









Anura3D MPM Research Community

Anura3D MPM Software

Scientific Manual

Version: 2022
21 April 2022

Anura3D MPM Software, Scientific Manual

With the contributions of:

Francesca Ceccato	Università degli Studi di Padova, Italy
James Fern	Amberg Engineering Ltd / EPFL, Switzerland
Mario Martinelli	Deltares Delft, The Netherlands
Alexander Rohe	Deltares Delft, The Netherlands
Alba Yerro	VirginiaTech, United States

Contributors are listed in alphabetical order. Many people contributed to the creation of this tutorial. If you feel that you are one of them but your name does not appear in this list, please contact us.

Published and printed by:
Anura3D MPM Research Community
c/o Stichting Deltares
Boussinesqweg 1
2629 HV Delft (The Netherlands)

Contact and information:
e-mail: info@Anura3D.com
web: www.Anura3D.com

Contents

Anura3D MPM Research Community
c/o Stichting Deltares
Boussinesqweg 1
2629 HV Delft (The Netherlands)

Contact and information
e-mail: info@Anura3D.com
web: www.Anura3D.com

1 Introduction

This manual presents the the theories and numerical methods used in Anura3D. It contains a chapter on the mathematical formulation and the dynamic-explicit material point formulation of the 1-phase solid, 1-phase liquid and 2-phase mixture. The boundary conditions and advanced concepts used in the current version of Anura3D are also described in two separated sections. Lastly, a dedicated chapter on material models available in this code is also added for completeness. Additional scientific material can be found in the *Publications* section of the website.

2 General Framework

The material point method (MPM) was initially developed by Harlow [1] in Los Alamos National Laboratory. The basis of that method, called initially Particle-In-Cell method (PIC), was to represent fluid flow by a set of material points moving through a background fixed mesh. Later, in 1994 in the New Mexico University, Sulsky and Schreyer [2–4] extended the approach for modelling problems of solid mechanics, where basically the equation of dynamic momentum balance is discretized.

MPM is considered as a method in between the particle-based methods and the Finite Element Method (FEM). This is because it discretizes the media in two different frames. First, the continuum is divided into a set of material points (MPs). Each material point represents a portion of the domain and the mass of such sub-domain, assumed to be fixed during all of the calculation (to ensure mass conservation). In the classical MPM, it is considered to be concentrated at the corresponding material point. Other quantities such as velocities, strains and stresses, are also initialized and carried by the MPs. Each material point moves attached with the deformations of the body and this provides the Lagrangian description of the media.

The second frame is a computational mesh. It is the same as the one used in the conventional FEM and it is built to cover the full domain of the problem. The discrete governing equations are solved at the nodes of the computational mesh. The variables required to solve the equations in the mesh at any step of the analysis are transferred from the material points to the nodes of the mesh by using mapping functions. For instance, the typical linear shape functions used in the FEM. Boundary conditions can be imposed at the mesh nodes or at the material points and the governing equations are solved by using an incremental scheme. Then the quantities carried by the material points are updated through the interpolation of the mesh results, using the same mapping functions. The information associated with the mesh is not required for the next step of the analysis; therefore it can be discarded avoiding any mesh distortion.

Fig. 2.1 is a typical problem setup. The material domain is represented by a set of material points. The computational mesh covers all the computational domain.

At the beginning of each time increment, the discretized governing equations are defined by mapping information from the MPs to the computational nodes of the mesh by means of the interpolation functions (Fig. 2.2a). The governing equations of motion are solved for the primary unknown variables, e.g. the nodal accelerations (Fig. 2.2b). This nodal values are used to update acceleration, velocity and position of MPs, as well as to compute strains and stresses at the MPs (Fig. 2.2c). No permanent information is stored in the mesh, thus it can be freely redefined at the end of the time step, but commonly it is kept fixed. The assignment of MPs to finite elements is updated after mesh adjustment (Fig. 2.2d).

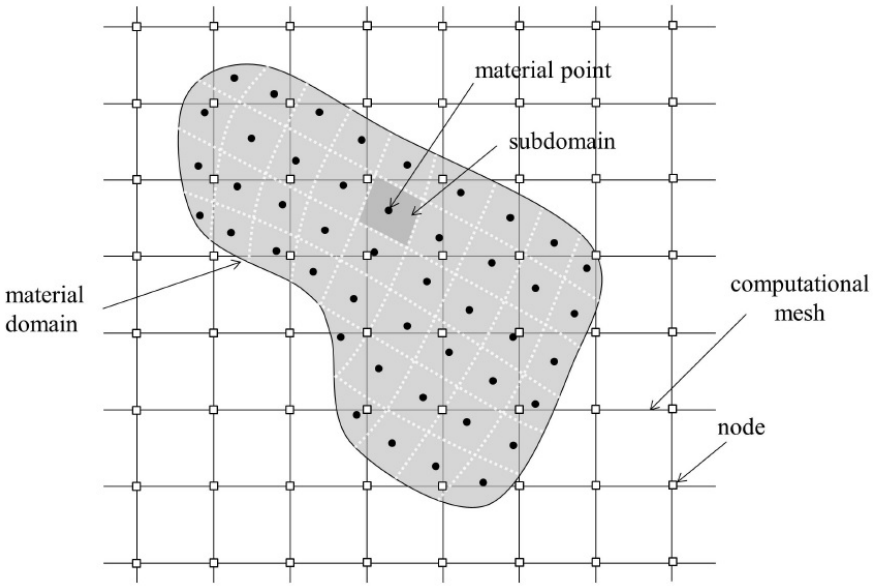


Figure 2.1: *Space discretisation. Nodes of the computational mesh and material points, after [5]*

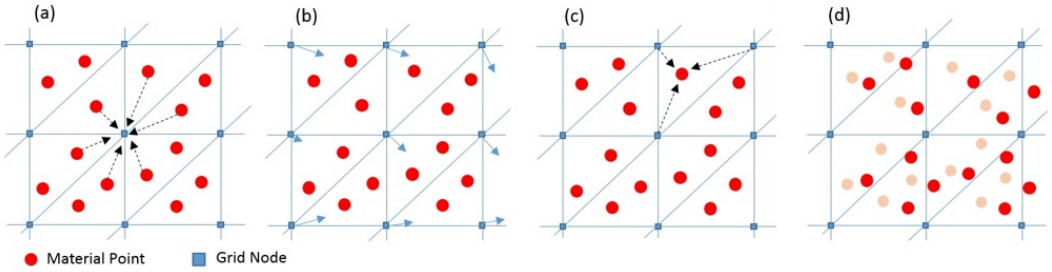


Figure 2.2: *Computation scheme of MPM: (a) map information from MPs to grid nodes; (b) solve governing equations of motion at the nodes; (c) update MP information; (c) update MP position and housekeeping*

3 Mathematical formulation

3.1 Introduction

This chapter presents the basic equations for the dynamic equilibrium of the solid body and the porous media within the framework of continuum mechanics. The variational form, or weak form, of the corresponding partial differential equations is also developed.

3.2 One-phase Single-point formulation for solid analysis

The MPM formulation for mechanical problems (1-phase MPM formulation) is presented in this Section. It is based on [3], in which only a single phase is considered. Because the MPM can be viewed as an extension of the FEM procedure, is important to note that they have much in common. For instance, the weak form of the governing equations as well as the final system of equations posed at the nodes of the computational mesh are identical in both schemes. The most distinguishing feature between them is the calculation of the numerical integrations over the volume of a finite element. Meanwhile in FEM the integrations are carried out using Gaussian quadrature, in MPM they are calculated based on the material points.

Three-dimensional domains are considered in this Chapter. Vectors and tensors are identified with bold type.

3.2.1 Governing equations

From a mathematical point of view, the continuum can be described by a set of differential governing equations. This includes conservation of mass, conservation of momentum, balance of energy and the corresponding boundary conditions. Furthermore, constitutive equations such as stress-strain relationship, which are characteristic of the material forming the continuum, are needed.

3.2.1.1 Mass conservation

The conservation of mass of the solid phase in Lagrangian form is written with the following partial differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho + \rho\nabla\cdot\mathbf{v}_S = 0 \tag{3.1}$$

in which $\mathbf{v}$ is the velocity vector of the solid phase. One of the features of the MPM formulation for a 1-phase material is that the mass of each material point m_{MP} remains constant during all of the calculation. This fact implies that mass conservation is automatically satisfied.

3.2.1.2 Momentum conservation

The momentum conservation of the continuum can be written with the differential Eq. 3.2. A dynamic formulation is considered, which means that the acceleration term is taken into account.

$$\rho\frac{\mathrm{d}\mathbf{v}_S}{\mathrm{d}t} = \nabla\cdot\boldsymbol{\sigma}_S + \rho\mathbf{b} \tag{3.2}$$

The density of the continuum is represented by ρ, $\frac{d\mathbf{v}_S}{dt}$ is the acceleration, $\boldsymbol{\sigma}_S$ is the Cauchy stress tensor, and **b** is the body force vector.

Two kind of boundary conditions are defined: prescribed traction (Eq. 3.3) and prescribed displacements (Eq. 3.4). Each one is applied on the corresponding domain, $\partial\Omega^{\boldsymbol{\sigma}}$ and $\partial\Omega^{\mathbf{u}}$ respectively,

$$\boldsymbol{\sigma}_S(\mathbf{x}, t) \cdot \mathbf{n} = \widehat{\mathbf{t}}_S(t) \tag{3.3}$$

$$\mathbf{u}_S(\mathbf{x}, t) = \widehat{\mathbf{u}}_S(t) \tag{3.4}$$

where $\boldsymbol{\sigma}_S(\mathbf{x}, t)$ is the stress tensor, **n** is the outward unit normal vector of the free surface, and $\widehat{\mathbf{t}}_S(t)$ is the surface traction vector. **x** is the position vector, $\mathbf{u}_S$ is the displacement vector, and t is time.

3.2.2 Weak form

The conservation of the momentum (Eq. 3.2) 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 δv and is integrated over the current domain Ω occupied by the continuum, i.e.,

$$\int_\Omega \delta\mathbf{v}_S \rho \frac{d\mathbf{v}_S}{dt} d\Omega = \int_\Omega \delta\mathbf{v}_S (\nabla \cdot \boldsymbol{\sigma}_S) d\Omega + \int_\Omega \delta\mathbf{v}_S \rho \mathbf{g} d\Omega \tag{3.5}$$

with $\delta\mathbf{v} = 0$ on $\partial\Omega_u$.

The first term on the right-hand side of Eq. (3.5) can be expressed as

$$\int_\Omega \delta\mathbf{v}_S (\nabla \cdot \boldsymbol{\sigma})_S \, d\Omega = \int_\Omega (\nabla \cdot \delta\mathbf{v}_S \boldsymbol{\sigma}_S) \, d\Omega - \int_\Omega (\nabla \cdot \delta\mathbf{v}_S) \boldsymbol{\sigma}_S \, d\Omega \tag{3.6}$$

Using Gauss theorem, also called divergence theorem, the first term on the right-hand side of Eq. (3.6) can be written as

$$\int_\Omega (\nabla \cdot \delta\mathbf{v}_S \boldsymbol{\sigma}_S) \, d\Omega = \int_{\partial\Omega_\sigma} \delta\mathbf{v}_S \, \boldsymbol{\tau}_S \, d\partial\Omega \tag{3.7}$$

Substituting all previous terms in Eq. (3.5), it follows

$$\int_\Omega \delta\mathbf{v}_S \rho \frac{d\mathbf{v}_S}{dt} d\Omega = \int_{\partial\Omega_\sigma} \delta\mathbf{v}_S \, \boldsymbol{\tau}_S \, d\partial\Omega - \int_\Omega (\nabla \cdot \delta\mathbf{v}_S) \boldsymbol{\sigma}_S \, d\Omega + \int_\Omega \delta\mathbf{v}_S \rho \mathbf{g} d\Omega \tag{3.8}$$

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

3.3 One-phase Single-point formulation for liquid analysis

3.3.1 Governing equations

In this section, the conservation equations of mass and linear momentum are written with respect to the liquid phase.

3.3.1.1 Mass conservation

The conservation of mass of the liquid phase in a lagrangian form is written as follows:

$$\frac{d\rho_L}{dt} + \rho_L \nabla \cdot \mathbf{v}_L = 0 \tag{3.9}$$

in which $\mathbf{v_L}$ is the velocity vector of the liquid phase. As written for the solid phase, one of the features of the MPM formulation is that the mass of each material point m_{MP} remains constant during all of the calculation. This fact implies that mass conservation is automatically satisfied.

3.3.1.2 Momentum conservation

The momentum conservation of the continuum can be written with the differential Eq. 3.10. A dynamic formulation is considered, which means that the acceleration term is taken into account.

$$\rho_L \frac{d\mathbf{v}_L}{dt} = \nabla \cdot \boldsymbol{\sigma}_L + \rho_L \mathbf{b} \tag{3.10}$$

The liquid density of the continuum is represented by ρ_L, $\mathbf{v}_L$ is the velocity, $\boldsymbol{\sigma_L}$ is the Cauchy stress tensor, and $\mathbf{b}$ is the body force vector.

Two kind of boundary conditions are defined: prescribed traction (Eq. 3.11) and prescribed displacements (Eq. 3.12, Fig. 3.1). Each one is applied on the corresponding domain, $\partial\Omega^{\boldsymbol{\sigma}}$ and $\partial\Omega^{\mathbf{u}}$ respectively. The boundary condition along $\partial\Omega^{\boldsymbol{\sigma}}$ is defined as follows:

$$\boldsymbol{\sigma}_L(\mathbf{x}, t) \cdot \mathbf{n} = \widehat{\mathbf{t}}(\mathbf{x}, t) \tag{3.11}$$

where $\boldsymbol{\sigma}_L(\mathbf{x}, t)$ is the stress tensor, $\mathbf{n}$ is the outward unit normal vector of the free surface and $\widehat{\mathbf{t}}(\mathbf{x}, t)$ is the surface traction vector. The boundary condition along $\partial\Omega^{\mathbf{u}}$ is defined as follows:

$$\mathbf{u}(\mathbf{x}, t) = \widehat{\mathbf{u}}(t) \tag{3.12}$$

$\mathbf{x}$ is the position vector, $\widehat{\mathbf{u}}$ is the displacement vector and t is time.

The boundary between liquid and gas phase is the *free surface* and is modelled as a boundary condition on $\partial\Omega_\sigma$. In case of zero surface tension and setting the atmospheric pressure to zero, such a boundary condition is set as $\widehat{\mathbf{t}}(\mathbf{x}, t) = p(\mathbf{x}, t) = 0$.

3.3.2 Weak form

The discretization of the equation of conservation of linear momentum for pure liquid is written as follows. The strong form of the equation is 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 δv and is integrated over the current domain Ω occupied by the continuum, i.e.,

$$\int_\Omega \delta\mathbf{v}_L \rho \frac{d\mathbf{v}_L}{dt} d\Omega = \int_\Omega \delta\mathbf{v}_L (\nabla \cdot \boldsymbol{\sigma}_L) d\Omega + \int_\Omega \delta\mathbf{v}_L \rho \mathbf{g} d\Omega \tag{3.13}$$

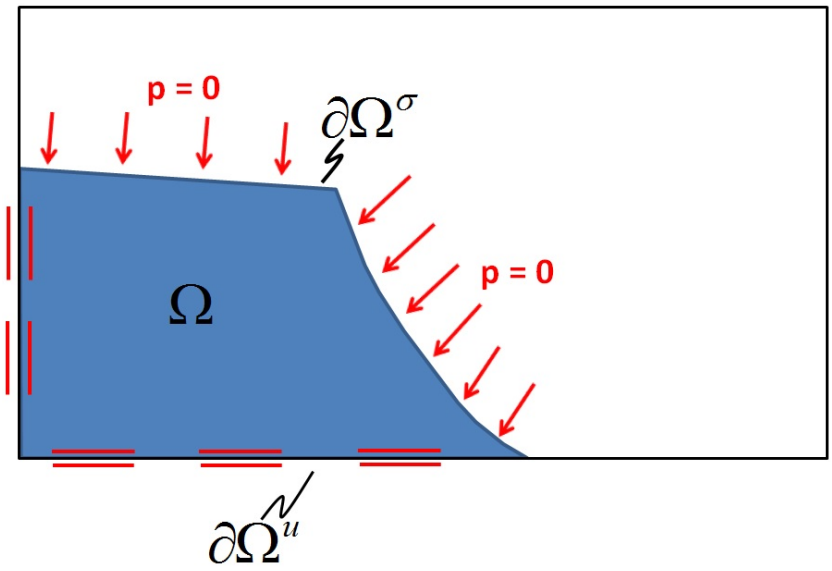


Figure 3.1: *Water column collapse. Boundary conditions: displacements on* $\partial\Omega^{u}$ *and pressure on* $\partial\Omega^{\sigma}$.

with $\delta v = 0$ on $\partial\Omega_u$.

The first term on the right-hand side of Eq. (3.13) can be expressed as

$$\int_{\Omega} \delta \mathbf{v}_L (\nabla \cdot \boldsymbol{\sigma}_L)\, d\Omega = \int_{\Omega} (\nabla \cdot \delta \mathbf{v}_L \boldsymbol{\sigma}_L)\, d\Omega - \int_{\Omega} (\nabla \cdot \delta \mathbf{v}_L) \boldsymbol{\sigma}_L\, d\Omega \tag{3.14}$$

Using Gauss theorem, also called divergence theorem, the first term on the right-hand side of Eq. (3.14) can be written as

$$\int_{\Omega} (\nabla \cdot \delta \mathbf{v}_L \boldsymbol{\sigma}_L)\, d\Omega = \int_{\partial\Omega_\sigma} \delta \mathbf{v}_L\, \boldsymbol{\tau}_L\, d\partial\Omega \tag{3.15}$$

Substituting all previous terms in Eq. (3.5), it follows

$$\int_{\Omega} \delta \mathbf{v}_L \rho \frac{d\mathbf{v}_L}{dt} d\Omega = \int_{\partial\Omega_\sigma} \delta \mathbf{v}_L\, \boldsymbol{\tau}_L\, d\partial\Omega - \int_{\Omega} (\nabla \cdot \delta \mathbf{v}_L) \boldsymbol{\sigma}_L\, d\Omega + \int_{\Omega} \delta \mathbf{v}_L \rho \mathbf{g} d\Omega \tag{3.16}$$

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

3.4 Two-phase Single-point formulation for saturated soil

3.4.1 Governing equations

This section is devoted to the presentation of the governing equations that are implemented in Anura3Dfor the solution of coupled dynamic, two-phase problems. It's worth mentioning that several coupled two-phase formulation have been discussed in the literature since the pioneering work by Terzaghi and Biot. The reader can refer to the work by Zienkiewicz and co-workers ([6], [7] and [8]) for a comprehensive discussion on this topic.

With the formulation used in Anura3D, the equilibrium equations are solved for the accelerations of pore liquid and soil skeleton as the primary unknown variables. This formulation proved to be able to capture the physical response of saturated soil under dynamic as well as static loading [9]. The governing equations are the mass conservation, and the momentum conservation of each phase and the mixture.

3.4.1.1 Mass conservation

The mass conservation of the solid phase is expressed as:

$$\frac{d}{dt}[(1-n)\rho_S] + (1-n)\rho_S \nabla \cdot \mathbf{v}_S = 0 \tag{3.17}$$

in which $\mathbf{v}_S$ is the velocity vector of the solid phase and n is the porosity of the solid skeleton.

On denoting the components of the vector of the (true) velocity of the liquid phase as $\mathbf{v}_L$, the conservation of mass of this phase can be written as:

$$\frac{d(n\rho_L)}{dt} + n\rho_L \nabla \cdot \mathbf{v}_L = 0 \tag{3.18}$$

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 liquid phases to

$$-\frac{dn}{dt} + (1-n)\nabla \cdot \mathbf{v}_S = 0 \tag{3.19}$$

and

$$\rho_L \frac{dn}{dt} + n\frac{d\rho_L}{dt} + n\rho_L \nabla \cdot \mathbf{v}_L = 0 \tag{3.20}$$

respectively. Substituting Equation 3.19 into Equation 3.20 allows to eliminate the term $\frac{dn}{dt}$. Hence,

$$(1-n)\nabla \cdot \mathbf{v}_S + \frac{n}{\rho_L}\frac{d\rho_L}{dt} + n\nabla \cdot \mathbf{v}_L = 0 \tag{3.21}$$

The liquid is considered as weakly compressible material. The effective volumetric strain $\bar{\varepsilon}_L$ in the liquid is defined as

$$\frac{d\bar{\varepsilon}_L}{dt} = -\frac{1}{\rho_L}\frac{d\rho_L}{dt} \tag{3.22}$$

Substituting Equation 3.22 into Equation 3.21 and rearranging terms yields:

$$\frac{d\bar{\varepsilon}_L}{dt} = \frac{1}{n}\left[(1-n)\nabla \cdot \mathbf{v}_S + n\nabla \cdot \mathbf{v}_L\right] \tag{3.23}$$

Equation 3.23 represents the conservation of mass of the saturated soil. It is also known as storage equation.

3.4.1.2 Momentum conservation

The conservation of momentum of the solid phase can be expressed as

$$(1-n)\rho_S \frac{d\mathbf{v}_S}{dt} - \nabla \cdot \boldsymbol{\sigma}' - (1-n)\nabla p_L - (1-n)\rho_S \mathbf{g} - \frac{n^2 \rho_L g}{k}(\mathbf{v}_L - \mathbf{v}_S) = 0 \tag{3.24}$$

where k is the isotropic Darcy permeability. It can be expressed in terms of the intrinsic permeability κ and the dynamic viscosity of the liquid μ_d as:

$$k = \kappa \frac{\rho_L g}{\mu_d} \tag{3.25}$$

The conservation of momentum of the liquid phase is written as:

$$n\rho_L \frac{d\mathbf{v}_L}{dt} - n\nabla p_L - n\rho_L \mathbf{g} + \frac{n^2 \rho_L g}{k}(\mathbf{v}_L - \mathbf{v}_S) = 0 \tag{3.26}$$

the term $(\mathbf{v}_L - \mathbf{v}_S)$ represents the relative velocity of the liquid respect to the solid.

Adding the momentum of the solid phase (Eq. 3.24) to the momentum of the liquid phase (Eq. 3.26), the momentum conservation for the mixture can be written as:

$$(1-n)\rho_S \frac{d\mathbf{v}_L}{dt} + n\rho_L \frac{d\mathbf{v}_L}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho_{sat} \mathbf{g} \tag{3.27}$$

Summarizing, the two-phase problem is described by two momentum equations, i.e. 3.26 for the liquid and 3.27 for the mixture, the storage Eq. 3.23, and the constitutive equation for the soil skeleton. These equations are derived neglecting the spatial variation of densities and porosity, assuming incompressible soil grains, and assuming the validity of the Darcy's law.

3.4.1.3 Boundary conditions

The proposed formulation requires that the boundary of the domain is the union of the following components

$$\partial\Omega = \partial\Omega_u \cup \partial\Omega_\sigma = \partial\Omega_{v_L} \cup \partial\Omega_p \tag{3.28}$$

where $\partial\Omega_{v_L}$ and $\partial\Omega_p$ are the prescribed velocity and prescribed pressure boundaries of the liquid phase, respectively, whereas $\partial\Omega_u$ is the prescribed displacement (velocity) boundary of the solid phase and $\partial\Omega_\sigma$ is the prescribed total stress boundary.

