RESEARCH PAPER

Nonconforming Dirichlet boundary conditions in implicit material point method by means of penalty augmentation

Bodhinanda Chandra[1,2] · Veronika Singer[1] · Tobias Teschemacher[1] · Roland Wüchner[1] ·
Antonia Larese[3]

Abstract

In many geomechanics applications, material boundaries are subjected to large displacements and deformation. Under these circumstances, the application of boundary conditions using particle methods, such as the material point method (MPM), becomes a challenging task since material boundaries do not coincide with the background mesh. This paper presents a formulation of penalty augmentation to impose nonhomogeneous, nonconforming Dirichlet boundary conditions in implicit MPM. The penalty augmentation is implemented utilizing boundary particles, which can move either according to or independently from the material deformation. Furthermore, releasing contact boundary condition, as well as the capability to accommodate slip boundaries, is introduced in the current work. The accuracy of the proposed method is assessed in both 2D and 3D cases, by convergence analysis reaching the analytical solution and by comparing the results of nonconforming and classical grid-conforming simulations.

Keywords Implicit time integration · Material point method · Nonconforming boundary conditions · Penalty method

1 Introduction

In recent years, continuum-based particle methods, including, among others, the smoothed particle hydrodynamics (SPH) [1], the particle finite element method (PFEM) [2], the reproducing kernel particle method [3], the element-free Galerkin method [4], and the moving particle semi-implicit method (MPS) [5], are the natural choice when simulating large deformation solid dynamics problems. While classical mesh-based approaches, such as the finite element method (FEM) [6], have been proven to be very powerful and accurate to model materials under small deformation, they are likely to suffer from mesh distortion and entanglement. When the material deformation becomes too large, mesh-based methods generally require remeshing and remapping of variables, which do not only introduce an error but are also computationally costly. Particle-based methods on the other side overcome this drawback by switching to either meshless or hybrid approach. SPH, MPS, and meshless methods in general discretize the problem domain into moving particles governed solely by the interaction between the neighboring particles. Even though remeshing processes can be avoided in this case, searching procedures and the usage of high-order basis functions are necessary to obtain accurate results [7]. Moreover, the difficulty to enforce Dirichlet boundary conditions (i.e., due to the absence of the Kronecker delta property) should be addressed properly, on top of the presence of tensile instability caused by an insufficient number of neighbor particles. Even though many attempts have been continuously proposed to overcome these issues [8], a valuable alternative could be represented by the hybrid approaches, combining the strengths of mesh-based and meshless techniques while simultaneously minimizing their drawbacks.

The material point method (MPM), belonging to the hybrid techniques, was originally proposed by

✉ Antonia Larese
    antonia.larese@unipd.it

    Bodhinanda Chandra
    bchandra@berkeley.edu

[1] Department of Civil, Geo and Environmental Engineering, Technical University of Munich, 80333 Munich, Germany

[2] Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720, USA

[3] Department of Mathematics "Tullio Levi Civita", Università degli Studi di Padova, 35121 Padua, Italy

Sulsky et al. [9] as an extension of the particle-in-cell (PIC) [10] and fluid-implicit-particle (FLIP) [11] method for treating solid dynamics problems. The key difference between MPM and its preceding methods is that MPM performs the integration of constitutive stress at the Lagrangian moving particles, while keeping the fixed "background" mesh in order to approximate continuous fields and their spatial gradients. MPM can be interpreted as a modified updated-Lagrangian finite element technique with moving integration points and a final mesh reset at each time step. MPM can simulate history-dependent materials with considerably large deformation avoiding mesh-distortion issue since the material properties are stored within the particles, called material points. The method since then has been successfully employed for the simulation of a variety of problems that are not easily addressed with mesh-based approaches. We can mention, for instance, hyper-velocity impact [12, 13], explosions [14], failure [15], and multi-phase geomechanical problems [16–18]. Some applications of the MPM are listed as a general review in the books published by Zhang et al. [19] and Fern et al. [20].

The use of MPM and its extended variants, such as the generalized interpolation material point (GIMP) method [21] or the convected particle domain interpolation (CPDI) [22, 23], to simulate large deformation solid dynamics problems, indeed, provides tremendous advantages, as it successfully tackles and solves the problem with mesh entanglement. Nevertheless, the method still suffers from some major difficulties. One open issue to be addressed is related to the nonconforming boundary conditions. In MPM, differently from other Lagrangian mesh-based approaches, boundary conditions often need to be applied on boundaries which does not match with the background mesh. Similar problems appears in the unfitted methods for computational fluid dynamics (CFD) problems [24–28].

This MPM problem has been addressed by few other authors. One possibility is to move the background fixed mesh according to the material displacements and deformation [29–31]. By doing this, the material boundary always coincides with the mesh edges in 2D or surface in 3D and boundary conditions can be applied in a FEM fashion to the mesh nodes. However, the accuracy of this method is strongly limited by the boundary shape and motion, i.e., the material boundary has to maintain its surface description and be subjected to small deformation movements without severe rotations. Otherwise, remeshing should be performed, losing the main motivation behind the choice of MPM instead of the standard FEM. Furthermore, the moving-mesh technique is also observed to produce numerical noises [32] due to the cell-crossing instabilities, which occur when material points cross the computational mesh boundaries [21]. In alternative the

desired boundary conditions can be applied to the boundary particles. This can be done either by identifying the outer layer of material points [33] or by building a layer of massless boundary particles which track the deformation of material boundaries. In both cases the carried information is mapped onto the connectivity's nodes through the element shape functions. The main disadvantage of this approach is that the applied boundary condition is not exactly lying on the material boundary, but across a boundary band, which therefore, results in an increase of the thickness of the actual boundary. This leads to significant errors and element size dependency as pointed out by Beuth [33] and Kafaji [29].

Another potential method was proposed by Mast et al. [34], assuming a dual-grid approach. Essentially, this method introduces an additional grid which follows the geometry of the material boundary, and the communication between the additional grids and the original background grids is performed to enforce the nonconforming essential boundary conditions. However, as explained by Bing et al. [35], the dual-grid configuration is problematic even for simple 1D problems, since it is sensitive to different boundary cell size and location. Mast's colleague, Yang [36], proposed a scheme for handling nonconforming Dirichlet boundaries by giving a specific boundary force field. This technique defines an influence region nearby the boundary, divided into three zones: constrained, decay, and free. Within the decay zone, the magnitude of boundary forces is reduced by a prescribed decay function to reflect a decreasing influence to the material. It should be noted here that the choice of assumed decay function will influence the quality of the numerical results.

An interesting alternative work to address moving Neumann boundary conditions has been proposed by Remmerswall et al. [37]. In their work, Remmerswall and colleagues introduced a boundary detection algorithm using the proximity field method (PFM), which is based on the concept derived from free surface boundary detection used in CFD. However, the PFM approach is reported to be computationally costly, highly mesh dependent, and its accuracy, at present, is still questionable [37].

Last but not least, most recently, Bing and colleagues [35, 38, 39] have proposed a more general way to track the evolution of physical boundary by utilizing a cubic B-spline interpolation method. By using this technique, the integration of surface traction for Neumann boundary can be done directly along the B-spline surface by incorporating Gauss quadrature. The B-spline surface can also be combined with a Dirichlet boundary imposition using the implicit boundary method (IBM) adopted from FEM [40–43]. The key idea of the IBM is to enforce essential boundary conditions by introducing extra stiffness terms in the system of equations, which can be calculated through a

"Dirichlet function" that depends on the signed-distance function measured from the material boundaries. Bing et al. [35, 38] extended it to MPM to include the roller boundary conditions on inclined boundaries and the treatment of nonhomogeneous boundaries. However, since the definition of the B-spline surface has to be established from the beginning of the simulation, this approach is still doubted to work for large deformation cases involving crack, fragmentation, and new surface reconstruction.

IBM has been proven to be suitable for imposing homogeneous and nonhomogeneous Dirichlet boundary conditions for 2D quasi-static linear-elastic problems in MPM with structured quadrilateral elements [35, 38, 39]. Yet, so far, implementation for 3D cases, with unstructured background mesh, and for nonlinear dynamic problems involving elastoplastic materials is not yet presented. Within their method, there are some limitations observed rooted from the undertaken assumptions. First, the IBM enforces the Dirichlet imposition by modification of the trial and test function spaces. Here, the approximated displacement field is modified by including the influence of diagonal Dirichlet matrix as well as the boundary condition to be imposed. This may lead to the violation of continuity requirements between elements, i.e., between cut and regular elements, which, by further, may potentially cause numerical instability. Secondly, the method also utilizes the material constitutive matrix **D** within its boundary enforcement formulation; this particularly may be difficult to represent when complex mixture of elastoplastic materials are involved.

While the application of Neumann boundary conditions has its paramount importance in many solid dynamics simulations, which have been discussed thoroughly in many prior works [35, 37, 44, 45], the current paper solely focuses in developing methods to handle the application of Dirichlet boundary conditions, which are needed in many problems of transient dynamics. The discussion on Neumann boundary conditions will be kept as a future work. This paper offers an alternative approach of implicit boundary imposition. Instead of modifying the interpolation shape functions or field parameterization as in the IBM [35, 38, 39], a modification of the weak form is adopted by appending an extra term derived from the penalty formulation.

The penalty formulation in general has been well established in FEM to impose constraint conditions of nonconforming elements [46], as well as in isogeometric B-Rep analysis (IBRA) [47, 48] to couple boundary conditions between patches. Compared to methods such as the Lagrange multiplier [49], where additional degrees of freedom (DOF) are required, the penalty method is significantly less expensive in computational cost. Furthermore, for similar imposition purposes, this approach has also been extended to meshfree methods [50–52] and other

continuum-based methods, such as in the CutFEM [53] and the finite cell methods (FCM) [54]. In the present paper, the adaption of the penalty method to MPM is presented. The penalty augmentation is adopted using boundary particles, which can be fixed or can move not only according to the material deformation, but also following prescribed field values, independently from the deformation.

The objective of this work is to introduce a penalty augmentation to impose nonhomogeneous, nonconforming boundary conditions in MPM. The proposed approach can work independent of background meshes (e.g., triangular, quadrilateral, tetrahedral, hexahedral), all type of materials, in both small and large strain regimes, and in both 2D and 3D problems. An extension to accommodate inclined roller support and frictionless contact (detaching) condition is also added to the formulation. The developed MPM code employed in the current work utilizes implicit integration scheme [55–57] and has been developed by the authors within the Kratos Multiphysics open-source platform [58]. The code proposed is available in GitHub[1] under a BSD license.

The paper is structured as follows. First, Sect. 2 introduces the governing equations needed, including a summary of procedure to perform an implicit MPM simulation. Section 3 provides some numerical details on how the boundary conditions are treated in implicit MPM, for both the grid-conforming and the proposed nonconforming boundary imposition. In Sect. 4, some numerical tests are simulated and discussed to assess the capabilities of the proposed approach. Finally, Sect. 5 presents the conclusions and suggestions for future works.

2 Displacement-based implicit MPM formulation

This section presents an overview of the displacement-based implicit MPM formulation used in the current work. The reader is strongly recommended to refer to [55, 56, 59] for more detailed information on the algorithm.

2.1 Balance equations and weak formulation

Consider a body $\mathcal{B}$, which occupies an initial domain Ω of the three-dimensional Euclidean space $\mathcal{E}$, with a regular boundary $\partial\Omega$ in its reference configuration. A deformation of $\mathcal{B}$ at a given time t is defined by a one-to-one mapping φ,

$$\varphi : \Omega \to \mathcal{E}. \tag{1}$$

The deformation gradient $\mathbf{F}$ is defined as

$$\mathbf{F}(\mathbf{X}, t) = \nabla_{\mathbf{X}} \varphi(\mathbf{X}, t) = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \tag{2}$$

where $\mathbf{x}$ and $\mathbf{X}$ denote the current and the reference configurations, respectively.

Continuity equation reads

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} + \rho \nabla \cdot \mathbf{v} = 0 \quad \text{in} \quad \varphi(\Omega), \tag{3}$$

where ρ and $\mathbf{v}$ are the density and velocity vector and $\rho/\mathrm{d}t$ indicates the total derivative of ρ. The conservation of linear momentum of $\mathcal{B}$ is given by:

$$\rho \ddot{\mathbf{u}} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad \text{in} \quad \varphi(\Omega), \tag{4}$$

where $\ddot{\mathbf{u}}$ represents the acceleration vector, $\boldsymbol{\sigma}$ is the symmetric Cauchy stress tensor and $\mathbf{b}$ denotes the volume acceleration.

The governing Eqs. (3) and (4) have to be solved numerically in a three-dimensional field $\Omega \subseteq \mathbb{R}^3$, within the time range $t \in [0, T]$, considering the following boundary conditions:

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

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on} \quad \varphi(\partial\Omega_N), \tag{6}$$

where $\mathbf{n}$ is the outward unit normal vector, while $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}$ are the Dirichlet prescribed displacement and the Neumann traction on $\partial\Omega_D$ and $\partial\Omega_N$, respectively.

