

Article

Comparison of a Material Point Method and a Galerkin Meshfree Method for the Simulation of Cohesive-Frictional Materials

Ilaria Iaconeta ^{1,2,*}, Antonia Larese ^{1,2}, Riccardo Rossi ^{1,2} and Zhiming Guo ²

¹ International Center for Numerical Methods in Engineering (CIMNE), Edificio C1, Campus Norte, Jordi Girona 1-3, 08034 Barcelona, Spain; antoldt@cimne.upc.edu (A.L.); rrossi@cimne.upc.edu (R.R.)

² Department of Civil and Environmental Engineering (DECA), Technical University of Catalonia (UPC), Edificio C1, Campus Norte, Jordi Girona 1-3, 08034 Barcelona, Spain; guozhiming@live.cn

* Correspondence: iiaconeta@cimne.upc.edu; Tel.: +34-93-401-08-08

Received: 28 July 2017; Accepted: 25 September 2017; Published: 30 September 2017

Abstract: The simulation of large deformation problems, involving complex history-dependent constitutive laws, is of paramount importance in several engineering fields. Particular attention has to be paid to the choice of a suitable numerical technique such that reliable results can be obtained. In this paper, a Material Point Method (MPM) and a Galerkin Meshfree Method (GMM) are presented and verified against classical benchmarks in solid mechanics. The aim is to demonstrate the good behavior of the methods in the simulation of cohesive-frictional materials, both in static and dynamic regimes and in problems dealing with large deformations. The vast majority of MPM techniques in the literature are based on some sort of explicit time integration. The techniques proposed in the current work, on the contrary, are based on implicit approaches, which can also be easily adapted to the simulation of static cases. The two methods are presented so as to highlight the similarities to rather than the differences from “standard” Updated Lagrangian (UL) approaches commonly employed by the Finite Elements (FE) community. Although both methods are able to give a good prediction, it is observed that, under very large deformation of the medium, GMM lacks robustness due to its meshfree nature, which makes the definition of the meshless shape functions more difficult and expensive than in MPM. On the other hand, the mesh-based MPM is demonstrated to be more robust and reliable for extremely large deformation cases.

Keywords: particle methods; nonlinear finite element method; implicit MPM; Galerkin Meshfree Method; cohesive-frictional materials

1. Introduction

The numerical simulation of solid mechanics problems involving history-dependent materials and large deformations has historically represented one of the most important topics in computational mechanics. In the framework of discrete mechanics, the Discrete Element Method (DEM) [1] is a popular technique in geotechnical engineering [2] for its advantages in handling large deformation and failure problems and for its relatively easy algorithm and implementation. However, it is well known that time-consuming procedures should be followed to calibrate the material parameters, and it is still extremely hard to handle real scale problems for the huge computational effort required. Within the continuum mechanics framework, a higher number of numerical techniques has been tested, and their effectiveness in the context of cohesive-frictional materials, which undergo large deformations, have been assessed. In this regard, an important contribution is represented by the works of Cervera and coworkers [3–6], where the Finite Element Method (FEM) is employed to study strain localization in cohesive-frictional materials. However, their work is limited to the case of infinitesimal strains and does not consider the large deformation regime [7]. The Lagrangian viewpoint presents, in this context,

a rather obligatory choice, since the adoption of such a framework greatly simplifies the constitutive modeling and the tracking of the entire deformation process. In the case of mesh-based methods, the natural limitation of the Lagrangian approach is related to the deformation of the underlying discretisation, which tends to get tangled as the deformation increases. Aggressive remeshing techniques have proven to be capable of further extending the realm of applicability of Lagrangian approaches, effectively extending the limits of the approach well beyond its original boundaries. For instance, the Arbitrary Lagrangian-Eulerian method (ALE) [8] has been developed in an attempt to overcome the limitation of the Total Lagrangian (TL) and Updated Lagrangian (UL) techniques when severe mesh distortion occurs. Many studies on the application of the ALE method can be found in the literature such as, for example, the numerical modeling of friction stir welding [9] or the simulation of geotechnical problems [10]. Another Lagrangian technique, which exploits such remeshing procedures, is represented by the particle finite element method, first developed for the simulation of free surface flows and breaking waves [11,12] and then successfully adapted for structural mechanics problems involving large deformations [13–15] or for the simulation of viscoplastic materials [16–21] and geomaterials [22,23]. Although the method has broad capabilities, some disadvantages come from the use of remeshing procedures, such as, issues in the parallelization of the code, in the conservation of the mass and difficulties related to the storage of historical variables [24–26]. Among the meshfree techniques, it is worth mentioning the Smooth Particle Hydrodynamics method (SPH) [27,28], a Lagrangian meshfree method, initially designed for hydrodynamics problems [29,30]. Recently, this method has been successfully applied to large deformation problems of geomaterials [31–34]. However, some disadvantages come from its meshfree nature, such as the impossibility to directly include boundary conditions in the SPH formalism, penetration problems between continua when high speed or impact occurs and the introduction of artificial viscosity to avoid unstable solutions.

Alternative Lagrangian techniques proposed in the literature, which solve the drawbacks listed before, are represented by the Material Point Method (MPM) [35,36] and the Galerkin Meshless Method (GMM) [37], which are the objects of study of the current work. The MPM is a particle-based method, extensively used for geotechnical problems [38–40] for its capabilities in tracking extremely large deformation while preserving material properties of the material points. Most MPM codes make use of explicit time integrations, due to the ease of the formulation and implementation. However, some implicit versions of MPM can be found in the literature. For instance, Guilkey [41] exploits the similarities between MPM and FEM in an implicit solution strategy. Beuth [42] proposes an implicit MPM formulation for quasi-static problems using high order elements and a special integration procedure for partially-filled boundary elements. Finally, Sanchez [43] presented an implicit MPM for quasi-static problems using a Jacobian free algorithm. In the current work, the displacement-based formulation and the time scheme integration of the MPM algorithm are equivalent to those proposed by [41]. The GMM is a truly meshless method, based on a Galerkin formulation. Unlike other methods, such as the element-free Galerkin method [44] or the reproducing kernel particle method [45], the technique employed in this work does not need element connectivity for integration or interpolation purposes. The algorithm presented in [37] to simulate fluid-structure interaction problems is taken as a starting point and adapted to the simulation of deformable solids.

Both MPM and GMM combine a Lagrangian description of the body under analysis, which is represented by a set of particles, the so-called material points, with the use of a computational mesh: a background grid in the case of MPM and a cloud of nodes in the case of GMM. At each time step, the governing equations are solved on the computational nodes, while history-dependent variables and material information are saved on the particles during the entire deformation process. As MPM and GMM do not employ any kind of remeshing procedure, the calculation is performed always at a local level, allowing an easy adaptation of the code to parallel computation. Moreover, the conservation of the mass is guaranteed during the whole simulation time, as the total mass is distributed between the material points, representing the volume of the entire continuum under study. Last but not least, a remapping of the state variables is avoided, and the employment of complex

time-dependent constitutive laws can be used without committing any mapping error. Given the fulfillment of the aforementioned features, MPM and GMM could represent suitable choices for the solution of real scale large deformation problems, and a comparison within a unified framework would be beneficial for an objective evaluation of the capabilities of each method.

In the present paper, the reader can find a detailed presentation of the implicit MPM and GMM algorithms and, as the main novelty, an assessment of their performance in terms of accuracy, computational cost and robustness. Both techniques employ a variational Galerkin formulation for solid mechanics problems, defined within an updated Lagrangian kinematic framework, under the assumption of finite strains. The first one uses a fixed background grid, which is deformed and then reset at every solution step. The second one uses a purely Lagrangian meshless approach. The two methods differ conceptually only in the way the shape functions and their gradients are computed. Nevertheless, as we shall see, the use of one technique leads to advantages and disadvantages over the alternative one. Both techniques described are implemented by the authors within Kratos Multiphysics [46,47], a framework for building multi-disciplinary finite element programs in C++.

The paper is structured as follows: the governing equations are presented in their strong form in Section 2, and the algebraic linearized system is obtained in Section 3. After that, MPM and GMM are briefly revised, and the two algorithms are detailed in Section 4 to emphasize the similarities with an FEM algorithm. Some benchmark tests are simulated to assess the capabilities of the developed formulations (Section 5), and finally, Section 6 provides some concluding remarks and suggestions for future research.

2. Governing Equations

Let us consider the body \mathcal{B} , which occupies a region Ω of the three-dimensional Euclidean space \mathcal{E} with a regular boundary $\partial\Omega$ in its reference configuration. A deformation of \mathcal{B} is defined by a one-to-one mapping:

$$\varphi : \Omega \rightarrow \mathcal{E} \quad (1)$$

that maps each point \boldsymbol{p} of the body \mathcal{B} into a spatial point \boldsymbol{x} :

$$\boldsymbol{x} = \varphi(\boldsymbol{p}) \quad (2)$$

which represents the location of \boldsymbol{p} in the deformed configuration of \mathcal{B} . The region of \mathcal{E} occupied by \mathcal{B} in its deformed configuration is denoted as $\varphi(\Omega)$.

The problem is governed by mass and linear momentum balance equations:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \boldsymbol{v} = 0 \quad \text{in } \varphi(\Omega) \quad (3a)$$

$$\rho \boldsymbol{a} - \nabla \cdot \boldsymbol{\sigma} = \rho \boldsymbol{b} \quad \text{in } \varphi(\Omega) \quad (3b)$$

where ρ is the mass density, \boldsymbol{a} is the acceleration, \boldsymbol{v} is the velocity, $\boldsymbol{\sigma}$ is the symmetric Cauchy stress tensor and \boldsymbol{b} is the body force. Acceleration and velocity are, by definition, the material derivatives of the velocity, \boldsymbol{v} , and the displacement, \boldsymbol{u} , respectively. For a compressible material, the conservation of mass is satisfied by:

$$\rho = \frac{\rho_0}{\det(\boldsymbol{F})} \quad (4)$$

where ρ_0 is the density in the undeformed configuration and $\det(\boldsymbol{F})$ is the determinant of the total deformation gradient $\boldsymbol{F} = d\boldsymbol{x}/d\boldsymbol{X}$ with \boldsymbol{x} and \boldsymbol{X} representing the current and initial position, respectively. Equation (4) holds at any point and in particular at the sampling points where the equation is written, e.g., the material points. Thermal effects are not considered in the present work, so the energy balance is considered implicitly fulfilled.

The balance equations are solved numerically in a three-dimensional region $\Omega \subseteq \mathcal{R}^3$, in the time range $t \in [0, T]$, given the following boundary conditions on the Dirichlet ($\varphi(\partial\Omega_D)$) and Neumann boundaries ($\varphi(\partial\Omega_N)$), respectively:

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on} \quad \varphi(\partial\Omega_D) \quad (5a)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on} \quad \varphi(\partial\Omega_N) \quad (5b)$$

where \mathbf{n} is the unit outward normal.

A constitutive equation for evaluation of the stress-strain relation is also needed to fully define the boundary value problem.

3. Weak Form

In Section 2, the strong form of the problem has been defined. In this section, the weak form is derived, following the formulation explained in [48], a displacement-based finite element procedure.

Let the displacement space $\mathcal{V} \in [\mathbf{H}^1(\mathcal{B})]^d$ be the space of vector functions whose components and their first derivatives are square-integrable; the integral form of the problem is:

$$\int_{\varphi(\Omega)} (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{w} dv + \int_{\varphi(\Omega)} \rho (\mathbf{b} - \mathbf{a}) \cdot \mathbf{w} dv - \int_{\varphi(\partial\Omega_N)} (\boldsymbol{\sigma} \cdot \mathbf{n} - \bar{\mathbf{t}}) \cdot \mathbf{w} da = 0, \quad \forall \mathbf{w} \in \mathcal{V} \quad (6)$$

where \mathbf{w} is an arbitrary test function, such that $\mathbf{w} = \{\mathbf{w} \in \mathcal{V} \mid \mathbf{w} = \mathbf{0} \text{ on } \varphi(\partial\Omega_D)\}$, dv is the differential volume and da the differential boundary surface. By integrating by parts, applying the divergence theorem and considering the symmetry of the stress tensor, the following expression is obtained:

$$\int_{\varphi(\Omega)} \boldsymbol{\sigma} : (\nabla^S \mathbf{w}) dv - \int_{\varphi(\Omega)} \rho (\mathbf{b} - \mathbf{a}) \cdot \mathbf{w} dv - \int_{\varphi(\partial\Omega_N)} \bar{\mathbf{t}} \cdot \mathbf{w} da = 0 \quad (7)$$

Under the assumption that the stress tensor is a function of the current strain only:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\epsilon}) \quad (8)$$

the problem is reduced to find a kinematically-admissible field \mathbf{u} that satisfies:

$$G(\mathbf{u}, \mathbf{w}) = 0 \quad \forall \mathbf{w} \in \mathcal{V} \quad (9)$$

where G is the virtual work functional defined as:

$$G(\mathbf{u}, \mathbf{w}) = \int_{\varphi(\Omega)} \boldsymbol{\sigma} : (\nabla^S \mathbf{w}) dv - \int_{\varphi(\Omega)} \rho (\mathbf{b} - \mathbf{a}) \cdot \mathbf{w} dv - \int_{\varphi(\partial\Omega_N)} \bar{\mathbf{t}} \cdot \mathbf{w} da \quad (10)$$

3.1. Linearization of the Spatial Weak Formulation

In this work, we attempt to solve the general Boundary Value Problems (BVP), characterized by both geometrical and material non-linearity. When a non-linear BVP is considered, the discretisation of the weak form results in a system of non-linear equations; for the solution of such a system, a linearization is, therefore, needed. The most used and known technique is Newton–Raphson’s iterative procedure, which makes use of directional derivatives to linearize the non-linear equations.