The following conditions should also be satisfied at the boundary

$$\partial\Omega_u \cap \partial\Omega_\sigma = \emptyset \quad \text{and} \quad \partial\Omega_{v_L} \cap \partial\Omega_p = \emptyset \tag{3.29}$$

3.4.2 Weak form

Before the discretization, the strong form of the governing equations has to be transformed in the weak form. This is achieved by multiplying Equations 3.26 and 3.27 by weighting function $\mathbf{t}$ and integrating over the current domain Ω:

$$\int_\Omega \delta\mathbf{t}\rho_L \frac{d\mathbf{v}_L}{dt} d\Omega = \int_\Omega \delta\mathbf{t}\nabla p_L d\Omega + \int_\Omega \delta\mathbf{t}\rho_L \mathbf{g} d\Omega - \int_\Omega \delta\mathbf{t}\frac{n\rho_L g}{k}(\mathbf{v}_L - \mathbf{v}_S) d\Omega \tag{3.30}$$

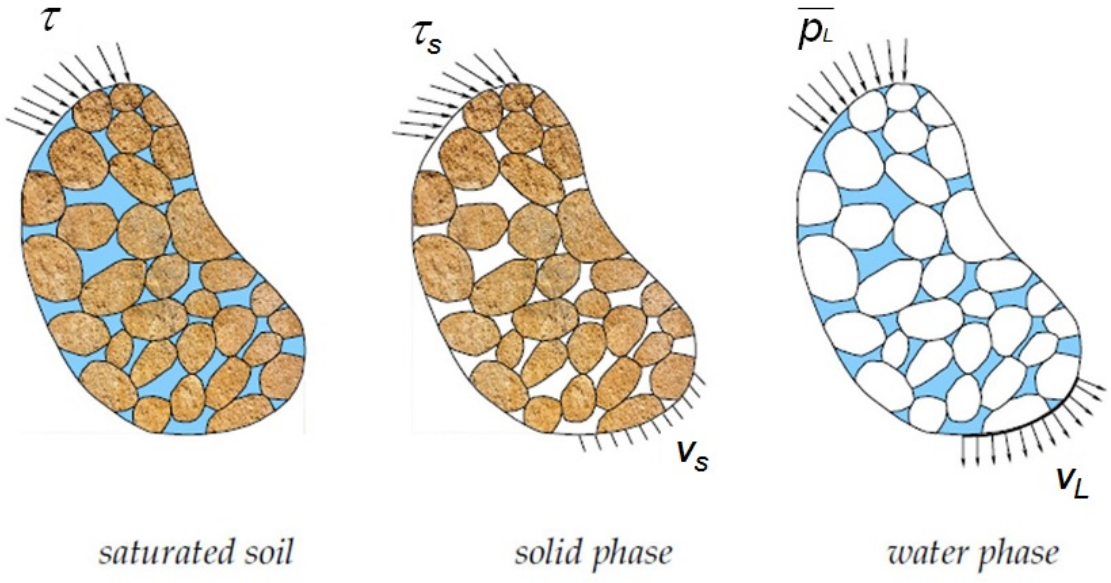


Figure 3.2: *Displacement and traction boundary conditions for two phase problem (Al-Kafaji 2013) [10].*

$$\int_\Omega \delta\mathbf{t}(1-\mathrm{n})\rho_S \frac{\mathrm{d}\mathbf{v}_S}{\mathrm{dt}}\mathrm{d}\Omega = \int_\Omega \delta\mathbf{t}\nabla\cdot\boldsymbol{\sigma}'\mathrm{d}\Omega + \int_\Omega \delta\mathbf{t}\rho_{\mathrm{sat}}\mathbf{g}\mathrm{d}\Omega - \int_\Omega \delta\mathbf{t}\mathrm{n}\rho_L \frac{\mathrm{d}\mathbf{v}_L}{\mathrm{dt}}\mathrm{d}\Omega \tag{3.31}$$

Applying the divergence theorem and the traction boundary conditions, the final weak forms are:

$$\int_\Omega \delta\mathbf{t}\rho_L \frac{\mathrm{d}\mathbf{v}_L}{\mathrm{dt}}\mathrm{d}\Omega = \int_{\partial\Omega_p} \delta\mathbf{t}\bar{\mathrm{p}}_L\mathrm{dS} - \int_\Omega \nabla\delta\mathbf{t}\mathrm{p}_L\mathrm{d}\Omega + \int_\Omega \delta\mathbf{t}\rho_L\mathbf{g}\mathrm{d}\Omega - \int_\Omega \delta\mathbf{t}\frac{\mathrm{n}\rho_L\mathrm{g}}{\mathrm{k}}(\mathbf{v}_L-\mathbf{v}_S)\mathrm{d}\Omega \tag{3.32}$$

$$\int_\Omega \delta\mathbf{t}(1-\mathrm{n})\rho_S \frac{\mathrm{d}\mathbf{v}_S}{\mathrm{dt}}\mathrm{d}\Omega = + \int_{\partial\Omega_\sigma} \delta\mathbf{t}\boldsymbol{\tau}\mathrm{dS} - \int_\Omega \nabla\delta\mathbf{t}\boldsymbol{\sigma}\mathrm{d}\Omega + \int_\Omega \delta\mathbf{t}\rho_{\mathrm{sat}}\mathbf{g}\mathrm{d}\Omega - \int_\Omega \delta\mathbf{t}\mathrm{n}\rho_L \frac{\mathrm{d}\mathbf{v}_L}{\mathrm{dt}}\mathrm{d}\Omega \tag{3.33}$$

The left-hand side terms in Equations 3.32 and 3.33 represent the inertia. In the right-hand side, the first terms represent the external load applied at the boundary, the second terms represent the internal force, and the third terms represent the gravity load. The last term in Equation 3.32 is the drag force. The last term in Equation 3.33 is the liquid inertia, where the porosity is taken into account.

3.5 Two-phase Single-point formulation for unsaturated soil

The two-phase formulation for unsaturated soil implemented in Anura3D refers to the work by [11]. It is derived considering dynamic momentum balance of the liquid phase, dynamic momentum balance of the mixture, and mass balances and constitutive relationships of solid

and liquid phases, for this reason it can be considered a two-phase formulation and it is an extension of the formulation presented in Section 3.4. All dynamic terms are taken into account; acceleration of the solid skeleton $\mathbf{a}_S$ and acceleration of the pore liquid $\mathbf{a}_L$ are the primary unknowns. Unsaturated conditions are accounted for considering that the liquid phase does not entirely fill the voids.

3.5.1 Governing equations

3.5.1.1 Mass conservation

When considering incompressible solid grains and disregarding the spatial variations in density and porosity, the expressions for the conservation of mass of the solid and the liquid reduce to Eqs. 3.34 and 3.35 respectively.

$$\frac{dn_S}{dt} + n_S \text{div}(\mathbf{v}_S) = 0 \tag{3.34}$$

$$\frac{d\left(\rho_L n_L\right)}{dt} = (\mathbf{v}_S - \mathbf{v}_L)\nabla(n_L \rho_L) - n_L \rho_L \text{div}(\mathbf{v}_L) \tag{3.35}$$

where n_L and n_S are the liquid and solid volumetric fraction, respectively.

Including Eq. 3.34 into Eq. 3.35, taking into account the definitions of liquid volumetric concentration ratios in terms of porosity and degree of saturation, $n_L = S_L n$, $n_S = 1 - n$, and rearranging terms gives Eq. 3.36

$$n\frac{d\left(\rho_L S_L\right)}{dt} = \text{div}\left[\rho_L n S_L(\mathbf{v}_S - \mathbf{v}_L)\right] - \rho_L S_L \text{div}(\mathbf{v}_S) \tag{3.36}$$

Finally, the material derivative in Eq. 3.36 is solved assuming liquid pressure as state variable, which yields to Eq. 3.37.

$$n\left(S_L \frac{\partial \rho_L}{\partial p_L} + \rho_L \frac{\partial S_L}{\partial p_L}\right)\frac{dp_L}{dt} = \text{div}\left[\rho_L n S_L(\mathbf{v}_S - \mathbf{v}_L)\right] - \rho_L S_L \text{div}(\mathbf{v}_S) \tag{3.37}$$

The derivative of liquid density with respect to pressure is given by the state equation of liquid; while the derivative of the degree of saturation is given by the soil-water retention curve (SWRC) (see Sec. 3.5.1.2).

3.5.1.2 Hydraulic constitutive equation

The relationship between pore liquid pressure and degree of saturation or liquid content is essential to model the behaviour of unsaturated soil. This is given by the SWRC which can assume different analytical expressions. In Anura3D, two alternative relationships are used: (i) a linear relation (Eq. 3.38) where a_v is a constant parameter, S_{min} is the residual degree of saturation and S_{max} is the maximum degree of saturation, and (ii) the Van Genuchten relationship (Eq. 3.39) where p_{ref}, λ, are fitting parameters.

$$S_L = S_{min} + (S_{max} - S_{min}) {p_L}^{a_v} \tag{3.38}$$

$$S_L = S_{min} + (S_{max} - S_{min}) \left[1 + \left(\frac{p_L}{p_{ref}} \right)^{\frac{1}{1-\lambda}} \right]^{-\lambda} \tag{3.39}$$

Partial saturation modifies the soil hydraulic conductivity; in general partially saturated soils are less permeable than fully saturated soils. The ratio between actual hydraulic conductivity and saturated hydraulic conductivity k/k_{sat} is a function of the degree of saturation as given by the hydraulic conductivity curve (HCC). A number of relationships have been proposed in the literature, the functions proposed by Hillel [12] (Eq. 3.40) and Mualem [13] (Eq. 3.41) are implemented in the applied software, in which r and λ are fitting parameters.

$$\frac{k}{k_{sat}} = S_L^r \tag{3.40}$$

$$\frac{k}{k_{sat}} = \sqrt{S_L} \left[1 - \left(1 - S_L^{\frac{1}{\lambda}} \right)^{\lambda} \right]^2 \tag{3.41}$$

3.5.1.3 Momentum conservation

The momentum balance of the liquid phase per unit of liquid volume is given in Eq. 3.42,

$$\rho_L \mathbf{a}_L = \nabla p_L - \mathbf{f}_L^d + \rho_L \mathbf{g} \tag{3.42}$$

where ρ_L is the liquid density, $\mathbf{f}_L^d$ is the drag force which accounts for solid-fluid interaction, p_L is the liquid pressure and $\mathbf{g}$ is the gravity vector.

The flow is considered laminar and stationary in the slow velocity regime. Hence, the drag force is governed by Darcy's law (Eq. 3.43),

$$\mathbf{f}_L^d = \frac{n_L \mu_L}{\kappa_L} (\mathbf{v}_L - \mathbf{v}_S) \tag{3.43}$$

where μ_L is the dynamic viscosity of the liquid, κ_L is the liquid intrinsic permeability, n_L is the liquid volumetric fraction and $\mathbf{v}_L$, $\mathbf{v}_S$ are the absolute liquid and solid velocities respectively.

The isotropic intrinsic permeability of the liquid κ_L can also be expressed in terms of Darcy permeability, or hydraulic conductivity, k_L (Eq. 3.44).

$$\kappa_L = k_L \frac{\mu_L}{\rho_L g} \tag{3.44}$$

The mixture dynamic momentum conservation can be written as Eq. 3.45,

$$n_S \rho_S \mathbf{a}_S + n_L \rho_L \mathbf{a}_L = \text{div}(\boldsymbol{\sigma}) + \rho_m \mathbf{g} \tag{3.45}$$

where ρ_S is the solid grain density, n_S is the volumetric concentration ratio of solid, and $\rho_m = n_S \rho_S + n_L \rho_L$ is the density of the mixture. Note that $n_S = 1 - n$ and $n_L = S_L n$, where n is the porosity of the solid skeleton and S_L is the degree of saturation. $\boldsymbol{\sigma}$ is the total stress tensor, which can be computed with the Bishop's effective stress equation for unsaturated soils and has the form of Eq. 3.46, where χ is an effective stress parameter, here assumed equal to S_L, and $\mathbf{m}$ is the unit vector, equal to $(1\ 1\ 1\ 0\ 0\ 0)^T$ in 3D. In this paper, stresses and pressures are positive for tension, thus suction is equal to p_L.

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' + \chi p_L \mathbf{m} \tag{3.46}$$

3.5.1.4 Boundary conditions

The proposed formulation requires the definition of the following boundary conditions (BC):

1. Prescribed liquid displacement or velocity on $\partial\Omega_{v_L}$ and prescribed pressure on $\partial\Omega_p$, where $\partial\Omega = \partial\Omega_{v_L} \cup \partial\Omega_p$ and $\partial\Omega_{v_L} \cap \partial\Omega_p = 0$
2. Prescribed solid displacement or velocity on $\partial\Omega_u$ and prescribed traction on $\partial\Omega_\tau$, where $\partial\Omega = \partial\Omega_{v_S} \cup \partial\Omega_\tau$ and $\partial\Omega_{v_S} \cap \partial\Omega_\tau = 0$.

Essential boundary conditions on $\partial\Omega_{v_L}$ and $\partial\Omega_{v_S}$ are imposed on the nodes of the computational grid. Natural boundary conditions on $\partial\Omega_p$ and $\partial\Omega_\tau$ are included in the weak form of the momentum balance equations.

In typical problems with partially saturated soil in geomechanics, prescribed liquid velocity can be applied on infiltration boundaries, and prescribed pressures can be applied either by defining a pressure load $\widehat{p}_L$ or by assigning a total hydraulic head $\widehat{H}$. Assuming the validity of Bernoulli's equation and neglecting the kinematic head, the total hydraulic head can be written as

$$\widehat{H} = h_g - \frac{\widehat{p}_L}{\rho_L g} \tag{3.47}$$

where h_g stands for the potential head or geometric head and $\widehat{p}_L/(\rho_L g)$ is the pressure head. The minus sign in Eq. 5.5 is introduced because pressure is assumed negative for compression.

Sometimes, the boundary condition is part of the problem solution, i.e. it is not known a priori if the boundary belongs to $\partial\Omega_{v_L}$ or $\partial\Omega_p$. This is typical of free surface flows across porous media along the so-called *potential seepage face*.

Figure 3.3 represents schematically how these BCs simulate different hydraulic loading acting on a levee. The implementation details of these boundary conditions can be found in 5.3.

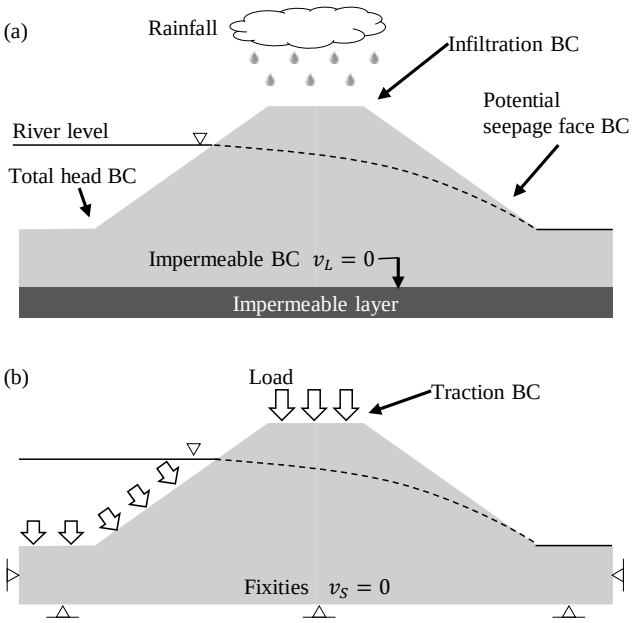


Figure 3.3: *Typical boundary conditions for liquid phase (a) and mixture (b), from [11].*

3.5.2 Weak form

The governing equations (liquid and mixture momentum balance) are transformed in the weak form by multipling by a weighting function **w** and integrating over the domain Ω

$$\int_V \rho_L \mathbf{a}_L \cdot \delta \mathbf{t} dV = \int_V \left[\nabla p_L + \rho_L \mathbf{g} - \frac{n_L \mu_L}{\kappa_L} (\mathbf{v}_L - \mathbf{v}_S) \right] \cdot \delta \mathbf{t} dV \tag{3.48}$$

The application of the divergence theorem to Eq. 3.48 results in Eq. 3.50

$$\int_V \rho_L \mathbf{a}_L \cdot \delta \mathbf{t} dV = \int \_\partial\Omega_p \widehat{\mathbf{p}}_L d\partial\Omega_p - \int_V p_L \nabla \cdot \delta \mathbf{t} dV + \int_V \rho_L \mathbf{g} \cdot \delta \mathbf{t} dV - \tag{3.49}$$

$$\int_V \left[\frac{n_L \mu_L}{\kappa_L} (\mathbf{v}_L - \mathbf{v}_S) \right] \cdot \delta \mathbf{t} dV \tag{3.50}$$

The mixture momentum balance equation in the weak form corresponds to Eq. 3.51

$$\int_V [(1-n)\rho_S \mathbf{a}_S] \cdot \mathbf{u} dV = \int_V \left[-n_L \rho_L \mathbf{a}_L + \nabla \sigma_m + ((1-n)\rho_S + n_L \rho_L)\mathbf{g} \right] \cdot \delta \mathbf{t} dV \tag{3.51}$$

The application of the divergence theorem to Eq. 3.51 results in Eq. 3.53

$$\int_V [(1-n)\rho_S \mathbf{a}_S] \cdot \delta \mathbf{t} dV = \int_V \left[-(n_L \rho_L \mathbf{a}_L) \cdot \delta \mathbf{t} dV + \int_{\partial\Omega_t} \boldsymbol{\tau} d\delta \mathbf{t} \partial\Omega_t \right] \tag{3.52}$$

$$- \int_V \sigma_m : (\nabla \delta \mathbf{t}) dV + \int_V [((1-n)\rho_S + n_L \rho_L)\mathbf{g}] \cdot \delta \mathbf{t} dV \tag{3.53}$$

Where $\widehat{\mathbf{p}}_L$ is the prescribed liquid pressure in the boundary $\partial\Omega_p$ and $\boldsymbol{\tau}$ is the prescribed traction vector in the boundary $\partial\Omega_\tau$.

4 Explicit-Dynamic Formulation

4.1 Introduction

This chapter presents the computation scheme adopted in the dynamic explicit implementation of Anura3D. It explains how the system of solving equations of the problem is assembled within the MPM framework and the main computational steps of the MPM solution procedure.

4.2 Notations and variables

The strain and stress tensors can be represented in vector form, taking advantage of the symmetry of the strain and stress tensors

$$\boldsymbol{\varepsilon}(\mathbf{x}, t) = [\varepsilon_{11}\ \varepsilon_{22}\ \varepsilon_{33}\ 2\varepsilon_{12}\ 2\varepsilon_{23}\ 2\varepsilon_{31}]^T \tag{4.1}$$

$$\boldsymbol{\sigma}(\mathbf{x}, t) = [\sigma_{11}\ \sigma_{22}\ \sigma_{33}\ \sigma_{12}\ \sigma_{23}\ \sigma_{31}]^T \tag{4.2}$$

where the first three terms in the vectors (σ_{ij} and ε_{ij} with i = j) are the normal components along the axis x_i of the coordinate system and the other three represent the shear terms acting on the $x_i x_j$ planes. This way to represent the stress and strain is known in the literature as *Voigt* notation and is very useful as it reduces the order of the symmetric tensors.

It should be realized 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.

$$\varepsilon_{ij}\sigma_{ij} = \varepsilon^T\boldsymbol{\sigma} \tag{4.3}$$

4.3 One-phase Single-point formulation for solid analysis

4.3.1 Discretized momentum balance equation

The weak form of the linear momentum equation over the domain Ω is given by

$$\int_{\Omega} \delta\mathbf{v}\rho\frac{d\mathbf{v}}{dt}d\Omega = \int_{\partial\Omega_\sigma} \delta\mathbf{v}\ \boldsymbol{\tau}\ d\partial\Omega - \int_{\Omega} (\nabla\cdot\delta\mathbf{v})\boldsymbol{\sigma}\ d\Omega + \int_{\Omega} \delta\mathbf{v}\rho\mathbf{g}d\Omega \tag{4.4}$$

The domain Ω is decomposed into finite subdomains Ω_{el} called finite elements. The union of these subdomains comprises the total domain, $\Omega = \cup_{el=1}^{nelm}\Omega_{el}$, where nelm denotes the total number of finite elements in the mesh. Each element is jointed with its surrounding elements by 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.

In finite element method, the discrete form is obtained by approximating the displacement and

eventually the velocity and acceleration as

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) &\approx \bar{\mathbf{N}}(\mathbf{x})\hat{\mathbf{u}}(t)\\ \mathbf{v}(\mathbf{x},t) &\approx \bar{\mathbf{N}}(\mathbf{x})\hat{\mathbf{v}}(t)\\ \mathbf{a}(\mathbf{x},t) &\approx \bar{\mathbf{N}}(\mathbf{x})\hat{\mathbf{a}}(t)\end{aligned} \tag{4.5}$$

respectively. The corresponding virtual quantities are approximated the same way, for instance,

$$\delta\mathbf{u}(\mathbf{x},t) \approx \bar{\mathbf{N}}(\mathbf{x})\delta\hat{\mathbf{u}}(t) \tag{4.6}$$

$$\delta\mathbf{v}(\mathbf{x},t) \approx \bar{\mathbf{N}}(\mathbf{x})\delta\hat{\mathbf{v}}(t) \tag{4.7}$$

$$\delta\mathbf{a}(\mathbf{x},t) \approx \bar{\mathbf{N}}(\mathbf{x})\delta\hat{\mathbf{a}}(t) \tag{4.8}$$

The interpolation function or the shape function matrix $\bar{\mathbf{N}}$ has the following form

$$\bar{\mathbf{N}}(\mathbf{x}) = \left[\bar{\mathbf{N}}_1(\mathbf{x})\ \bar{\mathbf{N}}_2(\mathbf{x})\ \ ...\ \ ...\ \ \bar{\mathbf{N}}_{nT}(\mathbf{x})\right] \tag{4.9}$$

with

$$\bar{\mathbf{N}}_i(\mathbf{x}) = \begin{bmatrix}\bar{N}_i(x) & 0 & 0\\ 0 & \bar{N}_i(x) & 0\\ 0 & 0 & \bar{N}_i(x)\end{bmatrix} \tag{4.10}$$

with nT 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. The vectors of nodal displacements, velocities and accelerations are denoted as $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$, respectively. These vectors can be written as follows

$$\hat{\mathbf{u}}(t) = [\hat{u}_{11}\ \ \hat{u}_{12}\ \ \hat{u}_{13}\ \ ...\ \ ...\ \ ...\ \ \hat{u}_{n_T1}\ \ \hat{u}_{n_T2}\ \ \hat{u}_{n_T3}]^T \tag{4.11}$$

$$\hat{\mathbf{v}}(t) = [\hat{v}_{11}\ \ \hat{v}_{12}\ \ \hat{v}_{13}\ \ ...\ \ ...\ \ ...\ \ \hat{v}_{n_T1}\ \ \hat{v}_{n_T2}\ \ \hat{v}_{n_T3}]^T \tag{4.12}$$

$$\hat{\mathbf{a}}(t) = [\hat{a}_{11}\ \ \hat{a}_{12}\ \ \hat{a}_{13}\ \ ...\ \ ...\ \ ...\ \ \hat{a}_{n_T1}\ \ \hat{a}_{n_T2}\ \ \hat{a}_{n_T3}]^T \tag{4.13}$$

where v_{12} indicates the velocity at node 1 in the direction of coordinate x_2.

The kinematic relation can be written using matrix notation as

$$\dot{\boldsymbol{\varepsilon}}(\mathbf{x},t) = \mathbf{L}\mathbf{v}(\mathbf{x},t) \tag{4.14}$$

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

$$\mathbf{L} = \begin{bmatrix}\frac{\partial}{\partial x_1} & 0 & 0\\ 0 & \frac{\partial}{\partial x_2} & 0\\ 0 & 0 & \frac{\partial}{\partial x_3}\\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0\\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2}\\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1}\end{bmatrix} \tag{4.15}$$

Substituting the second of Equation (4.3.1) into Equation (4.14) yields

$$\dot{\boldsymbol{\varepsilon}}(\mathbf{x},t) = \mathbf{L}\mathbf{N}(\mathbf{x})\hat{\mathbf{v}}(t) = \mathbf{B}(\mathbf{x})\hat{\mathbf{v}}(t) \tag{4.16}$$

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

$$\mathbf{B}(\mathbf{x}) = [\mathbf{B}_1(\mathbf{x}) \;\; \mathbf{B}_2(\mathbf{x}) \;\; \mathbf{B}_3(\mathbf{x}) \;\; ... \;\; ... \;\; ... \;\; \mathbf{B}_{nT}(\mathbf{x})] \tag{4.17}$$

with

$$\mathbf{B}_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_3} \\ \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_2} & \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_1} & 0 \\ 0 & \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_3} & \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_2} \\ \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_3} & 0 & \frac{\partial \bar{N}_i(\mathbf{x})}{\partial x_1} \end{bmatrix} \tag{4.18}$$

The weak form yields as follow:

$$\begin{aligned} &\delta\hat{\mathbf{v}}^T \int_{\partial\Omega} \bar{\mathbf{N}}^T \mathbf{t} \cdot \mathbf{n} \, d\partial\Omega - \delta\hat{\mathbf{v}}^T \int_\Omega B^T \boldsymbol{\sigma} \, d\Omega + \\ &+\delta\hat{\mathbf{v}}^T \int_\Omega \bar{\mathbf{N}}^T \rho \mathbf{g} \, d\Omega - \delta\hat{\mathbf{v}}^T \left[\int_\Omega \bar{\mathbf{N}}^T \rho \bar{\mathbf{N}} \, d\Omega \right] \hat{\mathbf{a}} = \mathbf{0} \end{aligned} \tag{4.19}$$

where $\mathbf{n}$ is the unit vector which is normal to the boundary of the domain.

The discretized form yields:

$$\mathbf{f}^{ext} - \mathbf{f}^{int} = \mathbf{M}\hat{\mathbf{a}} \tag{4.20}$$

with

$$\mathbf{f}^{ext} = \int_{\partial\Omega} \bar{\mathbf{N}}^T \mathbf{t} \cdot \mathbf{n} \, d\partial\Omega + \int_\Omega \rho \bar{\mathbf{N}}^T \mathbf{g} \, d\Omega \tag{4.21}$$

$$\mathbf{f}^{int} = \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}' \, d\Omega \tag{4.22}$$

$$\mathbf{M} = \int_\Omega \rho \bar{\mathbf{N}}^T \bar{\mathbf{N}} d\Omega \tag{4.23}$$

where $\mathbf{f}^{ext}$ is the vector of nodal external forces, $\mathbf{f}^{int}$ is the vector of nodal internal forces and $\mathbf{M}$ is the nodal mass matrix.

4.3.2 Initialization of material points

The MP carry all the information of the continuum. In this section, we discuss the initialization of MP within the background mesh. This includes association of mass, body forces, tractions and other properties of the continuum to MP. Elements filled with MP are called active elements and their nodes contribute to the solving system of equations; on the contrary the empty elements, i.e. those without any MP, are ignored thus reducing the computational cost.

