# Inhomogeneous Neumann boundary conditions for MPM and GIMP

Remmerswaal, Guido; Vardon, Philip J.; Hicks, Michael A.

**DOI**
10.1016/j.compgeo.2024.106494

**Publication date**
2024

**Document Version**
Final published version

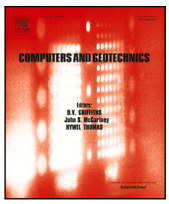
**Published in**
Computers and Geotechnics

Computers and Geotechnics 173 (2024) 106494

Contents lists available at ScienceDirect

Computers and Geotechnics

journal homepage: www.elsevier.com/locate/compgeo



Research paper

# Inhomogeneous Neumann boundary conditions for MPM and GIMP

Guido Remmerswaal [a,b], Philip J. Vardon [a], Michael A. Hicks [a,*]

[a] Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands
[b] Energy and Underground Infrastructure, Geo-Unit, Deltares, Delft, The Netherlands

ARTICLE INFO

*Keywords:*
Boundary conditions
GIMP
MPM
Stress oscillations

ABSTRACT

As the Material Point Method (MPM) uses both a mesh and a point discretisation scheme, the application of boundary conditions is difficult, currently limiting the flexibility of the method. While many boundary condition options have been used in the literature, the accuracy of Neumann boundary condition options has not yet been studied. Four options have here been evaluated for 1D and 2D benchmarks, although none of the options were found to be both accurate and generally applicable in MPM. However, for the generalised interpolation material point method (GIMP), the application of surface tractions on support domain boundaries or on a detected surface are valid options. Large differences between these two accurate options and the application of tractions at surface material points, a method regularly used in the literature, have been observed.

## 1. Introduction

Due to decades of development, the Finite Element Method (FEM) has become an accurate and flexible tool. The flexibility of the mesh to discretise arbitrary geometries and refine locations of interest, as well as the large assortment of (accurate) boundary conditions and constitutive models, have made the method applicable for a wide variety of problems (Bathe, 2014). In other words, the method is generally applicable. However, despite the accuracy and flexibility of FEM, and similar numerical models, they typically break down when large deformations occur. Moreover, the use of small deformation methods limits current practice to stability analysis, where the start of failure is estimated, either using factors of safety or failure probabilities (Griffiths and Lane, 1999; Hicks and Onisiphorou, 2005).

For many applications, modelling the process during and after failure is vital, especially when performing risk analyses. For example, the consequence of a landslide can only be estimated when the process during failure is known. Therefore, several large deformation alternatives for FEM have been developed. One group of alternatives are the so-called Meshless Methods (MMs), which solve governing equations on a set of points or particles (Belytschko et al., 1996). Due to the removal of a fixed nodal connectivity associated with a finite element mesh, mesh distortion is no longer a problem. MMs are promising, but the methods are often computationally intensive. The methods generally must reconstruct the point connectivity, i.e. which points influence each other, by searching and linking nearby particles, and high-order integration is usually required for accurate solutions (Nguyen et al., 2008; Bing et al., 2019). However, low-order integration schemes have recently been developed for MMs (Hillman and Chen, 2016; Wei et al., 2016, 2020).

The Material Point Method (MPM) on the other hand employs an FEM background mesh in combination with a particle scheme. Using an FEM mesh allows for a reuse of knowledge developed for FEM. MPM, originally designed in an explicit form by Sulsky et al. (1994), decouples the material from the mesh, by storing material properties and state variables on material points (MPs), while solving the governing equations on the background mesh. As with MMs, mesh distortion is not an issue for MPM, since only the MPs move and the background mesh can be reset after each computational step. MPM thus enables engineers to model the entire failure process and compute the consequence of failure, as shown in various examples (Sołowski and Sloan, 2015; Yerro et al., 2016; Wang et al., 2016a; Xu et al., 2019; González Acosta et al., 2021).

Recent developments in MPM have focused on solving stress oscillations occurring when MPs move across element boundaries. Many developments modify the procedure used to map properties to and from the background grid, e.g., the Generalised Interpolation Material Point method (GIMP) (Bardenhagen and Kober, 2004); Convected Particle Domain Interpolation (CPDI and CPDI2) (Sadeghirad et al., 2011, 2013); Dual Domain Material Point method (DDMP) (Zhang et al., 2011); and B-Spline Material Point Method (BSMPM) (Tielen et al., 2017; de Koster et al., 2021). The methods remove or reduce the effect of discontinuous shape function gradients at element boundaries.

* Corresponding author.
*E-mail address:* m.a.hicks@tudelft.nl (M.A. Hicks).

https://doi.org/10.1016/j.compgeo.2024.106494
Received 3 April 2024; Received in revised form 30 May 2024; Accepted 31 May 2024
Available online 25 June 2024

## Nomenclature

### Abbreviations

| | |
|---|---|
| BC | Boundary condition |
| BSMPM | B-Spline Material Point Method |
| CPDI | Convected Particle Domain Interpolation |
| DDMP | Dual Domain Material Point |
| FEM | Finite Element Method |
| GIMP | Generalised Interpolation Material Point |
| MM | Meshless method |
| MP | Material point |
| MPM | Material Point Method |
| PF | Proximity Field |
| PFM | Proximity Field Method |

### Latin symbols

| | |
|---|---|
| $\boldsymbol{a}$ | Acceleration |
| $a_j$ | Horizontal size of kernel function |
| $\boldsymbol{b}$ | Body forces |
| $\mathbf{B}$ | Shape function derivative matrix |
| $b_j$ | Vertical size of kernel function |
| $c_0$ | Initial undrained shear strength |
| $C_j$ | Kernel function constant |
| $c_r$ | Residual undrained shear strength |
| $E_{1D}$ | One-dimensional force error |
| $E_{F_x}$ | Horizontal force error |
| $E_{F_y}$ | Vertical force error |
| $\boldsymbol{F}_{\text{body}}$ | Nodal body forces |
| $\boldsymbol{F}^c$ | Point loads |
| $\boldsymbol{F}_{\text{ext}}$ | Nodal external forces |
| $\boldsymbol{F}_{\text{int}}$ | Nodal internal forces |
| $\boldsymbol{F}_{\text{point}}$ | Nodal point loads |
| $\boldsymbol{F}_{\text{traction}}$ | Nodal traction forces |
| $\boldsymbol{F}_p$ | Point load of MP $p$ |
| $\boldsymbol{F}_{\text{MP}}$ | Nodal loads method 1 |
| $\boldsymbol{F}_{\text{Nodes}}$ | Nodal loads method 2 |
| $\boldsymbol{F}_{\text{Surface}}$ | Nodal loads method 3 |
| $\boldsymbol{F}_{\text{Volume}}$ | Nodal loads method 4 (MPM) |
| $\boldsymbol{F}_{\text{GIMP}}$ | Nodal loads method 4 (GIMP) |
| $h$ | Element size |
| $H_{min}$ | Minimum gap height |
| $H_{total}$ | Total height of segments |
| $I$ | Combined invisibility number |
| $I^*$ | Combined invisibility number without gaps |
| $i$ | Node |
| $J$ | Invisibility number for rectangles |
| $\mathbf{K}$ | Stiffness matrix |
| $K$ | Invisibility number for extensions |
| $K_{i,j}$ | Kernel function |
| $lp$ | Support domain size |
| $\mathbf{M}$ | Mass matrix |
| $\mathbf{N}$ | Shape function matrix |
| $N_i$ | FEM shape function |
| $\nabla N_i$ | FEM shape function gradient |
| $nel$ | Number of elements |
| $nmp_{1\text{D}}$ | number of MPs per direction per element |
| $nmp$ | Number of MPs |
| $nn$ | Number of nodes |
| $p$ | MP |
| $\boldsymbol{q}$ | Traction |
| $q$ | Traction value |
| $S_{ip}$ | GIMP shape function |
| $\nabla S_{ip}$ | GIMP shape function gradient |
| $\mathbf{S}_j$ | Shape matrix |
| $t$ | time |
| $\Delta t$ | timestep size |
| $\boldsymbol{\Delta u}$ | Nodal incremental displacements |
| $\boldsymbol{u}_{i,j}$ | Local distance |
| $V$ | Volume |
| $W_{min}$ | Minimum gap width |
| $\boldsymbol{x}$ | global coordinates |
| $x_{\text{offset}}$ | Distance of mesh boundary from surface $\Gamma$ |

### Greek symbols

| | |
|---|---|
| $\Gamma$ | Surface area |
| $\Gamma_m$ | Surface area of element $m$ |
| $\Gamma_P$ | Surface area of MP $p$ |
| $\Gamma_V$ | Surface area proportional to volume in element $m$ |
| $\theta$ | Surface angle |
| $\boldsymbol{\sigma}$ | Cauchy stress |
| $\boldsymbol{\tau}$ | Surface traction |

GIMP, CPDI and CPDI2 assign a physical domain to each MP, and compute the influence of the domain on each element it overlaps. DDMP uses two background grids to reduce the cell crossing error and BSMPM employs a higher order shape function spanning multiple elements. Moreover, since MPs move within elements, they often do not coincide with the optimal locations for numerical integration and stress recovery. This causes additional oscillations on top of those caused by element crossing, and solutions have been developed to reduce these oscillations (Zhang et al., 2011; González Acosta et al., 2020).

A remaining key challenge for MPM, which is gaining attention, is the imposition of boundary conditions (BCs). Dirichlet BCs (fixed primary variable) and Neumann BCs (normal derivative of the primary variable or force in a mechanical governing equation) are most commonly used to solve the governing equations on the background grid. However, since the material, and thus the material boundary, has been decoupled from the grid, defining BCs at the material boundary can be troublesome (Cortis et al., 2018). Compared to MMs, defining BCs in MPM can be considered easier due to the presence of a background grid, because techniques designed for FEM may be applied. For example, FEM BCs can be applied directly when a problem is setup such that a boundary coincides with an edge of the background grid, i.e. conforming BCs. However, in order to make MPM generally applicable, non-conforming BCs, i.e. BCs decoupled from the grid, must be developed.

While several strategies for non-conforming BCs have been used during the development of MPM, of which a review is presented after a further introduction of boundary conditions in MPM in Section 2.2, the application of BCs is not yet fully understood. Therefore, the accuracy of several Neumann BC methods for MPM and GIMP are tested for a one and a two dimensional problem in Sections 3 and 4, respectively. Sections 3 and 4 highlight the BC options which are more or less consistent between the application of the Neumann BC and the computation of the internal force, resulting in conclusions on which BC application options can be employed. To use the most consistent and therefore accurate methods in practice, algorithms must be developed for general applicability. Therefore, the most consistent methods are

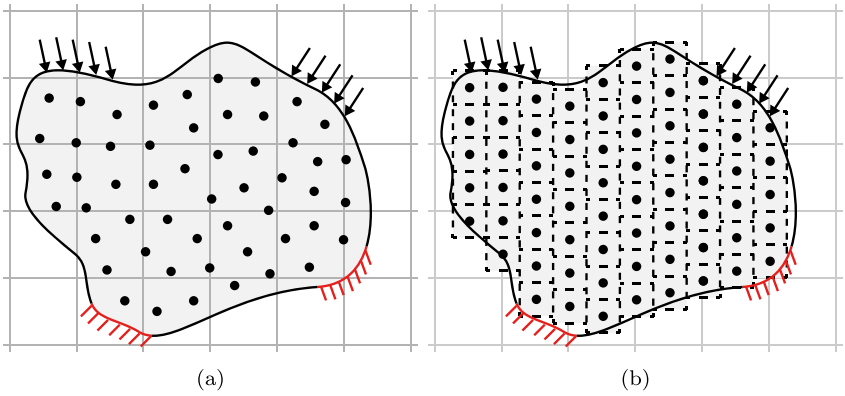



**Fig. 1.** A material body discretised using (a) MPM with MPs on a background grid, and (b) GIMP with MPs and support domains on a background grid.

extended for general applicability (Section 5). The algorithms are then applied to a submerged slope failure case study (Section 6).