The virtual work functional of Equation (10) is linearized with respect to the unknown \mathbf{u} , using an arbitrary argument \mathbf{u}^* , which is chosen to be the last known equilibrium configuration. The linearized problem is to find $\delta\mathbf{u}$ such that:

$$L(\delta\mathbf{u}, \mathbf{w}) \simeq G(\mathbf{u}^*, \mathbf{w}) + DG(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}] = 0, \quad \forall \mathbf{w} \in \mathcal{V} \tag{11}$$

where L is the linearized virtual work function and:

$$DG(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}] = \left. \frac{d}{d\gamma} \right|_{\gamma=0} G(\mathbf{u}^* + \gamma\delta\mathbf{u}, \mathbf{w}) \tag{12}$$

is the directional derivative of G at \mathbf{u}^* in the direction of $\delta\mathbf{u}$, given by:

$$DG(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}] = \left. \frac{d}{d\gamma} \right|_{\gamma=0} \int_{\varphi(\Omega)} [\boldsymbol{\sigma}(\boldsymbol{\epsilon}(\gamma)) : (\nabla^S \mathbf{w}) - \rho(\mathbf{b} - \mathbf{a}) \cdot \mathbf{w}] dv - \left. \frac{d}{d\gamma} \right|_{\gamma=0} \int_{\varphi(\partial\Omega_N)} \bar{\mathbf{t}} \cdot \mathbf{w} da \tag{13}$$

Under the assumption of conservative external loads, only the terms related to the internal and inertial forces are dependent on the deformation. Using the following definitions:

$$\boldsymbol{\epsilon}(\gamma) = \nabla^S(\mathbf{u}^* + \gamma\delta\mathbf{u}) = \boldsymbol{\epsilon}^* + \gamma\nabla^S(\delta\mathbf{u}) \tag{14}$$

where $\boldsymbol{\epsilon}^* = \nabla^S(\mathbf{u}^*)$ is the strain field at \mathbf{u}^* and $\mathbf{u}(\gamma) = \mathbf{u}^* + \gamma\delta\mathbf{u}$, the directional derivative $DG(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}]$ reduces to:

$$DG(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}] = \left. \frac{d}{d\gamma} \right|_{\gamma=0} \left(\int_{\varphi(\Omega)} [\boldsymbol{\sigma}(\boldsymbol{\epsilon}(\gamma)) : (\nabla^S \mathbf{w}) + \rho\mathbf{a}(\mathbf{u}(\gamma)) \cdot \mathbf{w}] dv \right) = DG^{static}(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}] + DG^{dynamic}(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}] \tag{15}$$

which can be split into a static and dynamic contribution.

Under the assumption of finite strains and adopting an updated Lagrangian kinematic framework, the expression of the directional derivative (Equation (15)) should be derived in spatial form. A common way to do that consists of linearizing the material weak form and in doing a push-forward operation to recover the spatial form [48]. Therefore, the linearization of the weak form derived with respect to the initial configuration reads:

$$DG(\mathbf{u}^*, \mathbf{w})[\delta\mathbf{u}] = \int_{\Omega} \nabla_X \delta\mathbf{u} \mathbf{S} \cdot \nabla_X \mathbf{w} dV + \int_{\Omega} [(\mathbf{F}^T \nabla_x^S \mathbf{w} \mathbf{F}) : \mathbb{C}(\mathbf{F}^T \nabla_x^S \delta\mathbf{u} \mathbf{F})] dV + \int_{\Omega} \rho_0 \frac{d\mathbf{a}}{d\mathbf{u}} \cdot \mathbf{w}[\delta\mathbf{u}] dV \tag{16}$$

where ∇_X and ∇_x are the material and spatial gradient operator, respectively, \mathbf{S} is the second Piola–Kirchhoff stress tensor, \mathbb{C} is the fourth order incremental constitutive tensor and dV is the differential volume element in the undeformed configuration. The linearization of the weak form with respect to the current configuration can be derived by pushing forward the linearization of Equation (16). The first term can be directly written in terms of the Kirchhoff stress $\boldsymbol{\tau} = \mathbf{F}\mathbf{S}\mathbf{F}^T$ as:

$$\nabla_X \delta\mathbf{u} \mathbf{S} \cdot \nabla_X \mathbf{w} = \nabla_X \delta\mathbf{u} \mathbf{F}^{-1} \boldsymbol{\tau} \mathbf{F}^{-T} \cdot \nabla_X \mathbf{w} \tag{17}$$

and using this standard identity $\nabla_x a = \nabla_X a F^{-1}$, Equation (17) can be written as:

$$\nabla_X \delta \mathbf{u} \mathbf{S} \cdot \nabla_X \mathbf{w} = \nabla_x \delta \mathbf{u} \boldsymbol{\tau} \cdot \nabla_x \mathbf{w} \quad (18)$$

The second integral of Equation (16) can be re-written as:

$$\int_{\Omega} \nabla_x^S \mathbf{w} : \widehat{\mathbb{C}} [\nabla_x^S \delta \mathbf{u}] dV \quad (19)$$

adopting the transformation of the fourth order incremental constitutive tensor \mathbb{C} in Voigt notation [48]:

$$\widehat{\mathbb{C}}_{iklm} = F_{iA} F_{lC} F_{mD} F_{kB} \mathbb{C}_{ABCD} \quad (20)$$

where lowercase indexes refer to the incremental constitutive tensor relative to the Kirchhoff stress, while uppercase indexes to the incremental constitutive tensor relative to the second Piola–Kirchhoff stress.

With these transformations, the linearization of the static contribution at the current configuration is:

$$DG^{static}(\mathbf{u}^*, \mathbf{w})[\delta \mathbf{u}] = \int_{\Omega} \nabla_x \delta \mathbf{u} \boldsymbol{\tau} \cdot \nabla_x \mathbf{w} + \nabla_x^S \mathbf{w} : \widehat{\mathbb{C}} [\nabla_x^S \delta \mathbf{u}] dV \quad (21)$$

Considering the definition of the determinant of the deformation gradient:

$$\det(\mathbf{F}) = J = \frac{dv}{dV} \quad (22)$$

the following relations hold

$$\boldsymbol{\sigma} = \frac{1}{J} \boldsymbol{\tau} \quad (23)$$

$$\overline{\mathbb{C}} = \frac{1}{J} \widehat{\mathbb{C}} \quad (24)$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are the Cauchy and Kirchhoff stress tensor, respectively, and $\overline{\mathbb{C}}$ is the incremental constitutive tensor relative to the Cauchy stress. Equation (16) can now be re-written in the current configuration as:

$$DG(\mathbf{u}^*, \mathbf{w})[\delta \mathbf{u}] = \int_{\varphi(\Omega)} \left(\nabla_x \delta \mathbf{u} \boldsymbol{\sigma} \cdot \nabla_x \mathbf{w} + \nabla_x^S \mathbf{w} : \overline{\mathbb{C}} [\nabla_x^S \delta \mathbf{u}] + \rho \frac{d\mathbf{a}}{d\mathbf{u}} \cdot \mathbf{w}[\delta \mathbf{u}] \right) dv \quad (25)$$

Equation (25) represents the linearization of the spatial weak formulation, also known as the updated Lagrangian formulation, since the deformation state \mathbf{u}^* is continuously updated during the non-linear incremental solution procedure, e.g., Newton–Raphson’s method.

3.2. Spatial Discretisation

For the sake of clarity, hereinafter, the p subscript is used to refer to variables attributed to the material points, while the I subscript is used to refer to variables attributed to the computational nodes. Let us assume discretizing the continuum body \mathcal{B} by a set of n_p material points and assigning a finite volume of the body Ω_p to each of those material points. Thus, the geometrical representation (\mathcal{B}_h) of \mathcal{B} reads:

$$\mathcal{B} \approx \mathcal{B}_h = \bigcup_{p=1}^{n_p} \Omega_p \quad (26)$$

and the integrals of the weak form can be written as:

$$\int_{\mathcal{B}} (\dots) dV \approx \int_{\mathcal{B}^h} (\dots) dV = \bigcup_{p=1}^{n_p} \int_{\Omega_p} (\dots) d\Omega_p \tag{27}$$

Let \mathcal{V}_h be a finite element space to approximate \mathcal{V} . The problem is now finding $\mathbf{u}_h \in \mathcal{V}_h$ such that:

$$DG(\mathbf{u}_h^*, \mathbf{w}_h)[\delta \mathbf{u}_h] = -G(\mathbf{u}_h^*, \mathbf{w}_h), \quad \forall \mathbf{w}_h \in \mathcal{V}_h \tag{28}$$

or using Equation (25):

$$\int_{\varphi(\Omega)} \left\{ \nabla_x \delta \mathbf{u}_h \boldsymbol{\sigma} \cdot \nabla_x \mathbf{w}_h + \nabla^S \mathbf{w}_h : \bar{\bar{\mathbf{C}}}[\nabla^S \delta \mathbf{u}_h] + \rho \frac{d\mathbf{a}_h}{d\mathbf{u}_h} \cdot \mathbf{w}_h[\delta \mathbf{u}_h] \right\} dv = \tag{29}$$

$$- \left(\int_{\varphi(\Omega)} \boldsymbol{\sigma} : (\nabla^S \mathbf{w}_h) dv - \int_{\varphi(\Omega)} \rho (\mathbf{b} - \mathbf{a}_h) \cdot \mathbf{w}_h dv - \int_{\varphi(\partial\Omega_N)} \bar{\mathbf{t}} \cdot \mathbf{w}_h da \right)$$

The detailed procedure to obtain the linearization of Equation (29) can be found in [48].

The final discretized form can be written as:

$$\bigcup_{p=1}^{n_p} \sum_{I=1}^n \sum_{K=1}^n \mathbf{w}_I^T \left((\nabla_x N_I)^T \boldsymbol{\sigma} (\nabla_x N_K) \mathbf{I} + \mathbf{B}_I^T \mathbf{D} \mathbf{B}_K + \frac{N_I \rho N_K}{\beta \Delta t^2} \mathbf{I} \right) V_p \delta \mathbf{u}_K \tag{30}$$

$$= - \bigcup_{p=1}^{n_p} \sum_{I=1}^n \mathbf{w}_I^T \left(\mathbf{B}_I \boldsymbol{\sigma} - \rho \mathbf{b} N_I + \sum_{K=1}^n N_I \rho N_K \mathbf{a}_K \right) V_p - \bigcup_{l=1}^{n_l} \sum_{I=1}^{n_m} \mathbf{w}_I^T N_I \bar{\mathbf{t}} A_l$$

where I and K are the indexes of the finite element's nodes, $\nabla_x N_I$ is the spatial gradient of the shape function evaluated at node I , \mathbf{D} is the matrix form of the incremental constitutive tensor $\bar{\bar{\mathbf{C}}}$, V_p is the volume relative to a single material point, A_l is the surface and \mathbf{B}_I is the deformation matrix relative to node I , expressed here for a 2D problem as:

$$\mathbf{B}_I = \begin{bmatrix} \frac{\partial N_I}{\partial x} & 0 \\ 0 & \frac{\partial N_I}{\partial y} \\ \frac{\partial N_I}{\partial y} & \frac{\partial N_I}{\partial x} \end{bmatrix} \tag{31}$$

The left-hand side of Equation (30) is given by three addends multiplied by the increment of the unknowns. The first one is commonly known as the geometric stiffness matrix:

$$\mathbf{K}_{IK}^{geo} = (\nabla_x N_I)^T \boldsymbol{\sigma} (\nabla_x N_K) \mathbf{I} V_p \tag{32}$$

while the second term is known as the material stiffness matrix:

$$\mathbf{K}_{IK}^{mat} = \mathbf{B}_I^T \mathbf{D} \mathbf{B}_K V_p \tag{33}$$

and their sum represents the static contribution to the tangent stiffness matrix:

$$\mathbf{K}_{IK}^{static} = \mathbf{K}_{IK}^{geo} + \mathbf{K}_{IK}^{mat} \tag{34}$$

The dynamic component is given by:

$$\mathbf{K}_{IK}^{dynamic} = \frac{N_I \rho N_K}{\beta \Delta t^2} \mathbf{I} V_p \tag{35}$$

and its definition depends on the adopted time scheme as explained in Section 4.