Assuming the following constitutive stress–strain relation:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}) = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})), \tag{7}$$

where $\boldsymbol{\varepsilon}$ is the strain tensor, and calling $\mathcal{V} \subset \mathbb{R}^3$ the space of the solution $\mathbf{u}$, we can derive the corresponding weak formulation. The problem is, hence, reduced to find a kinematically admissible displacement field $\mathbf{u} \in \mathcal{V}$ at each time t which satisfies:

$$\begin{aligned} R(\mathbf{u}, \mathbf{w}) = &\int_{\varphi(\Omega)} \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) : \nabla^s \mathbf{w} \mathrm{d}v \\ &- \int_{\varphi(\Omega)} \rho(\mathbf{b} - \ddot{\mathbf{u}}) \cdot \mathbf{w} \mathrm{d}v \\ &- \int_{\varphi(\partial\Omega_N)} \bar{\mathbf{t}} \cdot \mathbf{w} \mathrm{d}a = 0 \quad \forall \mathbf{w} \in \mathcal{V}, \end{aligned} \tag{8}$$

where $\mathbf{w} \in \mathcal{V}$ is an arbitrary test function.

2.2 Linearization of the weak formulation

Equation (8) is valid for any kind of strain definitions, including the large strain definitions. Since, in the current work, we attempt to solve problems involving strong material and geometric nonlinearities, a linearization of the weak form is necessary to solve the associated nonlinear

boundary value problems (BVP). By utilizing the Newton–Raphson's method, the approximation of the solution can be obtained iteratively by means of directional derivatives.

The linearization of the residual weak form following a Taylor's expansion yields to the following linear system matrix to be solved at every time step t:

$$\mathbf{K}^{\text{tan}} \delta \mathbf{u} = -\mathbf{R}. \tag{9}$$

where $\mathbf{R}$ is the residual vector expressed by Eq. (8), whereas $\mathbf{K}^{\text{tan}}$ is the corresponding tangent matrix associated to the directional derivative of $\mathbf{R}$. The displacement $\mathbf{u}$ is continuously updated during the Newton–Raphson's iterations by adding the previously unknown increment $\delta \mathbf{u}$ to its previous converged values.

The tangent stiffness matrix can be split into several sub-matrices given by:

$$\mathbf{K}^{\text{tan}} = \mathbf{K}^{\text{geo}} + \mathbf{K}^{\text{mat}} + \mathbf{K}^{\text{dyn}}. \tag{10}$$

The first two sub-matrices represent the static part of $\mathbf{K}$ and are known as the geometric and material stiffness matrices, respectively:

$$\mathbf{K}^{\text{geo}} := \int_{\varphi(\Omega)} \nabla_x \delta \mathbf{u} \cdot \boldsymbol{\sigma} \cdot \nabla_x \mathbf{w} \mathrm{d}v, \tag{11}$$

$$\mathbf{K}^{\text{mat}} := \int_{\varphi(\Omega)} \nabla_x^s \mathbf{w} \cdot \mathbf{D} \cdot \nabla_x^s \delta \mathbf{u} \mathrm{d}v. \tag{12}$$

Here, $\mathbf{D}$ is the tangent stiffness matrix defined for the used constitutive equation and ∇_x denotes the spatial gradient at the current configuration, $\nabla_x = \partial/\partial\mathbf{x}$.

Moreover, the dynamic component of the stiffness matrix can be computed as:

$$\mathbf{K}^{\text{dyn}} = \int_{\varphi(\Omega)} (\rho \, \alpha \, \mathbf{w} \cdot \delta \mathbf{u}) \mathrm{d}v, \tag{13}$$

where the value of α depends on the time integration scheme used. For instance, $\alpha = 1/\beta_N \Delta t^2$ in Newmark integration scheme, where β_N is typically equal to 0.25.

2.3 Spatial discretization

The integrals written in Eqs. (9)–(13) need to be approximated spatially by discretization techniques. In MPM, as the continuous geometry and domain are approximated by finite number of material points and background grids, the field quantities presented in the weak forms, e.g., stresses, displacements, etc., can be approximated through extrapolation and interpolation methods. In the following formulations, the variables at the mesh nodes will be identified with subscript I and J, while the properties associated with material points will be identified with subscript p. Moreover, the superscript n and $n + 1$ indicate the current (known) and next (unknown) time step,

respectively. All the variables with subscript h are nothing but the approximation of the variables themselves in the discretized geometry.

Following the standard nonlinear finite element discretization [60], Eq. (9) is discretized as:

$$\mathbf{K}^{\tan}\Delta\mathbf{u} = -\mathbf{R}, \tag{14}$$

where $\Delta\mathbf{u}$ is the increment of the discrete solution of the problem (i.e., the discrete counterpart of the incremental displacement $\delta\mathbf{u}$ used in Eq. 9). The elemental tangent stiffness matrix $\mathbf{K}^{\tan}$ and residual vector $\mathbf{R}$ are expressed as,

$$\mathbf{R}_I = \bigcup_{p=1}^{n_p}\left(\mathbf{B}_I\boldsymbol{\sigma} - \rho\mathbf{b}N_I + \sum_{J=1}^{n_n} N_I\rho N_J\mathbf{a}_J\right)v_p \\ - \bigcup_{a=1}^{n_a} N_I\bar{\mathbf{t}}A_a, \tag{15}$$

$$\mathbf{K}_{IJ}^{\tan} = \bigcup_{p=1}^{n_p}\left((\nabla_x N_I)^T\boldsymbol{\sigma}(\nabla_x N_J) + \mathbf{B}_I^T\mathbf{D}\mathbf{B}_J \\ + N_I\rho\alpha N_J\mathbf{I}\right)v_p. \tag{16}$$

where v_p and A_a are the current volume of a single material point and the area weight of surface boundary, respectively. $\mathbf{D}$ is the constitutive matrix and $\mathbf{B}_I$ is the deformation matrix related to each node I in the element. N_I is corresponding shape function value associated with grid node I, whereas n_n is the total number of computational node. Operator $\bigcup$ is used in Eqs. (15) and (16), instead of $\sum$ over all n_p, as the assembly procedure is not only adding up each of the material point contributions but also addressing the kinematic compatibility between the points.

2.4 Implicit MPM

The material point method (MPM) was originally proposed by [9] as an extension of the particle-in-cell (PIC) and fluid-implicit-particle (FLIP) method for treating solid dynamics problems. The method has been developed over the last two decades especially for solids undergoing large deformation [61], including many other multiphase, multi-physics, and multi-scale problems. MPM uses a fixed background mesh and a set of moving material points. While the mesh is used for calculation purpose in a FEM fashion and is reset at the end of each time step, all historical information is stored inside the material points which are moving in a Lagrangian way and employed as moving integration nodes (Fig. 1).

While MPM algorithms are generally written using an explicit time integration [62, 63], recently, implicit formulations [64–66] are often preferred as they are more accurate and robust to simulate cases with small rate of deformation, or static and quasi-static problems. On top of that, the stability of the method (for properly chosen dissipative approaches) does not depend on the wave propagation speed within the media and, thus, allows the usage of a relatively larger time steps. Finally, implicit formulations are advantageous to solve multi-physics coupling problems, especially if MPM should be coupled with other implicit-based methods such as FEM [67]. A more recent extension of the implicit MPM to GIMP has been proposed and showed to work well for large deformation elasto-plastic materials by [68, 69]. The three main phases of the classical implicit MPM are graphically summarized in Fig. 1.

(a) *Initialization phase*

- The material point-mesh connectivity is first defined by searching the element where each material point is located;
- The kinematic information obtained at the previous time step n is mapped by means of mass projection from each material point p to the connectivity nodes as initial conditions.

$$m_I^n = \sum_{p=1}^{n_p} m_p N_I(\boldsymbol{\xi}_p^n), \tag{17}$$

$$\mathbf{q}_I^n = \sum_{p=1}^{n_p} m_p\mathbf{v}_p^n N_I(\boldsymbol{\xi}_p^n), \tag{18}$$

$$\mathbf{f}_I^n = \sum_{p=1}^{n_p} m_p\mathbf{a}_p^n N_I(\boldsymbol{\xi}_p^n), \tag{19}$$

where n_p and $\boldsymbol{\xi}_p$ are the total number of material points inside one element and local coordinate of the material point p inside the connectivity. Here, $\mathbf{f}$, $\mathbf{q}$, $\mathbf{v}$, and $\mathbf{a}$ denote the force, momentum, velocity, and acceleration vectors, respectively.

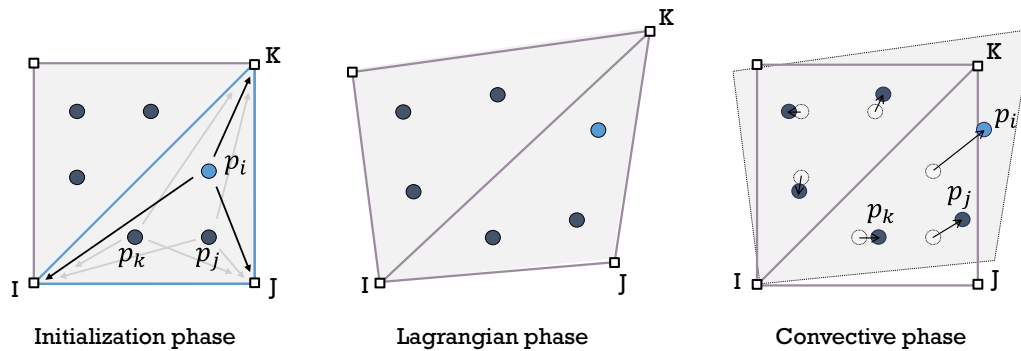- The corresponding initial nodal velocity and acceleration can be computed as:

$$\tilde{\mathbf{v}}_I^n = \frac{\mathbf{q}_I^n}{m_I^n}, \tag{20}$$

$$\tilde{\mathbf{a}}_I^n = \frac{\mathbf{f}_I^n}{m_I^n}. \tag{21}$$

(b) *Lagrangian phase*

- The discretized form of the governing equations (Eq. (14)) is solved on the active background mesh—those which contain material points—according to the scheme used.

(c) *Convective phase*

Fig. 1 The three phases of MPM: **a** initialization phase, **b** Lagrangian phase, and **c** convective phase. Square markers and capital letters identify the mesh nodes, while round markers and small letters indicate the material points

- The obtained solutions are interpolated back to the material points following:

$$\mathbf{u}_p^{n+1} = \sum_{I=1}^{n_n} N_I(\boldsymbol{\xi}_p^n)\mathbf{u}_I^{n+1}, \tag{22}$$

$$\mathbf{a}_p^{n+1} = \sum_{I=1}^{n_n} N_I(\boldsymbol{\xi}_p^n)\mathbf{a}_I^{n+1}, \tag{23}$$

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{1}{2}\Delta t\left(\mathbf{a}_p^n + \mathbf{a}_p^{n+1}\right), \tag{24}$$

where n_n is the total number of nodes per geometrical element.

- The position of the material points can be updated as:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{u}_p^{n+1}. \tag{25}$$

- The background mesh is recovered to their initial undeformed configuration, and all the nodal variables are cleared.

The implicit approach implemented in this paper is an extension of the displacement-based formulation, which was originally proposed by Guilkey and Weiss [64]. Further details of the used algorithm, such as the prediction–correction scheme derived from Newmark integration scheme and its extension to incorporate mixed formulation, can be found in [55, 56, 59].

3 Treatment of Dirichlet boundary conditions

The treatment of boundary conditions in MPM, for both the Neumann and Dirichlet boundaries, is not always straightforward, since the material surface boundary $\Gamma = \partial\Omega$, most of the time, does not coincide with the mesh boundary. As discussed in Sect. 1, a number of different strategies have been adopted to overcome this problem in

the past. From hereon, the boundary coinciding with the background mesh will be called *grid-conforming* boundary, while the *nonconforming* boundary will describe the boundary which does not coincide with the background element boundary (i.e., its nodes), see Fig. 2 for illustrative description. The current section will present the general formulation to apply grid-conforming boundary conditions in implicit MPM, followed by the proposed nonconforming boundary enforcement by means of the penalty method.

3.1 Grid-conforming boundary conditions

As commonly done in FEM, degrees of freedom (dofs) can be ordered such that the unknown dofs come first and the Dirichlet nodes afterward. The algebraic solution system now reads:

$$\begin{bmatrix} \mathbf{K}_{\mathbf{uu}}^{\tan} & \mathbf{K}_{\mathbf{uu}_{\Gamma_D}}^{\tan} \\ \mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}}^{\tan} & \mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}_{\Gamma_D}}^{\tan} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{u}_{\Gamma_D} \end{bmatrix} = -\begin{bmatrix} \mathbf{R}_{\mathbf{u}} \\ \mathbf{R}_{\mathbf{u}_{\Gamma_D}} \end{bmatrix}, \tag{26}$$

where $\Delta\mathbf{u}$ is the increment of displacement evaluated at each time step such that, $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta\mathbf{u}$. The sub-matrices $\mathbf{K}_{\mathbf{uu}}^{\tan}$ and $\mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}_{\Gamma_D}}^{\tan}$ are the stiffness matrix components which solely correspond to the regular displacement to solve $\mathbf{u}$ and the known displacement $\mathbf{u}_{\Gamma_D}$, whereas $\mathbf{K}_{\mathbf{uu}_{\Gamma_D}}^{\tan}$ and $\mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}}^{\tan}$ are the off-diagonal components connecting the solved and prescribed displacements. In general, the tangent stiffness matrix is symmetric, and thus, $\mathbf{K}_{\mathbf{uu}_{\Gamma_D}}^{\tan} = \left(\mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}}^{\tan}\right)^T$. In addition to that, the residual right-hand-side vector $\mathbf{R}$ can also be split into two components related to $\mathbf{u}$ and $\mathbf{u}_{\Gamma_D}$, i.e., $\mathbf{R}_{\mathbf{u}}$ and $\mathbf{R}_{\mathbf{u}_{\Gamma_D}}$, respectively.