Let us consider a single tetrahedral element to explain the full procedure of initialization of MP information. Each MP is initially positioned at a predefined local position inside the parent

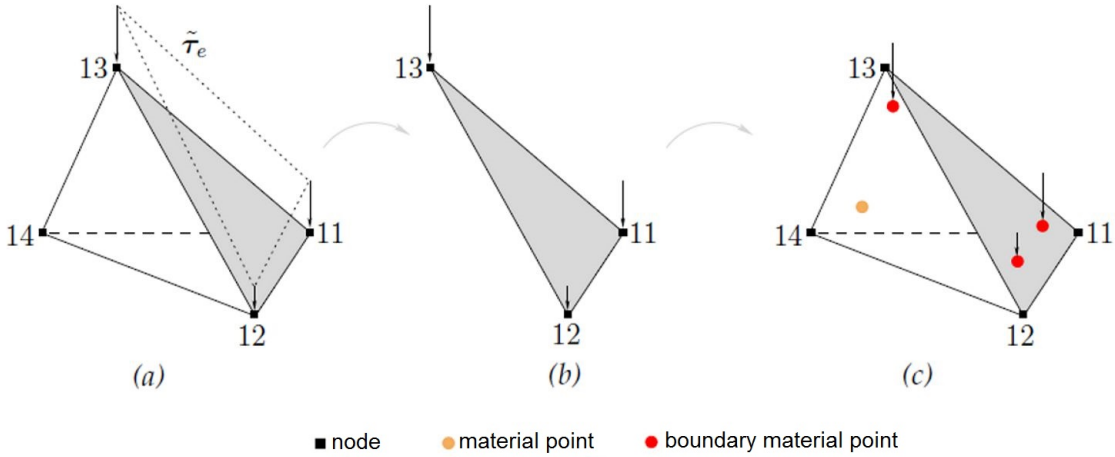


Figure 4.1: *Initialization of surface traction. (a) tetrahedral element (b) triangular element (c) traction is mapped to the boundary MP, after [10].*

element, and hence the local position vector $\boldsymbol{\xi}_{MP}$ of the material point MP is initialized. The global position vector $\mathbf{x}_{MP}$ is then obtained as

$$\mathbf{x}_{MP}(\boldsymbol{\xi}_{MP}) \approx \sum_{i=1}^{no_{nodes}} N_i(\boldsymbol{\xi}_{MP}^t)\mathbf{x}_i \tag{4.24}$$

in which no_{nodes} denotes the number of nodes per element, $N_i(\boldsymbol{\xi}_{MP})$ is the shape function of node i evaluated at the local position of material point MP and $\mathbf{x}_i$ are the nodal coordinates.

Volumes associated with MP are calculated so that all the MP inside the element have initially the same portion of the element volume, i.e.

$$\Omega_{MP} = \frac{1}{no_{MP,el}} \int_{\Omega_e} d\Omega \approx \frac{1}{no_{MP,el}} \sum_{q=1}^{no_{q,el}} w_{MP} \, |\mathbf{J}(\boldsymbol{\xi}_{MP})| \tag{4.25}$$

where Ω_{MP} is the volume associated with the material point MP, $no_{MP,el}$ denotes the number of MP in the element, $no_{q,el}$ is the number of Gauss points in the element, w_{MP} is the local integration weight associated with Gauss point MP, and $\mathbf{J}$ is the Jacobian matrix. This implies that, at the beginning of the calculation, all the active elements are assumed to be fully filled by the continuum body. An element is said to be partially filled if the sum of the volumes of the contained MP is less than the element volume.

The mass m_{MP} is then calculated as

$$m_{MP} = \Omega_{MP} \rho_{MP} \tag{4.26}$$

with ρ being the mass density of the material to which the material point MP belongs.

The gravity force $\mathbf{f}_{MP}^{grav}$ is simply calculated using the mass of the MP and the vector of gravitational acceleration $\mathbf{g}$ as

$$\mathbf{f}_{MP}^{grav} = m_{MP}\mathbf{g} \tag{4.27}$$

The external forces applied at the traction boundary are mapped to the MP located next to the element border, also called boundary MP (Fig. 4.1).

These MP carry surface traction throughout the computations. Considering a tetrahedral element, the traction vector $\boldsymbol{\tau}_e$ applied at the triangular surface is interpolated from the nodes of this surface to the boundary MP. Hence, the traction at boundary material point MP is

$$\boldsymbol{\tau}_e(\mathbf{x}_p) \approx \sum_{i=1}^{n_{tri}} N_i(\boldsymbol{\xi}_q)\boldsymbol{\tau}_e(\mathbf{x}_i) \tag{4.28}$$

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

$$\mathbf{f}_p^{trac} = \boldsymbol{\tau}_e(\mathbf{x}_{MP})\frac{S_e}{n_{ebMP}} = \frac{S_e}{n_{ebMP}}\sum_{i=1}^{n_{tri}} N_i(\boldsymbol{\xi}_{MP})\boldsymbol{\tau}_e(\mathbf{x}_i) \tag{4.29}$$

in which n_{ebMP} denotes the number of boundary MP located next to the loaded surface and S_e is the area of the corresponding loaded surface of the element e.

In the initialization of MP, 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.

4.3.3 Calculation of internal forces

In finite element formulation the numerical integration of all integrals above is not performed along the global coordinate system. Usually, it is more convenient to convert the global coordinates ($\mathbf{x}$) of each element into a reference (or *parent*) element system ($\boldsymbol{\xi}$), and the mapping procedure is done using the *Jacobian matrix*. In this work, a unit 4-node tetrahedron with linear shape functions is chosen to be the reference element. Details about the mapping procedure are described in Section 4.8. The internal force becomes

$$\mathbf{f}^{int} = \sum_{el=1}^{no_{elm}} \int_{\Omega_{el}} \mathbf{B}^T\boldsymbol{\sigma}\, d\Omega = \sum_{el=1}^{no_{elm}} \int_{\square} \mathbf{B}^T\boldsymbol{\sigma}|J|\, d\square \tag{4.30}$$

where Ω_{el} is the volume of the element el in the global coordinate system, no_{elm} is the number of active elements, $\square$ is volume of the parent element, $d\square = d\xi_1 d\xi_2 d\xi_3$ is the infinitesimal volume in the parent element system, $|J|$ is the determinant of the Jacobian, and $\mathbf{B}$ is the matrix of the shape function gradients calculated at location $\boldsymbol{\xi}$ with respect to the parent coordinate system. If numerical integration is applied, the equations yield as follows

$$\mathbf{f}^{int} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{k=1}^{no_{int,el}} \mathbf{B}_i^T(\boldsymbol{\xi}_k)\boldsymbol{\sigma}_k|J_k|W_k \tag{4.31}$$

where $no_{Nodes,el}$ and $no_{int,el}$ are respectively the number of nodes and integration points inside the element el, and W_k is the weight factor (or integration weight) of the integration point k.

In the material point method the MP carry all the information of the continuum, including the stresses. The internal forces are computed in the following form

$$\mathbf{f}^{\text{int}} = \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \sum_{\text{i}=1}^{\text{no}_{\text{Nodes,el}}} \sum_{\text{MP}=1}^{\text{no}_{\text{MP,el}}} \mathbf{B}_{\text{i}}^{\text{T}}(\boldsymbol{\xi}_{MP})\boldsymbol{\sigma}_{\text{MP}}\Omega_{\text{MP}} \tag{4.32}$$

where $\text{no}_{\text{MP,el}}$ is the number of material points within element el.

4.3.4 Calculation of external forces

The external forces can be subdivided in two parts. The body force, due to the gravity acceleration, and the external loading due to distributed forces applied to the body.

The body force can be computed in the following way

$$\mathbf{f}^{\text{ext,grav}} = \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \int_{\Omega_{\text{el}}} \rho\overline{\mathbf{N}}^{\text{T}}\mathbf{g}\, \text{d}\Omega = \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \int_{\square} \rho\mathbf{N}^{\text{T}}\mathbf{g}|\text{J}|\, \text{d}\square \tag{4.33}$$

where Ω_{el} is the volume of the element el in the global coordinate system, no_{elm} is the number of active elements, $\square$ is volume of the parent element, $\text{d}\square = \text{d}\xi_1\text{d}\xi_2\text{d}\xi_3$ is the infinitesimal volume in the parent element system, $|\text{J}|$ is the determinant of the Jacobian, and $\mathbf{N}$ is the shape function matrix calculated at location $\boldsymbol{\xi}$ with respect to the parent coordinate system. Applying the material point integration, the equation yield as follows

$$\mathbf{f}^{\text{ext,grav}} = \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \sum_{\text{i}=1}^{\text{no}_{\text{Nodes,el}}} \sum_{\text{MP}=1}^{\text{no}_{\text{MP,el}}} \text{m}_{\text{MP}}\mathbf{N}_{\text{i}}^{\text{T}}(\boldsymbol{\xi}_{MP})\mathbf{g}\Omega_{\text{MP}} \tag{4.34}$$

where $\text{no}_{\text{Nodes,el}}$ and $\text{no}_{\text{MP,el}}$ are respectively the number of nodes and material points inside the element el.

The external forces due to traction are computed in the following way

$$\mathbf{f}^{\text{ext,trac}} = \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \int_{\partial\Omega_{\text{el}}} \overline{\mathbf{N}}^{\text{T}}\mathbf{t}\cdot\mathbf{n}\, \text{d}\partial\Omega = \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \sum_{\text{i}=1}^{\text{no}_{\text{Nodes,el}}} \sum_{\text{MP}=1}^{\text{no}_{\text{MP,el}}} \mathbf{N}^{\text{T}}(\boldsymbol{\xi}_{MP})\, \mathbf{f}_{\text{MP}}^{\text{ext}} \tag{4.35}$$

where $\mathbf{f}_{\text{MP}}^{\text{ext}}$ is the force stored in the material point due to distributed external forces applied at the boundary of the body, as described in Section 4.3.2.

The complete form of the external force is summarized below:

$$\mathbf{f}^{\text{ext}} = \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \sum_{\text{i}=1}^{\text{no}_{\text{Nodes,el}}} \sum_{\text{MP}=1}^{\text{no}_{\text{MP,el}}} \mathbf{N}^{\text{T}}(\boldsymbol{\xi}_{MP})\, \mathbf{f}_{\text{MP}}^{\text{ext,trac}} + \sum_{\text{el}=1}^{\text{no}_{\text{elm}}} \sum_{\text{i}=1}^{\text{no}_{\text{Nodes,el}}} \sum_{\text{MP}=1}^{\text{no}_{\text{MP,el}}} \text{m}_{\text{MP}}\mathbf{N}_{\text{i}}^{\text{T}}(\boldsymbol{\xi}_{MP})\mathbf{g}\Omega_{\text{MP}} \tag{4.36}$$

4.3.5 Mass matrix

To solve the system of equations (Eq. 4.20), the mass matrix has to be inverted. In practice, to simplify computations, a lumped mass matrix may be used instead of the consistent mass matrix given by Eq. 4.23. The lumped mass matrix is a diagonal matrix in which each entry m_{i}

is obtained by summing over the corresponding row of the consistent mass matrix. Matrix inversions become trivial, although the result of using a lumped mass matrix is some dissipation of kinetic energy that has been quantified by [14]. Using the property $\sum_{j=1}^{N_n} N_j^p = 1$ it becomes

$$\mathbf{M}^{\text{lump}} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} \mathbf{N}^T(\xi_{MP})\, m_{MP} \tag{4.37}$$

Hereinafter, the superscript lump is removed from the lumped mass matrix $\mathbf{M}^{\text{lump}}$ to simplify the notation, and mass matrix will always refer to a lumped matrix.

In this study the 4-node tetrahedral element is used and the lumping procedure gives the following expression for the mass matrix **M**.

$$\mathbf{M} = \sum_{el=1}^{n_{elm}} \mathbf{M}_{el} \tag{4.38}$$

$$\mathbf{M}_{el} = \begin{bmatrix} \mathbf{m}_1 & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & \mathbf{m}_2 & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & ... & ... & \mathbf{m}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & \mathbf{m}_{\text{NodeEl}} \end{bmatrix} \tag{4.39}$$

$$\mathbf{m}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} ; \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.40}$$

$$m_i = \sum_{MP=1}^{no_{MP,el}} m_{MP} N_i(\xi_{MP}) \tag{4.41}$$

4.3.6 Time discretization

The time has been discretized into different instants (k). Considering Δt the time step size, the time discretization approach is $t^{k+1} = t^k + \Delta t$.

If the general system of equations (Eq. 4.20) is posed at time t^k, it can be rewritten as shown in Eq. 4.42, where the acceleration $\hat{\mathbf{a}}^k$ is the unknown.

$$\mathbf{M}^k \cdot \hat{\mathbf{a}}^k = \mathbf{f}^{\text{int}k} + \mathbf{f}^{\text{ext}k} \tag{4.42}$$

An explicit Euler time integration scheme is used to update the velocity (Eq. 4.43). This is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. Being $\hat{\mathbf{v}}^k$ the velocity at time t^k, the velocity at the next time step t^{k+1} is calculated using the acceleration at time t^k as follows.

$$\hat{\mathbf{v}}^{k+1} = \hat{\mathbf{v}}^k + \triangle t\, \hat{\mathbf{a}}^k \tag{4.43}$$

On the other hand, the displacements at time t^{k+1} are calculated using the updated velocity $\hat{\mathbf{v}}^{k+1}$ as indicated in Eq. 4.44.

$$\hat{\mathbf{u}}^{k+1} = \hat{\mathbf{u}}^k + \triangle t\, \hat{\mathbf{v}}^{k+1} \tag{4.44}$$

4.3.7 Solution algorithm for single time step

The algorithm presented here is based on the work presented by [3], which was improved respected the original one [2]. The key point is to work with momentum instead of velocity as much as possible, thus avoiding divisions by nodal masses. In each time step, the MPM computational cycle can be listed as follows.

1 The nodal mass is calculated using the shape functions and the lumped mass matrix at time t^k is formed (Eq. 4.37). The internal and external forces are evaluated in the nodes (Eqs.4.36 and 4.32).

2 The momentum balance equation (Eq. 4.42) is solved and the nodal accelerations $\hat{\mathbf{a}}_i^k$ are determined.

$$\hat{\mathbf{a}}_i^k = [\mathbf{M}_i^k]^{-1} \left(\mathbf{f}_i^{ext,k} - \mathbf{f}_i^{int,k}\right) \tag{4.45}$$

3 The velocity at the material points is updated considering Eq. 4.43 as

$$\mathbf{v}_{MP}^{k+1} = \mathbf{v}_{MP}^k + \Delta t \sum_{i=1}^{no_{Nodes}} N_i(\boldsymbol{\xi}_{MP}^k)\, \hat{\mathbf{a}}_i^k \tag{4.46}$$

4 Update nodal momentum

$$\mathbf{P}_i^{k+1} = \sum_{el=1}^{no_{el,i}} \sum_{MP=1}^{no_{MP,el}} m_{MP}\, N_j(\boldsymbol{\xi}_{MP}^k)\, \mathbf{v}_{MP}^{k+1} \tag{4.47}$$

5 Update nodal velocities

$$\hat{\mathbf{v}}_i^{k+1} = \frac{\mathbf{P}_i^{k+1}}{\mathbf{M}_i^k} \tag{4.48}$$

6 Compute incremental nodal displacement for solid and liquid constituent

$$\Delta\hat{\mathbf{u}}_i^{k+1} = \Delta t\, \hat{\mathbf{v}}_i^{k+1} \tag{4.49}$$

7 Compute strain increment

$$\Delta\varepsilon_{MP}^{k+1} = \mathbf{B}(\mathbf{x}_{MP})\Delta\hat{\mathbf{u}}_i{}^{k+1}$$

8 The stresses are updated using a material constitutive model

9 Update volume and density of MP

$$\Omega_{MP}^{k+1} = \Omega_{MP}^k(1 + \Delta\varepsilon_{vol,MP}) \qquad \rho_{MP}^{k+1} = \frac{\rho_{MP}^k}{(1 + \Delta\varepsilon_{vol,MP})} \tag{4.50}$$

10 Particle positions are updated considering Eq. 4.44 as

$$\mathbf{x}_{MP}^{k+1} = \mathbf{x}_{MP}^k + \sum_{i=1}^{no_{Nodes}} N_i(\boldsymbol{\xi}_{MP}^k)\, \Delta\hat{\mathbf{u}}_i^k \tag{4.51}$$

11 The computational grid is initialized for the next step, nodal values are discarded and the material points carry all the updated information.

4.3.8 Considerations on the use of one-phase solid analysis with porous media

Porous media are, in general, a mixture of three phases (solid, liquid and gas), which interact between each other, thus determining the mechanical response of the system. Taking rigorously into account these interactions may be in many cases unacceptably complicated, computationally expensive, and even unnecessary for engineering applications. For example, in numerical analyses dealing with drained and undrained conditions the presence of the pore liquid (e.g. water) can be considered in a simplified way. In the first case, the excess pore pressure is assumed to be zero, thus the presence of the liquid can be neglected and the solid skeleton (e.g. soil) can be regarded as one-phase even though it is saturated. In the latter, because of the negligible relative movement between solid and pore liquid, the equilibrium of the solid-liquid mixture can be considered rather than the equilibrium of solid skeleton and pore liquid as separate phases.

In undrained conditions, the stress state can be described in terms of total stresses or effective stresses. In the second case the excess pore pressures can be computed by means of the so-called Effective Stress Analysis, which is based on the assumption of strain compatibility between the solid skeleton and the enclosed liquid [15]. With the Effective Stress Analyses the pore pressure increment is computed as:

$$\Delta p = K_L \Delta \varepsilon_{vol} \tag{4.52}$$

where $\Delta \varepsilon_{vol}$ is the volumetric strain and K_L is the bulk modulus of the liquid.

Modeling of dry soil, and saturated soil in drained and undrained conditions are the field of applicability of the one-phase formulation. For partially drained conditions a fully-coupled two-phase formulation is necessary. This is presented in Section 4.5.

4.4 One-phase Single-point formulation for liquid analysis

4.4.1 Discretized momentum balance equation of liquid

The weak form can be written as follow:

$$\delta\hat{\mathbf{v}}_L^T \int_{\partial\Omega} \overline{\mathbf{N}}^T \mathbf{t}\cdot\mathbf{n}\, d\partial\Omega - \delta\hat{\mathbf{v}}_L^T \int_\Omega B^T \boldsymbol{\sigma}_L\, d\Omega + \\ +\delta\hat{\mathbf{v}}_L^T \int_\Omega \overline{\mathbf{N}}^T \rho_L \mathbf{g}\, d\Omega - \delta\hat{\mathbf{v}}_L^T \left[\int_\Omega \overline{\mathbf{N}}^T \rho_L \overline{\mathbf{N}}\, d\Omega\right] \dot{\hat{\mathbf{v}}}_L = \mathbf{0} \tag{4.53}$$

This equation has the same form as Eq. 4.3.1. More info about the derivation of the weak form of the momentum balance equation, the reader should refer to Sec.4.3.1.

The discretized form can be written in the following yields:

$$\mathbf{f}_L^{ext} - \mathbf{f}_L^{int} = \mathbf{M}_L \hat{\mathbf{a}} \tag{4.54}$$

with

$$\mathbf{f}_L^{ext} = \int_{\partial\Omega} \overline{\mathbf{N}}^T \mathbf{t}_L \cdot \mathbf{n}\, d\partial\Omega + \int_\Omega \rho_L \overline{\mathbf{N}}^T \mathbf{g}\, d\Omega \tag{4.55}$$

$$\mathbf{f}_L^{int} = \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}_L\, d\Omega \tag{4.56}$$

$$\mathbf{M}_L = \int_\Omega \rho_L \overline{\mathbf{N}}^T \overline{\mathbf{N}} d\Omega \tag{4.57}$$

where $\mathbf{f}_L^{ext}$ is the vector of nodal external forces, $\mathbf{f}_L^{int}$ is the vector of nodal internal forces and $\mathbf{M}_L$ is the nodal mass matrix.

4.4.2 Calculation of internal forces

In finite element formulation the numerical integration of all integrals above is usually performed using a reference (or *parent*) element system ($\boldsymbol{\xi}$), and the mapping procedure is done using the *Jacobian matrix*. In this manual, a unit 4-node tetrahedron with linear shape functions is chosen to be the reference element. Details about the calculation of the internal forces are discussed in Section 4.3.3 and details about the mapping procedure are described in Section 4.8.

The internal force vector becomes

$$\mathbf{f}_L^{int} = \sum_{el=1}^{no_{elm}} \int_{\Omega_{el}} \mathbf{B}^T \boldsymbol{\sigma}_L\, d\Omega = \sum_{el=1}^{no_{elm}} \int_\square \mathbf{B}^T \boldsymbol{\sigma}_L |J|\, d\square \tag{4.58}$$

where Ω_{el} is the volume of the element el in the global coordinate system, no_{elm} is the number of active elements, $\square$ is volume of the parent element, $d\square = d\xi_1 d\xi_2 d\xi_3$ is the infinitesimal volume in the parent element system, $|J|$ is the determinant of the Jacobian, and $\mathbf{B}$ is the matrix of shape function gradients calculated at location $\boldsymbol{\xi}$ with respect to the parent coordinate system.

In the material point method the MP carry all the information of the continuum, including the stresses. Thus, the internal forces are computed in the following form

$$\mathbf{f}_{\mathrm{L}}^{\mathrm{int}} = \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \sum_{\mathrm{i}=1}^{\mathrm{no_{Nodes,el}}} \sum_{\mathrm{MP}=1}^{\mathrm{no_{MP,el}}} \mathbf{B}_{\mathrm{i}}^{\mathrm{T}}(\boldsymbol{\xi}_{MP})\boldsymbol{\sigma}_{\mathrm{L,MP}}\Omega_{\mathrm{MP}} \tag{4.59}$$

where $\mathrm{no_{MP,el}}$ is the number of material points within element el.

4.4.3 Calculation of external forces

As for the one-phase-solid formulation, the external forces can be subdivided in two parts: the body force, which is generated by gravity acceleration, and the external loading due to distributed forces applied to the boundary of the liquid body.

The body force can be computed in the following way

$$\mathbf{f}_{\mathrm{L}}^{\mathrm{ext,grav}} = \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \int_{\Omega_{\mathrm{el}}} \rho_{\mathrm{L}} \bar{\mathbf{N}}^{\mathrm{T}} \mathbf{g} \, \mathrm{d}\Omega = \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \int_{\square} \rho \mathbf{N}^{\mathrm{T}} \mathbf{g} |\mathrm{J}| \, \mathrm{d}\square \tag{4.60}$$

where Ω_{el} is the volume of the element el in the global coordinate system, $\mathrm{no_{elm}}$ is the number of active elements, $\square$ is volume of the parent element, $\mathrm{d}\square = \mathrm{d}\xi_1\mathrm{d}\xi_2\mathrm{d}\xi_3$ is the infinitesimal volume in the parent element system, $|\mathrm{J}|$ is the determinant of the Jacobian, and $\mathbf{N}$ is the shape function matrix calculated at location $\boldsymbol{\xi}$ with respect to the parent coordinate system. Applying the material point integration, the equation yield as follows

$$\mathbf{f}_{\mathrm{L}}^{\mathrm{ext,grav}} = \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \sum_{\mathrm{i}=1}^{\mathrm{no_{Nodes,el}}} \sum_{\mathrm{MP}=1}^{\mathrm{no_{MP,el}}} \mathrm{m_{MP}} \mathbf{N}_{\mathrm{i}}^{\mathrm{T}}(\boldsymbol{\xi}_{MP})\mathbf{g}\Omega_{\mathrm{MP}} \tag{4.61}$$

where $\mathrm{no_{Nodes,el}}$ and $\mathrm{no_{MP,el}}$ are respectively the number of nodes and material points inside the element el.

The external forces due to traction are computed in the following way

$$\mathbf{f}_{\mathrm{L}}^{\mathrm{ext,trac}} = \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \int_{\partial\Omega_{\mathrm{el}}} \bar{\mathbf{N}}^{\mathrm{T}} \mathbf{t}_{\mathrm{L}} \cdot \mathbf{n} \, \mathrm{d}\partial\Omega = \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \sum_{\mathrm{i}=1}^{\mathrm{no_{Nodes,el}}} \sum_{\mathrm{MP}=1}^{\mathrm{no_{MP,el}}} \mathbf{N}^{\mathrm{T}}(\boldsymbol{\xi}_{MP}) \, \mathbf{f}_{\mathrm{L,MP}}^{\mathrm{ext,trac}} \tag{4.62}$$

where $\mathbf{f}_{\mathrm{L,MP}}^{\mathrm{ext}}$ is the force stored in the material point due to distributed external forces applied at the boundary of the body, as described in Section 4.3.2.

The complete form of the external force is summarized below:

$$\mathbf{f}_{\mathrm{L}}^{\mathrm{ext}} = \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \sum_{\mathrm{i}=1}^{\mathrm{no_{Nodes,el}}} \sum_{\mathrm{MP}=1}^{\mathrm{no_{MP,el}}} \mathbf{N}^{\mathrm{T}}(\boldsymbol{\xi}_{MP}) \, \mathbf{f}_{\mathrm{MP}}^{\mathrm{ext}} + \sum_{\mathrm{el}=1}^{\mathrm{no_{elm}}} \sum_{\mathrm{i}=1}^{\mathrm{no_{Nodes,el}}} \sum_{\mathrm{MP}=1}^{\mathrm{no_{MP,el}}} \mathrm{m_{MP}} \mathbf{N}_{\mathrm{i}}^{\mathrm{T}}(\boldsymbol{\xi}_{MP})\mathbf{g}\Omega_{\mathrm{MP}} \tag{4.63}$$

4.4.4 Mass matrix

To solve the system of equations (Eq. 4.54), the mass matrix has to be inverted. In practice, to simplify computations, a lumped mass matrix may be used instead of the consistent mass matrix given by Eq. 4.57. The lumped mass matrix is a diagonal matrix in which each entry m_i is obtained by summing over the corresponding row of the consistent mass matrix. Matrix inversions become trivial, although the result of using a lumped mass matrix is some dissipation of kinetic energy that has been quantified by [14]. Using the property $\sum_{MP=1}^{no_{MP}} N_i(\xi_{MP}) = 1$ it becomes

$$\mathbf{M}_L^{lump} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} \mathbf{N}^T(\xi_{MP})\, m_{MP} \tag{4.64}$$

Hereinafter, the superscript lump is removed from the lumped mass matrix $\mathbf{M}^{lump}$ to simplify the notation, and mass matrix always will refer to a lumped matrix.

