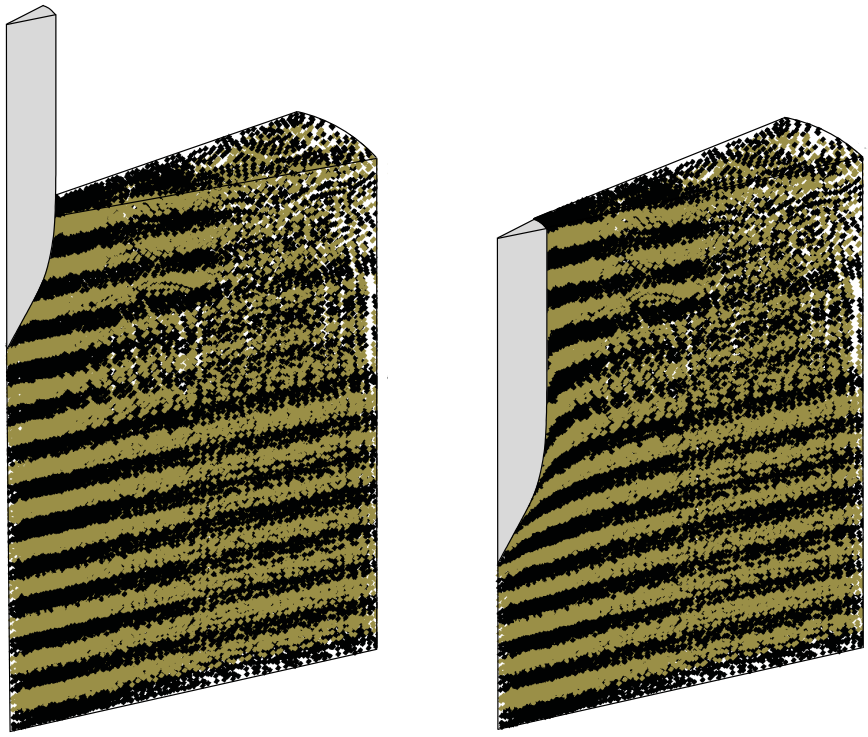


Figure 7.17: Number of blows versus penetration depth for the case of $\alpha = 6\%$ and $I_d = 63\%$: comparison of different values of the friction coefficient



(a) *(b)*

Figure 7.18: Particles distribution (a) initial configuration (b) final configuration for the case of $\alpha = 6\%$, $I_d = 63\%$ and $\mu = 0$

7.4.4 Results for varying sand density

Computations with different values of the relative density were carried out, in which $I_d = 30\%$, 63% and 80%, corresponding to loose, dense and very dense sand, respectively [179], were selected. The respective initial void ratios were $e_0 = 0.83$, 0.645 and 0.55. The computations were performed with $\mu = 0.5$ and $\alpha = 6\%$.

The results of the penetration with time are shown in Figure 7.19. The final penetration depth for the first blow was found to be $12.8\,[cm]$, $6\,[cm]$ and $3.36\,[cm]$ for the case of $I_d = 30\%$, 63% and 80%, respectively. It is observed that the driving in the loose sand ($I_d = 30\%$) showed practically no rebound of the pile, whereas the driving in the dense sand ($I_d = 63\%$) and very dense sand ($I_d = 80\%$) showed a rebound of 0.6 and $1.6\,[mm]$, respectively. The rebound of the pile in a blow was calculated as the difference between the maximum and the permanent penetration for a blow.

Figure 7.20 compares the drivability of the pile in sand for the three relative densities, indicating a slow rate of penetration for the very dense sand ($I_d = 80\%$). Driving the pile to a penetration depth of approximately $75\,[cm]$ in loose, dense and very dense sand required 9, 19 and 34 hammer blows, respectively, indicating the need for heavier hammer for deep penetration in the very dense sand.

Figure 7.21 shows the distribution of the void ratio along the depth for the case of very dense sand ($I_d = 80\%$) at different offsets relative to the pile after $27.5\,[cm]$ of driving. Also shown, is the initial distribution of the void ratio after applying gravity alone. The effect of the compaction (decrease of the void ratio) is observed in the figure. The compaction influenced the zone within a radius of $0.85\,[m]$ from the pile for the very dense

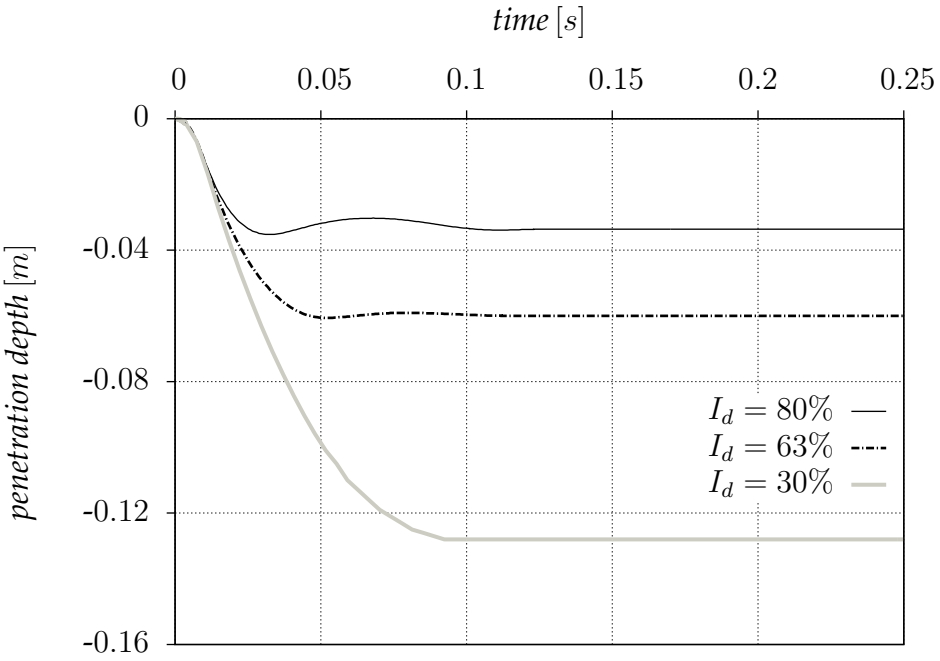


Figure 7.19: Penetration depth with time for the first blow with $\alpha = 6\%$ and $\mu = 0.5$: comparison of different relative densities

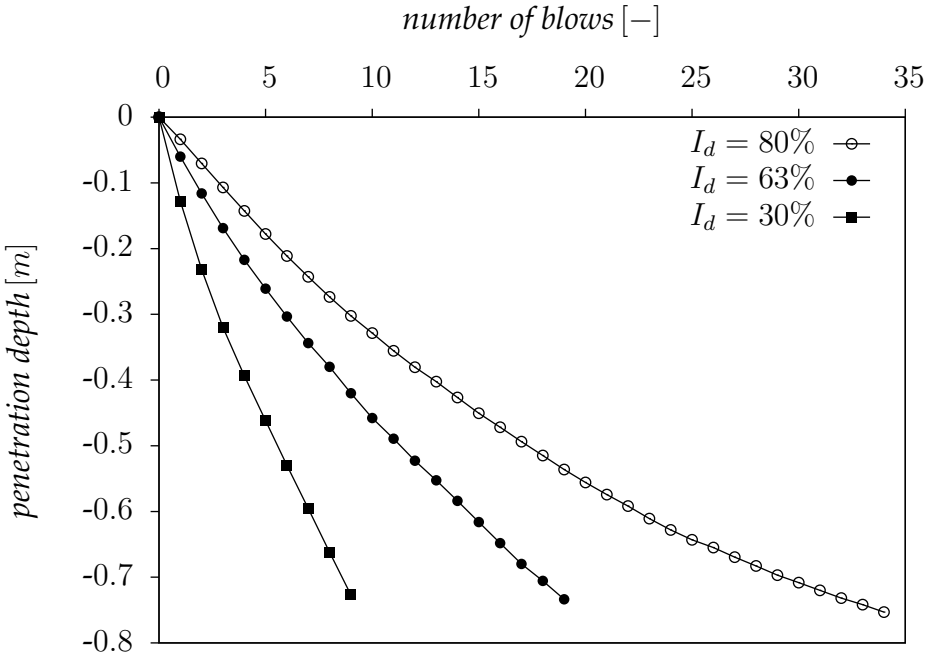


Figure 7.20: Number of blows versus penetration depth for the case of $\alpha = 6\%$ and $\mu = 0.5$: comparison of different relative densities

sand. Although not shown, compaction was observed for the dense sand ($I_d = 63\%$) and the loose sand ($I_d = 30\%$), with the zones of influence being limited to $0.7\,[m]$ and $0.65\,[m]$, respectively. Figure 7.22 shows the contours of the horizontal displacement after the final penetration was reached for the case of the very dense sand ($I_d = 80\%$). The contour scale ranges from 0 to $0.15\,[m]$, with the extremes corresponding to blue and red, respectively. The accumulated shear strains around the pile are shown in Figure 7.23 for the cases of very dense sand ($I_d = 80\%$). Considerable shearing of the sand was developed because of the high friction coefficient ($\mu = 0.5$).

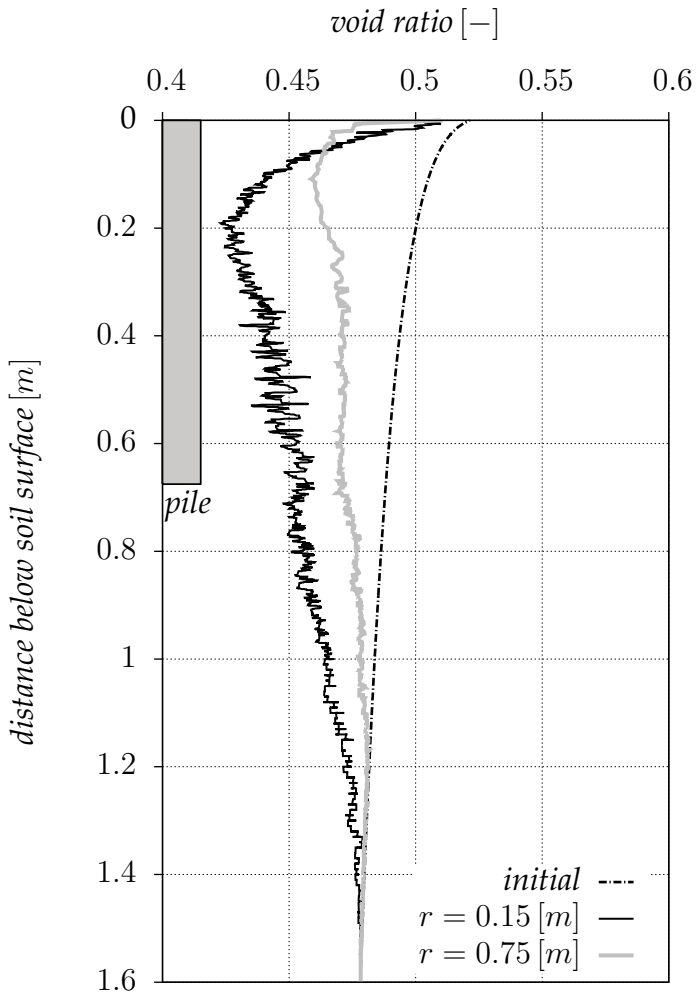


Figure 7.21: Void ratio distribution along depth for various offsets (r) form the pile for the case of very dense sand ($I_d = 80\%$) after a penetration of $27.5\,[cm]$

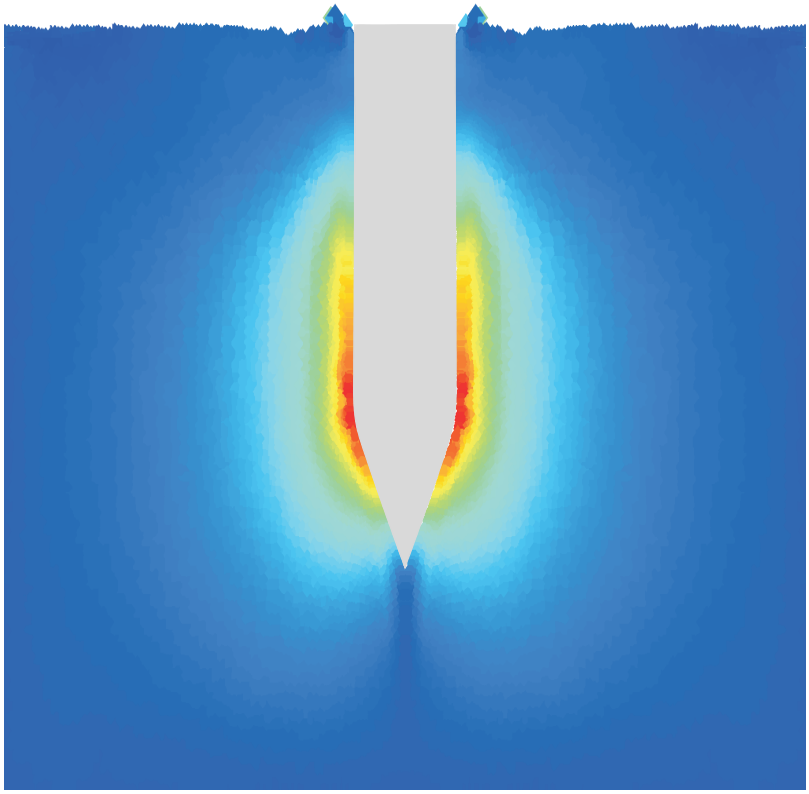


Figure 7.22: Horizontal displacement of sand in the final configuration for the case of $\alpha = 6\%$, $\mu = 0.5$ and $I_d = 80\%$, the scale ranges from 0 to 0.15 $[m]$

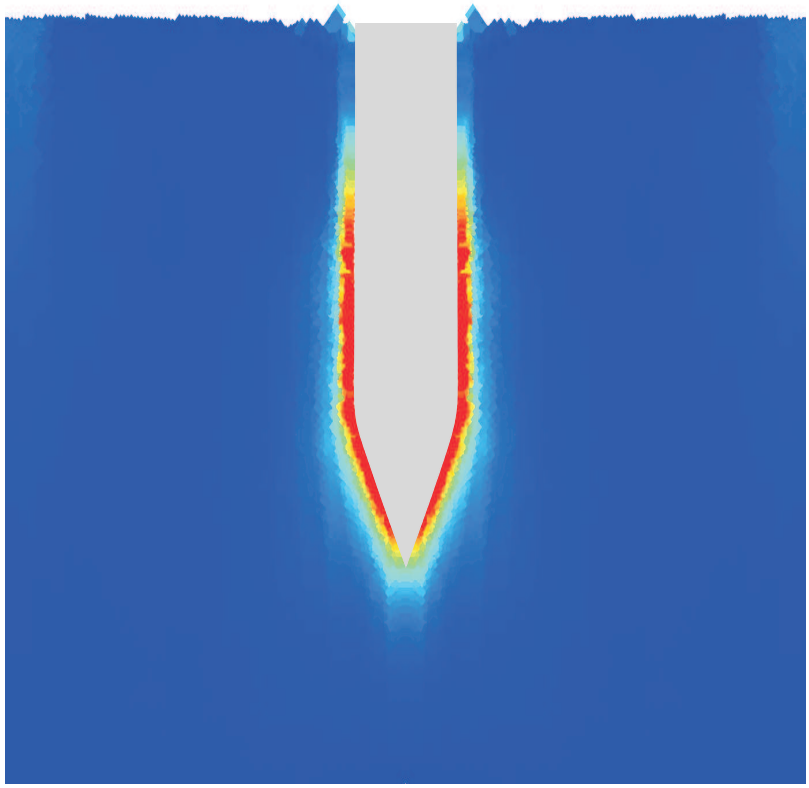


Figure 7.23: Accumulated shear strains in the final configuration for the case of $\alpha = 6\%$, $\mu = 0.5$ and $I_d = 80\%$

7.5 Deep penetration

7.5.1 Problem description

A pile driving problem for deep penetration is analyzed in this section. Figure 7.24 shows the finite element mesh of the problem along with the zones of the moving and compressed mesh. Similar to the problem of Section 7.4, only a sector of 20° was discretized. The soil domain had a height of $4.55\,[m]$ and a width of $1.2\,[m]$. The domain from the bottom of the mesh to a height of $4.0\,[m]$ represented the compressed mesh zone, whereas the moving mesh zone was represented by the upper domain including the initially empty mesh.

The mesh had a total number of elements of 6051 including the initially inactive elements with the total number of particles being 74856, distributed as 4 particles per each element for the pile, with each element in the compressed mesh zone having initially 20 particles, whereas each active element in the moving mesh zone had initially 10 particles. The aspect ratio (vertical to radial) of the compressed mesh zone was initially 3 to 1 up to a radial dimension of $0.15\,[m]$.

Similar to the previous example, the pile was initially embedded in the soil to a height of $0.4\,[m]$ from its tip point. The same procedure for calculating the initial stresses due to the weight and the same material properties and boundary conditions of the previous example were adopted in this example. All computations were performed with an initial relative density of $I_d = 63\%$.

7.5.2 Results for varying driving pressure

Two hammers with different masses were considered in this problem, one having a mass of $m = 200\,[kg]$ and the other a mass of $m = 300\,[kg]$, dropped from a height of $h = 1\,[m]$ with a load duration of $t_1 = 0.012\,[s]$. The respective peak pressures were found to be $p_{max} \approx 1000$ and $1500\,[kN/m^2]$. The time between successive blows was $t_2 = 0.5\,[s]$, corresponding to a driving frequency of $2\,[Hz]$.