## 2. Theoretical background of boundary conditions

In FEM, the governing equations can be discretised using standard FEM discretisation into nodes and elements (see for example Bathe (2014) for a derivation of the governing equations) and expressed in matrix form (González Acosta et al., 2020):

$$\mathbf{M}^t\boldsymbol{a} + \mathbf{K}^t\Delta\boldsymbol{u} = \left(\boldsymbol{F}_{\text{body}} + \boldsymbol{F}_{\text{traction}} + \boldsymbol{F}_{\text{point}} - \boldsymbol{F}_{\text{int}}\right)^{t+\Delta t} \tag{1}$$

where $\mathbf{M}$ is the mass matrix, $\boldsymbol{a}$ is the vector of nodal accelerations, $\mathbf{K}$ is the stiffness matrix, $\Delta\boldsymbol{u}$ is the vector of nodal incremental displacements, and $\boldsymbol{F}_{\text{body}}$, $\boldsymbol{F}_{\text{traction}}$, $\boldsymbol{F}_{\text{point}}$ and $\boldsymbol{F}_{\text{int}}$ are the nodal body forces, surface tractions, point loads and internal forces, respectively. Note that concentrated point loads are non-physical and can cause numerical issues, such as non-convergence, but they can be useful for comparisons with analytical solutions. Usually, the external forces are grouped together and the external and internal forces are then integrated per element (see Wang et al. (2016b) for further details), such that

$$\boldsymbol{F}_{\text{ext}} = \mathop{\mathbb{A}}_{m=1}^{nel} \int_{V_m} \mathbf{N}^{\mathrm{T}}\boldsymbol{b}\,\mathrm{d}V_m + \mathop{\mathbb{A}}_{m=1}^{nel} \int_{\Gamma_m} \mathbf{N}^{\mathrm{T}}\boldsymbol{\tau}\,\mathrm{d}\Gamma_m + \sum_{i=1}^{nn} \boldsymbol{F}_i^c \tag{2}$$

and

$$\boldsymbol{F}_{\text{int}} = \mathop{\mathbb{A}}_{m=1}^{nel} \int_{V_m} \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}^{t+\Delta t}\,\mathrm{d}V_m \tag{3}$$

where $\mathbb{A}_{m=1}^{nel}$ is the element assembly over $nel$ elements, $nn$ is the total number of nodes, $\boldsymbol{F}_{\text{ext}}$ are the external forces, $\mathbf{N}$ is the shape function matrix, $\mathbf{B}$ is the shape function derivative matrix, $\boldsymbol{b}$ are the body forces, $\boldsymbol{\tau}$ is the surface traction, $\boldsymbol{F}_i^c$ is a point load at node $i$, $\boldsymbol{\sigma}^{t+\Delta t}$ is the stress at the end of the timestep, $V_m$ is the volume of element $m$, and $\Gamma_m$ is the surface of element $m$ belonging to $\Gamma$. Note that a concentrated load may only be properly applied when a node is located at its position. The integration in Eqs. (2) and (3) are evaluated numerically using Gauss integration. Eq. (1) fulfils the external load conditions, but the displacement (Dirichlet) conditions must still be enforced (Bathe, 2014; Smith et al., 2014; Zienkiewicz et al., 2014).

### 2.1. BCs in MPM

In MPM, a body is no longer discretised into elements and nodes, since MPs are used instead, see Fig. 1(a). Gauss integration over the body volume $V$ is replaced by a summation over the MPs. Therefore, the body force and internal force become

$$\boldsymbol{F}_{\text{body}} = \sum_{p=1}^{nmp} \mathbf{N}_p^{\mathrm{T}}\boldsymbol{b}V_p \tag{4}$$

and

$$\boldsymbol{F}_{\text{int}} = \sum_{p=1}^{nmp} \mathbf{B}_p^{\mathrm{T}}\boldsymbol{\sigma}^{t+\Delta t}V_p \tag{5}$$

where $\mathbf{N}_p$ and $\mathbf{B}_p$ are the shape function and shape function derivative matrices evaluated at MP $p$, $V_p$ is the volume of $p$, and $nmp$ is the total number of MPs. Eqs. (4) and (5) transfer information between the MP and background mesh discretisation via this integration.

In FEM, the edge of the elements closely resembles the domain edge $\Gamma$ and the element nodes typically lie on the domain edge. Therefore, boundary conditions can be easily included. In MPM, neither the MPs nor the background grid align directly with the domain boundary, see Fig. 1(a), and it is not immediately obvious where surface tractions and point loads should be applied, nor where displacement conditions should be enforced.

Moreover, several of the techniques used to address the aforementioned stress oscillations inherent in MPM alter the point and/or mesh discretisation. For example, in GIMP, an MP is no longer discretised as a single point, but is assigned a rectangular area instead, see Fig. 1(b), with a size defined as $2lp_x \times 2lp_y$. The standard FEM shape functions and shape function gradients are replaced with functions that spread the influence of an MP over multiple elements, thereby accounting for the changed discretisation, see Fig. 2. While, in the initial position, the rectangular support domains of the MPs do not overlap, as shown in Fig. 1(b), the support domains may overlap after movement of the MPs. This change will impact the application of the boundary conditions. Here, BCs in MPM and GIMP are studied, to investigate the effects of different discretisation approaches on the BCs.

A straightforward strategy is to apply BCs on 'surface' MPs, i.e. the MPs closest to the material boundary. Dirichlet conditions can be enforced on surface MPs by fixing the primary variables of MPs (Moormann and Hamad, 2015; Wang et al., 2018). Neumann conditions may also be applied as point loads at surface MPs, by integrating the loads to the background grid (Chen et al., 2002; Hu and Chen, 2003; Martinelli et al., 2017; Fern et al., 2019). Thin membranes have also been tracked using 'surface' MPs (York et al., 1999), and methods have been proposed to compute boundary normal vectors (York, 1997; Torres and Brackbill, 2000). However, since MPs are, by definition, not at the material boundary, the exact location of the boundary is not modelled and therefore this is likely to cause significant inaccuracies and discretisation dependency of the solution. Moreover, for large distortions (which MPM is designed for), the surface MPs may change over time and must therefore be identified, which is a non-trivial task.

Since the governing equations are solved at the nodes, another solution could be to enforce the BCs directly at 'surface' nodes, i.e. those nodes which separate active from inactive elements. This presented reasonable results for a rainfall boundary condition in explicit MPM (Martinelli et al., 2021), even though the BC did not coincide exactly with the material boundary. The results can be further improved when the

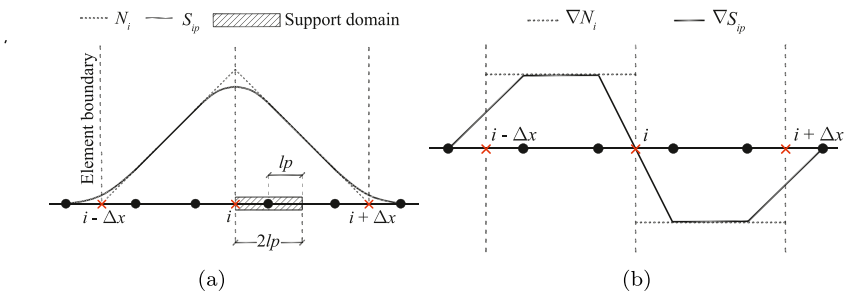



**Fig. 2.** (a) GIMP shape function ($S_{ip}$) and regular FE shape function ($N_i$) of node $i$, and (b) GIMP shape function gradient ($\nabla S_{ip}$) and regular FE shape function gradient ($\nabla N_i$) for node $i$ (Bardenhagen and Kober, 2004; González Acosta et al., 2020).

boundary is forced to be at the nodes; boundaries with an irregular shape can be applied on nodes of an irregular grid (Wang et al., 2005; Tjung et al., 2020; Tjung, 2020), while a moving mesh can be employed for BCs with known displacements (Beuth, 2012; Phuong et al., 2016; Fern et al., 2019; Martinelli and Galavi, 2021; Wang et al., 2021). However, while these methods show good results for a subset of problems, the solutions are not generally applicable. Moreover, standard stress oscillation reduction techniques often require structured grids, and versions for unstructured grids require new developments (de Koster et al., 2021).

MPM boundary conditions may benefit from developments in MMs, which do not benefit from an FEM mesh. In MMs, FEM techniques, such as Lagrange multipliers, the penalty method and Nitsche's method can still be employed with adjustments (Fernández-Méndez and Huerta, 2004). Moreover, new techniques have been developed, for example boundary FEM grids have been coupled to meshless points (Huerta and Fernández-Méndez, 2000; Mast et al., 2011), ghost points have been used to enforce BCs or to track the boundary (Colagrossi and Landrini, 2003; Federico et al., 2012; Mao et al., 2016), and truncation properties of shape functions near the boundary have been used to apply confining stresses (Zhao et al., 2019b).

Recent studies have looked at applying BCs within FEM elements instead of on element boundaries. Dirichlet BCs can be included into the system of equations of nodes surrounding the material boundary using the implicit boundary method (IBM) (Kumar et al., 2008; Cortis et al., 2018; Bing et al., 2019). In IBM, the displacement constraints are enforced using so-called Dirichlet functions. These functions enforce the constraint over a narrow band along the boundary, with the constraint only indirectly affecting the rest of the FEM mesh via the system of equations. While in IBM the BCs are included in the equations of elements surrounding the boundary, in the Shifted Boundary Method (SBM) the BCs are mapped onto the elements enclosed by the boundary (Main and Scovazzi, 2018; Liu and Sun, 2019). In SBM all elements intersecting or outside the boundary are removed from the mesh, and a surrogate boundary is then created at the boundary of the remaining mesh. A distance function is employed to map from the real to the surrogate boundary, such that the BC can be included using Nitche's method. Similar to IBM and SBM, Neumann BCs can be integrated to surrounding nodes using Gauss integration along the boundary (Remmerswaal, 2017; Bing et al., 2019) or mapped to the surrogate boundary (Main and Scovazzi, 2018). These FEM-like methods require the location of the surface, which must be predefined by the user or can be detected based on the MP location and properties (Remmerswaal, 2017).

Besides Dirichlet and Neumann BCs, inflow and outflow BCs, i.e. introduction and removal of material, respectively, can be required within MPM. These flux-based conditions are especially useful to reduce the computational cost in (pore-)fluid mechanics. For example, to model a steady flow through a boundary in space, a velocity (Dirichlet) condition must be enforced at the boundary. However, the MPs would flow away from the boundary, leaving the elements on which the condition is enforced empty, preventing the steady flow through the boundary. Therefore, new material (points) must be introduced through the boundary (Zhao et al., 2019a). Additional elements may be used to introduce the new MPs, and the Dirichlet BC should also be enforced on the new MPs to create a steady flow. To prevent a build up of MPs at the end of the computational domain, an outflow condition can be used, i.e. MPs may be removed when they enter specific elements (Zhao et al., 2019a). To ensure a specific solution, a Neumann condition can be enforced at the outflow elements, and in this case it is usually easy to ensure that these boundaries can conform to the background grid edge.

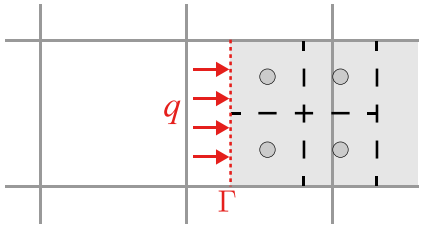



**Fig. 3.** Example of surface traction $q$ applied on the surface of a material $\Gamma$ (red dotted line). The material is discretised into MPs (circles) with their respective support domains (black dashed lines) located within a background mesh (grey solid lines).

### 2.2. Neumann boundary condition methods in MPM

Four possible methods to apply Neumann BCs on an MP discretisation are outlined below and investigated in Sections 3 and 4. The methods are explored using the example configuration in Fig. 3. In this example, the MPs are evenly spaced and translated horizontally with respect to the background grid. For simplicity, the MPs and the boundary $\Gamma$ are aligned with the grid in the vertical direction, but this is not necessary. $\Gamma$ (as indicated by the red dotted line in Fig. 3) is neither located at the edge of a grid cell nor at an MP. MP domains are also shown via the black dashed lines, but these are only defined in methods with particle domains such as GIMP.

#### 2.2.1. Equivalent point loads applied on surface MPs

To apply loads on surface MPs, as shown in Fig. 4(a), $\Gamma$ is split into segments $\Gamma_p$ associated with the closest surface MP. $\boldsymbol{q}$ is integrated along $\Gamma_p$ into an equivalent point load which acts at MP $p$,

$$\boldsymbol{F}_p = \int_{\Gamma_p} \boldsymbol{q}\, \mathrm{d}\Gamma_p \tag{6}$$

and the point loads are integrated to the nodes using shape functions (similar to Eq. (2)) by summing over all boundary MPs:

$$\boldsymbol{F}_{\mathrm{MP}} := \sum_p \mathbf{N}_p^{\mathrm{T}} \boldsymbol{F}_p \tag{7}$$

where $\boldsymbol{F}_{\mathrm{MP}}$ are the nodal external loads obtained from the point loads applied at the boundary MPs. This method is convenient for externally

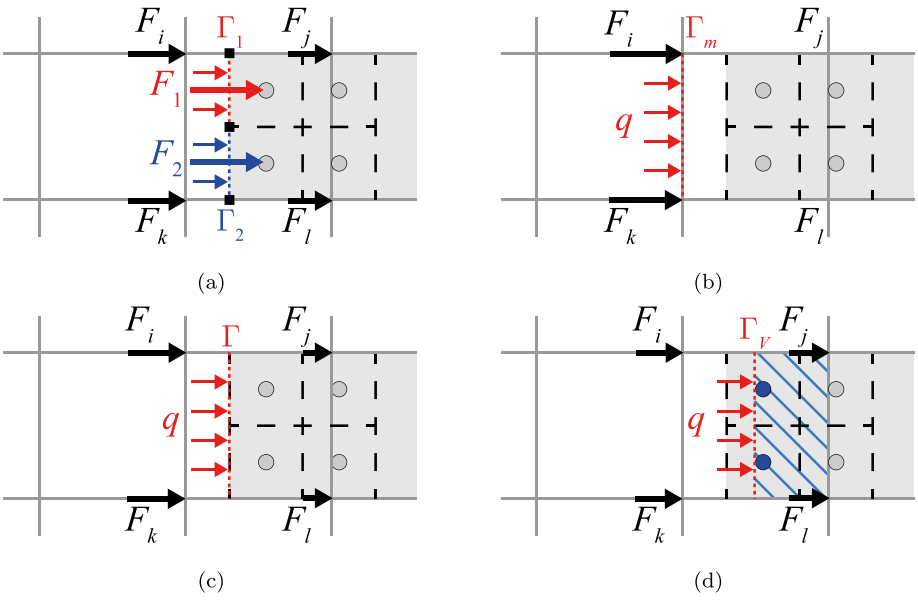



**Fig. 4.** Four Neumann boundary condition methods to transfer $\boldsymbol{q}$ to $\boldsymbol{F}_{\text{ext}}$: (a) application on surface MPs as point loads, obtained from integration over two surface segments; (b) application on surface nodes; (c) direct integration from (detected) surface; (d) application on a surface proportional to the volume of the MPs in the boundary element (blue hatched volume is equal to the volume of the blue MPs).

applied loads which move with the material, as once defined $\boldsymbol{F}_p$ does not need to be recalculated — this can even be done in a pre-processing stage. It is seen that, in contrast to FEM, all nodes of the element containing the BC have external forces associated with the BC, shown as $F_i$ to $F_l$ in Fig. 4(a).

