Received: 8 February 2023 | Revised: 7 June 2023 | Accepted: 4 July 2023

DOI: 10.1002/nme.7332

RESEARCH ARTICLE

WILEY

Ghost stabilisation of the material point method for stable quasi-static and dynamic analysis of large deformation problems

William M. Coombs

Department of Engineering, Durham University, Durham, UK

Correspondence
William M. Coombs, Department of Engineering, Durham University, Durham, UK.
Email: w.m.coombs@durham.ac.uk

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Numbers: EP/W000970/1, EP/R004900/1, EP/N006054/1

Abstract

The unstable nature of the material point method (MPM) is widely documented and is a barrier to the method being used for routine engineering analyses of large deformation problems. The vast majority of articles concerning this issue are focused on the instabilities that manifest when a material point crosses between background grid cells. However, there are other issues related to the stability of MPMs. This article focuses on the issue of the conditioning of the global system of equations caused by the arbitrary nature of the position of the physical domain relative to the background computational grid. The issue is remedied here via the use of a ghost stabilisation technique that penalises variations in the gradient of the solution field near the boundaries of the physical domain. This technique transforms the stability of the MPM, providing a robust computational framework for large deformation explicit dynamic and implicit *quasi*-static analysis.

KEYWORDS

explicit dynamics, finite deformation mechanics, implicit analysis, material point method, stabilisation

1 | INTRODUCTION

The material point method's (MPMs)[1] greatest advantage, the decoupling of the deformation of the physical material from the computational grid, is also its greatest challenge in terms of robust and efficient engineering computations. By allowing the physical bodies to deform through a background grid of finite elements, the MPM can be used to model large deformation solid mechanics problems involving history dependent materials without re-meshing or re-mapping of material parameters (see de Vaucorbeil et al.[2] and Solowski et al.[3] for recent reviews articles), especially when post failure behaviour is of interest.[4] However, this means the overlap between the physical body, represented by material points, and the background grid, which is used to solve the governing equations, varies over the analysis. This feature is not unique to the MPM and allows the method to be grouped within a larger class of non-matching mesh-based methods, such as the cut finite element method,[5-7] the finite cover method,[8] and the finite cell method,[9,10] amongst others, it also shares some parallels with enriched methods[11,12] in terms of the decoupling of physical features from the computational mesh. However, the variable interaction between the physical body and the material points in the MPM can cause numerical stability problems.[13] Specifically, background elements that are partially filled with material points cause problems in terms of the conditioning of the global system of equations. Similar problems have been observed in the cut finite element method,[5-7] where arbitrarily small intersections can occur between an element, K, and the physical problem domain, Ω. These intersections can be much smaller than the size of the elements, h. This impacts on the conditioning of the linear system of equations being solved as the smallest eigenvalue of the linear system is related to the smallest intersection between the elements and the background grid, which can be arbitrarily small. This means that the condition number (the ratio of the largest to smallest eigenvalue) of the linear system of equations is not bounded. These issues have a detrimental impact on both dynamic and *quasi*-static analysis via the conditioning of the mass and/or stiffness matrix, with the issues often being more severe for the stiffness matrix.[5] In terms of MPMs, the issue is compounded when using domain-based MPMs, such as the generalised interpolation material point method (GIMPM),[14] where arbitrarily small overlaps can occur between the material point domains, Ω_p, and the elements in the background mesh. The issue can also occur in B-spline-based material point approaches[15,16] due to the extended range of influence of the basis functions and the potential for material points to have a small contribution to a background grid degree of freedom. This was highlighted by Yamaguchi et al.[17] who reinforced that the issue can be more severe for MPMs than for cut finite element methods (and other embedded-domain/unfitted methods). A crude way avoid this issue is to exclude small overlaps between the material point domains and the background mesh (or rather small shape function values), or through excluding those nodes with small masses from the equilibrium solution or even by adding additional mass to these degrees of freedom. However, these *ad hoc* methods rely on the definition of *small*, which is arbitrary[18] and problem dependent and does not enforce a bound on the conditioning of the linear system of equations being solved. An alternative approach is to modify the MPM algorithm and weight the internal forces by the basis functions to reduce the impact of small material point-background grid interactions.[18] This *modified update stress last* (MUSL) approach changes the calculation of the updated nodal velocities such that when a small mass is calculated at a nodal location it is weighted by a correspondingly small basis function.[19] However, this approach does not directly resolve the potential for the consistent mass (or stiffness) matrix to be ill-conditioned. One approach that avoids this issue is the total Lagrangian MPM of de Vaucorbeil et al.[20] but the approach loses many of the benefits of the MPM, such as the ability to handle large distortional deformations and automatic contact detection.

The conditioning issues described above are largely avoid by the vast majority of explicit dynamic MPMs via the adoption of a *lumped* mass matrix, making the inversion of the mass matrix trivial[*]. However, adopting a lumped mass matrix has consequences for energy conservation,[18,21-25] resulting in artificial numerical dissipation. This issue was recognised by Nairn and Hammerquist[24] who proposed an approximation to the consistent mass matrix inverse based on an eXtended Particle-In-Cell (XPIC) concept[26] which allowed the mass matrix inverse to be written as an infinite series expansion. Small overlaps between material points and the background grid causes other issues. For example, Ma et al.[13] highlighted impact of small background grid nodal masses on the stability of MPMs when determining the grid node accelerations. They introduced an approach to redistribute the nodal forces associated with the *small* nodal masses to eliminate the spurious accelerations[†]. However, the method does not deal with the small values in the mass matrix, choosing instead to deal with the manifestation of these small values in the acceleration calculation. Very few solutions to this issue have been proposed for implicit *quasi*-static analyses, where the conditioning of the stiffness matrix is critical for stable convergence. Wang et al.[27] are proponents of a *soft stiffness* to stabilisation, where it is assumed that the background finite element mesh has a *small* elastic modulus, which is integrated and added to the global stiffness matrix (see Section 3.1.2 of Wang et al.[27] for details). However, the approach is not targeted at the regions that need stabilisation and degrades the equilibrium convergence rate of implicit solution methods due to the lack of a corresponding term in the weak form. An alternative technique was proposed by Yamaguchi et al.[17] who introduced extended B-spline basis functions in background grid cells interacting with the boundary of the physical domain in an attempt to reduce the stress oscillations and conditioning issues associated with poor integration. However, this approach requires the introduction of boundary points to track the deformation of the body, which limits the applicability of the methods to problems involving significant distortion, self-contact and so forth.

This article resolves the issue of ill-conditioning of the global consistent mass and stiffness matrices for dynamic and *quasi*-static MPMs using, and adapting, the *ghost stabilisation* technique of Burman[28]. The method is general in that it can be applied to any material point variant and does not require the physical boundary of the domain to be tracked explicitly. The ghost stabilisation opens the door for the consistent mass matrix to be used for routine dynamic material point analyses, eliminating the additional energy dissipation caused by a lumped mass matrix and provides flexibility in the stress and velocity updating procedures. It offers the first credible stabilisation technique for implicit *quasi*-static MPMs, answering a key question posed by Yamaguchi et al.,[17] and removing much of the uncertainty regarding if a given analysis will converge or not. However, the technique is equally practical for explicit and implicit MPMs, as demonstrated by the numerical simulations in Section 5. Overall, this article provides a key ingredient for MPMs to become a practical choice for large deformation engineering analyses.

2 | MATERIAL POINT METHOD

The MPM uses material points to represent the physical body/bodies being analysed but the governing equations are solved on a background mesh/grid of finite elements. This means that quantities must be mapped from points-to-grid and from grid-to-points during different stages of an analysis, and the key steps in a MPM algorithm are shown in Figure 1. A given problem is discretised into a number of time (or load) steps and for each of the steps the following procedure is applied:

(a) the physical body is discretised by a number of material points that carry information on the volume, mass, deformation, stress and so forth of the material that they represent and the position of these points on the background grid determined (i.e., identifying the element(s)/nodes(s) the points interact with);
(b) material point information on velocity/momentum, stress, mass, and stiffness is mapped to the background grid using the basis functions (or spatial derivatives) of the MPM variant under consideration to form nodal velocity, force, mass, and stiffness quantities, as required by the adopted time stepping algorithm;
(c) governing equations are assembled on the background grid nodes;

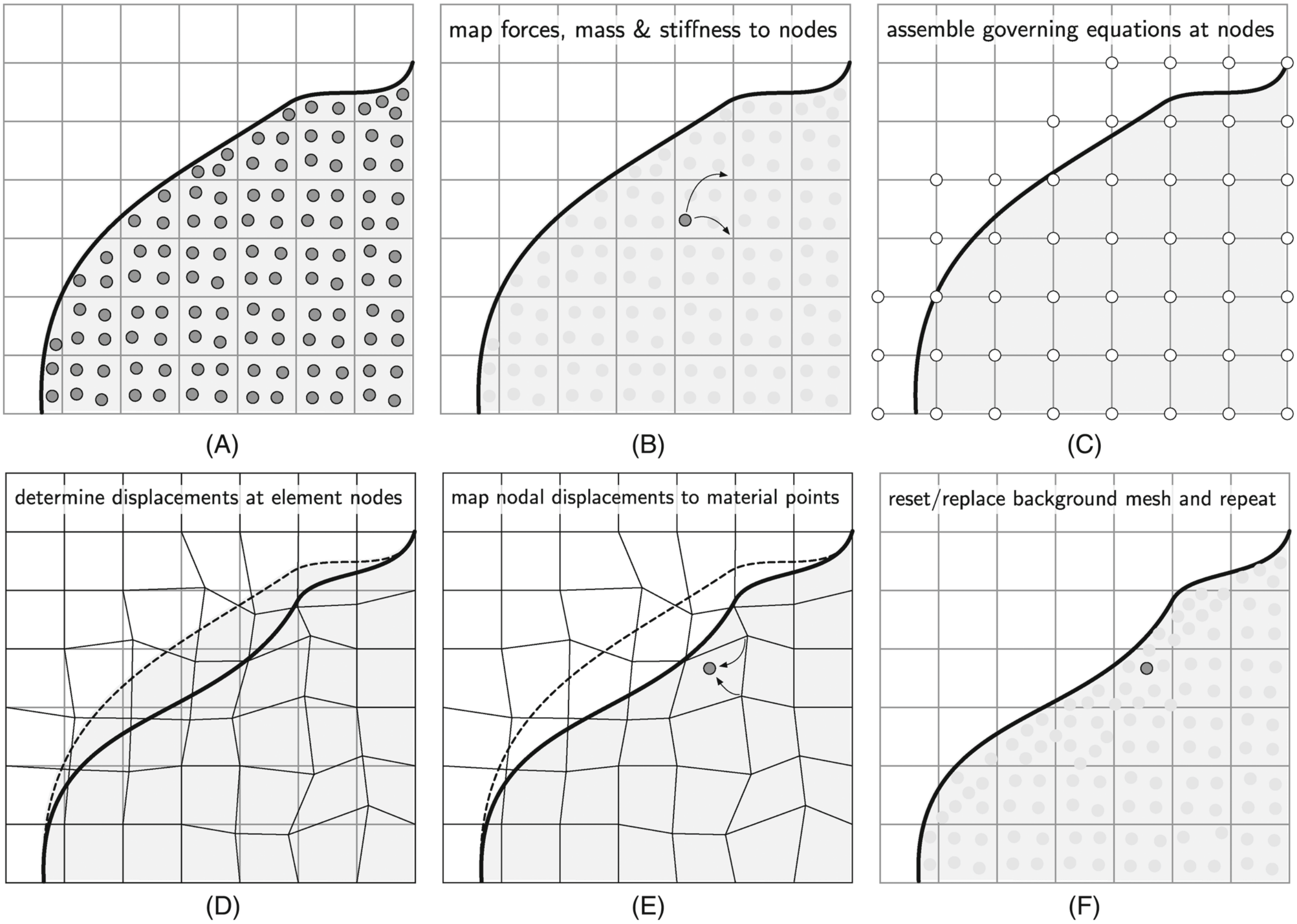
map forces, mass & stiffness to nodes
assemble governing equations at nodes
(A)
(B)
(C)
determine displacements at element nodes
map nodal displacements to material points
reset/replace background mesh and repeat
(D)
(E)
(F)

FIGURE 1 Material point method steps (adapted from Coombs and Augarde[29]). (A) Initial position. (B) Point-to-grid map. (C) Assembly. (D) Grid solution. (E) Grid-to-point map. (F) Update points and reset grid.

(d) governing equations are solved for the unknown nodal displacements/velocities (depending on the nature of the governing equations and the adopted time stepping algorithm);
(e) grid information is mapped to the material points to update deformation, velocity, stress, volume and so forth and to determine the material point displacement over the time step;
(f) material points are updated[‡] to their new positions and the background grid reset/replaced.

The above general steps are applicable to dynamic and *quasi*-static MPMs discretised in time via implicit and explicit approaches. However, specific details within each of the steps will change depending on algorithm/implementation choices, for example the material point stress updating algorithmic location will change if an update stress first (USF) explicit approach is adopted, as the above position corresponds to an update stress last (USL) method. Key to this procedure is the definition of the governing equations that will be solved during the analysis. This article is focused on dynamic and *quasi*-static analysis of elastic and elasto-plastic solids undergoing large deformations and the governing equations for this continuum behaviour will be covered in the next section.