Computations were carried out with $\alpha = 0\%$ and $\mu = 0.5$. The results of the number of blows versus the penetration depth are shown in Figure 7.25. A penetration depth of about $0.92\,[m]$ required 10 blows with a driving pressure of $p_{max} = 1000\,[kN/m^2]$ and 7 blows with a pressure of $p_{max} = 1500\,[kN/m^2]$. The computations with $p_{max} = 1000\,[kN/m^2]$ were stopped after the 10^{th} blow as it was realized that heavier hammer was required.

With the available computational resources, we computed 70 blows for the case of $\alpha = 6\%$ and $\mu = 0.5$ with the heavy hammer ($p_{max} = 1500\,[kN/m^2]$). The results are shown in Figure 7.26. The final penetration after 70 blow was found to be $2.1\,[m]$. The distribution of void ratios after a penetration of $0.5\,[m]$ is shown in Figure 7.28. The contour scale ranges from 0.36 to 0.6, with the extremes corresponding to blue and red, respectively. The value of 0.6 represents the initial void ratio of the red region approximately, indicating practically no compaction in this region, whereas one observes compaction around the pile.

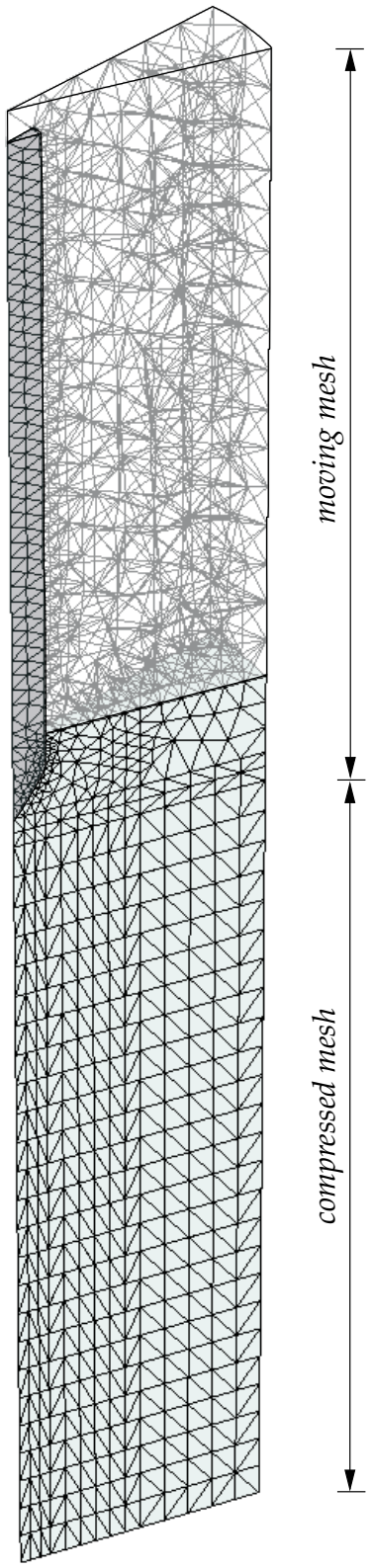


Figure 7.24: The mesh of the deep driving problem with the moving and compressed zones

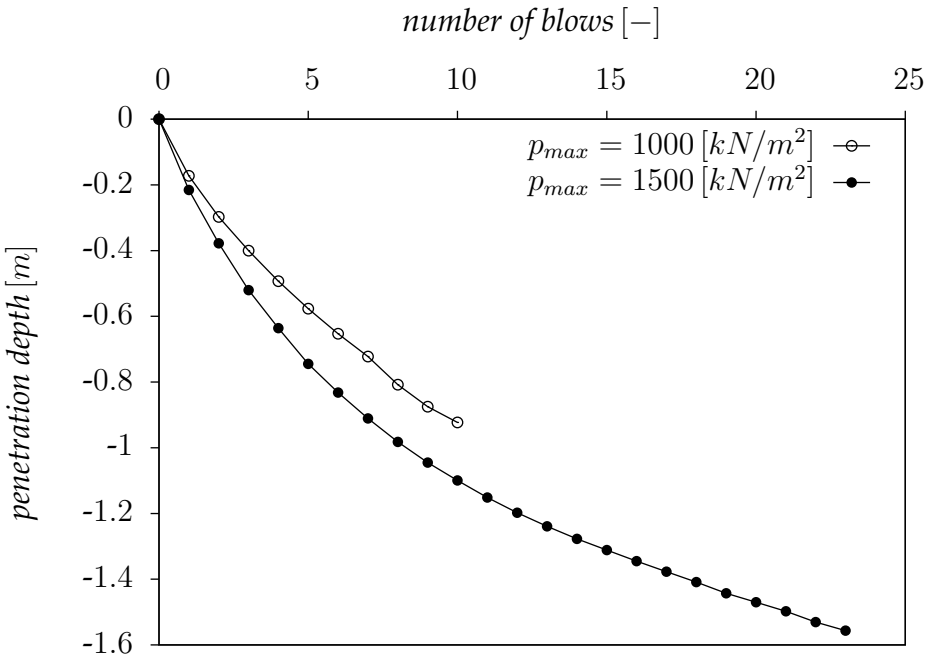


Figure 7.25: Number of blows versus penetration depth for the case of $\alpha = 0\%$, $\mu = 0.5$ and $I_d = 63\%$: comparison of different drving pressure

Nevertheless, a penetration depth of about $3\,[m]$ was reached with the case of $\alpha = 0\%$ and $\mu = 0$ using a driving pressure of $p_{max} = 1000\,[kN/m^2]$. The results are shown in Figure 7.27. High rate of penetration is observed for this idealized case. The $3\,[m]$ penetration required 11 blows. Figure 7.29 shows the particles distribution in the initial and the final configuration of this case.

Referring to the dynamic anchor problem of Section 5.3, much deeper penetration could be achieved ($5.5\,[m]$) as the problem required much less computational time, considering that the load was monotonic and the soil was modeled using Mohr-Coulomb.

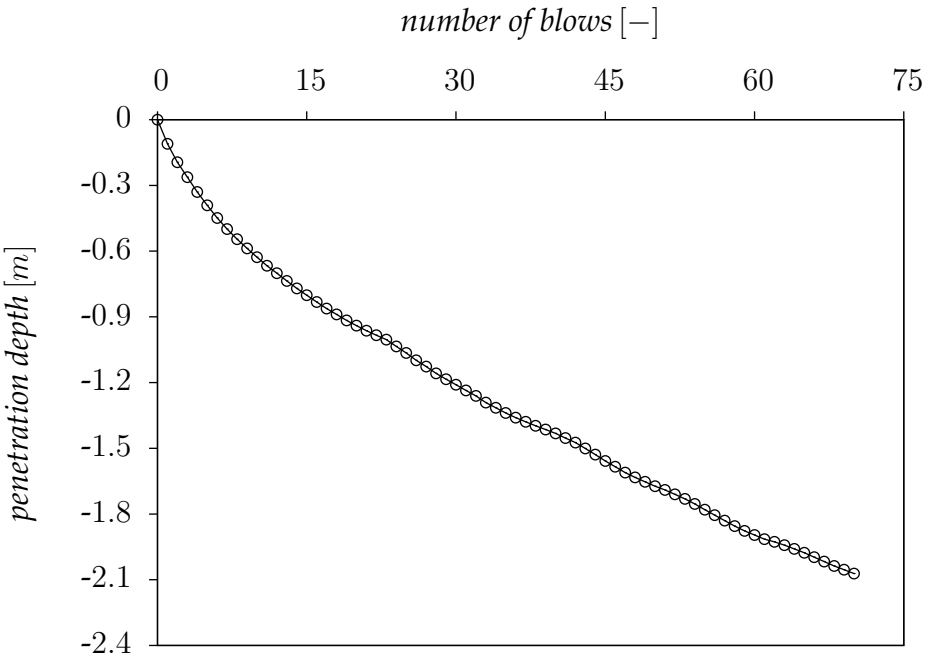


Figure 7.26: Number of blows versus penetration depth for the case of $\alpha = 6\%$, $\mu = 0.5$ and $I_d = 63\%$ with a driving pressure of $p_{max} = 1500\,[kN/m^2]$

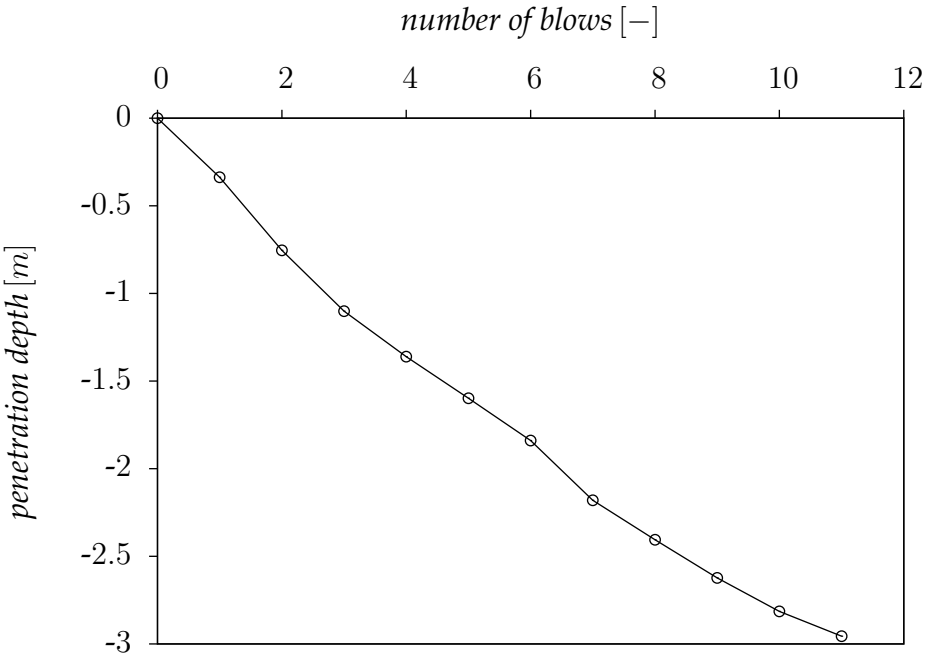


Figure 7.27: Number of blows versus penetration depth for the case of $\alpha = 0\%$, $\mu = 0$ and $I_d = 63\%$ with a driving pressure of $p_{max} = 1000\,[kN/m^2]$

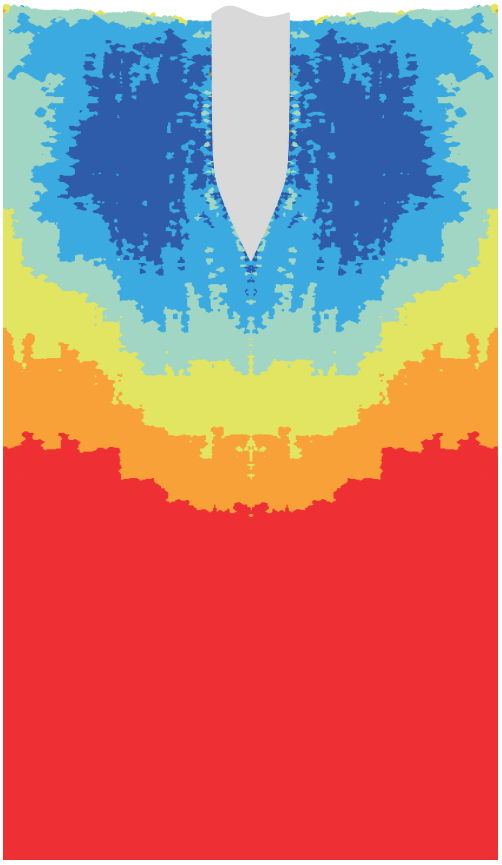
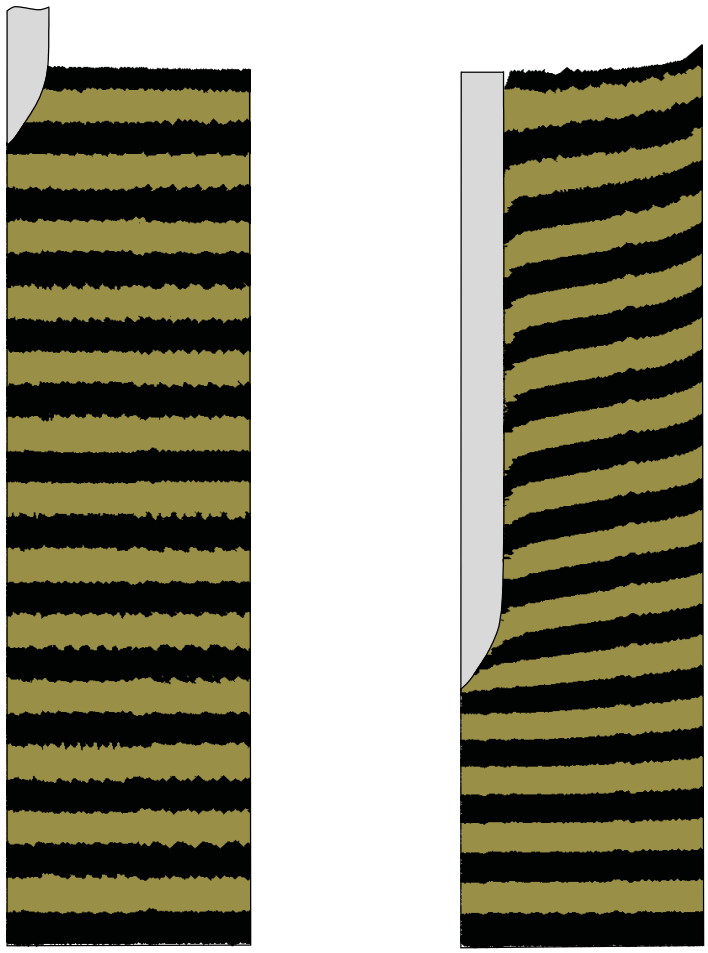


Figure 7.28: Void ratios for the case of $\alpha = 6\%$, $\mu = 0.5$, $I_d = 63\%$ and $p_{max} = 1500\,[kN/m^2]$ after a penetration of $0.5\,[m]$, the scale ranges from 0.36 (blue) to 0.6 (red)



(a) *(b)*

Figure 7.29: Particles distribution for the case of $\alpha = 0\%$, $\mu = 0$, $I_d = 63\%$ and $p_{max} = 1000\,[kN/m^2]$: (a) initial configuration (b) after a penetration of $3\,[m]$

Chapter 8

Concluding remarks

In the present study, the attention focused on the extension of the material point method to geomechanical applications. Many fundamental issues were discussed and treated. This chapter draws some conclusions from this study.

An extensive literature review was presented about the development of the material point method (MPM), starting from its infancy at Los Alamos where it was originally developed by Harlow in 1950. From this literature review, one concludes that MPM was successfully applied to solve a wide range of problems in different engineering fields. The method was shown to be a powerfully numerical technique, providing possibilities of modeling large deformation and interaction between solid bodies.

One-step integration schemes for a second-order ordinary differential equation (ODE) were studied and extended for application to the finite element governing equations, aiming to circumvent the time step size restriction associated with the conditionally-stable integration schemes. However, it was not possible to increase the size of the time step beyond the critical one. This is consistent with the experience of others [46, 177, 193].

A solution to the well-known numerical dilemma of grid-crossing in MPM was proposed. Gauss integration was adopted for quasi-static problems in a way that is identical to that of Lagrangian FEM, in which the integration points have fixed local positions inside the element and never cross element boundary. Furthermore, smoothening of the stresses variations was accomplished by using a stress averaging technique for those elements that are fully filled. It was shown that the problem of the spacial oscillations in stresses is mitigated by this procedure.

It is clear that MPM suffers from the same problem of pathological locking, which is encountered when using low-order finite element method for incompressible problems, if mitigation procedures are not introduced. The enhanced volumetric strain procedure, which extends the volumetric behavior to include neighboring elements via a least squares smoothening process, helps mitigating the problem of mesh locking. It was shown that, the problem of overpredicting the stiffness because of the incompressibility constraints induced by the low-order elements, could be treated by the enhanced volumetric strain procedure.

In the analysis of soil dynamics, absorbing boundaries are introduced to prevent the reflection of waves at the arbitrary selected boundary of the domain. One usually em-

ploys so-called dashpots that will continuously creep under load. In order to limit such non-physical displacements, the technique of the boundary dashpots was modified to a Kelvin-Voigt type of response by combining the dashpot with a spring.

The material point method was extended to model the behavior of saturated soils under dynamic loading. A velocity formulation was adopted, in which solid and fluid phase velocities are the variables. Key components of the proposed approach are the adoption of Verruijt's [183] sequence of update steps when integrating over time, and the enhancement of volumetric strains that mitigates the spurious pressure oscillations which plague low-order finite element implementations. It was shown that the adopted algorithm can predict dynamic response of saturated soil for various scenarios and can accurately capture the propagation of the undrained wave followed by the damped wave that is associated with the consolidation process. The algorithm was applied to the problem of a sea dike being subjected to cyclic waves. Predictions of the dynamic generation and dissipation of pore pressures were discussed as well as the effect of the revetment and water stiffness on soil stresses and permanent deformations.