Finally, the tangent stiffness matrix is given by:

$$\mathbf{K}_{IK}^{tan} = \mathbf{K}_{IK}^{static} + \mathbf{K}_{IK}^{dynamic} \quad (36)$$

and represents the submatrix relative to one node of the discretisation with dimension $[n_{dof} \times n_{dof}]$, where n_{dof} is the number of degrees of freedom of a single node. This matrix can be considered as the Jacobian matrix of the right-hand side of Equation (30), i.e., the residual \mathbf{R}_I . Equation (30) can be rewritten in compact form as:

$$\mathbf{K}_{IK}^{tan} \delta \mathbf{u}_K = -\mathbf{R}_I. \quad (37)$$

4. MPM and GMM Techniques

In Section 3, the formulation, common to both techniques, based on an updated Lagrangian description of the continuum, is presented. In this section, the characteristic features of MPM and GMM are highlighted.

As previously stated, MPM and GMM are two Lagrangian techniques and share the fact that the material points shall be understood as the integration points of the calculation, each carrying information about the material and kinematic response. Each material point represents a computational element with one single integration point (the material point itself), whose connectivity is defined by either the nodes of the elements in which it falls (in MPM) or by the nodes in the cloud around the material point (in GMM). In both algorithms, the initial position of the material points is chosen to coincide with the Gauss points of an FE grid, and the mass, which remains constant during the simulation, is equally distributed between the material points, falling, initially, in the same element.

In MPM, such material points may migrate from one element of the grid to another. In the evaluation of the FEM integrals, the shape functions are evaluated at the material point location on the basis of the grid element into which the material point falls (Figure 1a). To prevent mesh distortion, the nodal solution is deleted such that at each time step, the undeformed mesh is recovered; while in GMM, the material points move together with the computational nodes, and the shape functions are evaluated once the surrounding cloud of nodes is defined (Figure 1b). In this case the nodes preserve their history through the whole simulation, as in FEM.

4.1. MPM

MPM is a particle method, proposed for the first time by Harlow [49] for the solution of fluid flow problems under a large deformation regime and originally known by the name of the Particle-In-Cell (PIC) method. Some decades after, Sulsky and coworkers presented its extension to solid mechanics problems [35,36]. As previously stated, most MPM codes make use of explicit time schemes, which are generally preferable when simulating impacts at high velocities or fast transient problems. In other cases, for example when the driving force is gravity or when the rate of deformation is small, the adoption of an implicit time scheme is the best choice, since the stability of the method (for properly chosen dissipative methods) does not depend on the wave propagation speed within the media, which provides the typical time step limitation for explicit approaches [50].

In the current work, the displacement-based formulation and the time scheme integration of the MPM algorithm are equivalent to those proposed by [41].

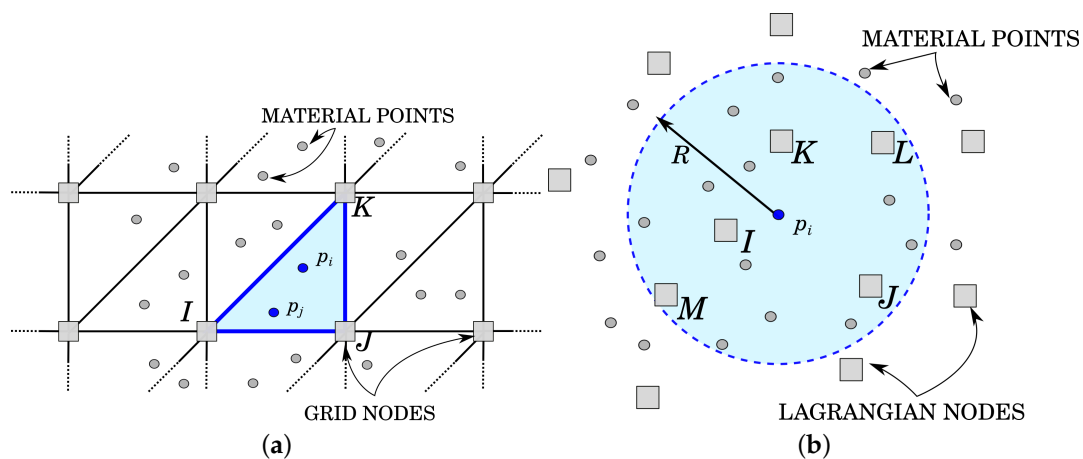


Figure 1. Shape functions calculation in MPM and Galerkin Meshfree Method (GMM). (a) MPM. The shape functions on the material point p_i are evaluated using the FE shape function of element I-J-K; (b) GMM. The shape functions on the material point p_i are evaluated using the information on the nodes sufficiently close to the material point itself.

4.1.1. MPM Algorithm

Traditionally, the MPM algorithm is composed of three different phases [36], as graphically represented in Figure 2:

- Initialization phase (Figure 2a): At the beginning of the time step, the connectivity is defined for each material points, and the initial conditions on the FE grid nodes are created by means of a projection of material points' information obtained at the previous time step t_n ;
- UL-FEM calculation phase (Figure 2b): The local matrix, represented by the left-hand-side (*lhs*) of Equation (30), and the local vector, constituted by the right-hand-side (*rhs*) of Equation (30), are evaluated in the current configuration according to the formulation presented in the previous section. The global left-hand-side matrix (*LHS*) and the global right-hand-side vector (*RHS*) are obtained by assembling the local contributions of each material point, and finally, the system is iteratively solved. During the iterative procedure, the nodes are allowed to move, according to the nodal solution, and the material points do not change their local position within the geometrical element until the solution has reached convergence;
- Convective phase (Figure 2c): During the third and last phase, the nodal information at time t_{n+1} is interpolated back to the material points. The position of the material points is updated and, in order to prevent mesh distortion, the undeformed FE grid is recovered.

Many features of the MPM are connected to the finite element method [35]. Indeed, Phase b coincides with the calculation step of a standard non-linear FE code, while Phases a and c define the MPM features.

At the beginning of each time step (t_n), during Phase a, the degrees of freedom and the variables on the nodes of the fixed mesh are defined gathering the information from the material points (Figure 2a).

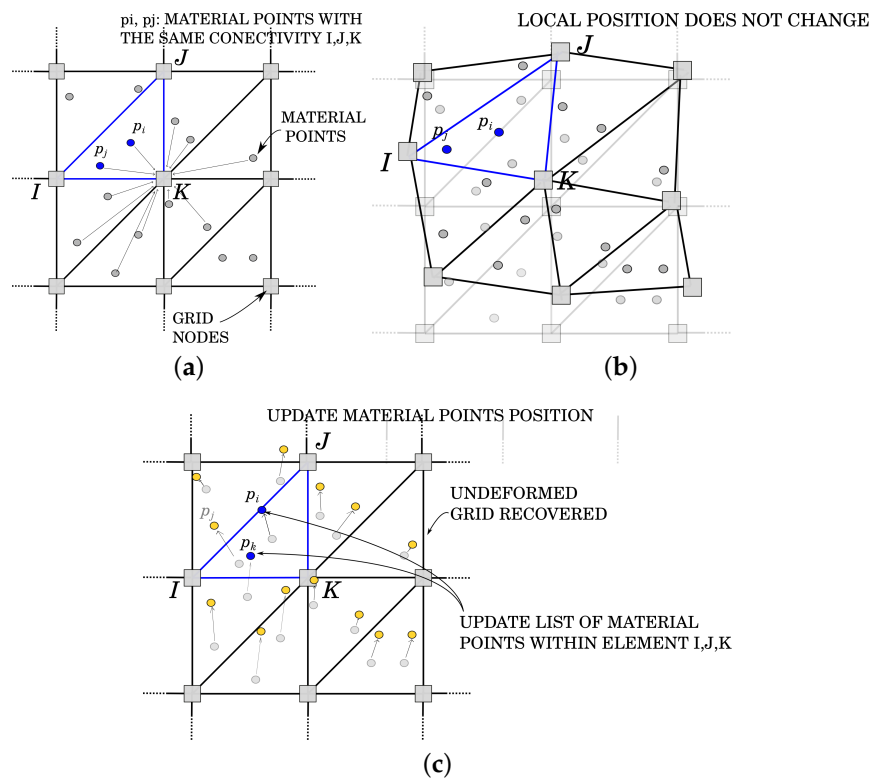


Figure 2. MPM phases. (a) Initialization phase; (b) Updated Lagrangian FEM phase; (c) Convective phase.

The momentum q_p and inertia f_p on the material points, which are defined as functions of mass m_p , velocity v_p and acceleration a_p :

$$q_p^n = v_p^n m_p \tag{38}$$

$$f_p^n = a_p^n m_p \tag{39}$$

are projected on the background grid by evaluating in a first step the global values of nodal mass m_I , momentum q_I and inertia f_I as described in Algorithm 1.

Once m_I , q_I and f_I are obtained, it is possible to compute the values of nodal velocity and nodal acceleration of the previous time step as:

$$\tilde{v}_I^n = \frac{q_I^n}{m_I} \tag{40}$$

$$\tilde{a}_I^n = \frac{f_I^n}{m_I} \tag{41}$$

It is worth mentioning that the initial nodal conditions are evaluated at each time step using material point information in order to have initial values even on grid elements empty at the previous time step ($t_{n-1} - t_n$).

Both Lagrangian techniques presented in this paper make use of a predictor/corrector procedure, based on the Newmark integration scheme.

In MPM, the prediction of the nodal displacement, velocity and acceleration reads:

$${}^{it+1}\Delta\mathbf{u}_I^{n+1} = 0.0 \quad (42)$$

$${}^{it+1}\mathbf{v}_I^{n+1} = \frac{\lambda}{\zeta\Delta t} [{}^{it+1}\Delta\mathbf{u}_I^{n+1}] - \left(\frac{\lambda}{\zeta} - 1\right) \tilde{\mathbf{v}}_I^n - \frac{\Delta t}{2} \left(\frac{\lambda}{\zeta} - 2\right) \tilde{\mathbf{a}}_I^n \quad (43)$$

$${}^{it+1}\mathbf{a}_I^{n+1} = \frac{1}{\zeta\Delta t^2} [{}^{it+1}\Delta\mathbf{u}_I^{n+1}] - \frac{1}{\zeta\Delta t} \tilde{\mathbf{v}}_I^n - \left(\frac{1}{2\zeta} - 1\right) \tilde{\mathbf{a}}_I^n \quad (44)$$

where the upper-left side index it indicates the iteration counter, while the upper-right index n the time step. λ and ζ are Newmark's coefficients equal to 0.5 and 0.25, respectively.

Once the nodal velocity and acceleration are predicted (Equations (42)–(44)), the system of linearized governing equations is formulated, according to Section 3, and the local matrix \mathbf{K}^{tan} and the residual \mathbf{R}_I are evaluated and assembled according to Equations (30) and (37), respectively (Phase b, Figure 2b).

The solution in terms of increment of nodal displacement is found iteratively solving the residual-based system of Equation (37). Once the solution ${}^{it+1}\delta\mathbf{u}_I^{n+1}$ is obtained, a correction of the nodal increment of displacement is performed:

$${}^{it+1}\Delta\mathbf{u}_I^{n+1} = {}^{it}\Delta\mathbf{u}_I^{n+1} + {}^{it+1}\delta\mathbf{u}_I^{n+1} \quad (45)$$

Velocity and acceleration are corrected according to Equations (43) and (44), respectively. This procedure has to be repeated until convergence is reached.