#### 2.2.2. Boundary moved to the surface nodes

With the second method, the boundary condition is applied on the boundary $\Gamma_m$ separating the active elements from the inactive elements, see Fig. 4(b). In MPM, an element is active when it contains at least one MP. In GIMP, an element is active when it overlaps with a part of the support domain of at least one MP. The surface traction part of Eq. (2) is used to integrate the traction to the surface nodes:

$$\boldsymbol{F}_{\text{Nodes}} := \int_{\Gamma_m} \mathbf{N}^{\mathrm{T}} \boldsymbol{q} \, \mathrm{d}\Gamma_m \tag{8}$$

where $\boldsymbol{F}_{\text{Nodes}}$ are the external nodal loads computed from the surface nodes. This method is the most similar to FEM, but also moves the boundary away from its intended location unless the material boundary aligns with the mesh. It is straightforward to apply if the load is fixed in space and not dependent on the movement of the material in the domain, but it requires a substantial housekeeping algorithm to identify the correct location to apply the boundary condition if it moves with the material.

#### 2.2.3. Boundary condition applied exactly

The third method (see Fig. 4(c)) integrates $\boldsymbol{q}$ directly from $\Gamma$. Standard FEM integration can be used within elements as well, such that

$$\boldsymbol{F}_{\text{Surface}} := \int_{\Gamma} \mathbf{N}^{\mathrm{T}} \boldsymbol{q} \, \mathrm{d}\Gamma \tag{9}$$

where $\boldsymbol{F}_{\text{Surface}}$ is the external nodal load integrated directly from $\Gamma$. This integration becomes more complicated in realistic scenarios where the shape of $\Gamma$ may be complex. In these cases, B-splines or Composite Bezier Curves may be used to represent $\Gamma$, and Gauss integration along $\Gamma$ can be used to evaluate Eq. (9), see Remmerswaal (2017) and Bing et al. (2019) for further details. This method (as in the first method) results in nodal forces on all of the nodes in the element, and requires the identification of the exact boundary location. For the case when $\Gamma$ is parallel to the GIMP domain orientation, this method is equivalent to applying loads on the GIMP support domain.

#### 2.2.4. Boundary condition moved proportionally to volume of MPs in background grid elements

The fourth method (Fig. 4(d)) evaluates the volume of the material points within each cell. A boundary surface $\Gamma_V$ is placed in partially filled cells, such that the volume enclosed by the surface is equal to the volume of MPs in each cell. For example, the surface is placed in the middle of the surface element in Fig. 4(d), since the element has only two MPs within it (50% of the original four). $\boldsymbol{q}$ is then integrated along $\Gamma_V$ to obtain $\boldsymbol{F}_{\text{Volume}}$, i.e. the external nodal loads computed from $\Gamma_V$:

$$\boldsymbol{F}_{\text{Volume}} := \int_{\Gamma_V} \mathbf{N}^{\mathrm{T}} \boldsymbol{q} \, \mathrm{d}\Gamma_V \tag{10}$$

This method is difficult to practically implement for arrangements where the MPs are not well aligned with the background grid. In GIMP, the volume of MPs is distributed according to the support domain of the MPs. Therefore, $\Gamma_V$ coincides with the boundary of the support domain and, for clarity, the name $\boldsymbol{F}_{\text{Volume}}$ is changed to $\boldsymbol{F}_{\text{GIMP}}$. In other words, $\boldsymbol{F}_{\text{GIMP}}$ is the nodal force when a traction is applied to the boundary of the support domain of GIMP, and is equivalent to $\boldsymbol{F}_{\text{Surface}}$ when $\Gamma$ is parallel to the GIMP domain orientation.

## 3. Application in one dimension

A simple benchmark is presented to demonstrate the performance and characteristics of the different BC application methods, and to indicate the inconsistencies of the BC methods with respect to the internal force calculation. A mismatch between $\boldsymbol{F}_{\text{ext}}$ and $\boldsymbol{F}_{\text{int}}$ will cause spurious accelerations, and can therefore cause (additional) stress oscillations which have been characteristic of many MPM implementations. A 1D problem is solved using a 2D plane strain discretisation and background grid. An elastic bar, presented in Fig. 5, is fixed at one end and loaded with a surface traction $q = 1$ kPa at the other end. The bar has a Poisson's ratio of 0. The MPs are distributed equally within the bar, with a distance of 0.5 m between adjacent points. The start of the background grid is placed at a distance $x_{\text{offset}}$ from the surface $\Gamma$, and consists of 1 m square elements. The MPs are fixed in place, i.e. no displacements are computed. The initial horizontal stress in all MPs is initialised as $q$, such that the bar is in equilibrium. Therefore, for a consistent solution, $\boldsymbol{F}_{\text{int}}$ and $\boldsymbol{F}_{\text{ext}}$ should be equal at each node for every position of the bar in the background grid, i.e. for all values of $x_{\text{offset}}$.

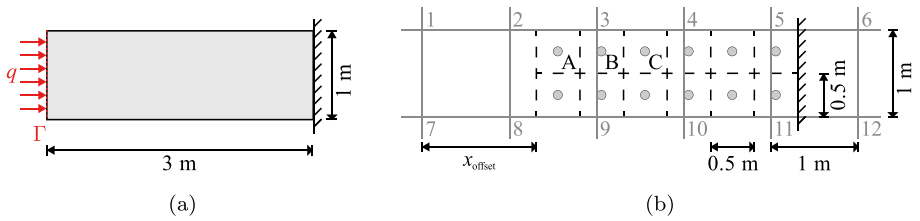



**Fig. 5.** (a) A 3 m by 1 m bar fixed at one end and compressed from the other end by a surface traction $q$; (b) the bar discretised into MPs (dots), representing a square support domain (dashed lines), and the domain discretised as a background grid constructed from 1 m square elements (solid grey lines), in which the start of the grid is located at a distance $x_{offset}$ from the surface $\Gamma$.

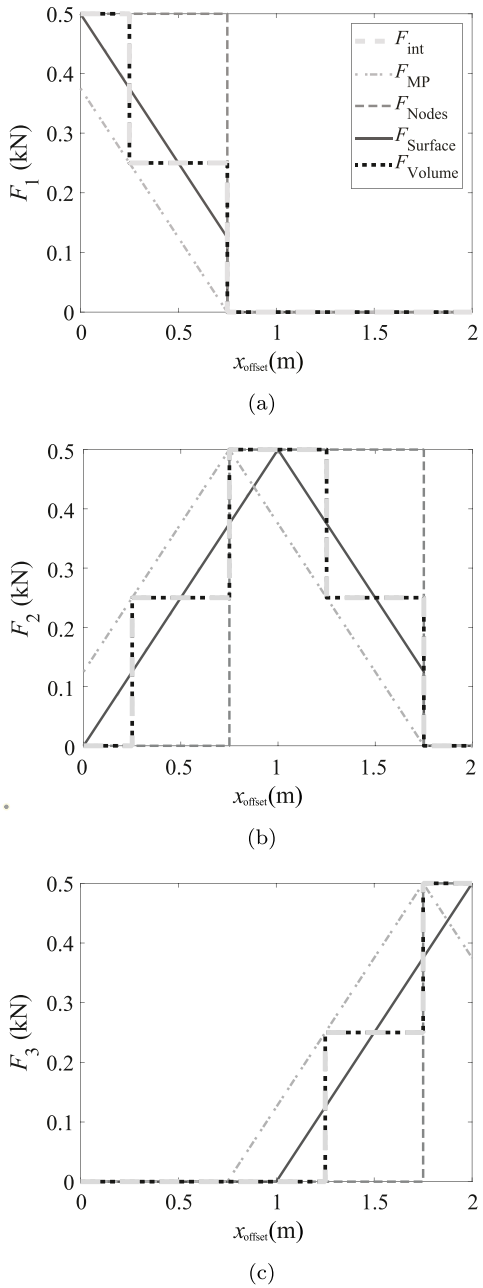



**Fig. 6.** Internal force for standard MPM compared with external force computed by four methods for (a) node 1, (b) node 2, and (c) node 3 of Fig. 5.

Fig. 6 presents $\boldsymbol{F}_{int}$ and $\boldsymbol{F}_{ext}$ at the first three nodes, calculated using standard MPM, for various distances $x_{offset}$. Due to the constant shape function gradients within an element (and the constant stress), $\boldsymbol{F}_{int}$ remains constant until MPs cross element boundaries, after which a jumps occurs. All methods show a qualitatively similar behaviour, i.e. as the boundary moves away from a node $\boldsymbol{F}_{ext}$ decreases, whereas $\boldsymbol{F}_{ext}$ increases as the boundary moves closer to a node. This behaviour is similar to the behaviour of $\boldsymbol{F}_{int}$, and therefore explains the qualitative success of boundary conditions in past MPM research.

However, by looking at the results in more detail some differences are observed. $\boldsymbol{F}_{MP}$ is inconsistent with $\boldsymbol{F}_{int}$ for most $x_{offset}$, since too much load is transferred to the inner nodes. In other words, load is applied inside the material instead of on the material surface. $\boldsymbol{F}_{Nodes}$ is consistent when the boundary element contains 4 MPs, but overestimates $\boldsymbol{F}_{ext}$ compared to $\boldsymbol{F}_{int}$ on the surface nodes after MPs cross element boundaries. $\boldsymbol{F}_{Nodes}$ may present consistent results for small deformation problems, which are often used during code development. $\boldsymbol{F}_{Surface}$ matches $\boldsymbol{F}_{int}$ when $\Gamma$ coincides with the background grid or when $\Gamma$ lies exactly in the middle of the boundary element. However, $\boldsymbol{F}_{Surface}$ and $\boldsymbol{F}_{MP}$ change linearly with $x_{offset}$, while $\boldsymbol{F}_{int}$ remains constant due to the shape function gradients. Finally, $\boldsymbol{F}_{Volume}$ is consistent with $\boldsymbol{F}_{int}$. The position of the surface in this method is dependent on the volume/number of MPs in the outermost active element, and jumps to a new position whenever MPs leave this element. The surface corresponds with the end of the material point discretisation, and therefore distributes $\boldsymbol{F}_{ext}$ in the same manner as the internal load, thereby avoiding stress oscillations.

Note that Neumann boundary conditions are only theoretically correct when applied on the exact material surface (i.e. using $\boldsymbol{F}_{Surface}$). However, the discretisation of MPM causes an incorrect internal force calculation near the boundary, and $\boldsymbol{F}_{Volume}$ is therefore a more consistent BC method for MPM. By increasing the number of material points, or by increasing the number of background grid elements, the difference between $\boldsymbol{F}_{Volume}$ and $\boldsymbol{F}_{Surface}$ will decrease.

Fig. 7 repeats the analysis using GIMP instead of MPM. $\boldsymbol{F}_{int}$ has changed due to the GIMP shape function gradients (see Fig. 2), which influence multiple elements and 'distribute' the MP stresses according to their support domain. Similar to $\boldsymbol{F}_{Volume}$ for MPM, $\boldsymbol{F}_{GIMP}$ computes an $\boldsymbol{F}_{ext}$ consistent with $\boldsymbol{F}_{int}$. Moreover, due to the simple problem geometry, the boundary of the support domain of GIMP coincides with $\Gamma$ such that $\boldsymbol{F}_{GIMP} = \boldsymbol{F}_{Surface}$. $\boldsymbol{F}_{Nodes}$ is less consistent in GIMP compared to MPM, since the internal force changes linearly with $x_{offset}$, while $\boldsymbol{F}_{MP}$ is more consistent due to the linear change. Note that standard MPM shape functions are used to integrate or transfer loads to the nodes. This agrees with the concept of GIMP, since the GIMP shape function for a surface should be created from a 'support domain' with zero width, which returns the standard FEM shape functions.