Numerical issues for quasi-static problems including single and two-phase were discussed, including the use of local damping and the prediction of the convergence to quasi-static equilibrium. It was shown that the local damping cannot replace the absorbing boundaries for dynamic problems. The interfaces between layers of different damping act as artificial constraints, leading to early reflection of the waves before reaching the boundaries of the domain. Mass scaling procedure was also presented and tested. It was shown that this procedure allows the use of large time steps when used to model problems with slow rate of loading, in which the inertia effect can be disregarded.

MPM was applied to analyze different geomechanical problems. The collapse of a tunnel face was analyzed and it was shown that the method can predict the dynamic collapsing of the ground with a crater being formed on the ground surface. The pattern of the tunnel face collapse was also investigated with the real collapse time. The instability of a slope and its quasi-static deformation was investigated. The installation of a dynamic anchor was simulated. Different values of the frictional coefficient were used at the contact surface between the soil and the anchor. Deep penetration of about $5.5\,[m]$ was reached, showing the capability of MPM in modeling deep penetration problems.

Considering cyclic loading in pile driving, a highly non-linear hypoplastic model was used for sand. In contrast to the previous implementation of this constitutive model, explicit Euler forward integration scheme with sub-stepping technique was adopted. A formula to calculate the constrained modulus was derived from the constitutive equation of the model. The constrained modulus was required in the calculation of the critical time step size and the dashpots and springs coefficients of the Kelvin-Voigt elements. Numerical simulations of oedometer and triaxial test were performed. Comparing the values of predicted peak friction angles to the experimental database with the same stress level, we observed that the hypoplastic model predicts realistic peak friction an-

gles for low relative densities, but overpredicts the peak friction angles for high relative densities. However, experimental data corresponding to lower stress level shows high peak friction angles that are in-line with the values of the numerical predictions. Hence, high stress levels in pile driving make the hypoplastic model less attractive in modeling pile driving in dense sand as the model does not include stress-level dependency of the peak friction angle. Considering stiffness data for different quartz sands, we observed that the calibration of the input data needs special attention when using the hypoplastic model. For further numerical simulations of pile driving, it is recommended to use data from cyclic test results, e.g., data from triaxial test, simple shear and hollow cylinder test. Such experimental data may be required to assess the performance of the considered model in cyclic loading.

Numerical simulation of pile driving using MPM was investigated. A procedure was proposed to convert the hammer impact into a surface traction applied at the pile head. Results of pile drivability in sand with different initial densities were presented. Also investigated, was a parametric study including the variation of the pile damping, skin friction and the driving pressure. Computations with shallow and deep penetration up to $3\,[m]$ were performed. It was shown that damping of the pile and the friction at the contact surface between the pile and the soil effect the penetration rate considerably. For further simulations, estimation of the damping factor and the frictional coefficients is recommended.

The moving mesh concept was used in the modeling of the drop anchor and the pile driving. The concept allows the fine mesh to move with the pile, avoiding elements that contain both pile and soil particles. Adopting this procedure, the pile is modeled in a way similar to that of a Lagrangian finite element, in which the computational mesh aligns with the surface where tractions are prescribed and hence the load is applied at the nodes of the pile head and need not be mapped to the particles. Furthermore, a Lagrangian pile eliminates the need of identifying the new interface between the pile and the soil during computations as the nodes of the interface surface coincide with the material points of the pile throughout the computations. Hence, the unit normal vectors which are required in the contact algorithm do not change and the inaccuracy of recomputing them is eliminated.

For further application of MPM, it is recommended that simulations be completed for piles driven into saturated soil. This would include the prediction of the generation and dissipation of pore pressures. Another application of the algorithm would be the simulation of sand liquefaction. Very loose sand has a tendency to collapse under shear with a decrease of the volume, which may be especially dangerous when the sand is saturated with water. A saturated sand that would densify in a short time under shear could develop large pore pressures leading to shear failure. This can have serious consequences for the stability of slopes. For example, the sand in the estuaries in the south west of the Netherlands is loosely packed because of the ever continuing process of erosion by tidal currents and deposition of the sand at the turning of the tide.

Bibliography

[1] S. Andersen and L. Andersen. Analysis of spatial interpolation in the material-point method. *Computers and Structures*, 88(7-8):506–518, 2010.

[2] S. Andersen and L. Andersen. Modelling of landslides with the material-point method. *Computational Geosciences*, 14(1):137–147, 2010.

[3] S.M. Andersen. *Material-Point Analysis of Large-Strain Problems: Modelling of Landslides*. PhD thesis, Department of Civil Engineering, Aalborg University, Aalborg, Denmark, 2009.

[4] J.H. Argyris and S. Kelsey. *Energy Theorems and Structural Analysis*. Butterworths, London,1960. Reprinted from a series of articles in Aircraft Engineering, 1954–1955.

[5] B. Banerjee. Material point method simulations of fragmenting cylinders. In *17th ASCE Engineering Mechanics Conference*, University of Delaware, Newark, DE, June 2004.

[6] S.G. Bardenhagen. Energy conservation error in the material point method for solid mechanics. *Journal of Computational Physics*, 180(1):383–403, 2002.

[7] S.G. Bardenhagen and J.U. Brackbill. Dynamic stress bridging in granular material. *Journal of Applied Physics*, 83(11):5732–5740, 1998.

[8] S.G. Bardenhagen and E.M. Kober. The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–496, 2004.

[9] S.G. Bardenhagen, J.U. Brackbill, and D. Sulsky. The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering*, 187 (3–4):529–541, 2000.

[10] S.G. Bardenhagen, J.U. Brackbill, and D. Sulsky. Numerical study of stress distribution in sheared granular material in two dimensions. *Physical Review E*, 62(3): 3882–3890, 2000.

[11] S.G. Bardenhagen, J.U. Brackbill, and D.L. Sulsky. Shear deformation in granular materials. In *Proceedings of the 11th International Detonation Symposium*, pages 547–555. VA. Office of Naval Research, 2000.

[12] S.G. Bardenhagen, J.E. Guilkey, K.M. Roessig, J.U. Brackbill, W.M. Witzel, and J.C. Foster. An improved contact algorithm for the material point method and application to stress propagation in granular material. *CMES: Computer Modeling in Engineering and Sciences*, 2(4):509–522, 2001.

[13] S.G. Bardenhagen, A.D. Brydon, and J.E. Guilkey. Insight into the physics of foam densification via numerical simulation. *Journal of the Mechanics and Physics of Solids*, 53(3):597–617, 2005.

[14] K.-J. Bathe and S. Bolourchi. Large displacement analysis of three-dimensional beam structures. *International Journal for Numerical Methods in Engineering*, 14(7): 961–986, 1979.

[15] K.-J. Bathe, E. Ramm, and E.L. Wilson. Finite element formulations for large deformation dynamic analysis. *International Journal for Numerical Methods in Engineering*, 9(2):353–386, 1975.

[16] K.J. Bathe and E.L. Wilson. *Numerical Methods in Finite Element Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[17] E. Bauer. Calibration of a comprehensive hypoplastic model for granular materials. *Soils and Foundations*, 36(1):13–26, 1996.

[18] T. Belytschko and B.J. Hsieh. Non-linear transient finite element analysis with convected co-ordinates. *International Journal for Numerical Methods in Engineering*, 7(3):255–271, 1973.

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

[20] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47, 1996.

[21] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

[22] P. Bettess. Infinite elements. *International Journal for Numerical Methods in Engineering*, 11(1):53–64, 1977.

[23] P. Bettess and O.C. Zienkiewicz. Diffraction and refraction of surface waves using finite and infinite elements. *International Journal for Numerical Methods in Engineering*, 11(8):1271–1290, 1977.

[24] L. Beuth. *Formulation and Application of a Quasi-Static Material Point Method*. PhD thesis, Institute of Geotechnical Engineering, University of Stuttgart, 2012.

[25] L. Beuth, T. Benz, P.A. Vermeer, C.J. Coetzee, P. Bonnier, and P. Van Den Berg. Formulation and validation of a quasi-static material point method. In *Proceedings of the 10th International Symposium on Numerical Methods in Geomechanics*, volume 10, pages 189–195., Rhodes, Greece, 2007.

[26] L. Beuth, T. Benz, P.A. Vermeer, and Z. Więckowski. Large deformation analysis using a quasi-static material point method. *Journal of Theoretical and Applied Mechanics, Bulgarian Academy of Sciences Sofia*, 38:45–60, 2008.

[27] L. Beuth, Z. Więckowski, and P.A. Vermeer. Solution of quasi-static large-strain problems by the material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(13):1451–1465, 2011.

[28] C.B. Biezeno and R. Grammel. *Technische Dynamik*. Springer–Verlag, Berlin, 1933.

[29] M.A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12(2):155–164, 1941.

[30] M.D. Bolton. The strength and dilatancy of sands. *Géotechnique*, 36(1):65–78, 1986.

[31] J. Bonet, H. Marriott, and O. Hassan. Stability and comparison of different linear tetrahedral formulations for nearly incompressible explicit dynamic applications. *International Journal for Numerical Methods in Engineering*, 50(1):119–133, 2001.

[32] P.G. Bonnier. *Testing, Modelling and Numerical Analysis of the Mechanical Behaviour of Bituminous Concrete.* PhD thesis, Delft University of Technology, Delft, The Netherlands, 1993.

[33] R. I. Borja. Dynamics of pile driving by the finite element method. *Computers and Geotechnics*, 5(1):39–49, 1988.

[34] J.U. Brackbill. On modelling angular momentum and vorticity in compressible fluid flow. *Computer Physics Communications*, 47(1):1–16, 1987.

[35] J.U. Brackbill. FLIP MHD: A particle-in-cell method for magnetohydrodynamics. *Journal of Computational Physics*, 96(1):163–192, 1991.

[36] J.U. Brackbill and H.M. Ruppel. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics*, 65(2):314–343, 1986.

[37] J.U. Brackbill, D.B. Kothe, and H.M. Ruppel. FLIP: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1):25–38, 1988.

[38] F. Brunel, J.N. Leboeuf, T. Tajima, J.M. Dawson, M. Makino, and T. Kamimura. Magnetohydrodynamic particle code: Lax-Wendroff algorithm with finer grid interpolations. *Journal of Computational Physics*, 43(2):268–288, 1981.

[39] L. Brusa and L. Nigro. A one-step method for direct integration of structural dynamic equations. *International Journal for Numerical Methods in Engineering*, 15(5): 685–699, 1980.

[40] A.D. Brydon, S.G. Bardenhagen, E.A. Miller, and G.T. Seidler. Simulation of the densification of real open-celled foam microstructures. *Journal of the Mechanics and Physics of Solids*, 53(12):2638–2660, 2005.

[41] D. Burgess, D. Sulsky, and J.U. Brackbill. Mass matrix formulation of the FLIP particle-in-cell method. *Journal of Computational Physics*, 103(1):1–15, 1992.

[42] O. Buzzi, D.M. Pedroso, and A. Giacomini. Caveats on the implementation of the generalized material point method. *Computer Modelling in Engineering and Sciences*, 1(1):1–21, 2008.

[43] Z. Chen, W. Hu, L. Shen, X. Xin, and R. Brannon. An evaluation of the MPM for simulating dynamic failure with damage diffusion. *Engineering Fracture Mechanics*, 69(17):1873–1890, 2002.

[44] R.W. Clough. The finite element method in plane stress analysis. In *Proceedings, American Society of Civil Engineers ASCE, 2nd Conference on Electronic Computation*, pages 345–378, Pittsburgh, PA, 1960.

[45] C.J. Coetzee. *The Modelling of Granular Flow Using the Particle-in-Cell Method*. PhD thesis, Department of Mechanical Engineering, University of Stellenbosch, Stellenbosch, South Africa, 2003.

[46] C.J. Coetzee and I. Jassim. Oral communications. June, 2009.

[47] 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.

[48] C.J. Costantino. Finite element approach to stress wave problems. *Journal of the Engineering Mechanics Division ASCE*, 93:153–177, 1967.

[49] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49:1–23, 1943.

[50] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[51] A. Cromer. Stable solutions using the Euler approximation. *American Journal of Physics*, 49(5):455–459, 1981.

[52] S.J. Cummins and J.U. Brackbill. An implicit particle-in-cell method for granular materials. *Journal of Computational Physics*, 180(2):506–548, 2002.

[53] P.A. Cundall. Distinct element models of rock and soil structure. In *Analytical and Conlputational Methods in Engineering Rock Mechanics*, pages 129–163. Allen and Unwin, London, 1987.

[54] G.G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3 (1):27–43, 1963.

[55] N.P. Daphalapurkar, H. Lu, D. Coker, and R. Komanduri. Simulation of dynamic crack growth using the generalized interpolation material point (GIMP) method. *International Journal of Fracture*, 143(1):79–102, 2007.

[56] R. De Boer and W. Ehlers. A historical review of the formulation of porous media theories. *Acta Mechanica*, 74(1):1–8, 1988.

[57] A. De Looff, R. `t Hart, K. Montauban, and M. van de Ven. Golfklap, a model to determine the impact on dike structures with an asphaltic concrete layer. In *Proceedings of the 30th ICCE*, pages 5106–5115, San Diego, USA, 2006.

[58] S.A. Degago. Impact tests on sand with numerical modeling, emphasizing on the shape of the falling onbject. Master's thesis, Norwegian University of Science and Technology, Department of Civil and Transport Engineering, 2007.

[59] C. Detournay and E. Dzik. Nodal mixed discretization for tetrahedral elements. In *Proceedings of the 4th International FLAC Symposium, Numerical Modeling in Geomechanics. Minnesota Itasca Consulting Group, Inc. Paper*, 2006.

[60] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. *Encyclopedia of Computational Mechanics*, 1:413–437, 2004.

[61] J.W. Eastwood. Low dissipation fluid and MHD simulation. *Computer Physics Communications*, 48(1):17–23, 1988.

[62] T. Erhart, W.A. Wall, and E. Ramm. Robust adaptive remeshing strategy for large deformation, transient impact simulations. *International Journal for Numerical Methods in Engineering*, 65(13):2139–2166, 2006.

[63] M.W. Evans, F.H. Harlow, and B.D. Meixner. Interaction of shock or rarefaction with a bubble. *The Physics of Fluids*, 5(6):651–656, 1962.

[64] I.A. Fernandes. Exploring quadratic shape functions in material point method. Master's thesis, Department of Mechanical Engineering, University of Utah, 2004.

[65] B.A. Finlayson and L.E. Scriven. The method of weighted residuals–A review. *Applied Mechanics Reviews*, 19(9):735–748, 1966.

[66] FLAC. *Fast Lagrangian Analysis of Continua in 3 Dimensions*. Online Manual, 1998.

[67] G.M. Flato. A particle-in-cell sea-ice model. *Atmosphere-Ocean*, 31(3):339–358, 1993.

[68] R.A. Frazer, W.P. Jones, and S.W. Skan. Approximations to functions and to the solution of differential equations. Technical Report 1799, Great Britain Aeronautical Research Committee Report, 1937.

[69] B.G. Galerkin. Series solution of some problems of elastic equilibrium of rods and plates. *Vestnik Inzhenerov Tekhnikov*, 19:897–908, 1915.

[70] D. Givoli. Non-reflecting boundary conditions. *Journal of Computational Physics*, 94 (1):1–29, 1991.

[71] G. G. Goble and Frank Rausche. The analysis of pile driving–A state-of-the-art. In *International Seminar on the Applications of Stress-Wave Theory on Piles*, Stockholm, June 1980.

[72] J.E. Guilkey and J.A. Weiss. Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method. *International Journal for Numerical Methods in Engineering*, 57(9):1323–1338, 2003.

[73] J.E. Guilkey, T. Harman, A. Xia, B. Kashiwa, and P. McMurtry. An Eulerian-Lagrangian approach for large deformation fluid structure interaction problems, part 1: Algorithm development. In *2nd International Conference on Fluid Structure Interactions*, Cadiz, Spain, 2003.

[74] J.E. Guilkey, J.B. Hoying, and J.A. Weiss. Computational modeling of multicellular constructs with the material point method. *Journal of Biomechanics*, 39(11):2074–2086, 2006.

[75] J.E. Guilkey, T.B. Harman, and B. Banerjee. An Eulerian-Lagrangian approach for simulating explosions of energetic devices. *Computers and structures*, 85(11-14): 660–674, 2007.

[76] Y. Guo and J.A. Nairn. Calculation of J-integral and stress intensity factors using the material point method. *Computer Modeling in Engineering and Sciences*, 6:295–308, 2004.

[77] Y.J. Guo and J.A. Nairn. Three-dimensional dynamic fracture analysis using the material point method. *Computer Modeling in Engineering and Sciences*, 16(3):141–156, 2006.

[78] G.D. Hahn. A modified Euler method for dynamic analyses. *International Journal for Numerical Methods in Engineering*, 32(5):943–955, 1991.

[79] L.N. Hand and J.D. Finch. *Analytical Mechanics*. Cambridge University Press, Cambridge, 1998.

[80] F.H. Harlow. The particle-in-cell computing method for fluid dynamics. *Methods in Computational Physics*, 3:319–343, 1964.

[81] F.H. Harlow. PIC and its progency. *Computer Physics Communications*, 48:1–10, 1988.

[82] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of fluids*, 8(12):2182–2189, 1965.