Hence, the goal is to obtain $\Delta\mathbf{u}_{\Gamma_D} = 0$, as if the following predictor–corrector scheme was used:

$$(\textbf{predictor on } \partial\Omega_D): \quad \mathbf{u}_{\Gamma_D}^{n+1,0} = \bar{\mathbf{u}}, \tag{27}$$

$$(\textbf{corrector on } \partial\Omega_D): \quad \mathbf{u}_{\Gamma_D}^{n+1,k} = \mathbf{u}_I^{n+1,k-1} + \Delta\mathbf{u}_{\Gamma_D}^k \\ = \bar{\mathbf{u}}, \tag{28}$$
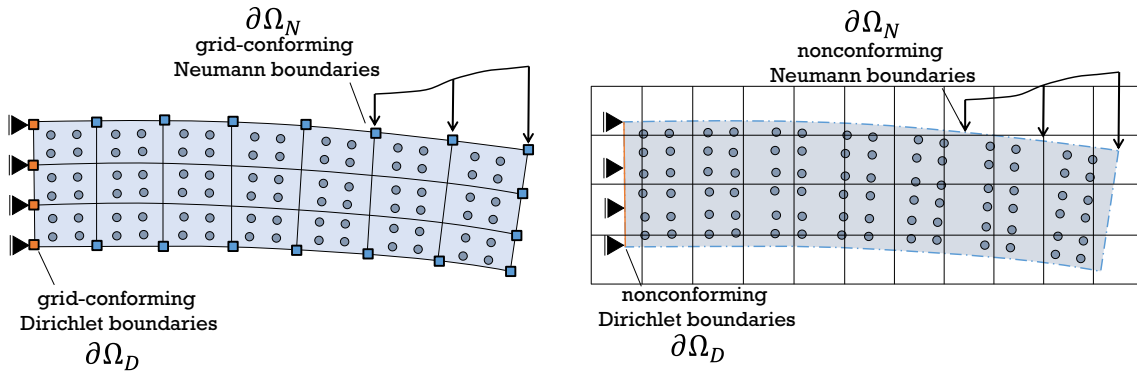
Fig. 2 Types of boundary conditions in MPM: grid-conforming (left) and nonconforming (right) boundary conditions

the Dirichlet boundary Eq. (5) is enforced automatically. The superscript k is used to denote iteration of the Newton–Raphson's method. This can be accomplished by setting the off-diagonal elements of the tangent stiffness matrix, $\mathbf{K}^{\text{tan}}_{\mathbf{uu}_{\Gamma_D}}$ and $\mathbf{K}^{\text{tan}}_{\mathbf{u}_{\Gamma_D}\mathbf{u}}$, to zero, and assigning the diagonal components as identity, $\mathbf{K}^{\text{tan}}_{\mathbf{u}_{\Gamma_D}\mathbf{u}_{\Gamma_D}} = \mathbf{I}$. Meanwhile, the right-hand-side components corresponding to the Dirichlet conditions are also modified as $\mathbf{R}_{\mathbf{u}_{\Gamma_D}} = \mathbf{0}$. Equation (26) can, therefore, be rewritten as:

$$\begin{bmatrix} \mathbf{K}^{\text{tan}}_{\mathbf{uu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{u}_{\Gamma_D} \end{bmatrix} = -\begin{bmatrix} \mathbf{R}_{\mathbf{u}} \\ \mathbf{0} \end{bmatrix}. \tag{29}$$

Solving the modified system yields the solution set $[\mathbf{u}^{n+1}, \bar{\mathbf{u}}]$, and the Dirichlet boundary conditions are automatically enforced.

Equation (29) works for both homogeneous and non-homogeneous boundaries, as well as for roller supports if the boundary conditions are described in the direction of Cartesian coordinate bases, i.e., $\mathbf{e}_i$. However, this will not work for roller boundaries with an arbitrary inclination (Fig. 3). To accommodate the roller boundary conditions in an inclined configuration, a rotation matrix $\mathbf{Q}$ is introduced:

$$\mathbf{Q} = \begin{bmatrix} \hat{n}_x & \hat{n}_y & \hat{n}_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \hat{q}_x & \hat{q}_y & \hat{q}_z \end{bmatrix}, \tag{30}$$

where $\hat{\mathbf{n}}$, $\hat{\mathbf{t}}$, and $\hat{\mathbf{q}}$ are the unit normal vector perpendicular to the inclined surface and the two unit tangent vectors parallel to the surface, respectively. Note that, $\mathbf{Q}$ is orthogonal, and therefore, $\mathbf{Q}^T = \mathbf{Q}^{-1}$. The mapping between (n, t, q) and (x, y, z) can be performed as:

$$\mathbf{u}^\theta = \mathbf{Q}\mathbf{u} = [u_n \ u_t \ u_q]^T, \tag{31}$$

where $\mathbf{u}^\theta$ is the displacement described in the axis-aligned configuration.
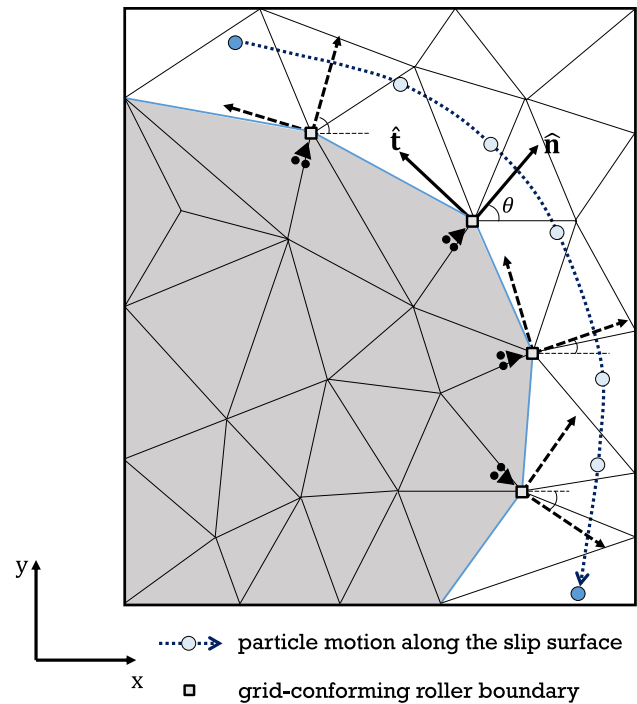

Fig. 3 Grid-conforming slip or roller support with arbitrary inclination

We then use a similar approach to compute a partially axis-aligned system by multiplying Eq. (26) with a modified rotation matrix $\widehat{\mathbf{Q}}$ as:

$$\begin{aligned} \widehat{\mathbf{Q}}\mathbf{K}^{\text{tan}}\left(\widehat{\mathbf{Q}}^T\widehat{\mathbf{Q}}\right)\Delta\mathbf{u} &= -\widehat{\mathbf{Q}}\mathbf{R} \\ \left(\widehat{\mathbf{Q}}\mathbf{K}^{\text{tan}}\widehat{\mathbf{Q}}^T\right)\widehat{\mathbf{Q}}\Delta\mathbf{u} &= -\widehat{\mathbf{Q}}\mathbf{R} \\ \mathbf{K}^\theta \Delta\mathbf{u}^\theta &= -\mathbf{R}^\theta. \end{aligned} \tag{32}$$

$\widehat{\mathbf{Q}}$ is a block matrix with the following structure:

$$\widehat{\mathbf{Q}} = \begin{bmatrix} [\mathbf{I}] & [\mathbf{0}] \\ [\mathbf{0}] & [\mathbf{Q}] \end{bmatrix}, \tag{33}$$

where the unsupported system is kept as it is, while the

Dirichlet boundary is rotated by multiplying it with the rotation matrix $\mathbf{Q}$. Equation (32) can be expanded as:

$$
\left[
\begin{array}{c|c}
\mathbf{K}_{\mathbf{uu}}^{\tan} & \mathbf{K}_{\mathbf{uu}_{\Gamma_D}}^{\tan} \mathbf{Q}^T \\
\hline
\mathbf{Q}\mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}}^{\tan} & \mathbf{Q}\mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}_{\Gamma_D}}^{\tan} \mathbf{Q}^T
\end{array}
\right]
\begin{bmatrix}
\Delta\mathbf{u} \\
\Delta u_{\Gamma_D,n} \\
\Delta u_{\Gamma_D,t} \\
\Delta u_{\Gamma_D,q}
\end{bmatrix}
= -
\begin{bmatrix}
\mathbf{R}_{\mathbf{u}} \\
R_{\mathbf{u}_{\Gamma_D,n}} \\
R_{\mathbf{u}_{\Gamma_D,t}} \\
R_{\mathbf{u}_{\Gamma_D,q}}
\end{bmatrix}.
\tag{34}
$$

As the partially rotated system has been obtained, we can then enforce support only in the normal direction, similar to the one in Eq. (29). This yields the following system:

$$
\left[
\begin{array}{c|c|c}
\mathbf{K}_{\mathbf{uu}}^{\tan} & 0 & \left(\mathbf{K}_{\mathbf{uu}_{\Gamma_D}}^{\tan}\mathbf{Q}^T\right)_{t,q} \\
 & \vdots & \\
\hline
\mathbf{0}\cdots & 1 & 0 \\
\left(\mathbf{Q}\mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}}^{\tan}\right)_{t,q} & 0 & \left(\mathbf{Q}\mathbf{K}_{\mathbf{u}_{\Gamma_D}\mathbf{u}_{\Gamma_D}}^{\tan}\mathbf{Q}^T\right)_{t,q}
\end{array}
\right]
\begin{bmatrix}
\Delta\mathbf{u} \\
\Delta u_{\Gamma_D,n} \\
\Delta u_{\Gamma_D,t} \\
\Delta u_{\Gamma_D,q}
\end{bmatrix}
= -
\begin{bmatrix}
\mathbf{R}_{\mathbf{u}} \\
0 \\
R_{\mathbf{u}_{\Gamma_D,t}} \\
R_{\mathbf{u}_{\Gamma_D,q}}
\end{bmatrix}.
\tag{35}
$$

Solving Eq. (35) will give:

$$
\Delta\mathbf{u}_{\Gamma_D}^{\theta} = [0 \ \Delta u_{\Gamma_D,t} \ \Delta u_{\Gamma_D,q}]^T,
\tag{36}
$$

which has to be rotated back to the original configuration before being added to the solution step. This modifies the predictor–corrector scheme specified by Eqs. (27) and (28) to:

$$
(\textbf{predictor on } \partial\Omega_D): \quad \mathbf{u}_{\Gamma_D}^{n+1,0} = \bar{u}_n\hat{\mathbf{n}},
\tag{37}
$$

$$
(\textbf{corrector on } \partial\Omega_D): \quad \mathbf{u}_{\Gamma_D}^{n+1,k} = \mathbf{u}_I^{n+1,k-1} + \mathbf{Q}^T\Delta\mathbf{u}_{\Gamma_D}^{\theta},
\tag{38}
$$

which results in a slip boundary condition that supports the normal direction but allows the tangential directions to move freely for arbitrary inclination.

3.2 Nonconforming boundary enforcement

The imposition of nonconforming Dirichlet boundary conditions in MPM should be as general and simple as possible, so that it can be used in practice, without the necessity of modifying the underlying parameterization. This is the main motivation behind the choice of implicit boundary imposition by means of penalty method proposed in the current work. The penalty method has been previously used just for contact algorithm in explicit [70, 71] and implicit [72] MPM. The method works seamlessly within the nonlinear implicit MPM framework established earlier for Dirichlet boundary enforcement. Furthermore, this work proposes and extends the usage of the method to include slip and releasing contact boundaries, which formulation will be elaborated in the current section and summarized in Algorithm 1 provided in Appendix A. While the following section will present an applicability of

penalty method to implicit MPM, the technique has never been used in explicit MPM. However, it was used to couple trimmed B-Rep patches through weak enforcement in explicit IBRA by [73]. The formulation of penalty method in explicit MPM should not differ substantially from the current work, but the usage of smaller time step might be necessary to provide stability of the boundary imposition. The extension of the proposed implementation towards implicit GIMP [68, 69] or other variants of implicit MPM can be done with minimum modifications, although more comprehensive assessment of the accuracy and the feasibility of the penalty augmentation should be investigated further and is left for future works.