In this study the 4-node tetrahedral element is used and the lumping procedure gives the following expression for the mass matrix **M**.

$$\mathbf{M}_L = \sum_{el=1}^{n_{elm}} \mathbf{M}_{L,el} \tag{4.65}$$

$$\mathbf{M}_{L,el} = \begin{bmatrix} \mathbf{m}_1 & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & \mathbf{m}_2 & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & ... & ... & \mathbf{m}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & \mathbf{m}_{NodeEl} \end{bmatrix} \tag{4.66}$$

$$\mathbf{m}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \; ; \; \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.67}$$

$$m_i = \sum_{MP=1}^{no_{MP,el}} m_{MP} N_i(\xi_{MP}) \tag{4.68}$$

4.4.4.1 Time discretization

The time has been discretized into different instants (k). Considering Δt the time step size, the time discretization approach is $t^{k+1} = t^k + \Delta t$.

If the general system of equations (Eq. 4.54) is posed at time t^k, it can be rewritten as Eq. 4.42, where the acceleration $\hat{\mathbf{a}}^k$ is the unknown.

$$\mathbf{M}_L^k \cdot \hat{\mathbf{a}}_L^k = \mathbf{f}_L^{int^k} + \mathbf{f}_L^{ext^k} \tag{4.69}$$

An explicit Euler time integration scheme is used to update the velocity (Eq. 4.43). This is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. Being $\hat{\mathbf{v}}^k$ the velocity at time t^k, the velocity at the next time step t^{k+1} is calculated using the acceleration at time t^k as follows.

$$\hat{\mathbf{v}}_L^{k+1} = \hat{\mathbf{v}}_L^k + \triangle t\, \hat{\mathbf{a}}_L^k \tag{4.70}$$

On the other hand, the displacements at time t^{k+1} are calculated using the updated velocity $\hat{\mathbf{v}}_L^{k+1}$ as indicated in Eq. 4.71.

$$\hat{\mathbf{u}}_L^{k+1} = \hat{\mathbf{u}}_L^k + \triangle t\, \hat{\mathbf{v}}_L^{k+1} \tag{4.71}$$

4.4.5 Solution algorithm for single time step

The algorithm presented here is based on the work presented by [3], which was improved respected the original one [2]. The key point is to work with momentum instead of velocity as much as possible, thus avoiding divisions by nodal masses. In each time step, the MPM computational cycle can be listed as follows.

1 The nodal mass is calculated using the shape functions and the lumped mass matrix at time t^k is formed (Eq. 4.64). The internal and external forces are evaluated in the nodes (Eqs. 4.63 and 4.59).

2 The momentum balance equation (Eq. 4.69) is solved and the nodal accelerations $\hat{\mathbf{a}}^k_{L,i}$ are determined.

$$\hat{\mathbf{a}}^k_{L,i} = [\mathbf{M}^k_{L,i}]^{-1} \left(\mathbf{f}^{ext,k}_{L,i} - \mathbf{f}^{int,k}_{L,i}\right) \tag{4.72}$$

3 The velocity at the material points is updated considering Eq. 4.70 as

$$\mathbf{v}^{k+1}_{MP} = \mathbf{v}^k_{MP} + \Delta t \sum_{i=1}^{no_{Nodes}} N_i(\xi^k_{MP})\, \hat{\mathbf{a}}^k_{L,i} \tag{4.73}$$

4 Update nodal momentum

$$\mathbf{P}^{k+1}_{L,i} = \sum_{el=1}^{no_{el,i}} \sum_{MP=1}^{no_{MP,el}} m_{MP}\, N_j(\xi^k_{MP})\, \mathbf{v}^{k+1}_{MP} \tag{4.74}$$

5 Update nodal velocities

$$\hat{\mathbf{v}}^{k+1}_{L,i} = \frac{\mathbf{P}^{k+1}_{L,i}}{\mathbf{M}^k_{L,i}} \tag{4.75}$$

6 . Compute incremental nodal displacement

$$\Delta\hat{\mathbf{u}}^{k+1}_{L,i} = \Delta t\, \hat{\mathbf{v}}^{k+1}_{L,i} \tag{4.76}$$

7 . Compute strain increment

$$\Delta\varepsilon^{k+1}_{MP} = \mathbf{B}(\mathbf{x}_{MP})\Delta\hat{\mathbf{u}}^{k+1}_{L,i}$$

8 The stresses are updated using a material constitutive model of water

9 Update volume and density of MP

$$\Omega^{k+1}_{MP} = \Omega^k_{MP}(1 + \Delta\varepsilon_{vol,MP}) \qquad \rho^{k+1}_{MP} = \frac{\rho^k_{MP}}{(1 + \Delta\varepsilon_{vol,MP})} \tag{4.77}$$

10 Particle positions are updated considering Eq. 4.44 as

$$\mathbf{x}^{k+1}_{MP} = \mathbf{x}^k_{MP} + \sum_{i=1}^{no_{Nodes}} N_i(\xi^k_{MP})\, \Delta\hat{\mathbf{u}}^k_{L,i} \tag{4.78}$$

11 The computational grid is initialized for the next step, nodal values are discarded and the material points carry all the updated information.

4.5 Two-phase Single-point formulation for saturated soil

4.5.1 Discretized momentum balance equation of liquid phase

The weak form can be written as follow:

$$\delta\hat{\mathbf{v}}_L^T \int_{\partial\Omega} \overline{\mathbf{N}}^T \mathbf{t}_L \cdot \mathbf{n}\, d\partial\Omega - \delta\hat{\mathbf{v}}_L^T \int_\Omega B^T p_L \boldsymbol{\delta}\, d\Omega - \delta\hat{\mathbf{v}}_L^T \left[\int_\Omega \overline{\mathbf{N}}^T n \frac{\rho_L g}{k} \overline{\mathbf{N}}\, d\Omega\right] (\hat{\mathbf{v}}_L - \hat{\mathbf{v}}_S) +$$
$$+\delta\hat{\mathbf{v}}_L^T \int_\Omega \overline{\mathbf{N}}^T \rho_L \mathbf{g}\, d\Omega - \delta\hat{\mathbf{v}}_L^T \left[\int_\Omega \overline{\mathbf{N}}^T \rho_L \overline{\mathbf{N}}\, d\Omega\right] \hat{\mathbf{a}}_L = \mathbf{0} \tag{4.79}$$

in which $\boldsymbol{\delta} = \begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\end{bmatrix}^T$.

The discretized form can be written in the following yields:

$$\mathbf{f}_L^{ext} - \mathbf{f}_L^{int} - \mathbf{f}^d = \mathbf{M}_L \hat{\mathbf{a}} \tag{4.80}$$

with

$$\mathbf{f}_L^{ext} = \int_{\partial\Omega} \overline{\mathbf{N}}^T \mathbf{t}_L \cdot \mathbf{n}\, d\partial\Omega + \int_\Omega \rho_L \overline{\mathbf{N}}^T \mathbf{g}\, d\Omega \tag{4.81}$$

$$\mathbf{f}_L^{int} = \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}_L\, d\Omega \tag{4.82}$$

$$\mathbf{f}^d = \int_\Omega \overline{\mathbf{N}}^T n \frac{\rho_L g}{k} \overline{\mathbf{N}}\, d\Omega\, (\hat{\mathbf{v}}_L - \hat{\mathbf{v}}_S) \tag{4.83}$$

$$\mathbf{M}_L = \int_\Omega \rho_L \overline{\mathbf{N}}^T \overline{\mathbf{N}} d\Omega \tag{4.84}$$

where $\mathbf{f}_L^{ext}$ is the vector of nodal external forces, $\mathbf{f}_L^{int}$ is the vector of nodal internal forces, $\mathbf{f}^d$ is the vector of drag forces and $\mathbf{M}_L$ is the nodal mass matrix.

4.5.2 Discretized momentum balance equation of mixture

The weak form can be written as follow:

$$\delta\hat{\mathbf{v}}_S^T \int_{\partial\Omega} \overline{\mathbf{N}}^T \mathbf{t} \cdot \mathbf{n}\, d\partial\Omega - \delta\hat{\mathbf{v}}_S^T \int_\Omega B^T \boldsymbol{\sigma}\, d\Omega +$$
$$+ \delta\hat{\mathbf{v}}_S^T \int_\Omega \overline{\mathbf{N}}^T \rho \mathbf{g}\, d\Omega - \delta\hat{\mathbf{v}}_S^T \left[\int_\Omega \overline{\mathbf{N}}^T n\, \rho_L \overline{\mathbf{N}}\, d\Omega\right] \hat{\mathbf{a}}_L - \delta\hat{\mathbf{v}}_S^T \left[\int_\Omega \overline{\mathbf{N}}^T (1-n)\, \rho_S \overline{\mathbf{N}}\, d\Omega\right] \hat{\mathbf{a}}_S = \mathbf{0} \tag{4.85}$$

The discretized form can be written in the following yields:

$$\mathbf{f}^{ext} - \mathbf{f}^{int} - \bar{\mathbf{M}}_L \hat{\mathbf{a}}_L = \mathbf{M}_S \hat{\mathbf{a}}_S \tag{4.86}$$

with

$$\mathbf{f}^{ext} = \int_{\partial\Omega} \overline{\mathbf{N}}^T \mathbf{t} \cdot \mathbf{n}\, d\partial\Omega + \int_\Omega \rho \overline{\mathbf{N}}^T \mathbf{g}\, d\Omega \tag{4.87}$$

$$\mathbf{f}^{int} = \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}\, d\Omega \tag{4.88}$$

$$\bar{\mathbf{M}}_L = \int_\Omega \overline{\mathbf{N}}^T n\, \rho_L \overline{\mathbf{N}}\, d\Omega \tag{4.89}$$

$$\mathbf{M}_S = \int_\Omega (1-n) \rho_S \overline{\mathbf{N}}^T \overline{\mathbf{N}} d\Omega \tag{4.90}$$

where $\mathbf{f}^{ext}$ is the vector of nodal external forces, $\mathbf{f}^{int}$ is the vector of nodal internal forces, $\bar{\mathbf{M}}_L$ is the nodal mass matrix (including porosity) of the liquid phase and $\mathbf{M}_S$ is the nodal mass matrix of the solid phase.

4.5.3 Calculation of internal forces

In finite element formulation the numerical integration of all integrals above is usually performed using a reference (or *parent*) element system ($\boldsymbol{\xi}$), and the mapping procedure is done using the *Jacobian matrix*. In this manual, a unit 4-node tetrahedron with linear shape functions is chosen to be the reference element. Details about the calculation of the internal forces are discussed in Section 4.3.3 and details about the mapping procedure are described in Section 4.8.

The internal force vector becomes

$$\mathbf{f}_L^{int} = \sum_{el=1}^{no_{elm}} \int_{\Omega_{el}} \mathbf{B}^T \boldsymbol{\sigma}_L \, d\Omega = \sum_{el=1}^{no_{elm}} \int_{\square} \mathbf{B}^T \boldsymbol{\sigma}_L |J| \, d\square \tag{4.91}$$

where Ω_{el} is the volume of the element el in the global coordinate system, no_{elm} is the number of active elements, $\square$ is volume of the parent element, $d\square = d\xi_1 d\xi_2 d\xi_3$ is the infinitesimal volume in the parent element system, $|J|$ is the determinant of the Jacobian, and $\mathbf{B}$ is the matrix of the shape function gradients calculated at location $\boldsymbol{\xi}$ with respect to the parent coordinate system.

In the material point method the MP carry all the information of the continuum, including the stresses. Thus, the internal forces are computed in the following form

$$\mathbf{f}_L^{int} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} \mathbf{B}_i^T(\boldsymbol{\xi}_{MP}) \boldsymbol{\sigma}_{L,MP} \Omega_{MP} \tag{4.92}$$

where $no_{MP,el}$ is the number of material points within element el.

4.5.4 Calculation of external forces

As for the one-phase-solid formulation, the external forces can be subdivided in two parts: the body force, which is generated by gravity acceleration, and the external loading due to distributed forces applied to the boundary of the liquid body.

The body force can be computed in the following way

$$\mathbf{f}_L^{ext,grav} = \sum_{el=1}^{no_{elm}} \int_{\Omega_{el}} \rho_L \bar{\mathbf{N}}^T \mathbf{g} \, d\Omega = \sum_{el=1}^{no_{elm}} \int_{\square} \rho \mathbf{N}^T \mathbf{g} |J| \, d\square \tag{4.93}$$

where Ω_{el} is the volume of the element el in the global coordinate system, no_{elm} is the number of active elements, $\square$ is volume of the parent element, $d\square = d\xi_1 d\xi_2 d\xi_3$ is the infinitesimal volume in the parent element system, $|J|$ is the determinant of the Jacobian, and $\mathbf{N}$ is the shape function matrix calculated at location $\boldsymbol{\xi}$ with respect to the parent coordinate system. Applying the material point integration, the equation yield as follows

$$\mathbf{f}_L^{ext,grav} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} m_{MP} \mathbf{N}_i^T(\boldsymbol{\xi}_{MP}) \mathbf{g} \Omega_{MP} \tag{4.94}$$

where $no_{Nodes,el}$ and $no_{MP,el}$ are respectively the number of nodes and material points inside the element el.

The external forces due to traction are computed in the following way

$$\mathbf{f}_L^{ext,trac} = \sum_{el=1}^{no_{elm}} \int_{\partial\Omega_{el}} \bar{\mathbf{N}}^T \mathbf{t}_L \cdot \mathbf{n} \, d\partial\Omega = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} \mathbf{N}^T(\xi_{MP}) \, \mathbf{f}_{L,MP}^{ext,trac} \tag{4.95}$$

where $\mathbf{f}_{L,MP}^{ext}$ is the force stored in the material point due to distributed external forces applied at the boundary of the body, as described in Section 4.3.2.

The complete form of the external force is summarized below:

$$\mathbf{f}_L^{ext} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} \mathbf{N}^T(\xi_{MP}) \, \mathbf{f}_{MP}^{ext} + \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} m_{MP} \mathbf{N}_i^T(\xi_{MP}) \mathbf{g} \Omega_{MP} \tag{4.96}$$

4.5.5 Lumping procedure: Mass matrix and Drag matrix

To solve the system of equations (Eq. 4.105), the mass matrix has to be inverted. In practice, to simplify computations, a lumped mass matrix may be used instead of the consistent mass matrix given by Eq. 4.90. The lumped mass matrix is a diagonal matrix in which each entry m_i is obtained by summing over the corresponding row of the consistent mass matrix. Matrix inversions become trivial, although the result of using a lumped mass matrix is some dissipation of kinetic energy that has been quantified by [14]. Using the property $\sum_{MP=1}^{no_{MP}} N_i(\xi_{MP}) = 1$ it becomes

$$\bar{\mathbf{M}}_L^{lump} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{MP=1}^{no_{MP,el}} \mathbf{N}^T(\xi_{MP}) \, m_{L,MP} \tag{4.97}$$

Hereinafter, the superscript lump is removed from the lumped mass matrix $\bar{\mathbf{M}}^{lump}$ to simplify the notation, and mass matrix always will refer to a lumped matrix.

In this study the 4-node tetrahedral element is used and the lumping procedure gives the following expression for the mass matrix $\bar{\mathbf{M}}_L$.

$$\bar{\mathbf{M}}_L = \sum_{el=1}^{n_{elm}} \mathbf{M}_{L,el} \tag{4.98}$$

$$\bar{\mathbf{M}}_{L,el} = \begin{bmatrix} \mathbf{m}_1 & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & \mathbf{m}_2 & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & ... & ... & \mathbf{m}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & \mathbf{m}_{NodeEl} \end{bmatrix} \tag{4.99}$$

$$\mathbf{m}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \; ; \; \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.100}$$

$$m_i = \sum_{MP=1}^{no_{MP,el}} m_{L,MP} N_i(\xi_{MP}) \tag{4.101}$$

The same procedure is used to calculate the lumped mass matrix for the solid phase $\mathbf{M}_S$.

The drag force $\mathbf{f}^d$ can be written as follows

$$\mathbf{f}^d = \mathbf{Q}\,(\hat{\mathbf{v}}_L - \hat{\mathbf{v}}_S) \tag{4.102}$$

with

$$\mathbf{Q} = \int_\Omega \bar{\mathbf{N}}^T n \frac{\rho_L g}{k} \bar{\mathbf{N}}\, d\Omega$$

The lumping technique is also applied to the drag matrix $\mathbf{Q}$ as follows:

$$\mathbf{Q}^{lump} = \sum_{el=1}^{no_{elm}} \sum_{i=1}^{no_{Nodes,el}} \sum_{k=1}^{no_{int,el}} \mathbf{N}_i^T n_k \frac{\rho_L g}{k} |J_k| \tag{4.103}$$

Hereinafter, likewise for the mass matrix, the superscript lump is removed from the lumped drag matrix $\mathbf{Q}^{lump}$ to simplify the notation, and drag matrix always will refer to a lumped matrix. In this study the 4-node tetrahedral element is used and the lumping procedure gives the following compact expression:

$$\mathbf{Q} = \sum_{el=1}^{n_{elm}} \mathbf{Q}_{el} \;;\; \mathbf{Q}_{el} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{q}_2 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{q}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{q}_{NodeEl} \end{bmatrix}$$

$$\mathbf{q}_i = \begin{bmatrix} q_i & 0 & 0 \\ 0 & q_i & 0 \\ 0 & 0 & q_i \end{bmatrix} \;;\; \mathbf{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$q_i = \sum_{k=1}^{no_{int,el}} n_k \frac{\rho_L g}{k} N_i(\boldsymbol{\xi}_k) |J_k| W_k$$

where $no_{Nodes,el}$ and $no_{int,el}$ are respectively the number of nodes and integration points inside the element el, and W_k is the weight factor (or integration weight) of the integration point k.

4.5.6 Time discretization

The discrete system of equations can be written as:

$$\mathbf{M}_L \hat{\mathbf{a}}_L^k = \mathbf{f}_L^{ext,k} - \mathbf{f}_L^{int,k} - \mathbf{f}^{d,k} \tag{4.104}$$

$$\mathbf{M}_S^t \hat{\mathbf{a}}_S^k = \mathbf{f}^{ext,k} - \mathbf{f}^{int,k} - \bar{\mathbf{M}}_L^k \hat{\mathbf{a}}_L^k \tag{4.105}$$

Note that in Eqs. 4.105 and 4.105 the subscript L and S denote that the quantity is referred to the fluid and water phase respectively; no subscript indicates that the quantity belongs to the mixture.

The Euler-Cromer scheme is used to integrate the equations in time. From Eq. 4.104 the fluid acceleration at time t is calculated and used to update the fluid velocity $\mathbf{v}_L^{t+\Delta t}$. The solid acceleration is calculated solving Eq. 4.105 and used to update the solid velocity $\mathbf{v}_S^{t+\Delta t}$. Incremental strains are calculated at the MP from the updated velocities, after that the constitutive relations are used to calculate the stresses and pore water pressure.

In the implementation used for this manual only one set of MP, representing the solid phase is considered. This means that the MP store all the informations regarding the soild and the liquid phase, and their positions are updated according to the solid displacement.

4.5.7 Solution algorithm for single timestep

The initialization of MP explained in Section 4.3.2 is easily extended to the two-phase problem, then no more details are given in this Section.

The solution sequence of a single time step is described in the following:

1 The momentum equations for the fluid and the mixture are initialized by mapping the significative quantities from the MP to the mesh nodes. The procedure is similar to step 1 of the algorithm presented in Section 4.3.7.

2 Eq. 4.104 is solved for $\hat{\mathbf{a}}_L^t$

$$\hat{\mathbf{a}}_L^t = \mathbf{M}_L^{t,-1}\left[\mathbf{f}_L^{ext,t} - \mathbf{f}_L^{int,t} - \mathbf{f}^{d,t}\right] \tag{4.106}$$

3 The acceleration vector $\mathbf{a}_S^t$ is calculated from Eq. 4.105 as:

$$\hat{\mathbf{a}}_S^t = \mathbf{M}_S^{t,-1}\left[\mathbf{f}^{ext,t} - \mathbf{f}^{int,t} - \bar{\mathbf{M}}_L^t \mathbf{a}_L^t\right] \tag{4.107}$$

4 The velocities of the MPs are updated using nodal accelerations and shape functions:

$$\hat{\mathbf{v}}_{L,MP}^{t+\Delta t} = \hat{\mathbf{v}}_{L,MP}^t + \sum_{i=1}^{n_{en}} \Delta t N_i(\xi_{MP}^t)\hat{\mathbf{a}}_{L,i}^t \tag{4.108}$$

$$\hat{\mathbf{v}}_{S,MP}^{t+\Delta t} = \hat{\mathbf{v}}_{S,MP}^t + \sum_{i=1}^{n_{en}} \Delta t N_i(\xi_{MP}^t)\hat{\mathbf{a}}_{S,i}^t \tag{4.109}$$

5 The nodal velocities $\hat{\mathbf{v}}_L^{t+\Delta t}$ and $\hat{\mathbf{v}}_S^{t+\Delta t}$ are then calculated from the updated momentum solving the following equation

$$\mathbf{P}_{L,i}^{t+\Delta t} = \bar{\mathbf{M}}_L^t \hat{\mathbf{v}}_L^{t+\Delta t} \approx \sum_{MP=1}^{n_{MP}} m_{L,MP}\mathbf{N}^T(\xi_{MP}^t)\hat{\mathbf{v}}_{L,MP}^{t+\Delta t} \tag{4.110}$$

$$\mathbf{P}_{S,i}^{t+\Delta t} = \mathbf{M}_S^t \hat{\mathbf{v}}_S^{t+\Delta t} \approx \sum_{MP=1}^{n_{MP}} m_{S,MP}\mathbf{N}^T(\xi_{MP}^t)\hat{\mathbf{v}}_{S,MP}^{t+\Delta t} \tag{4.111}$$

6 Nodal velocities are integrated to get nodal incremental displacements

$$\Delta\hat{\mathbf{u}}_L^{t+\Delta t} = \Delta t\hat{\mathbf{v}}_L^{t+\Delta t}\ ;\ \Delta\hat{\mathbf{u}}_S^{t+\Delta t} = \Delta t\hat{\mathbf{v}}_S^{t+\Delta t} \tag{4.112}$$

7 Strains at MPs are calculated as

$$\Delta\varepsilon_{L,MP}^{t+\Delta t} = \mathbf{B}(\xi_{MP}^t)\hat{\mathbf{v}}_L^{t+\Delta t}\Delta t \tag{4.113}$$

$$\Delta\varepsilon_{S,MP}^{t+\Delta t} = \mathbf{B}(\xi_{MP}^t)\hat{\mathbf{v}}_S^{t+\Delta t}\Delta t \tag{4.114}$$

and stresses are updated according to a constitutive relation

8 Water pressures at the material points are updated as:

$$\Delta\bar{\varepsilon}_{L,MP}^{t+\Delta t} = \frac{1}{n_{MP}^t}\left[(1-n_{MP}^t)\Delta\varepsilon_{S,MP}^{t+\Delta t} + n_{MP}^t\Delta\varepsilon_{L,MP}^{t+\Delta t}\right] \tag{4.115}$$

$$p_{L,MP}^{t+\Delta t} \approx p_{L,MP}^t + K_{L,MP}\Delta\bar{\varepsilon}_{L,MP}^{t+\Delta t} \tag{4.116}$$

where $\boldsymbol{\delta} = \begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\end{bmatrix}$, $\Delta\varepsilon_{vol,S}$ and $\Delta\varepsilon_{vol,L}$ are the volumetric strain, i.e. $\Delta\varepsilon_{vol} = \Delta\varepsilon_{11} + \Delta\varepsilon_{22} + \Delta\varepsilon_{33}$, at the MP for the solid and liquid phase respectively.

9 The total stress is calculated as:

$$\boldsymbol{\sigma}_{MP}^{t+\Delta t} = \boldsymbol{\sigma}'^{t+\Delta t}_{MP} + \delta p^{t+\Delta t}_{L,MP} \tag{4.117}$$

10 Volumes associated with MP are updated using the solid volumetric strain increment

$$\Omega_{MP}^{t+\Delta t} = (1 + \Delta\epsilon^{t+\Delta t}_{vol,S,MP}) \tag{4.118}$$

11 The positions of MP are updated using the displacements of the solid phase

$$\mathbf{x}_{MP}^{t+\Delta t} = \mathbf{x}_{MP}^{t} + \sum_{i=1}^{n_{no_{nodes}}} N_i(\boldsymbol{\xi}^t_{MP})\Delta\hat{\mathbf{u}}^{t+1}_{S,i} \tag{4.119}$$

12 The book-keeping is updated using the new position of particles

The reader should observe that, similarly to the one-phase solution procedure, MP velocity are calculated from nodal accelerations and nodal velocities are computed from the nodal momentum. This is called modified Lagrangian algorithm and allows to overcome the small mass problem.

4.6 Two-phase Single-point formulation for unsaturated soil

This section summarizes the algorithm for the solution of the two-phase Single-point MPM formulation for unsaturated soil. Since the procedure is very similar to the two-phase Single-point formulation for saturated soil, the reader is referred to Section 4.5 for further details.

