Computer Methods in Applied Mechanics and Engineering 442 (2025) 118012

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

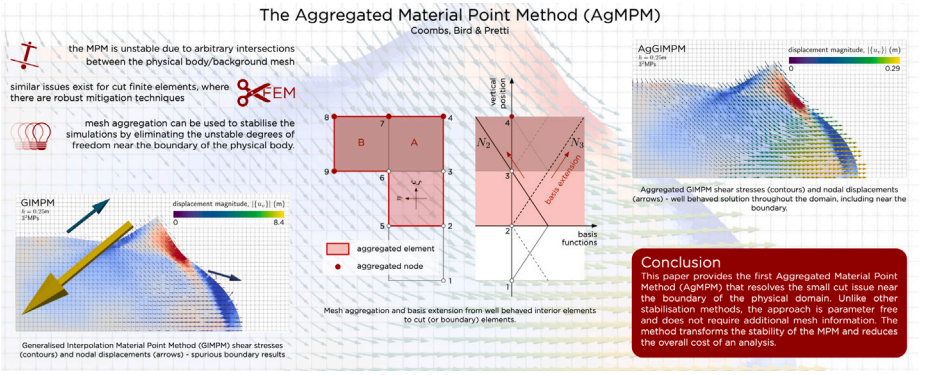
journal homepage: www.elsevier.com/locate/cma

The Aggregated Material Point Method (AgMPM)

William M. Coombs *, Robert E. Bird, Giuliano Pretti

Department of Engineering, Durham University, South Road, Durham, DH1 3LE, County Durham, UK

GRAPHICAL ABSTRACT



ARTICLE INFO

Dataset link: https://dx.doi.org/10.15128/r1gm80hv42k

Keywords:
Material point method
Finite deformation mechanics
Mesh aggregation
Implicit quasi-static
Explicit dynamic

ABSTRACT

The Material Point Method (MPM) has been shown to be an effective approach for analysing large deformation processes across a range of physical problems. However, the method suffers from a number of spurious artefacts, such as a widely documented cell crossing instability, which can be mitigated by adopting basis functions with higher order continuity. The larger stencil of these basis functions exacerbate a less widely discussed issue - *small cuts*. The small cut issue is linked to the arbitrary interaction between the physical body and the background mesh that is used to assemble and solve the governing equations in the MPM. There is the potential for degrees of freedom near the boundary of the body to have very small contributions from material points, which causes two problems: (i) artificially large accelerations/displacements at the boundary and (ii) ill conditioning of the global linear system. This paper provides a new mesh Aggregated MPM, or AgMPM, that mitigates the small cut issue by forming aggregated elements, tying the ill-behaved degrees of freedom to well posed interior elements. Implicit quasi-static and explicit dynamic formulations are provided and demonstrated through a series of numerical examples. The approach does not introduce any new numerical parameters and can be applied to implementations that adopt a lumped mass matrix. Aggregation is shown to significantly improve the stability of implicit implementations of the MPM, often at a lower

* Corresponding author.
E-mail address: w.m.coombs@durham.ac.uk (W.M. Coombs).

https://doi.org/10.1016/j.cma.2025.118012
Received 10 January 2025; Received in revised form 2 April 2025; Accepted 8 April 2025
Available online 30 April 2025

computational cost compared to standard, non-aggregated, implementations. The technique improves the energy conservation and the stress field of explicit dynamic MPMs.

1. Introduction

The Material Point Method (MPM) was proposed by Sulsky et al. [1] as an extension of Particle In Cell (PIC)/FLuid Implicit Particle (FLIP) approaches to solid, history-dependent materials. Over the last 30 years it has gained widespread traction in modelling large deformation problems where other mesh-based methods, such as the Finite Element Method (FEM), can struggle due to mesh distortion. The key concept is to remove the strong coupling between the physical body and the mesh that is used to solve the governing equations. The physical body, represented by a collection of material points, is allowed to deform through a background grid of finite elements, which is used to assemble and solve the governing equations. The method has parallels with immersed/unfitted FEMs [2], and the method can be grouped under the expression coined by Burman et al. [3] as *"mesh-free method[s] posed on a mesh"*. The methods have similar advantages but also struggle with some of the same issues, such as boundary representation [4–6], imposition of boundary conditions [4,7–13], accuracy of integration [14] and, the focus of this paper, instabilities caused by the arbitrary interaction between the body and the mesh.

Similar to immersed FEMs, the MPM suffers from stability and ill conditioning issues associated with *small cuts*. The first issue can result in physically spurious results close to the boundary, such as artificially large accelerations or displacements that impact on the quality of the stress field. The second issue impacts on the accuracy of the linear solution and the ability to use efficient iterative solvers, which are a requirement for large three-dimensional problems. The issue is linked to the arbitrary interaction between the physical body, defined by a collection of material points, and the background grid, which is used to construct and solve the governing equations. The stability issue was recognised by Sulsky et al. [15] and summarised as

> *"...if a single material point crosses into an element and is close to the element boundary, then the value of a nodal basis function identified with a node at the opposite side of the element may be small. However, the internal force vector involves the gradient of the basis function, which does not approach zero for points near the element boundary. The result is that computed accelerations at outer nodes can occasionally be unphysical..."*

The problem is compounded by adopting basis functions that generate higher order continuity between elements, such as generalised interpolation (GIMPM, [16]) and B-Spline basis functions [10]. This compounding factor was also identified by the immersed FEM community, where *"ill-conditioning problems become particularly apparent when higher-order methods"* are adopted [17]. The use of higher order basis functions increases the potential for a small contribution to the resulting linear system of equations from degrees of freedom near the boundary of the physical domain, this can be due to the extended stencil of spline-based basis functions or simply the non-linear form of the functions having small values (or gradients) near element boundaries. It was also proved by Prenter et al. [17] that the condition number of the linear system is inversely proportional size of the smallest volume contribution (i.e. the smallest cut), which scales to the power of the polynomial order of the basis functions, p.

In explicit MPM implementations the impact of the small cut issue can be reduced by adopting the Modified Updated Stress Last (MUSL, [15]) approach with a lumped mass matrix. Lumping means that inversion of the mass matrix to determine the nodal accelerations is trivial (avoiding the ill conditioning problem and removing the requirement for an efficient linear solver for large problems). Spurious accelerations can still be predicted close to the boundary (the stability issue), but these are damped by an additional momentum-based material point to node velocity projection step. This means that the spurious accelerations are weighted by the small masses that caused them, and a more reasonable velocity field is reconstructed prior to updating the stresses at the material points. Another explicit approach is to ignore [15] or redistribute [18] the small masses (and the associated forces) if they fall below a value which is deemed problematic. However, the definition of *small* is arbitrary and it has been shown that the technique does not stabilise the problem in all cases [19]. Wang et al. [20] was one of the first papers to recognise the stability issue within quasi-static implicit implementation of the MPM. They proposed including a *soft stiffness* integrated over the background grid using Gauss quadrature to improve the numerical stability. Interestingly, the soft stiffness approach has parallels with fictitious domain virtual stiffness adopted in some immersed FEMs [21,22], where a stiffness the material outside of the domain is set to a small value to avoid conditioning issues of the global matrix. However, this soft stiffness approach [20] is rather crude as, unlike the virtual stiffness approach, it is applied to the full domain, irrespective if elements are suffering from the small cut issue or are fully inside the physical body. An alternative solution was proposed by Chandra et al. [8] who suggested re-weighting the basis functions for implicit MPMs to mitigate the small cut issue. Similar to mass cut off approaches, the technique requires the definition of a threshold below which to cut and re-weight the basis functions. Yamaguchi et al. [10] proposed the use of Extended B-Spline (EBS) basis functions to reduce the likelihood of the small cut problem occurring. The approach modifies the basis functions near the edge of the physical domain such that the *"degenerated bases are replaced by the linear combination of those of the stable bases"* [10]. This method is part of the wider group of basis function manipulation methods, where the functions are constrained to geometrically nearby functions (for example, Web-splines [23,24] and i-splines [25]) or by excluding them completely [17]. Recently, Coombs [19] recognised

the parallels between the MPM and immersed FEM communities in suffering from the *small cut* issue and adopted Burman's Ghost stabilisation technique [26] to mitigate the problem for implicit and explicit MPMs. The approach has also been extended to bi-phase materials by Pretti et al. [27], who stabilised both the displacement and pressure field for a quasi-static *u-p* implicit MPM. However, the approach relies on the specification of an appropriate penalty, or stabilisation, parameter which depends on the equations being solved and the material being analysed. It also requires information on the face connectivity of the background mesh, which may not be available in many existing MPM implementations. Finally, the technique has zero impact on implementations that use a lumped mass matrix as the lumping of the stabilisation matrix (summation over each degree of freedom) results in a null matrix. Therefore, the approach cannot be applied to most explicit dynamic MPM implementations, where lumping of the mass matrix is the default option.

It is worth noting that specific preconditions have been developed to address the poor conditioning of the linear system equations (but not the underlying stability issue) that result from discretisations suffering from the small cut problem. Example papers in this area include [17,28–30], with Schwarz preconditioners being of particular interest in immersed/unfitted FEMs [31]. However, as far as the authors are aware, to date there has been little research on the use of immersed FEM-type preconditioners within the MPM to address the small cut issue. The likely reason for this is the dominance of lumped mass matrices in large explicit MPM analyses.

Cell aggregation or cell agglomeration techniques, such as the aggregated FEM, AgFEM [32], is an approach used in the immersed FEM literature, where cells with the small cut problem are merged with neighbouring full cells forming aggregates. FEM implementations of the technique have been shown to be successful at mitigating the small cut issue but they have yet to be applied within the MPM. As far as the authors are aware there is only one MPM paper that utilises the concept of aggregation. Zheng et al. [33] used an aggregation technique to reduce the cost of B-spline basis functions by aggregating internal degrees of freedom. However, the focus is on the *cost* of the analysis and not on the stabilisation of the method in terms of the small cut problem. In fact, we do not believe that approach would mitigate the problem due to the focus on internal degrees of freedom, which are not problematic in terms of numerical stability.

This paper represents the first time that cell/mesh aggregation techniques have been applied to the MPM with the specific intention to mitigate the small cut issue, resulting in the Aggregated MPM (AgMPM). The following sections explain key aspects of the adopted quasi-static implicit and dynamic explicit MPM formulations, the mesh aggregation technique and the adaptations required to implement the approach within the MPM. The proposed method is demonstrated on a number of test cases and compared with the ghost stabilisation technique as well as non-stabilised MPM implementations.

2. Material point method

The Material Point Method (MPM, [1]) is well established and this section does not attempt to provide a full description of the approach. The implementation adopted in this paper is based on the open-source AMPLE (A Material Point Learning Environment) code of Coombs and Augarde [34]. A basic outline of the approach is provided below along with the key equilibrium equations and a description of the adopted basis functions.

As shown in Fig. 1, a typical load (or time) step in the material point method requires the following steps:

(a) At the start of the current step, the physical body is represented by a number of material points (shown by the dark grey filled circles) which is positioned within a background grid of finite elements.
(b) Point-to-grid map: information about the physical body is mapped to the nodes of the background grid. The transferred information depends on the governing equations. For example, quasi-static stress analysis requires information about the stiffness, internal force and gravitational external forces at the material points to be mapped to the nodes of the background grid using the basis functions and spatial derivatives.
(c) Assemble the governing equations at the nodes (the white-shaded circles) of the background grid, where the material points act as quadrature points.
(d) Solve the governing equations at the nodes of the background grid with appropriate boundary conditions (imposed weakly or strongly at the nodes).
(e) Grid-to-point map: the nodal solution is mapped to the material points in terms of material point displacements, deformation state, etc.
(f) Update the positions of the material points and reset/replace the background grid. Return to step (a).

2.1. Large deformation continuum formulation

The form of the MPM adopted in this paper is based on the work of Charlton et al. [35] using the open source AMPLE code of Coombs and Augarde [34] but extended to include inertia. A brief outline of the updated Lagrangian large deformation continuum formulation is given below; interested readers are referred to the above papers for details. Within this approach, the Galerkin form of the updated Lagrangian weak statement of equilibrium over each background grid element, $\mathcal{K}$, is

$$\int_{\varphi_t(\mathcal{K})} [\nabla_x S_{vp}]^T \{\sigma\} \mathrm{d}V + \int_{\varphi_t(\mathcal{K})} \rho [S_{vp}]^T \{\dot{v}\} \mathrm{d}V - \int_{\varphi_t(\mathcal{K})} [S_{vp}]^T \{b\} \mathrm{d}V - \int_{\varphi_t(\partial\Omega)} [S_{vp}]^T \{t\} \mathrm{d}S = \{0\}, \tag{1}$$

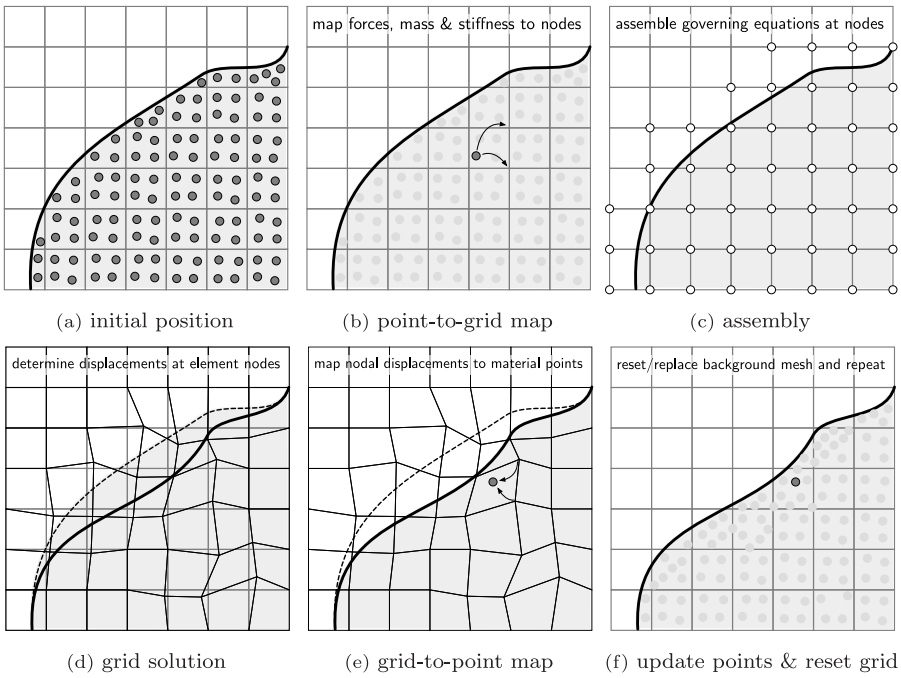


Fig. 1. Material point method steps. *Source:* Adapted from Coombs and Augarde [34].

where φ_t is the motion of the physical body, $[S_{vp}]$ is the basis/shape function matrix, which is used to transfer information between the material points (subscript p) and the vertices of the background grid (subscript v), $[\nabla_x S_{vp}]$ is the strain–displacement matrix containing derivatives of the basis functions with respect to the updated (deformed) coordinates, $\{\sigma\}$ is a vector representation of the Cauchy stress, ρ is the density of the material, $\{\dot{v}\}$ is the acceleration, $\{b\}$ is the body force per unit volume and $\{t\}$ is the surface traction acting over the boundary of the body, $\partial\Omega$, which does not necessarily coincide with the boundary of the element. In the MPM the material points act as quadrature points when integrating over the background grid. This means that the weak statement of equilibrium, (1), becomes

$$\underset{\forall p}{\mathbf{A}}\left([\nabla_x S_{vp}]^T\{\sigma_p\}V_p + [S_{vp}]^T\{\dot{v}_p\}m_p - [S_{vp}]^T\{b\}V_p\right) - \int_{\varphi_t(\partial\Omega)}[S_{vp}]^T\{t\}\mathrm{d}S = \{0\}, \tag{2}$$

where V_p and $m_p = \rho V_p$ are the volume in the updated frame and mass associated with the material point. The surface traction contribution remains expressed in terms of the boundary of the physical body. The large deformation elasto-plastic continuum mechanics formulation used in this paper can be found in Charlton et al. [35]. In brief, the formulation multiplicatively decomposes the deformation gradient into elastic and plastic components and combines this Lee decomposition with a linear relationship between Kirchhoff stress and logarithmic elastic strain along with an exponential map of the plastic flow rule. This allows isotropic small strain plasticity algorithms to be used directly within a large deformation setting without modifying the stress update procedure [36]. The approach is widely used in large deformation finite element methods (see for example [37]), and in material point methods. Details of the specific constitutive models used in this paper are included as part of the numerical examples in Section 4.