3.2.1 Penalty approach

The basic concept of the penalty method is introduced in this section, which can be used to impose Dirichlet boundary condition in two nonconforming discretizations. First, let's assume two coupling boundary edges Γ_1 and Γ_2, which can be discretized by using any methods, e.g., as surface meshes, B-spline surfaces, or even as particles. Each of the boundary edge contains a corresponding displacement field $\mathbf{u}_1 = \mathbf{u}_1(\mathbf{x}_1)$ and $\mathbf{u}_2 = \mathbf{u}_2(\mathbf{x}_2)$, respectively. The penalty approach formulation to couple the displacement between the two surfaces starts with the following virtual work formulation:

$$
\begin{aligned}
\delta W^{\text{penalty}} &= \beta \int_{\Gamma_1} (\mathbf{u}_1 - \mathbf{u}_2) \cdot (\delta\mathbf{u}_1 - \delta\mathbf{u}_2)\mathrm{d}\Gamma_1, \\
&= \beta \int_{\Gamma_2} (\mathbf{u}_2 - \mathbf{u}_1) \cdot (\delta\mathbf{u}_2 - \delta\mathbf{u}_1)\mathrm{d}\Gamma_2,
\end{aligned}
\tag{39}
$$

where β is the user-defined penalty factor. Here, $\delta W^{\text{penalty}}$ corresponds to the coupling between the displacement (i.e., the position) of two coupling surfaces, written in a weak sense. Notice that the subscripts 1 and 2 are interchangeable, which should result in the same virtual work formulation independently whether the integral is computed along Γ_1 or along Γ_2. If the discretization of the surface Γ_1 is matching and consistent with the discretization of Γ_2, which means that the displacement is conforming $\mathbf{u}_1 = \mathbf{u}_2$, Eq. (39) vanishes and the coupling is inherently satisfied.

Dirichlet boundary imposition actually can be seen as a special type of surface coupling, with a prescribed displacement $\bar{\mathbf{u}}$, instead of the displacement fields $\mathbf{u}_2$ in Γ_2. We can then rewrite Eq. (39), assuming $\mathbf{u} = \mathbf{u}_1$, as:

$$
\delta W^{\text{penalty}} = \beta \int_{\Gamma_D} (\mathbf{u} - \bar{\mathbf{u}}) \cdot \delta\mathbf{u}_D \ \mathrm{d}\Gamma_D,
\tag{40}
$$

where $\Gamma_D = \Gamma_1$ indicates the boundaries to be enforced.

Appending Eqs. (40) to (8), the modified system of equations can be written as:

$$\left(\mathbf{K}^{\text{tan}} + \mathbf{K}^{\text{penalty}}\right)\Delta\mathbf{u} = -\left(\mathbf{R} + \mathbf{R}^{\text{penalty}}\right), \quad (41)$$

where $\mathbf{K}^{\text{tan}}$ and $\mathbf{R}$ are the regular local tangent stiffness matrix and residual vector without the imposition (Eq. (10)). The additional penalty terms in the system matrix are given as:

$$\mathbf{K}^{\text{penalty}} = \beta \int_{\Gamma_D} \mathbf{H}^T \cdot \mathbf{H} \, d\Gamma_D, \quad (42)$$

while the additional residual term is:

$$\begin{aligned}\mathbf{R}^{\text{penalty}} &= \beta \int_{\Gamma_D} \mathbf{H}^T \cdot (\mathbf{u} - \bar{\mathbf{u}}) \, d\Gamma_D \\ &= \left[\beta \int_{\Gamma_D} \mathbf{H}^T \cdot \mathbf{H} \, d\Gamma_D\right](\mathbf{u}_I - \bar{\mathbf{u}}_I) \\ &= \mathbf{K}^{\text{penalty}}(\mathbf{u}_I - \bar{\mathbf{u}}_I)\end{aligned} \quad (43)$$

with matrix $\mathbf{H}$ being defined as follows:

$$\mathbf{H} = \begin{bmatrix} N_1 & 0 & 0 & \cdots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \cdots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \cdots & 0 & 0 & N_n \end{bmatrix}. \quad (44)$$

Here, the subscript n denotes the number of nodes for each element.

3.2.2 Extension to inclined roller boundary

An extension to the regular penalty imposition is to include the inclined roller (or slip) boundary condition, similar to what explained in Sect. 3.1 for grid-conforming boundary conditions (Eqs. (32)–(38)). However, instead of modifying the matrix and vector elements in the normal direction, the modification will interest only the tangential components.

Initially, the new linear system of equation including the penalty term should be rotated to the partially axis-aligned configuration as in Eq. (32).

Therefore, applying the modified block matrix to Eq. (41) yields:

$$\left(\widehat{\mathbf{Q}}\left[\mathbf{K}^{\text{tan}} + \mathbf{K}^{\text{penalty}}\right]\widehat{\mathbf{Q}}^T\right)\widehat{\mathbf{Q}}\Delta\mathbf{u} = -\widehat{\mathbf{Q}}\left(\mathbf{R} + \mathbf{R}^{\text{penalty}}\right) \quad (45)$$

which can be rewritten as:

$$\left(\widehat{\mathbf{Q}}\mathbf{K}^{\text{tan}}\widehat{\mathbf{Q}}^T + \widehat{\mathbf{Q}}\mathbf{K}^{\text{penalty}}\widehat{\mathbf{Q}}^T\right)\widehat{\mathbf{Q}}\Delta\mathbf{u} = -\left(\widehat{\mathbf{Q}}\mathbf{R} + \widehat{\mathbf{Q}}\mathbf{R}^{\text{penalty}}\right). \quad (46)$$

Calling

$$\begin{aligned}\mathbf{K}^{\theta} &:= \widehat{\mathbf{Q}}\mathbf{K}^{\text{tan}}\widehat{\mathbf{Q}}^T, \\ \mathbf{K}^{\text{penalty}, \theta} &:= \widehat{\mathbf{Q}}\mathbf{K}^{\text{penalty}}\widehat{\mathbf{Q}}^T, \\ \mathbf{R}^{\theta} &:= \widehat{\mathbf{Q}}\mathbf{R}, \\ \mathbf{R}^{\text{penalty}, \theta} &:= \widehat{\mathbf{Q}}\mathbf{R}^{\text{penalty}},\end{aligned}$$

Equation (46) becomes:

$$\left(\mathbf{K}^{\theta} + \mathbf{K}^{\text{penalty}, \theta}\right)\Delta\mathbf{u}^{\theta} = -\left(\mathbf{R}^{\theta} + \mathbf{R}^{\text{penalty}, \theta}\right) \quad (47)$$

Since Dirichlet constraint only interests $\mathbf{R}^{\text{penalty}, \theta}$ and $\mathbf{K}^{\text{penalty}, \theta}$, the other terms should remain untouched. The main idea of the following steps is to enforce the normal displacement by penalty terms, whereas the tangential directions are kept as it is (free to deform and not restricted by the additional penalty terms). The modified penalty stiffness and residual vector can be expanded as:

$$\mathbf{K}^{\text{penalty}, \theta}_{\text{modified}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \left(\mathbf{Q}\mathbf{K}^{\text{penalty}}_{\mathbf{u}_{\Gamma_D}\mathbf{u}_{\Gamma_D}}\mathbf{Q}^T\right)_n & \vdots & \vdots \\ \mathbf{0} & \cdots & 0 & 0 \\ \mathbf{0} & \cdots & 0 & 0 \end{bmatrix}, \quad (48)$$

$$\mathbf{R}^{\text{penalty}, \theta}_{\text{modified}} = \begin{bmatrix} \mathbf{0} \\ \hline R^{\text{penalty}}_{\mathbf{u}_{\Gamma_D}, n} \\ 0 \\ 0 \end{bmatrix}. \quad (49)$$

Finally, we can obtain the imposed rotated system with slip support as:

$$\left(\mathbf{K}^{\theta} + \mathbf{K}^{\text{penalty}, \theta}_{\text{modified}}\right)\Delta\mathbf{u}^{\theta} = -\left(\mathbf{R}^{\theta} + \mathbf{R}^{\text{penalty}, \theta}_{\text{modified}}\right). \quad (50)$$

Solving this modified system of equation will yield the Dirichlet imposition that is working for nonconforming boundaries with arbitrary inclination.

3.2.3 Imposing penalty conditions in MPM through boundary particles

Notice that the surface integrals in Eqs. (42) and (43) can be approximated through various approaches, which depend on how the surface topology is represented. In this work, a particle discretization technique is used to represent the continuous surface Γ_D with a discrete number of boundary particles b. Hence, if this is the case, the continuous surface integral can be approximated as:

$$\int_{\Gamma_D} (\cdots) \mathrm{d}\Gamma_D \approx \int_{\Gamma_{D_h}} (\cdots) \mathrm{d}\Gamma_{D_h}$$
$$= \bigcup_{b=1}^{n_b} \int_{\Gamma_b} (\cdots) \mathrm{d}\Gamma_b = \bigcup_{b=1}^{n_b} (\cdots) A_b, \qquad (51)$$

where the last term can be written assuming that all the quantities inside the surface integral are independent of the surface.

The surface particle discretization is performed at the beginning of the simulation to generate boundary particles from an initial surface mesh, where the penalty conditions are imposed to the MPM. Here, the surface unit normal vector is also initially defined at each particle, which is needed when a contact boundary is considered (please refer to Sect. 3.2.4 for further discussion on the topic). The boundary particles are mass-less and only carry some necessary kinematic variables, as well as the boundary variables for imposition. Each of the generated boundary particle is assigned with a corresponding area which is needed to approximate surface integrals as in (51). The surface area of the particles can be initiated by a simple division operation in 2D and 3D cases as:

$$A_b = \frac{A_{\mathrm{mesh}}^{3D}}{\bar{n}_b} = \frac{L_{\mathrm{mesh}}^{2D}}{\bar{n}_b} \cdot t_{\mathrm{mesh}}, \qquad (52)$$

where A_b indicates the area of particle condition (or boundary) b. Meanwhile, A_{mesh}, L_{mesh}, t_{mesh}, and $\bar{n}_b$ are the area, length, thickness, and the number of boundary particle per initial mesh, which is an input parameter. An illustrative example of the associated boundary area in 3D assigned in the particle boundary is shown in Fig. 4. Boundary particles can be initiated similarly in 2D from line segments with length L_{mesh} and unit thickness $t_{\mathrm{mesh}} = 1$. It is worth to note that, according to our experience, the number of boundary particles per line segment in 2D or a surface element in 3D should be more than, or at least equal to, the considered number of material point per cell.

Numerical instability could appear when the nonconforming boundary intersects the background mesh very close to one of its edges. This instability is commonly known as the "small-cut" instability, very common also in FEM-based unfitted approaches [28, 74]. To overcome this issue a simple modification to the shape function values is performed while constructing matrix $\mathbf{H}$ in Eq. (44). The modified shape function value $\bar{N}_{Ib}$ for background node I at boundary particle b can be obtained as:

$$\bar{N}_{Ib} = \frac{N_{Ib}^*}{\sum_I^{n_n} N_{Ib}^*} \quad \text{where,} \quad N_{Ib}^* = \begin{cases} \epsilon, & \text{if} \quad N_{Ib} \leq \epsilon \\ N_{Ib}, & \text{otherwise} \end{cases}, \qquad (53)$$

where ϵ is the stabilization tolerance, which is often set to be $\epsilon = 0.01$, and $N_{Ib} = N_I(\xi_b)$ is the original shape

function value. Notice that, to ensure the partition of unity, the weighting procedure is necessary.

Another aspect which is worth to be noticed, especially in the presence of inclined roller support, is the definition of the unit normal vector, i.e., of the outward normal direction of the imposed boundary. First, the normal vector needs to be defined on each boundary particle b, $\hat{\mathbf{n}}_b$ when creating the boundary particles. Hence, the value of the $\hat{\mathbf{n}}_b$ is mapped (Fig. 5) at the beginning of each time step n to the background grid nodes I as:

$$\hat{\mathbf{n}}_I^n = \frac{\sum_{b=1}^{n_b} \hat{\mathbf{n}}_b^n A_b^n N_I(\xi_b^n)}{\left\| \sum_{b=1}^{n_b} \hat{\mathbf{n}}_b^n A_b^n N_I(\xi_b^n) \right\|}. \qquad (54)$$

The L_2 normalized unit normal vector of the background nodes, $\hat{\mathbf{n}}_I$, is required here to compute the tangential directions and to construct the rotation matrix $\mathbf{Q}_I$ (Eq. (30)). With this matrix in hand, the transformation of the global system is straightforward as described in (47). Please note that $\hat{\mathbf{n}}_b$ should be updated according to the boundary particle deformation.

3.2.4 Contact consideration

One further extension feature of the nonconforming boundary conditions is to consider a releasing contact, which is exceptionally useful in many large deformation transient problems. Here, if nonsticking contact condition is assumed, a gap function g can be evaluated to describe the relative normal displacement between the kinematic fields and the imposed values. The gap function is normally evaluated at a certain boundary particle b, which is also assumed as the contact point, and expressed as:

$$g_b = \hat{\mathbf{n}}_b \cdot [\mathbf{u}(\mathbf{x}_b) - \bar{\mathbf{u}}_b] = \hat{\mathbf{n}}_b \cdot \left[\sum_{j=1}^{n_n} N_j(\xi_b) \mathbf{u}_j - \bar{\mathbf{u}}_b \right], \qquad (55)$$

where $\hat{\mathbf{n}}_b$ indicates the unit normal vector pointing outwards and u_p the prescribed displacement. The gap function g indicates whether the boundary condition should be imposed or not, i.e., if $g < 0$, the continuum field nearby the boundary is penetrating the interface, and hence, the penalty imposition (Eq. (40)) should be appended to the system of equations. However, if $g \geq 0$, no further treatment is needed, and the particles at the vicinity of the contact area are allowed to separate.