2.1 | Continuum formulation

The material point approach adopted in this article is based on the open-source AMPLE (A Material Point Learning Environment) code,[29] which is a *quasi*-static implicit finite deformation elasto-plastic MPM based on an updated Lagrangian formulation[§], but extended to include inertia effects for the purpose of this article. The method is defined by the following weak statement of equilibrium

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

where $\dot{v}_i$ and φ_t are the acceleration and motion of the material body with 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 volume, V of the domain, which has a 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. In the case of *quasi*-static analysis the acceleration of the body is assumed to be zero, which reduces (1) to that of the AMPLE framework.[29] The surface traction term in (1) is neglected as the focus of this article is on the stability of the MPM rather than the imposition of surface tractions within the method, which is a research question in its own right (for example, see Bing et al.[31] or Remmerswaal[32]). We also restrict the Dirichlet boundary conditions to coincide with the background grid as the imposition of general Dirichlet constraints is another active area of research.[31,33,34]

The large deformation elasto-plastic continuum mechanics formulation used in this article is identical to that of Charlton et al.[30] and Coombs et al.[29] and readers are referred to those papers for details. In brief, the formulation adopts a multiplicative decomposition of the deformation gradient into elastic and plastic components, with a linear relationship between logarithmic elastic strain and Kirchhoff stress. This is combined with an exponential map of the plastic flow rule which allows conventional small strain plasticity algorithms to be used without modification.[35] This powerful combination is widely used in large deformation finite element methods[36] and several MPM implementations.[29,30,34,37-40] These numerical implementations require the weak statement of equilibrium (1) to be discretised in space and time.

2.2 | Discrete formulation

Neglecting the traction term, the Galerkin form of the weak statement of equilibrium over each background grid element, K, can be obtained from (1) as

$$\int_{\varphi_t(K)} [\nabla_x S_{vp}]^T\{\sigma_p\}\mathrm{d}V - \int_{\varphi_t(K)} [S_{vp}]^T\{b\}\mathrm{d}V + \int_{\varphi_t(K)} \rho[S_{vp}]^T\{\dot{v}\}\mathrm{d}V = \{0\}, \tag{2}$$

where $[S_{vp}]$ contains the basis/shape functions that are used to transfer information between the material points and the background grid, $[\nabla_x S_{vp}]$ is he strain-displacement matrix containing derivatives of the basis functions with respect to the updated coordinates[¶]. Consistent with the majority of the published MPM literature, the subscripts $(\cdot)_v$ and $(\cdot)_p$ denote quantities associated with *vertices* (nodes) of the background grid and material *points*, respectively.

In the MPM, the physical domain is discretised by a material points, each representing a finite volume of material, V_p. This allows the Galerkin form of the weak statement of equilibrium (2) to be approximated as

$$\mathop{\mathcal{A}}_{\forall p}\left([\nabla_x S_{vp}]^T\{\sigma_p\}V_p - [S_{vp}]^T\{b\}V_p + [S_{vp}]^T\{\dot{v}\}m_p\right) = \{0\}, \tag{3}$$

where $\mathcal{A}$ is the standard assembly operator acting over all of the material points in the problem[#] and $m_p = \rho V_p$ is the mass associated with a material point. The first term in (3) represents the internal nodal forces generated by the stress within the material, the second term is the external actions caused by the body forces and the third term accounts for the inertia of the physical body.

2.3 | Explicit dynamics

Explicit time stepping approaches assumed that the next state can be determined from information known at the current state, essentially forward predicting the new accelerations, $\{\dot{v}_v\}$, via the difference between the internal and external forces. Assuming that the material point acceleration can be approximated as

$$\{\dot{v}_p\} = [S_{vp}]\{\dot{v}_v\}, \tag{4}$$

where $\{\dot{v}_v\}$ are the accelerations of the background grid nodes, (3) becomes

$$\mathop{\mathcal{A}}_{\forall p}\left([\nabla_x S_{vp}]^T\{\sigma_p\}V_p - [S_{vp}]^T\{b\}V_p + [S_{vp}]^T[S_{vp}]\{\dot{v}_v\}m_p\right) = \{0\}, \tag{5}$$

which can be rearranged to obtain the grid node accelerations

$$\{\dot{v}_v\} = [M_v]^{-1}\left\{\mathop{\mathcal{A}}_{\forall p}\left([S_{vp}]^T\{b\}V_p - [\nabla_x S_{vp}]^T\{\sigma_p\}V_p\right)\right\}. \tag{6}$$

where

$$[M_v] = \mathop{\mathcal{A}}_{\forall p}\left([S_{vp}]^T[S_{vp}]m_p\right), \tag{7}$$

is the consistent mass matrix assembled on the background grid nodes. The nodal accelerations can be used to increment the material point velocities via a FLuid Implicit Particle (FLIP) update[‖]

$$\{v_p\}_{n+1} = \{v_p\}_n + \Delta t \sum_{\forall v} S_{vp}\{\dot{v}_v\}, \tag{8}$$

where Δt is the time step size and the subscripts $(\cdot)_n$ and $(\cdot)_{n+1}$ denote the previous and updated states, respectively. The material point positions are updated using

$$\{x_p\}_{n+1} = \{x_p\}_n + \Delta t \sum_{\forall v} S_{vp}\{v_v\}_{n+1}. \tag{9}$$

The nodal velocities at the start of the step are obtained by mapping the momentum of the material points to the grid nodes and multiplying this momentum by the inverse of the mass matrix

$$\{v_v\}_n = [M_v]^{-1}\left\{\mathop{\mathcal{A}}_{\forall p}\left(m_p[S_{vp}]^T\{v_p\}_n\right)\right\}, \tag{10}$$

and the nodal velocities at the end of the step are obtained by adding nodal accelerations multiplied by the time step the previous velocity

$$\{v_v\}_{n+1} = \{v_v\}_n + \Delta t\{\dot{v}_v\}. \tag{11}$$

The stress updating algorithmic position is an important consideration for explicit MPMs. There are two basic options[22]:

1. USF: the stress is updated before the determination of the nodal velocities, (6), using the increment in the deformation field from the previous step; or
2. USL: the stress is updated after the determination of the nodal velocities using the increment in the deformation field form the current step.

The stress updating point has implications on the energy conservation of MPMs. For example, when combined with a lumped mass matrix, the USL approach leads to excessive energy dissipation.[22] However, the USL approach is often preferred as the dissipation is consistent with the accuracy of the solution and it tends to damped unresolved modes.[22,43] Differences between USF and USL, alongside the MUSL approach,[18] will be explored in more detail as part of the numerical analyses presented in Section 5, including a study on the energy conserving nature of the approaches with both lumped and consistent mass matrices.

2.4 | Implicit *quasi*-static analysis

For *quasi*-static analysis the inertia term in (1) is assumed to be negligible, reducing the discrete statement of equilibrium to

$$\underset{\forall p}{\mathcal{A}} \left([\nabla_x S_{vp}]^T \{\sigma_p\} V_p - [S_{vp}]^T \{b\} V_p \right) = \{0\}, \tag{12}$$

which is a non-linear statement of equilibrium in terms of the nodal displacements in the deformed configuration. This non-linear equation can be efficiently solved using an implicit Newton–Raphson approach, which requires the equilibrium statement to be linearised with respect to the nodal displacements to form the global stiffness matrix

$$[K] = \underset{\forall p}{\mathcal{A}} \left([\nabla_x S_{vp}]^T [a_p] [\nabla_x S_{vp}] V_p \right), \tag{13}$$

where $[a_p]$ is the spatial consistent tangent modulus of the material point under its current state of deformation.[30] This global stiffness matrix is used to iteratively update the nodal displacements until the equilibrium statement (12) converges within a given tolerance and at each iteration the material point stress and tangent modulus are updated. Once convergence has been achieved the material point positions, $\{x_p\}$, are updated using

$$\{x_p\}_{n+1} = \{x_p\}_n + \sum_{\forall v} S_{vp} \{u_v\}, \tag{14}$$

where $\{u_v\}$ is the displacement of a node over the current load step. This approach is adopted by the open source AMPLE code and the reader is referred to Coombs and Augarde[29] for details.

2.5 | Basis functions

Different MPMs are characterised by the choice of basis functions, $[S_{vp}]$, that map information between the material points and background grid nodes. In this article two options are considered, the:

1. standard MPM (sMPM),[1] where the shape functions of the underlying finite element grid are adopted and the material points are assumed to be concentrated point masses/volumes; and
2. GIMPM,[14] where the basis functions are formed via the convolution of the shape functions of the finite element grid with a particle characteristic function (usually taken to be a unity function), which generates C^1 continuous basis functions via the uniform distribution of the mass/volume associated with the material point over a domain centred on the material point. In this article, the domains are updated using the symmetric material stretch tensor, as detailed by Charlton et al.[30]

The GIMPM was developed to reduce the well documented cell crossing instability of MPMs, caused by the sudden transfer of internal force as material points cross between elements of the background grid. Other options include convected particle domain interpolation methods[44,45] and adopting spline-based basis functions.[46,47]

2.6 | Conditioning issues

Explicit and implicit MPMs suffer from stability issues associated with the conditioning of the consistent mass and stiffness matrices. This trivial test problem is designed to highlight these issues. A 2 × 1 m plane strain physical domain was discretised by eight material points, each representing 0.25 m^3 of material with a Young's modulus of $E = 1$ Pa, Poisson's ratio of $\nu = 0$ and a density of $\rho = 1\ \text{kg/m}^3$. A background grid comprised of $h = 1$ m square bi-linear quadrilateral elements was used to construct the global consistent mass and small-strain stiffness matrices. The background grid nodes were constrained as shown in Figure 2, where the grid nodes at $y = 0$ m are constrained vertically and the node at $x = 2$ m, $y = 0$ m is also constrained horizontally.

Figure 3 shows the evolution of the condition number, $\kappa([\cdot])$, the ratio of the largest to smallest eigenvalue[**], of the reduced[††] consistent mass and stiffness matrices as the physical domain is translated by $a/h = 2$ over 5000 steps for both the sMPM and GIMPM. Note that as the focus of this example is on the impact of the material point positions on the conditioning of the mass and stiffness matrices, the translation of the physical domain is imposed directly at the material points rather than being driven by any grid-based boundary conditions. Therefore, no governing equations are solved, or variables updated, as part of this example beyond updating the positions of the material points. Figure 3 also reports the condition number of the lumped mass matrix

$$[\overline{M}_v] = \text{diag}\left(\mathop{\mathcal{A}}_{\forall p} \left([S_{vp}]^T \{1\} m_p \right) \right), \tag{15}$$

where $\{1\}$ is a vector of ones of length equal to the number of physical dimensions. The lumped mass matrix is often used in MPMs instead of the consistent mass matrix[‡‡].

The condition number of all of the matrices varies as the physical domain translates through the background grid. For the sMPM (Figure 3A), sudden increases in the condition number of the mass matrices are observed when material points transition between background grid cells (for example, at $a/h = 0.25, 0.75, 1.25$, and 1.75). This is due to the transitioning material points having very small contributions to certain nodes in terms of the basis functions values. The stiffness matrix does not show the same spikes as it is formed using the spatial derivatives of the basis functions, which, as we are only concerned with the behaviour in the horizontal direction, are piecewise constant within each element. Therefore, the position of a material point within the element has less impact on the conditioning of the stiffness matrix. Beyond $a/h = 1.75$ the condition of the stiffness matrix $\kappa([K]) = \infty$ as there is no constraint in the horizontal direction as all of the material points have moved beyond the influence of the horizontally constrained node.

The response of the GIMPM is smoother due to the C^1 continuity of the basis functions. However, both the mass and stiffness matrices suffer from conditioning issues as the spatial derivatives of the basis functions are dependent on the overlap between the domains associated with the material points and the background grid cells. This causes spikes in the condition number when $a/h = 0, 1$, and 2. In all cases, the condition number increase is more pronounced for the consistent mass matrix than the lumped mass matrix as it contains the product of the basis function with themselves.

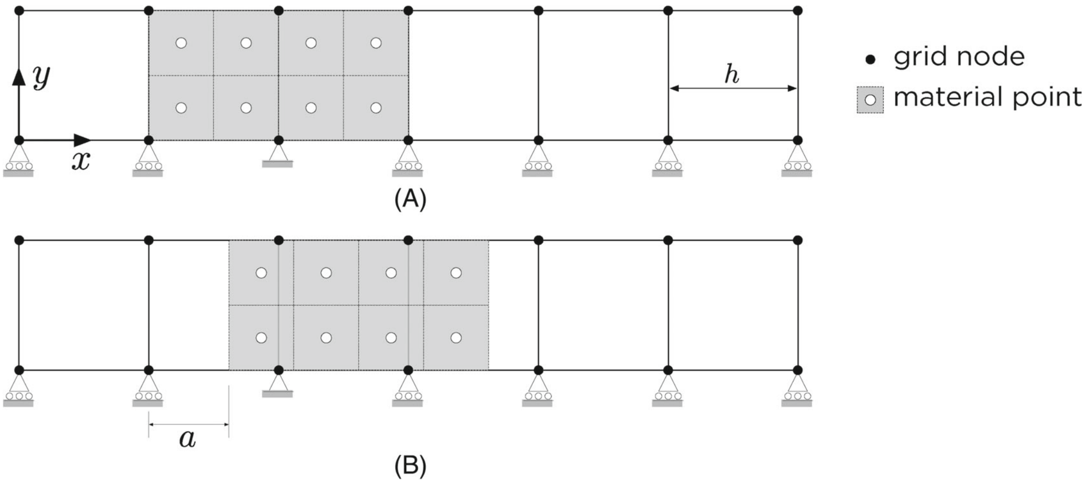

FIGURE 2 Translating material point domain problem setup. (A) Initial position. (B) Domain translation.