2.2. Implicit quasi-static analysis

Quasi-static implementations neglect the acceleration term in (1) and (2). This means that the adopted large deformation elasto-plastic continuum formulation, (2), is a non-linear equation in terms of the displacements of the background grid. Consistent with the AMPLE code [34], in this paper the quasi-static non-linear problem is solved using a Newton–Raphson approach which continues until the residual out of balance force of (2), defined as $\{f_R\}$, converges to within a given tolerance. This requires repeating steps (b) through (d) in Fig. 1 and only once (2) is satisfied are the material point positions updated using the basis functions linking the material points and the vertices of the background mesh. See Coombs and Augarde [34] for details.

2.3. Explicit dynamics

Explicit dynamic implementations of the MPM predict the next state of the material based on information at the current state. Assuming that the accelerations of the material points can be obtained from the background grid accelerations, $\{\dot{v}_v\}$, via

$$\{\dot{v}_p\} = \sum_{\forall v} S_{vp}\{\dot{v}_v\}, \tag{3}$$

and neglecting the external traction term, the accelerations of the background grid can be computed as

$$\{\dot{v}_v\} = [M]^{-1} \underset{\forall p}{\mathbf{A}} \left([S_{vp}]^T \{b\} V_p - [\nabla_x S_{vp}]^T \{\sigma_p\} V_p \right) \tag{4}$$

where $[M] = \underset{\forall p}{\mathbf{A}} \left([S_{vp}]^T [S_{vp}] m_p \right)$ is the consistent mass matrix. In practice the vast majority of explicit dynamics implementation of the MPM adopt a lumped mass matrix for the reasons outlined in the introduction.

Once obtained, the nodal accelerations can be used to provide a FLIP (FLuid Implicit Particle) update on the material point velocities

$$\{v_p\}_{n+1} = \{v_p\}_n + \Delta t \sum_{\forall v} S_{vp}\{\dot{v}_v\}, \tag{5}$$

where Δt is the time step, and n and $n+1$ are the previous and current time steps, respectively. In order to update the material point positions the nodal velocities at the start of the time step are required, which are obtained through

$$\{v_v\}_n = [M]^{-1} \underset{\forall p}{\mathbf{A}} \left(m_p [S_{vp}]^T \{v_p\}_n \right), \tag{6}$$

which can be used to determine the updated nodal velocities $\{v_v\}_{n+1} = \{v_v\}_n + \Delta t\{\dot{v}_v\}$ and the updated material point positions

$$\{x_p\}_{n+1} = \{x_p\}_n + \Delta t \sum_{\forall v} S_{vp}\{v_v\}_{n+1}. \tag{7}$$

There exist two options for explicit MPM implementations when updating the material point stress, with descriptive names: Update Stress First (USF) and Update Stress Last (USL). In USF approaches the stress at the material points is updated prior to determination of the nodal accelerations using the increment in the deformation gradient from the previous step. USL approaches update the stress state at the end of the current step. The Modified USL (MUSL), as the name suggests, is a modification to the USL approach that was described in the introduction. The approach includes an additional momentum-based determination of the nodal velocities using (6) with the updated material point velocities prior to updating the material point stress state. As described by Sulsky et al. [15], the aim of the MUSL is to mitigate unphysical accelerations at the boundary of the body by re-weighting by the small masses that caused the spurious values.

2.4. Basis functions

Generalised Interpolation Material Point Method (GIMPM, [16]) C^1-continuous basis functions are adopted in this paper to mitigate the well-documented cell crossing instability. The basis functions can be expressed as

$$S_{vp} = \frac{1}{V_d} \int_{\Omega_p} \chi_p N_v dV \tag{8}$$

and are obtained by integrating the product of the basis functions associated with the background grid, N_v, with a characteristic function, χ_p, over the rectangle defining the domain associated with the material point, Ω_p, normalised by the volume of the domain (see Fig. 2, plotted in terms of the undeformed coordinates at the start of a load or time step, $\tilde{X}$). For GIMPM basis functions, χ_p is a unity hat function of length $2l_p$ with a value of 1 inside the material point's domain and zero elsewhere. The integration of the background grid basis functions over the material point domain results in five equations for S_{vp}, as identified by Regions A through E in Fig. 2 for Node 2 of the background grid (see [34] for the five basis function equations and spatial derivatives). If the material point domain is fully within an element (Regions B and D), S_{vp} will be equal to N_v, the other regions represent cases where the material point domain overlaps multiple elements. In these overlapping cases (Regions A, C and E) the basis functions are C^1 continuous across element boundaries. Note that GIMPM basis functions are non-interpolatory at the nodal locations; $S_{vp} \neq 1$ at $\tilde{X}_p = \tilde{X}_v$. A key observation is that GIMPM basis functions extend the region of influence of the node to $h + l_p$, compared to h in the case of conventional polynomials basis functions. This has the consequence of increasing the stencil of the basis functions and therefore the bandwidth within the resulting linear system of equations. The basis functions of the background grid can be recovered by setting χ_p to a Dirac delta function, such that $S_{vp} = N_v$ as in the original MPM [1]. Note that the integral is normalised by the domain volume, V_d, as, depending on the domain updating procedure, it is not guaranteed this volume will be the same as V_p (see Coombs et al. [38] for a discussion on domain updating procedures). Here the domains are updated according to the approach included in AMPLE [34].

Although GIMPM basis functions mitigate the cell crossing instability, they increase the possibility of a problem suffering from the small cut instability. This is due to the basis functions being obtained from the integral of the background grid basis over the material point domain. If a small overlap exists between a domain and a background grid element (for example in Regions A or E

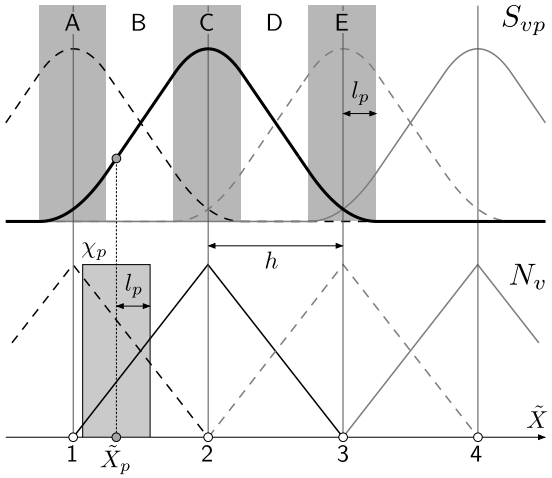


Fig. 2. 1D basis functions: linear finite element basis functions, N_v (bottom), and GIMPM basis functions, S_{vp} (top). The white-shaded circles are nodes of the background grid and the grey-shaded circle represents a material point. $\tilde{X}$ are the coordinates at the start of a load or time step.
Source: Adapted from [39],

of Fig. 2) this will result in very small basis function values between the material point and the vertices of the mesh. This is less of an issue for the original MPM, which uses the basis functions of the background grid sampled at the material point location; it is only when the material point is very close to an element boundary that the small cut problem manifests for the original MPM. The next section will detail how mesh aggregation can correct the small cut issue for both the original MPM and the GIMPM.

3. Mesh aggregation

Mesh aggregation or cell agglomeration techniques, such as the aggregated FEM, AgFEM [32], is an approach used in the immersed FEM community, where cells at the boundary of the domain with the small cut cell problem are merged with neighbour interior cells forming aggregates. The approach is designed to tackle both the ill conditioning issue and the stability problem introduced by small cuts, and relies on defining a discrete extension operator from degrees of freedom on well-posed elements to ill-posed (typically the cut) elements [31]. The approach has parallels with discrete extension operators [40], which stabilise the cut elements by modifying the finite element space close to the boundary through extension from stable interior elements. As far as the authors are aware, AgFEM techniques have yet to be applied to the MPM.

It is important to highlight the differences between aggregation and Extended B-Spline (EBS) and other basis manipulation methods. As commented by Badia et al. [32], mesh aggregation

> *"is related to the extension techniques in [24,41,42] in the sense that we also define linear constraints between each ill-conditioned DOF and a set of some well-behaved master DOFs. However, the way such constraints are built is completely different and it is not restricted to B-splines. In fact, the approach is quite insensitive to the functional space, since it relies on a geometrical aggregation of cells and a dof-to-cell map."*

While the EBS and the aggregation pursue the same goal and share similarities (the stencil of the basis functions is modified for both techniques), there are also substantial differences between the methods. The EBS trims or extends the basis function stencils to exclude the poorly-integrated nodes and modifies the basis functions of well-integrated neighbouring nodes to ensure the partition of unity. The aggregation instead only extends a basis function's stencil beyond their usual definition, creating macro-elements. In this fashion, the nodal values of displacements of well-behaved nodes can interpolate on bigger domains, encompassing those of elements with poorly-integrated nodes.

The aggregation approach relies on first identifying the cut, or boundary, elements, $\mathcal{K}_B$, and then forming element aggregations with neighbouring interior elements, $\mathcal{K}_I$, from which the basis functions are extended to eliminate the potentially ill-posed degrees of freedom. The aggregation process is shown in Fig. 3. Fig. 3(a) shows the physical body being analysed, Ω, within a computational domain. The boundary of the physical body intersects a number of *cut* or boundary elements, shown by the mid grey-shaded elements in Fig. 3(b), and contains a number of white-shaded interior elements. Elements fully outside of the body are inactive and shaded light grey with a dashed element boundary.[1] The boundary elements need to be associated with an interior element in order to form

[1] Note that there is the potential for confusion when describing elements "outside" the body, especially when considering basis functions with extended influence (such as B-spline basis functions). In this case, a more precise definition is to state that the inactive elements are the elements that have no influence on the solution in the physical body.

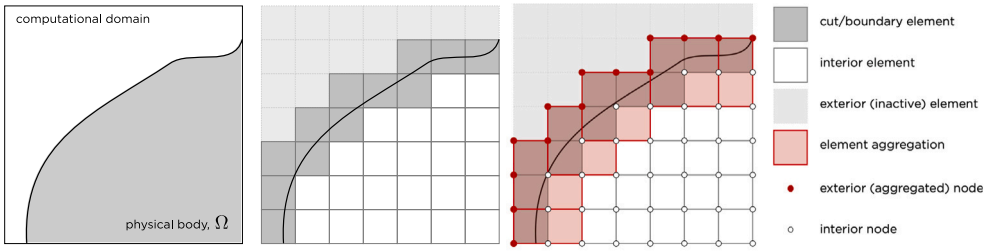


Fig. 3. Mesh aggregation definitions and steps.

an element aggregation. Each aggregate will contain one (and only one) interior element and one or more boundary elements [32], as shown in Fig. 3(c). Once the aggregates have been established, the active nodes that are attached to a boundary element but not an interior element (shown by the red circles in Fig. 3(c)) need to be tied to the nodes of their associated interior element. This involves extending the basis functions of the interior element to the position of the node that needs to be aggregated, linking the degrees of freedom of the exterior nodes to those of the interior nodes (the white circles in Fig. 3(c)). Note that in many cases the link between an exterior node and an interior element is non-unique as the exterior node may be connected to multiple boundary elements. As commented by Badia et al. [32], this is not important provided that each exterior node is tied to the degrees of freedom of the interior nodes.

The displacement of an aggregated node, $\{u_b\}$, is removed as an independent variable and instead is obtained by extending the basis functions of the interior element, such that

$$\{u_b\} = \sum_{a=1}^{n_e} N_a(\{x_b\})\{u_a\}, \tag{9}$$

where n_e is the number of nodes of the interior element, N_a are the basis functions of the interior element, $\{x_b\}$ is the global location of the node being aggregated and $\{u_a\}$ are the displacements of the nodes of the interior element. The displacements of all of the external aggregated nodes can be compactly expressed as

$$\{d^{\text{ext}}\} = [C]\{d^{\text{int}}\}, \tag{10}$$

where $[C]$ is a global constraint matrix that contains the basis functions that link the exterior to the interior elements. This allows the definition of an extension matrix with the form

$$[E] = \begin{bmatrix} [I] \\ [C] \end{bmatrix}, \qquad \text{such that} \qquad \begin{Bmatrix} \{d^{\text{int}}\} \\ \{d^{\text{ext}}\} \end{Bmatrix} = [E]\{d^{\text{int}}\}, \tag{11}$$

where $[I]$ is an identity matrix. $[E]$ has dimensions $(n_{\text{int}} + n_{\text{ext}})$ by n_{int}, where n_{int} and n_{ext} are the number of interior and exterior degrees of freedom, respectively.

3.1. Implementation

This section details key aspects of the implementation of mesh aggregation within the MPM, focusing on: (i) identification of boundary elements and nodes for aggregation, (ii) nodal aggregation, (iii) modification to the linear solution for implicit quasi-static aggregated methods and (iv) the changes required for an explicit dynamic MPM with mesh aggregation.

Boundary element and aggregated node identification

Determining the boundary elements is a necessary step for node aggregation. Compared to immersed/unfitted FEMs, a precise definition of the boundary is not usually available in the MPM, where the boundary of the physical body is implied by the distribution of material points. Therefore, an alternative approach is required for identifying the *cut* or boundary elements. Following the approach of Coombs [19], for each physical body:

(i) determine which elements are populated by the material points representing the body, that is the elements that are *active* in the analysis (the white-shaded elements in Fig. 4(b));
(ii) determine the faces on the boundary between the active and inactive parts of the mesh (faces where one of its associated elements is active and one that is inactive); and
(iii) using the faces from (ii), identify the *boundary elements* (the dark grey shaded elements in Fig. 4(c)).