[83] F.H. Harlow and J.E. Welch. Numerical study of large-amplitude free-surface motions. *The Physics of Fluids*, 9(5):842–851, 1966.

[84] F.H. Harlow, A.A. Amsden, and J.R. Nix. Relativistic fluid dynamics calculations with the particle-in-cell technique. *Journal of Computational Physics*, 20(2):119–129, 1976.

[85] T. Harman, J.E. Guilkey, B. Kashiwa, J. Schmidt, and P. McMurtry. An Eulerian-Lagrangian approach for large deformation fluid structure interaction problems, part 2: Multi-physics simulations within a modern computational framework. In *2nd International Conference on Fluid Structure Interactions*, Cadiz, Spain, 2003.

[86] R. Hart, P.A. Cundall, and J. Lemos. Formulation of a three-dimensional distinct element model–Part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 25(3):117–125, 1988.

[87] S. Henke. Influence of pile installation on adjacent structures. *International Journal for Numerical and Analytical Methods in Geomechanics*, 34(11):1191–1210, 2010.

[88] S. Henke and J. Grabe. Simulation of pile driving by 3-dimensional finite-element analysis. In *Proceedings of the 17th European Young Geotechnical Engineers*, pages 215–233, Zagreb, 2006.

[89] S. Henke, G. Qiu, and J. Grabe. A Coupled Eulerian-Lagrangian approach to solve geotechnical problems involving large deformations. In *7th European Conference on Numerical Methods in Geotechnical Engineering (NUMGE)*, Trondheim, Norway, 2010.

[90] R. Hill. *The Mathematical Theory of Plasticity*. Oxford, Clarendon Press, 1950.

[91] R. Hill. Some basic principles in the mechanics of solids without a natural time. *Journal of the Mechanics and Physics of Solids*, 7(3):209–225, 1959.

[92] W. Hu and Z. Chen. A multi-mesh MPM for simulating the meshing process of spur gears. *Computers and structures*, 81(20):1991–2002, 2003.

[93] W. Hu and Z. Chen. Model-based simulation of the synergistic effects of blast and fragmentation on a concrete wall using the MPM. *International Journal of Impact Engineering*, 32(12):2066–2096, 2006.

[94] M. Huang, Z.Q. Yue, L.G. Tham, and O.C. Zienkiewicz. On the stable finite element procedures for dynamic problems of saturated porous media. *International Journal for Numerical Methods in Engineering*, 61(9):1421–1450, 2004.

[95] P. Huang, X. Zhang, S. Ma, and X. Huang. Contact algorithms for the material point method in impact and penetration simulation. *International Journal for Numerical Methods in Engineering*, 85(4):498–517, 2011.

[96] Z.J. Huang and S.B. Savage. Particle-in-cell and finite difference approaches for the study of marginal ice zone problems. *Cold Regions Science and Technology*, 28(1): 1–28, 1998.

[97] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[98] 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.

[99] I. Ionescu, J. Guilkey, M. Berzins, R.M. Kirby, and J. Weiss. Computational simulation of penetrating trauma in biological soft tissues using the material point method. *Studies in Health Technology and Informatics*, 111:213–218, 2005.

[100] B. Jeremić, Z. Cheng, M. Taiebat, and Y. Dafalias. Numerical simulation of fully saturated porous materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(13):1635–1660, 2008.

[101] R.R. Kunar and L. Rodriguez-Ovejero. A model with non-reflecting boundaries for use in explicit soil-structure interaction analyses. *Earthquake Engineering and Structural Dynamics*, 8(4):361–374, 1980.

[102] J.N. Leboeuf, T. Tajima, and J.M. Dawson. A magnetohydrodynamic particle code for fluid simulation of plasmas. *Journal of Computational Physics*, 31(3):379–408, 1979.

[103] S.L. Lee, Y.K. Chow, G.P. Karunaratne, and K.Y. Wong. Rational wave equation model for pile-driving analysis. *Journal of Geotechnical Engineering*, 114(3):306–325, 1988.

[104] X. Li, X. Zhang, X. Han, and D.C. Sheng. An iterative pressure-stabilized fractional step algorithm in saturated soil dynamics. *International Journal for Numerical and Analytical Methods in Geomechanics*, 34(7):733–753, 2010.

[105] G. R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Publishing Company Incorporated, 2003.

[106] G.R. Liu and S.S. Quek Jerry. A non-reflecting boundary for analyzing wave propagation using the finite element method. *Finite Elements in Analysis and Design*, 39 (5-6):403–417, 2003.

[107] E. Love and D.L. Sulsky. An energy-consistent material-point method for dynamic finite deformation plasticity. *International Journal for Numerical Methods in Engineering*, 65(10):1608–1638, 2006.

[108] E. Love and D.L. Sulsky. An unconditionally stable, energy-momentum consistent implementation of the material-point method. *Computer Methods in Applied Mechanics and Engineering*, 195(33-36):3903–3925, 2006.

[109] L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, 1977.

[110] J. Lysmer and R.L. Kuhlemeyer. Finite dynamic model for infinite media. *Journal of the Engineering Mechanics Division*, 95(EM4):859–877, 1969.

[111] J. Ma, H. Lu, and R. Komanduri. Structured mesh refinement in generalized interpolation material point (GIMP) method for simulation of dynamic problems. *Computer Modeling in Engineering and Sciences*, 12:213–227, 2006.

[112] S. Ma, X. Zhang, and X. M. Qiu. Comparison study of MPM and SPH in modeling hypervelocity impact problems. *International Journal of Impact Engineering*, 36(2): 272–282, 2009.

[113] X. Ma, P.T. Giguere, B. Jayaraman, and D.Z. Zhang. Distribution coefficient algorithm for small mass nodes in material point method. *Journal of Computational Physics*, 229(20):7819–7833, 2010.

[114] M.E. Mabsout and S.M. Sadek. A study of the effect of driving on pre-bored piles. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(2):133–146, 2003.

[115] M.E. Mabsout and J.L. Tassoulas. A finite element model for the simulation of pile driving. *International Journal for Numerical Methods in Engineering*, 37(2):257–278, 1994.

[116] M.E. Mabsout, L.C. Reese, and J.L. Tassoulas. Study of pile driving by finite-element method. *Journal of Geotechnical Engineering*, 121(7):535–543, 1995.

[117] M.E. Mabsout, S.M. Sadek, and T.E. Smayra. Pile driving by numerical cavity expansion. *International Journal for Numerical and Analytical Methods in Geomechanics*, 23(11):1121–1140, 1999.

[118] P. Mackenzie-Helnwein, P. Arduino, W. Shin, J.A. Moore, and G.R. Miller. Modeling strategies for multiphase drag interactions using the material point method. *International Journal for Numerical Methods in Engineering*, 83(3):295–322, 2010.

[119] L.E Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

[120] B.M. Marder. GAP–a PIC-type fluid code. *Mathematics of Computation*, 29(130): 434–446, 1975.

[121] R.L. McCrory, R.L. Morse, and K.A. Taggart. Growth and saturation of instability of spherical implosions driven by laser or charged particle beams. *Nuclear Science and Engineering*, 64(1):163–176, 1977.

[122] C. Miehe. *Computational Mechanics of Materials*. A course in the frame of COMMAS master program, University of Stuttgart, 2009.

[123] J. J. Monaghan. An introduction to SPH. *Computer Physics Communications*, 48(1): 89–96, 1988.

[124] L. Moresi, H.-B. Mühlhaus, and F. Dufour. Particle-in-cell solutions for creeping viscous flows with internal interfaces. In *Bifurcation and Localisation Theory in Geomechanics*, pages 345–354. Swets & Zeitlinger B.V., 2001.

[125] L. Moresi, F. Dufour, and H.-B. Mühlhaus. A lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. *Journal of Computational Physics*, 184(2):476–497, 2003.

[126] J.A. Nairn. Material point method calculations with explicit cracks. *Computer Modeling in Engineering and Sciences*, 4(6):649–664, 2003.

[127] J.A. Nairn. Numerical simulations of transverse compression and densification in wood. *Wood and fiber science*, 38(4):576–591, 2006.

[128] J.A. Nairn. Material point method simulations of transverse fracture in wood with realistic morphologies. *Holzforschung*, 61(4):375–381, 2007.

[129] J.A. Nairn and Y.J. Guo. Material point method calculations with explicit cracks, fracture parameters, and crack propagation. In *11th International Conference on Fracture*, Turin, Italy, March 20–25, 2005.

[130] J.A. Nairn and N. Matsumoto. Fracture modeling of crack propagation in wood and wood composites including crack tip processes and fiber bridging mechanics. In *Proceedings of the 12th International Conference on Fracture*, pages 12–17, Ottawa, 2009.

[131] A. Niemunis and I. Herle. Hypoplastic model for cohesionless soils with elastic strain range. *Mechanics of Cohesive-frictional Materials*, 2(4):279–299, 1997.

[132] A. Nishiguchi and T. Yabe. Second-order fluid particle scheme. *Journal of Computational Physics*, 52(2):390–413, 1983.

[133] P. Nithiarasu. A matrix free fractional step method for static and dynamic incompressible solid mechanics. *International Journal for Computational Methods in Engineering Science and Mechanics*, 7(5):369–380, 2006.

[134] E. Oñate, S. R. Idelsohn, F. Del Pin, and R. Aubry. The particle finite element method–an overview. *International Journal of Computational Methods*, 1(2):267–307, 2004.

[135] X.F. Pan, A. Xu, G. Zhang, P. Zhang, J. Zhu, S. Ma, and X. Zhang. Three-dimensional multi-mesh material point method for solving collision problems. *Communications in Theoretical Physics*, 49(5):1129–1138, 2008.

[136] S.G. Parker, J. Guilkey, and T. Harman. A component-based parallel infrastructure for the simulation of fluid–structure interaction. *Engineering with Computers*, 22(3): 277–292, 2006.

[137] M. Paz. *Structural Dynamics: Theory and Computation*. Van Nostrand Reinhold, New York, 2nd edition, 1985.

[138] A.N.B. Poliakov, P.A. Cundall, Y.Y. Podladchikov, and V.A. Lyakhovsky. An explicit inertial method for the simulation of viscoelastic flow: An evaluation of elastic effects on diapiric flow in two-and three-layers models. In *Flow and Creep in the Solar System: Observations, Modeling and Theory*, pages 175–195. Kluwer Academic Publishers, 1993.

[139] W. Prager. *Introduction to Mechanics of Continua*. Dover Publications, New York, 1973.

[140] N. Prime, F. Dufour, and F. Darve. Modeling of landslides with a finite element method with Lagrangian integration points. In *Proceedings of the 2nd International Symposium on Computational Geomechanics (COMGEOII)*, pages 354–363, Cavtat-Dubrovnik, Croatia, April 2011.

[141] J.W.S. Rayleigh. *The Theory of Sound*. Macmillan, London, 1877.

[142] W. Ritz. Über eine neue methode zur lösung gewisser variationsprobleme der mathematischen physik. *Journal für die reine und angewandte Mathematik*, 135:1–61, 1909.

[143] A. Sadeghirad, R.M. Brannon, and J. Burghardt. A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. *International Journal for Numerical Methods in Engineering*, 86:1435–1456, 2011.

[144] T. Schanz. *Zur Modellierung des mechanischen Verhaltens von Reibungsmaterialien*. PhD thesis, Mitteilung 45 des Instituts für Geotechnik, Universität Stuttgart, 1998.

[145] T. Schanz and P.A. Vermeer. On the stiffness of sands. *Géotechnique*, 48:383–387, 1998.

[146] H.L. Schreyer, D.L. Sulsky, and S.-J. Zhou. Modeling delamination as a strong discontinuity with the material point method. *Computer Methods in Applied Mechanics and Engineering*, 191(23-24):2483–2507, 2002.

[147] S.A. Silling. Simulation of penetration and perforation with CTH. Technical report, Sandia National Laboratories, Albuquerque, New Mexico, United States, 1993.

[148] E.A. Smith. Pile-driving analysis by the wave equation. *American Society of Civil Engineers Transactions*, 127:1145–1193, 1962.

[149] I.M. Smith and Y.K. Chow. Three-dimensional analysis of pile drivability. In *Proceeding of the 2nd International Conference on Numerical Methods in Offshore Piling*, pages 1–20, Austin, April 1982.

[150] I.M. Smith and D.V. Griffiths. *Programming the Finite Element Method*. John Wiley & Sons Inc, 2004.

[151] W.D. Smith. A nonreflecting plane boundary for wave propagation problems. *Journal of Computational Physics*, 15(4):492–503, 1974.

[152] D.M. Snider. An incompressible three-dimensional multiphase particle-in-cell model for dense particle flows. *Journal of Computational Physics*, 170(2):523–549, 2001.

[153] M. Steffen, R.M. Kirby, and M. Berzins. Analysis and reduction of quadrature errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering*, 76(6):922–948, 2008.

[154] M. Steffen, P.C. Wallstedt, J.E. Guilkey, R.M. Kirby, and M. Berzins. Examination and analysis of implementation choices within the material point method (MPM). *Computer Modeling in Engineering and Sciences*, 32(2):107–127, 2008.

[155] M. Steffen, R.M. Kirby, and M. Berzins. Decoupling and balancing of space and time errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering*, 82(10):1207–1243, 2010.

[156] H. Sturm and L. Andresen. Large deformation analysis of the installation of dynamic anchor. In *7th European Conference on Numerical Methods in Geotechnical Engineering (NUMGE)*, pages 255–260, 2010.

[157] D. Sulsky. A numerical study of compaction of dry granular material. In *Proceedings of Materials Research Society Symposium*. Cambridge University Press, 2003.

[158] D. Sulsky and J.U. Brackbill. A numerical method for suspension flow. *Journal of Computational Physics*, 96(2):339–368, 1991.

[159] D. Sulsky and A. Kaul. Implicit dynamics in the material-point method. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1137–1170, 2004.

[160] D. Sulsky and H. Schreyer. A particle method with large rotations applied to the penetration of history-dependent materials. *Advances in Numerical Simulation Techniques for Penetration and Perforation of Solids, ASME, AMD*, 171:95–102, 1993.

[161] D. Sulsky and H. Schreyer. The particle-in-cell method as a natural impact algorithm. *Advanced Computational Methods for Material Modeling, ASME, AMD*, 268: 219–229, 1993.

[162] D. Sulsky and H.L. Schreyer. Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):409–429, 1996.

[163] D. Sulsky and L. Schreyer. MPM simulation of dynamic material failure with a decohesion constitutive model. *European Journal of Mechanics-A/Solids*, 23(3):423–445, 2004.

[164] D. Sulsky, Z. Chen, and H.L. Schreyer. A particle method for history-dependent materials. Technical Report SAND93-7044, Sandia National Laboratories, Albuquerque, New Mexico, United States, 1993.

[165] 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.

[166] 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.

[167] D. Sulsky, H. Schreyer, K. Peterson, R. Kwok, and M. Coon. Using the material-point method to model sea ice dynamics. *Journal of Geophysical Research*, 112(c2): C02S90, 2007.

[168] T. Tajima, J.N. Leboeuf, and J.M. Dawson. A magnetohydrodynamic particle code with force free electrons for fluid simulations. *Journal of Computational Physics*, 38 (2):237–250, 1980.

[169] H. Tan and J.A. Nairn. Hierarchical, adaptive, material point method for dynamic energy release rate calculations. *Computer Methods in Applied Mechanics and Engineering*, 191(19-20):2095–2109, 2002.

[170] R.L. Taylor. A mixed-enhanced formulation for tetrahedral finite elements. *International Journal for Numerical Methods in Engineering*, 47(1-3):205–227, 2000.

[171] K. Terzaghi. *Erdbaumechanik*. Deuticke, Wien, 1925.

[172] W. Thomson. *Theory of Vibration with Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[173] T. Timberlake and J.E. Hasbun. Computation in classical mechanics. *American Journal of Physics*, 76(4-5):334–339, 2008.

[174] P. To and I.M. Smith. Technical notes on practical applications-A note on finite element simulations of pile driving. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12:213–219, 1988.

[175] M.J. Turner, R.W. Clough, H.C. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *Journal of Aeronautical Sciences*, 23(9):805–823, 1956.

[176] M.J. Turner, E.H. Dill, H.C. Martin, and R.J. Melosh. Large deflections of structures subjected to heating and external loads. *Journal of Aeronautical Sciences*, 27(2):97–102, 1960.

[177] J. van Esch, P. Bonnier, and H. Teunissen. Biot consolidation and multiphase subsurface flow. Unpublished report, Deltares, October 2009.

[178] J. van Esch, D. Stolle, and I. Jassim. Finite element method for coupled dynamic flow-deformation simulation. In *2nd International Symposium on Computational Geomechanics (COMGEO II)*, Cavtat-Dubrovnik, Croatia, April 2011.

[179] P.A. Vermeer. *Geoengineering*. A course in the frame of WAREM master program, University of Stuttgart, 2009.

[180] P.A. Vermeer. *Elastoplastic Soil Modelling*. A course in the frame of COMMAS master program, University of Stuttgart, 2011.

[181] P.A. Vermeer. *On the Oedometer Stiffness of Soils*. Plaxis course, Schiphol Airport, the Netherlands, January 2012.

[182] P.A. Vermeer, L. Beuth, and T. Benz. A quasi-static method for large deformation problems in geomechanics. In *Proceedings of the 12th International Conference of International Association for Computer Methods and Advances in Geomechanics (IACMAG)*, pages 55–63, Goa, India, 2008.