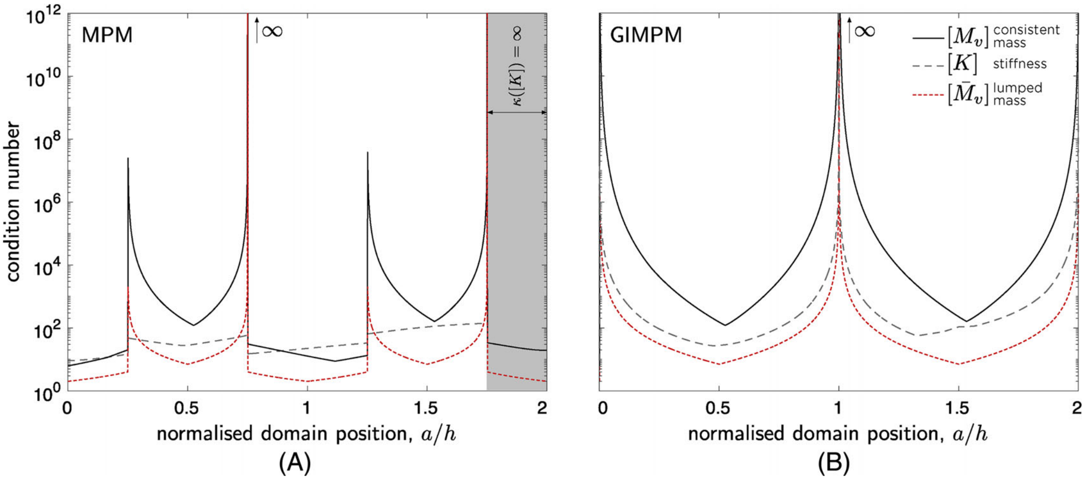

FIGURE 3 Translating material point domain: Condition number variation. (A) sMPM. (B) GIMPM.

This is a trivial problem but it demonstrates the conditioning issues faced by MPMs. The key issue is that we do not know how the material points, and their associated domains for the GIMPM, will coincide with the background grid through the analysis. These issues have resulted in most explicit MPMs adopting a lumped mass matrix and/or including thresholds on the minimum considered basis function values that contributes to the analysis. These numerical *fixes* are not universal and have implications on the accuracy, stability and general use of the methods. Although there have been some papers investigating the issues associated with poor conditioning of the mass[13] and stiffness[48] matrices, a general solution has yet to be provided.

3 | GHOST STABILISATION

A typical material point situation is shown in Figure 4, where a physical body is discretised by a number of material points on a regular background mesh. Although a boundary to the physical domain has been shown, most MPMs do not include a representation of the physical boundary and instead they rely on the location of the material points to represent the extent of the physical material. Figure 4B identifies the unpopulated elements (the elements that do not contain material points) in the background mesh, which have been shaded grey. Figure 4C identifies the boundary elements (the elements that are intersected by the physical boundary), which are shaded dark grey. It is these boundary elements that potentially contain small overlaps between material points and the background mesh, depending on how the material points are used to represent the physical material and their associated characteristic function.

One approach to overcome the issues associated with small overlaps between the physical domain and the computational grid is to add additional stabilising terms of the mass/stiffness matrix that provide a bound on the conditioning of the system. The *ghost* penalty stabilisation approach was first proposed by Burman,[28] and aims to bound the conditioning of a matrix by penalising jumps in the gradient of the solution field across elements cut by the physical boundary. The name *ghost* can be interpreted as projection of the physical solution field into the empty part of the computational mesh. This is achieved by adding an extra term to the linear system being solved, which strengthens the coupling in the system of equations and ties the solution field in the poorly conditioned part of the linear system to the well-conditioned part via the gradient of the solution field near the physical boundary. This extra *ghost* stabilisation term was expressed by Sticko et al.[5] (amongst others) as an integral over the faces of elements cut/intersected by the physical boundary, Γ, with the following bilinear form

$$j(u_i, w_i) = \sum_{k=1}^{q} \frac{h^{2k+1}}{(2k+1)(k!)^2} \int_{\Gamma} [[\partial_n^k u_i]]\ [[\partial_n^k w_i]] d\Gamma, \tag{16}$$

where q is the polynomial order of the basis functions, h is the size of the element face, w_i and u_i are the test and trial functions (i are the physical dimensions), $\partial_n^k u_i$ is the kth spatial derivative of u_i in the direction normal to the face under

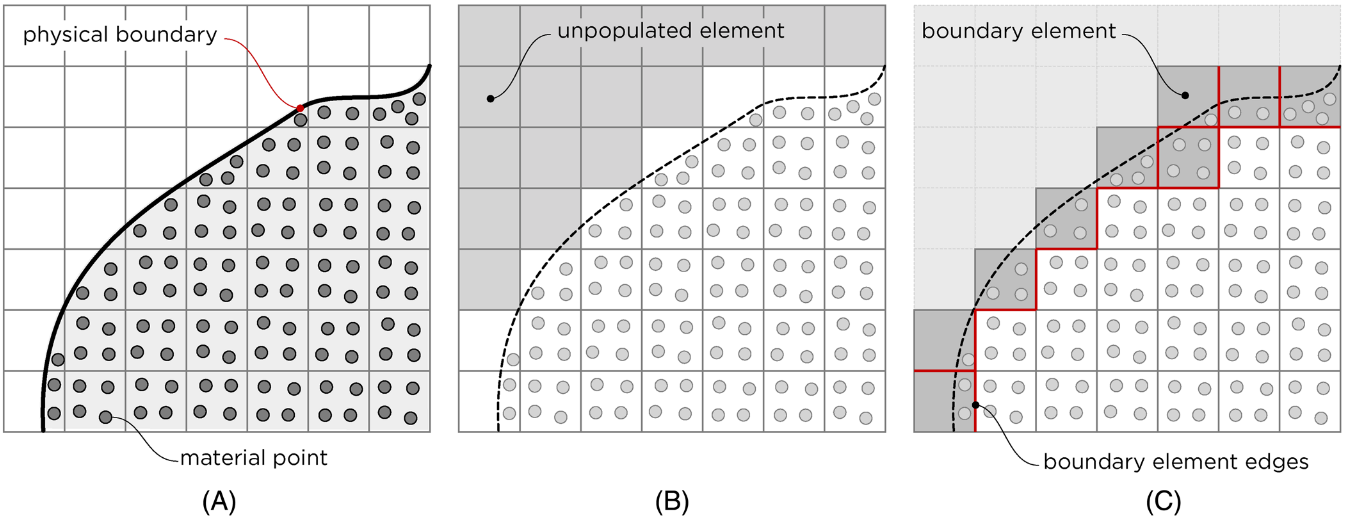

FIGURE 4 Material point method discretisation and boundary representation. (A) MP discretisation. (B) Active/inactive elements. (C) Boundary elements.

consideration, n_i, and $[[\cdot]]$ denotes the jump over a face, F

$$[[u_i]] = u_i|_{F^+} - u_i|_{F^-}. \tag{17}$$

F^+ and F^- denote the faces associated with the *positive* and *negative* elements attached to the boundary between two elements[§§].

3.1 | Boundary identification

A key aspect of the ghost stabilisation approach is identifying the element boundaries that are associated with the physical boundary. The selection of these element boundaries is a clear point of departure between cut-cell finite element and MPMs as most material point simulations do not explicitly define the boundary of the physical domain(s). Therefore, an approach is required to identify these element boundaries without constructing/tracking a representation of the physical boundary.

In this article, the following steps are taken to identify the element boundaries, Γ, for the ghost stabilisation integral:

1. Identify the boundary elements, which are the elements *attached* to (sharing a face with) any unpopulated elements (the light grey-shaded elements in Figure 5). These elements are the dark grey shaded elements shown in Figure 5.
2. The boundary element edges are defined as the boundaries of these elements with: (i) other boundary elements or (ii) elements that are populated by material points (the white-shaded elements in Figure 5). These element boundaries are shown by the thick red lines in Figure 5[¶¶].

For MPMs it is important to integrate over both the faces between boundary elements and the faces between boundary elements and other *active* elements. The stabilisation obtained from integrating over the latter faces will impose continuity between the gradient of the solution of the well-conditioned part of the domain and the partially filled elements. When considering an analysis with multiple bodies, this process should be adopted for each body and then the union of these edges used to determine the stabilisation.

3.2 | Ghost stabilisation for material point analysis

This article is restricted to MPMs that adopt bi-linear quadrilateral elements as the underlying background grid. However, the overall approach is general and it can be applied to other elements, for example with reference to the finite element

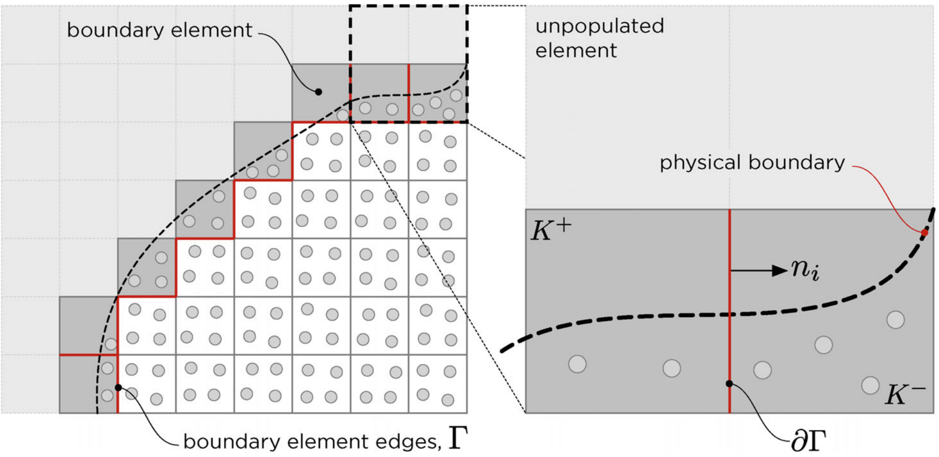

FIGURE 5 Material point method boundaries: Positive/negative elements and normal definition.

literature: higher order quadrilaterals,[5,6,49] including for coupled materials,[50] triangular/tetrahedral elements,[51-53] and different forms of basis functions, such as B-spline based cut/immersed finite element methods,[54-56] amongst others.

For bi-linear quadrilateral elements ($q = 1$) the bi-linear form of the stabilisation term (16) becomes

$$j(u_i, w_i) = \frac{h^3}{3}\int_\Gamma [[\partial_n u_i]]\ [[\partial_n w_i]] d\Gamma, \tag{18}$$

where for a two-dimensional problem the gradient of u_i normal to a boundary can be expressed as

$$\partial_n u_i = (\nabla u)_{ij} n_j = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} \end{bmatrix} \begin{Bmatrix} n_x \\ n_y \end{Bmatrix}. \tag{19}$$

Expanding the jump terms, the stabilisation term for linear elements, (18), can be expressed as

$$j(u_i, w_i) = \frac{h^3}{3}\int_\Gamma \left(\frac{\partial u_i^+}{\partial x_j} n_j - \frac{\partial u_i^-}{\partial x_j} n_j \right)\left(\frac{\partial w_i^+}{\partial x_j} n_j - \frac{\partial w_i^-}{\partial x_j} n_j \right) d\Gamma, \tag{20}$$

where n_j is the outward normal for the positive element, as shown in Figure 5. Expanding (20), introducing the finite element approximation space for the test and trial functions and eliminating the nodal values associated with the test function (full derivation given in A) results in a matrix comprised of four sub components multiplied by the physical displacements of the positive, $\{d^+\}$, and negative, $\{d^-\}$, elements

$$\{f_G\} = \begin{bmatrix} [J_G^{++}] & [J_G^{+-}] \\ [J_G^{-+}] & [J_G^{--}] \end{bmatrix} \begin{Bmatrix} \{d^+\} \\ \{d^-\} \end{Bmatrix} = [J_G]\{d\}, \tag{21}$$

the combined $[J_G]$ matrix can be compactly expressed as

$$[J_G] = \frac{h^3}{3}\int_\Gamma \left([G]^T[m][G]\right) d\Gamma, \qquad \text{where} \qquad [G] = \begin{bmatrix} [G^+] & -[G^-] \end{bmatrix}, \tag{22}$$

and $[m] = [n][n]^T$. For two-dimensional analysis, the normal matrix and the matrix containing the derivatives of the basis functions have the following format

$$[n]^T = \begin{bmatrix} n_x & 0 & 0 & n_y \\ 0 & n_y & n_x & 0 \end{bmatrix} \quad \text{and} \quad [G^+] = \begin{bmatrix} N_1^+,_x & 0 & \dots & N_n^+,_x & 0 \\ 0 & N_1^+,_y & \dots & 0 & N_n^+,_y \\ 0 & N_1^+,_x & \dots & 0 & N_n^+,_x \\ N_1^+,_y & 0 & \dots & N_n^+,_y & 0 \end{bmatrix}.$$

$[G^-]$ has the same format as $[G^+]$, with N_i^+ replaced with N_i^-. See Appendix B for three-dimensional forms of the $[n]$ and $[G^+]$ matrices.