By including the contact consideration using the aforementioned gap function, we can model a nonsticking contact condition for both nonslip and slip conditions. Moreover, this allows more accurate imposition of Dirichlet boundary conditions as it reduces the element band where the boundary condition is enforced. However, it requires smaller time increment for problems with high relative velocity towards the boundary, as otherwise, the

Initial surface meshes

Generated boundary particles with their area (integration weight) contour
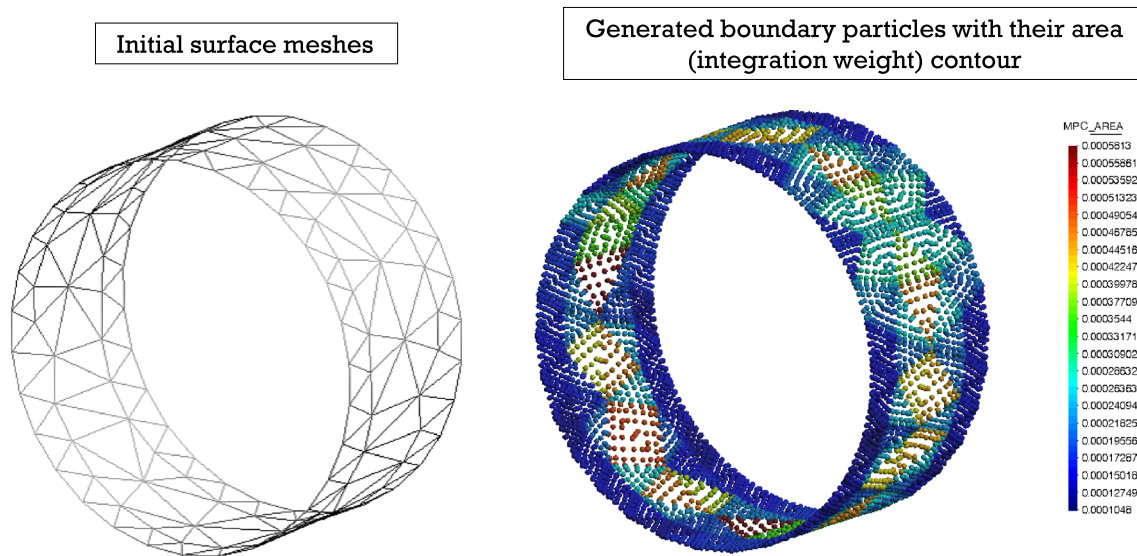
Fig. 4 Surface discretization and boundary particle generation: (left) initial triangle surface mesh and (right) generated boundary particle with its integration weight contour
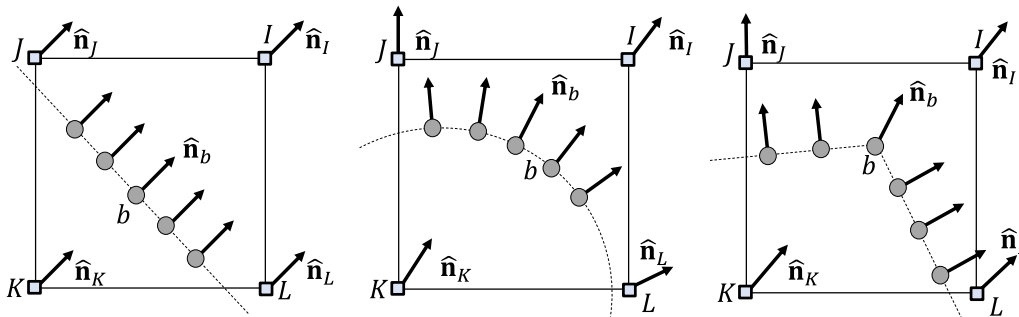
Fig. 5 Illustrative figure describing unit normal vector of the imposed boundary particles and the background nodes

boundary enforcement will be too late and the incoming material points may penetrate the nonconforming boundary surfaces.

4 Numerical examples

This section aims at assessing the quality of the numerical solutions achieved with the proposed MPM formulation with nonconforming penalty boundary condition. Four validation tests have been studied and will be presented in the following subsections. Here, the original MPM linear basis function with a Dirac delta characteristic function of MP volume is assumed for all cases.

1. A static hyperelastic cantilever beam under self-weight is used to investigate quantitatively the convergence of Dirichlet boundary imposition by changing the penalty factor β, the size of the background mesh, and material parameters.

2. The collapse of a granular column with nonconforming side walls is simulated to demonstrate the capability of the proposed boundary imposition in elastoplastic dynamic problems.

3. A rolling cylinder on an inclined slope is simulated in 2D and 3D to analyze the accuracy of the penalty-based contact and slip condition in comparison to analytical results.

4. A granular mixing problem in a rotating drum is addressed to demonstrate the applicability of a nonconforming (round) wall with prescribed displacement and detaching condition.

4.1 Static hyperelastic cantilever beam under self-weight

A two-dimensional static analysis of a hyperelastic cantilever beam is simulated to investigate quantitatively the convergence of the penalty boundary imposition by modifying numerical, geometrical, and material properties. The

cantilever beam considered in this case has a length of $l = 8\,\text{m}$ and a square cross section with unit area of $A = 1 \times 1\,\text{m}^2$ as sketched in Fig. 6. The material model is assumed to be neo-Hookean hyperelastic. Initially, the material parameters are set as follows: density $\rho = 1000\,\text{kg/m}^3$, Young's modulus $E = 90\,\text{MPa}$, and Poisson's ratio $v = 0.0$.

Three convergence studies were conducted to quantify the performance of the proposed method with respect to different numerical conditions. In this study, four material points per cell were considered for all simulations, whereas seven boundary particles per initial boundary segment were assumed. First, the influence of the penalty factor is measured by comparing the displacement obtained in the observation point A (Fig. 6) with the reference results conducted with direct enforcement of the boundary conditions on the nodes, as done in [55]. Both the absolute and the relative errors are evaluated according to

$$e_a = |\delta_A - \delta_A^{\text{ref}}|, \tag{56}$$

$$e_r = \left| \frac{\delta_A - \delta_A^{\text{ref}}}{\delta_A^{\text{ref}}} \right|, \tag{57}$$

where the reference vertical displacement is denoted by δ_A^{ref}, which is equal to the following expression according to [75]:

$$\delta_A^{\text{ref}} = -\left(\frac{\rho g (bhl) l^3}{8EI} + \frac{\rho g l^2}{2G A_s} \right). \tag{58}$$

where g is the gravity acceleration, $I = bh^3/12$ is the inertia of the beam section, and $A_s = 5/6A$ is the reduced cross section area due to the shear effect. G is the beam shear modulus, which is equal to $G = \frac{E}{2(1+v)}$.

Figure 7 (left) shows that both errors are converging linearly as the penalty factor β increases. It is observed that this linearly decreasing tendency is only valid up to a certain threshold. After passing a certain magnitude of penalty factor, the obtained results show a nonchanging behavior; this can be observed by the change of convergence slope around $\beta = 10^{15}$. Moreover, Fig. 7 (right) demonstrates the converging tendency with respect to background mesh size, which is linear for very coarse meshes and quadratic at smaller element size. At larger element size, the error is observed to be governed by the choice of penalty factor, in this case $\beta = 10^{15}$ is kept constant during the convergence study, whereas, at smaller element size, the error returns back to quadratic as the accuracy gained by reducing the element size is more dominant. However, this means that nonconforming imposition of boundary conditions in very coarse meshes exhibits lower accuracy than the standard grid-conforming approach which has a quadratic convergence rate as shown by [55].

Last but not least, Fig. 8 shows the convergence analysis for different values of the Young's modulus. As the Young's modulus influences the condition number of the tangent stiffness matrix, the effective value of penalty factor is inherently influenced by the value of the material Young's modulus. In Fig. 8 (left), the convergence lines, varying the penalty factor β for four different values the of Young's modulus, are plotted. There linear convergence is reached; this is expected and is previously shown in Fig. 7 (left). This is the motivation behind the convergence study shown in Fig. 8 (right). This convergence study gives an insight on how to choose the suitable value of penalty factor β given a material's Young's modulus and the intended relative error. From the author's experience, a
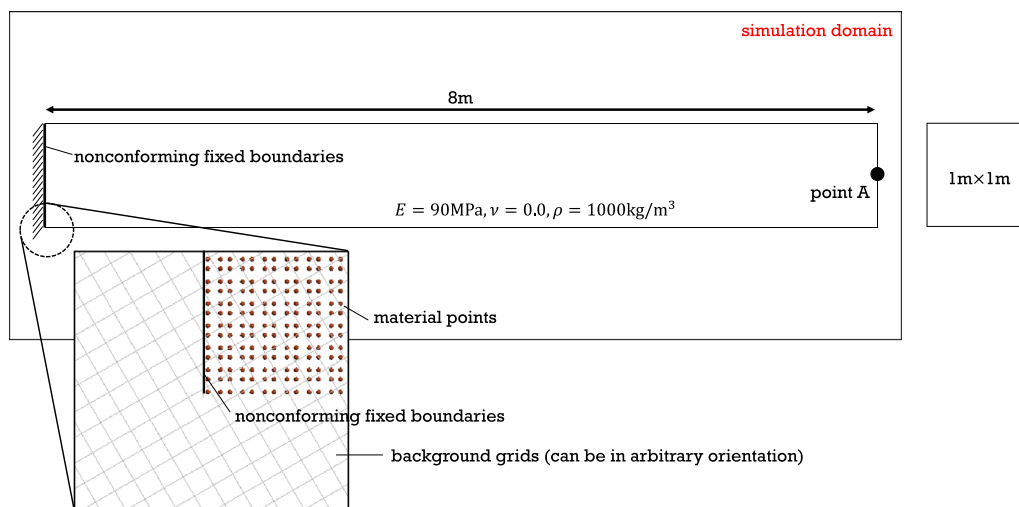
Fig. 6 Static hyperelastic beam: geometry, boundary conditions, and initial material parameters. The constrained left boundary is nonconforming with the background mesh
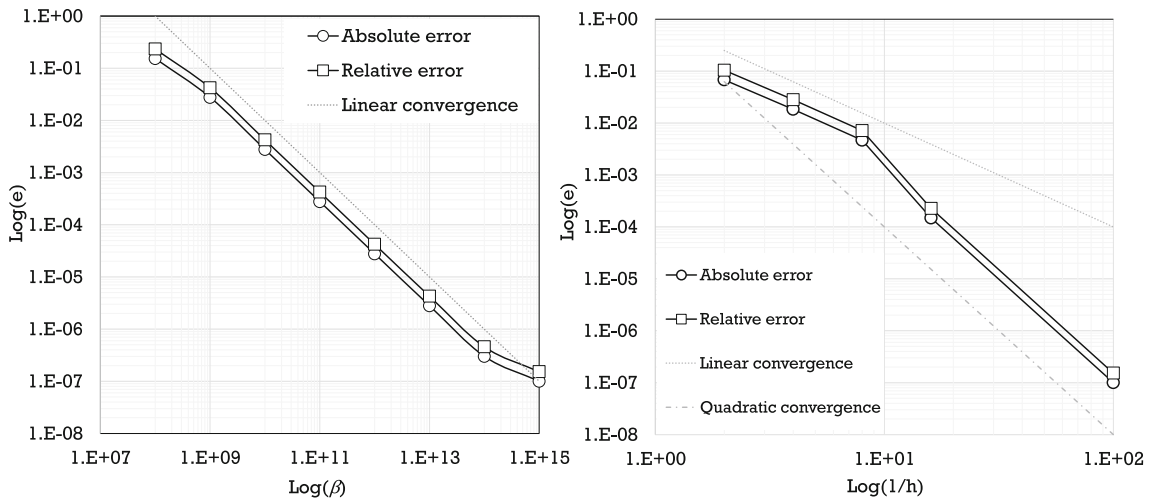
Fig. 7 Static hyperelastic beam: convergence analysis: (left) varying the penalty factor β and (right) varying the background element size

ratio of $\frac{\beta}{E} = 10^3 - 10^4$ is normally suitable to achieve a considerable accuracy for engineering analysis.

4.2 Granular column collapse with nonconforming side walls

A 2D granular column collapse simulation of noncohesive soil was simulated according to the two-dimensional experiment conducted by Bui [76]. In this validation case, the soil is modeled using a nonassociated elastoplastic model, assuming Mohr–Coulomb yield criterion as specified by Clausen [77]. However, the implementation of the model has been adapted to the finite strain assumption to handle relatively larger deformation at every time step. Here, the elastic material parameters of the granular soil are set to be: $E = 840$ kPa, $v = 0.3$, and $\rho = 2650$ kg/m^3,

for the material Young's modulus, Poisson's ratio, and density, respectively, whereas the plastic variables, such as the angle of internal friction, cohesion, and the angle of dilation are $\phi' = 19.8°$, $c' = 0.0$ kPa, and $\psi = 0.0°$, respectively, which describe a noncohesive, nondilative granular material.