4.6.1 Discretized momentum balance of the liquid

The discretized momentum balance equation of the liquid is

$$\widetilde{\mathbf{M}}_L\mathbf{a}_L = \mathbf{f}_L^{ext} - \mathbf{f}_L^{int} - \mathbf{Q}_L(\mathbf{v}_L - \mathbf{v}_S) \tag{4.120}$$

$\mathbf{a}_L$, $\mathbf{v}_L$, and $\mathbf{v}_S$ are the nodal acceleration and velocity vectors; $\widetilde{\mathbf{M}}_L$ is the intrinsic liquid lumped mass matrix (Eq. 4.121); $\mathbf{f}_L^{ext}$, $\mathbf{f}_L^{int}$ are internal and external nodal force vectors of the liquid phase (Eqs. 4.122-4.123), and $\mathbf{Q}_L$ is the drag force matrix (Eq. 4.124).

$$\widetilde{\mathbf{M}}_L \approx \sum_{MP=1}^{n_{MP}} \widetilde{m}_L^{MP}\mathbf{N} \tag{4.121}$$

$$\mathbf{f}_L^{ext} \approx \int_{\partial\Omega_p} \mathbf{N}^T\widehat{\mathbf{p}}_L d\partial\Omega_p + \sum_{MP=1}^{n_{NP}} m_L^{MP}\mathbf{N}^T\mathbf{g} \tag{4.122}$$

$$\mathbf{f}_L^{int} \approx \sum_{MP=1}^{n_{MP}} \mathbf{B}^T p_L^{MP}\mathbf{m}V_{MP} \tag{4.123}$$

$$\mathbf{Q}_L \approx \sum_{MP=1}^{n_{MP}} \mathbf{N}^T \frac{n_L^{MP} \rho_L g}{k_L^{MP}} \mathbf{N} V_{MP} \tag{4.124}$$

where $\widehat{\mathbf{p}}_L$ is the prescribed liquid pressure in the boundary $\partial\Omega_p$, $\mathbf{N}$ is the matrix of nodal shape functions and $\mathbf{B}$ is the matrix of the gradients of the nodal shape functions evaluated at local MP positions. The treatment of boundary conditions for unsaturated soils is described in details in Sec. 5.3.

The phase mass of the MP is calculated as Eq. 4.125, where V_{MP} is the volume of the MP.

$$\widetilde{m}_L^{MP} = \rho_L V_{MP} \tag{4.125}$$

4.6.2 Discretized momentum balance of the mixture

The discretized momentum balance equation of the mixture is:

$$\mathbf{M}_S \mathbf{a}_S + \mathbf{M}_L \mathbf{a}_L = \mathbf{f}^{ext} - \mathbf{f}^{int} \tag{4.126}$$

$\mathbf{a}_S$ and $\mathbf{a}_S$ are the nodal acceleration vectors; $\mathbf{M}_S$ and $\mathbf{M}_L$ are the solid liquid lumped mass matrices (Eq. 4.127-4.128); $\mathbf{f}^{ext}$, $\mathbf{f}^{int}$ are internal and external nodal force vectors of the mixture (Eqs. 4.129-4.130).

$$\mathbf{M}_L \approx \sum_{MP=1}^{n_{MP}} m_L^{MP} \mathbf{N} \tag{4.127}$$

$$\mathbf{M}_S \approx \sum_{MP=1}^{n_{MP}} m_S^{MP} \mathbf{N} \tag{4.128}$$

$$\mathbf{f}^{ext} \approx \int_{\partial\Omega_\tau} \mathbf{N}^T \boldsymbol{\tau} d\partial\Omega_\tau + \sum_{MP=1}^{n_{MP}} m_m^{MP} \mathbf{N}^T \mathbf{g} \tag{4.129}$$

$$\mathbf{f}^{int} \approx \sum_{MP=1}^{n_{MP}} \mathbf{B}^T \boldsymbol{\sigma}_{MP} V_{MP} \tag{4.130}$$

Where $\boldsymbol{\tau}$ is the prescribed traction vector in the boundary $\partial\Omega_\tau$.

The phase mass of the MP is calculated as Eqs. 4.131-4.133, where V_{MP} is the volume of the MP.

$$m_S^{MP} = n_S^{MP} \rho_S V_{MP} \tag{4.131}$$

$$m_L^{MP} = n_L^{MP} \rho_L V_{MP} \tag{4.132}$$

$$m_m^{MP} = \rho_m V_{MP} \tag{4.133}$$

4.6.3 Time discretization

Eqs. 4.120 and 4.126 are integrated in time using the Euler-Cromer explicit method. This time integration scheme is conditionally stable and the critical time step size is computed as indicated in Sec. 4.7.

$$\mathbf{M}_L \hat{\mathbf{a}}_L^k = \mathbf{f}_L^{ext,k} - \mathbf{f}_L^{int,k} - \mathbf{f}^{d,k} \tag{4.134}$$

$$\mathbf{M}_S^t \hat{\mathbf{a}}_S^k = \mathbf{f}^{ext,k} - \mathbf{f}^{int,k} - \bar{\mathbf{M}}_L^k \hat{\mathbf{a}}_L^k \tag{4.135}$$

The momentum balances (Eqs. 4.134 and 4.135) are solved at the nodes of the mesh. Mass balances (Eqs. 3.34 and 3.37) and constitutive equations are posed locally at the MPs to update secondary variables.

4.6.4 Solution algorithm for single time step

The numerical algorithm is based on the modified lagrangian algorithm originally proposed by Sulsky et al. [3] for one-phase material, and successfully extended to multiphase materials (see e.g. [11, 16, 17]). The MPM solution scheme for each time step can be summarised as follows.

1 Liquid momentum balance equation (Eq. 4.120) is assembled and solved for the liquid nodal acceleration $\mathbf{a}_L$.
2 Mixture momentum balance equation (Eq. 4.126) is assembled and solved to obtain the nodal acceleration of the solid $\mathbf{a}_S$.
3 Velocities and momentum of the MPs are updated from nodal accelerations of each phase.
4 Nodal velocities are calculated from nodal momentum and used to compute the strain increment at the MP location and the terms on the right-end-side of Eq. 3.37.
5 Mass balance equation (Eq. 3.37) and soil stress-strain equation give the increment of pore pressure and effective stress respectively.
6 State variables at MPs are updated. Degree of saturation and hydraulic conductivity are updated according to SWRC and HCC respectively.
7 Displacement and position of each MP are updated according to the updated velocity of the solid phase.
8 Nodal values are discarded, the MPs carry all the updated information, and the computational grid is initialised for the next time step.

Steps 1 and 2 are commonly referred to as *Lagrangian phase*, while steps 3 to 8 are called *convective* or *Eulerian phase* of the method to emphasize the fact that MPM combines the advantages of Lagrangian and Eulerian approaches.

4.7 Stability criterion: the Critical timestep

A numerical algorithm is stable and well posed when the numerical solution is maintained close to the exact one throughout the calculation, which means that the errors stay bounded. The stability of a numerical algorithm depends on the integration scheme. Explicit time integration schemes, such as the one presented above, are conditionally stable.

The Courant-Friedrichs-Levy condition is considered to obtain the critical time interval in order to achieve a stable solution.

4.7.1 Criterion for One-phase-solid

The criterion to determine the critical timestep in case of one-phase-solid simulation is the following

$$\Delta t_{cr} = \frac{L_{min}}{c} \; ; \; c = \sqrt{\frac{E_c}{\rho}} \tag{4.136}$$

where E_c is the constrained modulus of the solid media. The length L_{min} is the minimal length of the element and can be determined as the shortest distance from the side of the maximum area to the opposite node. The construction is shown in Figure 4.2.

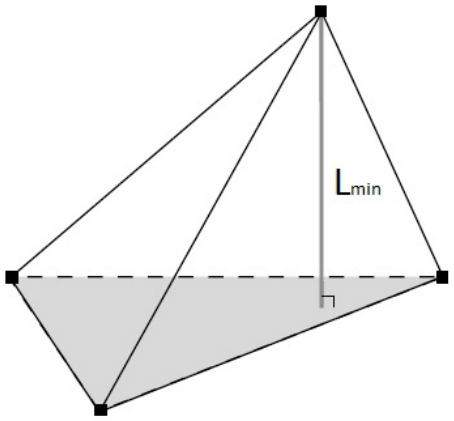


Figure 4.2: *Minimum length* L_{min} *for a tetrahedral element.*

4.7.2 Criterion for one-phase-liquid

The criterion to determine the critical timestep in case of one-phase-solid simulation is the following

$$\Delta t_{cr} = \frac{L_{min}}{c_L} \; ; \; c_L = \sqrt{\frac{K_L}{\rho_L}} \tag{4.137}$$

where K_L is the bulk modulus of the water. The length L_{min} is the minimal length of the element and can be determined as the shortest distance from the side of the maximum area to the opposite node. The construction is shown in Figure 4.2.

4.7.3 Criterion for two-phase-coupled

The stability analysis of the two-phase-coupled formulation is onerous; for this reason, a first simplified analysis was performed in [18] on a simplified formulation that is derived by assuming an incompressible pore fluid. Based on the eigenvalue analyses of a single linear finite element, it is observed that in addition to the CFL stability condition, the influence of the permeability must be included. The full stability criterion for the two-phase-coupled formulation is represented by the following equation

$$\Delta t_{cr} = \min\left\{\Delta t_{cr,1}; \Delta t_{cr,2}\right\} \tag{4.138}$$

where

$$\Delta t_{cr,1} = m\frac{L_{min}}{c_1} \tag{4.139}$$

$$\Delta t_{cr,2} = \frac{2\tilde{\rho}k}{\rho_L g} \tag{4.140}$$

$$c_1 = \sqrt{\frac{E_c^u}{\rho}} \tag{4.141}$$

$$E_c^u = E_c + K_L/n \tag{4.142}$$

$$\rho = (1-n)\rho_S + n\rho_L \tag{4.143}$$

$$\tilde{\rho} = \rho + (\frac{1}{n} - 2)\rho_L \tag{4.144}$$

E_c^u is the undrained constrained modulus of the saturated soil, E_c is the laterally confined modulus of the solid skeleton and K_L is the bulk modulus of the liquid. The length L_{min} is the minimal length of the element and can be determined as the shortest distance from the side of the maximum area to the opposite node. The construction is shown in Figure 4.2.

It is worth noticing that $\Delta t_{cr,1}$ is a function of the size of the mesh, the stiffness, the porosity and the density of the material, while $\Delta t_{cr,2}$ is a function of the permeability, the porosity and the density. In many cases, the time step required to ensure the stability in low permeability soils can be smaller than the one required for high permeability soils, i.e. $\Delta t_{cr,2} < \Delta t_{cr,1}$.

The condition expressed by Eq. 4.138 can be improved and reformulated in the single equation [19]:

$$\Delta t_{cr} = \frac{-2a + \sqrt{4a^2 + 8(b + \sqrt{b^2 - 4c})}}{b + \sqrt{b^2 - 4c}} \tag{4.145}$$

The coefficients a, b and c are given by

$$a = \frac{n\rho g}{(1-n)\rho_S k} \tag{4.146}$$

$$b = \frac{4(n\rho K_L + (1-2n)\rho_L K_L + n\rho_L E_c)}{n(1-n)\rho_S \rho_L L_{min}} \tag{4.147}$$

$$c = \frac{16 E_c K_L}{(1-n)\rho_S \rho_L L_{min}} \tag{4.148}$$

4.8 4-node tetrahedral element

The isoparametric element used in this manual is the 4-node tetrahedral. Figure 4.3 shows the element in the parent domain, in which the node number varies from 1 to 4, and in the global domain where the nodes numbers can vary arbitrarily depending on global mesh, and in this case number 11, 12, 13 and 14 are used as an example. One of the main features of the isoparametric elements is that the shape functions in both parent and global domain are the same. Therefore, the coordinates of a point in the global domain are determined as follows

$$\mathbf{x}(\boldsymbol{\xi}, t) = \sum_{i=1}^{4} N_i(\boldsymbol{\xi})\, \mathbf{x}_i(t) \tag{4.149}$$

where $\boldsymbol{\xi}$ is the vector of the parent coordinate system which is written as follows

$$\boldsymbol{\xi} = [\xi_1 \;\; \xi_2 \;\; \xi_3]^T \tag{4.150}$$

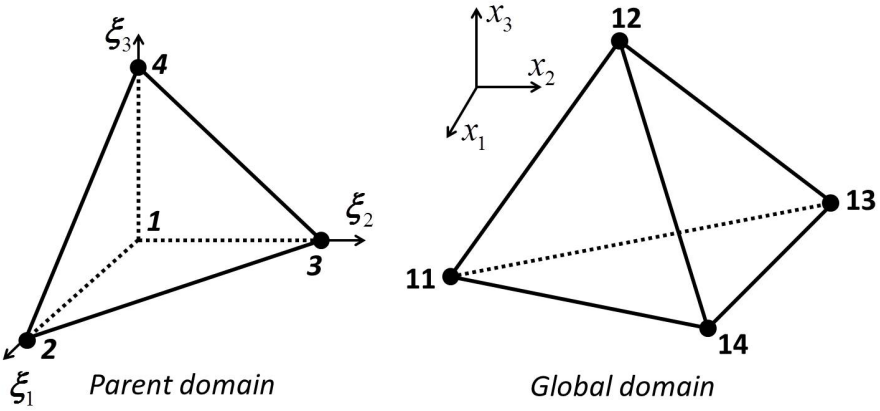


Figure 4.3: *4-node tetrahedral element. Parent and Global domain.*

The shape function is defined in the parent domain, i.e. in the local coordinate system, as follows:

$$N_1(\boldsymbol{\xi}) = 1 - \xi_1 - \xi_2 - \xi_3 \tag{4.151}$$
$$N_2(\boldsymbol{\xi}) = \xi_1 \tag{4.152}$$
$$N_3(\boldsymbol{\xi}) = \xi_2 \tag{4.153}$$
$$N_4(\boldsymbol{\xi}) = \xi_3 \tag{4.154}$$

where the subscript $1, 2, 3$ or 4 represents the node number in the parent domain.

Usually, the derivative of the shape function is also required to compute for instance the **B** matrix, which is needed to determine the internal forces of the system. The derivative is computed using the chain rule as follows

$$\begin{aligned}
\frac{\partial N_i(\boldsymbol{\xi})}{\partial \mathbf{x}_1} &= \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}\frac{\partial \xi_1}{\partial \mathbf{x}_1} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}\frac{\partial \xi_2}{\partial \mathbf{x}_1} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}\frac{\partial \xi_3}{\partial \mathbf{x}_1} \\
\frac{\partial N_i(\boldsymbol{\xi})}{\partial \mathbf{x}_2} &= \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}\frac{\partial \xi_1}{\partial \mathbf{x}_2} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}\frac{\partial \xi_2}{\partial \mathbf{x}_2} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}\frac{\partial \xi_3}{\partial \mathbf{x}_2} \\
\frac{\partial N_i(\boldsymbol{\xi})}{\partial \mathbf{x}_3} &= \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_1}\frac{\partial \xi_1}{\partial \mathbf{x}_3} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_2}\frac{\partial \xi_2}{\partial \mathbf{x}_3} + \frac{\partial N_i(\boldsymbol{\xi})}{\partial \xi_3}\frac{\partial \xi_3}{\partial \mathbf{x}_3}
\end{aligned} \tag{4.155}$$

The compact form can be written as follows

$$\begin{Bmatrix} \frac{\partial N_i(\boldsymbol{\xi})}{\partial \mathbf{x}_1} \\ \frac{\partial N_i(\boldsymbol{\xi})}{\partial \mathbf{x}_2} \\ \frac{\partial N_i(\boldsymbol{\xi})}{\partial \mathbf{x}_3} \end{Bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{\xi}_1}{\partial \mathbf{x}_1} & \frac{\partial \boldsymbol{\xi}_2}{\partial \mathbf{x}_1} & \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{x}_1} \\ \frac{\partial \boldsymbol{\xi}_1}{\partial \mathbf{x}_2} & \frac{\partial \boldsymbol{\xi}_2}{\partial \mathbf{x}_2} & \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{x}_2} \\ \frac{\partial \boldsymbol{\xi}_1}{\partial \mathbf{x}_3} & \frac{\partial \boldsymbol{\xi}_2}{\partial \mathbf{x}_3} & \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{x}_3} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_1} \\ \frac{\partial N_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_2} \\ \frac{\partial N_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_3} \end{Bmatrix} \tag{4.156}$$

The derivatives of the natural coordinates $\boldsymbol{\xi}$ to the global coordinates $\mathbf{x}$ are not explicitly available. The (3x3) matrix in Eq.(4.156) is the inverse of the Jacobian matrix $\mathbf{J}$, which is defined as follows

$$\begin{bmatrix} \frac{\partial \boldsymbol{\xi}_1}{\partial \mathbf{x}_1} & \frac{\partial \boldsymbol{\xi}_2}{\partial \mathbf{x}_1} & \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{x}_1} \\ \frac{\partial \boldsymbol{\xi}_1}{\partial \mathbf{x}_2} & \frac{\partial \boldsymbol{\xi}_2}{\partial \mathbf{x}_2} & \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{x}_2} \\ \frac{\partial \boldsymbol{\xi}_1}{\partial \mathbf{x}_3} & \frac{\partial \boldsymbol{\xi}_2}{\partial \mathbf{x}_3} & \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{x}_3} \end{bmatrix} = \mathbf{J}^{-1} = \begin{bmatrix} \frac{\partial \mathbf{x}_1}{\partial \boldsymbol{\xi}_1} & \frac{\partial \mathbf{x}_1}{\partial \boldsymbol{\xi}_2} & \frac{\partial \mathbf{x}_1}{\partial \boldsymbol{\xi}_3} \\ \frac{\partial \mathbf{x}_2}{\partial \boldsymbol{\xi}_1} & \frac{\partial \mathbf{x}_2}{\partial \boldsymbol{\xi}_2} & \frac{\partial \mathbf{x}_2}{\partial \boldsymbol{\xi}_3} \\ \frac{\partial \mathbf{x}_3}{\partial \boldsymbol{\xi}_1} & \frac{\partial \mathbf{x}_3}{\partial \boldsymbol{\xi}_2} & \frac{\partial \mathbf{x}_3}{\partial \boldsymbol{\xi}_3} \end{bmatrix}^{-1} \tag{4.157}$$

5 The boundary conditions

From a numerical point of view, two kinds of boundary conditions can be distinguished: natural and essential. On the one hand, essential boundary conditions are imposed directly on the solution. In this way, degrees of freedom are directly eliminated from the system of equations. Dirichlet conditions are an example. On the other hand, natural boundary conditions are imposed on a secondary variable, such as in Newman or Newton (Robin) conditions.

The boundary conditions required in a mechanical problem are summarized in Table 5.1. Those are prescribed traction and prescribed velocity, which correspond to natural and essential conditions respectively. In the numerical implementation, prescribed traction is included in the weak form of the governing equations, while calculated velocities are directly overwritten by the prescribed values.

Table 5.1: *Types of boundary conditions in a mechanical problem.*

Essential	Natural
Prescribed velocity	Prescribed traction

It is commonly accepted that boundary conditions are specified at the nodes of the mesh [20, 21]. This idea makes sense considering that the governing equations are solved in the computational grid, and consequently the MPM formulation persists almost identical to that used in the FEM.

The problem of applying the boundary conditions at the nodes appears when a moving boundary is considered. In these cases, the boundary condition has to move attached to the body. However, in the standard MPM formulation, the computational mesh remains constant. Therefore, in such cases, the boundary condition should be carried by the material points.

This way of handling the boundary conditions to solve the issue of movement leads to some inaccuracies, especially in natural conditions. If the boundary condition is applied on a material point, this condition has to be distributed all over the nodes of the element to solve the system of equations. If a material point is affected by a moving boundary condition, the element containing the point becomes part of the contour. Then, the contour has an equivalent thickness of the size of the element and the boundary condition is spread affecting all material points within the cell, even those which are not at the boundary. To minimize numerical errors, it is recommended to use a fine mesh.

In some cases a boundary condition must be applied at a deforming boundary, i.e. the computational mesh in MPM does not necessarily align with the boundary of the material. In these cases, the nodes belonging to the boundary are determined with the procedure proposed by [11]. For each time step, firstly the active elements (i.e. elements containing MPs) adjacent to an empty element (i.e. element without MPs) are detected (*boundary elements*); then the nodes belonging to the element side adjacent to an empty element (*boundary side*) are marked as *boundary nodes*. The MPs next to the boundary side are identified as *boundary material points* as shown in Fig. 5.1. Finally, if the boundary node lies inside the area where a specified condition has to be applied, e.g. in the infiltration zone or on the potential seepage face, the corresponding boundary condition is applied.

Another difficulty with moving boundaries appears when prescribed velocity or traction (pressure) do not have a constant direction, but it is normal to the boundary. This means that if the shape of the contour changes, the direction of the applied condition has to be updated during

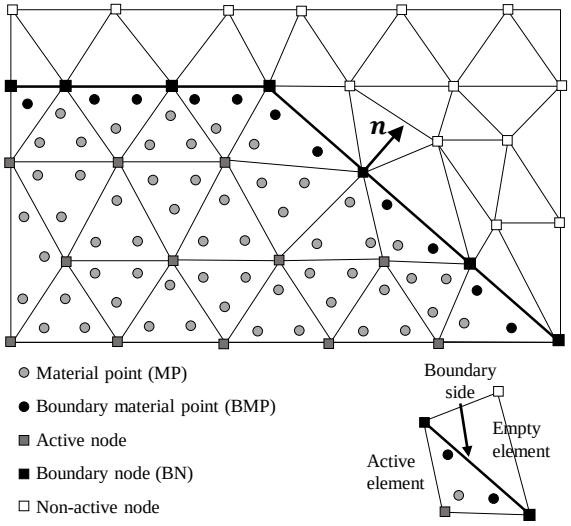


Figure 5.1: *Definition of boundary nodes and boundary material points*

the calculation. The normal direction at the node is determined by means of the gradient of mass as Eq. 5.1

$$\mathbf{n} \approx \frac{\sum_{MP=1}^{n_{MP}} m_m^{MP} \mathbf{B}}{\| \sum_{MP=1}^{n_{MP}} m_m^{MP} \mathbf{B} \|} \tag{5.1}$$

In Figure 5.2 a very simple problem is presented. It shows the movement of a beam that is fixed at one end and subjected to a vertical force F at the opposite end. In this example the fixity can be understood as an essential boundary condition in which the prescribed velocity is zero. Because it remain motionless, it can be applied at the nodes. On the other hand, the vertical force is a natural boundary condition which moves attached to the beam, therefore it should be carried by the material points.

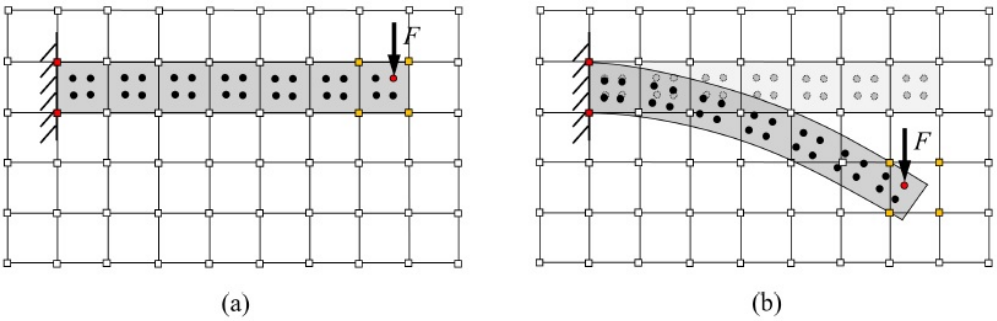


Figure 5.2: *Problem of a beam subjected to a vertical force at one end and embedded at the other one. (a) Initial discretization; (b) final discretization. The boundary conditions are carried in those nodes and material points colored in red. Yellow nodes are affected by the moving boundary condition although it is not stored in them [from [5]].*

5.1 Zero kinematic boundary conditions

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 5.3 illustrates such a situation. Regarding zero traction boundary conditions, they are automatically enforced to be satisfied by the solution of equations of motion.

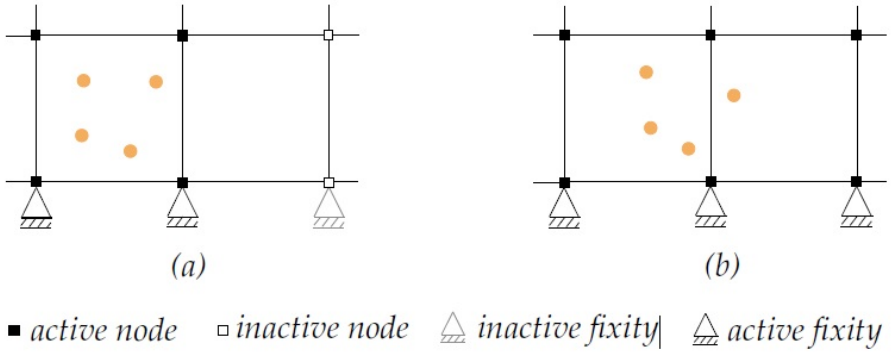


Figure 5.3: *Kinematic boundary conditions are prescribed at the active and inactive nodes (a) inactive fixity (b) active fixity. [from. Al-Kafaij 2013]*

5.2 Traction boundary conditions

Zero-traction boundary conditions are automatically enforced to be satisfied by the solution of equations of motion. Dealing with non-zero traction boundary conditions is more complex. Traction loads can be applied on either the element boundaries or the material points (MPs). The first option is applicable only when the boundary of the body remains aligned with loaded element boundaries throughout the computation. The second option consists in storing the load on selected MPs that move through the mesh.

When the traction is assigned at the element boundary, the nodal traction force is integrated like in FEM applying Gauss quadrature and then used in the momentum balance equation (Fig. 5.4 a). The applied load is thus integrated accurately and the traction nodal force is non-zero only for the nodes belonging to the loaded surface. When the traction is applied on the MPs, this is mapped to the material points located next to the element border, also called boundary material points (Fig. 5.4 b). These MP carry surface traction throughout the computations.

Considering, for instance, a tetrahedral element, the traction vector $\boldsymbol{\tau}_e$ applied at the triangular surface is interpolated from the nodes of this surface to the boundary MP. See figure 5.5.