These are the elements that potentially suffer from the small cut issue and need to be aggregated with the interior elements, as shown by the red shaded regions in Fig. 4(d). However, it is only the active nodes that are only connected to a boundary element that require aggregation (the red nodes in Fig. 4(e)). This set of aggregated (or boundary or exterior) nodes, $\mathcal{N}_B$, can be determined

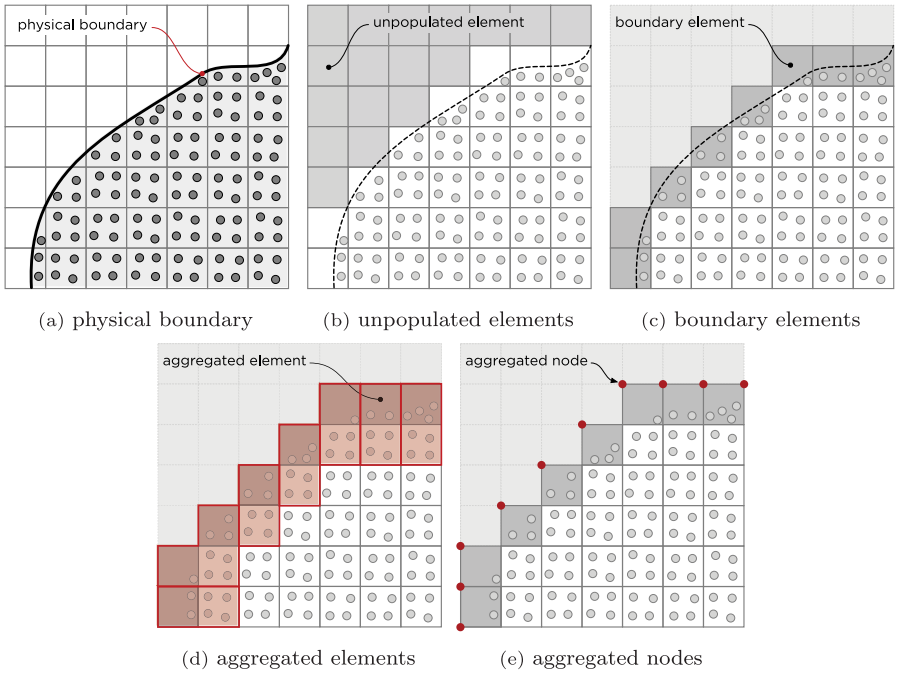


Fig. 4. Physical boundary to node aggregation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

by comparing all of the active nodes, $\mathcal{N}_A$, in the analysis with the nodes attached to the interior elements, $\mathcal{N}_I$, the nodes that require aggregation are ones in $\mathcal{N}_A$ but not in $\mathcal{N}_I$, that is the set difference $\mathcal{N}_B = \mathcal{N}_A \backslash \mathcal{N}_I$.

In this paper it is assumed that any degree of freedom that is at the edge of the domain, and therefore has the potential to suffer from the small cut issue, should be aggregated with interior degrees of freedom. However, there may well be boundary elements that are well populated with material points and would *behave* without aggregation or stabilisation. A threshold could be introduced where aggregation is, or is not, applied depending on the distribution of the physical material within the boundary elements and/or the resulting nodal quantities (basis functions, mass, stiffness, etc.) once the MP information has been mapped to the nodes of the background grid. One example of the use of thresholds in the MPM is the approach of Ma et al. [18], where small masses and their associated forces are redistributed to neighbouring degrees of freedom. However, it has been shown that this technique is sensitive to the choice of the cut off threshold and fails when other stabilisation techniques succeed [19]. Yamaguchi et al. [10] introduced an occupation index, the ratio between the volume of physical material within an element of the background grid (i.e. the volume of MPs in the element) and element volume, for the application of extended B-Splines. They demonstrated that the approach is insensitive to a 0.25 – 0.75 threshold range on the occupation index, which is not surprising given the small cut issue is linked to very small interactions between MPs and degrees of freedom of the background mesh and volume occupation of 0.25 is relatively large in this context. However, it is important to note that the distribution of the material within the element is critical, not just the volume itself, when assessing if a degree of freedom is likely to suffer from the small cut instability. A key advantage of mesh aggregation is that it is fully parameter free, being simply based on the connectivity of the background grid. Therefore, this paper follows the approach of Coombs [19], where stabilisation is applied over all faces that may require stabilisation, and applies aggregation to all exterior degrees of freedom rather than introduce an arbitrary numerical parameter.

Aggregation

As described above, the first step in mesh aggregation is to associate an interior element with each of the boundary elements. The basis functions of the interior element will be extended into the boundary element and used to constrain out the potentially problematic nodes from the linear system of equations. The interior (or connected) element associated with each boundary element is determined through the following steps:

(i) Loop over each face of the boundary element. IF the connected element is on the interior, mark the element as the boundary element's connected element for aggregation.

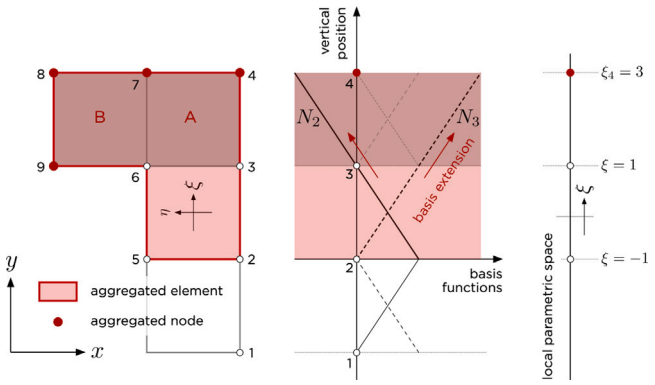


Fig. 5. Aggregated basis. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(ii) IF the boundary element does not share a face with an interior element, determine the closest interior element to the boundary element (in terms of element centres) and mark the element as the boundary element's connected element for aggregation.[2] Note that this is the "corner" case, where the boundary element shares faces with other boundary elements and a node with an interior element.

In both cases the operation is non-unique, but this is not important for the subsequent node aggregation, which is applied node by node.

For each of the boundary nodes in $\mathcal{N}_B$

(i) Determine the local position of the node in the connected element, $\{\xi_B\}$, using the following residual

$$\{x_B\} - [N(\{\xi_B\})]\{x_c\} = \{0\}, \tag{12}$$

where $\{x_B\}$ is the global position of the boundary node, $\{x_c\}$ are the global coordinates of the nodes of the connected element, $[N]$ is the basis function matrix of the connected element defined in terms of the element's local coordinates, $\{\xi\}$. For a linear element the solution can be obtained in a single Newton–Raphson step, which has the added benefit of providing the required basis function values.

(ii) Position the basis function values linking the boundary node to the nodes of the connected element into the global extension matrix, $[E]$.

The above operation is shown for a number of boundary nodes in Fig. 5. The left of the figure shows a situation where two boundary elements (labelled A and B) are associated with an interior element (defined by nodes 2, 5, 6, 3) and form an aggregation. In this case four nodes (the red nodes in Fig. 5) need to be aggregated. First, considering Node 4, due to the Cartesian nature of the boundary grid, only the nodes with the same global x-position as Node 4 will have a non-zero basis function value and in this case we will only consider Nodes 2 and 3 from the interior element, as labelled on the figure. The aggregation process can therefore be considered using the one-dimensional projection shown on the right of the discretised domain (centre of the figure), where the basis function of the red and white-shaded nodes are shown.

Considering the standard isoparametric local coordinate system of a linear two-noded element, $\xi \in [-1, 1]$ as shown on the right of Fig. 5, the position of the red node within the connected node's local system is $\xi_4 = 3$. Calculating the basis functions of the nodes of the connected element gives

$$N_2 = \frac{1}{2}(1 - \xi) = -1 \qquad \text{and} \qquad N_3 = \frac{1}{2}(1 + \xi) = 2.$$

the basis functions of the other nodes of the connected element will be equal to zero. The extension matrix for Nodes 2, 3 and 4 shown on the right of Fig. 5 follows as

$$[E_{([2,3,4],[2,3])}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 2 \end{bmatrix},$$

[2] It is also possible to take the closest interior element approach for all elements that require aggregation, which might be advantageous if face connectivity is not available in the code's data structure. However, if the face information is available it is likely to be more efficient to first check the boundary element faces.

where the modified rows, those associated with Node 4 (the aggregated node), have been highlighted and represent elements of the constraint matrix, $[C]$, as shown in (11).

The other nodes requiring aggregation follow a similar process. It is worth highlighting Node 8, which will have non-zero aggregation values from four nodes as its local position associated with the interior element is $(\xi, \eta) = (3, 3)$. However, all nodes follow the same process in terms of determining the local position of the aggregated node and the associated basis functions of the interior element.

Implicit quasi-static linear solution

For implicit quasi-static analysis, once the global extension matrix, $[E]$, has been formed, it is applied in the linear solution of the non-linear problem. The aggregated stiffness matrix and out of balance residual force vector are defined as

$$[K^{\mathrm{A}}] = [E]^T[K][E] \qquad \text{and} \qquad \{f_R^{\mathrm{A}}\} = [E]^T\{f_R\}, \tag{13}$$

which are used to obtain the aggregated displacement vector that contains the displacements of the interior nodes, $\{d^{\mathrm{int}}\}$, where $[K]$ and $\{f_R\}$ are the non-aggregated global stiffness matrix and residual force vector. The form of the aggregated stiffness and residual force are obtained by substituting (11) into the continuum weak statement of equilibrium. Galerkin's method is then applied to obtain the discrete form of the weak equilibrium statement, which can be linearised for the unknown internal displacements, $\{d^{\mathrm{int}}\}$ – see Appendix A for a summary of the derivation. The full displacement vector is recovered from (11), that is

$$\{d\} = [E]\{d^{\mathrm{int}}\}. \tag{14}$$

Iterations continue until the norm of the aggregated residual force vector, $\{f_R^{\mathrm{A}}\}$, normalised by the norm of the external forces, converges within a given tolerance. Note that as $[E]$ is based on the basis functions, it does not change through the Newton–Raphson process and can be computed once per load step.

Explicit dynamic MPM with aggregation

Explicit dynamic implementations of the MPM with mesh aggregation require all operations that use the global mass matrix to be modified. This includes: (i) determining the nodal velocities based on the momentum of the material points; and (ii) calculation of the nodal accelerations from the dynamic force residual. The first operation is required twice for MUSL implementation, at the start of the time step and immediately prior to updating the material point stresses.

Nodal velocities: The aggregated global mass matrix is defined as

$$[M^{\mathrm{A}}] = [E]^T[M][E] \tag{15}$$

where $[M]$ is the global mass matrix (consistent or lumped). A lumped mass matrix is used in all of the examples presented in this paper. The velocities of the interior nodes are obtained using

$$\{v^{\mathrm{int}}\} = [M^{\mathrm{A}}]^{-1}\left\{[E]^T \mathop{\mathbf{A}}_{\forall p} m_p\{v_p\}[S_{vp}]\right\} \tag{16}$$

and the full nodal velocity vector recovered using

$$\{v\} = [E]\{v^{\mathrm{int}}\}. \tag{17}$$

Nodal accelerations: The accelerations of the interior the nodes are obtained from

$$\{\dot{v}^{\mathrm{int}}\} = [M^{\mathrm{A}}]^{-1}\{f_R^{\mathrm{A}}\}, \tag{18}$$

where $\{f_R^{\mathrm{A}}\}$ is the residual external force vector (the difference between the external and internal forces). The full nodal acceleration vector recovered using

$$\{\dot{v}\} = [E]\{\dot{v}^{\mathrm{int}}\}. \tag{19}$$

It is important to note that the process of aggregation results in off diagonal terms in $[M^{\mathrm{A}}]$, even if $[M]$ is a lumped (i.e. diagonal) matrix. This means that it is necessary to apply a linear solver when determining the nodal velocities and accelerations rather than simply dividing the forcing term by the lumped mass of the associated degree of freedom. When using a lumped matrix, $[M]$, the aggregated mass matrix, $[M^{\mathrm{A}}]$, will be block diagonal for the interior degrees of freedom that support external, aggregated nodal values. The aggregated mass matrix, $[M^{\mathrm{A}}]$, will be symmetric positive definite provided that the lumped mass matrix does not contain any negative mass values, which could only occur if the basis functions are negative at material point locations. Given that $[M^{\mathrm{A}}]$ is block diagonal, with the majority of the diagonal elements being 1×1 lumped entries (the degrees of freedom that are not attached to element aggregations), the matrix can be inverted block by block. This substantially reduces the cost of the inversion of $[M^{\mathrm{A}}]$ compared to a consistent mass matrix, which will have higher connectivity. It also makes the use of preconditioners unnecessary as the aggregated mass matrix scales very well; the solver time will be proportional to the maximum block size, rather than being strongly related to the problem size.[3] As mentioned, interior degrees of freedom not part of element aggregations will have lumped entries and their associated velocities/accelerations can be determined directly from the lumped mass.

[3] The solver cost, will be $\approx\mathcal{O}(n)$ where n is the global system size, with an additional, but small, cost (i.e. the cost of an efficient solver, such as Cholesky factorisation – $\mathcal{O}(n_B^3)$, where n_B is the block size) for resolving each aggregated block separately, which can be performed in parallel.

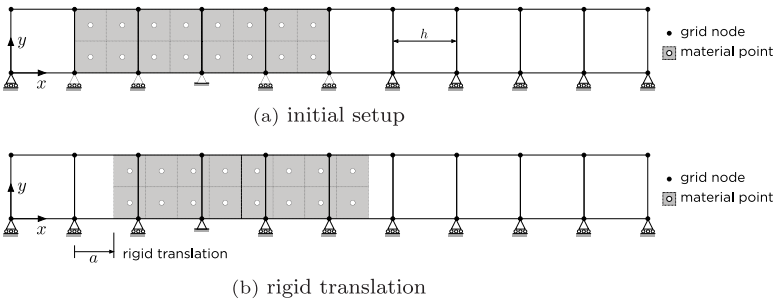


Fig. 6. Rigid translation: problem definition.

It is noted that there appears to be a choice regarding when to apply lumping versus aggregation of the mass matrix. For example, it may seem advantageous to aggregate the consistent (or lumped) mass matrix and then apply diagonal lumping to recover a trivial-to-solve linear system. However, it is not appropriate to lump the aggregated mass matrix as it has the potential to introduce negative values on the leading diagonal of the resulting matrix. This means that the matrix is no longer guaranteed to be positive definite, which is caused by the negative entries in the extension matrix, $[E]$. These issues are revealed in Appendix B using the case shown in Fig. 5. This example also shows that the only way to avoid negative definite matrices consists of aggregating the considered mass matrix (either lumped or consistent) and not altering it further.

Global versus local aggregation

Note that in the above algorithm the extension matrix, $[E]$, has been expressed in terms of a global quantity however it is entirely possible to implement aggregation at a local aggregated element level and assemble the resulting aggregated quantities. In fact, entries in the constraint matrix, $[C]$, are by their nature, local operations considering the interaction between an exterior (or aggregated) node and its parent aggregated element. However, depending on the context, it might be better to assemble this matrix globally and multiply it with the other global matrices (e.g., the stiffness or the mass matrix) or swap the order of this procedure, that is multiply matrices locally and assemble the result in a global one. For implicit simulations, the former option seems preferable, as the matrix $[E]$ is constant within a load step, while the stiffness matrix changes at every iteration. Conversely, the better option for explicit simulations may be multiplying the local extension matrix by the local mass matrix, as this is performed only once per time step and avoids global multiplication operations. It is also acknowledged that details of the preferred implementation approach will also depend convenience within the base MPM code, and a local or global approach will not change the resulting system of equations. Implementation efficiency is another important consideration and the interested reader is referred to [43] for a highly efficient parallel implementation of AgFEM on distributed memory architectures.

4. Numerical examples

This section includes six numerical examples of increasing complexity to demonstrate the ability of the AgGIMPM approach to produce stable simulations of large deformation problems with non-linear material behaviour.

4.1. Rigid translation

This numerical example investigates the evolution of the condition number of the global stiffness matrix and global mass matrix with rigid body motion, or rather the range of condition numbers depending on the exact interaction of the physical body with the background mesh. A 4×1 m plane strain physical domain was discretised by 16 material points, each representing 0.25 m^3 of material with a Young's modulus of $E = 1$ Pa, a Poisson's ratio of $\nu = 0$ and a density of $\rho = 1$ kg/m^3. A background grid comprised of $h = 1$ m square bi-linear quadrilateral elements was used to construct the global matrices. The background grid nodes were constrained as shown in Fig. 6, where the grid nodes at $y = 0$ m are constrained vertically and the node at $x = 3$ m, $y = 0$ m is also constrained horizontally. The focus of this example is on the impact of the material point positions on the global matrix conditioning, therefore the translation of the physical domain is imposed directly at the material points rather than being driven by any grid-based boundary conditions; the stiffness and mass matrix are assembled, but no governing equations are solved using the matrices.