3.3 | Mass stabilisation for dynamics

For a dynamic MPM, mass stabilisation can be added to reduce the conditioning issues associated with small overlaps/contributions from material points to the background grid. This involves adding the following stabilisation matrix to the consistent mass matrix

$$[M_G] = \gamma_M [J_G], \tag{23}$$

where γ_M is a scalar constant that controls how much stabilisation is added to the linear system. Sticko et al.[5] suggested $\gamma_M = \frac{1}{4}\rho$, where ρ is the density of the material under consideration, is a suitable stabilisation parameter. The impact of the variation of this parameter will be explored in the numerical analyses in Section 5. Note that the summation of all of the terms in $[J_G]$ is always equal to zero and therefore the addition of the stabilisation does not change the total mass in the system. However, changing the consistent mass matrix does have consequences for the conservation of linear and angular momentum. In the case of a constant velocity field, the conservation of momentum is conserved irrespective of the magnitude of the stabilisation parameter, γ_M, as the overall mass of the system remains unchanged. For varying velocity fields, ghost stabilisation has the potential to introduce an error in the conservation of momentum but this error will reduce proportional to h^3 (see Equation (22)) with mesh refinement.

3.4 | Stiffness stabilisation of quasi-static analysis

Ghost stabilisation for *quasi*-static analysis is essentially a penalty approach that modifies the *quasi*-static discrete form of the weak form statement of equilibrium (2) to

$$\int_{\varphi_t(K)} [\nabla_x S_{vp}]^T \{\sigma_p\} \mathrm{d}V - \int_{\varphi_t(K)} [S_{vp}]^T \{b\} \mathrm{d}V + \beta \int_{\Gamma} [G]^T [n] \{g\} d\Gamma = \{0\}, \tag{24}$$

where $\beta = \gamma_k h^3/3$ and γ_k is the penalty parameter for ghost stiffness stabilisation[##], and $\{g\}$ is the jump in the displacement gradient over the boundary element edges, Γ

$$\{g\} = [n]^T [G] \{d\}. \tag{25}$$

The third term in (24) is an additional force associated with the ghost stabilisation that penalises changes in gradient of the solution field over the boundary element edges. Linearising (24) with respect to the unknown nodal displacements results in an additional stiffness component linked to this force

$$[K_G] = \frac{\gamma_k h^3}{3} \int_{\Gamma} \left([G]^T [m] [G] \right) d\Gamma = \gamma_k [J_G]. \tag{26}$$

There is significant variation in the value of penalty parameter, γ_k, used in the literature and also variation in the exponent on the face length, h. For example, when enforcing Dirichlet boundary conditions for elastic wave analysis, Sticko et al.[5] adopted the following stiffness stabilisation

$$[K_G^D] = \frac{\gamma_k}{h^2} [J_G],$$

with the penalty parameter set to $\gamma_k = \mu + \frac{\lambda}{2}$, where λ and μ are the Lamé parameters, which sets the stabilisation parameter to be half of the P-wave modulus of the material. Hansbo et al.[6] suggested a modification to this approach, where a different strength of penalisation was imposed on Neumann (traction) and Dirichlet boundaries. The Neumann boundary

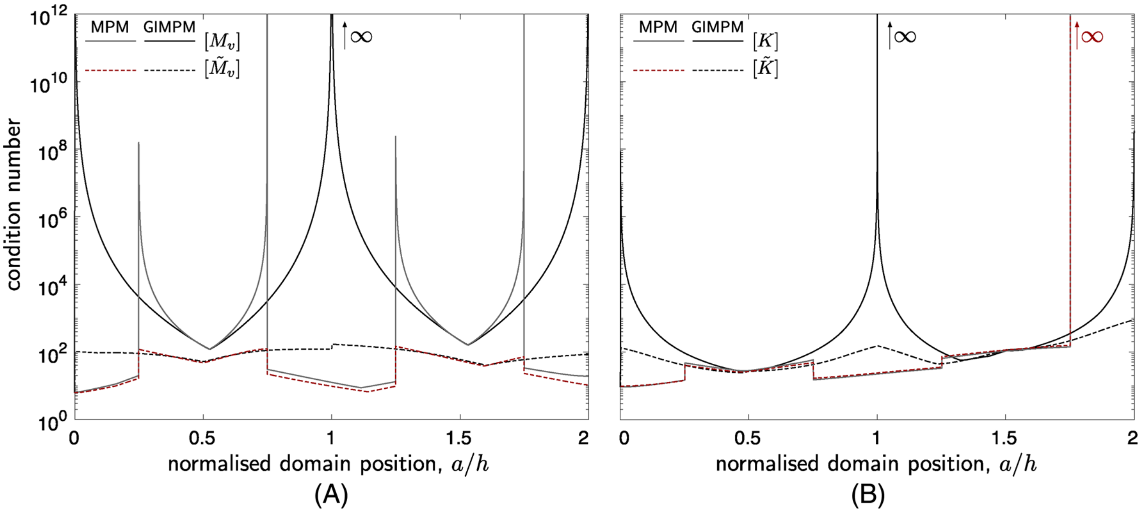

FIGURE 6 Translating material point domain: Condition number variation of (A) the consistent mass and (B) stiffness matrices for the sMPM and GIMPM with and without ghost stabilisation (the stabilised matrices are denoted by the $[\tilde{\cdot}]$).

stabilisation was weakened to

$$[K_G^N] = \gamma_k [J_G],$$

whereas $[K_G^D]$ was adopted on boundaries with Dirichlet (displacement) constraints. Hansbo et al.[6] found that $[K_G^N]$ was sufficient to bound the condition number of the linear system, whilst producing more accurate numerical results due to the less onerous penalisation (via the omission of the h^{-2} multiplier on the stiffness stabilisation). Hansbo et al.[6] set $\gamma_k = (2\mu + \lambda) \times 10^{-4}$, which is much smaller than that used by Sticko et al.[5] for elastic wave problems. Burman[7] set $\gamma_k = (\mu + \lambda) \times 10^{-7}$ when analysing shape optimisation problems and a number of authors have commented that the results are relatively insensitive to the stabilisation parameter. This article is focused on the stabilisation of MPMs due to small overlaps between the background mesh and homogeneous Neumann (traction free) physical boundaries defined by the location of the material points and/or their associated domains and therefore Hansbo et al.'s[6] Neumann boundary stabilisation is adopted. The impact of the value of the stabilisation parameter, γ_k, will be explored in the numerical analysis presented in Section 5.

3.5 | Ghost stabilised conditioning

The trivial test problem described in Section 2.6 is now re-examined with ghost stabilisation. Figure 6 shows the variation of the condition numbers of the stabilised, $[\tilde{\cdot}]$, and unstabilised consistent mass and stiffness matrices as the domain translates through the background mesh for the sMPM and the GIMPM. The following stabilisation parameters were adopted

$$\gamma_M = \frac{1}{4}\rho \qquad \text{and} \qquad \gamma_K = 1.$$

The ghost stabilisation removes the spikes in the condition number of the consistent mass matrix for both the sMPM and the GIMPM. The is most clearly seen through the maximum condition numbers reported in Table 1. The stabilisation has also removed the spikes in the conditioning of the GIMPM stiffness matrix, reducing the maximum condition number by 24 orders of magnitude. The stiffness matrix of the MPM is well conditioned for $a/h \in [0, 7/4)$ and therefore the stabilisation has minimal impact (note Table 1 reports the maximum MPM stiffness condition number for $a/h \in [0, 7/4)$). The stabilisation does not remove the requirement to constrain rigid body motion and beyond $a/h = 7/4$ the condition of the stabilised stiffness matrix is effectively $\kappa([\tilde{K}]) = \infty$ as the domain is not constrained in the horizontal direction.

TABLE 1 Translating material point domain: Condition number variation of the consistent mass and stiffness matrices for the sMPM and GIMPM with and without ghost stabilisation.

	$\max(\kappa([\boldsymbol{M}_v]))$	$\max(\kappa([\tilde{\boldsymbol{M}}_v]))$	$\max(\kappa([\boldsymbol{K}]))$	$\max(\kappa([\tilde{\boldsymbol{K}}]))$
MPM	9.08×10^{26}	1.53×10^{2}	4.72×10^{1}[a]	4.22×10^{1}[a]
GIMPM	4.43×10^{43}	1.63×10^{2}	3.32×10^{26}	1.66×10^{2}

[a] Maximum for $a/h \in [0, 7/4)$.

The maximum time step size for an explicit time stepping algorithm is limited by the Courant–Friedrichs–Lewy (CFL) number, C_{CFL}, via

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

where α is a constant that depends on the adopted time stepping algorithm.[5] 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 $[K]\{x\} = \lambda[M]\{x\}$. The ideal situation is where the C_{CFL} number is independent of the position of the physical domain relative to the background mesh and scales linearly with the size of the elements in the background grid. However, ill-conditioning of the mass and stiffness matrices can cause severe time step limitations due to the unbounded nature of the matrices' eigenvalues.[5] This issue is demonstrated in Figure 7, where the CFL number is given as the physical domain translates through the background mesh for the sMPM and GIMPM with and without ghost stabilisation. The CFL number for the non-stabilised methods are highly dependent on the position of the physical body relative to the background mesh, mirroring the results shown in Figure 6 in terms of the condition numbers of the mass and stiffness matrices. The CFL number for the ghost stabilised methods are almost independent of the domain's position on the background grid.

This conceptual problem has demonstrated the impact of the ghost stabilisation technique on the conditioning, and therefore accuracy and stability, of the consistent mass and stiffness matrices. It has taken two matrices that are effectively un-invertible for certain intersections of the background mesh with the physical domain introduced an additional term

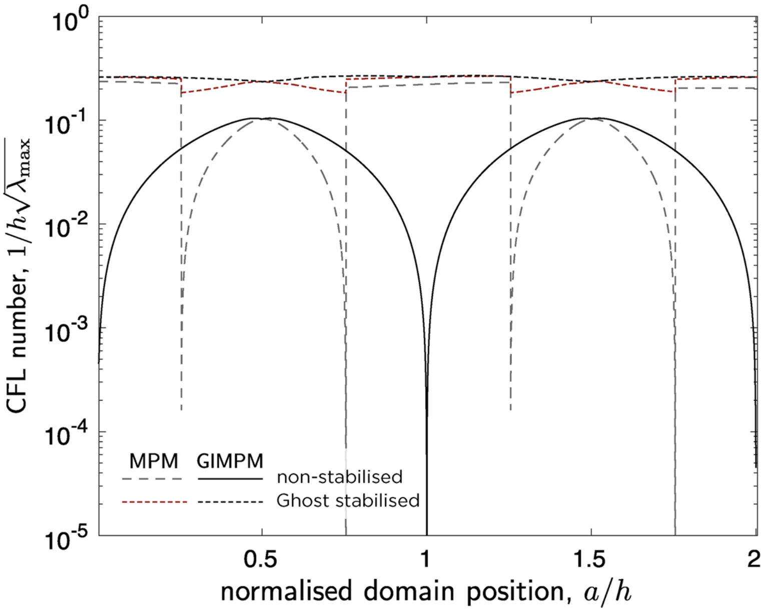

FIGURE 7 Translating material point domain: CFL number with and without ghost stabilisation.

into the linear system that stabilises the equations and removes the strong dependency of the position of the domain within the background mesh. It also removes the CFL number dependence on the position physical domain within the background grid. The impact of this stabilisation will be demonstrated further via the numerical examples presented in Section 5, prior to this key aspects of the implementation of the ghost stabilisation within MPMs will be detailed.

4 | IMPLEMENTATION

Implementation of the ghost stabilisation technique only requires minor modification to standard MPM code. These key differences are described in this section.

4.1 | Data structures

The ghost stabilisation technique requires information on the *skeleton*[‖‖] of the background mesh that is not routinely stored for the MPM (or continuous Galerkin finite element methods), namely:

face connectivity between elements: this is simply a unique list of all of the internal faces in the analysis and the elements to which they are connected.

face nodal topology: although this could be derived from the face connectivity between elements, it is convenient to also store the nodes that are connected to each face, in a similar way to storing the topology of each element in conventional finite element methods. This face topology information will be used to determine face lengths, normal directions, global quadrature point locations and so forth.

4.2 | Boundary element edge identification

Determining the boundary element edges is a critical stage of the ghost stabilisation technique and needs to be performed at each time/load step. 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 Figure 4B/C);

(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);

(iii) using the faces from (ii), identify the *boundary elements* (the dark grey shaded elements in Figure 4C);

(iv) using the boundary elements from (iii), loop over each face of the element and if both elements associated with the face are *active*, flag as a *boundary element edge* (the thick red lines in Figure 4C).

4.3 | Ghost stabilisation determination

The boundary element edges, Γ, identified using the process described in Section 4.2, must be integrated over to form the ghost stabilisation matrix (22). Gauss–Legendre quadrature is a convenient and efficient choice, allowing $[J_G]$ to be approximated as

$$[J_G] \approx A_{\forall\Gamma} \left(\frac{h^3}{3} \sum_{i=1}^{n_{Gp}} \left([G_i]^T [m] [G_i] \right) w_i \det([\mathcal{J}_i]) \right), \tag{27}$$

where the $(\cdot)_i$ subscripts denote quantities that are potentially dependent on the Gauss point location, w_i are the Gauss point weights, $\det([\mathcal{J}])$ is the determinant of the surface Jacobian (the ratio of the global/local face lengths, $h/2$ for two-dimensional linear elements) and n_{Gp} is the number of Gauss points used to approximate the integral.