Hence, the traction at boundary material point MP is

$$\boldsymbol{\tau}_e(\mathbf{x}_{MP}) \approx \sum_{i=1}^{n_{tri}} N_i(\boldsymbol{\xi}_{MP})\boldsymbol{\tau}_e(\mathbf{x}_i) \tag{5.2}$$

where N_i is the shape function of node i of the triangular surface element and $\boldsymbol{\xi}_{MP}$ are the coordinates of the boundary material point MP inside the parent triangular element. These

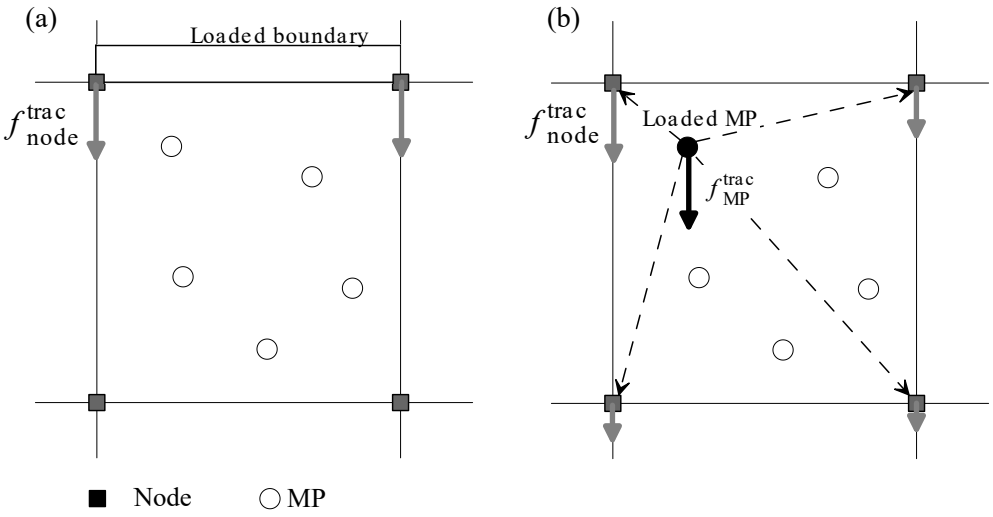


Figure 5.4: *(a) Traction is integrated from element boundary to adjacent nodes. (b) Traction is mapped from a loaded MP to the nodes of the element.*

coordinates simply represent the projection of the MP on the triangular surface element. The traction force vector $\mathbf{f}^{trac}_{MP}$ is then

$$\mathbf{f}^{trac}_{MP} = \tau_e(\mathbf{x}_{MP})\frac{S_e}{n_{ebMP}} = \frac{S_e}{n_{ebMP}}\sum_{i=1}^{n_{tri}} N_i(\xi_{MP})\tau_e(\mathbf{x}_i) \tag{5.3}$$

in which n_{ebMP} denotes the number of boundary MP located next to the loaded surface. This force vector keep the same direction and intensity throughout the calculation.

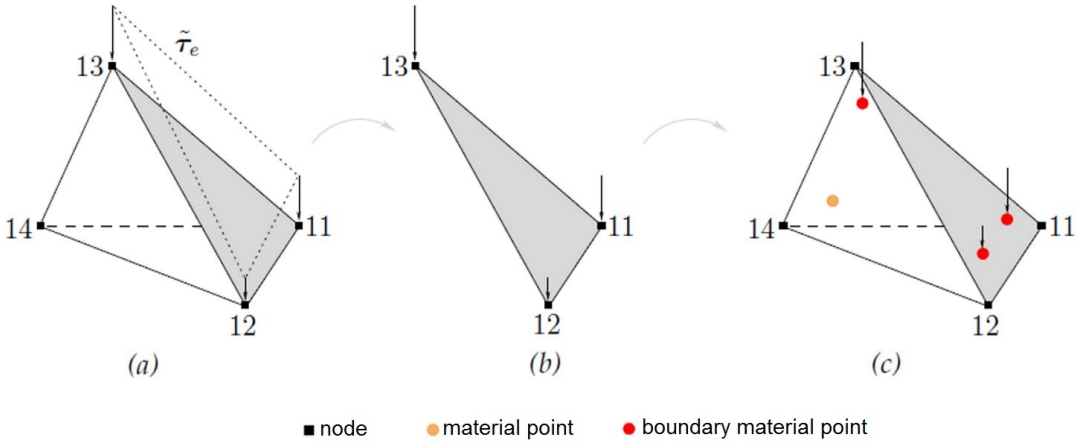


Figure 5.5: *Traction is mapped from a boundary particle to all nodes of the element. [from. Al-Kafaij 2013]*

During the simulation, the boundary particles move throughout the computational mesh, and a way to treat such tractions during the MPM simulation is to map them from boundary material points to all nodes of the element where a boundary particle is located, see Figure 5.4b. Such mapping procedure is carried out according to the following equation.

$$\mathbf{f}^{tract} = \sum_{el=1}^{no_{el}}\sum_{MP=1}^{no_{MP,el}} \mathbf{N}^T(\xi_{MP})\mathbf{f}^{trac}_{MP} \tag{5.4}$$

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.

5.3 Hydraulic boundary conditions

Typical geotechnical problems with unsaturated soils such as the study of dams, levees, and natural slopes may need the treatment of the following BCs for the liquid phase (Figure 3.3a):

Impermeable BC: fluid is not permitted to flow across this boundary, thus the fluid velocity perpendicular to the boundary $\partial\Omega_{v_{L,0}} \subset \partial\Omega_{v_L}$ is zero. This condition is implemented as explained in Sec. 5.1.

Infiltration BC: imposed flux along a boundary $\partial\Omega_{v_{L,infilt}} \subset \partial\Omega_{v_L}$ able to target infiltration and evaporation processes (see Sec. 5.3.2);

Pressure BC: prescribed pressure applied on $\partial\Omega_p$. This condition is very similar to the traction boundary condition described in Sec. 5.2.

Hydraulic head BC: this is an alternative definition of pressure BC to simulate water levels in reservoirs. A total head value $\widehat{H}$ is prescribed on the boundary $\partial\Omega_H \subset \partial\Omega_p$, which is related to the applied pressure $\widehat{p}_L$ by means of Bernoulli's equation (Eq. 5.5) where h_g is the geometric head

$$\widehat{H} = h_g - \frac{\widehat{p}_L}{\rho_L g} \tag{5.5}$$

Seepage face BC: this condition occurs at the interface between soil and the atmosphere, like the downstream side of a river embankment, where it is not known a priori if the condition will be Dirichelet or Neumann type. At this boundary, if soil is saturated, then fluid is free to exit at zero pressure (i.e. $\widehat{p}_L = 0$, Neumann BC). Otherwise, the boundary becomes closed (i.e. prescribed velocity, Dirichelet BC). Implementation details are in Sec. 5.3.3.

5.3.1 Total head BC

The implementation of the total head boundary condition follows [11, 22]. The total head is converted to an imposed pressure condition $\widehat{p}_L$, using Eq. 5.5 solved for the liquid pressure. This means that a hydrostatic approximation is adopted, neglecting local turbulence induced by the river flow or by the potential collapse of the internal bank. A series of hydraulic head values in time, like the ones provided by water gauge readings, is assigned with an input file, and at each time step the current hydraulic head H_{curr} is computed by linear interpolation.

The resulting nodal vector is part of the vector of external forces in the liquid momentum equation ($\mathbf{f}_L^{ext}$ in Eq. 4.120) and in the mixture momentum equation ($\mathbf{f}^{ext}$ in Eq. 4.126) to account also for water weight.

The *potential seepage face BC* is automatically assigned at the boundary nodes included in the hydraulic head BC but lying above.

5.3.2 Infiltration BC

This boundary condition is necessary to simulate rainfall or evaporation boundary condition and has been elaborated by [11, 23]. It consists of applying a prescribed specific discharge

$q = \widehat{\mathbf{w}} \cdot \mathbf{n}$ along the boundary, where $\mathbf{n}$ is the outward normal unit vector (Eq. 5.1) and the seepage velocity $\widehat{\mathbf{w}}$ is defined as Eq. 5.6,

$$\widehat{\mathbf{w}} = n_L(\mathbf{v}_L - \mathbf{v}_S) \tag{5.6}$$

The application of this boundary condition is based on a predictor-corrector scheme: liquid and solid velocity are first predicted assuming zero-pressure boundary conditions at the infiltration boundary and then (eventually) corrected to ensure the prescribed infiltration rate. The procedure can be summarized in the following steps, also shown in Fig. 5.6:

1 $\mathbf{a}_L^t$ and $\mathbf{a}_S^t$ are computed by solving Eqs. 4.120 and 4.126 assuming $\widehat{p}_L = 0$ at the infiltration boundary
2 Nodal velocities are predicted as $\tilde{\mathbf{v}}_L^{t+\Delta t} = \mathbf{v}_L^t + \mathbf{a}_L \Delta t$ and $\tilde{\mathbf{v}}_S^{t+\Delta t} = \mathbf{v}_S^t + \mathbf{a}_S \Delta t$
3 Infiltration condition is checked. If the net infiltration discharge q_{net} (Eq. 5.7) is positive, ponding conditions occur and if fluid accumulation above the boundary is not allowed (it must remain at zero pressure) no correction is necessary. If the net infiltration discharge is negative, or liquid ponding is allowed above the surface, then liquid and solid velocity must be corrected to ensure the correct infiltration rate and mixture momentum balance

$$q_{net} = (n_L(\tilde{\mathbf{v}}_L^{t+\Delta t} - \tilde{\mathbf{v}}_S^{t+\Delta t}) - \widehat{\mathbf{w}}) \cdot \mathbf{n} \tag{5.7}$$

4 If necessary, the liquid nodal velocity is corrected by Eq. 5.8 and 5.9

$$\mathbf{v}_L^{t+\Delta t} = \tilde{\mathbf{v}}_L^{t+\Delta t} + \Delta \mathbf{v}_L \tag{5.8}$$

$$\mathbf{v}_S^{t+\Delta t} = \tilde{\mathbf{v}}_S^{t+\Delta t} + \Delta \mathbf{v}_S \tag{5.9}$$

where

$$\Delta \mathbf{v}_L = -\frac{(n_L(\tilde{\mathbf{v}}_L^{t+\Delta t} - \tilde{\mathbf{v}}_S^{t+\Delta t}) - \widehat{\mathbf{w}}) \cdot \mathbf{n})\mathbf{n}}{n_L\left(1 + \frac{m_L}{m_S}\right)} \tag{5.10}$$

$$\Delta \mathbf{v}_S = -\frac{m_L \Delta \mathbf{v}_L}{m_S} \tag{5.11}$$

m_L and m_S are the liquid and solid nodal masses.
5 The corrected liquid and solid acceleration is computed as $\mathbf{a}_{L(S)}^t = (\mathbf{v}_{L(S)}^{t+\Delta t} - \mathbf{v}_{L(S)}^t)/\Delta t$
6 The MPM solution scheme can proceed with the convective phase as explained in Sec. 4.6.4.

Note that this boundary condition is applied at nodal level, thus nodal liquid volumetric fraction is necessary. This can be computed by mapping $n_L = nS_L$ from the MPs to the nodes of the mesh.

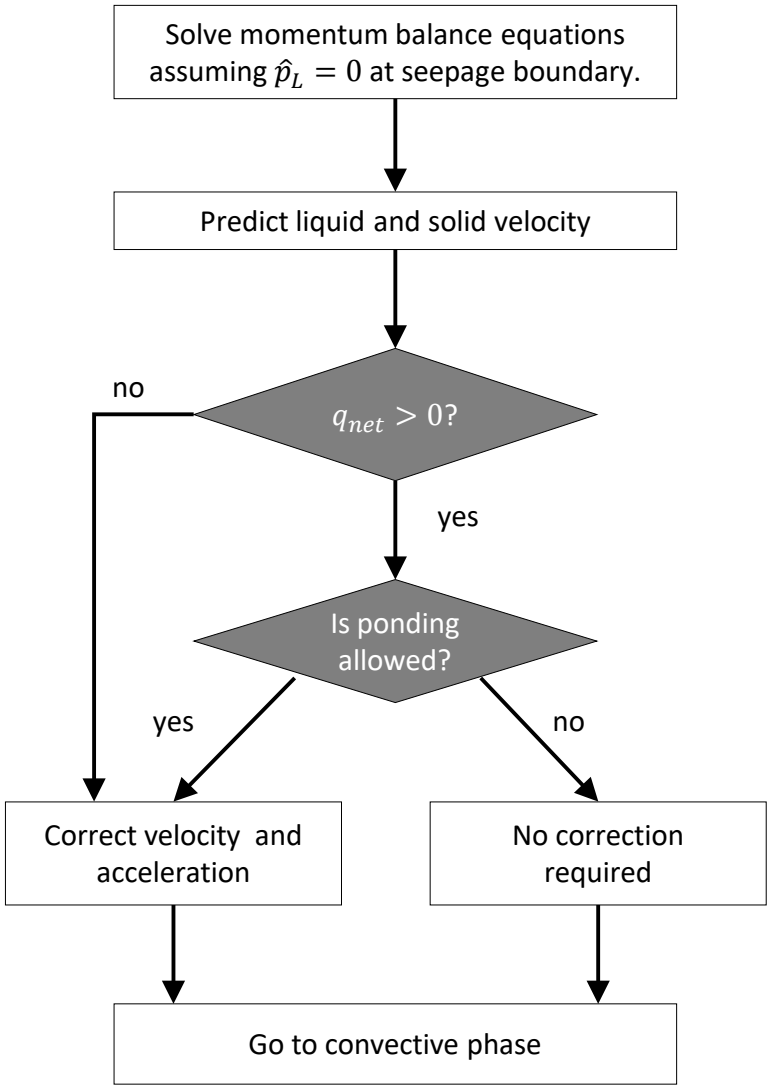


Figure 5.6: *Flow chart of the infiltration boundary condition, from [11].*

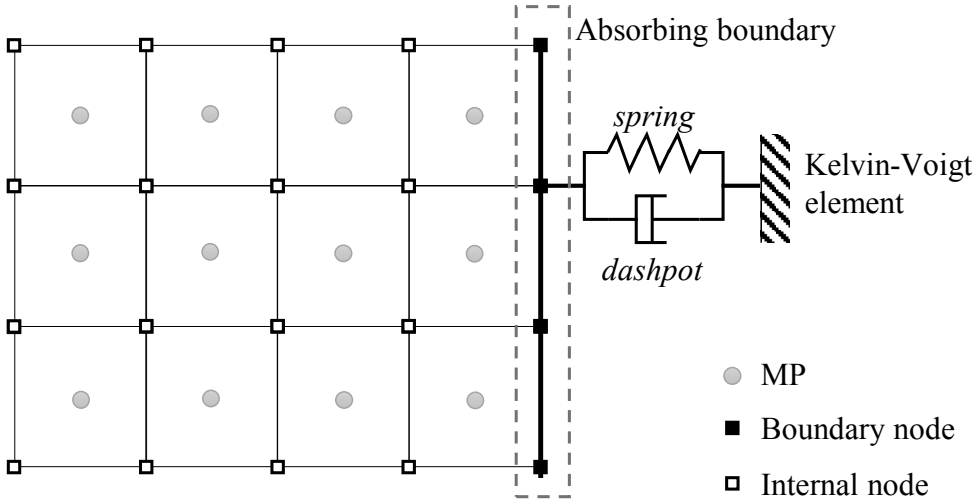


Figure 5.7: *Absorbing boundary with Kelvin-Voigt element.*

5.3.3 Potential seepage face BC

A *potential seepage face* can be defined as an interface between soil and atmosphere where the fluid is free to exit at zero pressure when the soil is saturated, but it cannot enter when the soil is partially saturated. An example of potential seepage face is the downstream surface of a river embankment, or it can also arise after a rapid drawdown of the river level. This boundary condition is necessary where it is unknown if the boundary is an essential or a natural boundary condition.

The implementation can be considered as a special case of the infiltration boundary condition described in Sec. 5.3.2, where $\widehat{\mathbf{w}} = 0$. Liquid and solid nodal velocities are predicted assuming zero-pressure at the potential seepage face (natural boundary condition). If $(n_L(\tilde{\mathbf{v}}_L^{t+\Delta t} - \tilde{\mathbf{v}}_S^{t+\Delta t})) \cdot \mathbf{n} > 0$ it means that fluid is flowing out of the soil at zero pressure and no correction is required. If $(n_L(\tilde{\mathbf{v}}_L^{t+\Delta t} - \tilde{\mathbf{v}}_S^{t+\Delta t})) \cdot \mathbf{n} < 0$, then fluid is flowing into the soil at zero pressure, thus velocity must be corrected with Eqs. 5.8 and 5.9 in which $\widehat{\mathbf{w}} = 0$ (switch to essential boundary condition).

5.4 Absorbing boundaries

The use of finite boundaries produces wave reflections that do not characterise a naturally unbounded domain. Different strategies have been proposed in literature to treat wave reflections. One of the most common approaches in dynamic codes is the use of absorbing boundaries. This method [24] applies viscous damping forces (dashpot) along the artificial boundary. Al-Kafaji [10] observed that the boundary that is supported by dashpots tends to creep continuously, particularly under sustained loading. In order to control creep, dashpots are replaced by Kelvin-Voigt elements, which consider the parallel combination of a spring and a dashpot as shown in Fig. 5.7.

The traction vector corresponding to the absorbing boundary has three components: a component normal to the boundary τ_n^{ab}, and two components in tangential directions $\tau_{t_1}^{ab}$ and $\tau_{t_2}^{ab}$.

The response of the Kelvin-Voigt element is described by Eq. 5.12.

$$\tau_n^{ab} = -a\rho c_p v_n - k_p u_n \tag{5.12a}$$

$$\tau_{t_1}^{ab} = -b\rho c_s v_{t_1} - k_s u_{t_1} \tag{5.12b}$$

$$\tau_{t_2}^{ab} = -b\rho c_s v_{t_2} - k_s u_{t_2} \tag{5.12c}$$

where a and b are dimensionless parameters, v_n, v_{t_1} and v_{t_2} the velocities, u_n, u_{t_1} and u_{t_2} the displacements, ρ the unit mass, c_p and c_s the velocities of the compression and shear waves, respectively. k_p and k_s represent the stiffness per unit area associated to the elastic component.

In an isotropic linear elastic material, the wave velocities (Eq. 5.13) are functions of the constrained modulus E_c, the elastic shear modulus G and the mass density ρ.

$$c_p = \sqrt{\frac{E_c}{\rho}} \quad \text{and} \quad c_s = \sqrt{\frac{G}{\rho}} \tag{5.13}$$

The first term in the right-hand side of Eq. 5.12 represents the traction given by the dashpot, which is proportional to the velocity. The second term represents the traction given by the spring, which is proportional to the displacement. The matrix form of the discretised equation of motion, including the force at the absorbing boundary, reads as Eq. 5.14.

$$\mathbf{Ma} = \mathbf{f}^{\,\text{trac}} - \mathbf{f}^{\,\text{ab}} + \mathbf{f}^{\,\text{grav}} - \mathbf{f}^{\,\text{int}} \tag{5.14}$$

where the contribution of the absorbing boundary $\mathbf{f}^{\,\text{ab}}$ is computed by integrating $\boldsymbol{\tau}^{\text{ab}}$ over the surface $\partial\Omega_{\text{ab}}$ wherever the absorbing boundary is applied. It can then be written in matrix form as Eqs. 5.15 to 5.17.

$$\mathbf{f}^{\text{ab}} = \mathbf{C}^{\text{ab}}\mathbf{v} + \mathbf{K}^{\text{ab}}\mathbf{u} \tag{5.15}$$

$$\mathbf{C}^{\text{ab}} = \int_{\partial\Omega_{\text{ab}}} \mathbf{N}\boldsymbol{\eta}^{\text{ab}}\mathbf{N}^{\mathsf{T}}\,\mathrm{dS} \tag{5.16}$$

$$\mathbf{K}^{\text{ab}} = \int_{\partial\Omega_{\text{ab}}} \mathbf{N}\mathbf{k}^{\text{ab}}\mathbf{N}^{\mathsf{T}}\,\mathrm{dS} \tag{5.17}$$

where the dashpot matrix $\mathbf{C}^{\text{ab}}$ contains the dashpot coefficients (Eq. 5.16), and the spring matrix $\mathbf{K}^{\text{ab}}$ contains the spring coefficients (Eq. 5.17).

The coefficients k_p and k_s can be expressed as functions of the elastic moduli of the material adjacent to the absorbing boundary and a virtual thickness δ as Eq. 5.18.

$$k_p = \frac{E_c}{\delta} \quad \text{and} \quad k_s = \frac{G}{\delta} \tag{5.18}$$

The virtual thickness δ can be interpreted as the thickness of a virtual layer, which extends outside the boundary. Note that for $\delta \rightarrow 0$ the absorbing boundary reduces to a rigid boundary. It reduces to a dashpot boundary for $\delta \rightarrow \infty$.

Since MPs move through the mesh, the scheme is implemented in an incremental form such that displacement and velocity increments of a MP are only accounted for when it enters the considered element (Eqs. 5.19 and 5.20).

$$\mathbf{f}^{ab,t} = \mathbf{f}^{ab,t-1} + \Delta\mathbf{f}^{vb,t} \tag{5.19}$$

$$\Delta\mathbf{f}^{vb,t} = \mathbf{C}^{t}\Delta\mathbf{v} + \mathbf{K}^{t}\Delta\mathbf{u} \tag{5.20}$$

For the two-phase formulation, two sets of Kelvin-Voigt elements are defined. For the liquid phase, the traction of the absorbing boundary has only the normal component and is given by Eq. 5.21.

$$p_L^{ab} = -a_L \rho_L c_L v_{L,n} - k_L u_{L,n} \tag{5.21}$$

where $v_{L,n}$ and $u_{L,n}$ are the normal component of liquid velocity and displacement, respectively. The velocity of the compression wave in the liquid is given by Eq. 5.22.

$$c_L = \sqrt{\frac{K_L}{\rho_L}} \tag{5.22}$$

The stiffness coefficient k_L can be expressed as a function of the bulk modulus of the liquid K_L and the virtual thickness δ_L as Eq. 5.23.

$$k_L = \frac{K_L}{\delta_L} \tag{5.23}$$

Further details and an example can be found in [25].

6 Advanced concepts

6.1 Introduction

Some additional features are described in this chapter. The contact algorithm first, then the local damping for both 1-phase and 2-phase material, and then the mass scaling procedure for quasi-static problems are initially discussed. Lastly, the strategies for mitigation of pathological locking for low order elements is also examined.

6.2 Contact problems

Problems of soil-structure interaction are common in geotechnical engineering. MPM is naturally capable of handling non-slip contact between different bodies, indeed inter-penetration cannot occur because the bodies' velocities belong to the same vector field. However, when continuum bodies come into contact, in most cases frictional sliding occurs at the contact surface. To simulate such a sliding interaction a specific contact algorithm that allows relative motion at the interface between contacting bodies is required.

In this section the contact algorithm proposed by [26] is presented as well as its extension to the adhesive contact. The advantage of this algorithm is that it does not require any special interface element. It proved to be efficient in modeling interaction between solid bodies as well as shearing in granular materials.

The adhesive type of contact is well suited to simulate soil-structure interaction in case of cohesive soil under undrained conditions. Indeed, in this case the tangential force cannot exceed the undrained shear strength.

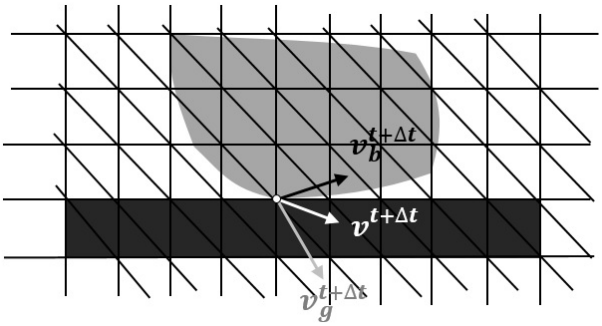


Figure 6.1: *Example of two bodies in contact.*

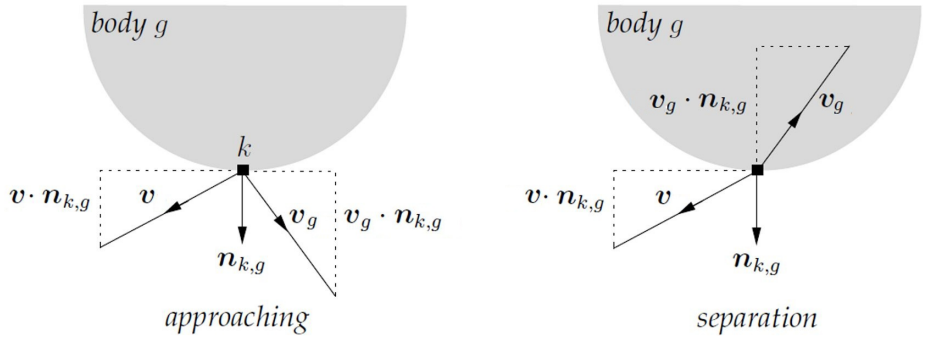


Figure 6.2: *Cases of approaching bodies (left) and separating bodies (right) (Al-Kafaji 2013).*

6.2.1 Formulation

The contact algorithm used in Anura3Dcan be considered 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 the contact law.

Consider body *g* (gray in Fig. 6.1) and body *b* (black in Fig. 6.1), which are in contact at time t. The procedure starts with the initialization of the equation of motion (Eq. 4.20) for each body separately, as well as for the combined system.