Unlike an FEM code, the nodal information is available only during the calculation of a time step: at the beginning of each time step, a reset of all the nodal information is performed, and the accumulated displacement information is deleted. The computational mesh is allowed to deform only during the iterative procedure of a time step, avoiding the typical element tangling of a standard FEM. When convergence is achieved, the position of the nodes is restored to the original one (Phase c, Figure 2c). Before restoring the undeformed configuration of the FE grid, the solution in terms of nodal displacement, velocity and acceleration is interpolated on the material points, as:

$$\Delta\mathbf{u}_p^{n+1} = \sum_{n=1}^{n_n} N_I(\xi_p, \eta_p) \Delta\mathbf{u}_I^{n+1} \quad (46)$$

$$\mathbf{a}_p^{n+1} = \sum_{n=1}^{n_n} N_I(\xi_p, \eta_p) \mathbf{a}_I^{n+1} \quad (47)$$

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{1}{2}\Delta t (\mathbf{a}_p^n + \mathbf{a}_p^{n+1}) \quad (48)$$

where n_n is the total number of nodes per geometrical element, (ξ_p, η_p) are the local coordinates of material point p and $N_I(\xi_p, \eta_p)$ is the shape function evaluated at the position of the material point p , relative to node I .

Finally, the current position of the material points is updated as:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta\mathbf{u}_p^{n+1} \quad (49)$$

The details of the MPM algorithm are presented in Algorithm 1.

Algorithm 1 MPM algorithm.(we will use $(\bullet)^n = (\bullet)(t_n)$)Material DATA: E, ν, ρ Initial data on material points: $m_p, \mathbf{x}_p^n, \Delta t, \mathbf{u}_p^n, \mathbf{v}_p^n, \mathbf{a}_p^n, \mathbf{F}_p^n = \sum_I \frac{\partial N_I}{\partial \mathbf{x}_I^0} \cdot \mathbf{x}_I^n \Delta \mathbf{F}_p = \sum_I \frac{\partial N_I}{\partial \mathbf{x}_I^n} \cdot \mathbf{x}_I^{n+1}$ Initial data on nodes: **NONE - everything is discarded in the initialization phase**OUTPUT of calculations: $\Delta \mathbf{u}_I^{n+1}, \sigma_p^{n+1}$ **1. INITIALIZATION PHASE**

- Clear nodal info and recover undeformed grid configuration
- Calculation of initial nodal conditions.
 - (a) for $p = 1:N_p$
 - * Calculation of nodal data
 - $\mathbf{q}_I^n = \sum_p N_I m_p \mathbf{v}_p^n$
 - $\mathbf{f}_I^n = \sum_p N_I m_p \mathbf{a}_p^n$
 - $m_I^n = \sum_p N_I m_p$
 - (b) for $I = 1:N_I$
 - * $\tilde{\mathbf{v}}_I^n = \frac{\mathbf{q}_I^n}{m_I^n}$
 - * $\tilde{\mathbf{a}}_I^n = \frac{\mathbf{f}_I^n}{m_I^n}$
- Newmark method: PREDICTOR. Evaluation of $^{it+1}\Delta \mathbf{u}_I^{n+1}$, $^{it+1}\mathbf{v}_I^{n+1}$ and $^{it+1}\mathbf{a}_I^{n+1}$ using Equations (42)–(44)

2. UL-FEM PHASE

- for $p = 1:N_p$
 - (a) Evaluation of local residual (*rhs*) (Equation (10))
 - (b) Evaluation of local Jacobian matrix of residual (*lhs*) (Equation (25))
 - (c) Assemble rhs and lhs to the global vector *RHS* and global matrix *LHS* (Equations (30) and (37))
- Solving system ($\Delta \mathbf{u}_I^{n+1}$)
- Newmark method: CORRECTOR (Equations (43)–(45))
- Check convergence
 - (a) NOT converged: go to Step 2
 - (b) Converged: go to Step 3

3. CONVECTIVE PHASE

- Update the kinematics on the material points by means of an interpolation of nodal information (Equations (46)–(49))
- Save the stress σ_p^{n+1} , strain ϵ_p^{n+1} and total deformation gradient \mathbf{F}_p^{n+1} on material points (the latter by $\mathbf{F}_p^{n+1} = \Delta \mathbf{F}_p \cdot \mathbf{F}_p^n$)

4.2. GMM

The GMM is a truly meshless method that can be seen as the application of the MPM idea extended to the case in which both the nodes and the material points behave as purely Lagrangian through the whole analysis. Thus, it is relatively easy to enforce conservation properties at the integration points, while also maintaining the history of nodal results during all the simulation

time, provided that a reliable technique is chosen for the computation of the meshless shape functions. The difficulty is, hence, moved to the construction of such an effective meshless base, which is addressed in Section 4.2.2.

4.2.1. GMM Algorithm

The GMM algorithm is based on three principal steps (see Figure 3). The initialization phase (Figure 3a) is the step that mostly distinguishes GMM from MPM. During this phase, the connectivity of each integration point (i.e., each material point) is computed as the “cloud of nodes”, centered on the material point and obtained by a search-in-radius. Such a cloud is then employed for the calculation of the shape functions. Unlike MPM, the Newmark prediction is performed by using the nodal information of the previous time step, like in FEM. Once N and ∇N are suitably defined, MPM and GMM essentially coincide in the following steps. This is reflected in Steps 2 (Figure 3b) and 3 (Figure 3c) of the algorithm being coincident with the MPM case. The details of the GMM algorithm are presented in Algorithm 2.

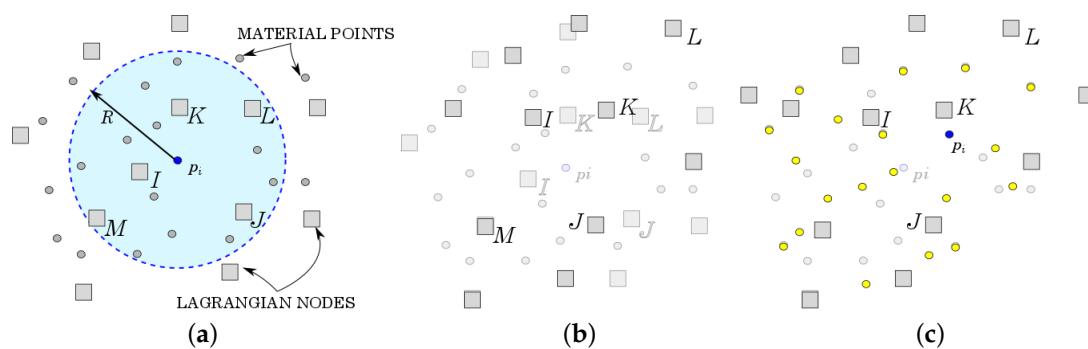


Figure 3. GMM phases. (a) Initialization phase; (b) Updated Lagrangian FEM phase; (c) Convective phase.

Algorithm 2 GMM algorithm.

Material DATA: E, ν, ρ

Initial data on material points: $m_p, \mathbf{x}_p^n, \Delta t, \mathbf{F}_p^n = \sum_I \frac{\partial N_I}{\partial \mathbf{x}_I^0} \cdot \mathbf{x}_I^n \Delta \mathbf{F}_p = \sum_I \frac{\partial N_I}{\partial \mathbf{x}_I^n} \cdot \mathbf{x}_I^{n+1}$

Initial data on nodes: $\mathbf{u}_I^n, \mathbf{v}_I^n, \mathbf{a}_I^n$

OUTPUT of calculations: $\mathbf{u}_I^{n+1}, \boldsymbol{\sigma}_p^{n+1}$

1. INITIALIZATION PHASE

- for every material point with position \mathbf{x}_p^n gather the cloud of nodes with position \mathbf{x}_I^n such that $|\mathbf{x}_p - \mathbf{x}_I| < R$
- compute the shape functions $N_I(\mathbf{x}_p^n)$ for all nodes I in the cloud
- Newmark method: PREDICTOR
for the prediction of displacement, unlike Equation (42), ${}^{it+1}\mathbf{u}_I^{n+1} = \mathbf{u}_I^n$,
while for the prediction of nodal velocity and nodal acceleration, see Equations (43) and (44)

2. UL-FEM PHASE (identical to MPM)

3. CONVECTIVE PHASE

- Update position of the material points by means of an interpolation of nodal solution
 - (a) for $p = 1:N_p$
 $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \sum_I N_I \Delta \mathbf{u}_I^{n+1}$
 - Save state of stress $\boldsymbol{\sigma}_p^{n+1}$, state of strain $\boldsymbol{\epsilon}_p^{n+1}$ and total deformation gradient \mathbf{F}_p^{n+1} on material points (the latter by $\mathbf{F}_p^{n+1} = \Delta \mathbf{F}_p \cdot \mathbf{F}_p^n$)
-

4.2.2. Calculation of GMM Shape Functions

While the computation of the shape functions is trivial for the standard MPM, thanks to the presence of a background grid (Figure 1a), the evaluation of the shape functions in GMM is more complex. From a technical point of view, GMM is based on a conceptually simple operation: given an arbitrary position x_p in space (which will, in practice, coincide with the position of the material point) and a search radius R , one may find all of the nodes I such that $\|x_I - x_p\| < R$. Given such a cloud of nodes, one may then compute, at the position x_p , the shape functions $N_I(x_p)$ (together with their gradients), such that, a given function u , whose nodal value is u_I , can be interpolated at the position x_p as $u(x_p) = \sum_I N_I(x_p) u_I$ (Figure 1b).

However, in order to construct a convergent solution, some guarantees must be provided by the shape functions. In particular, they shall comply with the Partition of Unity (PU) property, as a very minimum at all of the positions x_p at which the shape functions are evaluated. A number of shape functions exist complying with such a property [51–53]. Among the available options, two appealing classes of meshless functions are considered in our work: the first choice is constituted by the so-called Moving Least Square (MLS) method and the second one represented by the Local Maximum Entropy (LME) technique.

The first technique is based on the MLS approach, first introduced by Lancaster [54] and Belytschko [44,52]. The MLS-approximation fulfills the reproducing conditions by construction, so no corrections are needed.

The fundamental principle of MLS approximants is based on a weighted least square fitting of a target solution, sampled at a given, possibly randomly-distributed, set of points, via a function of the type:

$$P(\mathbf{x}) = a_1 + a_2x + a_3y + a_4xy + \dots \quad (50)$$

where the coordinates x, y are to be understood as relative to the sampling position.

The reconstruction of a continuous function $h(\mathbf{x})$ can be obtained considering the data h_I be located at points \mathbf{x}_I and an arbitrary, smooth and compactly supported, weight function $W_I(\mathbf{x})$, such that the \mathbf{x}_I fall within the support of W . Assuming now that the reconstructed function (h_x^h) is computed as:

$$h(\mathbf{x}) \approx h_x^h = \mathbf{P}^T(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}) \quad (51)$$

the fitting to $h(\mathbf{x})$ is done by minimizing the error function J , defined as:

$$J = \sum_I (\mathbf{P}_I^T \cdot \mathbf{a}(\mathbf{x}) - h_I)^2 W_I(\mathbf{x}) \quad (52)$$

where $\mathbf{P}_I = \mathbf{P}(\mathbf{x}_I)$.

This allows defining a set of approximating shape functions N such that:

$$h(\mathbf{x}) = \sum_I N_I(\mathbf{x}) h_I \quad (53)$$

where:

$$N_I = \mathbf{P}^T(\mathbf{x}) \cdot \mathbf{M}^{-1}(\mathbf{x}) \cdot \mathbf{P}_I W_I(\mathbf{x}) \quad (54)$$

with \mathbf{M} defined as:

$$\mathbf{M}(\mathbf{x}) = \sum_I \mathbf{P}_I \mathbf{P}_I^T W_I(\mathbf{x}) \quad (55)$$

It can be readily verified that the shape functions are able to reproduce exactly a polynomial up to the order used in the construction. This fact can also be used to prove compliance with the partition of unity property. Namely, if one assumes $h_I = \mathbf{P}_I(\mathbf{x}_I)$ and substitutes into Equation (53), then:

$$\begin{aligned}
\sum_I N_I(\mathbf{x}) \mathbf{P}_I^T &= \mathbf{P}^T(\mathbf{x}) \cdot \mathbf{M}^{-1}(\mathbf{x}) \cdot \sum_I \mathbf{P}_I \mathbf{P}_I^T W_I(\mathbf{x}) \\
&= \mathbf{P}^T(\mathbf{x}) \cdot \mathbf{M}^{-1}(\mathbf{x}) \cdot \mathbf{M}(\mathbf{x}) \\
&= \mathbf{P}^T(\mathbf{x})
\end{aligned}
\tag{56}$$

Hence, considering the special case of a constant polynomial $\mathbf{P}(\mathbf{x}) = 1$ or of a linear variation in \mathbf{x} , $\mathbf{P}(\mathbf{x}) = \mathbf{x}$ we obtain, respectively:

$$\begin{aligned}
\sum_I N_I &\equiv 1 \\
\sum_I \mathbf{x}_I N_I &\equiv \mathbf{x}
\end{aligned}
\tag{57}$$

A similar reasoning also gives:

$$\begin{aligned}
\sum_I \nabla N_I &\equiv 0 \\
\sum_I \mathbf{x}_I \cdot \nabla N_I &\equiv 1
\end{aligned}
\tag{58}$$

thus proving the compliance with the PU property.