[183] A. Verruijt. *An introduction to soil dynamics*, volume 24. Springer Verlag, 2010.

[184] P.-A. von Wolffersdorff. A hypoplastic relation for granular materials with a predefined limit state surface. *Mechanics of Cohesive-frictional Materials*, 1(3):251–271, 1996.

[185] P.C. Wallstedt. *On the Order of Accuracy of the Generalized Interpolation Material Point Method*. PhD thesis, University of Utah, 2008.

[186] P.C. Wallstedt and J.E. Guilkey. Improved velocity projection for the material point method. *Computer Modeling in Engineering and Sciences*, 19:223–232, 2007.

[187] P.C. Wallstedt and J.E. Guilkey. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics*, 227(22):9628–9642, 2008.

[188] P.C. Wallstedt and J.E. Guilkey. A weighted least squares particle-in-cell method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 85 (13):1687–1704, 2011.

[189] J. Wan. *Stabilized Finite Element Methods for Coupled Geomechanics and Multiphase Flow*. PhD thesis, Stanford University, USA, 2002.

[190] Z. Więckowski. Analysis of granular flow by the material point method. In *European Conference on Computational Mechanics ECCM*, Cracow, Poland, June 2001.

[191] Z. Więckowski. The dynamic analysis of large strain problems by the material point method. In *5th World Congress on Computational Mechanics WCCM V*, Vienna, Austria, July 2002.

[192] Z. Więckowski. The material point method in large strain engineering problems. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4417–4438, 2004.

[193] Z. Więckowski and I. Jassim. Oral communications. Stuttgart, 2009.

[194] Z. Więckowski, S.-K. Youn, and J.-H. Yeon. A particle-in-cell solution to the silo discharging problem. *International Journal for Numerical Methods in Engineering*, 45 (9):1203–1225, 1999.

[195] A.R. York II. *The Development of Modifications to the Material Point Method for the Simulation of Thin Membranes, Compressible Fluids, and Their Interactions*. PhD thesis, The University of New Mexico, 1997.

[196] A.R. York II, D. Sulsky, and H.L. Schreyer. The material point method for simulation of thin membranes. *International Journal for Numerical Methods in Engineering*, 44(10):1429–1456, 1999.

[197] A.R. YorkII, D. Sulsky, and H.L. Schreyer. Fluid–membrane interaction based on the material point method. *International Journal for Numerical Methods in Engineering*, 48(6):901–924, 2000.

[198] D.Z. Zhang, Q. Zou, W.B. VanderHeyden, and X. Ma. Material point method applied to multiphase flows. *Journal of Computational Physics*, 227(6):3159–3173, 2008.

[199] H.W. Zhang, K.P. Wang, and Z. Chen. Material point method for dynamic analysis of saturated porous media under external contact/impact of solid bodies. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20):1456–1472, 2009.

[200] X. Zhang, K.Y. Sze, and S. Ma. An explicit material point finite element method for hyper-velocity impact. *International Journal for Numerical Methods in Engineering*, 66(4):689–706, 2006.

[201] S. Zhou. *The Numerical Prediction of Material Failure Based on the Material Point Method*. PhD thesis, University of New Mexico, Albuquerque, 1998.

[202] S. Zhou, J. Stormont, and Z. Chen. Simulation of geomembrane response to settlement in landfills by using the material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 23(15):1977–1994, 1999.

[203] O.C. Zienkiewicz. *The Finite Element Method*. McGraw–Hill, Berkshire, 1977.

[204] O.C. Zienkiewicz and Y.K. Cheung. *The Finite Element Method in Structural and Continuum Mechanics*. McGraw–Hill, London, 1967.

[205] O.C. Zienkiewicz and T. Shiomi. Dynamic behaviour of saturated porous media: The generalized Biot formulation and its numerical solution. *International Journal for Numerical and Analytical Methods in Geomechanics*, 8(1):71–96, 1984.

[206] O.C. Zienkiewicz, M. Watson, and I.P. King. A numerical method of visco-elastic stress analysis. *International Journal of Mechanical Sciences*, 10:807–827, 1968.

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

[208] O.C. Zienkiewicz, D.K. Paul, and A.H.C. Chan. Unconditionally stable staggered solution procedure for soil-pore fluid interaction problems. *International Journal for Numerical Methods in Engineering*, 26(5):1039–1055, 1988.

[209] O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, D.K. Paul, and T. Shiomi. Static and dynamic behaviour of soils: A rational approach to quantitative solutions. I. Fully saturated problems. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 285–309. The Royal Society, 1990.

[210] O.C. Zienkiewicz, J. Rojek, R.L. Taylor, and M. Pastor. Triangles and tetrahedra in explicit dynamic codes for solids. *International Journal for Numerical Methods in Engineering*, 43(3):565–583, 1998.

[211] O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, B.A. Schrefler, and T. Shiomi. *Computational Geomechanics with Special Reference to Earthquake Engineering*. Wiley, Chichester, 1999.

Appendix A

Formulation of frictional and adhesive contact

This appendix is devoted to the formulation of contact in the framework of the dynamic MPM. Section A.1 presents the formulation and solution procedure of frictional contact algorithm. This algorithm is extended to a more general adhesive contact in Section A.2.

A.1 MPM formulation and solution procedure of frictional contact

Let us consider a general representative case of two solid continuums in contact as shown in Figure A.1. A boundary representing the contact surface must be defined. Such a contact surface is denoted as $\partial\Omega_{\tau^c}$ and indicated in Figure A.1. Hence, the boundary of a continuum is now defined as

$$\partial\Omega_u \cup \partial\Omega_\tau \cup \partial\Omega_{\tau^{vb}} \cup \partial\Omega_{\tau^c} = \partial\Omega \tag{A.1}$$

and

$$\partial\Omega_i \cap \partial\Omega_j = \varnothing \qquad i,j = u,\tau,\tau^{vb},\tau^c. \tag{A.2}$$

The contact surface $\partial\Omega_{\tau^c}$ represents the intersection of the surfaces of the two contacting bodies, i.e.,

$$\partial\Omega_{\tau^c} = \partial\Omega^1 \cap \partial\Omega^2 \tag{A.3}$$

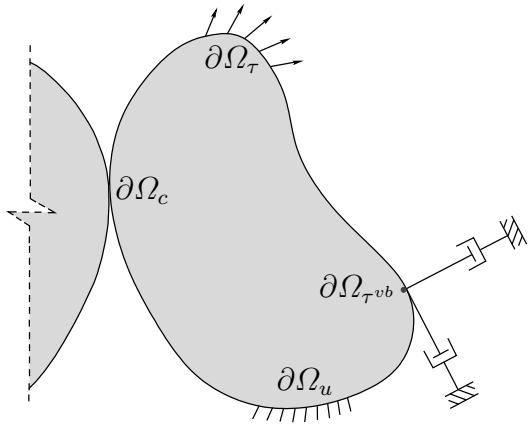


Figure A.1: Two solid continuums in contact

with $\partial\Omega^1$ and $\partial\Omega^2$ being the boundaries of the two contacting bodies.

The contact algorithm adopted in this thesis might be viewed as a predictor-corrector scheme, in which the velocity is predicted from the solution of each body separately and then corrected using the velocity of the coupled bodies following Coulomb friction. In this section, we consider the detailed formulation and solution procedure of the frictional contact problem.

A.1.1 Initialization of equations of motion

To keep matters simple, we consider the case of two bodies in contact as shown in Figure A.2. One body is labeled as g indicating grey and the other as b indicating black. Let us assume that the bodies are in contact at time t. The procedure for solving contact starts with initializing the equations of motion for each body separately as well as for the combined bodies on virtual meshes, requiring $n_c + 1$ virtual meshes, where n_c denotes the number of contacting bodies. In our example, we require 3 virtual meshes; two for solving the equations of motion of body g and body b separately and one to solve the equations of motion of the combined bodies. Considering body g as example for the initialization of the equations of motion on its own mesh, one can write

$$\boldsymbol{M}_g^t \, \boldsymbol{a}_g^t = \boldsymbol{F}_g^{trac,t} + \boldsymbol{F}_g^{grav,t} - \boldsymbol{F}_g^{int,t} = \boldsymbol{F}_g^t. \tag{A.4}$$

We presume that the reader is familiar with the above equation from previous chapters. The subscript indicates body g. In this appendix, we disregard the forces induced by the viscous boundaries as well as damping forces. This is only to simplify the equations. Such forces can be added easily to the equations of motion as done in Chapter 4. The contact forces are not added to the equations of motion, as we predict the solution for each body separately. The effect of such forces will be taken into account later as an additional constraint to the predicted velocity. The mass matrix of body g is constructed as

$$\boldsymbol{M}_g^t = \bigA_{e=1}^{\bar{n}_{elm}} \boldsymbol{M}_{e,g}^t \tag{A.5}$$

with

$$\boldsymbol{M}_{e,g}^t = \begin{bmatrix} \boldsymbol{m}_{1,g}^t & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{m}_{2,g}^t & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{m}_{en,g}^t \end{bmatrix}. \tag{A.6}$$

The matrix $\boldsymbol{0}$ is a null matrix and $\boldsymbol{m}_{i,g}$ represents the mass of node i of body g. It is defined as

$$\boldsymbol{m}_{i,g}^t = \begin{bmatrix} m_{i,g}^t & 0 & 0 \\ 0 & m_{i,g}^t & 0 \\ 0 & 0 & m_{i,g}^t \end{bmatrix}. \tag{A.7}$$

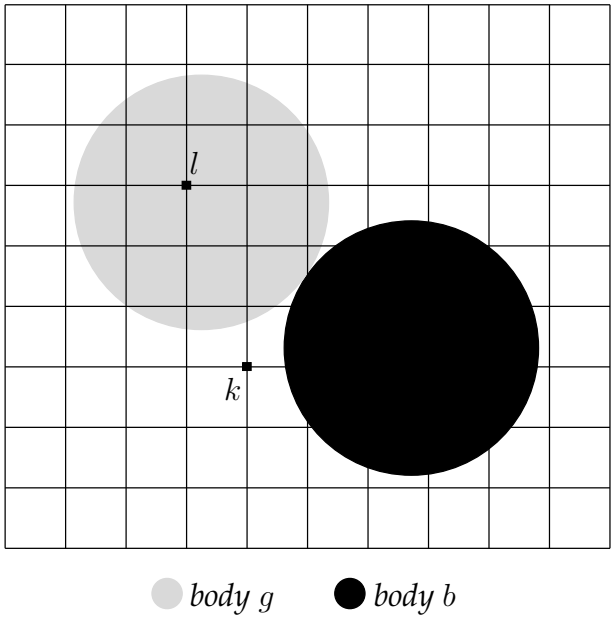


Figure A.2: Illustrative example of two bodies in contact[a]

[a] Although particles are not shown for simplicity, each body represents a cloud of particles.

The mass $m_{i,g}^t$ is integrated considering only particles that belong to body g, i.e.,

$$m_{i,g}^t \approx \sum_{p=1}^{n_{ep,g}} m_p \, N_i \left(\boldsymbol{\xi}_p^t\right) \tag{A.8}$$

with $n_{ep,g}$ being the number of particles that belong to body g in the element. The initial nodal velocity vector of body g at time t is calculated from the following equation

$$\boldsymbol{M}_g^t \, \boldsymbol{v}_g^t = \boldsymbol{P}_g^t, \tag{A.9}$$

in which

$$\boldsymbol{P}_g^t \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep,g}} m_p \, \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \hat{\boldsymbol{v}}_p^t \tag{A.10}$$

is the nodal momentum vector of body g at time t. The traction force vector is then assembled as follows

$$\tilde{\boldsymbol{F}}_g^{trac,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{\tau elm}} \sum_{p=1}^{n_{ebp,g}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \tilde{\boldsymbol{f}}_p^{trac} \tag{A.11}$$

with $n_{ebp,g}$ being the number of boundary particles that belong to body g in the element. Then

$$\boldsymbol{F}_g^{trac,t} = \tilde{\boldsymbol{F}}_g^{trac,t} \, \mathcal{T}\left(t\right). \tag{A.12}$$

The nodal vector of body forces due to the weight of body g is integrated as follows

$$\boldsymbol{F}_g^{grav,t} \approx \mathop{\mathrm{A}}_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep,g}} \boldsymbol{N}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{f}_p^{grav} \tag{A.13}$$

and the internal forces vector due to stresses of body g is

$$\boldsymbol{F}_g^{int,t} \approx \bigwedge_{e=1}^{\bar{n}_{elm}} \sum_{p=1}^{n_{ep,g}} \boldsymbol{B}^T \left(\boldsymbol{\xi}_p^t\right) \boldsymbol{\sigma}_p^t \, \Omega_p^t. \tag{A.14}$$

The same procedure is repeated for body b to construct its equations of motion. Hence, the resulting equations of motion for that body are

$$\begin{aligned} \boldsymbol{M}_b^t \, \boldsymbol{a}_b^t &= \boldsymbol{F}_b^{trac,t} + \boldsymbol{F}_b^{grav,t} - \boldsymbol{F}_b^{int,t} \\ &= \boldsymbol{F}_b^t. \end{aligned} \tag{A.15}$$

The equations of motion of the combined bodies g and b are obtained by simply adding the masses and forces of the two bodies g and b, i.e.,

$$\boldsymbol{M}^t \, \boldsymbol{a}^t = \boldsymbol{F}^t, \tag{A.16}$$

in which

$$\boldsymbol{M}^t = \boldsymbol{M}_g^t + \boldsymbol{M}_b^t \quad \text{and} \quad \boldsymbol{F}^t = \boldsymbol{F}_g^t + \boldsymbol{F}_b^t. \tag{A.17}$$

A.1.2 Prediction of nodal velocities

Having the equations of motion of body g, b and the combined bodies initialized, they are then solved for the nodal acceleration as

$$\boldsymbol{a}_g^t = \boldsymbol{M}_g^{t,-1} \boldsymbol{F}_g^t \quad , \quad \boldsymbol{a}_b^t = \boldsymbol{M}_b^{t,-1} \boldsymbol{F}_b^t \quad \text{and} \quad \boldsymbol{a}^t = \boldsymbol{M}^{t,-1} \boldsymbol{F}^t. \tag{A.18}$$

The predicted nodal velocities at time $t + \Delta t$ are calculated from the accelerations as

$$\boldsymbol{v}_g^{t+\Delta t} = \boldsymbol{v}_g^t + \Delta t \, \boldsymbol{a}_g^t \quad , \quad \boldsymbol{v}_b^{t+\Delta t} = \boldsymbol{v}_b^t + \Delta t \, \boldsymbol{a}_b^t \quad \text{and} \quad \boldsymbol{v}^{t+\Delta t} = \boldsymbol{v}^t + \Delta t \, \boldsymbol{a}^t. \tag{A.19}$$

The initial velocity vector, $\boldsymbol{v}^t$, of the combined bodies is computed using the same procedure explained in Equations A.9 and A.10 considering particles of both bodies.

A.1.3 Detecting contact nodes

Contact at nodes is detected by comparing the velocity of a single body to that of the combined bodies. When these velocities differ, the node is identified as a contact node. Let us, for example, consider nodes l and k of body g, see Figure A.2. It is clear that the velocity of node l from body g is identical to that of the combined bodies as this node is interior node and hence body b does not contribute to its equations of motion. Hence, one can write the following

$$\boldsymbol{v}_{l,g}^{t+\Delta t} = \boldsymbol{v}_l^{t+\Delta t}, \tag{A.20}$$

which tells that node l is not a contact node and hence no correction is required to its predicted velocity. On considering node k, its combined velocity is different from the single body velocity as this node is shared between the two bodies and hence both of them contribute to equation of motion of the combined bodies. This means that

$$\boldsymbol{v}_{k,g}^{t+\Delta t} \neq \boldsymbol{v}_k^{t+\Delta t} \tag{A.21}$$

and consequently this node is identified as a contact node. In the rest, we consider the contact node k of body g for the formulation of the contact algorithm.

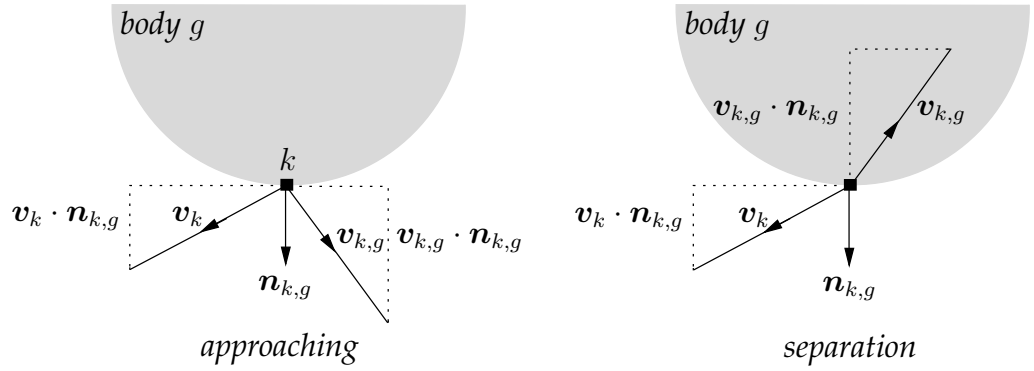


Figure A.3: Approaching and separation cases at a contact node[a]

[a]The time index is dropped and node k is assumed to be on the interface for simple illustration.