Fig. 7 shows the evolution of the stiffness matrix condition number, $\kappa([\cdot])$ the ratio of the largest to smallest eigenvalue, of the reduced stiffness matrices (the stiffness matrices after the rows/columns associated with the constrained and inactive degrees of freedom have been removed) as the physical domain is translated by $a/h = 4$ over 5000 steps for the standard GIMPM, Ghost-stabilised GIMPM (GsGIMPM) [19], where the stabilisation parameter is set to the Young's modulus of the material, and Aggregated GIMPM (AgGIMPM). Note that for $a/h \in [3, 4]$ the body is no longer constrained by the boundary condition that restricts rigid body motion in the x-direction and all of the method have conditions numbers above 1×10^{16}.

Condition numbers above the reciprocal of the precision of the machine being used will cause accuracy problems when solving the linear system of equations. It is clear from Fig. 7 that the GIMPM suffers from instabilities when the boundary physical domain

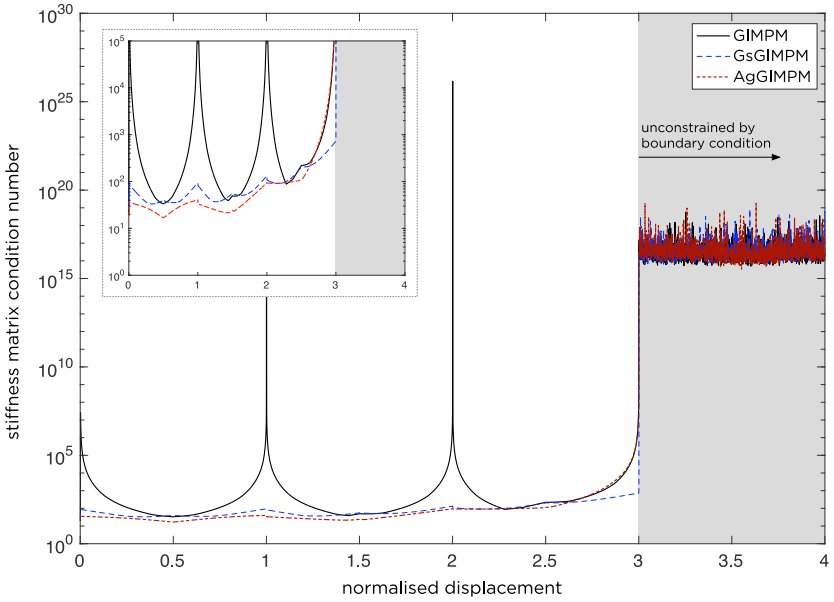
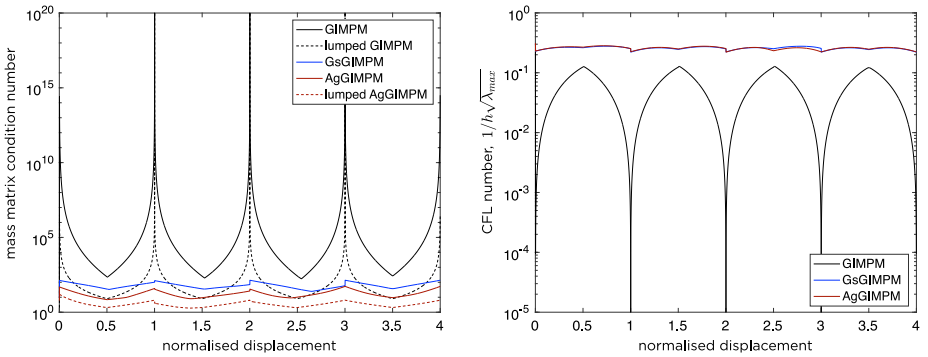


Fig. 7. Rigid translation: stiffness matrix conditioning (inset figure shows detail for $\kappa \in [10^0, 10^5]$).



(a) mass matrix (b) CFL

Fig. 8. Rigid translation: (a) mass matrix condition number and (b) CFL number.

is very close to a grid boundary. Spikes in the condition number above 10^{25} are seen at $a/h = 1$ and 2, which will cause problems for any double precision linear solver. This issue is corrected by both ghost stabilisation and mesh aggregation, with similar resulting condition numbers from both methods. The biggest difference between the two approaches is seen for $a/h \in [2.7, 3.0]$, which is due to the difference in the influence of the boundary condition. The ghost stabilisation weakly constrains the gradient of the solution at the boundary of the physical domain, which has the effect of increasing the influence of the x-direction boundary condition.

The condition of the mass matrix has important consequences for dynamic explicit and implicit implementations of the MPM. The evolution of the condition number of the reduced mass matrix is shown in Fig. 8(a) for the standard GIMPM, GsGIMPM [19] (stabilisation parameter set to $\gamma_M = \rho/4$) and AgGIMPM, along with the lumped matrix for GIMPM and AgGIMPM. As with the stiffness matrix, aggregation improves the condition number of the mass matrix and removes the spikes seen in the condition number of the GIMPM consistent and lumped mass matrices and offers a slight improvement over the GsGIMPM. Note that ghost stabilisation does not change the lumped mass matrix as the sum of the stabilisation terms over a given degree of freedom is equal to zero. This is an important distinction between mesh aggregation and ghost stabilisation; mesh aggregation can be applied to explicit MPMs that use a lumped mass matrix whereas ghost stabilisation has zero impact.

The conditioning of the stiffness and mass matrices have a combined impact on the maximum time step size for a dynamic analysis. The maximum time step size for an explicit time stepping algorithm is limited by the Courant–Friedrichs–Lewy (CFL) number, C_{CFL}, via

$$\Delta t \leq \alpha C_{\text{CFL}} \min(h),$$

Table 1
Rigid translation: minimum and maximum CFL numbers.

	GIMPM	GsGIMPM	AgGIMPM
$\max(C_{\text{CFL}})$	2.25×10^{-1}	3.03×10^{-1}	3.13×10^{-1}
$\min(C_{\text{CFL}})$	1.30×10^{-14}	2.21×10^{-1}	2.21×10^{-1}

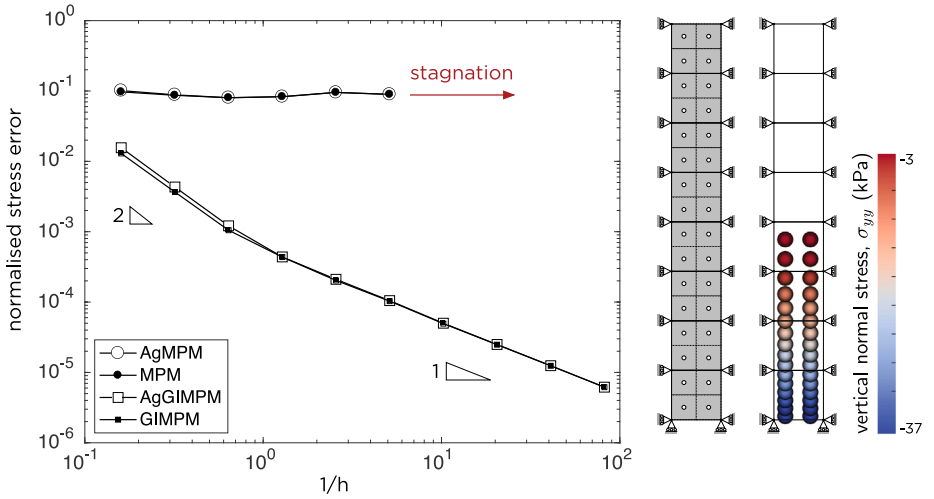


Fig. 9. Column under self weight: convergence with mesh refinement, and initial problem setup and deformed material point positions coloured according to the vertical stress, σ_{yy} for GIMPM with $h = 6.25$ m. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where α is a constant that depends on the time stepping algorithm [44]. The CFL number can be determined from

$$C_{\text{CFL}} = \frac{1}{h\sqrt{\lambda_{\max}}},$$

where $\lambda_{\max}$ is the largest eigenvalue of the generalised eigenvalue problem $[K]\{x\} = \lambda[M]\{x\}$. Ideally C_{CFL} should be independent of the position of the physical domain relative to the background mesh. Ill-conditioning of the mass and stiffness matrices can cause severe time step limitations [45], which is demonstrated in Fig. 8(b) and Table 1 for the standard GIMPM without stabilisation or aggregation where C_{CFL} reduces by 13 orders of magnitude when the boundary physical domain is very close to a grid boundary. This problematic behaviour is corrected by both the ghost-stabilisation and the mesh aggregation, with a similar CFL range for both approaches.

4.2. Quasi-static column under self weight

This example considers the implicit analysis of a one dimensional quasi-static elastic column with an initial height of $l_0 = 50$ m under its own self weight. The material had a Young's modulus of $E = 10$ kPa and a Poisson's ratio of $\nu = 0$. The background mesh was comprised of square elements with roller boundary conditions on the base and sides and the column was discretised by a 2 by 2 grid of equally spaced material points in each initially populated background grid element. A body force of 800 N/m^2 ($g = 10$ m/s^2 and an initial density of $\rho_0 = 80$ kg/m^3) was applied over 40 equal load steps. The magnitude of the load causes the column to compress to approximately half of its initial height, as shown on the right of Fig. 9 for an analysis with a background size of $h = 6.25$ m using the GIMPM.

The analytical solution for the normal stress in the vertical (y) direction for this problem is

$$\sigma^a_{yy} = \rho_0\, g(l_0 - Y), \tag{20}$$

where Y is the original position of the point in the body and l_0 is the original height of the column. Fig. 9 shows the convergence with background mesh refinement whilst maintaining 4 material points per initially populated element of the standard MPM and the GIMPM with and without aggregation. The reported normalised stress error is

$$\text{error} = \sum_{p=1}^{n_p} \frac{|\sigma^p_{yy} - \sigma^a_{yy}(Y_p)|\; V^0_p}{(g\rho_0 l_0)V_0}, \tag{21}$$

where $V_0 = \sum V^0_p$ is the initial volume of the column and σ^p_{yy} is the vertical stress at each of the material points.

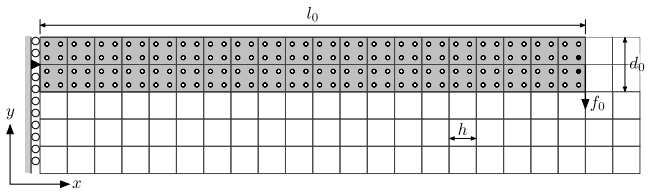


Fig. 10. Elastic beam: initial discretisation and boundary conditions with $h = 0.5$ m and 2^2 MPs/element (only part of the background mesh is shown).

Table 2
Elastic beam stability for GIMPM, GsGIMPM ($\gamma_K = E$) and AgGIMPM: for entries marked ✗, the number indicates the final stable load step; for entries marked ✓, the numbers indicate the total and (maximum) number of Newton iterations for the analysis.

(a) GIMPM: total and (max.) iterations or final stable iteration

MPs/elem.	$h = 0.250$	$h = 0.125$	$h = 0.0625$
2^2	✗ 11	✗ 1	✗ 18
3^2	✓ 215(8)	✗ 5	✓ 222(6)
4^2	✗ 26	✓ 215(5)	✗ 15
5^2	✓ 207(5)	✓ 214(5)	✗ 15
6^2	✓ 207(5)	✓ 216(6)	✓ 217(5)
mean	209.7(6)	215.0(5.3)	219.5(5.5)

(b) GsGIMPM: total and (max.) iterations

MPs/elem.	$h = 0.250$	$h = 0.125$	$h = 0.0625$
2^2	✓ 203(5)	✓ 205(5)	✓ 206(5)
3^2	✓ 204(5)	✓ 203(5)	✓ 201(5)
4^2	✓ 204(5)	✓ 201(5)	✓ 202(5)
5^2	✓ 203(5)	✓ 202(5)	✓ 202(5)
6^2	✓ 204(5)	✓ 202(5)	✓ 202(5)
mean	203.6(5)	202.6(5)	202.6(5)

(c) AgGIMPM: total and (max.) iterations

MPs/elem.	$h = 0.250$	$h = 0.125$	$h = 0.0625$
2^2	✓ 203(5)	✓ 204(5)	✓ 205(5)
3^2	✓ 203(5)	✓ 204(5)	✓ 206(5)
4^2	✓ 203(5)	✓ 204(5)	✓ 206(5)
5^2	✓ 202(5)	✓ 204(5)	✓ 205(5)
6^2	✓ 203(5)	✓ 204(5)	✓ 205(5)
mean	202.8(5)	204.0(5)	205.4(5)

As widely documented [19,35,38,39], Fig. 9 demonstrates that the standard MPM, irrespective of aggregation, does not converge with mesh refinement due to the cell crossing instability. The GIMPM and the AgGIMPM converge with mesh refinement, with only minor differences in the error values for a given mesh size, h. The convergence rate is initially quadratic, but reduces to a linear rate as the mesh is refined. This is due to the accuracy of the solution is limited by the lowest order of the basis function; GIMPM basis functions are linear when a material point is fully within one element and quadratic when the particle domain overlaps multiple elements. The limiting theoretical convergence rate of the stress error is the same as the basis order. Fig. 9 confirms that aggregation has a limited impact on the accuracy of the solution as the mesh is refined. It is important to note that due to the constrained nature of the problem (1D deformation) that stability is not an issue in this case. The impact of aggregation on a more challenging problem is explored by the next example.

4.3. Quasi-static elastic cantilever beam

This example considers the implicit quasi-static analysis of a large deformation beam as a challenging problem for the material point method due to numerous small overlaps between the material point domains and the background grid. The elastic cantilever beam was subjected to a point load at its free end and modelled using the generalised interpolation material point method. The beam was $l_0 = 10$ m long and $d_0 = 1$ m deep and the material had a Young's modulus of 12 MPa, a Poisson's ratio of 0.2 and was assumed to be weightless ($\rho = 0$). The $f_0 = 100$ kN end point load was split between the two material points closest to the end of the beam either side of the neutral axis and applied over 50 equal load steps. The initial discretisation of the beam is shown in Fig. 10 with $h = 0.5$ m and with 2^2 material points per initially populated background grid cells. The loaded material points are shown by the black-filled circles.

Table 2 provides information on the stability of the *quasi*-static analysis with different numbers of generalised interpolation material points per element when aggregation, or ghost stabilisation, is and is not applied. The table reports different information depending on if an analysis was able to complete all load steps:

Table 3
Elastic beam run time distribution (in seconds) for AgGIMPM with 3^2 MPs per element.

process	AgGIMPM		
	$h = 0.250$	$h = 0.125$	$h = 0.0625$
initialisation	0.072	0.093	0.169
MP search & basis calc.	6.166	27.247	175.810
form $[K]$ and $\{f^{\text{int}}\}$	9.943	40.257	168.046
form $\{f^{\text{ext}}\}$	0.154	0.553	2.297
mesh aggregation	**0.291**	**1.014**	**5.194**
linear solver	0.186	0.794	4.205
MP update	0.392	0.568	6.212
miscellaneous	0.141	0.568	2.397
total	**17.35**	**72.01**	**364.33**

✗ unstable analysis: the number indicates the final stable (converged) load step; and
✓ stable analysis: the numbers indicate the total and (maximum) number of Newton–Raphson iterations for the analysis.