However, MLS shape functions are not able to guarantee the Kronecker-delta property at the nodes. This implies that two nodal shape functions may be simultaneously non-zero at a given nodal position. This has practical implications at the moment of imposing Dirichlet boundary conditions, namely in order to impose $u(x_d) = 0$ at a given point on the Dirichlet boundary x_d , one must impose that $\sum(N_I(x_d)u_I) = 0$, which constitutes a classical multipoint constraint [55].

Interestingly, the choice of different shape functions could ease this particular problem. An appealing choice could be the use of LME approximants, which guarantee complying with a weak Kronecker-delta property until the cloud of nodes is represented by a convex hull.

The LME technique is based on the evaluation of the local max-ent approximants [56], which represents the solution that exhibits a (Pareto) compromise between competing objectives: the principle of max-ent [57] subject to the constraints:

$$\begin{aligned}
\text{For fixed } \mathbf{x} \text{ maximize } H(N_1, N_2, \dots, N_m) &= -\sum_I N_I \ln N_I, \\
\text{subject to } N_I &\geq 0, \quad I = 1, \dots, n_{node}, \\
\sum_I N_I &= 1, \quad \sum_I N_I \mathbf{x}_I = \mathbf{x}
\end{aligned}
\tag{59}$$

and the objective function interpreted as a measure of locality of the shape functions of the Delaunay triangulation:

$$\begin{aligned}
\text{For fixed } \mathbf{x} \text{ minimize } U(\mathbf{x}, N_1, N_2, \dots, N_m) &= -\sum_I N_I |\mathbf{x} - \mathbf{x}_I|^2, \\
\text{subject to } N_I &\geq 0, \quad I = 1, \dots, n_{node}, \\
\sum_I N_I &= 1, \quad \sum_I N_I \mathbf{x}_I = \mathbf{x}
\end{aligned}
\tag{60}$$

The solution to the problem can be found minimizing $\beta U(x, N_1, N_2, \dots, N_m) - H(N_1, N_2, \dots, N_m)$ subjected to the usual constrains. The optimization problem takes the form:

$$\begin{aligned} \text{For fixed } x \in \text{conv}X, \text{ minimize } & \sum_I \beta_I N_I |x - x_I|^2 + \sum_I N_I \ln N_I, \\ \text{subject to } & N_I \geq 0, \quad I = 1, \dots, n_{node}, \\ & \sum_I N_I = 1, \quad \sum_I N_I x_I = x \end{aligned} \quad (61)$$

with $\beta = \gamma/h^2$ representing a non-negative locality coefficient, where γ is a dimensionless parameter and h is a measure of nodal spacing. The value of γ is always chosen in a range between 0.6, relative to spread-out meshfree shape functions, and four, relative to linear finite element basis functions. Unlike MLS approximants, the LME basis functions possess the weak Kronecker-delta property at the boundary of the convex hull of the nodes, and they are a C^∞ function of β in $(0, +\infty)$. However, the computation of the LME approximation scheme is more onerous than MLS basis functions, as the problem described by Equation (61) is a convex problem. In Section 5, a comparison between these two procedures is performed through some benchmark tests, and an assessment in terms of computational cost, accuracy and robustness is provided.

5. Numerical Examples

In this section, three benchmark tests are considered for the comparison of the MPM and GMM formulations. Firstly, the static analysis of a 2D cantilever beam subjected to its self-weight is analyzed, and a mesh convergence study is performed. Secondly, the rolling of a rigid disk on an inclined plane is studied. Finally, a cohesive-soil column collapse is analyzed. All the numerical experiments have been performed on a PC with one Intel(R) Core(TM) i7-4790 CPU at 3.60 GHz.

5.1. 2D Cantilever Beam: Static Analysis

The static analysis of a 2D cantilever beam subjected to its self-weight under the assumption of plain strain is presented. The cantilever beam has a length $l = 8$ m and a square cross-section of unit side ($b = h = 1$ m) (Figure 4). The beam is modeled with a hyperelastic material: the density is $\rho = 1000$ kg/m³; the Young's modulus is $E = 90$ MPa; and the Poisson's ratio is $\nu = 0$. The results obtained with the MPM and GMM algorithms are compared with a standard FEM code using the same UL formulation.

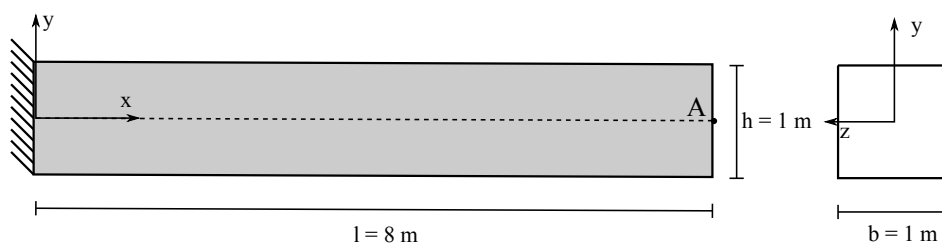


Figure 4. Static 2D cantilever analysis: geometry.

A mesh convergence study is carried out adopting five different mesh sizes, $h = 0.5, 0.25, 0.125, 0.0625$ and 0.01 m, respectively. Quadrilateral elements are used in FEM, MPM and GMM with four integration points per cell (in the case of MPM and GMM, the integration points coincide with the material points). In GMM, the mesh is only initially used for the creation of the material points and then deleted. Regarding the spatial search and the evaluation of the shape functions in GMM, a search radius $R = \sqrt{2}h^2$, dilation parameters $R_{eff} = R/2$ and $\gamma = 1.8$ are adopted in GMM-MLS

and GMM-LME, respectively. Under the assumption of a linear regime, the vertical deflection at Point A of the free edge can be evaluated analytically according to Timoshenko [58] as:

$$\delta = - \left(\frac{\rho g (bhl)^3}{8EI} + \frac{\rho g l^2}{2GA_s} \right) = -0.67806m \quad (62)$$

where g is the gravity acceleration, $I = \frac{bh^3}{12}$ the inertia of the beam section and $A_s = \frac{5}{6}A$ the reduced cross-section area due to the shear effect. However, the solution is computed under the assumption of non-linearity, and as a benchmark solution, the deflection evaluated through the finest mesh is considered. This value is $\delta = -0.67433$ m and is equally reached by all the methods.

Figures 5 and 6 compare the solutions obtained with an updated Lagrangian FEM, MPM, GMM-MLS and GMM-LME code, respectively, in terms of vertical displacement and Cauchy stress along the horizontal direction. One can observe that the results are in good agreement for all the methods.

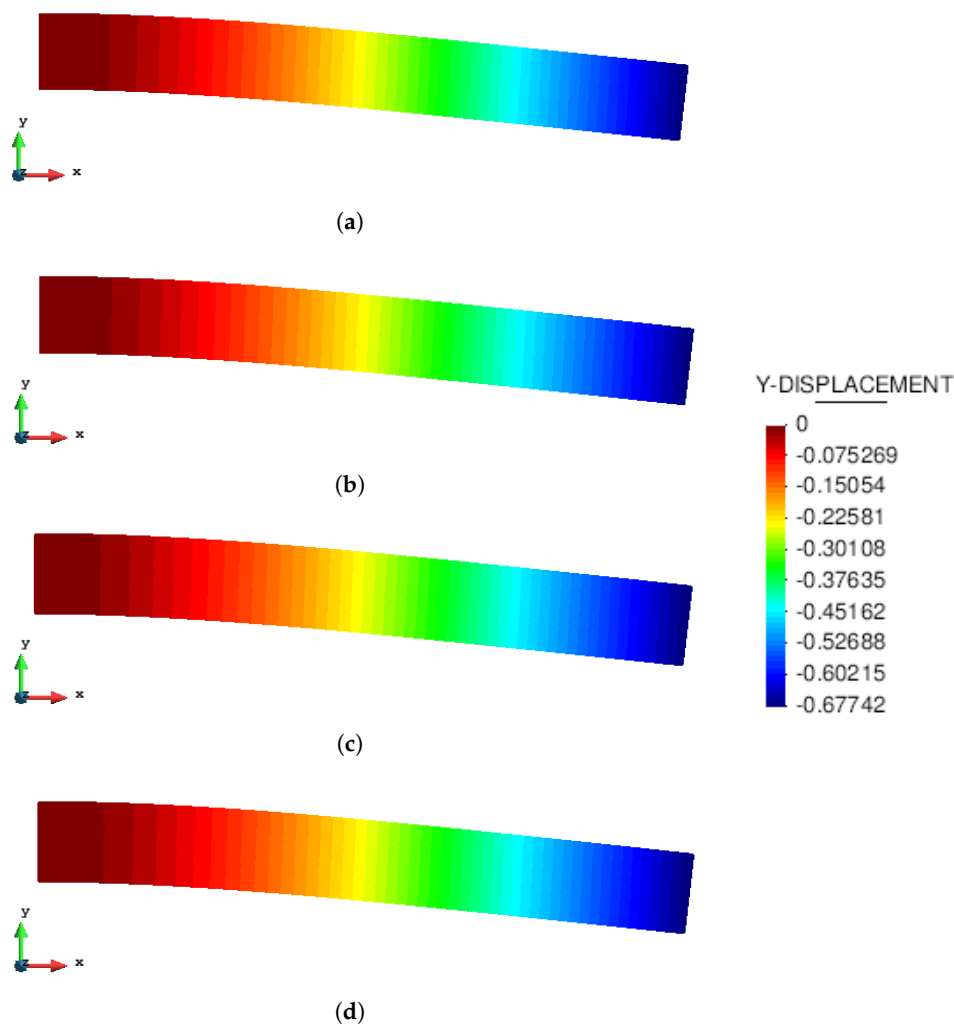


Figure 5. Static cantilever. Displacement along the y-direction.(a) FEM code; (b) MPM code; (c) GMM-MLS code; (d) GMM-LME code.

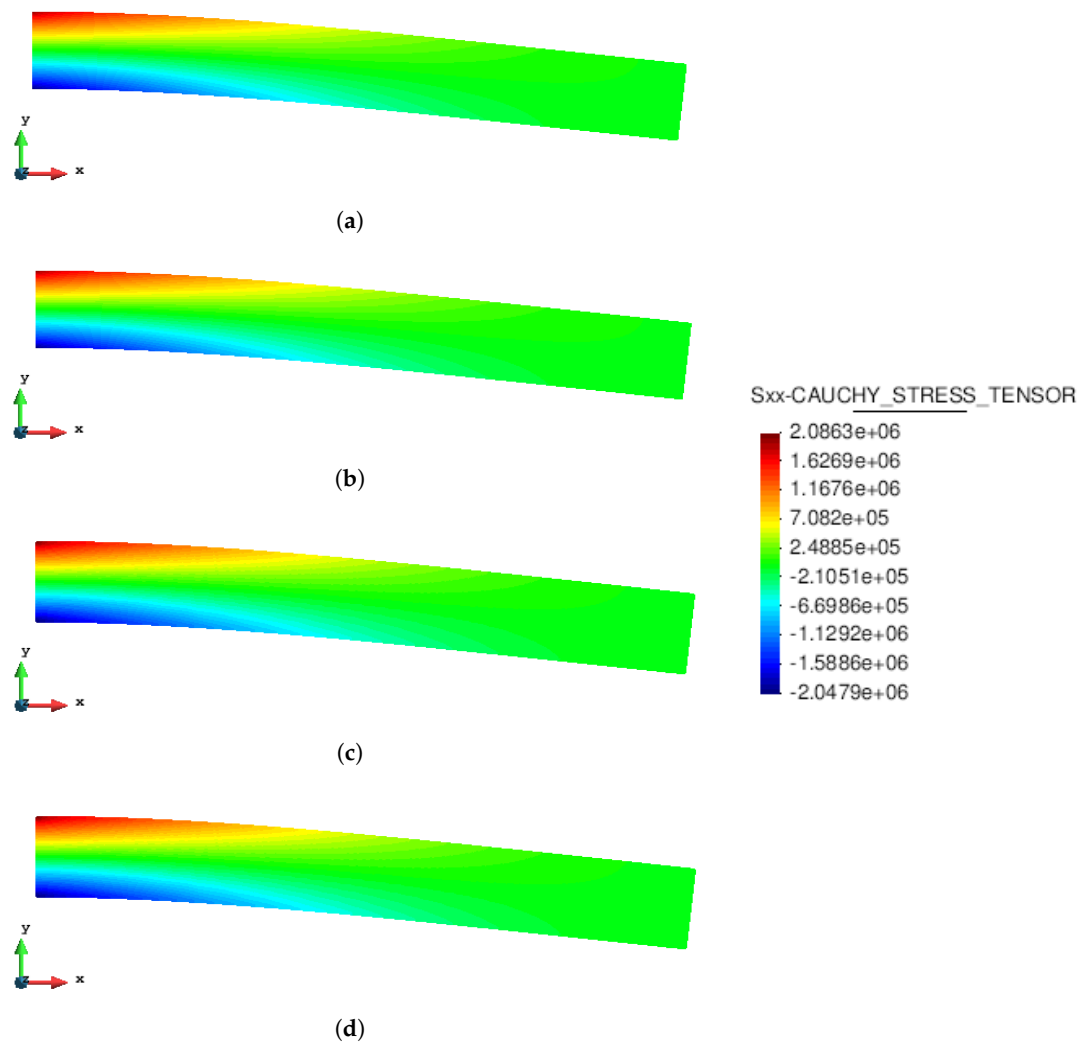


Figure 6. Static cantilever. Cauchy stress along the x-direction. (a) FEM code; (b) MPM code; (c) GMM-MLS code; (d) GMM-LME code.