In this example, the left and bottom side walls are assumed to be nonslip and implemented either as grid-conforming boundaries [57] or as nonconforming boundaries. The detailed simulation model can be seen in Fig. 9 with a zoomed region indicating the nonconforming boundary. Linear structured triangular elements with mesh size $h = 0.002$ m are used as background mesh. Here, three material points per cell and six boundary particles per line segment are considered. The simulations are run until reaching the final runout ($t_{end} = 2.0$ s).
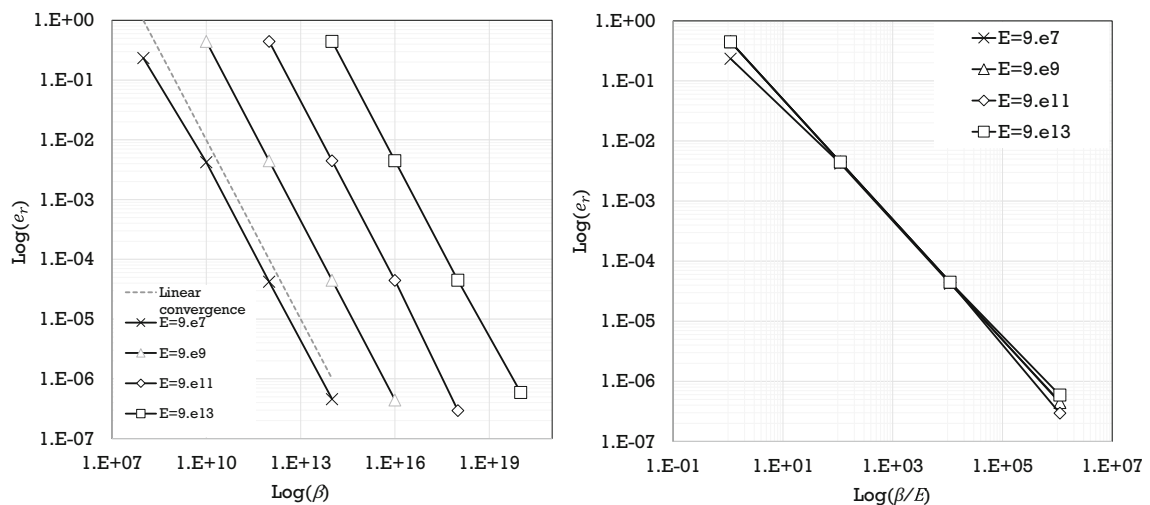

Fig. 8 Static hyperelastic beam: convergence analysis varying the Young's modulus E for (left) regular and (right) weighted result

In Fig. 10, the obtained granular soil configurations, for both grid-conforming and nonconforming boundary conditions, are compared with the experimental data and the simulation results conducted by [76] using the smoothed particle hydrodynamics (SPH) method. One can observe, as indicated in Fig. 10c, that the obtained surface configuration and the failure line show a good agreement with the experimental measurement as well as the simulations conducted by other methods. As discussed before, the quality of penalty boundary imposition strongly depends on the input value of the penalty factor β. For this validation test, the penalty factor is chosen to be $\beta = 10^{11}$. All in all, the nonconforming Dirichlet boundary imposition by penalty method can also be confirmed to work well with elastoplastic granular material model.

4.3 Cylinder on an inclined slope

A cylinder rolling down an inclined slope of $60°$ is simulated in both 2D and 3D. As the cylinder is solely driven by gravitational force, the achieved calculation results can be compared to the analytical solution as proposed by [78] for both rolling without slip and friction-less sliding conditions, which are enforced by the proposed Penalty Method. Figure 11 shows the geometry of the cylinder and the plane in 3D, where the diameter and height of the cylinder are both 100 cm. The 2D geometry is nothing but the cross section cut of the 3D model presented in Fig. 11.

The material model is considered to be linear elastic with a very large stiffness and a density of $\rho = 3000\,\mathrm{kg/m^3}$. For the numerical model, an unstructured tetrahedral mesh is utilized having a mesh size of 8 cm. The cylinder itself is discretized initially by three material points located within each element. The condition is imposed by introducing nine boundary particles for each surface element, each of them having a penalty factor of 10^{18}. The time step for the implicit calculation is chosen to be 10^{-3}. The evolution of the position of the center axis of the cylinder in time is plotted in Fig. 12 and compared with the analytical solution for 2D and 3D case. One can observe that the obtained results match very well to the analytical solution for both the rolling and sliding conditions. This confirms the proposed method to being applicable for arbitrarily inclined slip conditions independently of the considered background mesh.

It is worth noticing that in this numerical example the implemented contact algorithm, essential for rolling condition without slip, generally requires higher mesh resolution in comparison to the slip condition only. For instance, in 2D rolling condition, the solution plotted in Fig. 12 can be achieved by assuming mesh size of $h = 1.25\,\mathrm{cm}$, whereas, for the slip condition, the mesh size can be $h = 8\,\mathrm{cm}$, which is more than 6 times larger. This leads to the need of a significantly larger number of particles, especially for 3D case, increasing sensibly the computational cost and calling for distributed memory calculation capability.
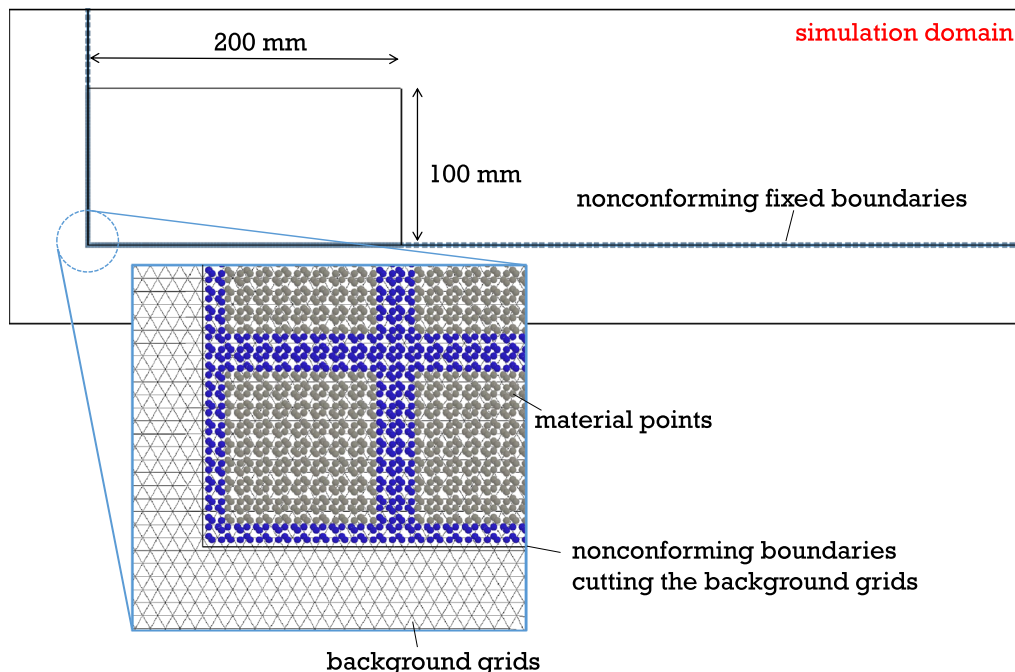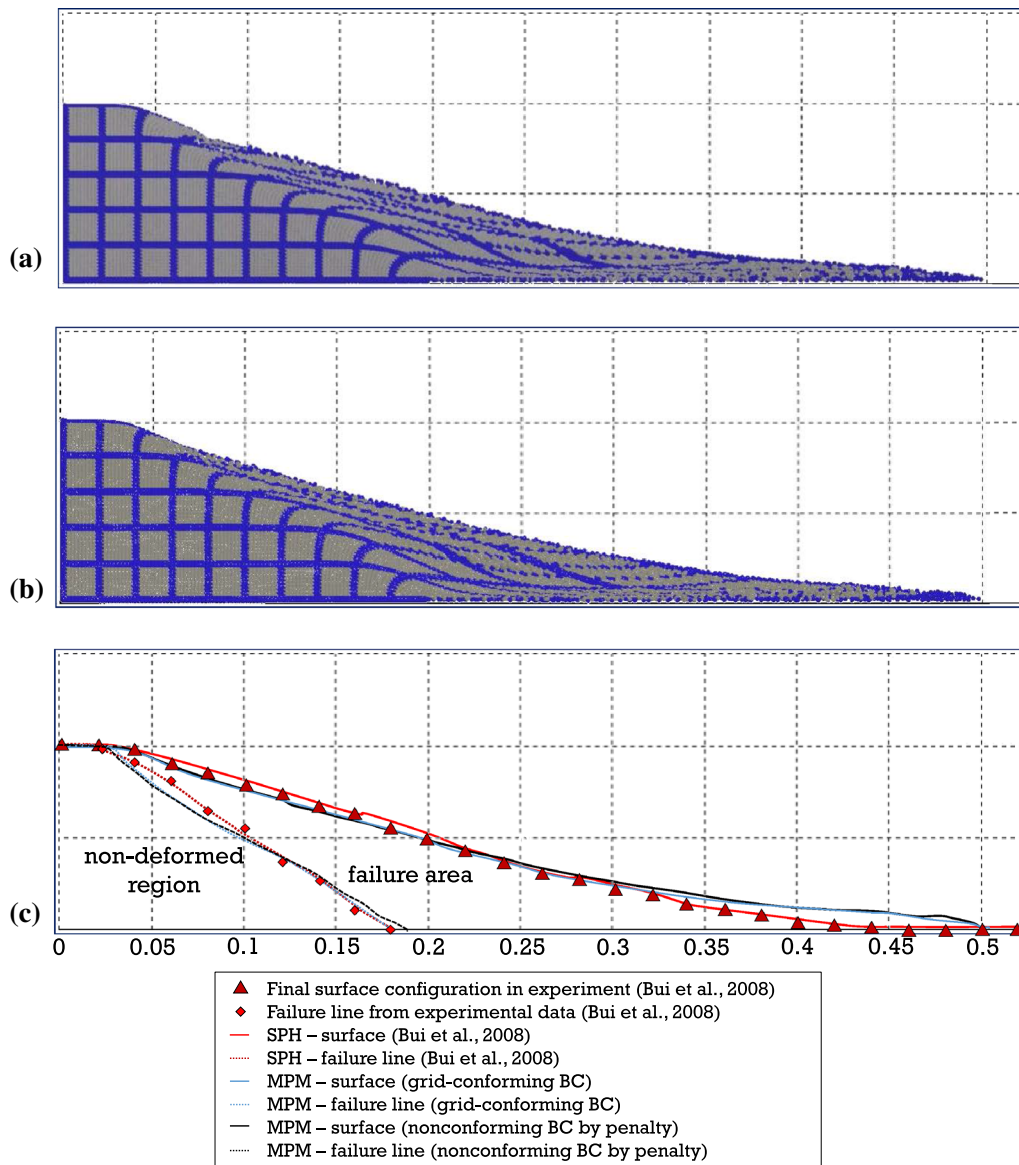

Fig. 9 Granular flow: simulation model with nonconforming Dirichlet boundaries

Fig. 10 Granular flow: simulation results obtained with **a** grid-conforming (default) and **b** nonconforming left and bottom walls; and **c** comparison of final surface configuration and failure with results obtained by [76]

## 4.4 Granular mixing in a rotating drum

To show the performance of the moving boundaries including the releasing contact algorithm, the example of the rotating drum proposed by Zuo et al. [79] is chosen. They compared the experimental data to the simulations obtained using the discrete element method as well as MPM and GIMP. It is worth noticing that, while this is a classical example to study the mixing of granular materials, in this paper it was chosen for the challenging imposition of boundary conditions: there is a circular, moving boundary, which requires releasing contact treatment. The mixing procedure itself is out of scope for the current work and will not be analyzed in detail.

The considered rotating drum, partially filled with dry granular material, has a diameter of 0.1 m and a height of 0.05 m. The three-dimensional geometry is reduced to a 2D plane strain model, represented in Fig. 13.