The tolerance on the normalised out of balance force was set to 1×10^{-6}, with a maximum of 10 iterations in the Newton–Raphson (NR) algorithm. Consistent with [19], for the GsGIMPM analysis the stabilisation parameter was set to the Young's modulus of the material, $\gamma_K = E$. The key observation from Table 2 is that all of the AgGIMPM and GsGIMPM analyses completed whereas seven out of the 15 GIMPM analyses failed to apply the full load. It is difficult to predict which GIMPM analyses will complete and which will fail, as there is no clear trend from the pattern of the failed simulations beyond that 2^2 material points is insufficient. Another observation is that both AgGIMPM and GsGIMPM are relatively insensitive to the size of the background mesh and the number of material points in terms of the total and maximum number of NR iterations. The total number of iterations for the standard GIMPM tends to increase with mesh refinement.

Table 3 details the run time distribution for the AgGIMPM with three mesh sizes with 3^2 MPs per element. The aggregation step (highlighted in bold) accounts for between 1.4 and 1.6% of the overall run time and scales approximately proportional to the number of degrees of freedom of the background mesh (or $1/h^2$). Note that this is the only significant additional algorithm that is required to implement the mesh aggregation approach versus a standard GIMPM analysis. The *linear solver* step is also modified to calculate the aggregated stiffness matrix and residual force, Eq. (13), but this has almost zero impact on the run time, being balanced by the solution of a smaller, aggregated, linear system. For comparison, the run time for forming the ghost stabilisation matrix with $h = 0.0625$ m and 3^2 MPs per element was 4.70 s and also scales approximately proportional to $1/h^2$.

The normalised force–displacement response of the 15 AgGIMPM simulations is shown in Fig. 11 along with the non closed-form analytical displacement solution of Molstad [46] (white-filled markers) for both the horizontal and vertical displacements.[4] The plotted displacement for the material point analyses is the mean of the displacement of the two material points closest to the end of the beam either side of the neutral axis (i.e. the mean displacement of the loaded material points). The resolution of the figure means that the responses of the 15 simulations are essentially the same and show good agreement with the analytical solutions, especially for the horizontal displacement. Note that the closed-form solution provided by Molstad [46] makes a number of assumptions, namely: the direction of the tip load remains vertical, the length of the deformed beam is the same as the original length, and the beam has constant flexural rigidity. The first assumption is consistent with the numerical analyses in this paper, but the other two assumptions can explain the minor discrepancies between the MPM results and Molstad's solution. A simulation for the standard GIMPM with $h = 0.25$ m and 3^2 material points per element is also shown by the black-filled discrete points. The vertical displacement response is very similar to the AgGIMPM simulations. A more significant difference is seen for the horizontal displacement, with the AgGIMPM predicting a stiffer response than GIMPM. GsGIMPM results are not included on the figure as the global force–displacement response of the simulations is essentially the same as that of the AgGIMPM analyses. The right of Fig. 11 shows the final deformed configuration of the material points, which have been coloured according to the vertical normal stress, σ_{yy}, for $h = 0.125$ m and 3^2 material points per element.

4.4. Quasi-static elasto-plastic collapse

This example considers an implicit quasi-static analysis of the collapse of an elasto-plastic rectangular body of material subject to a gravitational body force. The initial discretisation of the problem is shown in Fig. 12 with $l_0 = 8$ m, $h = 1$ m and 2^2 material points per initially filled background cell (the grey-shaded region). Note that the size of the background grid and the number of material points per element are different in Fig. 12 than used in the analysis to help with clarity of the figure. Roller boundary conditions are imposed on the base and left-hand side of the background grid. The material is modelled using a hyper-elastic Hencky, perfectly-plastic von Mises constitutive formulation with the following yield function

$$f = \sqrt{s_{ij} s_{ji}} - \rho_y, \qquad \text{where} \qquad s_{ij} = \sigma_{ij} - \frac{\sigma_{kk}}{3}\delta_{ij}$$

[4] Note that Molstad [46] also includes large deformation finite element simulations of this cantilever beam problem and the boundary conditions adopted in this paper are consistent with those analyses.

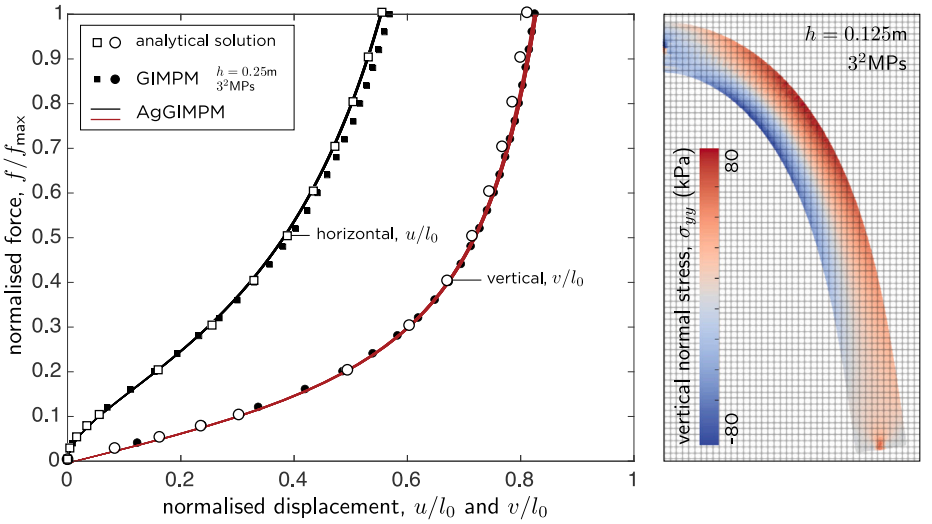


Fig. 11. Elastic beam: normalised force–displacement response for all AgGIMPM analyses and a single GIMP analysis ($h = 0.25$ m, 3^2 MPs) plus the analytical solution [46]; and final deformed beam for AgGIMPM with $h = 0.125$ m and 3^2 MPs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

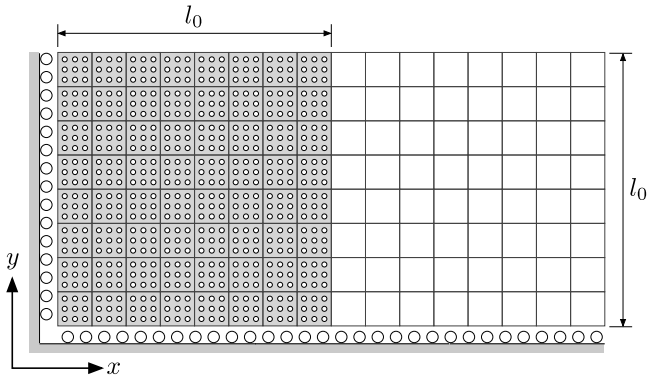
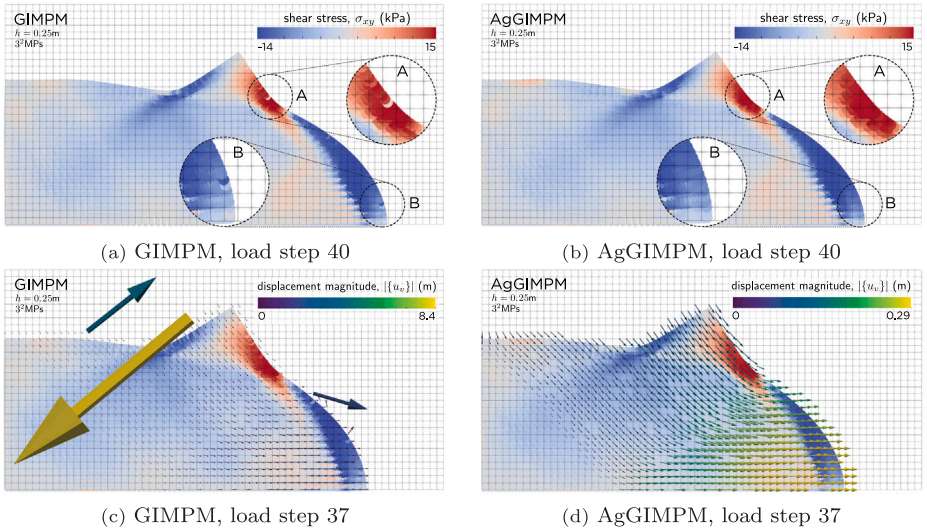


Fig. 12. Elasto-plastic collapse: boundary conditions and illustrative discretisation with $h = 1$ m and 3^2 MPs/element. Note that the actual analyses used $h = 0.25$ m.

and a yield strength of $\rho_y = 20$ kPa, a Young's modulus of 1 MPa and Poisson's ratio of 0.3. The analysis was conducted with the standard GIMPM, GsGIMPM and AgGIMPM with a mesh size of $h = 0.25$ m and 3^2 material points per initially populated element. A body force of 10 kN/m^3 (density of 1000 kg/m^3 and gravitational acceleration of 10 m/s^2) is applied over 40 equal loadsteps. The ghost stabilisation penalty parameter for the GsGIMPM was set to $\gamma_K = \frac{E}{100}$.

Fig. 13 shows the deformed material point positions at the end of the analysis shaded according to the material point's shear stress, σ_{xy}, value for (a) the GIMPM and (b) the AgGIMPM. Two areas are highlighted in the figure by the dashed circles labelled A and B, where there are spurious stress oscillation at the boundary of the domain for the GIMPM analysis that are not present in the AgGIMPM analysis. These spurious stress oscillations are caused by artificially large incremental displacements of the nodes of the background grid in the GIMPM analysis. These displacements can be seen in Figs. 13(c) and (d), where vectors of incremental nodal displacements are shown scaled according to their magnitude for load step 37 as a representative example. The incremental nodal displacement solution, $\{\Delta d\}$, contains both the internal, $\{\Delta d^{\text{int}}\}$, and aggregated (or external) displacements, $\{\Delta d^{\text{ext}}\}$, that have been recovered from the internal displacements using (11). Therefore all active degrees of freedom have an incremental displacement solution vector. Note that size of the vectors in Fig. 13(d) are scaled by a factor of two to better visualise the incremental displacement of the nodes; no scaling is applied to Fig. 13(c).

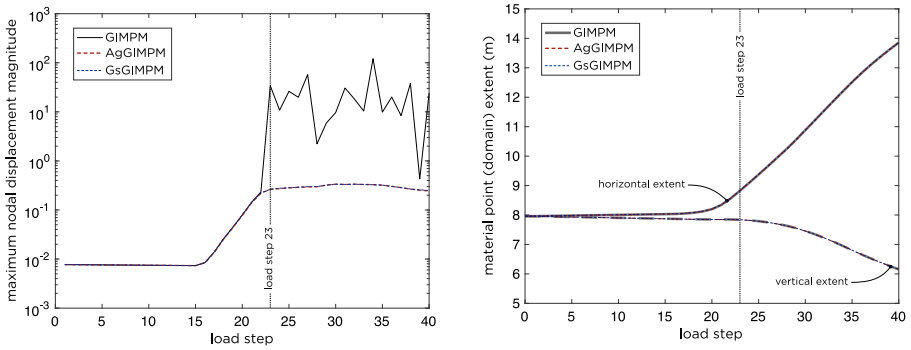
Large nodal displacements in seemingly random directions are observed close to the boundary in Fig. 13(c) for the GIMPM analysis, caused by the small cut issue, and it is these displacements that cause the random fluctuations of the stress field near the boundary of the physical domain. The maximum displacement magnitude of the nodes of the background mesh, $\max(|\{u_v\}|)\ \forall\ v$, for all load steps is shown in Fig. 14(a) for GIMPM, GsGIMPM and AgGIMPM. Little difference is seen between the analysis until load step 23, where the maximum value for the GIMPM analysis increases by approximately two orders of magnitude, this coincides with



(a) GIMPM, load step 40 (b) AgGIMPM, load step 40

(c) GIMPM, load step 37 (d) AgGIMPM, load step 37

Fig. 13. Elasto-plastic collapse: shear stress distribution at the end of the analysis for (a) GIMPM and (b) AgGIMPM and nodal displacement vectors at load step 37 for (c) GIMPM and (d) AgGIMPM. $h = 0.25$ m and 3^2 MPs/element for all cases. See supplementary information for an animation of all load steps.



(a) nodal displacement magnitude (b) physical domain extent

Fig. 14. Elasto-plastic collapse: (a) maximum displacement magnitude of the nodes of the background mesh, $\max(|\{u_v\}|)\ \forall\ v$, and (b) extent of the physical domain, $\max(\{x_p\})\ \forall\ p$.

the start of the large deformation collapse mechanism and significant non-mesh conformity of the physical body. This is confirmed by Fig. 14(b), which shows the extent of the physical domain based on the maximum coordinates of the material points, $\max(\{x_p\})\ \forall\ p$. GIMPM, GsGIMPM and AgGIMPM predict very similar overall deformation patterns, with only a 0.055% and 0.060% difference in the horizontal and vertical extents, respectively, across the three methods.

It is also worth noting that the GIMPM analysis had a total of 240 iterations whereas the AgGIMPM analysis had a total of 212 iterations (a 12% reduction), all of which had a maximum number of 8 iterations within any load step. Therefore, not only are the spurious nodal displacements, and associated oscillations in the stress field, removed but the cost of the analysis is also reduced despite requiring additional calculation steps.[5]

4.5. Quasi-static soil–structure interaction: cone penetration test

The final implicit quasi-static example is focused on axi-symmetric soil–structure interaction in the form of a Cone Penetration Test (CPT) using the rigid-deformable body contact approach of Bird et al. [47]. The predicted cone resistance is compared to experimental data obtained in a geotechnical centrifuge by Davidson et al. [48] and available from Cerfontaine et al. [49].

[5] 9.4% reduction: GIMPM run time 121.3 s, AgGIMPM run time 109.9 s. Software: MATLAB R2023b on MacOS Sonoma V. 14.7.1. Hardware: Apple M3 Pro, 18 GB RAM.

Table 4
CPT simulation information for 3^2 MPs/element and 0.75 MRF (Mesh Refinement Factor). The ✗ indicate analyses that failed to complete.

	final p/r_c	no. contact steps	time (s)	$\frac{\text{time}}{\text{steps}}$ (s)
GIMPM	0.881 ✗	27	–	–
GsGIMPM ($\gamma_K = E_{\min}$)	9.600	97	1407.0	14.51
GsGIMPM ($\gamma_K = \frac{E_{\min}}{10}$)	9.650	98	1516.7	15.47
GsGIMPM ($\gamma_K = \frac{E_{\min}}{100}$)	9.600	104	1718.3	16.52
GsGIMPM ($\gamma_K = \frac{E_{\min}}{1000}$)	1.528 ✗	43	–	–
AgGIMPM	9.675	107	1843.2	17.23

The material was modelled using a hyper-elastic Hencky, perfectly-plastic non-associated flow constitutive formulation with a Drucker–Prager (DP) yield function of the form

$$f = \varrho - \alpha\left(\xi - \xi_c\right), \qquad \text{where} \qquad \varrho = \sqrt{s_{ij}s_{ji}} \qquad \text{and} \qquad \xi = \frac{\sigma_{kk}}{\sqrt{3}}.$$

$\alpha = \tan(\phi)$ is the opening angle of the DP cone, $\xi_c = c\sqrt{3}\cot(\phi)$ is the tensile hydrostatic extent of the yield surface (the position of the cone apex on the hydrostatic axis), ϕ is the friction angle and c the cohesion. The slope of the plastic potential surface is controlled by the dilation angle, ψ, associated plastic flow is recovered when $\psi = \phi$.