A convergence study is performed to analyse the accuracy of MPM and GMM in comparison with the UL-FEM. The error is evaluated as:

$$error = \left| \frac{\delta - u_{num}}{\delta} \right| \quad (63)$$

where u_{num} is the numerical solution measure at Point A (see Figure 4). Figure 7 depicts the error evolution as a function of the inverse of the mesh size h . It is demonstrated that all the methods have a quadratic rate of convergence. In particular, the UL-FEM, MPM and GMM-MLS error curves coincide. Regarding the error, evaluated with the GMM-LME algorithm, the quadratic rate is maintained, but the curve is shifted a bit upwards, which makes this technique less accurate than GMM-MLS in the benchmark case studied.

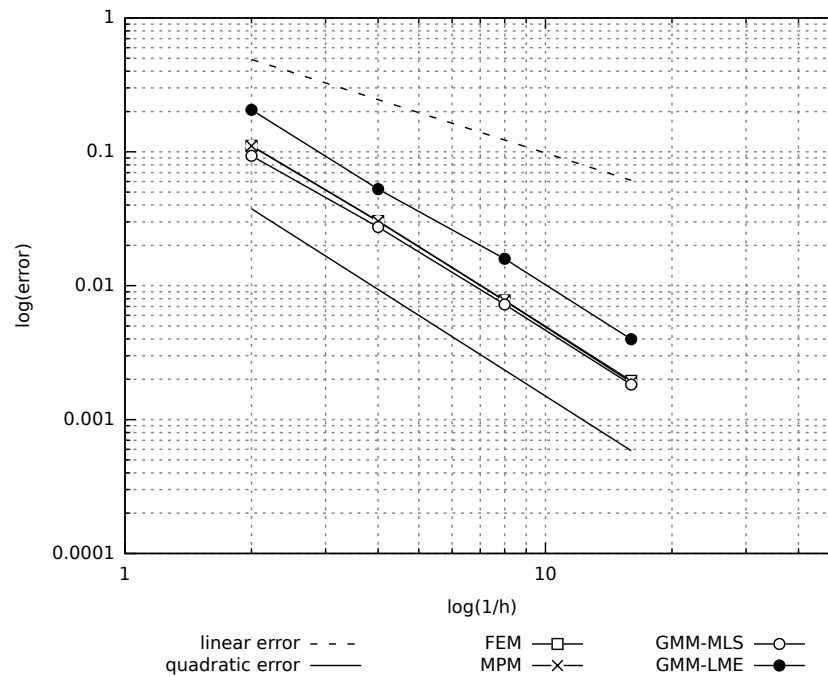


Figure 7. Static cantilever. Convergence analysis.

5.2. Rolling of a Rigid Disk on an Inclined Plane

The second benchmark test is a rigid disk rolling without slipping on an inclined plane. The geometry of the problem is depicted in Figure 8. The disk is made of a hyperelastic material: the density is $\rho = 7800 \text{ kg/m}^3$; the Young's modulus is $E = 200 \text{ MPa}$; and the Poisson's ratio is $\nu = 0.3$.

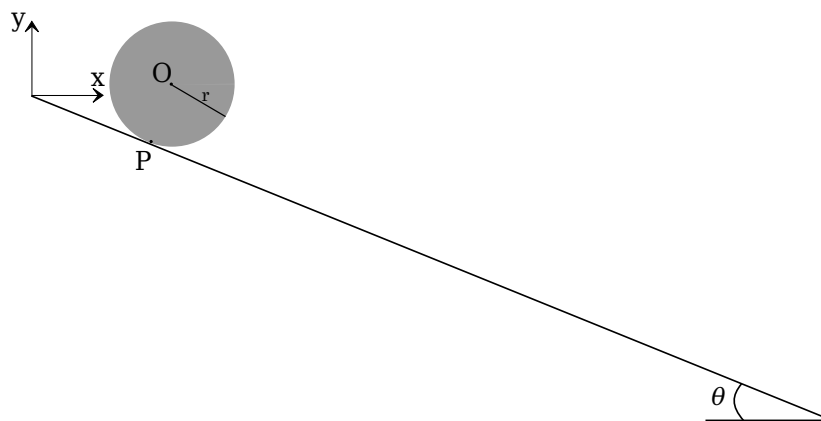


Figure 8. Rolling disk. Geometry.

This test is chosen for an objective assessment of the robustness of the MPM and GMM algorithm. The rolling on the plane implies a contact between the nodes belonging to the inclined plane and the nodes belonging to the disk. In a UL-FEM code, a contact algorithm would be necessary to set this boundary condition. On the contrary, by using either MPM or GMM, the contact is implicitly caught. The analytical acceleration (a) can be computed imposing the equilibrium of momentum at the contact point P :

$$a = \frac{2}{3}g\sin\theta \quad (64)$$

where g is gravity and θ the angle of the inclined plane. Integrating over time, the acceleration, velocity and displacement projected on the x-axis can be obtained as a function of time:

$$v(t) = a(t)\cos \theta t \quad (65)$$

$$u(t) = v(t)\cos \theta t + \frac{1}{2}a(t)\cos \theta t^2 \quad (66)$$

For the study of this test case, the analytical solution of Equation (66) is used for the assessment of the absolute error obtained with MPM, GMM-MLS and GMM-LME, evaluated as:

$$error = \sqrt{\sum_{t_i} (u(t_i) - u_{num}(t_i))^2} \quad (67)$$

where t_i is the time where the numerical result is calculated. As mesh-based and meshless techniques are compared in a dynamic test, for a more objective comparison, the error is analyzed along with the total computational time needed to finalize the simulation.

A triangular mesh with mesh size $h = 0.01$ m is used for MPM and GMM simulations. In both techniques, the same initial distribution of material points is used, which counts for three initial particles for the cell. Regarding the GMM-MLS the approximants are constructed by adopting a search radius $R = 1.5\sqrt{2h^2}$ and a dilation parameter $R_{eff} = \sqrt{2h^2}$. In GMM-LME, the basis functions are evaluated using a search radius $R = \sqrt{2h^2}$ and three values of dilation parameter $\gamma = 0.8, 1.8, 2.8$. All the numerical tests are repeated for three different time steps with $\Delta t_1 = 2\Delta t_2 = 4\Delta t_3$.

Table 1 shows the results of the analysis, in terms of errors and computational times, performed through MPM, GMM-MLS and GMM-LME.

Table 1. Rolling disk. Absolute errors and computational times.

Technique	Δt_1		Δt_2		Δt_3	
	error (m)	t_{comp} (s)	error (m)	t_{comp} (s)	error (m)	t_{comp} (s)
MPM	2.34	232.72	1.27	472.92	0.91	894.91
GMM-MLS	0.84	264.78	0.26	517	0.20	981.82
GMM-LME $\gamma = 0.8$	1.37	234.32	0.07	460.73	0.06	971.29
GMM-LME $\gamma = 1.8$	0.9	237.22	0.23	460.50	0.07	1005.21
GMM-LME $\gamma = 2.8$	0.52	232.42	0.07	466.63	0.06	1038.70

Regarding the absolute errors, it can be observed that, for a given computational cost, GMM is generally more accurate than MPM, because of the use of smooth basis functions, which provide a better approximation of the unknown variables. In particular, GMM-LME presents smaller errors in comparison to GMM-MLS. In all three cases considered (with $\gamma = 0.8, 1.8, 2.8$), the errors converge to a unique value at the same computational time, while in the case of GMM-MLS, the advantage of using higher order elements is lost for the smallest delta time. Regarding MPM, it is established that to achieve the same order of accuracy of GMM, a higher computational time must be expected, due to either a finer discretisation in space or in time. However, it is worth highlighting that GMM is much more time consuming than MPM, showing an increment of computational time of 10% in the case of GMM-MLS and from 8.5% up to 16% in the case of GMM-LME.

In this example, some essential conclusions can be drawn. In Section 5.1, it was observed that in a static case, the rate of convergence is the same for all the methods under analysis, but the accuracy of GMM-MLS and MPM is better than that of GMM-LME. On the contrary, in a dynamic case with a contact problem, the result is overturned. In fact, a better behavior is noted if LME approximants are employed.

5.3. Cohesive Soil Column Collapse

The third example is the simulation of a soil column collapse. The column is modeled with a cohesive–frictional material, defined by a cohesion $c = 5$ kPa, a friction angle $\phi = 25^\circ$, an elastic bulk modulus $K = 1.5$ MPa and a density $\rho = 1850$ kg/m³. In the current work, a Mohr–Coulomb plastic law in finite strains is considered, and the implicit integration scheme in principal stress space, presented in [59,60], is followed for its implementation.

This test has been chosen for the assessment of the robustness of MPM and GMM when the body undergoes really large deformation. The results are compared with the work of [31], where a Smooth Particle Hydrodynamics method (SPH) is applied to geotechnical problems.

The initial geometry and the boundary conditions are described by Figure 9.

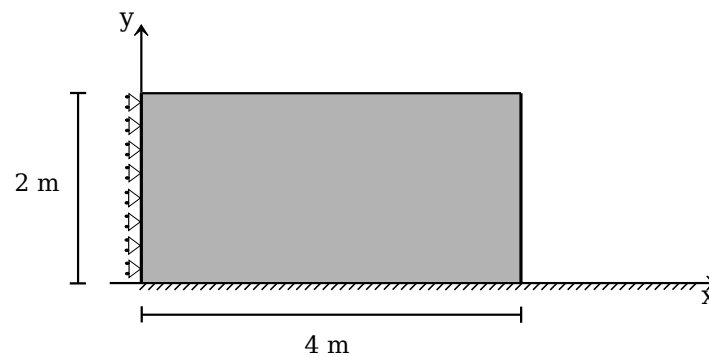


Figure 9. Granular column collapse. Geometry.

Quadrilateral elements with an initial distribution of four material points per cell are used in the simulations. Two different mesh sizes are considered: $h_1 = 0.05$ m (Mesh 1) and $h_2 = 0.025$ m (Mesh 2). In GMM, the basis functions are evaluated using an initial search radius $R = \sqrt{2h^2}$, a dilation parameter $R_{eff} = 0.5\sqrt{2h^2}$ and $\gamma = 1.8$, in GMM-MLS and GMM-LME, respectively. In this particular case, the procedure for the evaluation of the basis functions in the MLS and LME techniques has been modified to avoid the creation of a non-convex hull of nodes, which might lead to an incorrect set of approximants. This is required because the column is subjected to extremely large deformations. While in the previous examples, a constant radius was used for the definition of the cloud of nodes surrounding a material point, in the current example, a variable radius is adopted to guarantee a minimum number of nodes in each connectivity. In the case of LME, as a Newton iterative procedure is used for the evaluation of the shape functions, a measure of the goodness of the solution is represented by the condition number $k(A)$ of the Hessian matrix A , defined in [56]. If $k(A)$ exceeds a user-defined tolerance, the LME algorithm is repeated considering the old connectivity plus an additional node, chosen as the next node closer to the material point. In the case of MLS, it has been sufficient to impose a minimum number of six nodes in each cloud of nodes.

In Figure 10a, a comparison of the column deformation at different representative time instants is shown. The SPH model taken from [31] predicts a higher final run-out of the column collapse, while the final configurations at time 2.0 s of MPM, GMM-LME and GMM-MLS are almost coincident using Mesh 1 and Mesh 2. It is worth highlighting that GMM-MLS and MPM results of Figure 10 are always in good agreement. However, this is not the case if the evolution of the equivalent plastic strains is observed (see Figures 11 and 12). In the case of GMM-LME, an improvement of the results is noted by using the finer mesh (Mesh 2) in terms of displacements (Figure 10b) and an equivalent plastic strain distribution (Figure 13b). Regarding MPM, it is proven that a good approximation can be obtained using both meshes.