The implicit *quasi*-static approach described in Section 2.4 adopts an updated Lagrangian approach and therefore both the normal to each face and the spatial derivatives of the basis functions will change over the load step. However, there would be minimal benefit to including this variation as the ghost stabilisation is not a physical quantity. The *cost* of including this variation would be that the stabilisation matrix and the associated force contribution to the weak statement of equilibrium would need to be recalculated at each Newton–Raphson iteration within every load step. Assuming that the additional stiffness is constant over a load step also allows the force contribution from the ghost stabilisation to the weak statement of equilibrium to be calculated via the product of the ghost stabilisation stiffness matrix, $[K_G]$, with the background grid nodal displacements from the current step. In summary, as the stabilisation is a numerical fix to improve the stability of the method, it is assumed that $[J_G]$ is constant over each load/time step.

5 | NUMERICAL INVESTIGATIONS

All of the simulations presented in this section are conducted under the assumption of plane strain and adopt a background grid of bi-linear quadrilateral elements. The GIMPM is adopted for all analyses, unless stated otherwise.

5.1 | Mass stabilisation: Translating, rotating, and stretching

The first set of numerical analyses are focused on mass stabilisation and key steps that are required in dynamic material point simulations, namely the mapping of velocity between the material points and the grid at the start of a time step.

5.1.1 | Mass stabilisation: Rigid body translation

This experiment investigates the variation of the condition number of the consistent mass matrix of the active elements as a ghost emoji translates through a regular background mesh. A unit background mesh of bi-linear square elements was adopted and the ghost had a height and width of 0.4 m, as shown in Figure 8. The ghost underwent a rigid body motion of

$$\{u\} = \{0.5 \quad 0.5\}^T \text{ m},$$

over 1000 steps and the condition number of the global (non-stabilised) consistent mass matrix, $[M_v]$, was recorded at each step along with the condition of the stabilised, $[\tilde{M}_v]$, and lumped, $[\overline{M}_v]$, mass matrices. The ghost had a density of $\rho = 1$ kg/m^3 and was represented by 646 generalised interpolation material points[†††], as shown in Figure 8.

Figure 9 shows the evolution of the condition number of the consistent (black line), stabilised (thick red line), and lumped (grey dashed line) mass matrices as the ghost translates through the background mesh with $h = 0.0312$ m and a mass stabilisation parameter of $\gamma_M = \frac{1}{4}\rho$. The maximum condition number of the consistent mass matrix was 1.87×10^{41}, whereas the stabilised mass matrix had a maximum value of 4.33×10^5.

Table 2 explores the variation of the maximum condition number of the non-stabilised, stabilised and lumped mass matrices with different numbers of generalised interpolation material points with $\gamma_M = 0.25$. Due to the significant difference between the stabilised and non-stabilised matrices, the condition number of the stabilised matrix scales approximately linearly with the stabilisation parameter, γ_M. That is, reducing/increasing the stabilisation parameter by an order of magnitude will increase/reduce the condition number of the stabilised mass matrix by an order of magnitude until the condition number of $[J_G]$ approaches $[M_v]$.

Velocity mapping

The stabilised and non-stabilised consistent mass matrices were also used to map the velocity from the coarse material point distribution to nodes using (10). In this numerical experiment $\{v_p\}$ was set to

$$\{v_p\} = \{1 \quad 1\}^T,$$

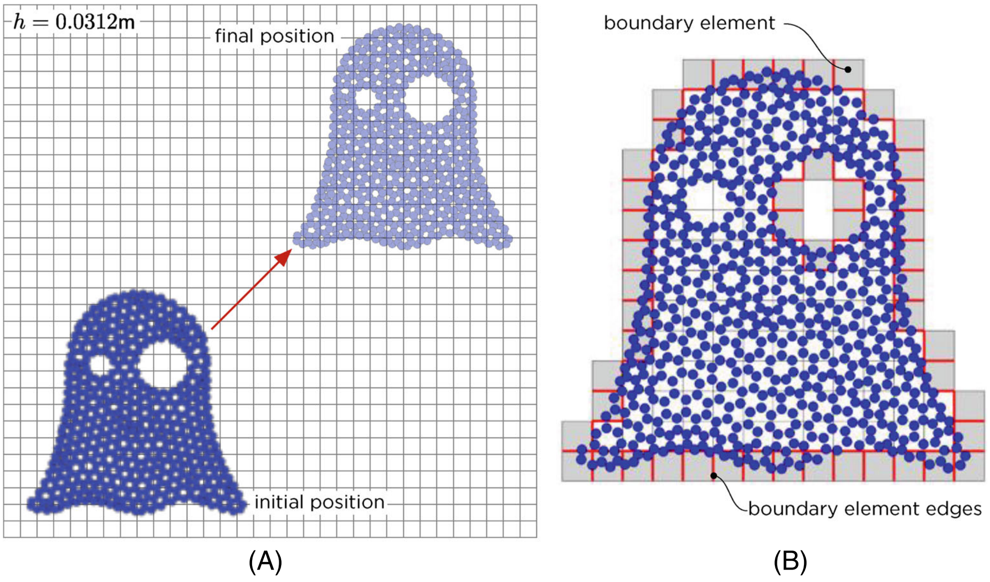

FIGURE 8 Translating ghost discretisation: (A) Initial and final material point positions and (B) boundary elements (grey shaded) and the faces (thick red lines) for the initial material point distribution (only the *active* elements in the background mesh are shown).

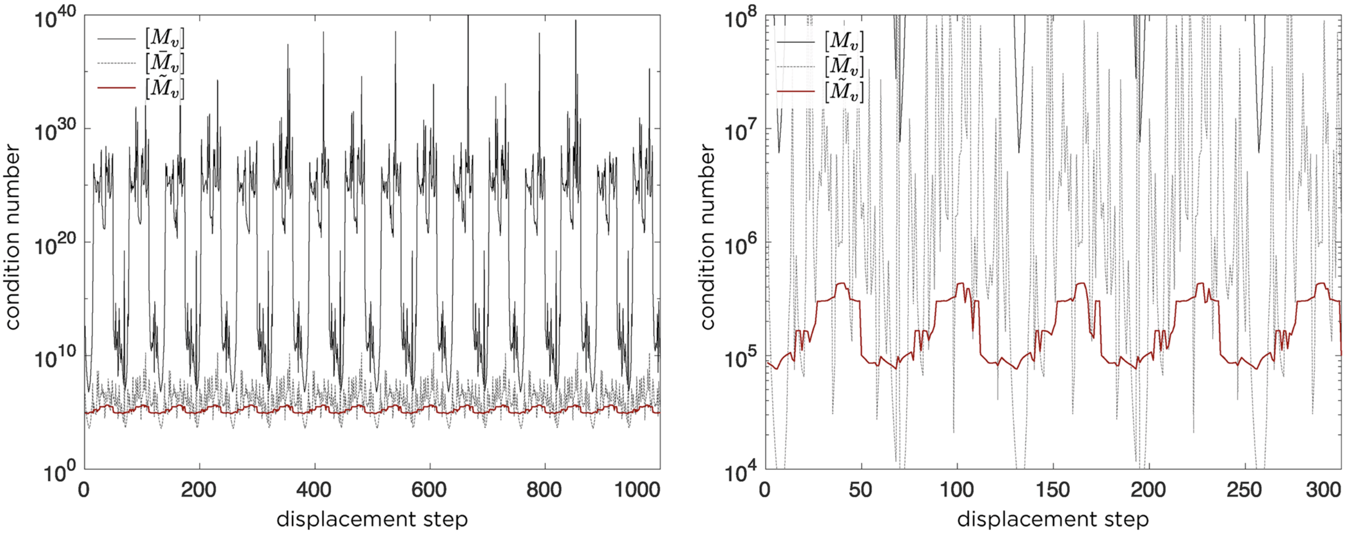

FIGURE 9 Translating ghost: Mass matrix condition number variation with displacement step (the stabilised matrices are denoted by $[\tilde{\cdot}]$ and the lumped by $[\bar{\cdot}]$).

and therefore to be consistent with this constant velocity field, all *active* nodes in the background mesh should have a velocity equal to $\{v_v\} = \{1 \quad 1\}^T$. The variation of the maximum nodal velocity with displacement step for the stabilised (red line) and non-stabilised consistent (black line) mass matrices are shown in Figure 10. The nodal values of velocity obtained from the non-stabilised consistent mass matrix are highly dependent on the position of the ghost relative to the background mesh, whereas the stabilised mass matrix correctly predicts nodal velocities of $\{1 \quad 1\}^T$, with a maximum error of 3.32×10^{-13}, irrespective of the position of the ghost within the background mesh. The lumped mass matrix correctly predicts the nodal velocities as, for constant velocity fields, the summation of the mass basis functions in the lumped matrix and nodal momentum cancel out irrespective of the basis function values, allowing the lumped mass matrix to achieve machine precision in the nodal velocities with a maximum error of 1.11×10^{-15}. This is however only true for velocity fields that are constant in space, as will be demonstrated in the next section.

TABLE 2 Translating ghost: Maximum mass matrix condition number for different numbers of material points per ghost with $h = 0.0312$ and $\gamma_M = 0.25\rho$.

	Coarse	Medium	Fine
No. material points/ghost	**646**	**2389**	**9613**
Non-stabilised $[M_v]$	1.87×10^{41}	2.38×10^{36}	3.01×10^{33}
Stabilised $[\tilde{M}_v]$	4.33×10^{5}	3.12×10^{5}	2.97×10^{5}
Lumped (diagonal) $[\overline{M}_v]$	1.71×10^{10}	7.66×10^{11}	1.20×10^{11}

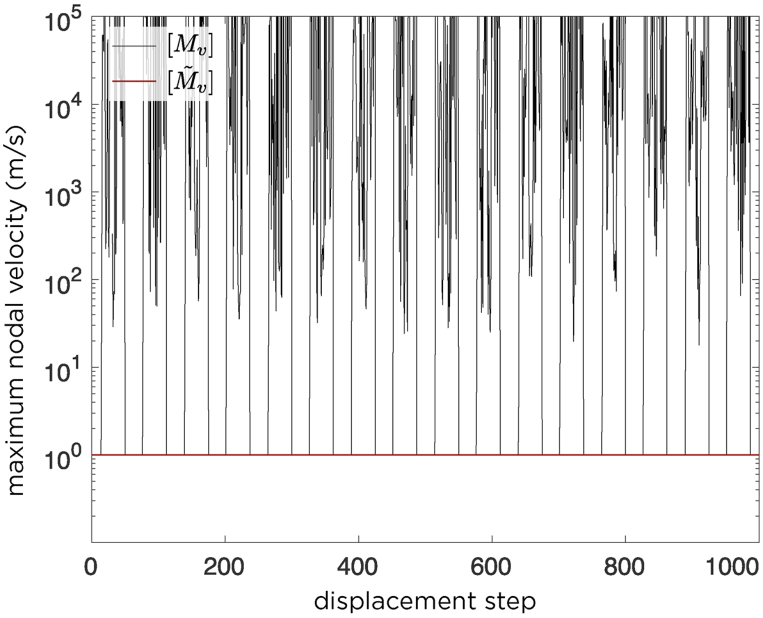

FIGURE 10 Translating ghost: Maximum nodal velocity variation with displacement step (the result from the stabilised matrix is denoted by $[\tilde{\cdot}]$).

5.1.2 | Explicit dynamics: Stretching ghost and velocity update

This experiment investigates the differences in the velocity mapping process using the consistent and lumped mass matrices as a ghost emoji expands on a regular background mesh using the explicit dynamic material point formulation described in Section 2.3. A unit background mesh with $h = 0.1$ m was adopted. The ghost had a height and width of 0.4 m and was centred within the background mesh, as shown by the dark blue points in Figure 11A. The *coarse* material point discretisation (646 material points), as described in the previous section, was adopted for this analysis. For this artificial problem, the ghost emoji had zero stiffness ($E = 0$ Pa) and a density of $\rho = 1000$ kg/m^3. The effect of gravity was ignored and the behaviour modelled over 1 second using 1000 time steps. A mass stabilisation parameter of $\gamma_M = \frac{1}{4}\rho$ was adopted for the ghost-stabilised simulations. The problem was initialised such that the material points had a velocity of

$$\{v_p^0\} = \frac{1}{t_t}\left\{\{x_p\} - \{x_{\text{cen}}\}\right\},$$

where $\{x_{\text{cen}}\} = \{0.5 \quad 0.5\}^T$ m was the centre of ghost emoji and $t_t = 1$ s the total simulation time. This velocity field represents a uniform expansion about $\{x_{\text{cen}}\}$. As the ghost has zero stiffness, there is nothing to resist the expansion of the body and the material points should move with a constant velocity throughout the analysis, doubling the size of the ghost over 1 second (as shown by the light blue points in Figure 11A). This means that at any point in the analysis, the norm of