The nodal accelerations for each body and the combined system are calculated solving the momentum equations and then used to predict the nodal velocities at time t + Δt as follows:

$$\mathbf{v}_g^{t+\Delta t} = \mathbf{v}_g^t + \Delta t \mathbf{a}_g^t \tag{6.1}$$

$$\mathbf{v}_b^{t+\Delta t} = \mathbf{v}_b^t + \Delta t \mathbf{a}_b^t \tag{6.2}$$

$$\mathbf{v}^{t+\Delta t} = \mathbf{v}^t + \Delta t \mathbf{a}^t \tag{6.3}$$

In this code, the contact surface is predefined by the user during the pre-processing phase. For each contact node, the algorithm proceeds with checking if the bodies are approaching or separating. 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:

$$(\mathbf{v}_g^{t+\Delta t} - \mathbf{v}^{t+\Delta t}) \cdot \mathbf{n}_k^t > 0 \quad \Rightarrow \quad \text{approaching}$$

$$(\mathbf{v}_g^{t+\Delta t} - \mathbf{v}^{t+\Delta t}) \cdot \mathbf{n}_k^t < 0 \quad \Rightarrow \quad \text{separating}$$

where $\mathbf{n}_k^t$ is the unit outward normal to body g at node k. The algorithm allows for free separation, i.e. no correction is required in this case and each body moves with the single body velocity $\mathbf{v}_{g(b)}^{t+\Delta t}$. If the bodies are approaching, than we need to check whether sliding occurs.

The predicted relative normal and tangential velocities at an approaching contact node can be respectively written as:

$$\mathbf{v}_{norm}^{t+\Delta t} = \left[\left(\mathbf{v}_g^{t+\Delta t} - \mathbf{v}^{t+\Delta t} \right) \cdot \mathbf{n}_k^t \right] \mathbf{n}_k^t \tag{6.4}$$

$$\mathbf{v}_{tan}^{t+\Delta t} = \mathbf{n}_k^t \times \left[\left(\mathbf{v}_g^{t+\Delta t} - \mathbf{v}^{t+\Delta t} \right) \times \mathbf{n}_k^t \right] \tag{6.5}$$

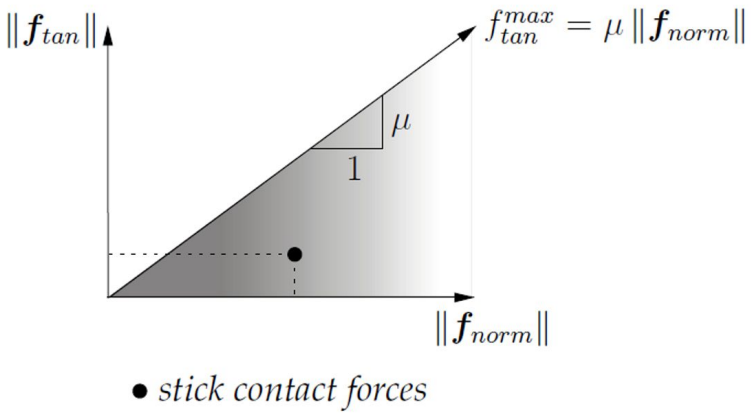


Figure 6.3: *Contact forces for the stick case (Al-Kafaji 2013).*

These components can be used to predict the contact forces at the node as:

$$\mathbf{f}_{norm}^{t+\Delta t} = \frac{m_{k,g}^{t}}{\Delta t}\mathbf{v}_{norm}^{t+\Delta t} \tag{6.6}$$

$$\mathbf{f}_{tan}^{t+\Delta t} = \frac{m_{k,g}^{t}}{\Delta t}\mathbf{v}_{tan}^{t+\Delta t} \tag{6.7}$$

where $m_{k,g}^{t}$ is the nodal mass integrated from MP of body g.

The maximum tangential force is:

$$f_{tan}^{max,t+\Delta t} = f_{adh}^{t+\Delta t} + \mu \left|\mathbf{f}_{tan}^{t+\Delta t}\right| \tag{6.8}$$

where $f_{adh}^{t+\Delta t}$ is the adhesive force at the contact and μ is the friction coefficient.

Depending on the magnitude of the predicted contact forces we can distinguish between stick and slip contact:

$$\text{If} \quad \left|\mathbf{f}_{tan}^{t+\Delta t}\right| < f_{tan}^{max,t+\Delta t} \quad \Rightarrow \quad \text{stick contact}$$

$$\text{If} \quad \left|\mathbf{f}_{tan}^{t+\Delta t}\right| > f_{tan}^{max,t+\Delta t} \quad \Rightarrow \quad \text{slip contact}$$

In the first case, i.e. the bodies stick to each others, no correction is required and the velocity corresponds to $\mathbf{v}^{t+\Delta t}$. In the second case, i.e. the bodies are sliding one respect to the other, the velocity needs to be corrected in such a way that no inter-penetration is allowed and the magnitude of the tangential force respect Eq. 6.8.

The predicted single body velocity $\mathbf{v}_{g}^{t+\Delta t}$ is corrected to a new velocity $\tilde{\mathbf{v}}_{k,g}^{t+\Delta t}$ such that the normal component coincide with the normal component of the combined bodies velocity, i.e.,

$$\tilde{\mathbf{v}}_{g}^{t+\Delta t} \cdot \mathbf{n}_{k}^{t} = \mathbf{v}^{t+\Delta t} \cdot \mathbf{n}_{k}^{t} \tag{6.9}$$

which can also be written as

$$\tilde{\mathbf{v}}_{g}^{t+\Delta t} = \mathbf{v}_{g}^{t+\Delta t} + \mathbf{c}_{norm}^{t+\Delta t} \tag{6.10}$$

where

$$\mathbf{c}_{norm}^{t+\Delta t} = -\left[\left(\mathbf{v}_{g}^{t+\Delta t} - \mathbf{v}^{t+\Delta t}\right) \cdot \mathbf{n}_{k}^{t}\right]\mathbf{n}_{k}^{t} \tag{6.11}$$

is the correction of the normal component of the predicted velocity.

This correction is equivalent to apply the following normal contact force:

$$\tilde{\mathbf{f}}_{norm}^{t+\Delta t} = \frac{m_{k,g}^{t}}{\Delta t}\mathbf{c}_{norm}^{t+\Delta t} \tag{6.12}$$

When sliding occurs, the maximum tangential contact force assumes the expression:

$$\tilde{\mathbf{f}}_{tan}^{t+\Delta t} = f_{tan}^{max,t+\Delta t}\mathbf{t} \tag{6.13}$$

with **t** being a unit vector indicating the direction of the tangent.

Substituting Eq. 6.8 into Eq. 6.13 we get:

$$\tilde{\mathbf{f}}_{tan}^{t+\Delta t} = \left(f_{adh}^{t+\Delta t} + \mu \left|\mathbf{f}_{norm}^{t+\Delta t}\right|\right)\mathbf{t} \tag{6.14}$$

The total contact force is:

$$\tilde{\mathbf{f}}_{cont}^{t+\Delta t} = \tilde{\mathbf{f}}_{tan}^{t+\Delta t} + \tilde{\mathbf{f}}_{norm}^{t+\Delta t} \tag{6.15}$$

which is used to correct the velocity as:

$$\tilde{\mathbf{v}}_{g}^{t+\Delta t} = \mathbf{v}_{g}^{t+\Delta t} + \frac{\tilde{\mathbf{f}}_{cont}^{t+\Delta t}}{m_{k,g}^{t}}\Delta t \tag{6.16}$$

It can also be written as:

$$\tilde{\mathbf{v}}_{g}^{t+\Delta t} = \mathbf{v}_{g}^{t+\Delta t} + \mathbf{c}_{norm}^{t+\Delta t} + \mathbf{c}_{tan}^{t+\Delta t} \tag{6.17}$$

where the correction for the tangential component assumes the form:

$$\mathbf{c}_{tan}^{t+\Delta t} = \frac{\Delta t}{m_{k,g}^{t}}\left(f_{adh}^{t+\Delta t} + \mu \left|\mathbf{f}_{norm}^{t+\Delta t}\right|\right)\mathbf{t} \tag{6.18}$$

The force introduced by the adhesion a can be expressed as;

$$f_{adh}^{t+\Delta t} = aA_{k}^{t} \tag{6.19}$$

where A_k^t is the contact area associated with the node k. It is integrated from the contact elements that share node k.

Substituting Eqs. 6.19 and 6.12 in Eq. 6.18, the corrected velocity can be written as:

$$\tilde{\mathbf{v}}_{g}^{t+\Delta t} = \mathbf{v}_{g}^{t+\Delta t} - \left[\left(\mathbf{v}_{g}^{t+\Delta t} - \mathbf{v}^{t+\Delta t}\right)\cdot\mathbf{n}_{k}^{t}\right]\mathbf{n}_{k}^{t} - \left\{\left[\left(\mathbf{v}_{g}^{t+\Delta t} - \mathbf{v}^{t+\Delta t}\right)\cdot\mathbf{n}_{k}^{t}\right]\mu + \frac{aA_{k}^{t}\Delta t}{m_{k,g}^{t}}\right\}\mathbf{t} \tag{6.20}$$

Fig. 6.4 illustrates with a flow chart the main steps of the implemented contact algorithm.

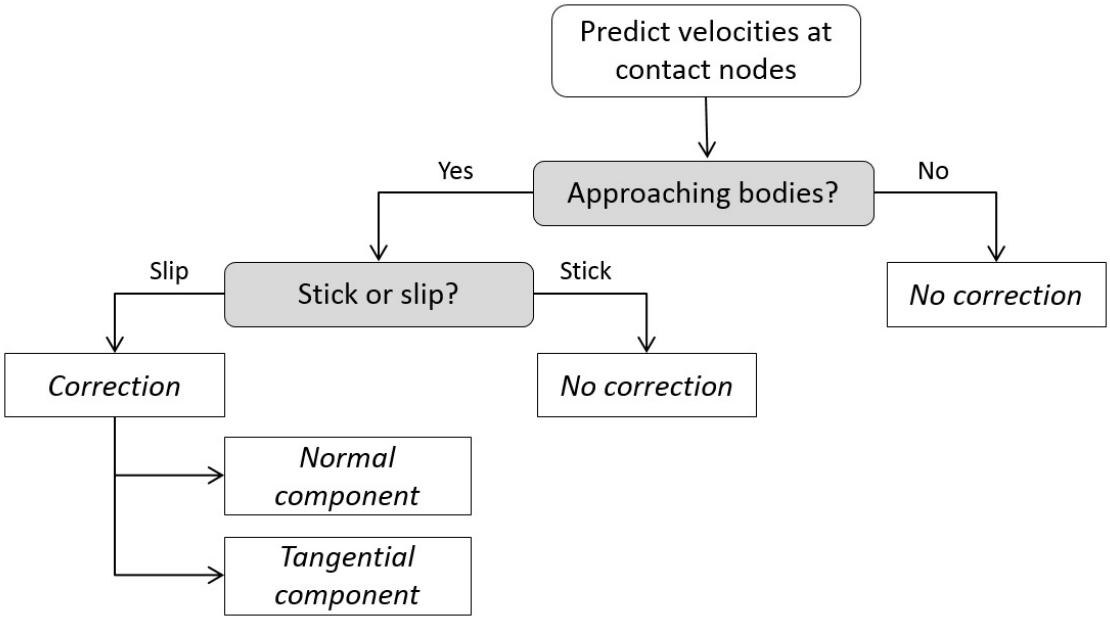


Figure 6.4: *Flow chart illustrating the contact algorithm.*

Having calculated the velocity of the contact node k at time t + Δt, the corrected acceleration vector at the node must be recalculated as:

$$\tilde{\mathbf{a}}_g^t = \frac{\tilde{\mathbf{v}}_g^{t+\Delta t} - \mathbf{v}_g^t}{\Delta t} \tag{6.21}$$

This corrected acceleration is used to update the MP velocity according to the algorithm presented previously. It should be remarked that the contact algorithm is applied between the Lagrangian phase and the convective phase. Indeed the nodal velocities are first predicted in the Lagrangian phase, then the corrected nodal velocities and accelerations are computed by the contact algorithm and these new values of nodal accelerations are used to compute the velocities at the MP in the convective phase. The same procedure explained here for body g must be applied to body b.

6.3 Rigid body

Anura3D uses explicit analysis, and as a result, it is conditionally stable. The stability condition (i.e., critical time step) for one-phase analysis (i.e., dry soil) follows the CFL condition [27], which depends on the bulk modulus of the material into consideration. The larger the elastic modulus, the lower the critical time step. Therefore, the computational cost of simulating rigid intrutions is high. For example, the stiffnes of steel is about to 200,000 times higher than the stiffnes of sand. Simulating rigid intrusors as rigid body save computational time because i) there is no need for integrating stresses on rigid MPs, and ii) the critical time step is independent of the rigid body stiffness. In this way the calculation time is minimized.

The rigid body algorithm is embedded into the contact framework and can work together with the moving mesh algorithm. The proposed algorithm consist of the following steps which are also illustrated in Figure 6.5 [28]:

1 In the Lagrangian phase, the nodes at the contact between the rigid body and surrounding material are identified (Figure 6.6), the sum of internal and external forces is balanced

with the rate of momentum, and the acceleration of the rigid body is obtained as the ratio between the rate of moment and its total mass as follows:

$$\mathbf{a}_{rb} = \frac{\left(m_{rb}\mathbf{g} + \mathbf{t}_{rb} - \sum_{i=1}^{n_c} \mathbf{f}_{soil}^{i}\right)}{m_{rb}} \tag{6.22}$$

where $\mathbf{a}_{rb}$ is the acceleration vector of the rigid body, m_{rb} is the total mass of the rigid body, $\mathbf{g}$ is the gravity acceleration vector, $\mathbf{t}_{rb}$ is the external load vector applied to the rigid body (equal to 0 for this study), n_c is the number of nodes within the contact surface (see Figure 6.6), i is the node relative index within the contact surface. Finally, $\mathbf{f}_{soil}^{i}$ is the force transmitted from the soil to the FFP at the i–th contact node, which is calculated as the i–th nodal internal force only considering the contribution from soil material points. Note that the model analyzed in this work is axisymmetric around the FFP axis; hence, only the vertical component of the reaction force is required.

2 At the end of the Lagrangian phase, the nodal acceleration of the rigid body ($\mathbf{a}_{rb}^{i}$) is updated based on the acceleration of the rigid body calculated in the previous step ($\mathbf{a}_{rb}$):

$$\mathbf{a}_{rb}^{i} = \mathbf{a}_{rb} \qquad \text{for: } 1 \leq i \leq n_{rb} \tag{6.23}$$

where i denotes the node number within a total of n_{rb} nodes (blue and red zone in Figure 6.6)

3 The nodal velocity of the rigid body ($\mathbf{v}_{rb}^{i}$) is calculated as:

$$\mathbf{v}_{rb}^{i} = \mathbf{v}_{0}^{i} + \mathbf{a}_{rb}^{i}\Delta t \qquad \text{for: } 1 \leq i \leq n_{rb} \tag{6.24}$$

with $\mathbf{v}_{0}^{n_i}$ being the velocity at the beginning of each time-step, $\mathbf{a}_{rb}^{n_i}$ is the current nodal acceleration, and Δt is the time-step increment. Note that in this manner, the rigid motion is ensured in all the nodes of the rigid body.

4 During the convective phase, the velocity and acceleration of the material points in the rigid body are updated based on the nodal values. Finally, the position of the material points is updated accordingly.

In this paper, the rigid body algorithm is used together with the contact algorithm proposed by [26]. To account for relative displacement and free separation of different bodies, the contact algorithm requires the evaluation of different acceleration and velocity fields at the nodes: one for each body ($\mathbf{v}_{soil}$, $\mathbf{a}_{soil}$, $\mathbf{v}_{rb}$, and $\mathbf{a}_{rb}$) and one for the system ($\mathbf{v}_{sys}$, $\mathbf{a}_{sys}$) (i.e., combined body fields) [26]. In order to ensure that the contact algorithm is compatible with the rigid body motion, the acceleration and velocity obtained in steps 2 and 3 must be prescribed to the system (i.e., $\mathbf{v}_{sys} = \mathbf{v}_{rb}$ and $\mathbf{a}_{sys} = \mathbf{a}_{rb}$) before the contact algorithm is applied (Figure 6.5).

Note that the rigid body algorithm proposed here is only valid for rigid body linear motions since we only consider the linear momentum balance. This could be extended to simulate rigid body rotation by solving the angular momentum.

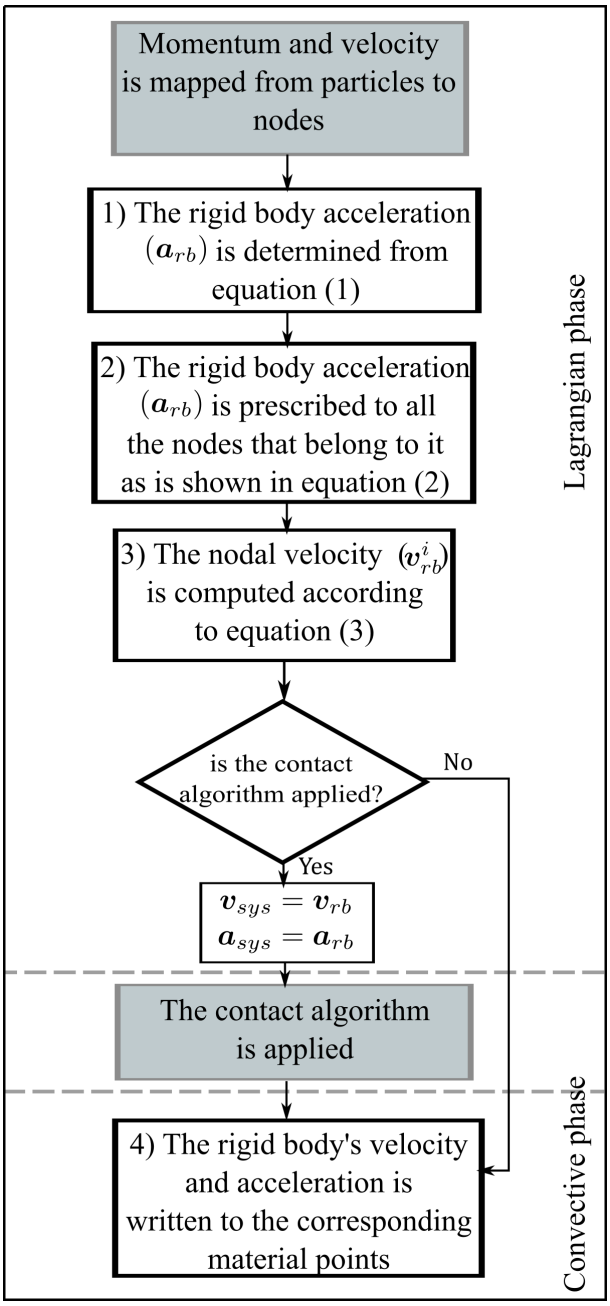


Figure 6.5: *Rigid body algorithm flow diagram. The shaded rectangles are processes that run independently of the rigid body algorithm. MPM phases and the contact algorithm are included for reference.*

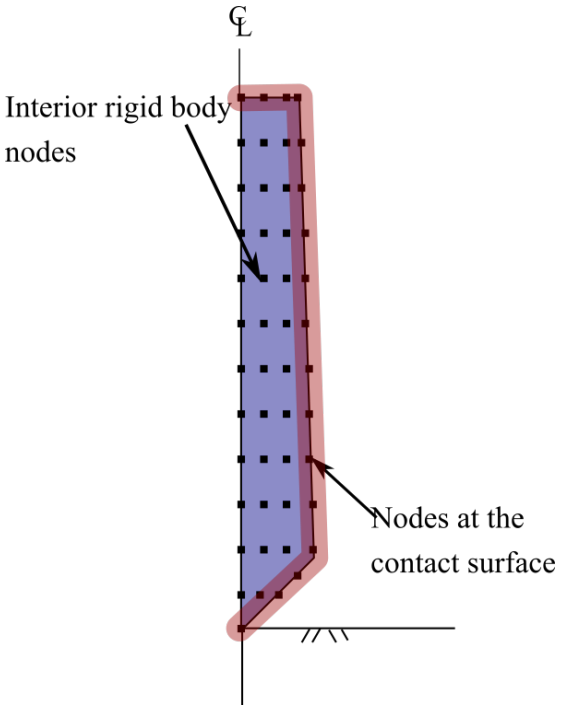


Figure 6.6: *Nodes belonging to the rigid intrusor (blue and red zone), and nodes belonging to the contact surface (red zone).*

6.4 Local damping

Natural dynamic systems contain some degree of damping of the vibration energy within the system; otherwise, the system would oscillate indefinitely when subjected to driving forces. Damping is due, in part, to energy loss as a result of internal friction in the intact material and slippage along interfaces, if these are present.

6.4.1 Single-phase problem

For a dynamic analysis, the damping in the numerical simulation should reproduce in magnitude and form the energy losses in the natural system when subjected to a dynamic loading. In soil and rock, natural damping is mainly hysteretic, i.e. independent of frequency, see [29] and [30]. This type of damping is difficult to reproduce numerically. However, if a constitutive model is found that contains an adequate representation of the hysteresis that occurs in a real material, then no additional damping would be necessary [31].

Rayleigh damping is commonly used to provide damping that is approximately frequency-independent over a restricted range of frequencies. However, this kind of damping introduces body forces that hinder the steady state collapse and might influence the mode of failure [32]. [33] describes a local non-viscous damping to overcome the difficulty associated with the viscous damping.

An alternative to Rayleigh damping is the so-called local damping. The local damping force is proportional to the out of balance force $f = f^{ext} - f^{int}$ and acts opposite to the direction of the velocity. For any degree-of-freedom in the considered system, the local damping can be described as follows

$$ma = f + f^{damp} \tag{6.25}$$

where

$$f^{damp} = -\alpha \, |f| \, \frac{v}{|v|} \tag{6.26}$$

is the damping force at the considered degree of freedom. The dimensionless parameter α is called local damping factor.

6.4.2 Two-phase problem

In the two-phase formulation the fluid and the solid phase are damped separately. At any degree of freedom the momentum equation for the water assumes the form:

$$m_w a_w = f_w^{trac} + f_w^{grav} - f_w^{int} - f_w^{drag} + f_w^{damp} \tag{6.27}$$

if $f_w = f_w^{trac} + f_w^{grav} - f_w^{int}$ is the unbalanced force for the water, the damping force for the water can be written as:

$$f_w^{damp} = -\alpha_w \, |f_w| \, \frac{w}{|w|} \tag{6.28}$$

The momentum equation for the mixture is:

$$m_s a_s = -\bar{m}_w a_w + f^{trac} + f^{grav} - f^{int} + f^{damp} \tag{6.29}$$

where

$$f^{damp} = f_s^{damp} + f_w^{damp} \tag{6.30}$$

The out of balance force for the solid phase is $f = f^{trac} + f^{grav} - f^{int}$, hence the damping force for the solid can be written as:

$$f_s^{damp} = -\alpha_s \, |f - f_w| \, \frac{v}{|v|} \tag{6.31}$$

In this manual the local damping factor for the fluid (α_w) and the solid (α_s) phase always assume the same value.

Local damping was originally designed for static simulations. However, it has some characteristics that make it attractive for dynamic simulations if proper values of the damping coefficient are used. In quasi-static problems high value of α, i.e. 0.7-0.8, can be used to accelerate convergence. In slow-process problems a small value of α, i.e. 0.05-0.15, can simulate natural energy dissipation of the material, if it is not taken into account by the constitutive model. Local damping should not be used or used only with high care in highly dynamic problem, where wave propagation is of great importance.

6.5 Convergence to static equilibrium

The dynamic MPM can also be adopted to model quasi-static problems, therefore, a criteria for the detection of the static equilibrium is required. The 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 F as

$$F = \frac{\|F_{ext} - F_{int}\|}{\|F_{ext}\|} \tag{6.32}$$

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

$$E = \frac{KE}{W_{ext}} \tag{6.33}$$

where KE denotes the kinetic energy of the system and W_{ext} is the work induced by the external forces. The system is assumed to have reached static equilibrium when F and E reach a pre-defined tolerance.

6.6 Mass scaling for quasi-static problems

The simulation of quasi-static or slow-process problems with a dynamic explicit code can require high computational effort because the time step size is bounded by the stability condition. However, if the inertia effect is negligible, then the time step size can be artificially increased by scaling the density. Introducing the mass scaling factor β, the critical time step size increases by a factor $\sqrt{\beta}$. Indeed:

$$\Delta t^{\beta}_{crit} = \frac{l_e}{\sqrt{\frac{E_c}{\beta\rho}}} = \sqrt{\beta}\Delta t^{1}_{crit} \tag{6.34}$$

where Δt^{1}_{crit} is the critical time step for $\beta = 1$, i.e. no mass scaling is applied.

Mass scaling is a very useful technique to improve computational efficiency of dynamic codes in simulating quasi-static ans slow-process motion. However, sensitivity analysis is necessary to calibrate the mass scaling factor in slow process problems, indeed extremely high values of β can significantly effect the result.

6.7 Cell-crossing instability and *MPM-MIXED*

The classical MPM approach suffers from a "cell-crossing instability" for problems involving large displacements. Whenever material points cross boundaries of any element in the computational background grid, an unphysical unbalance force appears at the nodes that are shared between previous and new elements of that crossing points. The explanation for that is the lack of smoothness of the nodal shape functions used in the interpolation of information between grid and material points. The shape functions are linear C^0, which means that the gradients of the shape functions within the element are constant but they are discontinuous at the element borders (they change the sign). These gradients are used in the calculation of the nodal internal forces as described in Section 4.3.3. Therefore, when one material point crosses a border, the internal forces of those involved nodes suffer an unphysical instability due to a jump discontinuity in the gradient of linear shape functions.