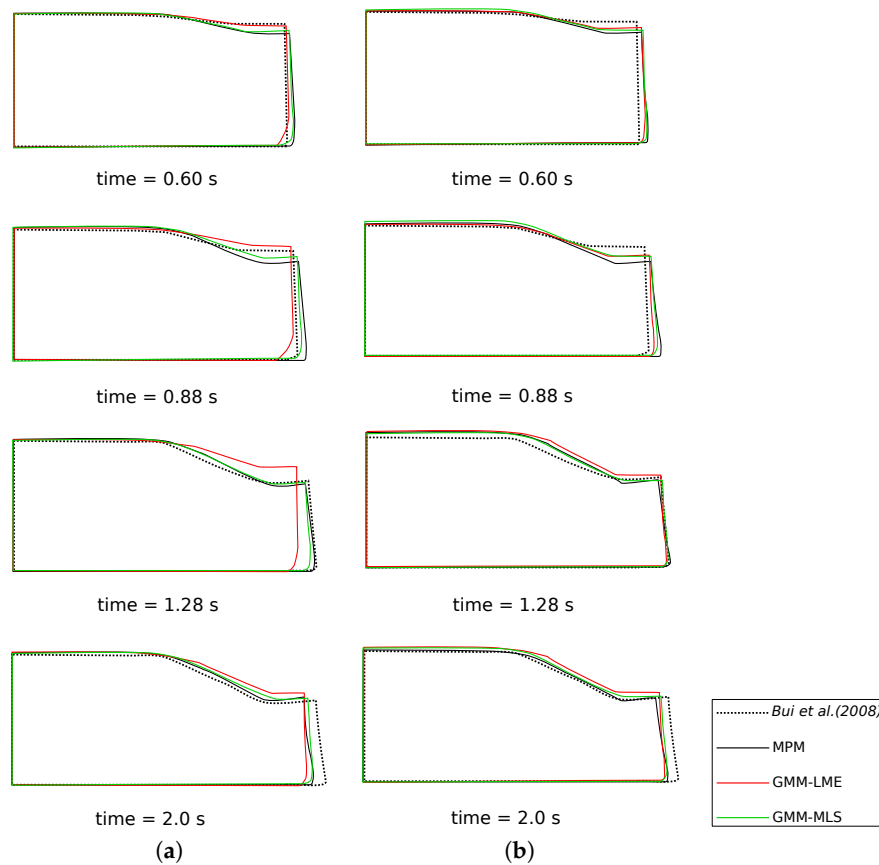


Figure 10. Soil column collapse. Configurations of the column at different representative time instants. (a) Mesh 1; (b) Mesh 2

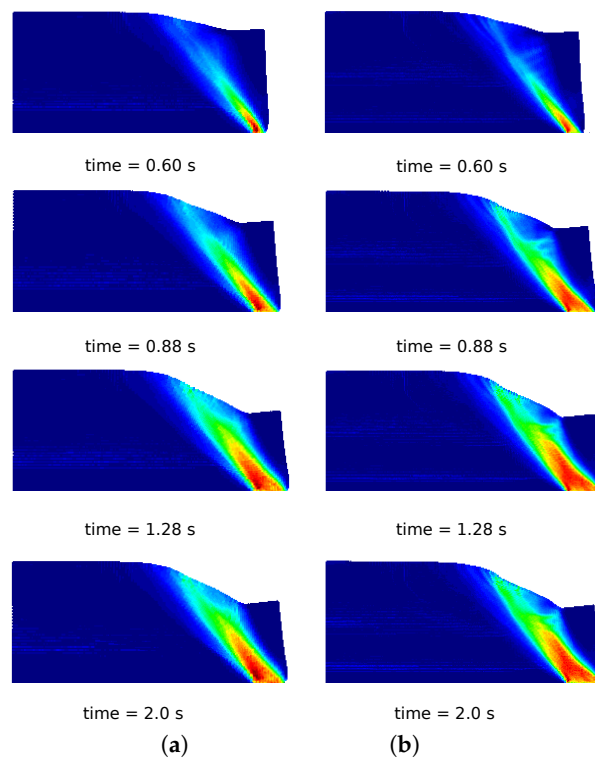


Figure 11. Soil column collapse. Distribution of equivalent plastic strains for different representative time instants in the MPM results. (a) Mesh 1; (b) Mesh 2.

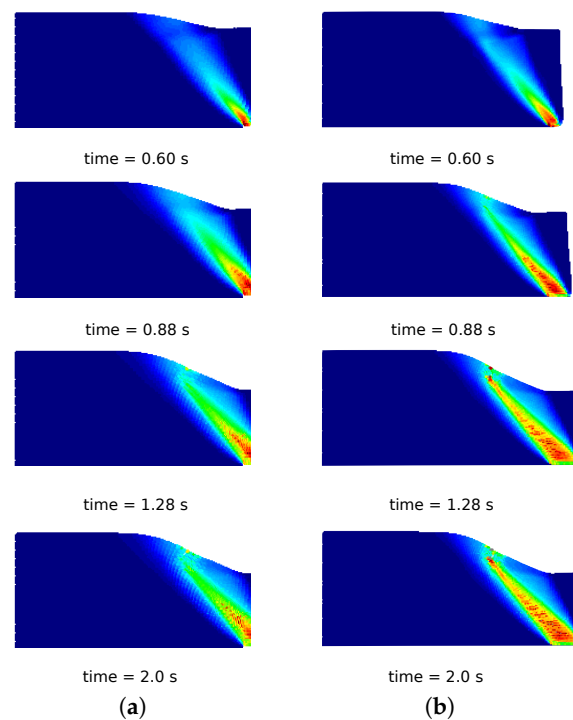


Figure 12. Soil column collapse. Distribution of equivalent plastic strains for different representative time instants in the GMM-MLS results. (a) Mesh 1; (b) Mesh 2.

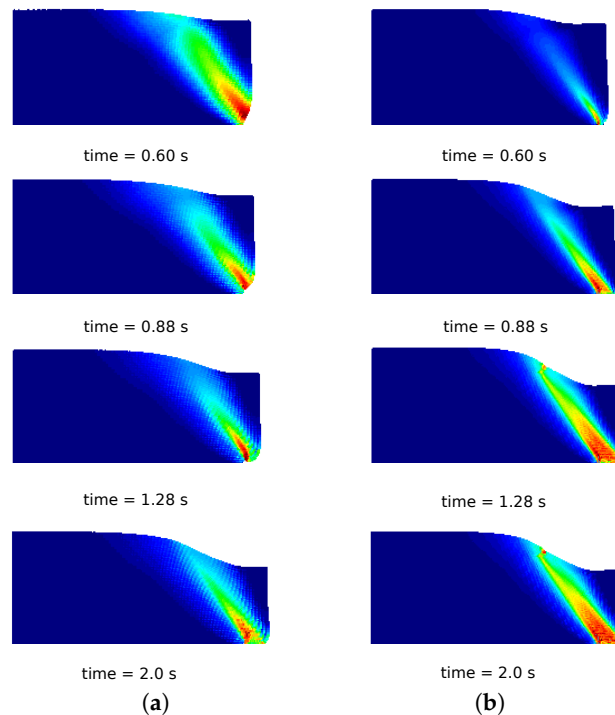


Figure 13. Soil column collapse. Distribution of equivalent plastic strains for different representative time instants in the GMM-LME results. (a) Mesh 1; (b) Mesh 2.

In this example, the capability of handling history-dependent materials, such as cohesive-frictional materials, is verified for both methods. It is noted that, in MPM, large deformations can be naturally tracked without modifying the algorithm, and accurate results are obtained also using the coarser mesh. In GMM, despite the remarkable features highlighted in the previous benchmark tests, when

the continuum undergoes extremely large deformations, special care should be taken in the definition of the MLS and LME approximants. In this regard, a lack of robustness of the GMM algorithm is observed due to the impossibility of guaranteeing a correct evaluation of the shape functions during the whole deformation process without an ad hoc modification of the procedure for the definition of the connectivity. Thus, for the solution of this example, a correction of the algorithm has been performed and verified to work properly, albeit an increase in the computational time is registered. The establishment of a more general procedure is left for future work.

6. Conclusions

In the present paper, two particle methods—a material point method and a Galerkin meshless method—are tested and compared to assess their capabilities in solving large displacement and large deformation problems. A variational displacement-based formulation, based on an updated Lagrangian description, is presented, and the algorithms are described in detail.

A comparison of MPM and GMM is performed through three benchmark tests, and the methods are assessed in terms of accuracy, computational time and robustness. The first example is a static cantilever beam. A convergence analysis is performed, and all the techniques have a quadratic convergence rate (compared to an FEM code). Secondly, the dynamic test of a rolling disk on an inclined plane is considered. The robustness of MPM and GMM in dealing with contact between two rigid bodies is tested, and an analysis in terms of computational time and error is performed. We found that GMM, in dynamic cases, has a higher accuracy than MPM, despite a higher computational cost. This is because in MPM, linear basis functions are considered, while in GMM, smooth basis functions are computed allowing one to obtain a superior approximation of the unknown variables. As a last example, a cohesive soil column collapse is analyzed. In this case, we assess the robustness of both methods when the continuum undergoes extremely large deformation. Firstly, it is demonstrated that MPM and GMM can be easily coupled with local plastic laws. Furthermore, it is noted that MPM leads to more accurate results, and the algorithm does not need to be modified in a large deformation case. On the contrary, in GMM, a modification of the algorithm has to be considered to avoid the formation of a non-convex hull of nodes, when the connectivity is defined. Nonetheless, in spite of this modification, a discrepancy in the results is noted, by using either the MLS or the LME technique.

In conclusion, the standard version of MPM represents a good choice to handle problems involving history-dependent materials and large deformations. Regarding GMM, the accuracy of the solution strictly depends on the basis functions chosen. If large deformation of the continuum is not taken into account, this method could be preferred to MPM due to its remarkable feature in obtaining accurate results in a limited computational time. However, under the finite strain regime, independent of the material to model, the construction of a connectivity in the meshless method becomes more complex, and at least to the authors' knowledge, a general methodology is still missing to properly define the correct connectivity under any deformation condition. Thus, despite the promising features of this approach, an improvement in the robustness of the GMM algorithm is needed to obtain more accurate and reliable solutions in large deformation and failure problems.

Acknowledgments: The research was supported by the Research Executive Agency through the “Training in Multiscale Analysis of MultiPhase Particulate Processes and Systems” (T-MAPPP) project (FP7 PEOPLE 2013 ITN-G.A.n607453). The Spanish Ministry of Economy and Competitiveness (Ministerio de Economía y Competitividad, MINECO) through the project “Desarrollo y validación de una aplicación para la determinación del Hidrograma de Rotura de presas de MAteriales sueltos, a partir de la configuración geomecánica particular” (HIRMA) (RTC-2016-4967-5) is also greatly acknowledged.