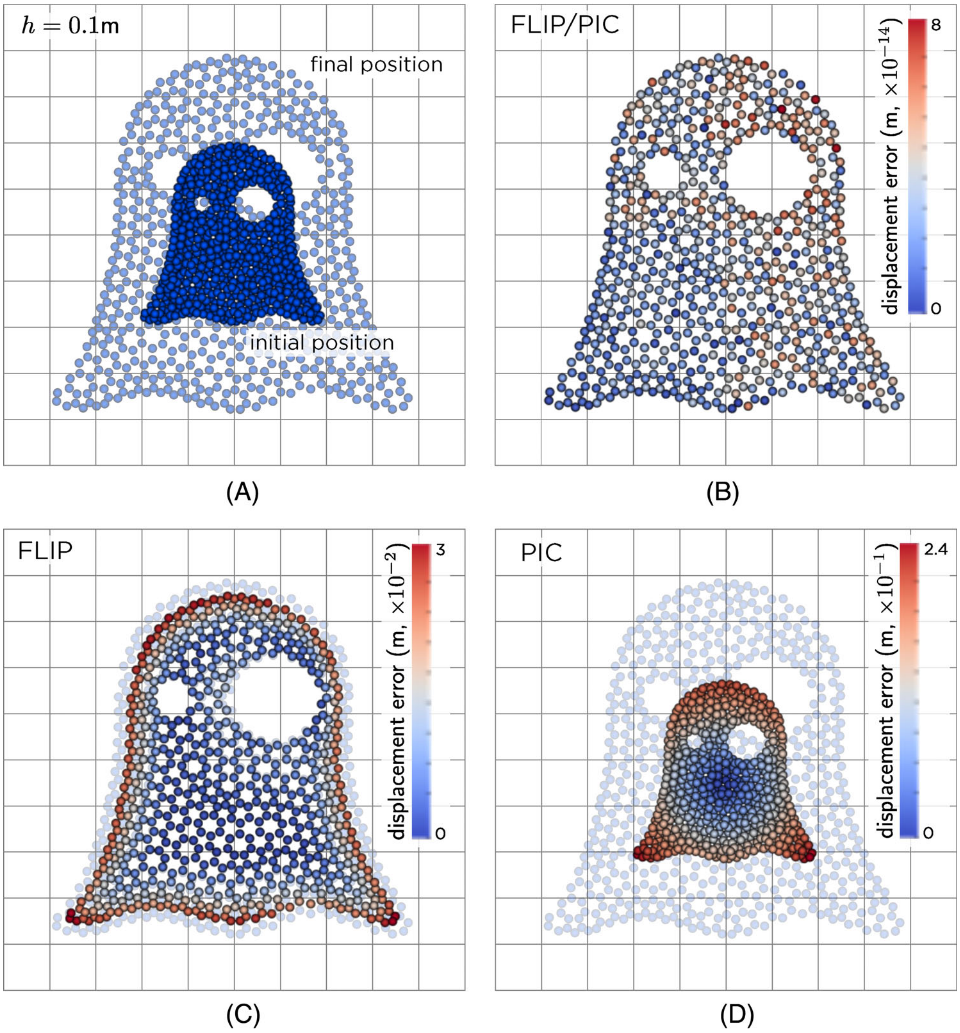

FIGURE 11 Stretching ghost: (A) initial discretisation and expected final material point positions for $h = 0.1$ m and the material point positions, coloured according to the norm of the displacement error, at the end of the analysis for: (B) the ghost stabilised consistent mass matrix and the lumped mass matrix using a (C) FLIP and (D) PIC material point velocity update.

the displacement error for any material point in the body can be evaluated using the following Euclidean (or L^2) norm

$$u^p_{\text{error}} = \left|\{u_p\} - t\{v^0_p\}\right|,$$

where $\{u_p\}$ is the displacement of a material point at time t. In terms of deformation, all material points should have a deformation gradient of $[F] = (1 + t)[I]$ with a volume ratio of $J = (1 + t)^2$. Note that it was not possible to use the non-stabilised consistent mass matrix for this analysis, with the simulation diverging after 21 time steps.

Figure 11B,C show the material point positions, coloured according to the norm of the displacement error, at the end of the analysis for the (b) ghost stabilised consistent and (c) lumped mass matrices using a FLIP velocity update (8). It is worth highlighting that the scales on the colour bars are different for the two figures, with Figure 11C having a range of $[0, 3] \times 10^{-2}$ whereas Figure 11B has a range 12 orders of magnitude smaller, at $[0, 8] \times 10^{-14}$. Figure 11C also shows the expected positions of the material points at the end of the analysis using light blue shaded circles. Figure 11D shows the final material point positions using a PIC material point velocity update[‡‡‡], where again the material points are coloured according to the norm of the displacement error. A number of points can be observed from the results presented in Figure 11B–D:

(i) for this zero acceleration problem, a FLIP material point velocity update maintains the correct velocities at the material points throughout the analysis irrespective if a lumped of consistent mass matrix is adopted[25];

(ii) the ghost stabilised consistent mass matrix correctly predicts the material point velocities throughout the analysis with a displacement error of less than 10^{-13} m across all material points and time steps;

(iii) the lumped mass matrix causes in an error in the updated material point positions for non-rigid body motions due to errors in the mapping of the material point velocities to the background grid vertices using (10) as these nodal velocities are used to update the material point positions at the end of the time step such that the deformation of the material points are consistent with the background grid. Therefore, adopting a lumped mass matrix means that the material point velocities are no longer consistent with their deformation over the time step, which is mapped from the grid incremental displacements via (9);

(iv) the ghost stabilised consistent mass matrix predicts vertex velocities, $\{v_v\}$, that are consistent with the physical linear velocity variation due to the ghost stabilisation penalising variations of the gradient of the basis functions across the boundary element edges this means that the method predicts the correct material point displacements (and velocities) using both FLIP and PIC velocity updates; and

(v) combining a PIC velocity update with a lumped mass matrix destroys the initial material point velocity field due to inaccuracies in the vertex velocities (as explained under point (ii) and also by Pretti et al.[25]), leading to a physically unrealistic solution as the total velocity field is repeatedly mapped between the material points and the grid vertices.

This simple problem has demonstrated the benefits of adopting a ghost stabilised consistent mass matrix over a lumped mass matrix. It is important to highlight that adopting a lumped mass matrix also has consequences for the energy conservation of MPMs (as discussed in detail by Love and Sulsky[23]). This point is explored in the following numerical example.

5.2 | Explicit dynamics: Collision of elastic ghosts

This example considers the collision of two elastic ghosts and has been designed to investigate the energy conservation implications of adopting a lumped or a stabilised consistent mass matrix. The analyses were run on a 1 m×1 m background grid, with 5, 10, and 20 elements in each direction ($h = 0.20, 0.10,$ and 0.05 m). The ghosts were initially placed in the upper right and lower left corners of the background grid, with velocities of $\{-0.1, -0.1\}$ m/s and $\{0.1, 0.1\}$ m/s respectively (as shown in Figure 12). Each ghost had width and height of 0.4 m, a Young's modulus of $E = 1000$ Pa, a Poisson's ratio of $\nu = 0.3$ and a density of $\rho = 1000$ kg/m^3. The same three generalised interpolation material point resolutions used in the previous sections, and detailed in Table 2, were adopted for this analysis. The effect of gravity was ignored and the

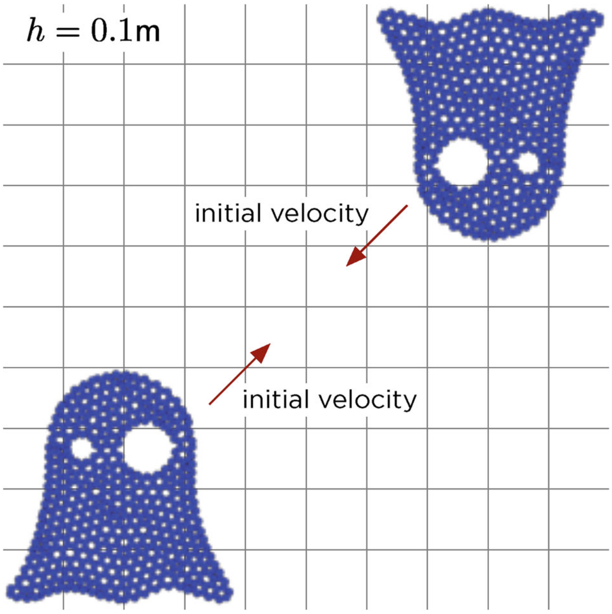

FIGURE 12 Collision of elastic ghosts: Initial *coarse* material point discretisation with $h = 0.1$ m.

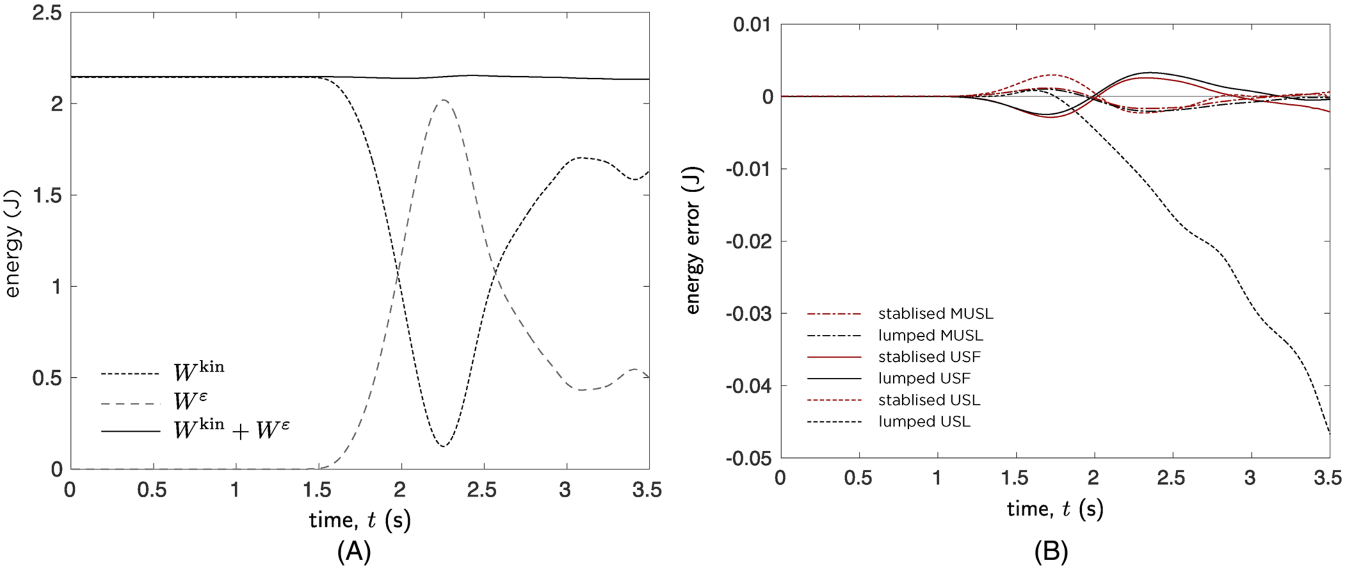

FIGURE 13 Collision of elastic ghosts: (A) USF ghost-stabilised energy variation for $h = 0.1$ m, $n_t = 1000$ and the *medium* material point discretisation and (B) energy error for $h = 0.1$ m, $n_t = 2000$ and the *medium* material point discretisation.

behaviour modelled over 3.5 s using a total of $n_t = 1000$, 2000, and 4000 time steps. The system was undamped and was analysed using the explicit dynamic material point formulation described in Section 2.3 with a lumped and stabilised consistent mass matrix ($\gamma_M = \frac{1}{4}\rho$). USF, USL, and MUSL[18] material point stress updating procedures were considered. The total initial energy in the system for all of the analyses was $W_0 = 2.14726$ J.

Note that it is not possible to analyse this problem using a non-stabilised consistent mass matrix for any of the combinations of parameters used with the lumped/stabilised analyses. When trying to use the non-stabilised consistent mass matrix, the simulation becomes unstable after a small number of time steps and the ghosts *explode* due to spurious nodal velocity values associated with the inversion of $[M_v]$.

Figure 13A shows the USF ghost-stabilised energy variation for $h = 0.1$ m, $n_t = 1000$ and the *medium* (2389 material points for each ghost, see Section 5.1.1) material point discretisation and Figure 13B shows the energy error

$$\text{energy error} = W_i^{\rm kin} + W_i^{\varepsilon} - W_0,$$

where $W_i^{\rm kin}$ and W_i^{ε} are the kinetic and strain energies associated with time step i, for $h = 0.1$ m, $n_t = 4000$ and the *medium* material point discretisation over the duration of the analysis for the lumped and stabilised consistent mass matrices using the USF, USL and MUSL approaches. As reported in the literature,[22,23] combining an USL approach with a lumped mass matrix (dashed black line) results in significant energy dissipation. This excessive dissipation is removed via the adoption of the ghost stabilised consistent mass matrix (dashed red line), resulting a simulation that approximately mirrors the USF analysis (solid red line) in terms of the energy dissipated at any point in the analysis. The lumped and stabilised consistent mass matrices yield very similar results in terms of the energy error for the USF and MUSL approaches.

Table 3 provides the normalised mean energy error

$$\text{normalised mean error} = \frac{1}{n_t W_0}\sum_{i=1}^{n_t}\left|W_i^{\rm kin} + W_i^{\varepsilon} - W_0\right|,$$

for the 162 analyses, where n_t is the total number of time steps. The numbers are coloured according to the magnitude of the average energy error, when yellow and green correspond to the minimum and maximum errors, respectively.

A number of points can be observed from Table 3:

(i) increasing the number of time steps reduces the average energy error over the simulation for all analyses;

TABLE 3 Collision of elastic ghosts: Normalised mean energy errors (×10^{-3}) for different discretisations using USF, USL, and MUSL approaches and the lumped and ghost-stabilised consistent mass matrices.