Figs. 8 and 9 show the effect of the discretisation on the error in the internal force. This error is computed by integrating the absolute difference between the internal and external forces over the offset $x_{offset}$, and dividing by the surface traction integrated over $x_{offset}$, i.e.

$$E_{1D} = \frac{\int |F_{int} - F_{ext}| \, \mathrm{d}x_{offset}}{\int q \, \mathrm{d}x_{offset}} \tag{11}$$

The integration is performed over $x_{offset}$ in the range 0 to 1 m, i.e. the largest element size. The error with respect to element size is computed using 2 MPs per Cartesian direction per element, while the error with

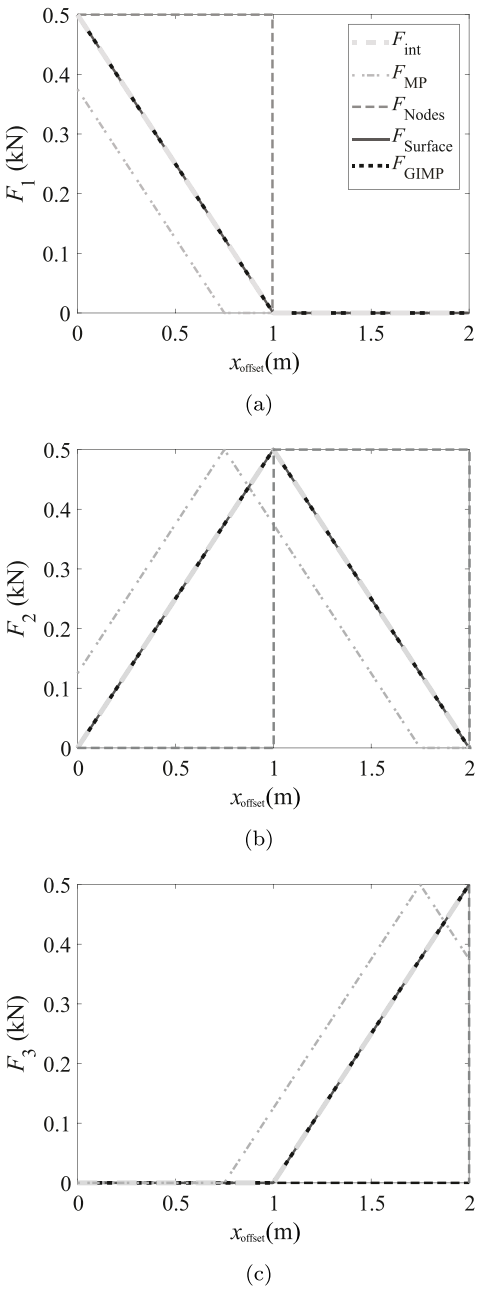



**Fig. 7.** Internal force for GIMP compared with external force computed by four methods for (a) node 1, (b) node 2, and (c) node 3 of Fig. 5. (Note the change from $F_{\text{Volume}}$ to $F_{\text{GIMP}}$ in the legend).

respect to number of MPs is computed using an element size of 1.0 m. As shown by Figs. 8(a) and 9(a), the error does not decrease with a decreasing element size. This is caused by the fact that, although the error at each node decreases, the increase in the number of nodes compensates the total error. Figs. 8(b) and 9(b) show a decrease in error with an increase in the number of MPs for $\boldsymbol{F}_{\text{MP}}$ (with both MPM and GIMP) and $\boldsymbol{F}_{\text{Surface}}$ (with MPM). With a larger number of MPs these methods are more consistent with the internal force calculation. $\boldsymbol{F}_{\text{Volume}}$ (with MPM), $\boldsymbol{F}_{\text{GIMP}}$ (with GIMP) and $\boldsymbol{F}_{\text{Surface}}$ (with GIMP) give no error between the internal and external force calculation for any discretisation, as was expected based on previous results. $\boldsymbol{F}_{\text{Nodes}}$ decreases in accuracy (with MPM) for a higher number of material points, as nodes far from the material are activated for longer with more material points. Due to the assigned domain of the MPs in GIMP, the activated nodes no longer depend on the number of material points, and the error in $\boldsymbol{F}_{\text{Nodes}}$ is therefore constant (and high).

This analysis indicates the importance of developing a (Neumann) BC method which agrees with the discretisation of the material and the adopted shape functions, i.e. the BC method should be consistent with the internal force calculation, which may differ depending on the formulation variant being used. Moreover, it shows another advantage of GIMP compared to MPM, since the material surface coincides with the optimal position for BCs in one dimension. Therefore, GIMP is adopted for the remainder of this paper.

## 4. Application in two dimensions

A bar with an inclined surface (angle $\theta$) is used to analyse the effect of a surface unaligned with the background grid, see Fig. 10(a). A traction $\boldsymbol{q}$ is applied normal to the bar's surfaces, i.e. normal to the inclined, horizontal and vertical surfaces. The bar is discretised using MPs with a 0.5 m square support domain, which are aligned with the background grid in the vertical direction, and located in the horizontal direction such that the support domains align with the sloped surface (see Fig. 10(b)). The slope cuts the middle of the left edge of the support domains as shown in Fig. 10(b), for all angles $\theta$. The background grid is offset by a distance $x_{\text{offset}}$ from the tip of the bar, and consists of 1 m square elements.

The analytical solution for the internal stress at all positions in the bar is $q$ in both the vertical and horizontal directions, i.e. the bar is in equilibrium with the applied external force. Once again, the initial stresses are set to this equilibrium condition. The calculated internal forces at the boundary nodes are compared with the external forces computed using the methods proposed in the previous sections, and should be equal for a consistent BC method. The normalised differences in the horizontal and vertical forces, $E_{F_x}$ and $E_{F_y}$, respectively, are defined as

$$E_{F_x} = \frac{\sum_i |F^{\text{ext}}_{x,i} - F^{\text{int}}_{x,i}|}{3q} \tag{12}$$

$$E_{F_y} = \frac{\sum_i |F^{\text{ext}}_{y,i} - F^{\text{int}}_{y,i}|}{3q \tan(\theta)} \tag{13}$$

where $F^{\text{ext}}_{j,i}$ and $F^{\text{int}}_{j,i}$ are the external and internal forces, respectively, of node $i$ in direction $j$. $E_{F_x}$ and $E_{F_y}$ express the difference between $\boldsymbol{F}_{\text{ext}}$ and $\boldsymbol{F}_{\text{int}}$ summed over all nodes, decomposed in the horizontal and vertical directions. $E_{F_x}$ and $E_{F_y}$ are normalised with respect to the theoretical external force acting along the sloped surface in the horizontal and vertical directions, respectively. Due to the geometry of the problem, the theoretical horizontal external force is independent of the slope angle, whereas the theoretical vertical external force depends on the slope angle. The denominators normalise the differences for the slope angle.

The computed differences with the four methods for $\tan(\theta) = 0.5$, i.e. the slope shown in Fig. 10(b), are shown in Fig. 11. The figure presents the normalised differences $E_{F_x}$ and $E_{F_y}$ for different offsets ($x_{\text{offset}}$) of the grid. To clarify, $\boldsymbol{F}_{\text{Surface}}$ is computed using the sloped surface of the actual bar, while $\boldsymbol{F}_{\text{GIMP}}$ is computed by the boundary of the GIMP domains constructed from vertical and horizontal segments (see Fig. 10(b)). For clarity, Figs. 11 and 12 are limited between 0 m $< x_{\text{offset}} <$ 1 m, i.e. one grid cell, since $E_{F_x}$ and $E_{F_y}$ are periodic with respect to the offset of the grid with a period of one grid cell. This period is independent of the slope angle.

Similar to the 1D case, $\boldsymbol{F}_{\text{Nodes}}$ presents the largest inconsistencies (oscillating around a normalised total difference of 100%), due to the fact that loads are applied at large distances from the domain (in some locations) for all values of $x_{\text{offset}}$. Whereas $\boldsymbol{F}_{\text{Nodes}}$ applies loads too far outside of the material, $\boldsymbol{F}_{\text{MP}}$ instead applies loads too far into the material. However, the differences for $\boldsymbol{F}_{\text{MP}}$ are smaller than the differences for $\boldsymbol{F}_{\text{Nodes}}$ and more or less constant: i.e. differences of

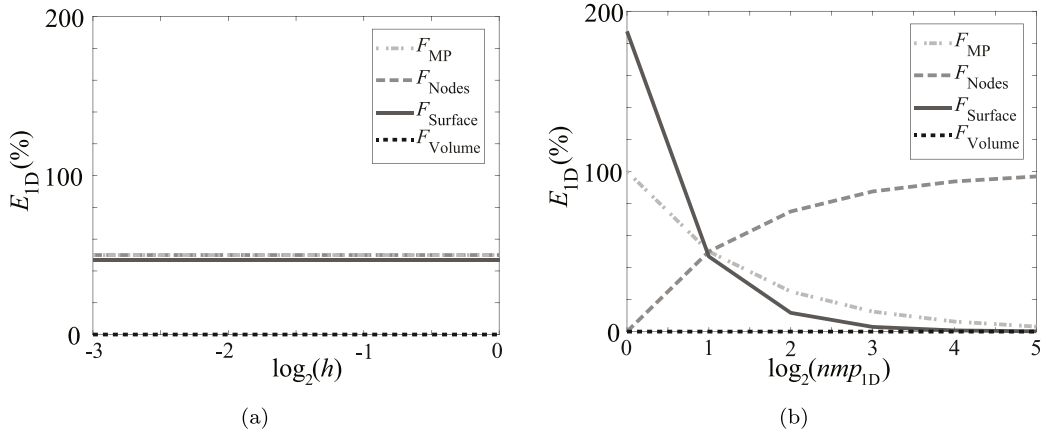



**Fig. 8.** Dependency of the error in the external force calculation, for standard MPM with four methods, on (a) the element size and (b) the number of material points per cell in each direction.

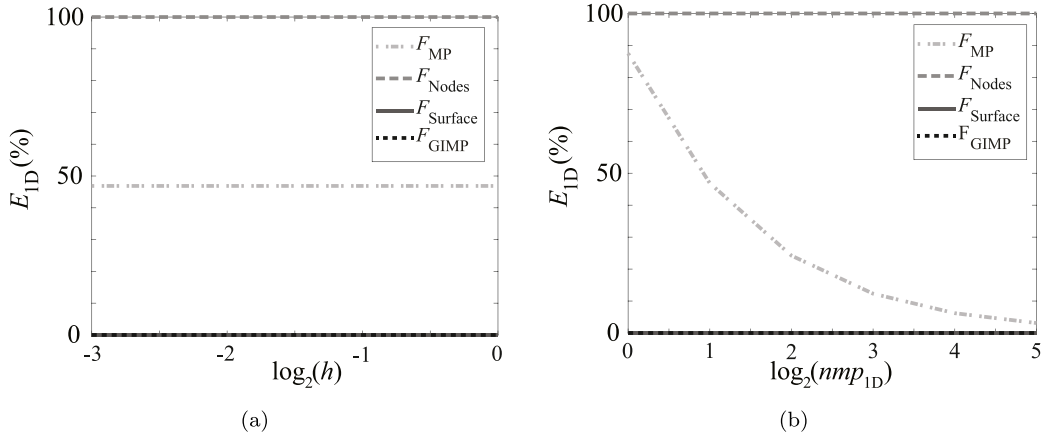



**Fig. 9.** Dependency of the error in the external force calculation, for GIMP with four methods, on (a) the element size and (b) the number of material points per cell in each direction.

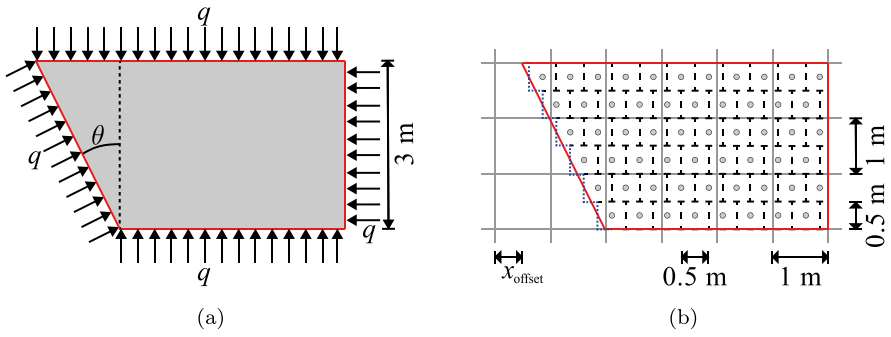



**Fig. 10.** (a) A bar with a surface inclined at an angle $\theta$, compressed from all sides by a surface traction $q$. (b) The bar is discretised into MPs (grey filled circles) with a square support domain (dashed lines), and a background grid (solid lines) constructed from 1 m square elements which is offset by a distance $x_{\text{offset}}$ with respect to the tip of the bar. The surfaces used to compute $\boldsymbol{F}_{\text{Surface}}$ and $\boldsymbol{F}_{\text{GIMP}}$ are indicated by the solid red line and dotted blue line, respectively.

approximately 50% in the $x$- and $y$-directions. The two methods which demonstrated an exact application of the BC in one dimension, $\boldsymbol{F}_{\text{Surface}}$ and $\boldsymbol{F}_{\text{GIMP}}$, also perform well in two dimensions. $\boldsymbol{F}_{\text{Surface}}$ has an average inconsistency of 2.1% and 1.3% in the $x$- and $y$-directions respectively, while $\boldsymbol{F}_{\text{GIMP}}$ has zero error, i.e. is consistent.