The rotating speed of the drum is $\omega = 10$ rpm, while the gravitational acceleration is set as 9.81 m/s². The material model chosen for this example is the Mohr–Coulomb model, whereas Zuo et al. [79] used a Drucker–Prager yield surface with an equivalent set of parameters. The material parameters are set to be: $E = 2700$ Pa, $v = 0.2$, and $\rho = 920$ kg/m³, for the Young's modulus, Poisson's ratio, and density of the material, respectively. The plastic variables, such as the angle of internal friction, cohesion, and the angle of dilation, are $\phi' = 27°$, $c' = 0.0$ kPa, and
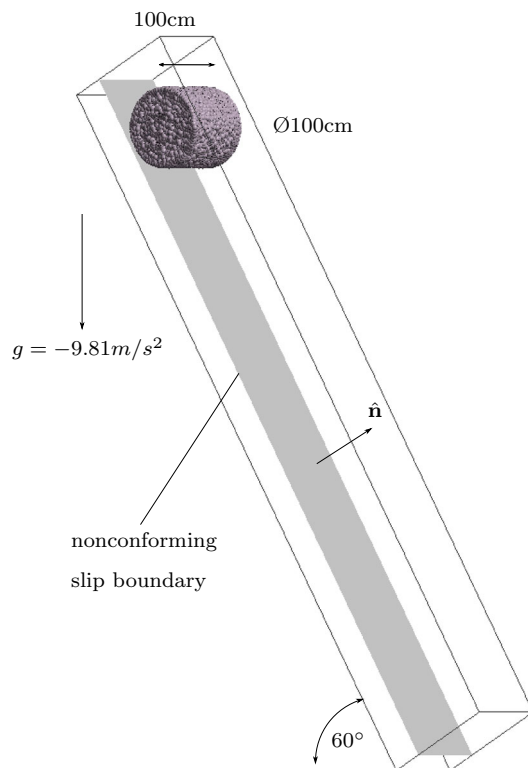
Fig. 11 Cylinder on an inclined slope: geometry and boundary conditions
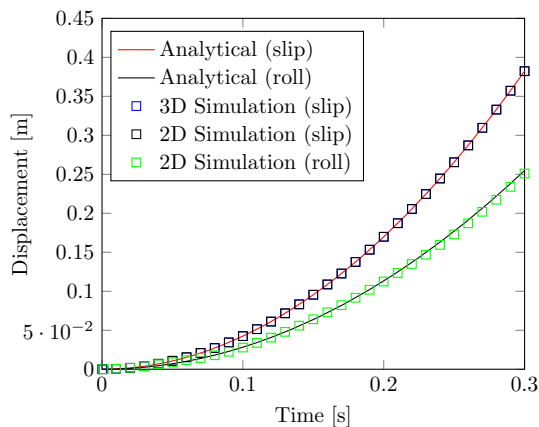
Fig. 12 Cylinder on an inclined slope: simulation results for frictionless slip condition in comparison to the analytical solution

$\psi = 0.0°$, respectively, which describe a noncohesive, nondilative granular material.

While a quadrilateral background mesh with element size of 3 mm is used for the simulation, an unstructured triangular mesh is employed for the discretization of the material points. The latter mesh size is set as 2 mm, and three material points per element are generated. Three mass-less boundary particles are located within each boundary segment, getting the corresponding information

of the normal direction and the prescribed angular velocity which are updated during the calculation process by means of unit quaternions. For the implicit calculation scheme a time step of 5×10^{-4} is used.

Figure 14 presents the results of the simulation at time step $t = 0$ s and $t = 3.5$ s in comparison to [79]. It can be observed that the obtained results show less mixing of the two materials. This is due to the fact that in the present simulation a contact model between the material points which allows relative sliding and separation between elements is not considered, while [79] considers friction contact condition between the granular material and drum particles. Nevertheless, the simulation results are in good agreement with the results by [79]. Furthermore, focusing on the boundary imposition two main achievements can be observed. First, the imposition of the nonconforming curved moving boundary can be imposed accurately without modeling the rotating drum as done by [79]. This may significantly reduce the computational cost. Secondly, even more important, by imposing the Dirichlet boundary condition using the proposed Penalty method, the problem of material points penetrating through the boundary layer, as observed in [79], can be resolved. Future works to include frictional contact boundary are necessary to improve the quality of the mixing.

5 Conclusions

The proposed penalty method, specially designed for an implicit MPM formulation, allows the imposition of Dirichlet boundary conditions on nonconforming, arbitrarily inclined and moving boundaries both in two and three dimensions. This is obtained by introducing moving boundary particles carrying an appended stiffness term determined by the penalty factor β. Hence, the implementation of the proposed method is straightforward as it solely requires an extra stiffness term as well as the corresponding residual force term to be appended to the global systems of equations, without modifying the discretization assumptions of the MPM algorithm. Furthermore, using the proposed approach, nonhomogeneous boundary conditions on top of the slip condition for arbitrary inclined boundary and releasing contact feature can be imposed without solving any additional equation.

However, the accuracy of the method has the classical limitations of the penalty approaches: it strongly depends on the value of the penalty factor β. While β is required to be large enough to enforce the prescribed boundary condition, too large value of β may result in an ill-conditioned tangent stiffness matrix. The penalty factor is influenced by various simulation parameters, such as, among others, the
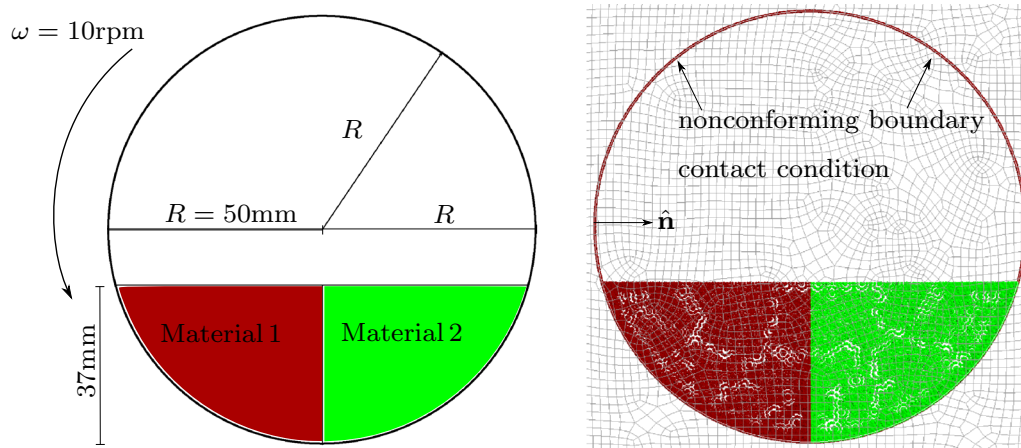
Fig. 13 Rotating drum: geometry and boundary conditions (left) and discretized simulations model (right)
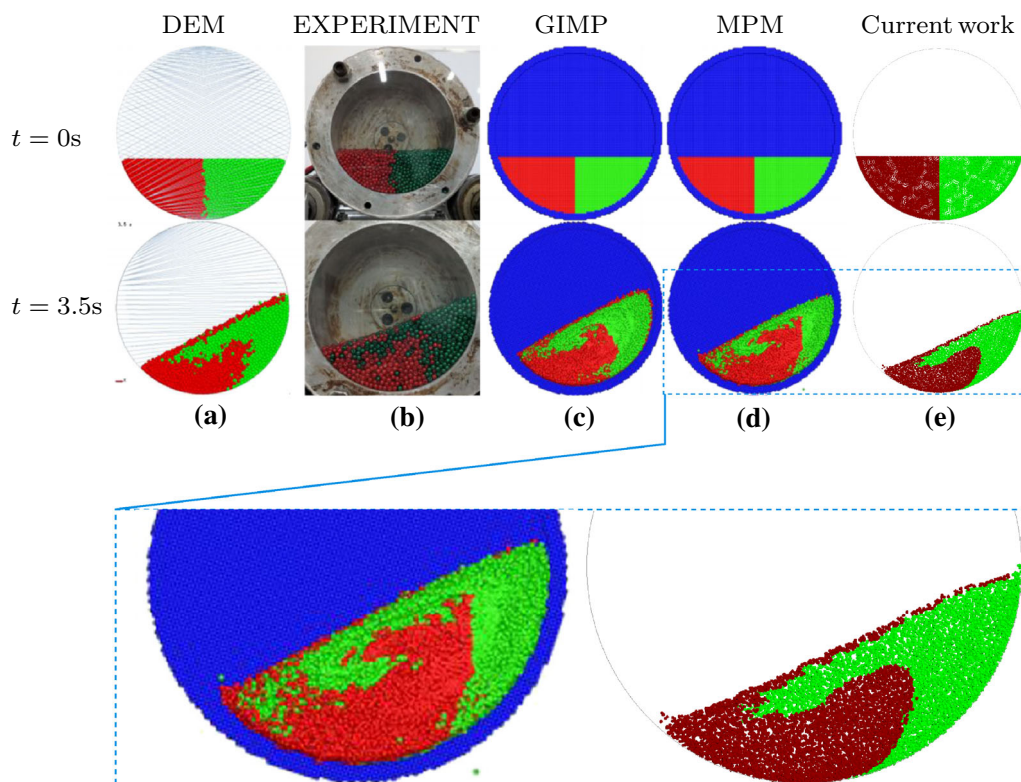

Fig. 14 Rotating drum: comparison of the flow patterns from [79] (**a–d**) and results using penalty method (**e**)

Young's Modulus of the material and the element size of the background mesh, as well as by the prescribed Neumann boundary condition and other external loading. Hence, β needs to be calibrated by the user. Nevertheless, this is inherent for the penalty method and not related to its adaption to MPM.

It is worth noting that the current implementation requires a predefined set of boundary particles to enforce the nonconforming boundary condition. The extension of the method to track and handle dynamic evolution of

boundary particles will be left for future works since it may provide a tremendous benefits in simulating many hydro-geomechanical problems such as erosion and fracture problems. Nevertheless, as long as the boundary particles can be identified and reconstructed, the proposed technique can be applied straightforwardly to the newly constructed surface.

Several numerical examples have been simulated to assess the quality of the proposed work. The comparison of the results with the analytical solution proves the accuracy

and expected convergence rate of the proposed method in two- as well as three-dimensional models. The method shows good agreement with experimental data available in the literature also in elastoplastic regimes. On top of that, the example of the granular mixing in a rotating drum shows the robustness and broad applicability of this method, as curved, moving nonconforming Dirichlet boundary conditions including a releasing contact formulation can be imposed in MPM using the proposed penalty method.

To conclude, the proposed enhancements make MPM more complete for the simulation of large deformation problems of granular and geomaterials and are of paramount importance for the predictive simulation and assessment of the effects caused by environmental phenomena like mud flow and avalanches.

Algorithm 1: Implicit MPM algorithm with penalty augmentation

Input: Simulation parameters (Δt, t_n) and material parameters (E, ν, ρ)

Initialize material point variables: m_p, v_p, $\mathbf{x}_p$, $\mathbf{v}_p$, $\mathbf{a}_p$

Initialize boundary particle variables: A_b, $\hat{\mathbf{n}}_b$, β, $\bar{\mathbf{u}}_b$

1 **PHASE:** 1. Initialization Phase

2   Clear all nodal information.

3   Search background connectivity for all material points p and boundary particles b.

4   Map MP mass, momentum, and force to background connectivity by means of mass extrapolation: Eqs. (17)-(19).

5   Compute initial nodal velocity and acceleration: Eqs. (20) and (21).

6   Compute nodal normal vector from boundary particles: Eq. (54).

7 **end**

8 **PHASE:** 2. Lagrangian Phase:

9   Initialize counter $k = 0$ and compute initial nodal displacement $\mathbf{u}_I^0$ by Newmark's method: see [55] and Eq. (37).

10   Newton-Raphson's loop: **while** *converged == false* **do**

11   Calculate material point contributions: **for** $p \leftarrow 1$ **to** n_p **do**

12   Compute and assemble residual vector $\mathbf{R}^k$ and tangent stiffness matrix $\mathbf{K}^{\text{tan},k}$: Eqs. (15) and (16).

13   **end**

14   Calculate boundary particle contributions: **for** $b \leftarrow 1$ **to** n_b **do**

15   Evaluate contact condition if intended: Eq. (55).

16   Compute and assemble penalty residual vector $\mathbf{R}^{\text{penalty},k}$ and tangent stiffness matrix $\mathbf{K}^{\text{penalty},k}$: Eqs. (43) and (42).

17   **end**

18   Rotate and impose roller support if intended: Eqs. (45)-(49).

19   Solve the imposed linear system of equations for fixed, Eq. (41), or roller support, Eq. (50).

20   Newmark's update for $\mathbf{u}_I^{n+1,k}$, $\mathbf{v}_I^{n+1,k}$ and $\mathbf{a}_I^{n+1,k}$: see [55] and Eq. (38).

21   Check convergence criteria: **if** *passed* **then**

22   Set $(\cdot)_I^{n+1} = (\cdot)_I^{n+1,k}$; *converged = true*.

23   **end**

24       $k = k + 1$

25   **end**

26 **end**

27 **PHASE:** 3. Convective Phase

28   Update the kinematic properties of the material points: Eqs. (22)-(25)

29 **end**

Output: Updated material point and grid variables, $(\cdot)_p^{n+1}$ and $(\cdot)_I^{n+1}$

Appendix A: Implicit MPM algorithm

The proposed formulation described in Sects. 2 and 3 is summarized in this appendix within Algorithm 1, which implementation is accessible in GitHub[2] under the Kratos-Multiphysics Particle Mechanics Application.

---

[2] https://github.com/KratosMultiphysics/Kratos/tree/master/applications/ParticleMechanicsApplication.