In this code there is a procedure to mitigate the cell-crossing instability. This simple procedure arises from calculating the stress of each element as a constant value that corresponds to the average of the stresses of the material points that at time t^k are located within the element. Based on that, Gauss integration is adopted to determine the internal forces (as in FEM), in which a single point with an averaged stress $\boldsymbol{\sigma}_{av}$ is considered in each element. The stress averaging is calculated according to the following expression (Eq. 6.35), where $no_{MP,el}$ is the number of material points within the element, and $\boldsymbol{\sigma}_{MP}$ and Ω_{MP} are the stress and the

volume of a material point, respectively.

$$\boldsymbol{\sigma}_{av} = \frac{\sum_{MP=1}^{no_{MP,el}} \boldsymbol{\sigma}_{MP}\, \Omega_{MP}}{\sum_{MP=1}^{no_{MP,el}} \Omega_{MP}} \tag{6.35}$$

The calculation of the internal forces by means of the Gauss point integration is only considered in the elements located in the interior of the continuum. The Gauss point integration is also used for the elements located at the boundaries only if the following condition is fulfilled:

$$\sum_{MP=1}^{no_{MP,el}} \Omega_{MP} > F_{fill}\, \Omega_{el} \tag{6.36}$$

in which the factor F_{fill} is set to 0.9. It the aforementioned condition (Eq. (6.36)) is not fulfilled, the internal force in the boundary elements is calculated with the classic MPM procedure. This is a mixed approach which uses material points (MP) and Gauss points to calculate the internal force, and in the current code is named *MPM-MIXED*.

6.8 Pathological locking in low order elements

6.8.1 Strain smoothing procedure

The formulation of the MPM is very close to the FEM in many aspects. At some point, this similarity is beneficial because most of the existing knowledge can be transferred to the MPM. However, several shortcomings associated with the FEM have been inherited by the MPM. Probably one of the most important limitation is the kinematic locking that can occur due to the linearity of the traditional shape functions constructed on the computational mesh. Difficulties arise when determining the displacement field for a nearly incompressible solid.

As a simple example of this phenomenon, consider Fig. 6.7. Element e_1, in Fig. 6.7a, is defined by nodes 1 and 2 on the x axis, and node 3 on the y axis. The area of the triangular element must remain constant if it is incompressible. If nodes 1 and 2 are fixed, y_3 must remain constant and $u_y^3 = 0$. Therefore, the remaining degree of freedom is the horizontal displacement u_x^3. Similarly, the only remaining degree of freedom in the element e_2 (Fig. 6.7b) defined by nodes 4, 5, and 6, is the vertical displacement u_y^5. Two triangles may be assembled, see Fig. 6.7c. Since incompressibility for element e_1 requires $u_y^3 = 0$ and incompressibility for element e_2 requires $u_x^3 = 0$, node 3 cannot move, and both elements are completely locked up. With nodes 1 through 4 locked up, the nodes for elements 3, 4, 5, 6, 7 and 8 will also be locked (Fig. 6.7c). Such locking usually propagates throughout the entire mesh yielding an unrealistic stiff response and an erroneous velocity field.

Several anti-locking approaches have been presented by different authors to mitigate this numerical problem associated with the linear shape functions [34, 35]. In this work, the Nodal Mixed Discretization (NMD) technique for linear tetrahedra elements presented by [35] has been used to mitigate the over-stiff behaviour. This procedure has showed effectiveness in mitigating the locking associated with (near) incompressible deformations [10]. In this technique, the element volumetric behaviour is averaged over the elements sharing its nodes via a least squares smoothing process. The effect of applying the NMD scheme is to increase the number of degrees of freedom per element. Being $\dot{\varepsilon}$ the strain rate of an element calculated

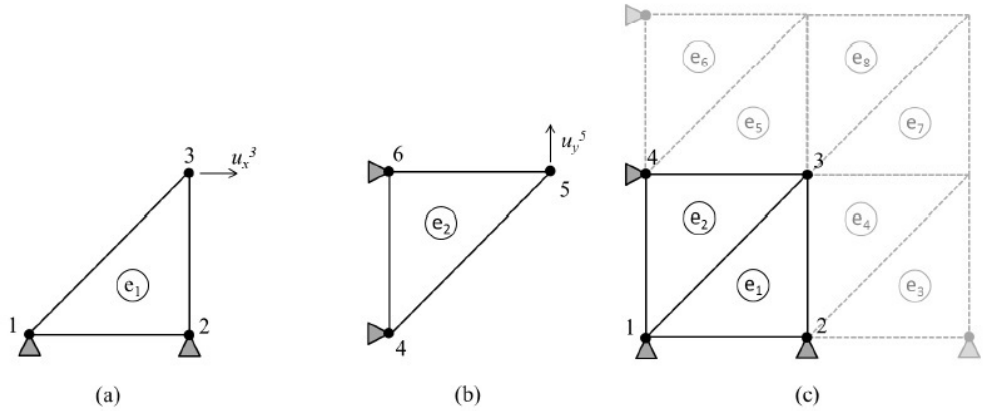


Figure 6.7: *A pathological case of volumetric locking in triangular elements. Degrees of freedom are indicated.*

from nodal velocities, it can be divided into deviatoric $\dot{\mathbf{e}}$ and volumetric $\dot{\varepsilon}_{vol}$ components as

$$\dot{\boldsymbol{\varepsilon}} = \dot{\mathbf{e}} + \dot{\varepsilon}_{vol}\boldsymbol{\delta} \tag{6.37}$$

where $\boldsymbol{\delta} = (1, 1, 1, 0, 0, 0)^T$.

The volumetric strain rate for a node i, can be defined as weighted average of the surrounding element values $\dot{\varepsilon}_{vol,e}$ with the following equation:

$$\dot{\varepsilon}_{vol,i} = \frac{\sum_{el=1}^{no_{el,i}} \dot{\varepsilon}_{vol,el}\Omega_{el}}{\sum_{el=1}^{no_{el,i}} \Omega_{el}} \tag{6.38}$$

where $no_{el,i}$ are the elements surrounding the node i, and Ω_{el} is the volume of the element el.

Then, a mean value for the element, $\bar{\dot{\varepsilon}}_{vol}$ is calculated by taking the average of nodal quantities as

$$\bar{\dot{\varepsilon}}_{vol} = \frac{1}{no_{node,el}} \sum_{i=1}^{no_{node,el}} \dot{\varepsilon}_{vol,i} \tag{6.39}$$

where $no_{node,el}$ is the number of nodes in an element.

Finally, the element strain rates is redefined by superposition of the deviatoric part and the volumetric average as

$$\dot{\boldsymbol{\varepsilon}} = \dot{\mathbf{e}} + \bar{\dot{\varepsilon}}_{vol}\boldsymbol{\delta} \tag{6.40}$$

6.8.2 Pore pressure increment smoothing

In unsaturated soils, the pore pressure is calculated incrementally according to 3.37, which can be rewritten as:

$$\frac{dp_L}{dt} = \widetilde{K}_L \text{div}\left[\rho_L n S_L(\mathbf{v}_S - \mathbf{v}_L)\right] - \rho_L S_L \text{div}(\mathbf{v}_S) \tag{6.41}$$

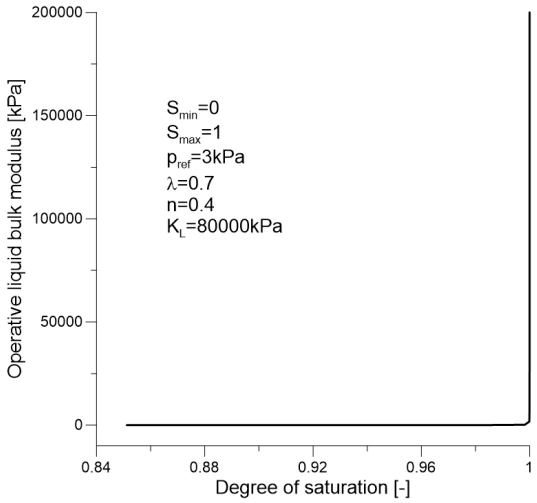


Figure 6.8: *Bulk modulus variation with degree of saturation.*

Where

$$\widetilde{K}_L = \left[n \left(S_L \frac{\partial \rho_L}{\partial p_L} + \rho_L \frac{\partial S_L}{\partial p_L} \right) \right]^{-1} \tag{6.42}$$

is an operative bulk modulus depending on the derivative of the SWRC, bulk modulus of the liquid, porosity, and degree of saturation. When the degree of saturation is close to 1, the operative bulk modulus increases very rapidly, see Fig. 6.8, generating large pressure oscillations. To mitigate these oscillations, a pressure increment smoothing procedure is implemented.

With this feature, the liquid pressure increments are initially computed at the MP and then averaged at the nodal level with a similar procedure used in Sec. 6.8. An average nodal liquid pressure increment is computed:

$$\Delta p_N = \frac{\left(\sum_{Elem} \sum_{MP} \Delta p_{Lp} * \Omega_p \right)}{\sum_{Elem} \sum_{MP} \Omega_p} \tag{6.43}$$

Where $\sum_{Elem}$ runs over the elements that share node N and $\sum_{MP}$ loops over the integration points of the element. For each element, an average pressure increment is computed averaging the nodal contributions using Eq. 6.44 where NEleNodes is the number of nodes of the element.

$$\Delta p_{El} = \frac{\sum_{NEleNodes} \Delta p_N}{NElementNodes} \tag{6.44}$$

This average pressure increment Δp_{El} is assigned to each MP inside the element.

6.9 Liquid free surface detection

The free surface problem is an important topic in coastal applications and many engineering problems. The expression for free surface is due to the contact of two fluids having different densities where free surface is developed at the interface. It is important to detect this surface

properly. In most applications where single fluid flow is involved, a fluid experiences a free surface as exposed to the atmospheric pressure, where the air pressure corresponds to gauge pressure. The important feature of this type of problems that the shape of the surface is unknown, a priori, as it depends on the developed flow.

In MPM, mass density field can be evaluated at nodes each time increment. In this approach, the density field is represented at the grid nodes using the following formula

$$\rho_{L,i} = \frac{\sum_{el=1}^{no_{el,i}} \sum_{MP=1}^{no_{MP,el}} N_i(\xi_{MP})\, m_{MP}}{\sum_{el=1}^{no_{el,i}} \frac{\Omega_{el}}{no_{nodes,el}}} \tag{6.45}$$

It is important to highlight here that the denominator of Eq. 6.45 involves only the non–empty elements, so that the density is approximated properly.

Consequently, the density at any location **x** inside elements can be interpolated using

$$\tilde{\rho}_{L,MP} = \sum_{i=1}^{no_{node,el}} N_i(\xi_{MP})\, \rho_{L,i} \tag{6.46}$$

In which the $\tilde{\rho}_{L,MP}$ is the interpolated liquid density field. Using this procedure, the interpolated density $\tilde{\rho}_{L,MP}$ is evaluated for all material points, which is used in the current MPM formulation to capture the free surface; i.e.

$$\tilde{\rho}_{L,MP} \leq F_{FreeSurf}\, \rho_{L,0} \tag{6.47}$$

where $\rho_{L,0}$ is the reference value of the liquid density and $F_{FreeSurf}$ is the factor that controls the continuity of the free surface. The bigger this value is, but always smaller than 1, the greater the number of particles detected. This factor is mesh dependent and needs to be set accordingly. The value suggested for this factor is 0.7.

6.10 Moving mesh

The moving mesh procedure exploits the fact that the computational grid in MPM does not store any permanent information. Therefore, it can be kept fixed or be freely modified at the end of each time step. This approach is useful when applying non-zero traction and kinematic boundary conditions because it ensures that the computational mesh aligns with the surface where tractions (kinematics) are prescribed [36]. It is particularly convenient when simulating the movement of a body inside the soil such as in pile driving or anchor installation where it is necessary to keep a refined mesh always near the structure [37–41].

The positions of the grid nodes are modified according to predefined rules after having updated stress, strain, velocity and position of MPs at the end of the convective phase. The MPs are assigned to the elements and the computation proceeds with a new time step.

The discretised domain is divided as follows with the moving mesh procedure.

Moving mesh portion of the grid whose elements do not deform along the simulation but move rigidly with certain predefined rules.

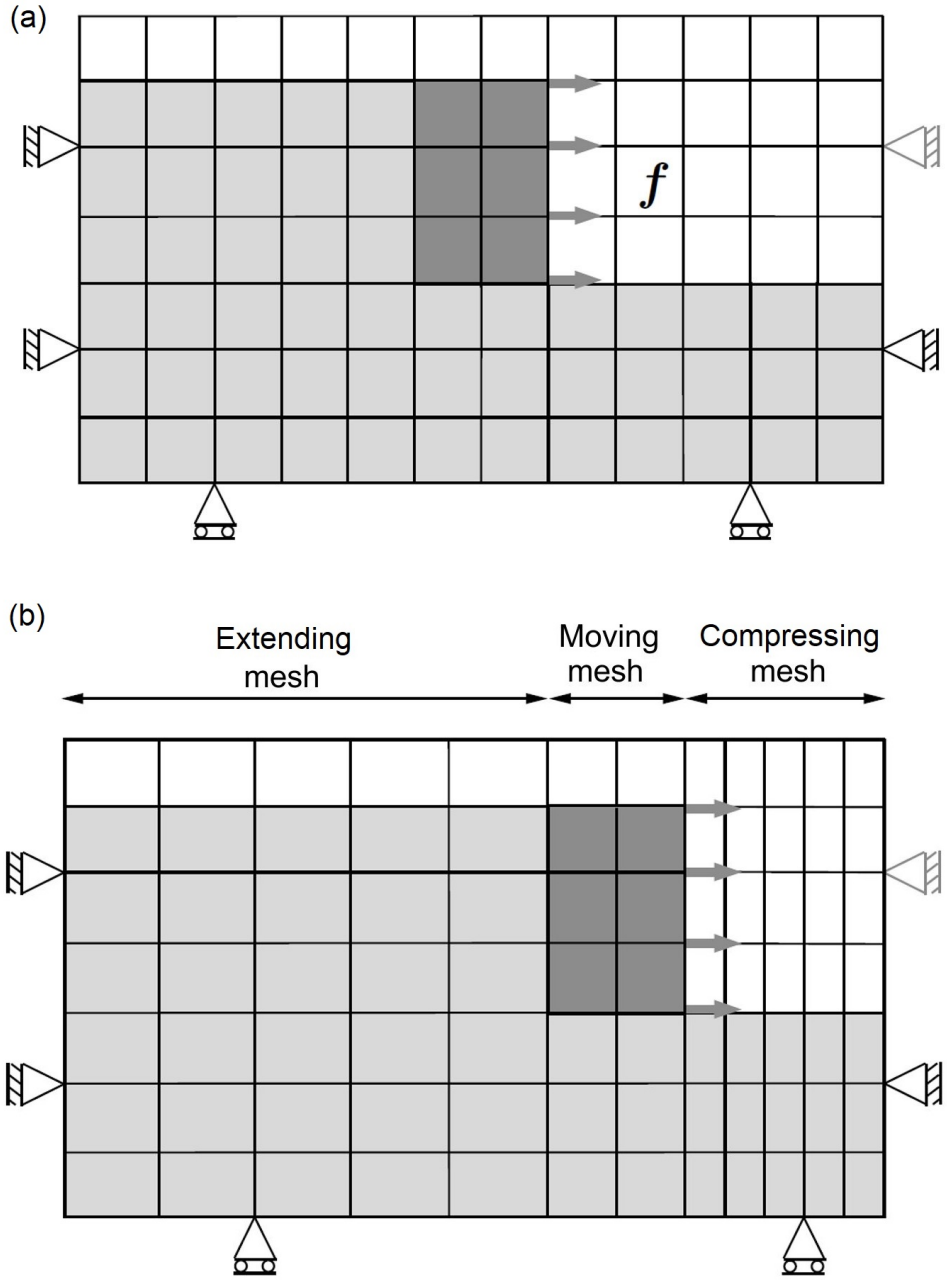
Deforming mesh portion of the grid located between the moving mesh and the fix boundary of the domain that modifies its dimensions with time. Thus, its elements deform accordingly.

All the elements of the discretised domain must be associated to only one of the aforementioned parts. In general, the moving mesh portion can translate in one or more directions and rotate rigidly following predefined rules, but in Anura3D 2021it is limited to translation in one direction.

The moving mesh usually contains a rigid body and it moves with the same displacement of this body, thus keeping the boundary of the elements aligned to the boundary of this structure. This is convenient when the contact algorithm is applied because it prevents the need for identifying the contact surface and computing the normal directions at each time increment. It is also possible to attach the moving mesh to a surface over which a traction is applied and that moves with a prescribed velocity.

Because of the movement of the moving mesh zone, the elements of the deforming mesh zones change their aspect ratio with time. It is important to choose a discretisation that ensures a reasonable aspect ratio of all the elements during the entire simulation.

Further detail and an example can be found in [42].


(a)
f
(b)
Extending mesh
Moving mesh
Compressing mesh

References

[1] Francis H Harlow. The particle-in-cell computing method for fluid dynamics. *Methods in computational physics*, 3(3):319–343, 1964.

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

[3] Deborah Sulsky, Shi-Jian Zhou, and Howard L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1-2):236–252, May 1995. ISSN 00104655. doi: 10.1016/0010-4655(94)00170-7. URL http://linkinghub.elsevier.com/retrieve/pii/0010465594001707.

[4] D Sulsky and HL Schreyer. Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems. *Computer Methods in Applied Mechanics and Engineering*, 0457825(96), 1996. URL http://www.sciencedirect.com/science/article/pii/S0045782596010912.

[5] A. Yerro. *MPM modelling of landslides in brittle and unsaturated soils*. PhD Thesis, Universitat Politècnica de Catalunya (UPC Barcelona), Spain, 2015.

[6] OC 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 Geomechanicsnumerical and analytical methods . . .*, 8:71–93, 1984. URL http://onlinelibrary.wiley.com/doi/10.1002/nag.1610080106/abstract.

[7] 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. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 429(1877):285–309, June 1990. ISSN 1364-5021. doi: 10.1098/rspa.1990.0061. URL http://rspa.royalsocietypublishing.org/cgi/doi/10.1098/rspa.1990.0061.

[8] Olgierd C Zienkiewicz, AHC Chan, M Pastor, BA Schrefler, and T Shiomi. *Computational geomechanics*. Wiley Chichester, 1999.

[9] J Van Esch, Dieter Stolle, Issam Jassim, and J M van Esch. Finite element method for coupled dynamic flow-deformation simulation. *2nd International Symposium on . . .*, (1), 2011. URL http://www.uni-stuttgart.de/igs/content/publications/218_Issam_Dubrovnik.pdfhttp://www.researchgate.net/publication/259579508_FINITE_ELEMENT_METHOD_FOR_COUPLED_DYNAMIC_FLOW-DEFORMATION_SIMULATION/file/e0b4952cbd16168e83.pdf.

[10] I.K.J. Al-Kafaji. *Formulation of a dynamic material point method (MPM) for geomechanical problems*. PhD Thesis, Institut für Geotechnik der Universität Stuttgart, Germany, 2013.

[11] Francesca Ceccato, Alba Yerro, Veronica Girardi, and Paolo Simonini. Two-phase dynamic mpm formulation for unsaturated soil. *Computers and Geotechnics*, 129:103876, 2021.

[12] D. Hillel. *Soil and water physical principles and processes*. Academic press, London (UK), 1971.

[13] Yechezkel Mualem. Hysteretical models for prediction of the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(6):1248–1254, dec 1976. doi: 10.1029/WR012i006p01248. URL http://doi.wiley.com/10.1029/WR012i006p01248.

[14] Jeremiah U Brackbill, Douglas B Kothe, and Hans M Ruppel. Flip: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1):25–38, 1988.

[15] P. A. Vermeer. *PLAXIS 2D Reference Manual Version 5*. Balkema, Rotterdam / Brookfield, 1993.

[16] I. Jassim, D. Stolle, and P.A. Vermeer. Two-phase dynamic analysis by material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(15):2502–2522, 2013. doi: 10.1002/nag.2146.

[17] A. Yerro, E.E. Alonso, and N.M. Pinyol. The material point method for unsaturated soils. *Géotechnique*, 65(3):201–217, 2015. doi: 10.1680/geot.14.P.163.

[18] M.M.J. Mieremet. Numerical stability for velocity-based 2-phase formulation for geotechnical dynamic analysis. Report 15-03, ISSN 1389-6520, Reports of the Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands, 2015.

[19] M.M.J. Mieremet. *Numerical stability for velocity-based 2-phase formulation for geotechnical dynamic analysis*. MSc Thesis, Applied Mathematics, Delft University of Technology, Delft, The Netherlands, 2015.

[20] E. Alonso and F. Zabala. Progressive failure of Aznalcóllar dam using the material point method. *Géotechnique*, 61(9):795–808, 2011. doi: 10.1680/geot.9.P.134.

[21] 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, July 2006. ISSN 00457825. doi: 10.1016/j.cma.2005.06.027. URL http://linkinghub.elsevier.com/retrieve/pii/S0045782505003592.

[22] Veronica Girardi, Alba Yerro, Francesca Ceccato, and Paolo Simonini. Modelling deformations in water retention structures with unsaturated material point method. *Proceedings of the Institution of Civil Engineers-Geotechnical Engineering*, 174(5):577–592, 2021.

[23] Mario Martinelli, Wei-Lin Lee, Chjeng-Lun Shieh, and Sabatino Cuomo. Rainfall Boundary Condition in a Multiphase Material Point Method. In *Fifth World Landslide Forum*, pages 303–309. 2021. doi: 10.1007/978-3-030-60706-7_29.

[24] John Lysmer and R. L. Kuhlemeyer. Finite dynamic model for infinite media. In *Proc. of ASCE*, pages 859–877, 1969.

[25] F. Ceccato and P. Simonini. Numerical Features used in Simulations. In E.J. Fern, A. Rohe, K. Soga, and E.E. Alonso, editors, *Mater. Point Method Geotech. Eng. A Pract. Guid.*, chapter 6, pages 101–123. CRC Press, London, 2019. ISBN 1138323314.

[26] 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. *Computer Modeling in Engineering and Sciences*, 2: 509–522, 2001.

[27] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11:215–234, 1967.

[28] Luis Zambrano-Cruzatty and Alba Yerro. Numerical simulation of a free fall penetrometer deployment using the material point method. *Soils and Foundations*, 60(3):668–682, June 2020. ISSN 0038-0806. doi: 10.1016/j.sandf.2020.04.002. URL http://www.sciencedirect.com/science/article/pii/S0038080620336179.

[29] Andrew Gemant and Willis Jackson. Xciii. the measurement of internal friction in some solid dielectric materials. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(157):960–983, 1937.

[30] RL Wegel and H Walther. Internal dissipation in solids for small cyclic strains. *Journal of Applied Physics*, 6(4):141–157, 1935.

[31] PA Cundall. *FLAC Manual: A Computer Program for Fast Lagrangian Analysis of Continua*, 2001.

[32] R Hart, PA 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. In *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, volume 25, pages 117–125. Elsevier, 1988.

[33] PA Cundall. Distinct element models of rock and soil structure. *Analytical and computational methods in engineering rock mechanics*, 4:129–163, 1987.

[34] C.M. Mast, P. Mackenzie-Helnwein, P. Arduino, G.R. Miller, and W. Shin. Mitigating kinematic locking in the material point method. *Journal of Computational Physics*, 231 (16):5351–5373, June 2012. ISSN 00219991. doi: 10.1016/j.jcp.2012.04.032. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999112002148.

[35] C Detournay and E Dzik. Nodal mixed discretization for tetrahedral elements. In *4th international FLAC symposium, numerical modeling in geomechanics. Minnesota Itasca Consulting Group, Inc. Paper*, number 07-02, 2006.

[36] L. Beuth. *Formulation and application of a quasi-static material point method*. PhD Thesis, Institut für Geotechnik der Universität Stuttgart, Germany, 2012.

[37] F. Ceccato, L. Beuth, P.A. Vermeer, and P. Simonini. Two-phase material point method applied to the study of cone penetration. *Computers and Geotechnics*, 80:440–452., 2016. doi: 10.1016/j.compgeo.2016.03.003.

[38] N.T.V. Phuong, A.F. Van Tol, A.S.K. Elkadi, and A. Rohe. Numerical investigation of pile installation effects in sand using material point method. *Computers and Geotechnics*, 73:58–71, 2016. doi: 10.1016/j.compgeo.2015.11.012.

[39] Francesca Ceccato, Alberto Bisson, and Simonetta Cola. Large displacement numerical study of 3D plate anchors Large displacement numerical study of 3D plate anchors. *European Journal of Environmental and Civil Engineering*, 2017. ISSN 1964-8189. doi: 10.1080/19648189.2017.1408498. URL https://doi.org/10.1080/19648189.2017.1408498.

[40] Mario Martinelli and Vahid Galavi. Investigation of the material point method in the simulation of cone penetration tests in dry sand. *Computers and Geotechnics*, 130:103923, 2021.

[41] Kaleigh M Yost, Alba Yerro, Russell A Green, Eileen Martin, and Jon Cooper. Mpm modeling of cone penetrometer testing for multiple thin-layer effects in complex soil stratigraphy. *Journal of Geotechnical and Geoenvironmental Engineering*, 148(2):04021189, 2022.

[42] F. Ceccato and P. Simonini. Cone Penetration Tests. In E.J. Fern, A. Rohe, K. Soga, and E.E. Alonso, editors, *Mater. Point Method Geotech. Eng. A Pract. Guid.*, chapter 18, pages 311–324. CRC Press, London, 2019. ISBN 1138323314.