Fig. 12 further investigates the difference between $\boldsymbol{F}_{\text{Surface}}$ and $\boldsymbol{F}_{\text{GIMP}}$, for several slope angles $\theta$. $\boldsymbol{F}_{\text{GIMP}}$ is seen to be an error free solution for all slope angles, while for $\boldsymbol{F}_{\text{Surface}}$ the error grows with a larger slope angle (it would then decrease with further increase in slope angle above 45°), yet remains reasonably low. The larger errors are caused by a growing difference between the actual surface and the discretisation of GIMP. Moreover, when the slope angle is close to 45°, the surface is more often partially located in inactive elements, reducing the applied total load and increasing the error. Note that, even though the error of $\boldsymbol{F}_{\text{Surface}}$ grows with the slope angle (up to $\theta = 45°$), the error is still significantly smaller than for $\boldsymbol{F}_{\text{MP}}$ and $\boldsymbol{F}_{\text{Nodes}}$.

The effect of element and material point discretisations have been investigated in 2D. Minor differences have been observed compared to the 1D discretisation case, but the conclusions are consistent. Separate figures are therefore not presented here. Increasing the number of

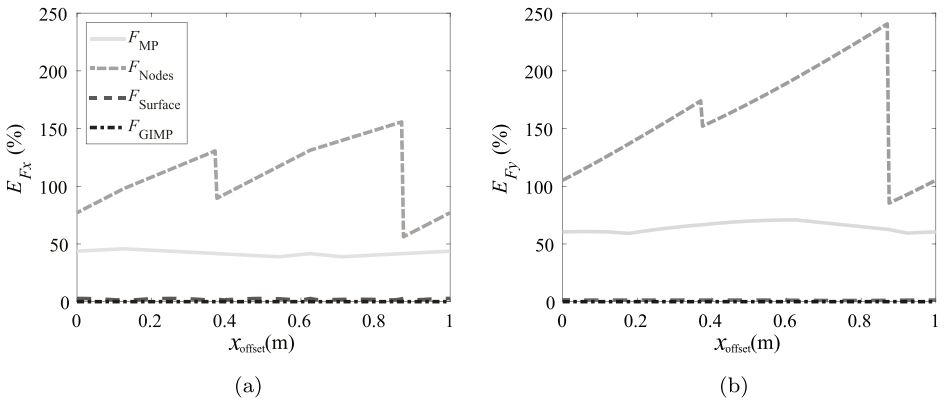



**Fig. 11.** Normalised difference between $\boldsymbol{F}_{\text{ext}}$ and $\boldsymbol{F}_{\text{int}}$ for an inclined bar with $\tan(\theta) = 0.5$ with different boundary condition methods, (a) $E_{Fx}$, (b) $E_{Fy}$.

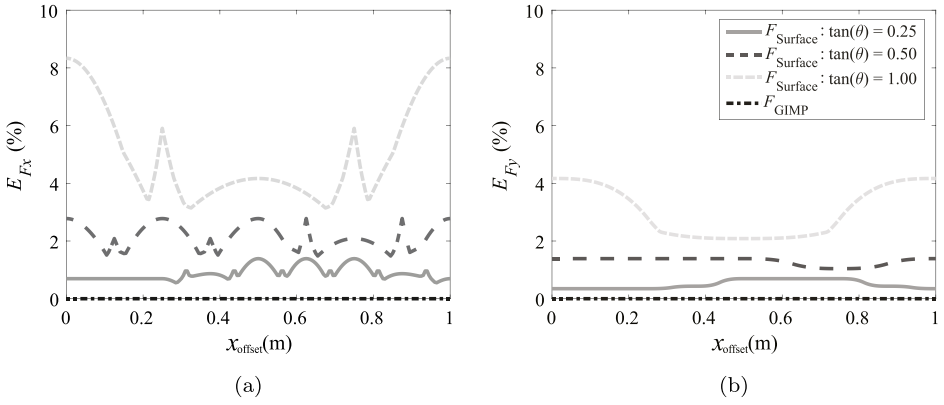



**Fig. 12.** Normalised difference between $\boldsymbol{F}_{\text{ext}}$ and $\boldsymbol{F}_{\text{int}}$ for an inclined bar with various slope angles using $\boldsymbol{F}_{\text{Surface}}$ and $\boldsymbol{F}_{\text{GIMP}}$, (a) $E_{Fx}$, (b) $E_{Fy}$.

material points improves the computation of $\boldsymbol{F}_{\text{MP}}$, while the error remains approximately constant for the other methods. The error is zero for $\boldsymbol{F}_{\text{GIMP}}$ as was also observed in Figs. 11 and 12.

## 5. Boundary detection

Sections 3 and 4 show that, in GIMP, $\boldsymbol{F}_{\text{Surface}}$ and $\boldsymbol{F}_{\text{GIMP}}$ are (almost) consistent with the internal force. Sections 3 and 4 involved a fixed geometry, where the material surface could be pre-defined. However, in an analysis where MPs move with respect to the background grid, and also with respect to each other, the material surface should be detected automatically. Algorithms are therefore needed to construct the boundary required for the consistent methods, i.e. boundary detection methods for $\boldsymbol{F}_{\text{Surface}}$ and $\boldsymbol{F}_{\text{GIMP}}$. Remmerswaal (2017) investigated various approaches to construct the material surface for $\boldsymbol{F}_{\text{Surface}}$ in MPM, and showed that the Proximity Field Method (PFM), which constructs linear segments within grid cells based on the level set method, performed well. A brief explanation of PFM is provided. In addition, this paper expands a contour algorithm for iso-rectangles to construct the surface for $\boldsymbol{F}_{\text{GIMP}}$ (Prusinkiewicz and Raghavan, 1985).

### 5.1. Proximity field method

The Proximity Field Method (PFM) (Remmerswaal, 2017) is based on the level set method, where the boundary location is defined as the (zero) level set of an auxiliary field (Sethian, 1996), and can be used to compute $\boldsymbol{F}_{\text{Surface}}$. Epanechnikov kernel functions are used to compute the proximity, i.e. the distance, to nearby MPs. In other words, the kernel function represents the influence domain of the MP. A local coordinate system is used for the kernel function (see Fig. 13), i.e.

$$K_{i,j} = C_j(1 - \boldsymbol{u}_{i,j}^{\mathrm{T}}\boldsymbol{u}_{i,j}) \tag{14}$$

where $K_{i,j}$ is the kernel function of MP $j$ evaluated at point $i$, $\boldsymbol{u}_{i,j}$ is a distance in local coordinates and $C_j$ is a constant used to control the magnitude of the kernel function. The local distance is computed as

$$\boldsymbol{u}_{i,j} = \mathbf{S}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j) \tag{15}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the global coordinates of point $i$ and MP $j$, and $\mathbf{S}_j$ is the shape matrix given by

$$\mathbf{S}_j = \begin{bmatrix} a_j & 0 \\ 0 & b_j \end{bmatrix} \tag{16}$$

in which $a_j$ and $b_j$ define the size of the kernel function.

The constant $C_j$ has here been defined as

$$C_j = \frac{2}{\pi \det(S_j)} \tag{17}$$

This ensures that the volume under each kernel function is equal to 1, i.e. each MP distributes the same total influence independent of its size. A $C_j$ correlated with MP mass can be used when large differences in MP mass occur within the domain.

Fig. 14(a) shows how the proximity field (PF) is constructed (for a 1D example) by summing up the kernel functions of all MPs. The PF is computed at the nodes and the boundary points are found by comparing the PF with a user specified threshold (see Fig. 14(b)). In 1 dimension, the boundary points denote the 2 ends of each 1D domain as illustrated for the 2 1D domains in Fig. 14(b), and connection of the points is therefore not required. In 2 dimensions, the complete

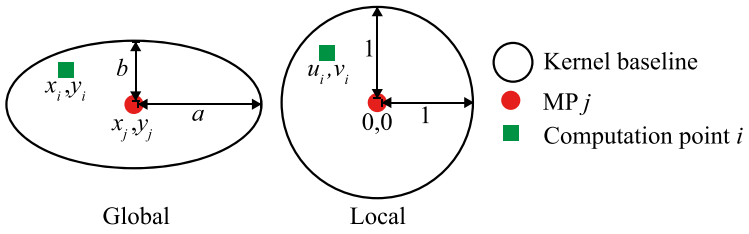



**Fig. 13.** Conversion of a global elliptical kernel baseline to a local circular kernel baseline around MP $j$. A point for the computation of the kernel function is indicatively shown in the global and local coordinate systems.

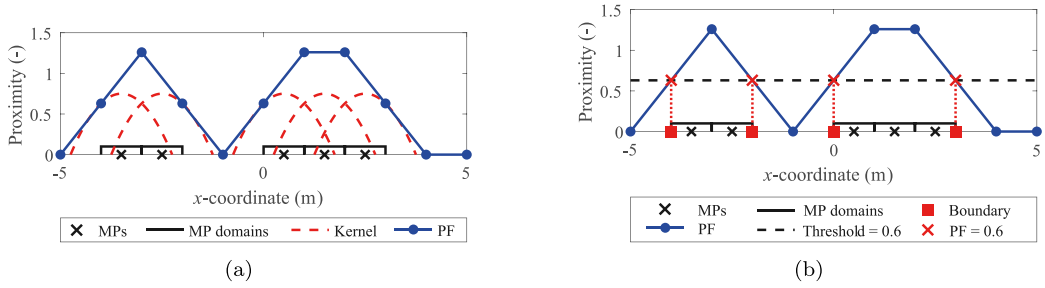



**Fig. 14.** (a) 1D example of constructing a PF using kernel functions; (b) 1D example of boundary detection by comparing the PF with a threshold.

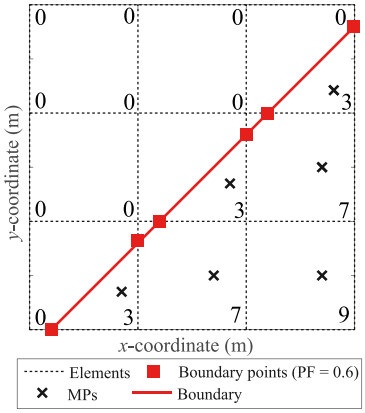



**Fig. 15.** Computation of boundary points from a proximity field in two dimensions. Proximity field values displayed at the nodes. Boundary constructed by connecting the boundary points with linear segments.

boundary is created by connecting the points using Composite Bézier Curves or B-splines as shown in Fig. 15. Here linear segments are used. The surface is then used to compute $\boldsymbol{F}_{\text{Surface}}$ (see Section 2.2).

### 5.2. Contour algorithm for iso-rectangles

The boundary can also be constructed by merging all the GIMP support domains, such that $\boldsymbol{F}_{\text{GIMP}}$ can be computed. This merging is similar to finding the contour of a set of iso-rectangles (Prusinkiewicz and Raghavan, 1985), i.e. domains aligned in two directions (the vertical and horizontal directions). For example, 5 domains are shown in Fig. 16(a), although, in practice, any number of domains can be merged.

Prusinkiewicz and Raghavan (1985) designed an algorithm to find the contour of a set of iso-rectangles, by converting the rectangles from Cartesian space (Fig. 16(a)) to a so-called slab space representing only the topology (Fig. 16(b)). In slab space, edges with the same horizontal or vertical coordinate are separated. For example, in Fig. 16 the bottom edge of rectangle 3 and the top edge of rectangle 4 have the same vertical Cartesian coordinate, but they have a separate slab-space position. Lines with the same coordinate are separated such that left edges precede right edges and bottom edges precede top edges. Two edges with the same coordinate which are on the same side of their rectangles, for example the bottom edges of rectangles 2 and 5 in Fig. 16, are separated according to the lowest MP number. Prusinkiewicz and Raghavan (1985) used the slab space to find the horizontal and vertical edges belonging to the contour. These edges in slab space also result in the correct contour in real space, as shown in Fig. 16.

Due to the movement of MPs, and their support domains, small gaps between domains can appear. As the support domains represent large continuous sections of material, instead of individual material particles, the small gaps between the support domains do not represent the behaviour of the material accurately. If external loads are applied within a small gap, the gap can grow larger causing unrealistic behaviour. It is therefore preferred to fill any gaps in the boundary that are smaller than a minimum gap width, i.e. gaps with a width or height smaller than a specified minimum width ($W_{min}$) or height ($H_{min}$), respectively. When a gap is larger than this specified minimum width it is assumed to be a gap occurring in the material.