(a) USF ghost coarse

	h (m)	0.20	0.10	0.05
time steps	1000	1.370	0.888	1.386
	2000	0.682	0.445	0.707
	4000	0.340	0.223	0.357

(b) USF ghost medium

	h (m)	0.20	0.10	0.05
time steps	1000	1.294	0.889	1.559
	2000	0.643	0.447	0.792
	4000	0.321	0.224	0.399

(c) USF ghost fine

	h (m)	0.20	0.10	0.05
time steps	1000	1.261	0.899	1.629
	2000	0.627	0.451	0.828
	4000	0.313	0.226	0.418

(d) USF lumped coarse

	h (m)	0.20	0.10	0.05
time steps	1000	1.418	0.956	0.871
	2000	0.706	0.478	0.438
	4000	0.352	0.239	0.220

(e) USF lumped medium

	h (m)	0.20	0.10	0.05
time steps	1000	1.325	0.916	1.011
	2000	0.660	0.458	0.511
	4000	0.329	0.229	0.257

(f) USF lumped fine

	h (m)	0.20	0.10	0.05
time steps	1000	1.283	0.908	1.122
	2000	0.638	0.454	0.567
	4000	0.319	0.227	0.285

(g) USL ghost coarse

	h (m)	0.20	0.10	0.05
time steps	1000	4.533	0.859	0.980
	2000	2.738	0.493	0.557
	4000	1.542	0.272	0.409

(h) USL ghost medium

	h (m)	0.20	0.10	0.05
time steps	1000	4.319	0.751	1.242
	2000	2.576	0.346	0.519
	4000	1.435	0.174	0.216

(i) USL ghost fine

	h (m)	0.20	0.10	0.05
time steps	1000	4.183	0.789	1.425
	2000	2.440	0.368	0.639
	4000	1.337	0.175	0.275

(j) USL lumped coarse

	h (m)	0.20	0.10	0.05
time steps	1000	12.32	8.417	10.58
	2000	6.647	4.726	6.790
	4000	3.460	2.521	3.989

(k) USL lumped medium

	h (m)	0.20	0.10	0.05
time steps	1000	12.39	8.207	8.899
	2000	6.684	4.607	5.674
	4000	3.481	2.457	3.319

(l) USL lumped fine

	h (m)	0.20	0.10	0.05
time steps	1000	12.35	7.992	8.022
	2000	6.666	4.484	5.105
	4000	3.473	2.391	2.982

(m) MUSL ghost coarse

	h (m)	0.20	0.10	0.05
time steps	1000	1.699	0.997	1.238
	2000	0.853	0.499	0.613
	4000	0.427	0.250	0.305

(n) MUSL ghost medium

	h (m)	0.20	0.10	0.05
time steps	1000	1.619	0.938	1.326
	2000	0.813	0.469	0.653
	4000	0.408	0.234	0.324

(o) MUSL ghost fine

	h (m)	0.20	0.10	0.05
time steps	1000	1.585	0.929	1.359
	2000	0.796	0.464	0.670
	4000	0.399	0.232	0.332

(p) MUSL lumped coarse

	h (m)	0.20	0.10	0.05
time steps	1000	1.369	1.221	1.088
	2000	0.685	0.612	0.550
	4000	0.343	0.306	0.276

(q) MUSL lumped medium

	h (m)	0.20	0.10	0.05
time steps	1000	1.284	1.128	0.932
	2000	0.643	0.565	0.466
	4000	0.322	0.283	0.233

(r) MUSL lumped fine

	h (m)	0.20	0.10	0.05
time steps	1000	1.246	1.097	0.905
	2000	0.624	0.550	0.452
	4000	0.312	0.275	0.226

Note: The cells are coloured according to the magnitude of the error from yellow (small) to green (large).

TABLE 4 Collision of elastic ghosts: Normalised mean energy error ($\times 10^{-3}$) variation with γ_M for USF with coarse ghosts and 1000 time steps.

	Normalised penalty parameter, γ_M/ρ											
***h* (m)**	2^2	2^0	2^{-2}	2^{-4}	2^{-6}	2^{-8}	2^{-10}	2^{-12}	2^{-14}	2^{-16}	2^{-18}	2^{-20}
0.20	0.834	1.197	1.370	1.414	1.420	1.407	1.390	1.388	1.375	1.375	1.388	-
0.10	0.858	0.872	0.888	0.919	0.925	0.940	0.953	0.943	-	-	-	-

(ii) the ghost stabilised consistent mass matrix is more sensitive to changes in background grid resolution compared to the lumped mass matrix when an USF approach is adopted, with the average energy error being smaller for the $h = 0.2$ and 0.1 m analyses when using $[\tilde{M}_v]$ than $[\overline{M}_v]$ but the opposite being true for the $h = 0.05$ m analysis;

(iii) the combination of an USL approach with a lumped mass matrix, $[\overline{M}_v]$, yields excessive energy dissipation, as shown by sub Table 3j–l, where the average energy error an order of magnitude larger than that of the USL with a stabilised consistent mass matrix, $[\tilde{M}_v]$, for mesh sizes $h = 0.1$ and 0.05 m;

(iv) the errors associated with the lumped and stabilised MUSL approaches are very similar for all combinations of h and n_t, with no clear trend in terms of the advantages of one approach over the other in terms of energy conservation.

The final result in this section is a study on the impact the magnitude of the mass stabilisation parameter, γ_M, on the energy conservation of the elastic ghost collision problem with the USF approach. The results are summarised in Table 4, which details the normalised mean energy error for analyses with a coarse ghost and 1000 time steps with $h = 0.2$ and 0.1 m. The mass stabilisation parameter is varied between $\gamma_M = 2^2\rho$ and $\gamma_M = 20^{-20}\rho$. Most of the analyses are stable, but the analyses fail to complete for $\gamma_M < 20^{-18}\rho$ and $\gamma_M < 20^{-12}\rho$ for the $h = 0.2$ and 0.1 m background grid sizes, respectively, as the penalty is insufficient to stabilise the consistent mass matrix. Note that the energy error reduces with increasing penalty but this is at the expense of increasing stiffness of the results due to the gradient of the solution being increasing constrained over the boundary of the physical domain. Therefore, the selection of the penalty parameter is a compromise between stability of the analyses whilst minimising the impact on the physical response of the problem. The experience of the author is that the $\gamma_M = \rho/4$ value suggested in the cut FEM literature[5] is a good starting point to balance these issues.

The analyses presented for this problem demonstrate that the ghost stabilisation allows the consistent mass matrix to be used for explicit dynamic analysis and also corrects an issue with the USL approach when combined with a lumped mass matrix. The following section will analyse the impact of including elasto-plasticity.

5.3 | Explicit dynamics: Collision of elasto-plastic ghosts

This example considers the collision of two elasto-plastic ghosts to demonstrate that the ghost stabilisation approach can be adopted for non-linear materials. The analyses were run on a 1 m×1 m background grid, with 20 elements in each direction ($h = 0.05$ m). The ghosts were initially placed in the upper right and lower left corners of the background grid, with velocities of $\{-0.1, -0.1\}$ m/s and $\{0.1, 0.1\}$ m/s respectively (as shown in Figure 12). Each ghost had width and height of 0.4 m, a Young's modulus of $E = 1000$ Pa, a Poisson's ratio of $\nu = 0.3$ and a density of $\rho = 1000$ kg/m^3. Yielding of the material was governed by a von Mises function of the form

$$f = \varrho - \varrho_y = 0, \tag{28}$$

where $\varrho = \sqrt{2J_2}$, $J_2 = s_{ij}s_{ji}/2$, $s_{ij} = \tau_{ij} - \tau_{kk}\delta_{ij}/3$ and τ_{ij} is the Kirchhoff stress. The deviatoric yield stress of both ghosts was set to $\varrho_y = 100$ Pa and the constitutive model adopted a elastic predictor, plastic corrector closest point projection algorithm (see Coombs[57] amongst others). The *medium* generalised interpolation material point resolution used in the previous sections, and detailed in Table 2, was adopted for this analysis. The effect of gravity was ignored and the behaviour modelled over 3.5 s using 1000 time steps. The system was undamped and was analysed using the USF explicit dynamic material point formulation described in Section 2.3 with a stabilised consistent mass matrix ($\gamma_M = \frac{1}{4}\rho$).

Figure 14 shows the evolution of the energy components over the analysis, showing the kinetic (W^{kin}, dashed black line) and strain (W^{ε}, dashed grey line) energies, the plastic dissipation (W^{p}, red line) and the total energy of the bodies ($W^{\text{kin}} + W^{\varepsilon}$, black line). The initial collision starts at approximately 1.5 s, which initiates strain energy generation and

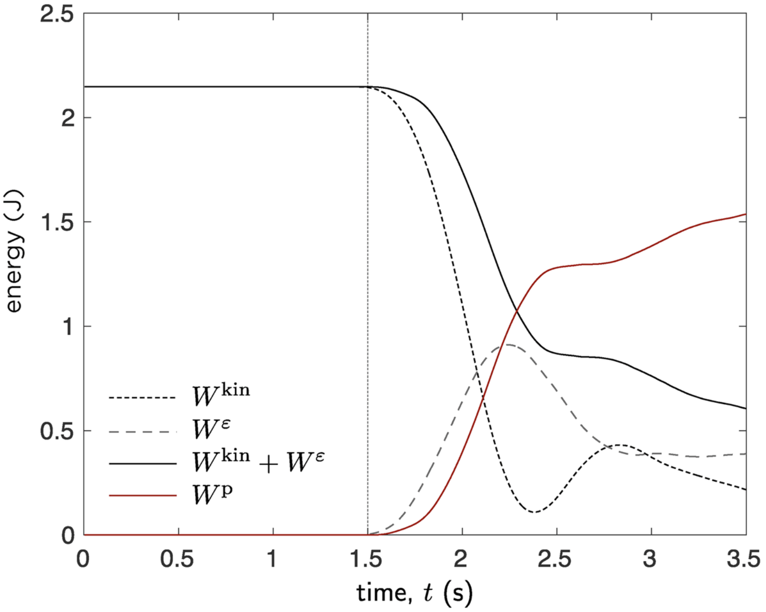

FIGURE 14 Collision of elasto-plastic ghosts: Energy components for a *medium* material point discretisation, a background mesh size of $h = 0.05$ m and 1000 time steps.

plastic dissipation within the ghosts. Beyond this initial collision there is additional plastic dissipation from around 2.7 s due to stress waves through the material. The plastic deformation also *traps* a degree of strain energy in the ghosts, as seen by the plateau in the strain energy beyond 3.0 s.

Figure 15 shows the material point positions at $t = 1.5$, 2.0, 2.5, and 3.0 s, coloured according to the magnitude of the total plastic logarithmic strains. It is worth noting that the native no slip contact in MPMs is based on the background grid and the proximity between the two bodies when contact occurs is based on the grid spacing rather than a description of the surface of the objects. With this native formulation (based on the work of Sulsky et al.[1]) *contact* occurs when material points from the two bodies interact via shared degrees of freedom on the background grid. This means that the bodies always maintain some separation proportional to the size of the background grid. Several papers (for example, see relevant papers in the MPM contact literature,[58-65] amongst others, and de Vaucorbeil et al.[2] and Solowski et al.[3] for recent reviews) have investigated alternative contact formulations in the MPM but these techniques are not explored in this article, where the focus is on the stabilisation of MPMs in general and comparison with existing techniques, which will be explored in the next example.

5.4 | Explicit dynamics: Elastic cantilever beam

This example considers an elastic cantilever beam under dynamic loading and is focused on examining the capabilities of the ghost stabilisation technique compared to other methods available in the literature designed to mitigate small nodal masses and/or nodal force contributions. The elastic cantilever beam was subjected to a point load at its free end and modelled using the GIMPM. The beam was $l_0 = 10$ m long and $d_0 = 1$ m deep and 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 end point 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 $h = 0.5$ m. The beam was positioned such that the outer boundary of the beam aligned with element boundaries, the beam's left hand edge coincided with the left hand edge of the background grid and the beam's neutral axis was centralised vertically within the background grid. The initial discretisation of the beam is shown in Figure 16 with $h = 0.5$ m and with 2^2 material points per initially populated background grid cells. Note that the figure uses 2^2 material points for clarity of the illustration, for all of the analyses 4^2 generalised interpolation material points per background grid cell was used. The loaded material points are shown by the black-filled circles.