The Young's modulus of the material varies with depth based on the relationship proposed by Schanz et al. [50]

$$E = E^*\left(\frac{\sigma_v K_0}{p^*}\right)^{m_E} \qquad \text{where} \qquad \sigma_v = d_p\rho \tag{22}$$

is the vertical stress at the material point based on its depth within the sample, d_p, $K_0 = 1 - \sin(\phi)$ is the coefficient of lateral earth pressure at rest, E^* is the reference Young's modulus, $p^* = 100$ kPa is the reference atmosphere pressure and m_E is an exponent controlling the variation of stiffness with depth (see Brinkgreve et al. [51] for details). A sand with a Relative Density (RD) of 82% is considered, where the associated material properties are: density $\rho = 1820$ kg/m^3, reference Young's modulus $E^* = 49.2$ MPa, Poisson's ratio $\nu = 0.3$, stiffness exponent $m_E = 0.56$, friction angle $\phi = 38.3°$, dilation angle $\psi = 8.3°$ and cohesion $c = 0.3$ kPa. Most of these parameters are obtained from the relative density relationships provided by Brinkgreve et al. [51]. Two exceptions are the Poisson's ratio, which is taken as a representative value for the type of sand, and the cohesion, which is set to the very small value of 0.3 kPa as a balance between numerical stability and the fact that this dry sand will essentially have zero cohesion.

An indicative background mesh is shown in Fig. 15, where the overall size of the initially populated background mesh is 12.5 m in diameter by 25.0 m high. The mesh is graded from a uniform fine 1.0 m by 10.0 m region (the grey shaded region in Fig. 15) in the immediate vicinity of the CPT penetration path towards the radial and lower limits of the mesh using a power law scaling rule. Roller boundary conditions were applied to the lower and outer radial limit of the mesh. Note that no boundary condition is applied at the line of symmetry, where $r = 0$ m, as this would artificially constrain the movement of the material points when interacting with the rigid body. Any material points moving in the negative radial direction close to the line of symmetry will generate compressive normal stresses in the circumferential direction that will act to limit the movement of the material and stop it crossing the line of symmetry. The CPT probe had a tip angle of 60° and radius of 0.4 m, which is scaled from the geotechnical centrifuge radius of 8 mm by the 50 g acceleration applied in the centrifuge tests. The CPT probe was moved 4 m vertically downwards over 100 steps, however the actual value of penetration will depend on the material point discretisation as penetration is measured from the first contact interaction with a material point. The level of penetration will also depend on the number of step size reductions required by the solver, as the displacement increment of the CPT is automatically halved if equilibrium is not obtained within a given load step. Once equilibrium is achieved the following step size is doubled, but with a limit of the original equal displacement step size (0.04 m in this case). All analyses will continue until the CPT probe has been displaced by at least 4 m.

The CPT, due to its high stiffness compared to the soil, was assumed to be a rigid body and the frictional penalty contact approach of Bird et al. [47] was adopted to model the soil-rigid body interaction. In this approach the corners of the GIMPM domains are used to detect contact between the material points and the rigid body, which is discretised into a number of linear segments (or triangular facets in 3D, [52]). Overlaps between material point domains and the rigid body generate penalty reaction forces proportional to the penetration (or overlap) length, which are multiplied by user specified penalty parameters in the normal and tangential directions. The penalty forces are imposed on the overlapping corner of the material point domain before being mapped to the nodes of the background grid and included in the solution of the non-linear equilibrium equation. In the case of frictional interfaces, the maximum force in the tangential direction is limited by the normal reaction force and a coefficient of friction between the soil and the rigid body. The contact the normal and tangential penalty parameters were set to $20A_pE_p$ and $10A_pE_p$, respectively, where E_p is the Young's modulus of the material point and A_p is area of contact associated with the material point, which is taken to be diameter of that circle that encompasses the material point domain multiplied by the appropriate length in the out of plane direction (see Bird et al. [47] for details). The coefficient of friction between the material points and the rigid body was set to 0.3.

Fig. 16 shows the normalised penetration resistance (f_c/E^*, where f_c is the vertical resistive force acting on the CPT probe) versus normalised penetration depth (p/r_c, where p is the penetration and r_c the cone radius) response for the standard GIMPM,

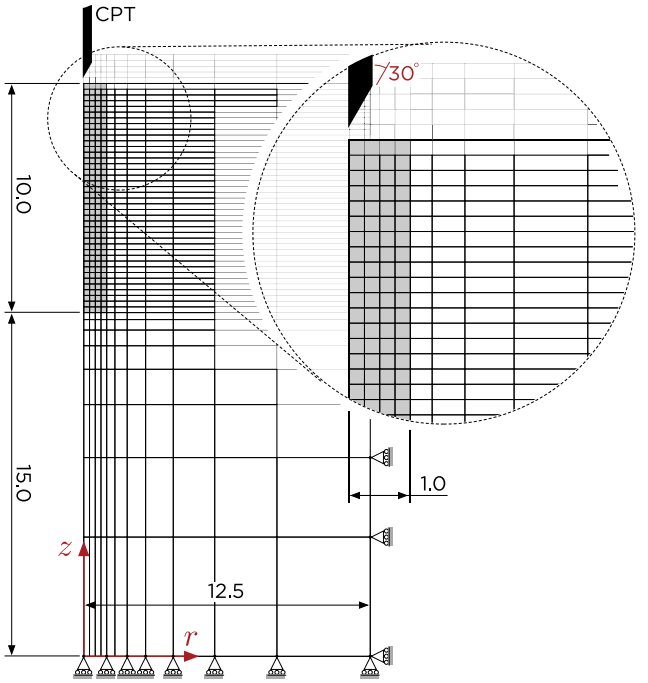
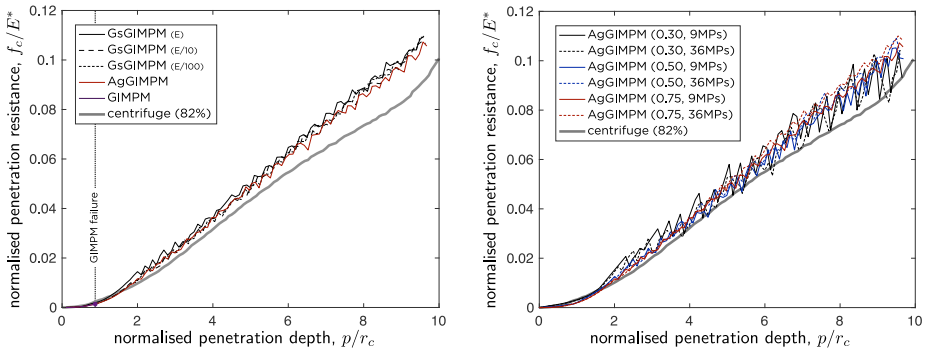


Fig. 15. CPT: indicative discretisation (dimensions in m).



(a) 0.75 MRF, 3^2MPs/element (b) AgGIMPM parameter variation

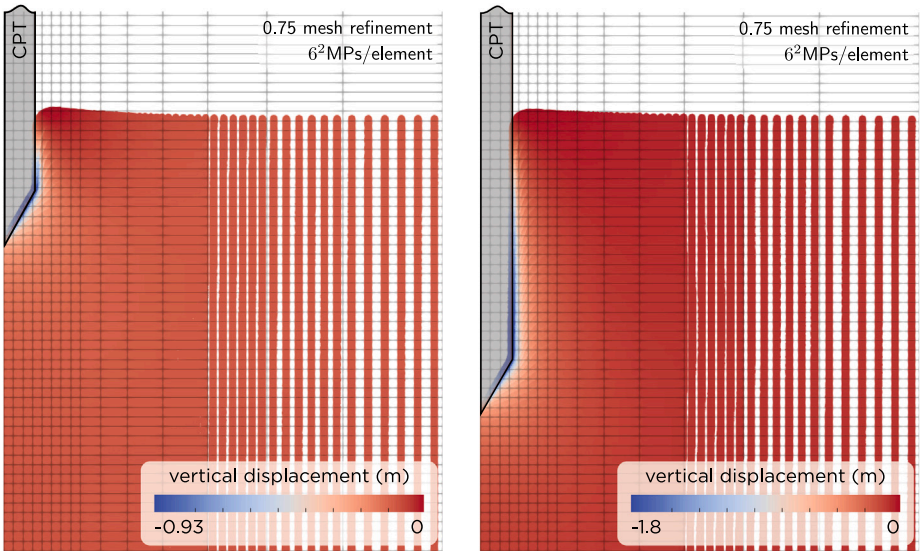
Fig. 16. CPT: normalised penetration resistance.

GsGIMPM and AgGIMPM for with various simulation parameters along with the centrifuge result. In Fig. 16(a) responses are compared between the methods with a fixed background mesh and material point discretisation. The standard GIMPM analyses fails to converge beyond a normalised penetration depth of 0.88 due to instabilities associated with the small cut problem. Fig. 16(a) also includes GsGIMPM results for three levels of penalty parameter ($\gamma_K = E_{\min}$, $\frac{E_{\min}}{10}$ and $\frac{E_{\min}}{100}$), which are very similar but behave differently in terms of their stability, as summarised in Table 4. The $\gamma_K = \frac{E_{\min}}{1000}$ result is not included in Fig. 16(a) as the simulation failed to complete. It can be seen that the simulations require more step size reductions and take longer with reducing stabilisation parameter; the GsGIMPM results approach the AgGIMPM number of steps and run time as the stabilisation is reduced. The GsGIMPM analysis with $\gamma_K = \frac{E_{\min}}{1000}$ failed to converge beyond a normalised penetration of 1.53, with the additional stiffness insufficient to mitigate the small cut issue. It can also be observed in Fig. 16(a) that, as expected, reducing the ghost penalty parameter leads to a slightly less stiff response. The AgGIMPM result is slightly less stiff than the Ghost stabilised simulations, with both approaches providing a good prediction of the centrifuge result.

Fig. 16(b) shows the normalised penetration resistance versus normalised penetration depth response for AgGIMPM with different background meshes (Mesh Refinement Factor (MRF), a relative measure of refinement between the different meshes, of 0.3, 0.5 and 0.75, where larger numbers give finer meshes) and numbers of material points per element (3^2 and 6^2). All of the simulations produce a globally similar response but have varying levels of oscillations, caused by the discrete nature of material points coming

Table 5
CPT AgGIMPM simulation information, where MRF is the Mesh Refinement Factor.

MRF	0.30		0.50		0.75	
no. nodes	826		1456		2489	
no. elements	754		1350		2340	
$h_{\min}$ (m)	0.333		0.200		0.125	
MPs/element	3^2	6^2	3^2	6^2	3^2	6^2
total no. MPs	4914	19,656	8550	34,200	14,580	58,320
final p/r_c	9.650	9.600	9.688	9.600	9.675	9.644
no. contact steps	99	96	101	97	107	104
run time (s)	404.8	1394.3	880.6	2810.2	1843.2	7083.4



(a) $p/r_c = 4.34$ (b) $p/r_c = 9.64$

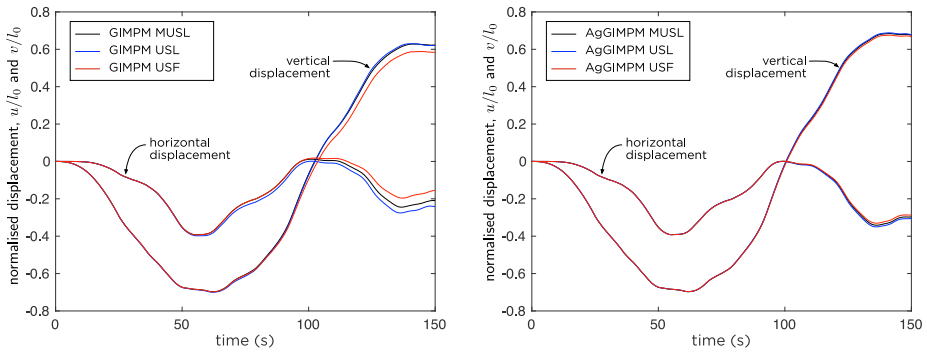
Fig. 17. CPT: deformed material point positions with 0.75 MRF and 6^2 MPs/element. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

into contact with the rigid CPT probe. The oscillations in the penetration resistance reduce with mesh and material point refinement as each material point is representing less volume and therefore generates less resistive force against penetration. As expected the run time of the simulations increase with mesh and material point refinement, as detailed in Table 5. It is also not surprising that the number of contact load steps (i.e. the number of times the step size is reduced) increases with mesh refinement as all simulations use the same maximum increment on the CPT probe vertical displacement.

The deformed material point positions for two different penetration depths are shown in Fig. 17 for the AgGIMPM analysis with a MRF of 0.75 and 6^2 MPs/element. The material points are coloured according to their vertical displacement, which is highly localised around the CPT. The impact of friction between the CPT and the material points can be observed by the blue region, showing material points being dragged downwards with the CPT as it penetrates the soil body.

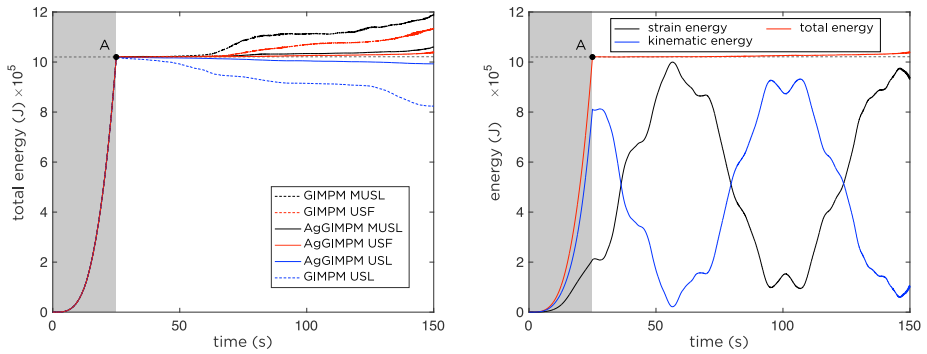
4.6. Dynamic elastic cantilever beam

This example considers the explicit forward Euler solution of an elastic cantilever beam under dynamic loading and is focused on the impact of mesh aggregation on stability, energy conservation and the quality of the stress field. The beam was $l_0 = 10$ m long and $d_0 = 1$ m deep and was subjected to a point load at its free end. The load was split between the two material points closest to the end of the beam either side of the neutral axis and increased linearly from zero to $f_0 = 500$ kN over the first 25 s of the analysis. Beyond 25 s the force was removed and the beam allowed to oscillate until the end of the analysis at 150 s, over a total of 6000 equal time steps ($\Delta t = 0.025$ s). All analyses were run on a 11 m×16 m background grid of bi-linear square elements with $h = 0.25$ m and 3^3 generalised interpolation material points per initially populated element. The material had a Young's modulus of 100 MPa, a Poisson's ratio of 0.2 and a density of $\rho = 7750 \times 10^3$ kg/m^3. The initial discretisation of the beam is shown in Fig. 10 with $h = 0.5$ m and with 2^2 material points per initially populated background grid cells. Note that the number of material points and the mesh size were selected for clarity of the illustration; $h = 0.25$ m and 3^2 material points were used for all analyses. The loaded material points are shown by the black-filled circles.



(a) GIMPM (b) AgGIMPM

Fig. 18. Dynamic elastic beam: MUSL, USL and USF normalised horizontal and vertical tip displacement response with $h = 0.25$ m, 3^2 MPs/element and 6000 time steps for (a) GIMPM and (b) AgGIMPM.