A.1.4 Checking for approaching and separation at a contact node

Now we detect whether or not the contact at the contact node k is broken by separation or continued by approaching. This is done by comparing the normal component of the single body velocity with the normal component of the combined bodies velocity. Hence, the following two cases are possible

$$\boldsymbol{v}_{k,g}^{t+\Delta t} \neq \boldsymbol{v}_{k}^{t+\Delta t} \Longrightarrow \begin{cases} \left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^{t} > 0 & \text{approaching} \\ \\ \left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^{t} < 0 & \text{separation} \end{cases} \tag{A.22}$$

with $\boldsymbol{n}$ being the unit outward normal to body g at node k. The cases of approaching and separation are illustrated in Figure A.3. The algorithm allows for free separation and no correction is required in that case. The solution for the separating node k is then $\boldsymbol{v}_{k,g}^{t+\Delta t}$. In case the two bodies are approaching each other at a contact node, we check whether or not sliding occurs.

A.1.5 Check for sliding at an approaching contact node

Let us denote the predicted relative normal and tangential velocities at an approaching contact node k as $\boldsymbol{v}_{k,norm}$ and $\boldsymbol{v}_{k,tan}$, respectively. One can now write

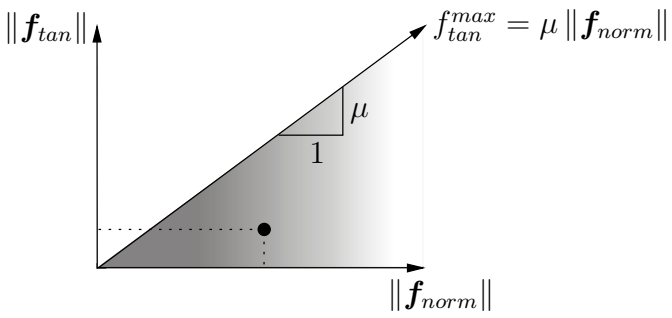
$$\boldsymbol{v}_{k,norm}^{t+\Delta t} = \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^{t}\right] \boldsymbol{n}_{k,g}^{t} \tag{A.23}$$

and

$$\begin{aligned} \boldsymbol{v}_{k,tan}^{t+\Delta t} &= \left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) - \boldsymbol{v}_{k,norm}^{t+\Delta t} \\ &= \left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) - \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^{t}\right] \boldsymbol{n}_{k,g}^{t}. \end{aligned} \tag{A.24}$$

From vector algebra, one can show that

$$\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{a} \cdot \boldsymbol{c})\, \boldsymbol{b} - (\boldsymbol{a} \cdot \boldsymbol{b})\, \boldsymbol{c} \tag{A.25}$$



• *stick contact forces*

Figure A.4: Contact forces for a stick case

with $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ being arbitrary vectors. If the above rule is applied on Equation A.24, the relative tangential velocity can then be re-written as

$$\boldsymbol{v}_{k,tan}^{t+\Delta t} = \boldsymbol{n}_{k,g}^{t} \times \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t} \right) \times \boldsymbol{n}_{k,g}^{t} \right]. \tag{A.26}$$

Having calculated the normal and tangential components of the predicted relative velocity, they can be used to predict the contact forces at the node. Hence, the normal force, $\boldsymbol{f}_{k,norm}$, and tangential force, $\boldsymbol{f}_{k,tan}$, can be computed as

$$\boldsymbol{f}_{k,norm}^{t+\Delta t} = \frac{m_{k,g}^{t}}{\Delta t}\, \boldsymbol{v}_{k,norm}^{t+\Delta t} \qquad \text{and} \qquad \boldsymbol{f}_{k,tan}^{t+\Delta t} = \frac{m_{k,g}^{t}}{\Delta t}\, \boldsymbol{v}_{k,tan}^{t+\Delta t}, \tag{A.27}$$

in which $m_{k,g}$ is the mass of node k integrated from particles of body g, see Equation A.8. According to Coulomb, the magnitude of the maximum allowable tangential force is defined as

$$\begin{aligned} f_{k,tan}^{max,t+\Delta t} &= \mu \, \|\boldsymbol{f}_{k,norm}^{t+\Delta t}\| \\ &= \mu \, \frac{m_{k,g}^{t}}{\Delta t} \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t} \right) \cdot \boldsymbol{n}_{k,g}^{t} \right] \end{aligned} \tag{A.28}$$

with μ being the coefficient of friction between the contacting bodies and $\| \, \|$ indicates the Euclidean norm of a vector, i.e., the magnitude of a vector. Depending on the predicted contact forces, two cases can be distinguished at an approaching contact node as follows

Case I: stick or non-slip contact The contacting bodies stick to each other at the contact node k when

$$\|\boldsymbol{f}_{k,tan}^{t+\Delta t}\| < f_{k,tan}^{max,t+\Delta t}. \tag{A.29}$$

Hence, the solution at this node corresponds to the predicted non-slip velocity, i.e., $\boldsymbol{v}_{k}^{t+\Delta t}$ and no correction is required. The friction force, in this case, balances the tangential force and hence no sliding occurs, i.e.,

$$\boldsymbol{f}_{k,fric}^{t+\Delta t} = -\boldsymbol{f}_{k,tan}^{t+\Delta t}. \tag{A.30}$$

Figure A.4 illustrates the contact forces for a stick contact.

Case II: slip or sliding contact Sliding occurs between the contacting bodies at the contact node k when

$$\|\boldsymbol{f}_{k,tan}^{t+\Delta t}\| > f_{k,tan}^{max,t+\Delta t}. \tag{A.31}$$

In this case, correction is required to ensure that the contacting bodies do not interpenetrate each other and the magnitude of the tangential force should be brought back to the maximum allowable value, i.e., $f_{k,tan}^{max,t+\Delta t}$. Such corrections are discussed in the following section.

A.1.6 Correction for a sliding node

In this section, we present the procedure of correcting the normal and tangential components of the predicted (forces / velocities) at a sliding node.

Correction of the normal component The predicted single body velocity $\boldsymbol{v}_{k,g}^{t+\Delta t}$ is corrected to a new velocity $\tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t}$ such that the normal component of the new velocity is equal to the normal component of the combined bodies velocity, i.e.,

$$\tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t} \cdot \boldsymbol{n}_{k,g}^{t} = \boldsymbol{v}_{k}^{t+\Delta t} \cdot \boldsymbol{n}_{k,g}^{t}, \tag{A.32}$$

which can also be written as

$$\tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t} = \boldsymbol{v}_{k,g}^{t+\Delta t} + \boldsymbol{c}_{k,norm}^{t+\Delta t} \tag{A.33}$$

with

$$\boldsymbol{c}_{k,norm}^{t+\Delta t} = -\left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^{t}\right] \boldsymbol{n}_{k,g}^{t} \tag{A.34}$$

being the correction of the normal component of the predicted velocity. The above correction ensures no interpenetration between the contacting bodies, i.e., it enforces the impenetrability condition to be satisfied at the contact node. The impenetrability condition is written, following the book of Belytschko et al. [21], as

$$\Omega^{g} \cap \Omega^{b} = \varnothing \tag{A.35}$$

with Ω^{g} and Ω^{b} being the domains of the considered contacting bodies.

It is clear from Equation A.33 that the correction of the normal component is equivalent to applying the following normal contact force

$$\begin{aligned} \tilde{\boldsymbol{f}}_{k,norm}^{t+\Delta t} &= \frac{m_{k,g}^{t}}{\Delta t}\, \boldsymbol{c}_{k,norm}^{t+\Delta t} \\ &= -\frac{m_{k,g}^{t}}{\Delta t}\, \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^{t}\right] \boldsymbol{n}_{k,g}^{t}. \end{aligned} \tag{A.36}$$

The graphical illustration of such a correction is shown in Figure A.5.

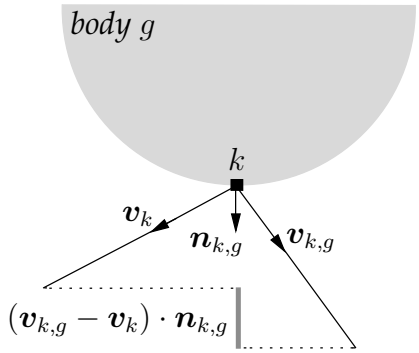


Figure A.5: Correction of the normal component of the velocity

Correction of the tangential component When sliding occurs, the tangential component of the single body velocity should be corrected such that the magnitude of the resulting tangential contact force is bounded by the maximum allowed tangential force, see Equation A.28. Hence, the new tangential force is written as

$$\tilde{\boldsymbol{f}}_{k,tan}^{t+\Delta t} = f_{k,tan}^{max,t+\Delta t}\, \boldsymbol{t} \tag{A.37}$$

with $\boldsymbol{t}$ being a unit vector indicating the direction of the tangential force. It is however calculated from the predicted tangential force, Equation A.27, as

$$\boldsymbol{t} = \frac{\boldsymbol{f}_{k,tan}^{t+\Delta t}}{\|\boldsymbol{f}_{k,tan}^{t+\Delta t}\|} = \frac{\frac{m_{k,g}^t}{\Delta t}\, \boldsymbol{n}_{k,g}^t \times \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t}\right) \times \boldsymbol{n}_{k,g}^t\right]}{\left\| \frac{m_{k,g}^t}{\Delta t}\, \boldsymbol{n}_{k,g}^t \times \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t}\right) \times \boldsymbol{n}_{k,g}^t\right] \right\|}. \tag{A.38}$$

Substituting Equations A.28 into Equation A.37 yields

$$\tilde{\boldsymbol{f}}_{k,tan}^{t+\Delta t} = \mu\, \frac{m_{k,g}^t}{\Delta t} \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^t\right] \boldsymbol{t}. \tag{A.39}$$

Hence, the frictional force is

$$\begin{aligned} \boldsymbol{f}_{k,fric}^{t+\Delta t} &= -\tilde{\boldsymbol{f}}_{k,tan}^{t+\Delta t} \\ &= -\mu\, \frac{m_{k,g}^t}{\Delta t} \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^t\right] \boldsymbol{t}. \end{aligned} \tag{A.40}$$

Figure A.6 illustrates the correction of the tangential forces. Having calculated the contact normal force (Equation A.36) and the frictional force (Equation A.40), one can write the resulting total contact force, $\tilde{\boldsymbol{f}}_{k,cont}^{t+\Delta t}$, as

$$\begin{aligned} \tilde{\boldsymbol{f}}_{k,cont}^{t+\Delta t} &= \tilde{\boldsymbol{f}}_{k,norm}^{t+\Delta t} + \boldsymbol{f}_{k,fric}^{t+\Delta t} \\ &= -\frac{m_{k,g}^t}{\Delta t} \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t}\right) \cdot \boldsymbol{n}_{k,g}^t\right] \left(\boldsymbol{n}_{k,g}^t + \mu \boldsymbol{t}\right), \end{aligned} \tag{A.41}$$

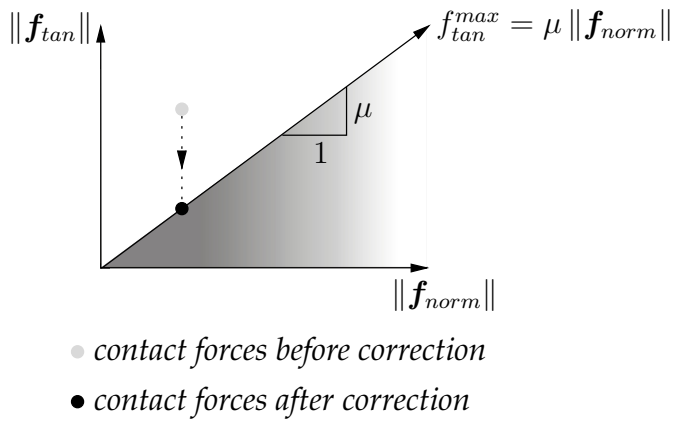


Figure A.6: Contact forces for a slip case

which is used to calculate the corrected velocity, $\tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t}$, as

$$\begin{aligned}\tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t} &= \boldsymbol{v}_{k,g}^{t+\Delta t} + \frac{\tilde{\boldsymbol{f}}_{k,cont}^{t+\Delta t}}{m_{k,g}^{t}}\Delta t \\ &= \boldsymbol{v}_{k,g}^{t+\Delta t} - \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right)\cdot \boldsymbol{n}_{k,g}^{t}\right]\left(\boldsymbol{n}_{k,g}^{t} + \mu \boldsymbol{t}\right)\end{aligned} \tag{A.42}$$

or

$$\tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t} = \boldsymbol{v}_{k,g}^{t+\Delta t} + \boldsymbol{c}_{k,norm}^{t+\Delta t} + \boldsymbol{c}_{k,tan}^{t+\Delta t} \tag{A.43}$$

with $\boldsymbol{c}_{k,norm}^{t+\Delta t}$ being the correction of the normal component of the velocity as given in Equation A.34 and

$$\boldsymbol{c}_{k,tan}^{t+\Delta t} = -\mu\left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k}^{t+\Delta t}\right)\cdot \boldsymbol{n}_{k,g}^{t}\right]\boldsymbol{t} \tag{A.44}$$

is the correction of the tangential component. The flowchart of Figure A.7 summarizes the complete procedure of the adopted contact algorithm.

A.1.7 Updating particles

Having calculated the velocity of the contact node k at time $t+\Delta t$, one must re-calculate the corrected acceleration vector of that node as

$$\tilde{\boldsymbol{a}}_{k,g}^{t} = \frac{\tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t} - \boldsymbol{v}_{k,g}^{t}}{\Delta t}. \tag{A.45}$$

Consequently, the acceleration vector of body g is computed by repeating the same procedure for all active nodes of this body. This acceleration vector is then used to update the velocity and other information of particles that belong to body g following the modified Lagrangian algorithm of Section 4.4. We would like to emphasize here that the same procedure of calculating the corrected velocity and acceleration of body g is then repeated for body b.

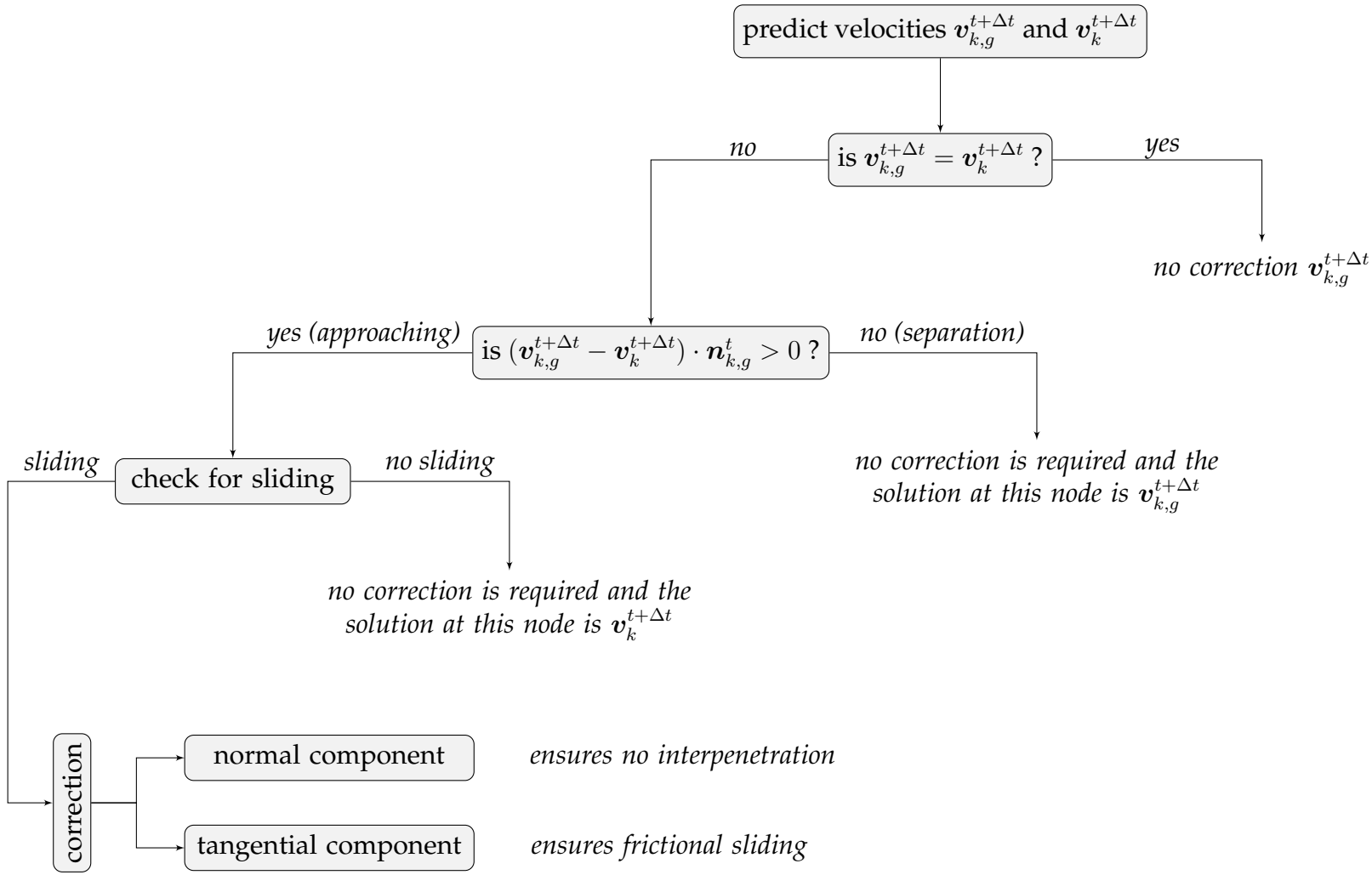


Figure A.7: Flowchart of the contact algorithm applied to node k of body g

A.2 Extending the algorithm to adhesive contact

As the need of adhesive contact algorithm is essential in modeling geomechanical problems, we find it necessary to extend the existing frictional contact to a more general adhesive contact. Equation A.28 is now extended to

$$f_{k,tan}^{max,t+\Delta t} = f_{k,adh}^{t+\Delta t} + \mu \, \| \boldsymbol{f}_{k,norm}^{t+\Delta t} \| \tag{A.46}$$

with $f_{k,adh}^{t+\Delta t}$ being the force induced by the adhesion a which can be written as

$$f_{k,adh}^{t+\Delta t} = a \, A_k^t, \tag{A.47}$$

where A_k is the contact area associated with node k. This area is integrated from the contact elements that share node k. On re-considering Equation A.37 for the correction of the tangential forces, one can write that

$$\tilde{\boldsymbol{f}}_{k,tan}^{t+\Delta t} = f_{k,tan}^{max,t+\Delta t} \, \boldsymbol{t} \tag{A.48}$$

and hence

$$\tilde{\boldsymbol{f}}_{k,tan}^{t+\Delta t} = \left(a \, A_k^t + \mu \, \frac{m_{k,g}^t}{\Delta t} \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t} \right) \cdot \boldsymbol{n}_{k,g}^t \right] \right) \boldsymbol{t}. \tag{A.49}$$