References

1. Lucy LB (1977) A numerical approach to the testing of the fission hypothesis. Astron J 82:1013–1024
2. Oñate E, Idelsohn SR, Del Pin F, Aubry R (2004) The particle finite element method: an overview. Int J Comput Methods 1(02):267–307
3. Huang T, Wei H, Chen J, Hillman M (2020) RKPM2D: an open-source implementation of nodally integrated reproducing kernel particle method for solving partial differential equations. Comput

Part Mech 7(2):393–433. https://doi.org/10.1007/s40571-019-00272-x

4. Belytschko T, Lu YY, Gu L (1994) Element-free galerkin methods. Int J Numer Methods Eng 37(2):229–256

5. Koshizuka S, Oka Y (1996) Moving-particle semi-implicit method for fragmentation of incompressible fluid. Nucl Sci Eng 123(3):421–434. https://doi.org/10.13182/NSE96-A24205

6. Zienkiewicz OC, Taylor RL (1977) The finite element method, vol 36. McGraw-Hill, London

7. Peng C, Wang S, Wu W, Wang C, Cheng J (2019) LOQUAT: an open-source GPU-accelerated SPH solver for geotechnical modeling. Acta Geotech 15(5):1269–1287. https://doi.org/10.1007/s11440-019-00839-1

8. Liu M-B, Liu G-R (2016) Particle methods for multi-scale and multi-physics. World Scientific, Singapore

9. Sulsky D, Chen Z, Schreyer HL (1994) A particle method for history-dependent materials. Comput Methods Appl Mech Eng 118(1–2):179–196

10. Harlow FH (1964) The particle-in-cell computing method for fluid dynamics. Methods Comput Phys 3:319–343

11. Brackbill JU, Ruppel HM (1986) FLIP: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. J Comput Phys 65(2):314–343

12. Ma S, Zhang X, Qiu X (2009) Comparison study of MPM and SPH in modeling hypervelocity impact problems. Int J Impact Eng 36(2):272–282

13. Zhang X, Sze K, Ma S (2006) An explicit material point finite element method for hyper-velocity impact. Int J Numer Methods Eng 66(4):689–706

14. Ma S, Zhang X, Lian Y, Zhou X (2009) Simulation of high explosive explosion using adaptive material point method. Comput Model Eng Sci (CMES) 39(2):101

15. Chen Z, Hu W, Shen L, Xin X, Brannon R (2002) An evaluation of the MPM for simulating dynamic failure with damage diffusion. Eng Fract Mech 69(17):1873–1890

16. Zhang H, Wang K, Chen Z (2009) Material point method for dynamic analysis of saturated porous media under external contact/impact of solid bodies. Comput Methods Appl Mech Eng 198(17–20):1456–1472

17. Bandara SS (2013) Material point method to simulate large deformation problems in fluid-saturated granular medium. University of Cambridge, Cambridge Ph.D. thesis

18. Yerro Colom A, Alonso Pérez de Agreda E, Pinyol Puigmartí NM (2015) The material point method for unsaturated soils. Géotechnique 65(3):201–217

19. Zhang X, Chen Z, Liu Y (2016) The material point method: a continuum-based particle method for extreme loading cases. Academic Press, Cambridge

20. Fern J, Rohe A, Soga K, Alonso E (2019) The material point method for geotechnical engineering. A practical guide. CRC Press, Boca Raton

21. Bardenhagen SG, Kober EM (2004) The generalized interpolation material point method. Comput Model Eng Sci 5(6):477–496

22. Sadeghirad A, Brannon RM, Burghardt J (2011) A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. Int J Numer Methods Eng 86(12):1435–1456

23. Sadeghirad A, Brannon RM, Guilkey J (2013) Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces. Int J Numer Methods Eng 95(11):928–952

24. Codina R, Baiges J (2009) Approximate imposition of boundary conditions in immersed boundary methods. Int J Numer Methods Eng 80(11):1379–1405. https://doi.org/10.1002/nme.2662

25. Urquiza J, Garon A, Farinas M-I (2014) Weak imposition of the slip boundary condition on curved boundaries for stokes flow.

J Comput Phys 256:748–767. https://doi.org/10.1016/j.jcp.2013.08.045

26. Massing A, Schott B, Wall W (2018) A stabilized Nitsche cut finite element method for the Oseen problem. Comput Methods Appl Mech Eng 328:262–300. https://doi.org/10.1016/j.cma.2017.09.003

27. Winter M, Schott B, Massing A, Wall W (2018) A Nitsche cut finite element method for the Oseen problem with general Navier boundary conditions. Comput Methods Appl Mech Eng 330:220–252. https://doi.org/10.1016/j.cma.2017.10.023

28. Zorrilla R, Larese A, Rossi R (2019) A modified finite element formulation for the imposition of the slip boundary condition over embedded volumeless geometries. Comput Methods Appl Mech Eng 353:123–157

29. al Kafaji IK (2013) Formulation of a dynamic material point method (MPM) for geomechanical problems. Ph.D. thesis, University of Stuttgart

30. Wang L, Coombs WM, Augarde CE, Brown M, Knappett J, Brennan A, Richards D, Blake A (2017) Modelling screwpile installation using the mpm. Proc Eng 175:124–132

31. Ceccato F, Beuth L, Vermeer PA, Simonini P (2016) Two-phase material point method applied to the study of cone penetration. Comput Geotech 80:440–452

32. Galavi V, Beuth L, Coelho BZ, Tehrani FS, Hölscher P, Van Tol F (2017) Numerical simulation of pile installation in saturated sand using material point method. Proc Eng 175:72–79

33. Beuth L (2012) Formulation and application of a quasi-static material point method. Ph.D. thesis, University of Stuttgart

34. Mast CM, Mackenzie-Helnwein P, Arduino P, Miller GR (2011) Landslide and debris flow-induced static and dynamic loads on protective structures. In: Multiscale and multiphysics processes in geomechanics. Springer, pp 169–172

35. Bing Y, Cortis M, Charlton T, Coombs W, Augarde C (2019) B-spline based boundary conditions in the material point method. Comput Struct 212:257–274

36. Yang W-C (2016) Study of tsunami-induced fluid and debris load on bridges using the material point method. Ph.D. thesis, University of Washington

37. Remmerswaal G, Vardon P, Hicks M, Acosta JG (2017) Development and implementation of moving boundary conditions in the Material Point Method. ALERT Geomater 28

38. Bing Y (2017) B-spline based boundary method for the material point method. Ph.D. thesis, Durham University

39. Cortis M, Coombs W, Augarde C, Brown M, Brennan A, Robinson S (2018) Imposition of essential boundary conditions in the material point method. Int J Numer Methods Eng 113(1):130–152

40. Burla RK, Kumar AV (2008) Implicit boundary method for analysis using uniform B-spline basis and structured grid. Int J Numer Methods Eng 76(13):1993–2028

41. Kumar AV, Padmanabhan S, Burla R (2008) Implicit boundary method for finite element analysis using non-conforming mesh or grid. Int J Numer Methods Eng 74(9):1421–1447

42. Kumar AV, Burla R, Padmanabhan S, Gu L (2008) Finite element analysis using nonconforming mesh. J Comput Inf Sci Eng 8(3):031005

43. Zhang Z, Kumar AV (2017) Immersed boundary modal analysis and forced vibration simulation using step boundary method. Finite Elem Anal Des 126:1–12

44. Chen Z, Brannon R (2002) An evaluation of the material point method. In: SAND Report, SAND2002-0482 (February 2002)

45. Hamad F, Moormann C (2014) Formulation of a dynamic material point method and applications to soil-water-geotextile systems . https://books.google.it/books?id=5wz8oQEACAAJ

46. Babuška I, Zlámal M (1973) Nonconforming elements in the finite element method with penalty. SIAM J Numer Anal 10(5):863–875

47. Breitenberger M, Apostolatos A, Philipp B, Wüchner R, Bletzinger K-U (2015) Analysis in computer aided design: nonlinear isogeometric b-rep analysis of shell structures. Comput Methods Appl Mech Eng 284:401–457

48. Teschemacher T, Bauer A, Oberbichler T, Breitenberger M, Rossi R, Wüchner R, Bletzinger K-U (2018) Realization of cad-integrated shell simulation based on isogeometric b-rep analysis. Adv Model Simul Eng Sci 5(1):19

49. Hughes TJ, Taylor RL, Sackman JL, Curnier A, Kanoknukulchai W (1976) A finite element method for a class of contact-impact problems. Comput Methods Appl Mech Eng 8(3):249–276

50. Zhu T, Atluri S (1998) A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput Mech 21(3):211–222

51. Atluri S, Zhu T-L (2000) The meshless local Petrov–Galerkin (MLPG) approach for solving problems in elasto-statics. Comput Mech 25(2–3):169–179

52. Fernández-Méndez S, Huerta A (2004) Imposing essential boundary conditions in mesh-free methods. Comput Methods Appl Mech Eng 193(12–14):1257–1275

53. Burman E, Claus S, Hansbo P, Larson MG, Massing A (2015) CutFEM: discretizing geometry and partial differential equations. Int J Numer Methods Eng 104(7):472–501

54. Schillinger D, Ruess M, Zander N, Bazilevs Y, Düster A, Rank E (2012) Small and large deformation analysis with the p-and B-spline versions of the finite cell method. Comput Mech 50(4):445–478

55. Iaconeta I, Larese A, Rossi R, Guo Z (2017) Comparison of a material point method and a galerkin meshfree method for the simulation of cohesive-frictional materials. Materials 10(10):1150

56. Iaconeta I, Larese A, Rossi R, O nate E (2018) A stabilized mixed implicit material point method for non-linear incompressible solid mechanics. Comput Mech 1–18

57. Chandra B, Larese A, Iaconeta I, Rossi R, Wüchner R (2019) Soil-structure interaction simulation of landslides impacting a structure using an implicit material point method. In: Proceedings of the 2nd international conference on the material point method for modelling soil-water-structure interaction. Cambridge, pp 72–78

58. Dadvand P, Rossi R, O nate E (2010) An object-oriented environment for developing finite element codes for multi-disciplinary applications. Arch Comput Methods Eng 17(3):253–297

59. Iaconeta I (2019) A discrete-continuum hybrid modelling of flowing and static regimes. Ph.D. thesis Universitat Politécnica de Catalu na

60. Wriggers P (2008) Nonlinear finite element methods. Springer, Berlin

61. Sulsky D, Zhou S-J, Schreyer HL (1995) Application of a particle-in-cell method to solid mechanics. Comput Phys Commun 87(1–2):236–252

62. Bardenhagen S (2002) Energy conservation error in the material point method for solid mechanics. J Comput Phys 180(1):383–403

63. Nairn JA (2003) Material point method calculations with explicit cracks. Comput Model Eng Sci 4(6):649–664

64. Guilkey JE, Weiss JA (2003) Implicit time integration for the material point method: quantitative and algorithmic comparisons with the finite element method. Int J Numer Methods Eng 57(9):1323–1338

65. Wang B, Vardon PJ, Hicks MA, Chen Z (2016) Development of an implicit material point method for geotechnical applications. Comput Geotech 71:159–167

66. Wang L, Cortis M, Coombs WM, Augarde C, Brown M, Knappett J, Brennan A, Davidson C, Richards D, Blake A (2019) On implementation aspects of implicit MPM for 3D analysis. In: Proceedings of the 2nd international conference on the "material point method for modelling soil-water-structure interaction"

67. Chandra B (2019) Soil-structure interaction simulation using a coupled implicit material point - finite element method. Technische Universität München, Masterarbeit

68. Charlton T, Coombs W, Augarde C (2017) igimp: an implicit generalised interpolation material point method for large deformations. Comput Struct 190:108–125

69. Coombs WM, Augarde CE, Brennan AJ, Brown MJ, Charlton TJ, Knappett JA, Motlagh YG, Wang L (2020) On lagrangian mechanics and the implicit material point method for large deformation elasto-plasticity. Comput Methods Appl Mech Eng 358:112622

70. Hamad F, Giridharan S, Moormann C (2017) A penalty function method for modelling frictional contact in MPM. Proc Eng 175:116–123

71. Ma J, Wang D, Randolph M (2014) A new contact algorithm in the material point method for geotechnical simulations. Int J Numer Anal Methods Geomech 38(11):1197–1210

72. Liu C, Sun W (2020) ILS-MPM: an implicit level-set-based material point method for frictional particulate contact mechanics of deformable particles. arXiv:2001.02412

73. Leidinger L, Breitenberger M, Bauer A, Hartmann S, Wüchner R, Bletzinger K-U, Duddeck F, Song L (2019) Explicit dynamic isogeometric B-Rep analysis of penalty-coupled trimmed NURBS shells. Comput Methods Appl Mech Eng 351:891–927

74. Massing A, Larson MG, Logg A, Rognes ME (2014) A stabilized Nitsche fictitious domain method for the Stokes problem. J Sci Comput 61(3):604–628

75. Timoshenko S, Goodier J (1951) Theory of elasticity. McGraw-Hill, London

76. Bui HH, Fukagawa R, Sako K, Ohno S (2008) 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 32(12):1537–1570

77. Clausen J, Damkilde L, Andersen L (2007) An efficient return algorithm for non-associated plasticity with linear yield criteria in principal stress space. Comput Struct 85(23–24):1795–1807

78. Bardenhagen S, Brackbill J, Sulsky D (2000) The material-point method for granular materials. Comput Methods Appl Mech Eng 187(3–4):529–541

79. Zuo Z, Gong S, Xie G (2020) Numerical simulation of granular mixing in a rotary drum using a generalized interpolation material point method. Asia Pac J Chem Eng. https://doi.org/10.1002/apj.2426