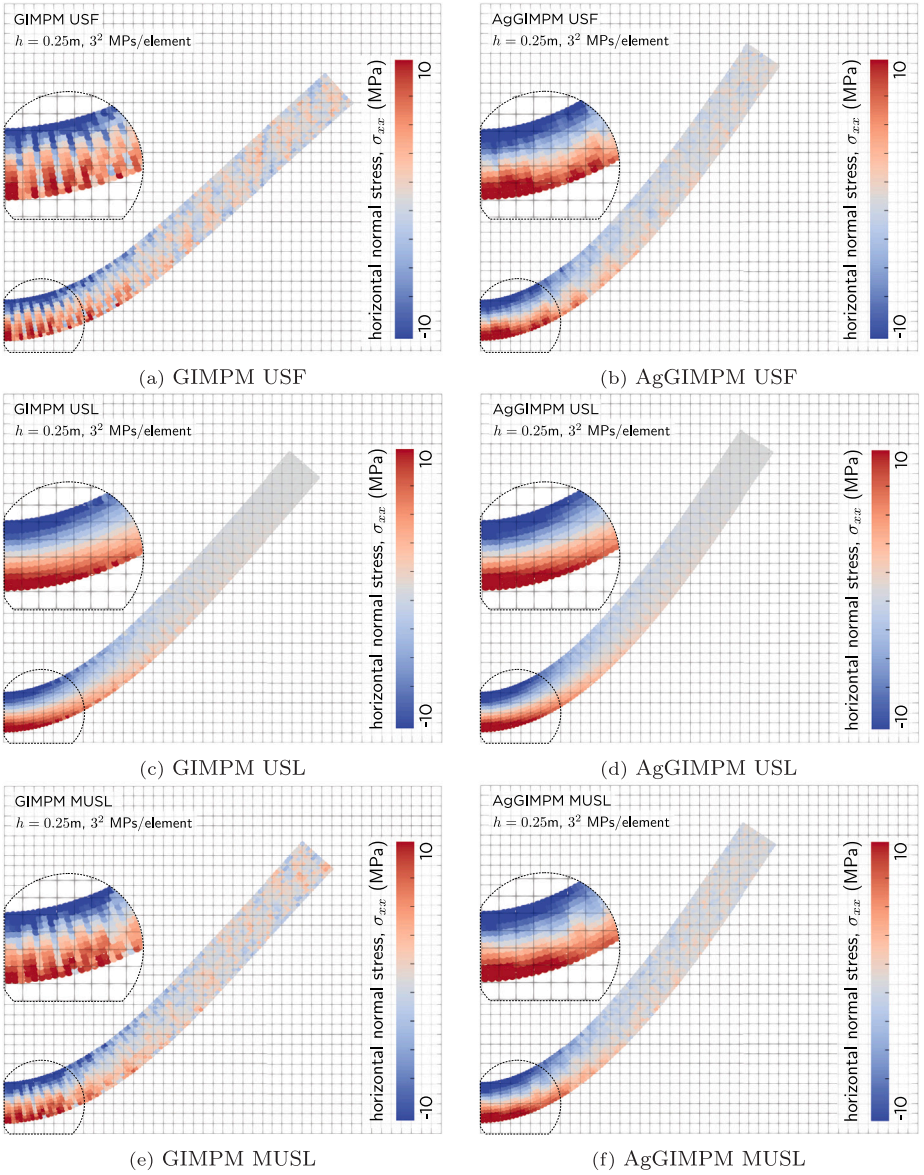
(a) total energy (b) AgGIMPM USF

Fig. 19. Dynamic elastic beam with $h = 0.25$ m, 3^2 MPs/element and 6000 time steps: (a) total energy USF, USL and MUSL for AgGIMPM and standard GIMPM, and (b) energy components for AgGIMPM USF.

The normalised displacement response of the tip of the beam and the total energy evolution over time is shown for six simulations in Fig. 18 and Fig. 19(a), respectively, representing the combinations of USF, USL and MUSL with standard and aggregated GIMPM, all using a lumped mass matrix. Note that the comparison does not include the ghost stabilisation technique as it has zero impact on a lumped mass matrix. One observation from Fig. 18 is that aggregation reduces the difference in the force–displacement response between the USF, USL and MUSL methods, with the AgGIMPM results almost indistinguishable for $t \in [0, 100]$ s. The grey-shaded region in Fig. 19 represents the period of time over which the external force is increased from zero to the maximum value; the force is removed at Point A. A key observation is that energy conservation is improved by mesh aggregation, with the dissipation/generation of energy reduced for each of the methods when comparing AgGIMPM with GIMPM.

Aggregation also improves the quality of the stress field in all cases, as shown in Fig. 20, where the deformed positions of the material points at the end of the analysis ($t = 150$ s) are shown shaded according to their horizontal normal stress value, σ_{xx}. The impact of aggregation on the stress distribution is particular evident for the USF and MUSL simulations. The dissipative nature of USL, and the associated damping of spurious accelerations, means that there is less impact of aggregation in this case. In the non-aggregated USF and MUSL simulations, artefacts in the acceleration field close to the boundary cause oscillations in the stress field that propagate through the beam.

The impact of the number of time steps on the stability and energy conservation of USF, USL and MUSL with standard and aggregated GIMPM with a lumped mass matrix is provided by Table 6. GsGIMPM with a stabilised consistent mass matrix is also included where, consistent with Coombs [19], the mass stabilisation parameter was set to $\gamma_M = \frac{1}{4}\rho$. The reported numbers are the relative change in the total energy from the point at which the external force is removed ($t = 25$ s, Point A in Fig. 19) and the end of the analysis ($t = 150$ s). The USL standard GIMPM analysis with 3000 time steps was unstable and failed at $t = 12.15$ s (step 243), as indicated by the ✗. Although they complete, the USF and MUSL analyses with 3000 time steps, irrespective of aggregation, are unstable and generate significant amounts of energy. For 6000 time steps and above, aggregation improves the energy conservation of USF, USL and MUSL, with the relative improvement reducing with increasing time steps. This is expected as lowering the time step will lessen the impact of spurious nodal accelerations at the boundary of the domain on the velocity and deformation field of the material points. Reducing the time step size will also help to mitigate the issue of the highly variable CFL number for the non-aggregated simulations. The GsGIMPM simulations with a consistent mass matrix are less stable than both the AgGIMPM and



(a) GIMPM USF (b) AgGIMPM USF

(c) GIMPM USL (d) AgGIMPM USL

(e) GIMPM MUSL (f) AgGIMPM MUSL

Fig. 20. Dynamic elastic beam: horizontal normal stress, σ_{xx}, distribution at the end of the analysis ($t = 150$ s) for $h = 0.25$ m, 3^2 MPs/element and 6000 time steps. The inset figures show detail around the cantilever root.

the GIMPM analyses, requiring a time step size of less than 0.0125 s for all of the USF, USL and MUSL approaches to be stable. GsGIMPM is also less energy conserving compared to AgGIMPM except for the $\Delta t = 0.010$ s USF analysis, where GsGIMPM marginally outperforms AgGIMPM. Combining GsGIMPM with MUSL leads to high levels of energy generation, even for the stable analyses.

Finally, mean run time (s) per time step for lumped GIMPM, lumped AgGIMPM and consistent mass GsGIMPM with $h = 0.25$ m, 3^2 MPs/element and $\Delta t = 0.01$ s are given in Table 7 for the USF, USL and MUSL approaches. The GIMPM has the lowest run time per load step, but as shown by the previous results, GIMPM without stabilisation or aggregation is unstable and in the case of USL can lead to excessive energy dissipation. The lumped AgGIMPM, as well as being more stable, is more efficient than GsGIMPM in all cases.

5. Conclusion

The small cut problem is a serious issue for the material point method, but there are relatively few general approaches that mitigate the issue, especially in the case of implicit implementations. This paper has provided the first Aggregated Material Point Method (AgMPM) and demonstrated the efficacy of the method versus standard MPM and Ghost-stabilised MPM (GsMPM)

Table 6
Dynamic elastic beam: energy loss/gain compared to the energy at $t = 25$ s (≈102 kJ) for AgGIMPM, GIMPM and GsGIMPM with $h = 0.25$ m and 3^2 MPs/element. The ✗ denotes a failed analysis, where the failed time step is reported.

no. time steps	3000	6000	9000	12,000	15,000
time step, Δt (s)	0.0500	0.0250	0.0167	0.0125	0.0100
GIMPM USF	+45.9%	+11.1%	+0.988%	+0.321%	+0.190%
GIMPM USL	243 ✗	−19.2%	−16.4%	−14.8%	−13.2%
GIMPM MUSL	+89.5%	+16.2%	+4.09%	+0.885%	+0.376%
AgGIMPM USF	+45.9%	+2.00%	+0.363%	+0.223%	+0.161%
AgGIMPM USL	−2.37%	−2.69%	−2.82%	−2.87%	−2.90%
AgGIMPM MUSL	+89.5%	+3.76%	+0.841%	+0.406%	+0.244%
GsGIMPM USF	10 ✗	234 ✗	+9.48%	+2.08%	−0.143%
GsGIMPM USL	9 ✗	171 ✗	296 ✗	−3.18%	−3.06%
GsGIMPM MUSL	9 ✗	235 ✗	+133%	+77.2%	+43.3%

Table 7
Dynamic elastic beam: mean run time (s) per time step for lumped GIMPM, lumped AgGIMPM and consistent GsGIMPM with $h = 0.25$ m, 3^2 MPs/element and $\Delta t = 0.01$ s.

	USF	USL	MUSL
GIMPM	0.0643	0.0645	0.0674
AgGIMPM	0.0686	0.0665	0.0711
GsGIMPM	0.0764	0.0769	0.0819

implementations. Similar to ghost stabilisation, mesh aggregation transforms the stability of the MPM and removes much of the uncertainty if an analysis will complete or not. Key advantages of the AgMPM versus GsMPM is that it can be applied to MPMs that use a lumped mass matrix and that no additional numerical parameters are required as it is simply based on the topology of the background mesh. It also does not require any face connectivity information to be stored within an implementation's data structure, which means it may be easier to implement within codes that do not routinely store this information. Although requiring additional operations, in the case of quasi-static implicit analysis, AgMPM is often more efficient than the standard MPM as the non-linear equations require fewer iterations to solve and are generally more stable. For explicit dynamic analysis, AgMPM improves the energy conservation properties and stress field representations for USF, USL and MUSL approaches as well as reducing the differences between the global structural response of the different methods. It has been shown that the AgGIMPM outperforms the GsGIMPM for explicit dynamic analyses in terms of stability, energy conservation and run time. Although this paper has focused on two-dimensional problems, it is straightforward to apply the technique to three-dimensional analysis, as it can be noted from the implementation details provided in Section 3.1.

CRediT authorship contribution statement

William M. Coombs: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization. **Robert E. Bird:** Writing – review & editing, Software, Methodology, Investigation, Conceptualization. **Giuliano Pretti:** Writing – review & editing, Methodology.

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.

Acknowledgements

The authors would like to acknowledge the contributions of the Computational Mechanics Research Node in the Department of Engineering of Durham University. The research presented in this article has benefited from discussions with and feedback from Professor Charles Augarde. The second author was supported by the Engineering and Physical Sciences Research Council, United Kingdom [grant number EP/W000970/1]. The third author was supported by the Faculty of Science at Durham University, United Kingdom. All data created during this research are openly available at doi.org/10.15128/r1gm80hv42k. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Appendix A. Aggregated weak form derivation

The starting point for deriving the discretised aggregated weak statement of equilibrium is the standard updated Lagrangian weak statement of equilibrium for dynamic stress analysis

$$\int_{\varphi_t(\Omega)} \Big(\sigma_{ij}(\nabla_x \eta)_{ij} - (b_i - \rho \dot{v}_i)\eta_i \Big) \mathrm{d}V - \int_{\varphi_t(\partial\Omega)} \big(t_i \eta_i \big) \mathrm{d}s = 0, \tag{A.1}$$

where $\dot{v}_i$ and φ_t are the acceleration and motion of the material body with current domain, Ω, which is subjected to tractions, t_i, on the boundary of the domain (with surface, s), $\partial\Omega$, and body forces, b_i, acting over the current volume, V of the domain, which has a current density, ρ, and leads to a Cauchy stress field, σ_{ij}, through the body. The weak form is derived in the current frame assuming a field of admissible virtual displacements, η_i. Introducing the standard finite element approximation, the displacements, velocities and accelerations can be expressed in terms of nodal values and basis functions, N_v, for example

$$\{\eta\} = \sum_{\forall v} N_v \{\eta_v\} = [N_v]\{d_\eta\} \qquad \text{and} \qquad \{u\} = [N_v]\{d\} \tag{A.2}$$

where $\{d_\eta\}$ and $\{d\}$ are global vectors of nodal virtual and physical displacements. For aggregated methods, the global vector of displacements is expressed in terms of the interior nodes as

$$\{d_\eta\} = [E]\{d_\eta^{\text{int}}\} \qquad \text{and} \qquad \{d\} = [E]\{d^{\text{int}}\}$$

where $[E]$ is the extension matrix. Neglecting the traction terms and introducing the aggregated finite element approximation yields

$$\int_{\varphi_t(\mathcal{K})} \Big(\Big[[\nabla_x N_v][E]\{d_\eta^{\text{int}}\} \Big]^T \{\sigma\} - \big\{ [N_v][E]\{d_\eta^{\text{int}}\} \big\}^T \big\{ \{b\} - \rho [N_v][E]\{\dot{v}^{\text{int}}\} \big\} \Big) \, \mathrm{d}V = 0. \tag{A.3}$$

Eliminating the nodal virtual displacements

$$\int_{\varphi_t(\mathcal{K})} \Big([E]^T [\nabla_x N_v]^T \{\sigma\} - [E]^T [N_v]^T \big\{ \{b\} - \rho [N_v][E]\{\dot{v}^{\text{int}}\} \big\} \Big) \mathrm{d}V = 0 \tag{A.4}$$

and expanding out the terms gives

$$\int_{\varphi_t(\mathcal{K})} \Big([E]^T [\nabla_x N_v]^T \{\sigma\} - [E]^T [N_v]^T \{b\} + \rho [E]^T [N_v]^T [N_v][E]\{\dot{v}^{\text{int}}\} \Big) \mathrm{d}V = 0. \tag{A.5}$$

Introducing the material point discretisation replaces the volume integral with an assembly operation over all of the material points in the analysis that represent the physical body results in

$$\underset{\forall p}{\mathbf{A}} \Big([E]^T [\nabla_x S_{vp}]^T \{\sigma_p\} V_p - [E]^T [S_{vp}]^T \{b\} V_p + \rho V_p [E]^T [S_{vp}]^T [S_{vp}][E]\{\dot{v}^{\text{int}}\} \Big) = 0, \tag{A.6}$$

which can be expressed as

$$[E]^T \left\{ \underset{\forall p}{\mathbf{A}} \Big([\nabla_x S_{vp}]^T \{\sigma_p\} V_p - [S_{vp}]^T \{b\} V_p \Big) \right\} + \Big[[E]^T [M][E] \Big] \{\dot{v}^{\text{int}}\} = 0. \tag{A.7}$$

The combined assembly term is the force residual (the difference between the internal and external forces). $[M^A] = [E]^T [M][E]$ is the aggregated mass matrix.

A.1. Explicit dynamic analysis

For explicit dynamic analysis, (A.7) allows the nodal accelerations to be forward predicted using

$$\{\dot{v}^{\text{int}}\} = [M^A]^{-1} \overbrace{[E]^T \underbrace{\left\{ \underset{\forall p}{\mathbf{A}} \Big([S_{vp}]^T \{b\} V_p - [\nabla_x S_{vp}]^T \{\sigma_p\} V_p \Big) \right\}}_{\text{force residual, } \{f_R\}}}^{\text{aggregated force residual, } \{f_R^A\}}. \tag{A.8}$$

$[M^A] = [E]^T [M][E]$ is the consistent aggregated mass matrix, however most MPMs adopt a lumped mass matrix. For AgMPM, this is achieved by replacing $[M]$ with a lumped mass matrix, as is done in the examples presented in this paper.