In order to fill the small gaps, the algorithm presented by Prusinkiewicz and Raghavan (1985) is here extended. Specifically, the horizontal edges of the rectangles are extended to the left and right by $W_{min}$, see Fig. 17. Then, for each slab $i$ the horizontal edges belonging to the contour are determined. Each slab $i$ of the slab space is divided into segments (numbered from bottom to top) by the horizontal edges intersecting $i$. For example, slabs 10 and 11 of Fig. 17(b) are divided into eleven segments as shown in Fig. 18. Prusinkiewicz and Raghavan (1985) counted the number of rectangles overlapping each segment using a so-called invisibility number. Here multiple invisibility numbers are used (see Fig. 18(a)). Invisibility number $J$ is an integer which counts how many rectangles overlap a specific segment, and can be used to determine if a segment is inside any rectangles ($J > 0$) or outside all rectangles ($J = 0$). $J$ is computed for segment $k$ in slab $i$ based on the previous segment $k-1$ in slab $i$ and the edge $m$ separating the segments $k$ and $k-1$:

$$J(k) = \begin{cases} 0 & \text{if } k = 0, \\ J(k-1)+1 & \text{if } m \text{ is a real bottom edge} \\ J(k-1)-1 & \text{if } m \text{ is a real top edge} \\ J(k-1) & \text{otherwise.} \end{cases} \tag{18}$$

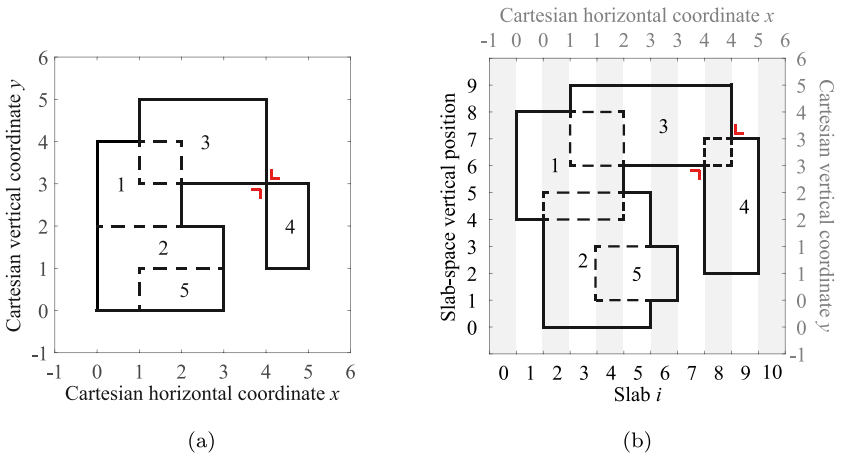



**Fig. 16.** Conversion from real space (a) to slab space (b) for a set of 5 rectangles according to the algorithm designed by Prusinkiewicz and Raghavan (1985). The contour is marked by the solid line and the internal edges are marked by the dashed lines. The slab space solves the ambiguity for the connection of rectangles 3 and 4, and the red right angle corners mark this connection.

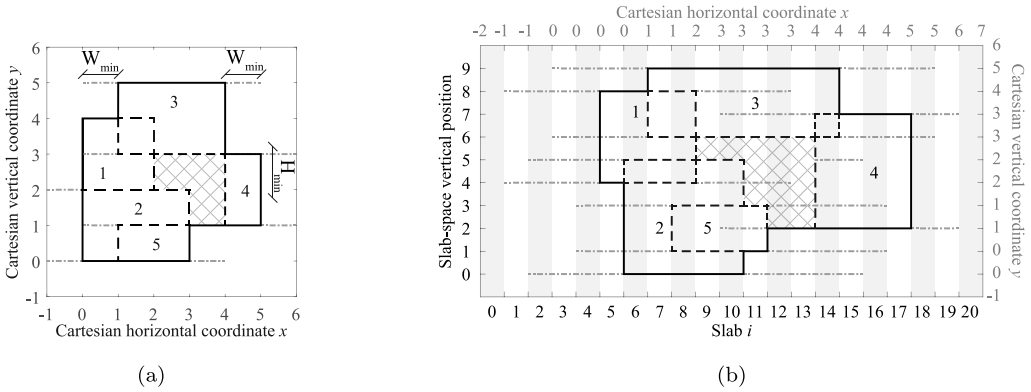



**Fig. 17.** Conversion from real space (a) to slab space (b) for a set of 5 rectangles according to new gap removal algorithm. Grey horizontal dashed-dotted lines mark the extended horizontal edges due to the minimum width ($W_{min}$). The contour is marked by the solid line and the internal edges are marked by the dashed lines, and the cross hatching highlights the removed gap.

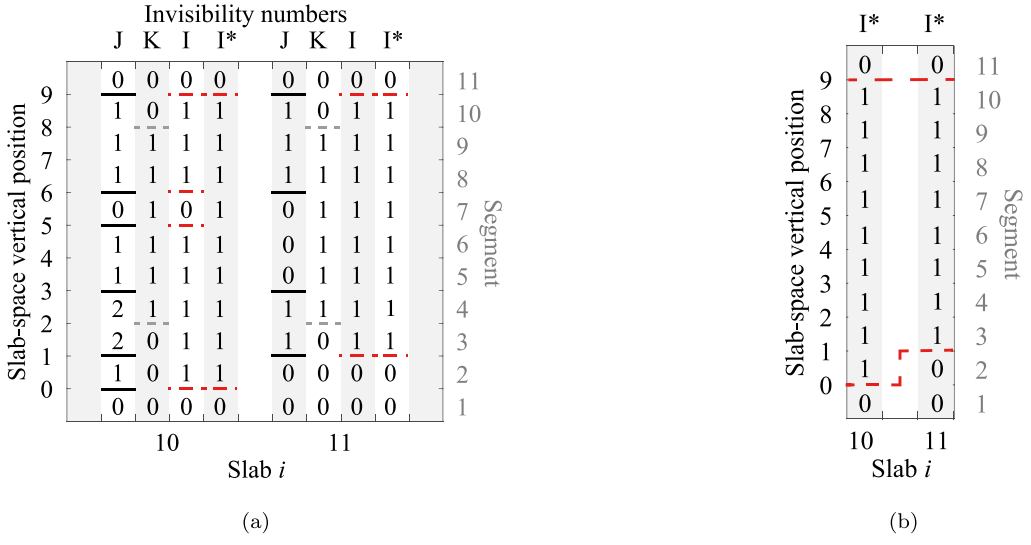



**Fig. 18.** (a) Computation of the 3 invisibility numbers ($I$, $J$ and $K$) together with the associated edges for slabs 10 and 11 of Fig. 17(b) ($I^*$ indicates the invisibility number after vertical gap removal). (b) The required connection between the contour edges found in slabs 10 and 11.

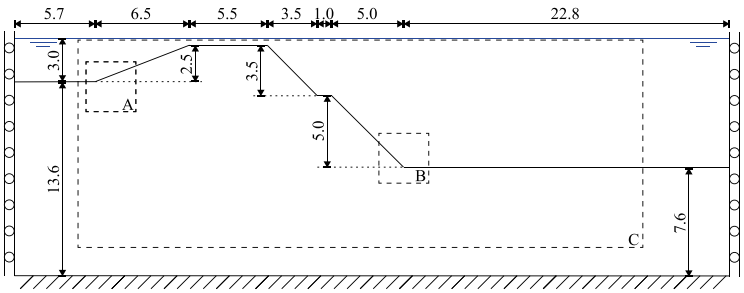



**Fig. 19.** Submerged slope example problem (dimensions shown in metres).

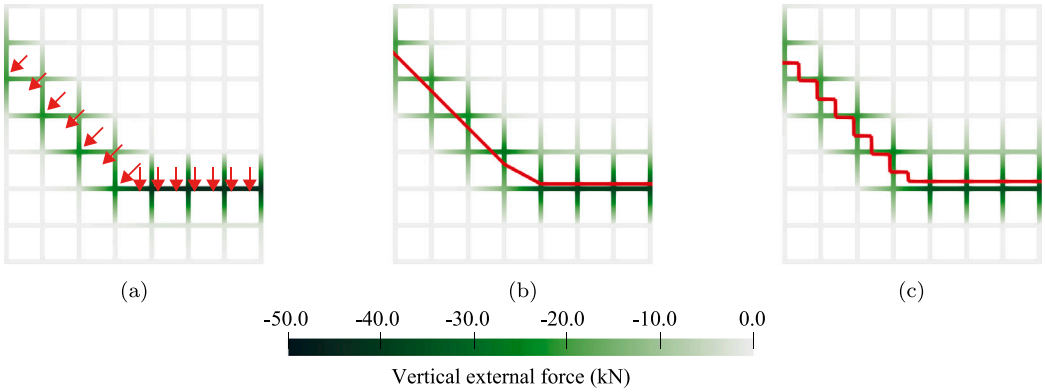



**Fig. 20.** Vertical external nodal loads (kN) coloured on the background grid at location B, computed using (a) point loads applied at MPs, (b) loads applied on a PFM surface, and (c) loads applied on a surface constructed from GIMP domains. Point loads, PFM surface and GIMP support domain surface are indicatively shown in red.

Similarly, invisibility number $K$ counts the number of left extensions of rectangles overlapping a segment $k$, and is computed using a similar procedure as $J$:

$$K(k) = \begin{cases} 0 & \text{if } k = 0, \\ K(k-1)+1 & \text{if } m \text{ is a left extension of a bottom edge} \\ K(k-1)-1 & \text{if } m \text{ is a left extension of a top edge} \\ K(k-1) & \text{otherwise.} \end{cases} \tag{19}$$

$J$ and $K$ are combined to form the invisibility number $I$:

$$I(k) = \begin{cases} 1 & \text{if } J(K) > 0, \\ 1 & \text{if } K(K) > 0 \text{ and } k \text{ was inside the contour in slab } l-1, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

A segment $k$ is inside the contour if $I(k) = 1$ and outside when $I(k) = 0$. In other words, by combining $J$ and $K$, $I$ ensures that all rectangles are included in the final contour and horizontal gaps have been removed. To remove small vertical gaps the total height ($H_{total}$) of consecutive segments with $I = 0$ is computed. $I^*$ is increased to 1 in the segments where $H_{total} <= H_{min}$.

The horizontal edges belonging to the final contour are those edges which separate segments with $I^* = 0$ from segments with $I^* = 1$. The vertical edges of the contour can be determined by comparing the horizontal edges of the contour from two consecutive slabs (Prusinkiewicz and Raghavan, 1985), see Fig. 18(b). The final contour is output as a sequence of contour edges. The coordinates of the contour polygon can be back-calculated from this sequence and the MP properties. More details on connecting relevant edges, the output and the space/time efficiency of the algorithm can be found in Prusinkiewicz and Raghavan (1985).

## 6. Submerged slope failure example

The applicability of the boundary detection methods for the most consistent Neumann BC techniques are evaluated for a submerged slope failure problem. The problem involves a two dimensional submerged clay slope, shown in Fig. 19, which is unstable under its own weight. The material has a unit weight of 20 kN/m$^3$, and the elastic deformation is governed by a Young's modulus of 1000 kPa and a Poisson's ratio of 0.45. The numerical model uses a Von-Mises softening model (Wang et al., 2016a). The initial undrained shear strength ($c_0$) is 5.4 kPa at the ground surface and increases linearly with depth by 3.0 kPa every meter (giving a maximum of 54 kPa at the base of the domain immediately under the slope crest). The residual undrained shear strength ($c_r$) is equal to 0.5 $c_0$, and the softening modulus is equal to −10 kPa. The background grid uses 0.5 m square elements, while the slope is discretised using MPs with rectangular support domains of varying sizes. The sizes vary such that (1) roughly 4 MPs are placed within each filled element, and (2) the slope is represented accurately, meaning that sloped surfaces cut the GIMP support domains in a similar manner to Fig. 10(b). The simulation uses implicit time integration, as originally presented by Charlton et al. (2017), with a timestep size of 0.01 s.

The hydrostatic pressures acting on the slope are converted to nodal loads computed using either equivalent point loads at the surface MPs, traction applied on a PFM surface, or traction applied on a GIMP domain surface. In other words, the most consistent methods for GIMP are compared against the application of loads at surface MPs, a technique often used in the literature. The point loads are presented as vectors in Figs. 20–22, while the PFM surface and GIMP domain surface are plotted as solid lines.