The friction force is

$$\begin{aligned} \boldsymbol{f}_{k,fric}^{t+\Delta t} &= -\tilde{\boldsymbol{f}}_{k,tan}^{t+\Delta t} \\ &= -\left(a \, A_k^t + \mu \, \frac{m_{k,g}^t}{\Delta t} \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t} \right) \cdot \boldsymbol{n}_{k,g}^t \right] \right) \boldsymbol{t} \end{aligned} \tag{A.50}$$

and then the contact force is written as

$$\begin{aligned} \tilde{\boldsymbol{f}}_{k,cont}^{t+\Delta t} &= \tilde{\boldsymbol{f}}_{k,norm}^{t+\Delta t} + \boldsymbol{f}_{k,fric}^{t+\Delta t} \\ &= -\frac{m_{k,g}^t}{\Delta t} \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t} \right) \cdot \boldsymbol{n}_{k,g}^t \right] \left(\boldsymbol{n}_{k,g}^t + \mu \boldsymbol{t} \right) - a \, A_k^t \boldsymbol{t}. \end{aligned} \tag{A.51}$$

Having calculated the contact force, the corrected velocity is calculated as

$$\begin{aligned} \tilde{\boldsymbol{v}}_{k,g}^{t+\Delta t} &= \boldsymbol{v}_{k,g}^{t+\Delta t} + \frac{\tilde{\boldsymbol{f}}_{k,cont}^{t+\Delta t}}{m_{k,g}^t} \Delta t \\ &= \boldsymbol{v}_{k,g}^{t+\Delta t} - \left[\left(\boldsymbol{v}_{k,g}^{t+\Delta t} - \boldsymbol{v}_k^{t+\Delta t} \right) \cdot \boldsymbol{n}_{k,g}^t \right] \left(\boldsymbol{n}_{k,g}^t + \mu \boldsymbol{t} \right) - \frac{a \, A_k^t \Delta t}{m_{k,g}^t} \boldsymbol{t}. \end{aligned} \tag{A.52}$$

Appendix B

The tetrahedral 4-noded element

The isoparametric tetrahedral 4-noded element adopted in this thesis is shown in Figure B.1 in both parent and global domains. For isoparametric elements, the same functions that are used to approximate the field variable are used to map the geometry of the parent domain to the global domain, i.e.

$$\boldsymbol{x}\left(\boldsymbol{\xi}, t\right) \approx \sum_{i=1}^{4} N_i\left(\boldsymbol{\xi}\right) \boldsymbol{x}_i\left(t\right), \tag{B.1}$$

where $\boldsymbol{\xi}$ is a vector indicating the natural or local coordinates system and written as

$$\boldsymbol{\xi} = \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^T \tag{B.2}$$

and $\boldsymbol{x}_i\left(t\right)$ is the vector of the global coordinates of node i at time t. The symbol N_i denotes the shape function of node i written in terms of the local coordinates system $\boldsymbol{\xi}$. The shape functions of the tetrahedral 4-noded element in terms of the local coordinates system $\boldsymbol{\xi}$ are

$$\begin{aligned} N_1\left(\boldsymbol{\xi}\right) &= 1 - \xi_1 - \xi_2 - \xi_3, \\ N_2\left(\boldsymbol{\xi}\right) &= \xi_3, \\ N_3\left(\boldsymbol{\xi}\right) &= \xi_1, \\ N_4\left(\boldsymbol{\xi}\right) &= \xi_2. \end{aligned} \tag{B.3}$$

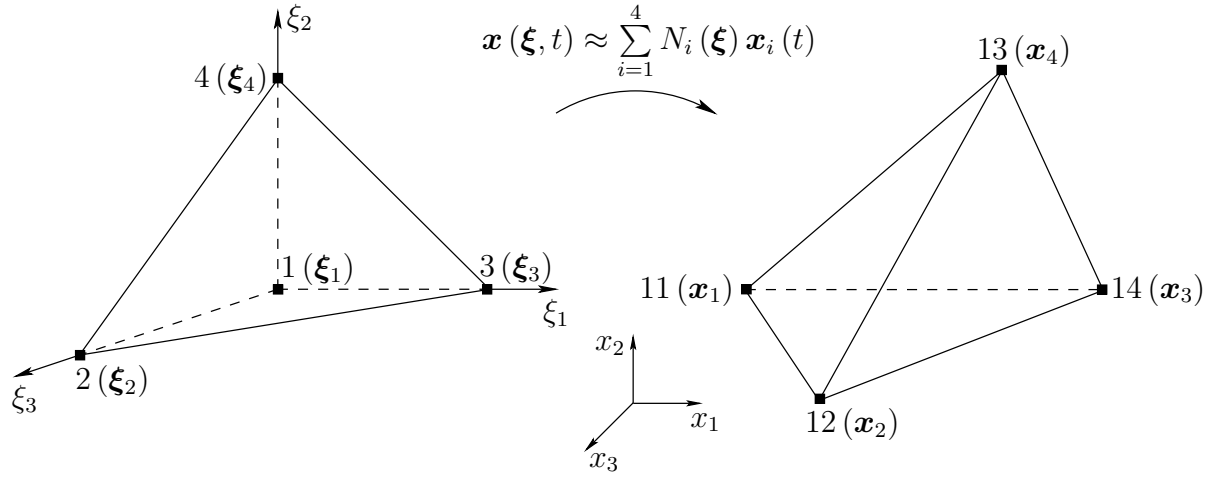
The shape functions matrix for an element reads

$$\boldsymbol{N}\left(\boldsymbol{\xi}\right) = \begin{bmatrix} \boldsymbol{N}_1\left(\boldsymbol{\xi}\right) & \boldsymbol{N}_2\left(\boldsymbol{\xi}\right) & \boldsymbol{N}_3\left(\boldsymbol{\xi}\right) & \boldsymbol{N}_4\left(\boldsymbol{\xi}\right) \end{bmatrix} \tag{B.4}$$

with

$$\boldsymbol{N}_i\left(\boldsymbol{\xi}\right) = \begin{bmatrix} N_i\left(\boldsymbol{\xi}\right) & 0 & 0 \\ 0 & N_i\left(\boldsymbol{\xi}\right) & 0 \\ 0 & 0 & N_i\left(\boldsymbol{\xi}\right) \end{bmatrix}. \tag{B.5}$$

The derivatives of the shape functions with respect to the global coordinates system $\boldsymbol{x}$ are required in the calculation of the strain-displacement matrix. They can be written using the chain rule as follows



$$\boldsymbol{\xi}_1 = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T \quad \boldsymbol{\xi}_2 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T \quad \boldsymbol{\xi}_3 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T \quad \boldsymbol{\xi}_4 = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$$

Figure B.1: The tetrahedral 4-noded element in parent (left) and global (right) domains

$$\begin{aligned}
\frac{\partial N_i(\boldsymbol{\xi})}{\partial x_1} &= \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_1} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_1} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}\frac{\partial \xi_3}{\partial x_1},\\
\frac{\partial N_i(\boldsymbol{\xi})}{\partial x_2} &= \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_2} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_2} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}\frac{\partial \xi_3}{\partial x_2},\\
\frac{\partial N_i(\boldsymbol{\xi})}{\partial x_3} &= \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_3} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_3} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}\frac{\partial \xi_3}{\partial x_3}.
\end{aligned} \tag{B.6}$$

As the derivatives of the natural coordinates $\boldsymbol{\xi}$ to the global coordinates $\boldsymbol{x}$ are not explicitly available, the derivatives in Equation B.6 are determined using an alternative indirect procedure as follows. Let us write

$$\begin{bmatrix}\dfrac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}\\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}\\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}\end{bmatrix} = \begin{bmatrix}\dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_3}{\partial \xi_1}\\ \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_2} & \dfrac{\partial x_3}{\partial \xi_2}\\ \dfrac{\partial x_1}{\partial \xi_3} & \dfrac{\partial x_2}{\partial \xi_3} & \dfrac{\partial x_3}{\partial \xi_3}\end{bmatrix}\begin{bmatrix}\dfrac{\partial N_i(\boldsymbol{\xi})}{\partial x_1}\\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial x_2}\\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial x_3}\end{bmatrix} \qquad i = 1, 4. \tag{B.7}$$

The first matrix on the right-hand side of Equation B.7 is known as the *Jacobian matrix* and denoted $\boldsymbol{J}$. With the use of Equation B.1, $\boldsymbol{J}$ can be put as

$$\boldsymbol{J} = \begin{bmatrix}\dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_3}{\partial \xi_1}\\ \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_2} & \dfrac{\partial x_3}{\partial \xi_2}\\ \dfrac{\partial x_1}{\partial \xi_3} & \dfrac{\partial x_2}{\partial \xi_3} & \dfrac{\partial x_3}{\partial \xi_3}\end{bmatrix} \approx \begin{bmatrix}\displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}x_{i1} & \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}x_{i2} & \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}x_{i3}\\ \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}x_{i1} & \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}x_{i2} & \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}x_{i3}\\ \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}x_{i1} & \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}x_{i2} & \displaystyle\sum_{i=1}^{4}\frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}x_{i3}\end{bmatrix} \tag{B.8}$$

with, for instance, x_{i1} denoting the coordinates x_1 of node i. The Jacobian matrix at any point within the element can be numerically calculated as the derivatives of the shape functions with respect to the local coordinates $\boldsymbol{\xi}$ are explicitly available. The Jacobian matrix can then be easily inverted so that

$$\begin{bmatrix} \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial x_1} \\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial x_2} \\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial x_3} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1} \\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2} \\ \dfrac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3} \end{bmatrix} \qquad i = 1, 4 \tag{B.9}$$

with $\boldsymbol{J}^{-1}$ being the inverse of the Jacobian matrix.

Appendix C

Numerical quadrature

C.1 Numerical quadrature over volume

The integration of a function $f(\boldsymbol{x})$ over the current global domain Ω_e of an element e is related to its integration over the element parent domain $\square$ by

$$\int_{\Omega_e} f(\boldsymbol{x})\, d\Omega = \int_{\square} f(\boldsymbol{x}(\boldsymbol{\xi}))\, |\boldsymbol{J}|\, d\square, \tag{C.1}$$

where $|\boldsymbol{J}|$ denotes the determinant of the Jacobian matrix. The right-hand side of Equation C.1 can be expressed as

$$\int_{\square} f(\boldsymbol{x}(\boldsymbol{\xi}))\, |\boldsymbol{J}|\, d\square = \int_{\xi_1}\int_{\xi_2}\int_{\xi_3} f(\boldsymbol{x}(\boldsymbol{\xi}))\, |\boldsymbol{J}|\, d\xi_1 d\xi_2 d\xi_3 \tag{C.2}$$

and the numerical integration of the function f is then determined as

$$\begin{aligned}\int_{\xi_1}\int_{\xi_2}\int_{\xi_3} & f(\boldsymbol{x}(\boldsymbol{\xi}))\, |\boldsymbol{J}|\, d\xi_1 d\xi_2 d\xi_3 \\ &\approx \sum_{i=1}^{n_{eq1}}\sum_{j=1}^{n_{eq2}}\sum_{k=1}^{n_{eq3}} w_i w_j w_k f(\boldsymbol{x}(\xi_{i1}, \xi_{j2}, \xi_{k3}))\, |\boldsymbol{J}(\xi_{i1}, \xi_{j2}, \xi_{k3})|\end{aligned} \tag{C.3}$$

with n_{eq1}, n_{eq2} and n_{eq3} denote the number of integration points in the directions of ξ_1, ξ_2 and ξ_3 respectively, w_i is the integration weight of quadrature point i and ξ_{i1} indicates the local coordinate ξ_1 of the quadrature point i. The integration of the function $f(\boldsymbol{x})$ can now be written for short as

$$\int_{\Omega_e} f(\boldsymbol{x})\, d\Omega \approx \sum_{q=1}^{n_{eq}} w_q f(\boldsymbol{x}(\boldsymbol{\xi}_q))\, |\boldsymbol{J}(\boldsymbol{\xi}_q)| \tag{C.4}$$

with $n_{eq} = n_{eq1} \times n_{eq2} \times n_{eq3}$ is the total number of quadrature points in an element and $w_q = w_i \times w_j \times w_k$ is the integration weight of quadrature point q. In this thesis, the numerical integrations over the volume of a finite element are carried out using *Gaussian quadrature*. As we consider low order shape functions, a single *Gauss point* per each finite element can be used to perform the numerical integration over the element volume. The coordinates of the single Gauss point in the tetrahedral parent element are $\xi_1 = \xi_2 = \xi_3 =$

$\frac{1}{4}$ and its weight is $w = \frac{1}{6}$ [97]. Let us consider Equation 3.55 again, but replacing $\bar{\boldsymbol{N}}$ by $\boldsymbol{N}$ as the numerical integration is performed in the parent domain. Hence,

$$\left(\mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T \varrho_e \, \boldsymbol{N} \, d\Omega\right) \boldsymbol{a} = \mathop{\mathrm{A}}_{e=1}^{n_{\tau elm}} \int_{\partial\Omega_{\tau e}} \boldsymbol{N}^T \boldsymbol{\tau}_e \, dS + \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{N}^T \varrho_e \, \boldsymbol{g} \, d\Omega - \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{\sigma}_e \, d\Omega. \tag{C.5}$$

Applying the numerical integration of Equation C.4 yields

$$\begin{aligned}\left(\mathop{\mathrm{A}}_{e=1}^{n_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{N}^T(\boldsymbol{\xi}_q) \, \varrho_e(\boldsymbol{x}(\boldsymbol{\xi}_q)) \, \boldsymbol{N}(\boldsymbol{\xi}_q) \, |\boldsymbol{J}(\boldsymbol{\xi}_q)|\right) \boldsymbol{a} \\ = \mathop{\mathrm{A}}_{e=1}^{n_{\tau elm}} \int_{\partial\Omega_{\tau e}} \boldsymbol{N}^T(\boldsymbol{\xi}) \, \boldsymbol{\tau}_e(\boldsymbol{x}(\boldsymbol{\xi}), t) \, dS + \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{N}^T(\boldsymbol{\xi}_q) \, \varrho_e(\boldsymbol{x}(\boldsymbol{\xi}_q)) \, \boldsymbol{g} \, |\boldsymbol{J}(\boldsymbol{\xi}_q)| \\ - \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{B}^T(\boldsymbol{\xi}_q) \, \boldsymbol{\sigma}_e(\boldsymbol{\xi}_q, t) \, |\boldsymbol{J}(\boldsymbol{\xi}_q)|\end{aligned} \tag{C.6}$$

and hence the terms of Equation 3.57 are

$$\begin{aligned}\boldsymbol{M}^c &\approx \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{N}^T(\boldsymbol{\xi}_q) \, \varrho_e(\boldsymbol{x}(\boldsymbol{\xi}_q)) \, \boldsymbol{N}(\boldsymbol{\xi}_q) \, |\boldsymbol{J}(\boldsymbol{\xi}_q)|, \\ \boldsymbol{F}^{trac} &= \mathop{\mathrm{A}}_{e=1}^{n_{\tau elm}} \int_{\partial\Omega_{\tau e}} \boldsymbol{N}^T(\boldsymbol{\xi}) \, \boldsymbol{\tau}_e(\boldsymbol{x}(\boldsymbol{\xi}), t) \, dS, \\ \boldsymbol{F}^{grav} &\approx \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{N}^T(\boldsymbol{\xi}_q) \, \varrho_e(\boldsymbol{x}(\boldsymbol{\xi}_q)) \, \boldsymbol{g} \, |\boldsymbol{J}(\boldsymbol{\xi}_q)|, \\ \boldsymbol{F}^{int} &\approx \mathop{\mathrm{A}}_{e=1}^{n_{elm}} \sum_{q=1}^{n_{eq}} w_q \, \boldsymbol{B}^T(\boldsymbol{\xi}_q) \, \boldsymbol{\sigma}_e(\boldsymbol{\xi}_q, t) \, |\boldsymbol{J}(\boldsymbol{\xi}_q)|.\end{aligned} \tag{C.7}$$

However, $\boldsymbol{F}^{trac}$ requires surface integration, which we shall discuss in the next section.