A.2. Implicit quasi-static analysis

Neglecting the inertia term reduces the aggregated equilibrium equation, (A.7), to a quasi-state expression with the form

$$[E]^T \left\{ \underset{\forall p}{\mathbf{A}} \Big([\nabla_x S_{vp}]^T \{\sigma_p\} V_p - [S_{vp}]^T \{b\} V_p \Big) \right\} = 0, \tag{A.9}$$

which in this paper is solved for the unknown aggregated nodal displacements using the conventional Newton–Raphson method. This requires linearisation of (A.9) with respect to the unknown aggregated nodal displacements, giving aggregated stiffness matrix as

$$[K^A] = [E]^T \left[\underset{\forall p}{\mathbf{A}} [\nabla_x S_{vp}]^T [a_p] [\nabla_x S_{vp}] V_p \right] [E] = [E]^T [K][E], \tag{A.10}$$

where $a_{ijkl} = (\partial \sigma_{ij} / \partial F_{km}) F_{lm}$ is the spatial consistent tangent modulus of the material in its current state of deformation, where $[F]$ is the deformation gradient (see Coombs and Augarde [34] for details), and $[K]$ is the stiffness matrix of the non-aggregated mesh.

Appendix B. Aggregation and lumping of the mass matrix

Explicit dynamic implementations of the MPM routinely adopt a lumped mass matrix, which makes inversion of the linear system of equations trivial. Lumping of the mass matrix is an algorithmic choice that alters the nature of the physical system of equations. Once this choice has been made the resulting system of equations needs to be solved, and this is when aggregation is applied to deal with the small cut issue. However, as discussed in Section 3.1, mesh aggregation reintroduces off diagonal terms into the lumped mass matrix, increasing the complexity and cost of solving the system of equations. It could be tempting to apply lumping to this aggregated mass matrix to recover a diagonal system. This appendix uses a simple problem to demonstrate that lumping after aggregation has the potential to lead to an unstable (and unphysical) system of equations.

Consider the problem shown in Fig. 5, and the associated extension matrix for Nodes 2, 3 and 4

$$[E_{([2,3,4],[2,3])}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 2 \end{bmatrix},$$

where the modified rows, those associated with Node 4 (the aggregated node), are highlighted. For illustrative purposes we can reduce this to an equivalent one-dimensional problem, where each node has a single degree of freedom. In this case the extension matrix becomes

$$[E_{([2,3,4],[2,3])}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 2 \end{bmatrix}.$$

Assuming that the lumped mass matrix for Nodes 2, 3 and 4 has the form

$$[M_{([2,3,4],[2,3,4])}] = \tilde{M} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tilde{m} \end{bmatrix},$$

where $\tilde{m}$ is the mass associated with Node 4, normalised by mass of Node 2 or 3, which are equal to $\tilde{M}$. Employing Sylvester's criterion to assess when a symmetric matrix is positive definite (see, for instance, [53]), the above matrix satisfies this feature as long as $\tilde{M} > 0$ and $\tilde{m} > 0$ are satisfied. This is expected in any mass matrix employing strictly non-negative basis functions, such as those used in the standard MPM and the GIMPM.

Applying mesh aggregation results in the following aggregated mass matrix

$$[M^A_{([2,3],[2,3])}] = [E]^T [M][E] = \tilde{M} \begin{bmatrix} 1+\tilde{m} & -2\tilde{m} \\ -2\tilde{m} & 1+4\tilde{m} \end{bmatrix},$$

where additional mass has been added to the leading diagonal and coupling introduced between the degrees of freedom supporting the aggregated Node 4. Employing again Sylvester's criterion, the above matrix is positive definite as long as $\tilde{M}(1+\tilde{m}) > 0$ and $5\tilde{m} > -\tilde{M}$. Both these criteria are satisfied for strictly positive values of $\tilde{m}$ and $\tilde{M}$. This suggests that lumping and successively aggregating does not result in negative semi-definite matrices as long as the initial lumped mass matrix is positive definite.

However, an interesting point to consider is what happens if lumping is applied to this aggregated mass matrix, which results in a diagonal matrix of the form

$$[\bar{M}^A_{([2,3],[2,3])}] = \tilde{M} \begin{bmatrix} 1-\tilde{m} & 0 \\ 0 & 1+2\tilde{m} \end{bmatrix}.$$

This matrix is positive definite if $\tilde{m} < 1$ and $1 + 2\tilde{m} > 0$. While the second requirement is not problematic, the first can lead to negative definite matrices. Subtracting the aggregated mass is fine when $\tilde{m} \ll 1$, but the requirement that $\tilde{m} < 1$ has important consequences if $\tilde{m}$ is not small, which can (and does) occur in mesh aggregation techniques. Additionally, it should be noted that an interior node may be supporting multiple aggregated nodes and therefore receive multiple negative aggregated mass contributions. This means that there is the potential for small, or even negative, masses to be generated on the leading diagonal of $[\bar{M}^A]$, making it unsuitable for general analysis.

It must also be noted that the same issue would manifest if the consistent mass matrix was aggregated and then lumped. This process could again lead to a resultant negative semi-definite matrix, and, as in the above-illustrated case, the root cause is the negative entries in the extension matrix, $[E]$.

Appendix C. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cma.2025.118012.

Data availability

All data created during this research are openly available at https://dx.doi.org/10.15128/r1gm80hv42k.

References

[1] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, Comput. Methods Appl. Mech. Engrg. 118 (1) (1994) 179–196.
[2] C.S. Peskin, Flow patterns around heart valves: A numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
[3] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, Internat. J. Numer. Methods Engrg. 104 (7) (2015) 472–501.
[4] Y. Bing, M. Cortis, T.J. Charlton, W.M. Coombs, C.E. Augarde, B-spline based boundary conditions in the material point method, Comput. Struct. 212 (2019) 257–274.
[5] D.A.B. Hyde, S.W. Gagniere, A. Marquez Razon, J. Teran, An implicit updated lagrangian formulation for liquids with large surface energy, ACM Trans. Graph. 39 (6) (2020).
[6] G. Remmerswaal, Development and Implementation of Moving Boundary Conditions in the Material Point Method, TU Delft, 2017.
[7] M. Cortis, W. Coombs, C. Augarde, M. Brown, A. Brennan, S. Robinson, Imposition of essential boundary conditions in the material point method, Internat. J. Numer. Methods Engrg. 113 (1) (2018) 130–152.
[8] B. Chandra, V. Singer, T. Teschemacher, R. Wüchner, A. Larese, Nonconforming Dirichlet boundary conditions in implicit material point method by means of penalty augmentation, Acta Geotech. 16 (2021) 2315–2335.
[9] V. Singer, T. Teschemacher, A. Larese, R. Wüchner, K.U. Bletzinger, Lagrange multiplier imposition of non-conforming essential boundary conditions in implicit material point method, Comput. Mech. 73 (2024) 1311–1333.
[10] Y. Yamaguchi, S. Moriguchi, K. Terada, Extended B-spline-based implicit material point method, Internat. J. Numer. Methods Engrg. 122 (7) (2021) 1746–1769.
[11] A. Chihadeh, W. Coombs, M. Kaliske, A coupled implicit MPM-FEM approach for brittle fracture and fragmentation, Comput. Struct. 288 (2023) 107143.
[12] Y. Liang, J. Given, K. Soga, The imposition of nonconforming Neumann boundary condition in the material point method without boundary representation, Comput. Methods Appl. Mech. Engrg. 404 (2023) 115785.
[13] G. Remmerswaal, P.J. Vardon, M.A. Hicks, Inhomogeneous Neumann boundary conditions for MPM and GIMP, Comput. Geotech. 173 (2024) 106494.
[14] D. Sulsky, M. Gong, Improving the material point method, in: Innovative Numerical Approaches for Multi-Field and Mulit-Scale Problems, in: Lecture Notes in Applied and Computational Mechanics, vol. 81, Springer, 2016, pp. 217–240.
[15] D. Sulsky, S.J. Zhou, H.L. Schreyer, Application of a particle-in-cell method to solid mechanics, Comput. Phys. Comm. 87 (1) (1995) 236–252.
[16] S.G. Bardenhagen, E.M. Kober, The generalized interpolation material point method, Comput. Model. Eng. Sci. 5 (6) (2004) 477–495.
[17] F. de Prenter, C.V. Verhoosel, G.J. van Zwieten, E.H. van Brummelen, Condition number analysis and preconditioning of the finite cell method, Comput. Methods Appl. Mech. Engrg. 316 (2017) 297–327.
[18] X. Ma, P.T. Giguere, B. Jayaraman, D.Z. Zhang, Distribution coefficient algorithm for small mass nodes in material point method, J. Comput. Phys. 229 (20) (2010) 7819–7833.
[19] W.M. Coombs, Ghost stabilisation of the material point method for stable quasi-static and dynamic analysis of large deformation problems, Internat. J. Numer. Methods Engrg. 124 (21) (2023) 4841–4875.
[20] B. Wang, P.J. Vardon, M.A. Hicks, Investigation of retrogressive and progressive slope failure mechanisms using the material point method, Comput. Geotech. 78 (2016) 88–98.
[21] D. Schillinger, M. Ruess, The finite cell method: A review in the context of higher-order structural analysis of CAD and image-based geometric models, Arch. Comput. Methods Eng. 22 (3) (2015) 391–455.
[22] M. Dauge, A. Düster, E. Rank, Theoretical and numerical investigation of the finite cell method, J. Sci. Comput. 65 (2015) 1039–1064.
[23] K. Höllig, C. Apprich, A. Streit, Introduction to the Web-method and its applications, Adv. Comput. Math. 23 (2005) 215–237.
[24] K. Höllig, U. Reif, J. Wipper, Weighted extended B-spline approximation of Dirichlet problems, SIAM J. Numer. Anal. 39 (2) (2001) 442–462.
[25] R.A.K. Sanches, P.B. Bornemann, F. Cirak, Immersed b-spline (i-spline) finite element method for geometrically complex domains, Comput. Methods Appl. Mech. Engrg. 200 (13) (2011) 1432–1445.
[26] E. Burman, Ghost penalty, C.R. Math. 348 (21) (2010) 1217–1220.
[27] G. Pretti, R. Bird, N. Gavin, W. Coombs, C. Augarde, A stable poro-mechanical formulation for Material Point Methods leveraging overlapping meshes and multi-field ghost penalisation, Internat. J. Numer. Methods Engrg. 126 (2025) e7630.
[28] S. Badia, F. Verdugo, Robust and scalable domain decomposition solvers for unfitted finite element methods, J. Comput. Appl. Math. 344 (2018) 740–759.
[29] S. Gross, A. Reusken, Analysis of optimal preconditioners for CutFEM, Numer. Linear Algebra Appl. 30 (5) (2023) e2486.
[30] F. de Prenter, C.V. Verhoosel, E.H. van Brummelen, Preconditioning immersed isogeometric finite element methods with application to flow problems, Comput. Methods Appl. Mech. Engrg. 348 (2019) 604–631.
[31] F. de Prenter, C.V. Verhoosel, E.H. van Brummelen, M.G. Larson, S. Badia, Stability and conditioning of immersed finite element methods: Analysis and remedies, Arch. Comput. Methods Eng. 30 (2023) 3617–3656.
[32] S. Badia, F. Verdugo, A.F. Martín, The aggregated unfitted finite element method for elliptic problems, Comput. Methods Appl. Mech. Engrg. 336 (2018) 533–553.
[33] Z. Zhu, T. Bao, X. Zhu, J. Gong, Y. Hu, J. Zhang, An improved material point method with aggregated and smoothed Bernstein functions, Mathematics 11 (4) (2023).
[34] W.M. Coombs, C.E. Augarde, AMPLE: A material point learning environment, Adv. Eng. Softw. 139 (2020) 102748.
[35] T.J. Charlton, W.M. Coombs, C.E. Augarde, iGIMP: An implicit generalised interpolation material point method for large deformations, Comput. Struct. 190 (2017) 108–125.
[36] J.C. Simo, Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory, Comput. Methods Appl. Mech. Engrg. 99 (1992) 61–112.
[37] E.A. de Souza Neto, D. Peric, D.R.J. Owen, Computational methods for plasticity: Theory and applications, in: Computational Methods for Plasticity, John Wiley & Sons, Ltd, 2008.
[38] W.M. Coombs, C.E. Augarde, A.J. Brennan, M.J. Brown, T.J. Charlton, J.A. Knappett, Y. Ghaffari Motlagh, L. Wang, On Lagrangian mechanics and the implicit material point method for large deformation elasto-plasticity, Comput. Methods Appl. Mech. Engrg. 358 (2020) 112622.

[39] W.M. Coombs, T.J. Charlton, M. Cortis, C.E. Augarde, Overcoming volumetric locking in material point methods, Compt. Methods Appl. Mech. Eng. 333 (2018) 1–21.
[40] E. Burman, P. Hansbo, M.G. Larson, CutFEM based on extended finite element spaces, Numer. Math. 152 (2022) 331–369.
[41] T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows, Comput. Methods Appl. Mech. Engrg. 209 (2012) 266–283.
[42] T. Rüberg, F. Cirak, A fixed-grid b-spline finite element technique for fluid–structure interaction, Internat. J. Numer. Methods Fluids 74 (9) (2014) 623–660.
[43] F. Verdugo, A.F. Martín, S. Badia, Distributed-memory parallelization of the aggregated unfitted finite element method, Comput. Methods Appl. Mech. Engrg. 357 (2019) 112583.
[44] S. Sticko, G. Ludvigsson, G. Kreiss, High-order cut finite elements for the elastic wave equation, Adv. Comput. Math. 46:45 (2020) 1–28.
[45] S.K.F. Stoter, S.C. Divi, E.H. van Brummelen, M.G. Larson, F. de Prenter, C.V. Verhoosel, Critical time-step size analysis and mass scaling by ghost-penalty for immersogeometric explicit dynamics, Comput. Methods Appl. Mech. Engrg. 412 (2023) 116074.
[46] T.K. Molstad, Finite Deformation Analysis Using the Finite Element Method (Ph.D. thesis), University of British Columbia, 1977.
[47] R.E. Bird, G. Pretti, W.M. Coombs, C.E. Augarde, Y.U. Sharif, M.J. Brown, G. Carter, C. Macdonald, K. Johnson, An implicit material point-to-rigid body contact approach for large deformation soil–structure interaction, Comput. Geotech. 174 (2024) 106646.
[48] C. Davidson, M.J. Brown, A.J. Brennan, J.A. Knappett, B. Cerfontaine, Y.U. Sharif, Physical modelling of screw piles for offshore wind energy, in: First International Symposium on Screw Piles for Energy Applications, University of Dundee, Dundee, UK, 2019.
[49] B. Cerfontaine, J. Knappett, M.J. Brown, C. Davidson, Y. Sharif, Optimised design of screw anchors in tension in sand for renewable energy applications, Ocean Eng. 217 (2020) 108010.
[50] T. Schanz, P.A. Vermeer, P.G. Bonnier, The hardening soil model: Formulation and verification, in: Beyond 2000 in Computational Geotechnics, Routledge, 2019, pp. 281–296.
[51] R.B.J. Brinkgreve, E. Engin, H.K. Engin, Validation of empirical formulas to derive model parameters for sands, Numer. Methods Geotech. Eng. 137 (2010) 142.
[52] R.E. Bird, G. Pretti, W.M. Coombs, C.E. Augarde, Y.U. Sharif, M.J. Brown, G. Carter, C. Macdonald, K. Johnson, A dynamic implicit 3D material point-to-rigid body contact approach for large deformation analysis, 2024, URL https://arxiv.org/abs/2412.01565.
[53] Alfio Quarteroni, Riccardo Sacco, Fausto Saleri, Numerical Mathematics, 37, Springer Science & Business Media, 2010.