The initial stresses are computed with a quasi-static load step, in which the MPs are fixed in place. Oscillations in initial stress, caused by the different external load methods, appear only locally at the

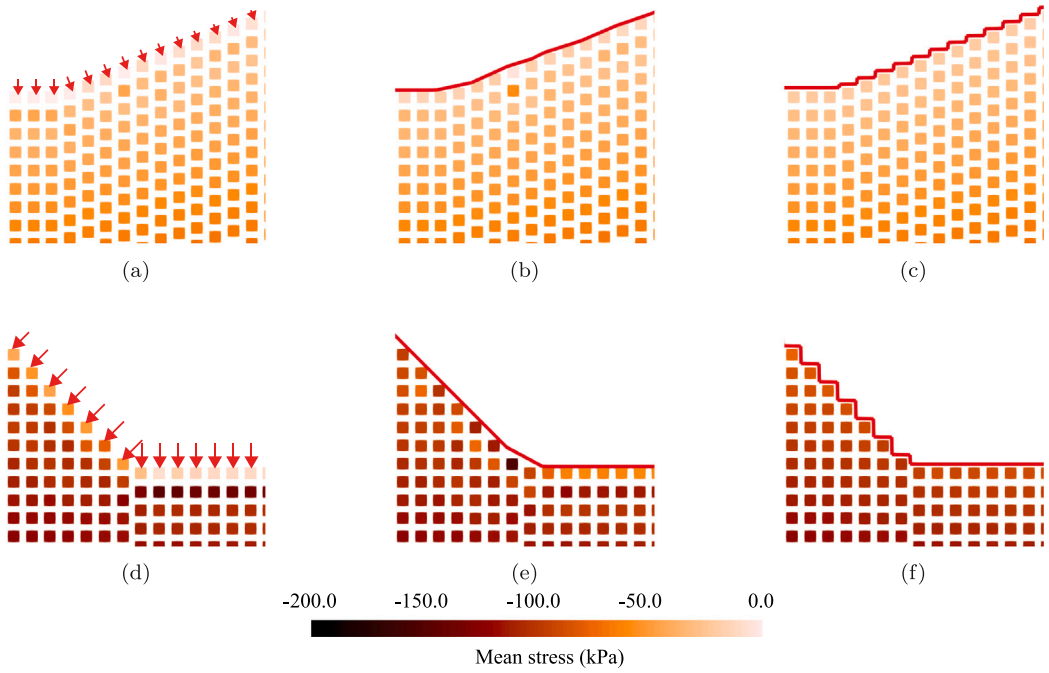



**Fig. 21.** Mean total stresses at the MPs (kPa) at location A (a–c) and location B (d–f), computed using (a, d) point loads applied at MPs, (b, e) loads applied on a PFM surface, and (c, f) loads applied on a surface constructed from GIMP domains. Point loads, PFM surface and GIMP support domain surface are indicatively shown.

surface. As these differences are difficult to observe, locations A and B, as shown in Fig. 19, are investigated in detail. Fig. 20 shows the vertical external forces computed with the three techniques at location B. The point loads cause higher nodal external forces inside the material (see Fig. 20(a)) compared to the other two methods. This results in lower mean stresses, as shown in Fig. 21, for the surface MPs, and, to compensate, higher stresses for the second layer of MPs compared to the other techniques. Similar stress errors are observed at the surface at location A (see Fig. 21(a)). Loading the PFM surface (Figs. 21(b) and 21(e)) reduces the mean stress error compared to the point loads, but still causes non-smooth stresses at the sloping faces, as was also observed in Section 4. Moreover, oscillations occur at the flat surface in Fig. 21(e) as the detected surface is a bit too close to the MPs. Finally, loading the GIMP support domain surface gives accurate (and smooth) initial stresses (see Figs. 21(c) and 21(f)).

The deviatoric stresses, investigated at the same locations, are less clearly affected by the placement of the loads (Fig. 22). This might be due to the mean stress nature of a hydrostatic pressure condition. Small deviations can only be observed along the gentle slope (Figs. 22(a) and 22(b)), while deviatoric stress errors appear negligible for the 45 degree slope (Figs. 22(d) and 22(e)). Similar to the mean stress, loading the GIMP support domain surfaces leads to no observable deviatoric stress errors (Figs. 22(c)-22(f)).

In Fig. 23 the failure process is shown as computed by the three methods. Softening can be observed along the failure surface for all three methods. The failure develops much faster when point loads are used compared to the other two approaches, i.e. the different BC methods can lead to differences in the failure process. Moreover, the surface MPs at which the point loads are applied can be pushed downwards (Fig. 23(b)), and, as the list of surface MPs remains fixed, the point loads can be pushed into the material. A gap in the soil appears, as seen at 4.5 s (Fig. 23(c)), and leads to a simulation crash.

The other two techniques present similar results, both during the simulation (Figs. 23(d), 23(e), 23(g) and 23(h)) and at the end of the simulation (Figs. 23(f) and 23(i)). The results are similar because the constitutive model is based on deviatoric stress, and the differences in deviatoric stress are small (as shown in Fig. 22). However, minor differences can be observed. For example, the failure after 7.0 s is developed slightly further when the PFM surface is used compared to when the GIMP domains are used. Moreover, if mean stress dependent constitutive models would be used, larger differences are expected. Note that the results described here are mainly dependent on the location of the boundary condition and not on the method used to determine its location. In other words, should other methods than those presented in Section 5 be used to construct the boundary or the contour of GIMP support domains, similar results would be expected given that the change in boundary location is small.

## 7. Conclusion

The application of traction (Neumann, or external load) boundary conditions should show consistency between the calculation of internal and external forces in order to provide accurate simulation results. In several of the possible methods examined here, and methods commonly presented in the literature, this is shown not to occur. In theory, Neumann boundary conditions should be applied on the exact material surface. However, in practice due to the MPM discretisation, the internal force may not be consistent with the application of a Neumann boundary condition on the exact material surface. These inconsistencies can lead to an incorrect (surface) stress distribution and influence the outcome of simulations. In the original MPM, the volume of material considered to be within a grid cell directly relates to the number of MPs within that grid cell, and, as such, the internal force jumps in value when MPs leave the element. Therefore, the BC application should also have this feature, although it is difficult to practically implement for arrangements where the material points are not well aligned with the background grid. Improving the discretisation, for example by using more material points or a different integration scheme such as GIMP, will reduce the difference between the consistent and exact boundary conditions. In GIMP, due to the MPs having their own domain, the material gradually leaves the grid cell as the MPs move, and the BC application should follow this behaviour. Therefore, a consistent

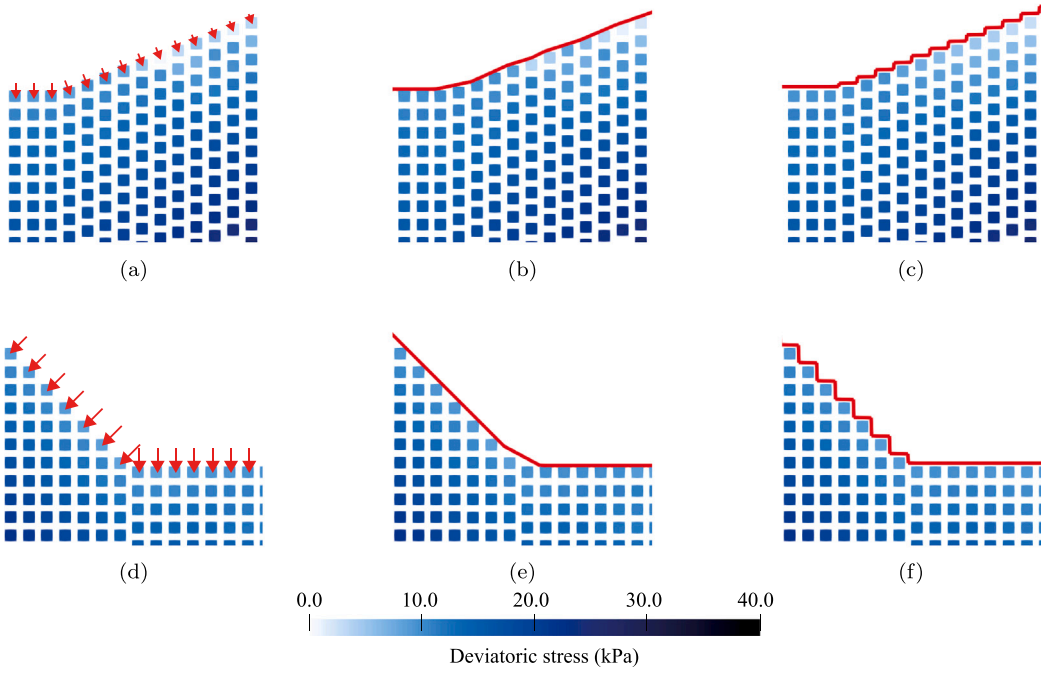



**Fig. 22.** Deviatoric stresses at the MPs (kPa) at location A (a-c) and location B (d-f), computed using (a, d) point loads applied at MPs, (b, e) loads applied on the PFM surface, and (c, f) loads applied on the surface constructed from GIMP domains. Point loads, PFM surface and GIMP support domain surface are indicatively shown.

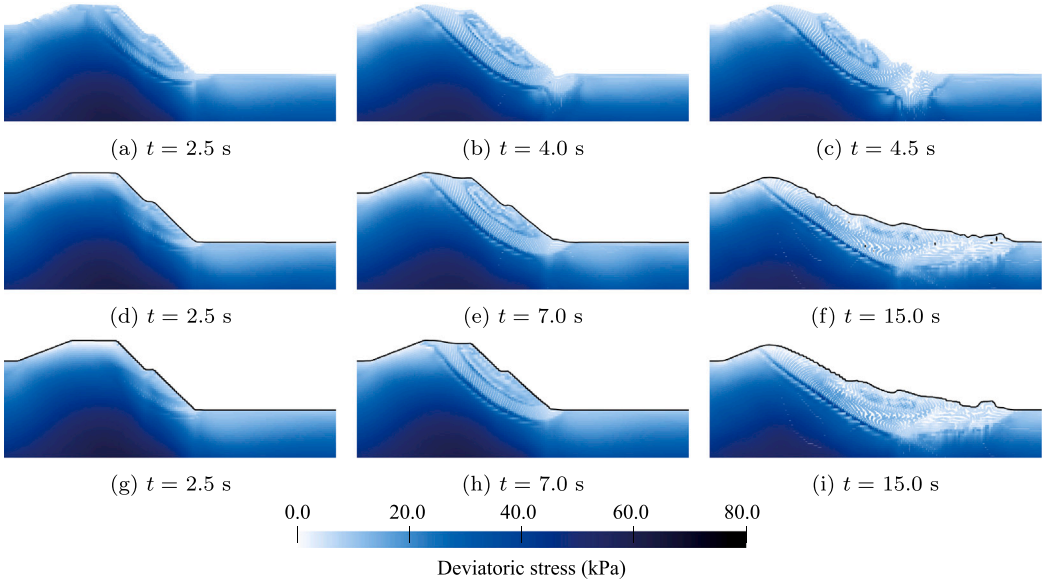



**Fig. 23.** Displacement of MPs with external loads computed using: (a–c) point loads at MPs, (d–f) loads applied on PFM surface, and (g–i) loads applied on GIMP domain surface. MPs coloured according to the deviatoric stress (kPa). PFM and GIMP domain surfaces plotted in black.

method involves applying loads on the material point support domain. The difference between this consistent method and applying loads on the exact material surface is small in GIMP. Note that the boundary condition scheme or aspects of it are likely to be useful in other MPM variants and meshless methods. For such methods, the boundary condition schemes should be similarly developed to follow the behaviour of the internal force discretisation.

Two (almost) consistent Neumann BC methods have been developed for GIMP, together with the required boundary detection methods. These techniques are generally applicable and significantly improve the calculated stresses of material points close to the surface, and have been shown to improve the failure process of a submerged slope. This removes one cause of stress oscillations regularly observed in MPM-type methods. Note that the computational efficiency of the boundary detection methods was not investigated in detail and optimisations could be developed. Also, the methods have here been developed for 2D problems, but there do not seem to be any barriers to extending the approach to 3D problems.

## CRediT authorship contribution statement

**Guido Remmerswaal:** Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Philip J. Vardon:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Michael A. Hicks:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

This work is part of the research programme AllRisk which is financed by the Netherlands Organisation for Scientific Research (NWO), project number P15-21 Project 4.

## References

Bardenhagen, S.G., Kober, E.M., 2004. The generalized interpolation material point method. CMES - Comput. Model. Eng. Sci. 5 (6), 477–495. http://dx.doi.org/10.3970/cmes.2004.005.477.

Bathe, K.J., 2014. Finite Element Procedures, Second ed. K.J. Bathe.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P., 1996. Meshless methods: An overview and recent developments. Comput. Methods Appl. Mech. Engrg. 139 (1–4), 3–47. http://dx.doi.org/10.1016/S0045-7825(96)01078-X.

Beuth, L., 2012. Formulation and Application of a Quasi-Static Material Point Method (Ph.D. thesis). University of Struttgart, Germany.

Bing, Y., Cortis, M., Charlton, T.J., Coombs, W.M., Augarde, C.E., 2019. B-spline based boundary conditions in the material point method. Comput. Struct. 212, 257–274. http://dx.doi.org/10.1016/j.compstruc.2018.11.003.

Charlton, T.J., Coombs, W.M., Augarde, C.E., 2017. iGIMP: An implicit generalised interpolation material point method for large deformations. Comput. Struct. 190, 108–125. http://dx.doi.org/10.1016/j.compstruc.2017.05.004.

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. http://dx.doi.org/10.1016/S0013-7944(02)00066-8.

Colagrossi, A., Landrini, M., 2003. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. J. Comput. Phys. 191 (2), 448–475. http://dx.doi.org/10.1016/S0021-9991(03)00324-3.