Author Contributions: Ilaria Iaconeta, Antonia Larese and Riccardo Rossi conceived of and developed the formulations presented in the current paper. Ilaria Iaconeta implemented the MPM algorithm. Zhiming Guo implemented the GMM algorithm with the help of Riccardo Rossi and Ilaria Iaconeta. Ilaria Iaconeta performed the examples. Antonia Larese and Riccardo Rossi supervised and coordinated the work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cundall, P.A.; Strack, O.D.L. *A Discrete Numerical Model for Granular Assemblies*; ICE Virtual Library: London, UK, 1979.
2. O’Sullivan, C. Particle-based discrete element modeling: Geomechanics perspective. *Int. J. Geomech.* **2011**, *11*, 449–464, doi:10.1061/(ASCE)GM.1943-5622.0000024.
3. Cervera, M.; Chiumenti, M.; Codina, R. Mixed stabilized finite element methods in nonlinear solid mechanics: Part II: Strain localization. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 2571–2589, doi:10.1016/j.cma.2010.04.005.
4. Wu, J.Y.; Cervera, M. *Strain Localization and Failure Mechanics for Elastoplastic Damage Solids*; International Center for Numerical Methods in Engineering: Barcelona, Spain, 2014.
5. Benedetti, L.; Cervera, M.; Chiumenti, M. Stress-accurate mixed fem for soil failure under shallow foundations involving strain localization in plasticity. *Comput. Geotech.* **2015**, *64*, 32–47, doi:10.1016/j.compgeo.2014.10.004.
6. Cervera, M.; Lafontaine, N.; Rossi, R.; Chiumenti, M. Explicit mixed strain—Displacement finite elements for compressible and quasi-incompressible elasticity and plasticity. *Comput. Mech.* **2016**, *58*, 511–532, doi:10.1007/s00466-016-1305-z.
7. Simo, J.; Hughes, T. *Computational Inelasticity*; Springer: New York, NY, USA, 1998.
8. Donea, J.; Giuliani, S.; Halleux, J.P. An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions. *Comput. Methods Appl. Mech. Eng.* **1982**, *33*, 689–723, doi:10.1016/0045-7825(82)90128-1.
9. Dialami, N.; Chiumenti, M.; Cervera, M.; de Saracibar, C.A. An apropos kinematic framework for the numerical modeling of friction stir welding. *Comput. Struct.* **2013**, *117*, 48–57, doi:10.1016/j.compstruc.2012.12.006.
10. Nazem, M.; Sheng, D.; Carter, J.P. Stress integration and mesh refinement for large deformation in geomechanics. *Int. J. Numer. Methods Eng.* **2006**, *65*, 1002–1027, doi:10.1002/nme.1470.
11. Idelsohn, S.R.; Oñate, E.; Pin, F.D. The Particle Finite Element Method: A powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int. J. Numer. Methods Eng.* **2004**, *61*, 964–984, doi:10.1002/nme.1096.
12. Larese, A.; Rossi, R.; Oñate, E.; Idelsohn, S. Validation of the particle finite element method (PFEM) for simulation of free surface flows. *Eng. Comput.* **2008**, *25*, 385–425, doi:10.1108/02644400810874976.
13. Carbonell, J.M.; Oñate, E.; Suárez, B. Modeling of Ground Excavation with the Particle Finite Element Method. *J. Eng. Mech.* **2010**, *136*, 455–463, doi:10.1061/(ASCE)EM.1943-7889.0000086.
14. Cante, J.; Davalos, C.; Hernandez, J.; Oliver, J.; Jonsén, P.; Gustafsson, G.; Häggblad, H. PFEM-based modeling of industrial granular flows. *Comput. Part. Mech.* **2014**, *1*, 47–70, doi:10.1007/s40571-014-0004-9.
15. Franci, A.; Oñate, E.; Carbonell, J.M. Unified Lagrangian formulation for solid and fluid mechanics and FSI problems. *Comput. Methods Appl. Mech. Eng.* **2016**, *298*, 520–547, doi:10.1016/j.cma.2015.09.023.
16. Cremonesi, M.; Ferrara, L.; Frangi, A.; Perego, U. Simulation of the flow of fresh cement suspensions by a Lagrangian finite element approach. *J. Non-Newton. Fluid Mech.* **2010**, *165*, 1555–1563, doi:10.1016/j.jnnfm.2010.08.003.
17. Cremonesi, M.; Frangi, A.; Perego, U. A Lagrangian finite element approach for the simulation of water-waves induced by landslides. *Comput. Struct.* **2011**, *89*, 1086–1093, doi:10.1016/j.compstruc.2010.12.005.
18. Larese, A.; Rossi, R.; Oñate, E.; Toledo, M. Physical and Numerical Modelization of the Behavior of rockfill dams during overtopping scenarios. In *Dam Maintenance and Rehabilitation II*; CRCpress Balkema: Leiden, The Netherlands, 2010; pp. 479–487.
19. Larese, A.; Rossi, R.; Oñate, E.; Idelsohn, S. A coupled PFEM—Eulerian approach for the solution of porous FSI problems. *Comput. Mech.* **2012**, *50*, 805–819, doi:10.1007/s00466-012-0768-9.
20. Larese, A.; Rossi, R.; Oñate, E.; Toledo, M.; Moran, R.; Campos, H. Numerical and experimental study of overtopping and failure of rockfill dams. *Int. J. Geomech.* **2013**, *15*, 04014060, doi:10.1061/(ASCE)GM.1943-5622.0000345.
21. Salazar, F.; Irazabal, J.; Larese, A.; Oñate, E. Numerical modeling of landslide-generated waves with the particle finite element method (PFEM) and a non-Newtonian flow model. *Int. J. Numer. Anal. Methods Geomech.* **2016**, *40*, 809–926, doi:10.1002/nag.2428.

22. Monforte, L.; Carbonell, J.M.; Arroyo, M.; Gens, A. Performance of mixed formulations for the particle finite element method in soil mechanics problems. *Comput. Part. Mech.* **2017**, *4*, 269–284, doi:10.1007/s40571-016-0145-0.
23. Monforte, L.; Arroyo, M.; Carbonell, J.M.; Gens, A. Numerical simulation of undrained insertion problems in geotechnical engineering with the Particle Finite Element Method (PFEM). *Comput. Geotech.* **2017**, *82*, 144–156, doi:10.1007/s40571-016-0145-0.
24. Oñate, E.; Franci, A.; Carbonell, J.M. Lagrangian formulation for finite element analysis of quasi-incompressible fluids with reduced mass losses. *Int. J. Numer. Methods Fluids* **2014**, *74*, 699–731, doi:10.1002/flid.3870.
25. Ryzhakov, P.; Oñate, E.; Rossi, R.; Idelsohn, S. Improving mass conservation in simulation of incompressible flows. *Int. J. Numer. Methods Eng.* **2012**, *90*, 1435–1451, doi:10.1002/nme.3370.
26. Franci, A.; Cremonesi, M. On the effect of standard PFEM remeshing on volume conservation in free-surface fluid flow problems. *Comput. Part. Mech.* **2017**, *4*, 331–343, doi:10.1007/s40571-016-0124-5.
27. Lucy, L.B. A numerical approach to the testing of the fission hypothesis. *Astron. J.* **1977**, *82*, 1013–1024, doi:10.1086/112164.
28. Gingold, R.; Monaghan, J. Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* **1977**, *181*, 375–389, doi:10.1093/mnras/181.3.375.
29. Monaghan, J.J. Simulating free surface flows with SPH. *J. Comput. Phys.* **1994**, *110*, 399–406, doi:10.1006/jcph.1994.1034.
30. Dalrymple, R.; Rogers, B. Numerical modeling of water waves with the SPH method. *Coast. Eng.* **2006**, *53*, 141–147, doi:10.1016/j.coastaleng.2005.10.004.
31. Bui, H.H.; Fukagawa, R.; Sako, K.; Ohno, S. Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic–plastic soil constitutive model. *Int. J. Numer. Anal. Methods Geomech.* **2008**, *32*, 1537–1570, doi:10.1002/nag.688.
32. Pastor, M.; Haddad, B.; Sorbino, G.; Cuomo, S.; Drempetic, V. A depth-integrated, coupled SPH model for flow-like landslides and related phenomena. *Int. J. Numer. Anal. Methods Geomech.* **2009**, *33*, 143–172, doi:10.1002/nag.705.
33. Bui, H.; Fukagawa, R.; Sako, K.; Wells, J. Slope stability analysis and discontinuous slope failure simulation by elasto-plastic smoothed particle hydrodynamics (SPH). *Géotechnique* **2011**, *61*, 565–574, doi:10.1680/geot.9.P.046.
34. Nonoyama, H.; Moriguchi, S.; Sawada, K.; Yashima, A. Slope stability analysis using smoothed particle hydrodynamics (SPH) method. *Soils Found.* **2015**, *55*, 458–470, doi:10.1016/j.sandf.2015.02.019.
35. Sulsky, D.; Chen, Z.; Schreyer, H. A particle method for history-dependent materials. *Comput. Methods Appl. Mech. Eng.* **1994**, *118*, 179–196, doi:10.1016/0045-7825(94)90112-0.
36. Sulsky, D.; Zhou, S.J.; Schreyer, H.L. Application of a particle-in-cell method to solid mechanics. *Comput. Phys. Commun.* **1995**, *87*, 236–252, doi:10.1016/0010-4655(94)00170-7.
37. Urrecha Espluga, M. Analysis of Meshfree Methods for Lagrangian Fluid-Structure Interaction. Ph.D. Thesis, Universidad Politécnica de Madrid, Madrid, Spain, 2014.
38. Zabala, F.; Alonso, E. Progressive failure of Aznalcóllar dam using the material point method. *Géotechnique* **2011**, *61*, 795–808, doi:10.1680/geot.9.P.134.
39. Yerro, A.; Alonso, E.; Pinyol, N. The material point method for unsaturated soils. *Géotechnique* **2015**, *65*, 201–217, doi:10.1680/geot.14.P.163.
40. Sołowski, W.; Sloan, S. Evaluation of material point method for use in geotechnics. *Int. J. Numer. Anal. Methods Geomech.* **2015**, *39*, 685–701, doi:10.1002/nag.2321.
41. Guilkey, J.E.; Weiss, J.A. Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method. *Int. J. Numer. Methods Eng.* **2003**, *57*, 1323–1338, doi:10.1002/nme.729.
42. Beuth, L.; Wieckowski, Z.; Vermeer, P. Solution of quasi-static large-strain problems by the material point method. *Int. J. Numer. Anal. Methods Geomech.* **2011**, *35*, 1451–1465, doi:10.1002/nag.965.
43. Sanchez, J.; Schreyer, H.; Sulsky, D.; Wallstedt, P. Solving quasi-static equations with the material-point method. *Int. J. Numer. Methods Eng.* **2015**, *103*, 60–78.
44. Belytschko, T.; Lu, Y.Y.; Gu, L. Element-free Galerkin methods. *Int. J. Numer. Methods Eng.* **1994**, *37*, 229–256, doi:10.1002/nme.1620370205.
45. Liu, W.K.; Jun, S.; Zhang, Y.F. Reproducing kernel particle methods. *Int. J. Numer. Methods Fluids* **1995**, *20*, 1081–1106, doi:10.1002/flid.1650200824.

46. Dadvand, P. A Framework for Developing Finite Element Codes for Multi-Disciplinary Applications. Ph.D. Thesis, Universidad Politécnica de Cataluña, Barcelona, Spain, 2007.
47. Dadvand, P.; Rossi, R.; Oñate, E. An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Arch. Comput. Methods Eng.* **2010**, *17*, 253–297, doi:10.1007/s11831-010-9045-2.
48. Wriggers, P. *Computational Contact Mechanics*; Springer: Berlin, Germany, 2006; p. 521.
49. Harlow, F. The particle-in-cell computing method for fluid dynamics. *Methods Comput. Phys.* **1964**, *3*, 319–343.
50. Wang, B.; Vardon, P.; Hicks, M.; Chen, Z. Development of an implicit material point method for geotechnical applications. *Comput. Geotech.* **2016**, *71*, 159–167, doi:10.1016/j.compgeo.2015.08.008.
51. Fries, T.; Matties, H. *Classification and Overview of Meshfree Methods*; Technical Report; Department of Mathematics and Computer Sciences, TU Braunschweig: Braunschweig, Germany, 2004.
52. Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P. Meshless methods. An overview and recent developments. *Comput. Methods Appl. Mech. Eng.* **1996**, *139*, 3–47, doi:10.1016/S0045-7825(96)01078-X.
53. Li, S.; Liu, W. Meshfree and particle methods and their applications. *Appl. Mech. Rev.* **2002**, *55*, 1–34, doi:10.1115/1.1431547.
54. Lancaster, P.; Salkauskas, K. Surfaces Generated by Moving Least Squares Methods. *Math. Comput.* **2010**, *37*, 141–158, doi:10.1090/S0025-5718-1981-0616367-1.
55. Felippa, C. *Introduction to Finite Element Methods*; University of Colorado at Boulder: Boulder, CO, USA, 2015.
56. Arroyo, M.; Ortiz, M. Local maximum-entropy approximation schemes: A seamless bridge between finite elements and meshfree methods. *Int. J. Numer. Methods Eng.* **2006**, *65*, 2167–2202, doi:10.1002/nme.1534.
57. Jaynes, E.T. Information theory and statistical mechanics. *Phys. Rev.* **1957**, *106*, 620, doi:10.1103/PhysRev.106.620.
58. Timoshenko, S.; Goodier, J. *Theory Elasticity*; McGraw-Hill: New York, NY, USA, 1951; p. 506.
59. Clausen, J.; Damkilde, L.; Andersen, L. Efficient return algorithms for associated plasticity with multiple yield planes. *Int. J. Numer. Methods Eng.* **2006**, *66*, 1036–1059, doi:10.1002/nme.1595.
60. Clausen, J.; Damkilde, L.; Andersen, L. An efficient return algorithm for non-associated plasticity with linear yield criteria in principal stress space. *Comput. Struct.* **2007**, *85*, 1795–1807, doi:10.1016/j.compstruc.2007.04.002.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).