C.2 Numerical quadrature for surface tractions

Generally, the surface traction vector $\boldsymbol{\tau}(\boldsymbol{x}, t)$ can be seen as a multiplication of space vector $\tilde{\boldsymbol{\tau}}(\boldsymbol{x})$ and a time function $\mathcal{T}(t)$, i.e.,

$$\boldsymbol{\tau}(\boldsymbol{x}, t) = \tilde{\boldsymbol{\tau}}(\boldsymbol{x}) \, \mathcal{T}(t). \tag{C.8}$$

However, when integrating surface tractions, only the space vector $\tilde{\boldsymbol{\tau}}(\boldsymbol{x})$ is involved in the integration. Later during the computations, the time function $\mathcal{T}(t)$ is applied

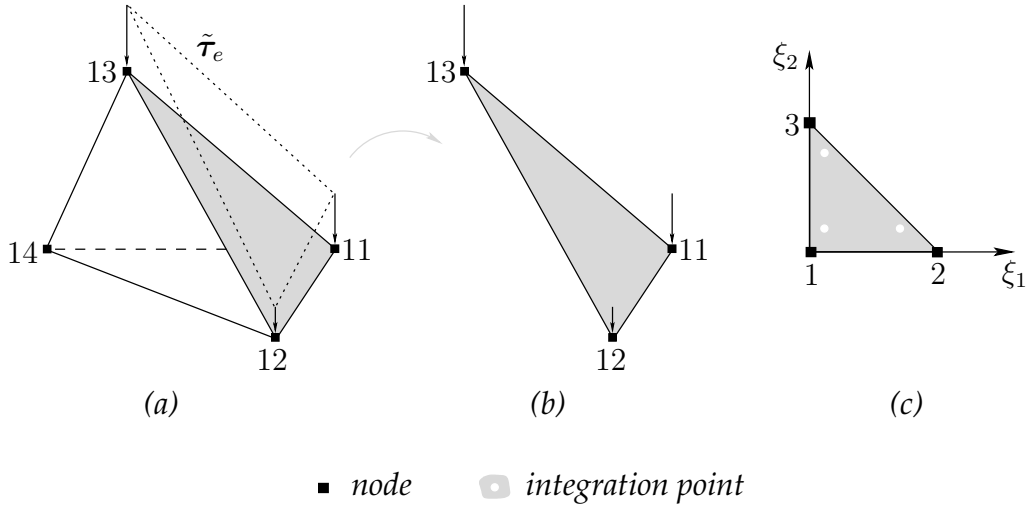


Figure C.1: Surface traction (a) tetrahedral element (b) triangular element in global domain (c) parent domain with three Gauss points

as a multiplier. Let us consider the surface traction acting on the element shown in Figure C.1. According to Equation C.8, the surface traction acting on a triangular element e can be written as

$$\boldsymbol{\tau}_e(\boldsymbol{x}, t) = \tilde{\boldsymbol{\tau}}_e(\boldsymbol{x})\, \mathcal{T}(t). \tag{C.9}$$

Its integration over the area of the triangular element is then

$$\int_{S_e} \boldsymbol{N}^T(\boldsymbol{\xi})\, \boldsymbol{\tau}_e(\boldsymbol{x}, t)\, dS = \left(\int_{S_e} \boldsymbol{N}^T(\boldsymbol{\xi})\, \tilde{\boldsymbol{\tau}}_e(\boldsymbol{x})\, dS \right) \mathcal{T}(t). \tag{C.10}$$

The above equation can be written for short as

$$\boldsymbol{F}_e^{trac}(\boldsymbol{x}, t) = \tilde{\boldsymbol{F}}_e^{trac}(\boldsymbol{x})\, \mathcal{T}(t), \tag{C.11}$$

in which $\boldsymbol{F}_e^{trac}(\boldsymbol{x}, t)$ is the traction force vector of element e at time t and $\tilde{\boldsymbol{F}}_e^{trac}(\boldsymbol{x})$ is the space vector of the traction force. The numerical integration of the space vector over the area of a triangular element e can be written as

$$\begin{aligned} \tilde{\boldsymbol{F}}_e^{trac}(\boldsymbol{x}) &= \int_{S_e} \boldsymbol{N}^T(\boldsymbol{\xi})\, \tilde{\boldsymbol{\tau}}_e(\boldsymbol{x})\, dS \\ &\approx \sum_{q=1}^{n_{seq}} w_q\, \boldsymbol{N}^T(\boldsymbol{\xi}_q)\, \tilde{\boldsymbol{\tau}}_e(\boldsymbol{x}(\boldsymbol{\xi}_q))\, S_e, \end{aligned} \tag{C.12}$$

where $\boldsymbol{\xi}$ is the local coordinate vector in two-dimensions. It is written as

$$\boldsymbol{\xi} = \begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix}^T, \tag{C.13}$$

n_{seq} indicates the number of integration points per triangular element, w_q is the local integration weight of the integration point q, S_e is the surface area of the considered

Table C.1: Positions and weights of Gauss points in triangular element [97]

point	ξ_1	ξ_2	w
1	0.16666 66666 66667	0.16666 66666 66667	0.33333 33333 33333
2	0.66666 66666 66667	0.16666 66666 66667	0.33333 33333 33333
3	0.16666 66666 66667	0.66666 66666 66667	0.33333 33333 33333

triangular element and $\boldsymbol{N}$ is a matrix containing the shape functions of the 3-noded triangular element, which is used to perform the integration of the surface traction in this thesis. The matrix $\boldsymbol{N}$ has the following form

$$\boldsymbol{N}(\boldsymbol{\xi}) = \begin{bmatrix} \boldsymbol{N}_1(\boldsymbol{\xi}) & \boldsymbol{N}_2(\boldsymbol{\xi}) & \boldsymbol{N}_3(\boldsymbol{\xi}) \end{bmatrix} \tag{C.14}$$

with

$$\boldsymbol{N}_i(\boldsymbol{\xi}) = \begin{bmatrix} N_i(\boldsymbol{\xi}) & 0 & 0 \\ 0 & N_i(\boldsymbol{\xi}) & 0 \\ 0 & 0 & N_i(\boldsymbol{\xi}) \end{bmatrix}. \tag{C.15}$$

The shape functions of the 3-noded triangular element are given in terms of the local coordinates system $\boldsymbol{\xi}$ as follows

$$\begin{aligned} N_1(\boldsymbol{\xi}) &= 1 - \xi_1 - \xi_2, \\ N_2(\boldsymbol{\xi}) &= \xi_1, \\ N_3(\boldsymbol{\xi}) &= \xi_2. \end{aligned} \tag{C.16}$$

The element traction $\tilde{\boldsymbol{\tau}}_e$ at an integration point q can be interpolated from the nodal values using the shape functions of the triangular element, i.e.,

$$\tilde{\boldsymbol{\tau}}_e(\boldsymbol{x}(\boldsymbol{\xi}_q)) \approx \sum_{i=1}^{n_{tri}} N_i(\boldsymbol{\xi}_q)\, \tilde{\boldsymbol{\tau}}_e(\boldsymbol{x}_i) \tag{C.17}$$

with n_{tri} represents the number of nodes in the considered triangular element and $\tilde{\boldsymbol{\tau}}_e(\boldsymbol{x}_i)$ denotes the traction vector at node i of element e. Substituting Equation C.17 into Equation C.12 yields

$$\begin{aligned} \tilde{\boldsymbol{F}}_e^{trac}(\boldsymbol{x}) &= \int_{S_e} \boldsymbol{N}^T(\boldsymbol{\xi})\, \tilde{\boldsymbol{\tau}}_e(\boldsymbol{x})\, dS \\ &\approx \sum_{q=1}^{n_{seq}} w_q\, \boldsymbol{N}^T(\boldsymbol{\xi}_q) \left(\sum_{i=1}^{n_{tri}} N_i(\boldsymbol{\xi}_q)\, \tilde{\boldsymbol{\tau}}_e(\boldsymbol{x}_i) \right) S_e. \end{aligned} \tag{C.18}$$

The positions and integration weights of Gauss points in the parent domain of the triangular elements with three integration points are listed in Table C.1. The local nodal coordinates of the 3-nodes in the triangular element, shown in Figure C.1 are

$$\begin{aligned} \boldsymbol{\xi}_1 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T, \\ \boldsymbol{\xi}_2 &= \begin{bmatrix} 1 & 0 \end{bmatrix}^T, \\ \boldsymbol{\xi}_3 &= \begin{bmatrix} 0 & 1 \end{bmatrix}^T. \end{aligned} \tag{C.19}$$

The numbering of Gauss points in the parent element takes the same sequence of numbering the nodes. In this thesis, we use the same time function $\mathcal{T}(t)$ for all elements where surface tractions are prescribed. The global vector of the space traction forces $\tilde{\boldsymbol{F}}^{trac}(\boldsymbol{x})$ can now be assembled from elements vectors

$$\tilde{\boldsymbol{F}}^{trac}(\boldsymbol{x}) = \mathop{\mathrm{A}}_{e=1}^{n_{\tau elm}} \tilde{\boldsymbol{F}}_e^{trac}(\boldsymbol{x}) \tag{C.20}$$

with $n_{\tau elm}$ denotes the number of triangular elements where surface tractions are prescribed. The global traction force vector can now be written as

$$\boldsymbol{F}^{trac}(\boldsymbol{x}, t) = \tilde{\boldsymbol{F}}^{trac}(\boldsymbol{x})\,\mathcal{T}(t)\,. \tag{C.21}$$

Appendix D

The assemblage procedure

For simple explanation of the assemblage procedure of global vectors and matrices from the local ones, we consider two elements from a two-dimensional mesh, see Figure D.1. The considered elements are 4-noded rectangular elements with two degrees-of-freedom per each node. Let us consider the assemblage of a vector $\boldsymbol{F}$ from the local vectors of elements e_1 and e_2, which are denoted as $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ respectively as shown in Figure D.1. Each element vector has 8 components, which are the degrees-of-freedom of the 4-noded rectangular element. The global vector $\boldsymbol{F}$ consists of $(2 \times n_T)$ component, where n_T denotes the total number of nodes in the considered mesh and 2 corresponds to the number of degrees-of-freedom per each node in the considered example. However, in Figure D.1, only a part of the global vector is shown for illustration. The assemblage procedure is done so that each node contributes to its corresponding global storages in

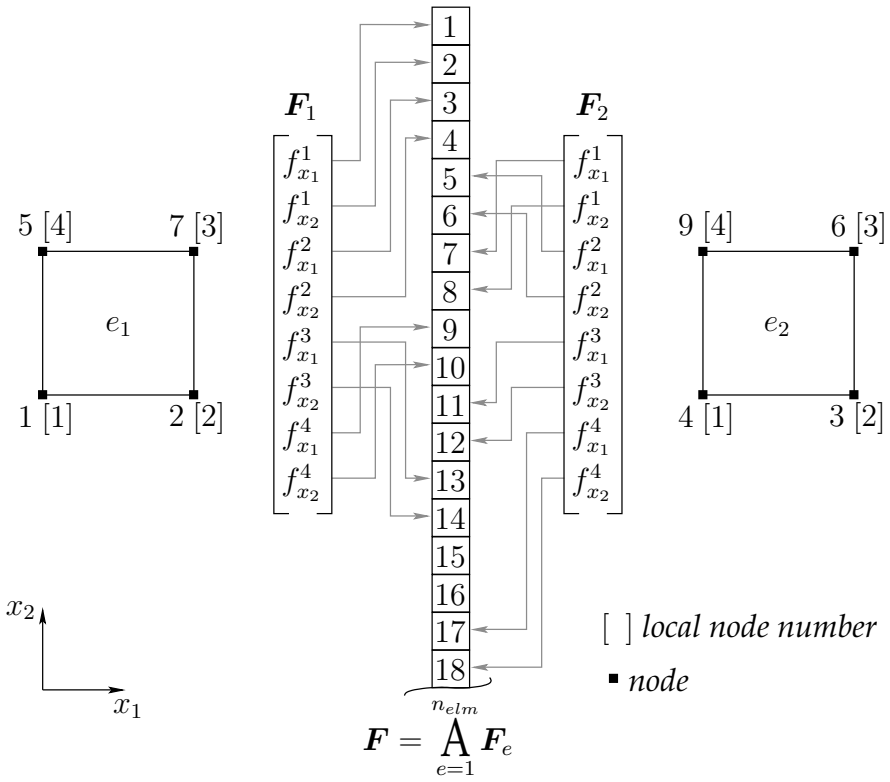


Figure D.1: Illustration of the assemblage procedure of a global vector from elements local vectors in two-dimensional mesh

the global vector. The global storages of a node n are

$$\begin{aligned} n_{x_1} &= (2 \times n) - 1 \\ n_{x_2} &= n_{x_1} + 1 \end{aligned}$$

with n_{x_1} and n_{x_2} are the global degrees-of-freedom of node n in x_1 and x_2 directions respectively. Let us now consider, for instance, node 7 from element e_1. Its global storages are

$$\begin{aligned} n_{x_1} &= (2 \times 7) - 1 \\ &= 13 \\ n_{x_2} &= 14 \end{aligned}$$

and hence node 7 should contribute to the global storages 13 and 14, which is illustrated in Figure D.1. Of course, in the computer implementation of the finite element method, one should store the global numbering of the nodes in a matrix usually called connectivity matrix and hence when looping over the local numbering of the nodes in each element, the corresponding global number of each node can be accessed using the connectivity matrix.

The above procedure can be used in three-dimensional FEM. In this case, the dimension of the global vector should then be $(3 \times n_T)$, where 3 indicates the number of degrees-of-freedom per node. The global storages of a node n in x_1, x_2 and x_3 directions are

$$\begin{aligned} n_{x_1} &= (3 \times n) - 2 \\ n_{x_2} &= n_{x_1} + 1 \\ n_{x_3} &= n_{x_1} + 2. \end{aligned}$$

Appendix E

Initial position of particles in the tetrahedral parent element

Let us define the following

$$a = \frac{5-\sqrt{5}}{20} \quad , \quad b = \frac{5+3\sqrt{5}}{20} \quad , \quad c = \frac{3a+b}{4} \quad \text{and} \quad d = \frac{a+b}{2}. \tag{E.1}$$

The initial position of particles are given in the following tables

Table E.1: One particle per element

particle	ξ_1	ξ_2	ξ_3
1	c	c	c

Table E.2: Four particles per element

particle	ξ_1	ξ_2	ξ_3
1	a	a	a
2	a	a	b
3	b	a	a
4	a	b	a

Table E.3: Eight particles per element

particle	ξ_1	ξ_2	ξ_3
1	$0.5c$	$0.5c$	$0.5c$
2	$0.5c$	$0.5c$	$2.5c$
3	$2.5c$	$0.5c$	$0.5c$
4	$0.5c$	$2.5c$	$0.5c$
5	c	$0.5c$	c
6	c	$1.5c$	c
7	$0.5c$	c	$1.5c$
8	$1.5c$	c	$0.5c$

Table E.4: Ten particles per element

particle	ξ_1	ξ_2	ξ_3
1	a	a	a
2	a	a	b
3	b	a	a
4	a	b	a
5	a	a	d
6	d	a	a
7	a	d	a
8	d	a	d
9	a	d	d
10	d	d	a

Table E.5: Twenty particles per element

particle	ξ_1	ξ_2	ξ_3
1	$0.2c$	$0.3c$	$0.3c$
2	$1.2c$	$0.4c$	$0.2c$
3	$2.2c$	$0.4c$	$0.3c$
4	$3.1c$	$0.4c$	$0.2c$
5	$0.3c$	$1.4c$	$0.2c$
6	$1.4c$	$1.4c$	$0.3c$
7	$2.1c$	$1.4c$	$0.2c$
8	$0.2c$	$2.4c$	$0.3c$
9	c	$2.4c$	$0.4c$
10	$0.3c$	$3.2c$	$0.4c$
11	$0.3c$	$0.4c$	$1.3c$
12	c	$0.4c$	$1.4c$
13	$1.6c$	$0.4c$	$1.3c$
14	$0.4c$	$1.2c$	$1.4c$
15	$0.8c$	$1.2c$	$1.4c$
16	$0.3c$	$1.8c$	$1.3c$
17	$0.3c$	$0.3c$	$2.1c$
18	$0.7c$	$0.3c$	$2.2c$
19	$0.3c$	$0.7c$	$2.2c$
20	$0.3c$	$0.3c$	$3c$

Curriculum Vitæ

9 April 1977	Born in Baghdad, Iraq
1996 – 2000	B.Sc. student in Mechanical Engineering, University of Baghdad
2000 – 2003	M.Sc. student in Mechanical Engineering, University of Baghdad
2003 – 2007	Teaching and research assistant, University of Baghdad
2007 – 2013	Teaching and research assistant, Stuttgart University
2010 – 2011	Secondment at Deltares in the framework of GEO-INSTALL project being funded by the EU
13. 03. 2013	Doctoral graduation with summa cum laude, Stuttgart University
since April 2013	Geomechanics engineer at Shell Global Solutions International B.V., Rijswijk, The Netherlands