Cortis, M., Coombs, W., Augarde, C., Brown, M., Brennan, A., Robinson, S., 2018. Imposition of essential boundary conditions in the material point method. Internat. J. Numer. Methods Engrg. 113 (1), 130–152. http://dx.doi.org/10.1002/nme.5606.

de Koster, P., Tielen, R., Wobbes, E., Möller, M., 2021. Extension of B-spline material point method for unstructured triangular grids using Powell–Sabin splines. Comput. Part. Mech. 8 (2), 273–288. http://dx.doi.org/10.1007/s40571-020-00328-3.

Federico, I., Marrone, S., Colagrossi, A., Aristodemo, F., Antuono, M., 2012. Simulating 2D open-channel flows through an SPH model. Eur. J. Mech. B Fluids 34, 35–46. http://dx.doi.org/10.1016/j.euromechflu.2012.02.002.

Fern, E.J., Rohe, A., Soga, K., Alonso, E.E., 2019. The Material Point Method for Geotechnical Engineering. CRC Press, Taylor & Francis Group, http://dx.doi.org/10.1201/9780429028090.

Fernández-Méndez, S., Huerta, A., 2004. Imposing essential boundary conditions in mesh-free methods. Comput. Methods Appl. Mech. Engrg. 193 (12–14), 1257–1275. http://dx.doi.org/10.1016/j.cma.2003.12.019.

González Acosta, J.L., Vardon, P.J., Hicks, M.A., 2021. Study of landslides and soil-structure interaction problems using the implicit material point method. Eng. Geol. 285, 106043. http://dx.doi.org/10.1016/j.enggeo.2021.106043.

González Acosta, J.L., Vardon, P.J., Remmerswaal, G., Hicks, M.A., 2020. An investigation of stress inaccuracies and proposed solution in the material point method. Comput. Mech. 65, 555–581. http://dx.doi.org/10.1007/s00466-019-01783-3.

Griffiths, D.V., Lane, P.A., 1999. Slope stability analysis by finite elements. Géotechnique 49 (3), 387–403. http://dx.doi.org/10.1680/geot.1999.49.3.387.

Hicks, M.A., Onisiphorou, C., 2005. Stochastic evaluation of static liquefaction in a predominantly dilative sand fill. Géotechnique 55 (2), 123–133. http://dx.doi.org/10.1680/geot.2005.55.2.123.

Hillman, M., Chen, J.S., 2016. An implicit gradient meshfree formulation for convection-dominated problems. In: Bazilevs, Y., Takizawa, K. (Eds.), Advances in Computational Fluid-Structure Interaction and Flow Simulation: New Methods and Challenging Computations. Springer International Publishing, Cham, pp. 25–37. http://dx.doi.org/10.1007/978-3-319-40827-9_3.

Hu, W., Chen, Z., 2003. A multi-mesh MPM for simulating the meshing process of spur gears. Comput. Struct. 81 (20), 1991–2002. http://dx.doi.org/10.1016/S0045-7949(03)00260-8.

Huerta, A., Fernández-Méndez, S., 2000. Enrichment and coupling of the finite element and meshless methods. Internat. J. Numer. Methods Engrg. 48 (11), 1615–1636. http://dx.doi.org/10.1002/1097-0207(20000820)48:11<1615::AID-NME883>3.0.CO;2-S.

Kumar, A.V., Padmanabhan, S., Burla, R., 2008. Implicit boundary method for finite element analysis using non-conforming mesh or grid. Internat. J. Numer. Methods Engrg. 74, 1421–1447. http://dx.doi.org/10.1002/nme.2216.

Liu, C., Sun, W., 2019. Shift boundary material point method: An image-to-simulation workflow for solids of complex geometries undergoing large deformation. Comput. Part. Mech. 7, 291–308. http://dx.doi.org/10.1007/s40571-019-00239-y.

Main, A., Scovazzi, G., 2018. The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. J. Comput. Phys. 372, 972–995. http://dx.doi.org/10.1016/j.jcp.2017.10.026.

Mao, S., Chen, Q., Li, D., Feng, Z., 2016. Modeling of free surface flows using improved material point method and dynamic adaptive mesh refinement. J. Eng. Mech. 142 (2), 04015069. http://dx.doi.org/10.1061/(ASCE)EM.1943-7889.0000981.

Martinelli, M., Al-Kafaji, I., Ceccato, F., Rohe, A., Yerro-Colom, A., Chmelnizkij, A., 2017. MPM Software – Anura3D. Anura3D MPM Research Community, Delft.

Martinelli, M., Galavi, V., 2021. Investigation of the material point method in the simulation of cone penetration tests in dry sand. Comput. Geotech. 130, 103923. http://dx.doi.org/10.1016/j.compgeo.2020.103923.

Martinelli, M., Lee, W.L., Shieh, C.L., Cuomo, S., 2021. Rainfall boundary condition in a multiphase material point method. In: Tiwari, B., Sassa, K., Bobrowsky, P.T., Takara, K. (Eds.), Understanding and Reducing Landslide Disaster Risk. WLF 2020. ICL Contribution To Landslide Disaster Risk Reduction. Springer, Cham, pp. 303–309. http://dx.doi.org/10.1007/978-3-030-60706-7_29.

Mast, C.M., Mackenzie-Helnwein, P., Arduino, P., Miller, G.R., 2011. Landslide and debris flow-induced static and dynamic loads on protective structures. In: Borja, R.I. (Ed.), Multiscale and Multiphysics Processes in Geomechanics. Springer-Verlag Berlin Heidelberg, pp. 169–172. http://dx.doi.org/10.1007/978-3-642-19630-0_43.

Moormann, C., Hamad, F., 2015. MPM dynamic simulation of a seismically induced sliding mass. IOP Conf. Ser.: Earth Environ. Sci. 26 (1), 012024. http://dx.doi.org/10.1088/1755-1315/26/1/012024.

Nguyen, V.P., Rabczuk, T., Bordas, S., Duflot, M., 2008. Meshless methods: A review and computer implementation aspects. Math. Comput. Simulation 79 (3), 763–813. http://dx.doi.org/10.1016/j.matcom.2008.01.003.

Phuong, N.T.V., van Tol, A.F., Elkadi, A.S.K., Rohe, A., 2016. Numerical investigation of pile installation effects in sand using material point method. Comput. Geotech. 73, 58–71. http://dx.doi.org/10.1016/j.compgeo.2015.11.012.

Prusinkiewicz, P., Raghavan, V.V., 1985. A simple space-optimal contour algorithm for a set of iso-rectangles. Congr. Numer. 46, 249–270.

Remmerswaal, G., 2017. Development and Implementation of Moving Boundary Conditions in the Material Point Method (Master's thesis). Delft University of Technology.

Sadeghirad, A., Brannon, R.M., Burghardt, J., 2011. A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. Internat. J. Numer. Methods Engrg. 86 (12), 1435–1456. http://dx.doi.org/10.1002/nme.3110, arXiv:1010.1724.

Sadeghirad, A., Brannon, R.M., Guilkey, J.E., 2013. Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces. Internat. J. Numer. Methods Engrg. 95 (11), 928–952. http://dx.doi.org/10.1002/nme.4526.

Sethian, J.A., 1996. Theory, algorithms, and applications of level set methods for propagating interfaces. Acta Numer. 5, 309–395. http://dx.doi.org/10.1017/S0962492900002671.

Smith, I.M., Griffiths, D.V., Margetts, L., 2014. Programming the Finite Element Method, Fifth ed. John Wiley & Sons, Chichester.

Sołowski, W.T., Sloan, S.W., 2015. Evaluation of material point method for use in geotechnics. Int. J. Numer. Anal. Methods Geomech. 39 (7), 685–701. http://dx.doi.org/10.1002/nag.2321.

Sulsky, D., Chen, Z., Schreyer, H.L., 1994. A particle method for history-dependent materials. Comput. Methods Appl. Mech. Engrg. 118 (1–2), 179–196. http://dx.doi.org/10.1016/0045-7825(94)90112-0.

Tielen, R., Wobbes, E., Möller, M., Beuth, L., 2017. A high order material point method. Procedia Eng. 175, 265–272. http://dx.doi.org/10.1016/j.proeng.2017.01.022.

Tjung, E.Y.S., 2020. Material Point Method for Large Deformation Modeling in Geomechanics Using Isoparametric Elements (Ph.D. thesis). University of California, Berkeley.

Tjung, E.Y.S., Kularathna, S., Kumar, K., Soga, K., 2020. Modeling irregular boundaries using isoparametric elements in material point method. In: Geo-Congress 2020. American Society of Civil Engineers, Reston, VA, pp. 39–48. http://dx.doi.org/10.1061/9780784482803.005, arXiv:1909.13389.

Torres, D.J., Brackbill, J.U., 2000. The point-set method: Front-tracking without connectivity. J. Comput. Phys. 165 (2), 620–644. http://dx.doi.org/10.1006/jcph.2000.6635.

Wang, L., Coombs, W.M., Augarde, C.E., Cortis, M., Brown, M.J., Brennan, A.J., Knappett, J.A., Davidson, C., Richards, D., White, D.J., Blake, A.P., 2021. An efficient and locking-free material point method for three-dimensional analysis with simplex elements. Internat. J. Numer. Methods Engrg. 122, 3876—3899. http://dx.doi.org/10.1002/nme.6685.

Wang, B., Karuppiah, V., Lu, H., Komanduri, R., Roy, S., 2005. Two-dimensional mixed mode crack simulation using the material point method. Mech. Adv. Mater. Struct. 12 (6), 471–484. http://dx.doi.org/10.1080/15376490500259293.

Wang, B., Vardon, P.J., Hicks, M.A., 2016a. Investigation of retrogressive and progressive slope failure mechanisms using the material point method. Comput. Geotech. 78, 88–98. http://dx.doi.org/10.1016/j.compgeo.2016.04.016.

Wang, B., Vardon, P.J., Hicks, M.A., 2018. Rainfall-induced slope collapse with coupled material point method. Eng. Geol. 239, 1–12. http://dx.doi.org/10.1016/j.enggeo.2018.02.007.

Wang, B., Vardon, P.J., Hicks, M.A., Chen, Z., 2016b. Development of an implicit material point method for geotechnical applications. Comput. Geotech. 71, 159–167. http://dx.doi.org/10.1016/j.compgeo.2015.08.008.

Wei, H., Chen, J.-S., Beckwith, F., Baek, J., 2020. A naturally stabilized semi-Lagrangian meshfree formulation for multiphase porous media with application to landslide modeling. J. Eng. Mech. 146 (4), 04020012. http://dx.doi.org/10.1061/(ASCE)EM.1943-7889.0001729.

Wei, H., Chen, J.-S., Hillman, M., 2016. A stabilized nodally integrated meshfree formulation for fully coupled hydro-mechanical analysis of fluid-saturated porous media. Comput. & Fluids 141, 105–115. http://dx.doi.org/10.1016/j.compfluid.2015.11.002.

Xu, X., Jin, F., Sun, Q., Soga, K., Zhou, G.G.D., 2019. Three-dimensional material point method modeling of runout behavior of the Hongshiyan landslide. Can. Geotech. J. 56 (9), 1318–1337. http://dx.doi.org/10.1139/cgj-2017-0638.

Yerro, A., Alonso, E.E., Pinyol, N.M., 2016. Run-out of landslides in brittle soils. Comput. Geotech. 80, 427–439. http://dx.doi.org/10.1016/j.compgeo.2016.03.001.

York, A.R., 1997. The Development of Modifications to the Material Point Method for the Simulation of Thin Membranes, Compressible Fluids, and Their Interactions (Ph.D. thesis). The University of New Mexico.

York, A.R., Sulsky, D., Schreyer, H.L., 1999. The material point method for simulation of thin membranes. Internat. J. Numer. Methods Engrg. 44 (10), 1429–1456. http://dx.doi.org/10.1002/(SICI)1097-0207(19990410)44:10<1429::AID-NME536>3.0.CO;2-4.

Zhang, D.Z., Ma, X., Giguere, P.T., 2011. Material point method enhanced by modified gradient of shape function. J. Comput. Phys. 230 (16), 6379–6398. http://dx.doi.org/10.1016/j.jcp.2011.04.032.

Zhao, X., Bolognin, M., Liang, D., Rohe, A., Vardon, P.J., 2019a. Development of in/outflow boundary conditions for MPM simulation of uniform and non-uniform open channel flows. Comput. & Fluids 179, 27–33. http://dx.doi.org/10.1016/j.compfluid.2018.10.007.

Zhao, S., Bui, H.H., Lemiale, V., Nguyen, G.D., Darve, F., 2019b. A generic approach to modelling flexible confined boundary conditions in SPH and its application. Int. J. Numer. Anal. Methods Geomech. 43 (5), 1005–1031. http://dx.doi.org/10.1002/nag.2918.

Zienkiewicz, O.C., Taylor, R.L., Fox, D., 2014. The Finite Element Method for Solid and Structural Mechanics, Seventh ed. Butterworth-Heinemann, Oxford, http://dx.doi.org/10.1016/B978-1-85617-634-7.00002-8.