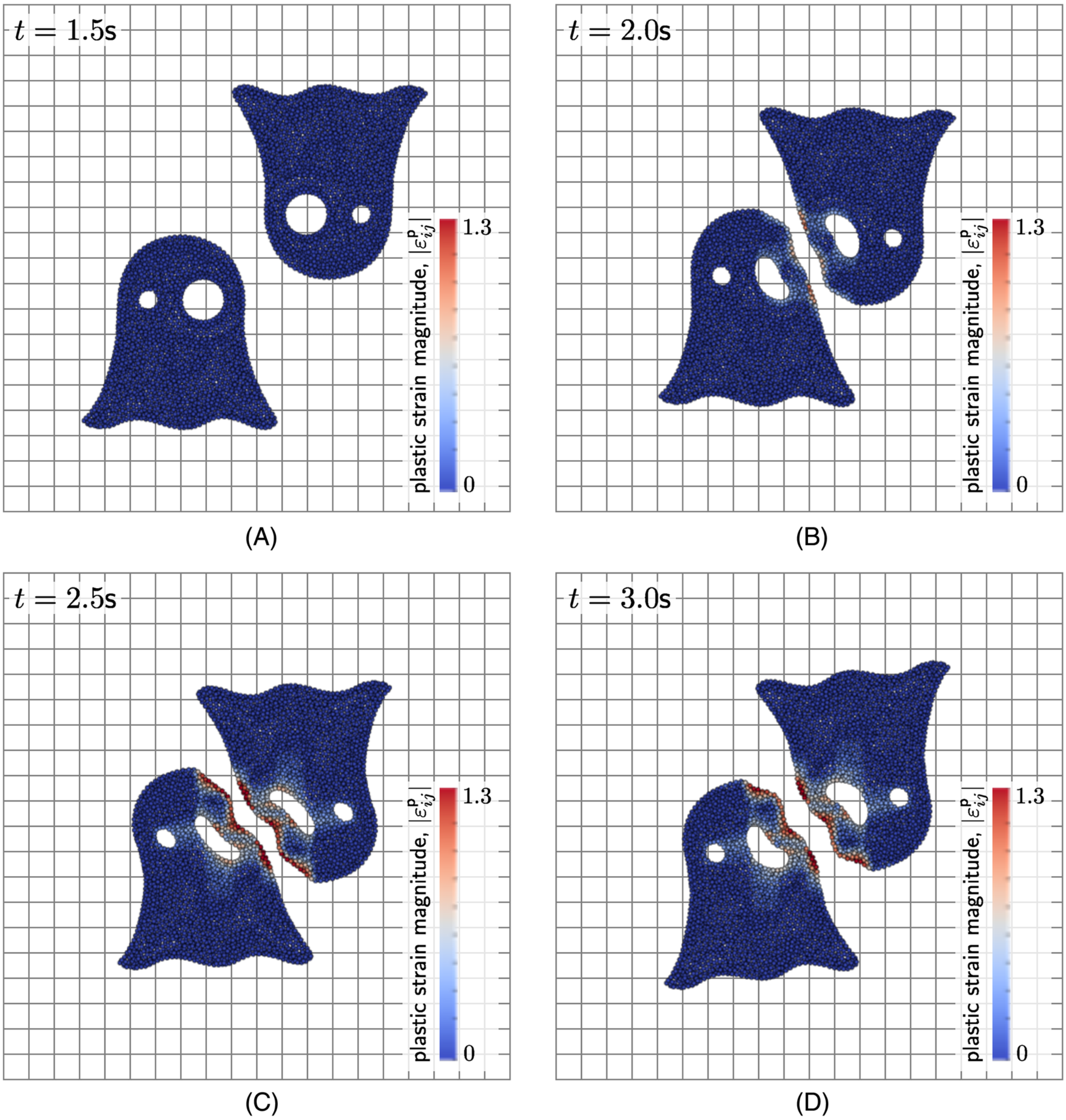

FIGURE 15 Collision of elasto-plastic ghosts: Deformed material point positions coloured according to the magnitude of the total plastic logarithmic strains using a *medium* material point discretisation, a background mesh size of $h = 0.05$ m and 1000 time steps. (A) $t = 1.5$ s. (B) $t = 2.0$ s. (C) $t = 2.5$ s. (D) $t = 3.0$ s.

Four methods were investigated:

1. USL GIMPM with a ghost-stabilised consistent mass matrix $(\gamma_M = \frac{1}{4}\rho)$;
2. MUSL[18] GIMPM with a lumped mass matrix; and
3. USF GIMPM with a lumped mass matrix and the *distribution coefficient algorithm for small mass nodes* of Ma et al.[13] with a threshold of 4% of the maximum mass value to trigger residual force redistribution[¶¶¶].
4. USF GIMPM with a lumped mass matrix.

Figure 17 provides the normalised horizontal and vertical displacement response of the tip of the cantilever by averaging the displacement of the loaded material points (the black filled circles in Figure 16) for all of the considered approaches. The result of the lumped USF, the lumped MUSL and the ghost stabilised USL approaches are very similar in terms of their global time-displacement response but the approach of of Ma et al.[13] failed at time step 4386 (at a time of 2.56 s, shown by the A points in Figure 17), with the material points *exploding* and exiting the background grid. If the

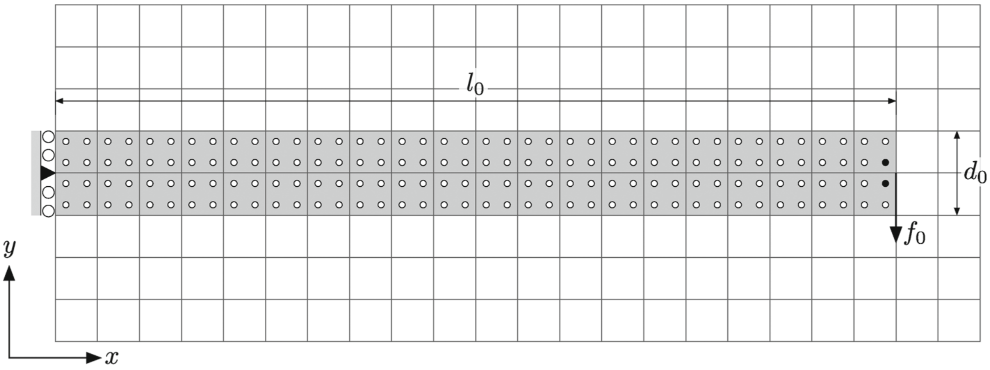

FIGURE 16 Elastic beam: Initial discretisation and boundary conditions with $h = 0.5$ m (only part of the background mesh is shown).

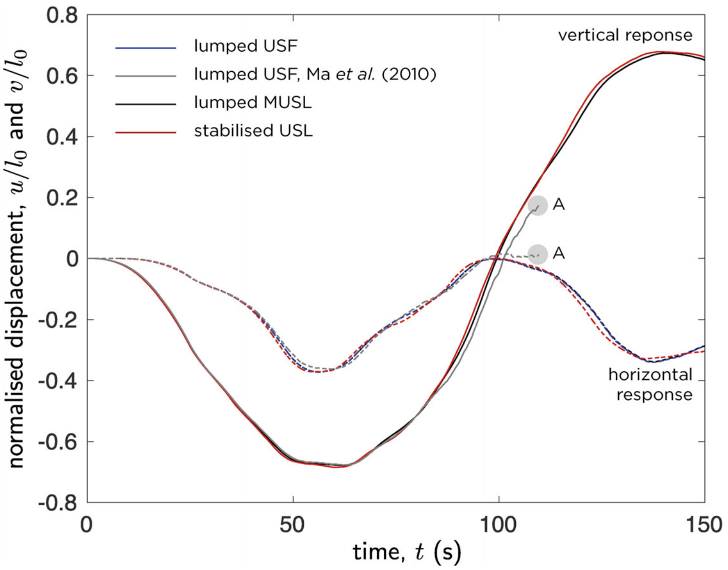

FIGURE 17 Elastic beam: Mean normalised displacement response of the loaded material points.

threshold on the percentage the maximum mass value to trigger residual force redistribution was dropped to 1% then the analysis was able to complete, giving a near identical result to the lumped USF simulation, which is not surprising given that the lumped USF is stable for this analysis.

The horizontal stress σ_{xx} distribution at $t = 150$ s for the ghost stabilised USL and MUSL approaches are shown in Figure 18. The overall stress distributions are similar but it is clear that the ghost stabilisation reduces the stress oscillations seen near the boundary of the beam. This is particularly evident in the region highlighted by the A-labelled dashed circle but can also be seen in other regions of the beam.

The examples explored so far have all focused on explicit dynamic analysis. The following sections will demonstrate the benefits of ghost stabilisation for implicit *quasi-static* analysis.

5.5 | Implicit *quasi*-static: Elastic compression under self weight

This example considers the one dimensional compression of an elastic column with an initial height of $l_0 = 50$ m under its own self weight and aims to demonstrate that including ghost stabilisation has minimal impact on the convergence of the method using a problem with an analytical solution. The material has a Young's modulus of 10 kPa and a Poisson's ratio of zero. The background mesh is comprised of square background elements with roller boundary conditions on the base and sides and the column is discretised by a 2 × 2 grid of equally spaced material points in each initially populated

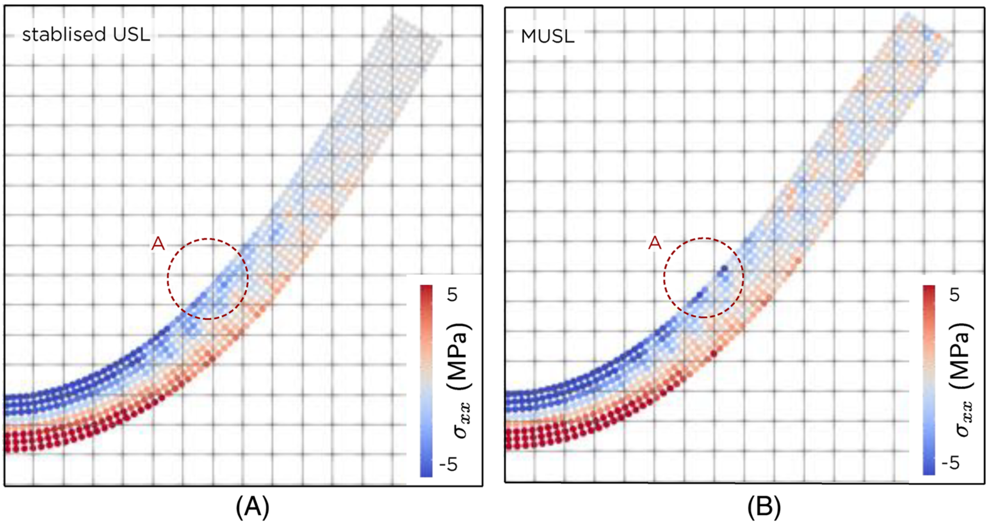

FIGURE 18 Elastic beam: Horizontal stress σ_{xx} distribution at $t = 150$ s for the (A) ghost-stabilised USL and (B) MUSL approaches.

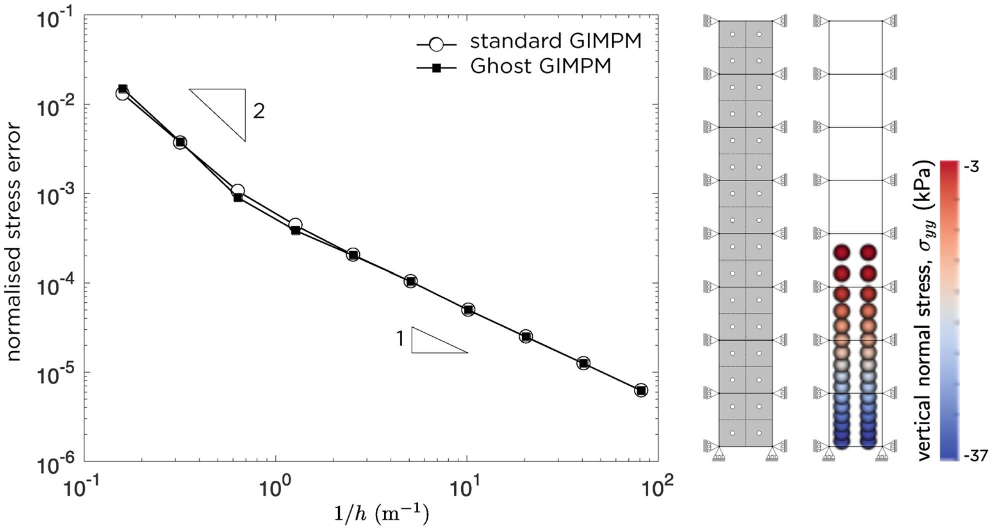

FIGURE 19 Compression under self weight: Normalised stress error convergence with grid refinement and initial problem discretisation and final material point positions coloured according to σ_{yy} with $h = 6.25$ m.

background grid element (as shown to the right of Figure 19 for $h = 6.25$ m). A body force of 800 N/m^2 ($g = 10$ m/s^2 and an initial density of $\rho_0 = 80$ kg/m^3) is applied over 40 equal load steps. The magnitude of the load causes the column to compress to approximately half of its initial height.

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

$$\sigma^a_{yy} = \rho_0 g(l_0 - Y), \tag{29}$$

where Y is the original position of the point in the body and l_0 is the original height of the column. Figure 19 shows the convergence of the GIMPM with background mesh refinement whilst maintaining 4 material points per initially populated element. The reported normalised stress error is

$$\text{error} = \sum_{p=1}^{n_p} \frac{|\sigma^p_{yy} - \sigma^a_{yy}(Y_p)|\ V^0_p}{(g\rho_0 l_0)V_0}, \tag{30}$$

TABLE 5 Compression under self weight: Stress error variation with γ_k.

	Normalised penalty parameter, γ_k/E				
h (m)	1^{-6}	1^{-3}	1^{0}	1^{3}	1^{6}
0.098	4.938×10^{-5}	4.938×10^{-5}	4.938×10^{-5}	4.926×10^{-5}	4.962×10^{-5}
1.563	1.049×10^{-3}	1.042×10^{-3}	8.912×10^{-4}	1.208×10^{-3}	1.210×10^{-3}

where $V_0 = \sum V_p^0$ is the initial volume of the column and σ_{yy}^p is the vertical stress at each of the material points. Figure 19 provides the errors with and without ghost stabilisation, where γ_k was set to 10 kPa for the stabilised analyses. The GIMPM with and without stabilisation converge towards the analytical solution at a rate between 1 and 2, which is consistent with the underlying basis of the method and there is very little difference between the error values for different background mesh resolutions. The difference in error also reduces with grid refinement due to the region influenced by the stabilisation reducing as the grid is refined. This point is reinforced by Table 5, which gives the normalised stress error for $h = 50/2^9 \approx 0.098$ m and $h = 50/2^5 \approx 0.163$ m with different stabilisation parameter values. It is clear from the table that reducing the background element size reduces the sensitivity of the result to the value of the penalty parameter for the reason explained above. For the analysis with a mesh size of 0.098 m, twelve orders of magnitude variation in the penalty parameter only changes the normalised stress error by 0.49%, and the variation in stress error is 15% for the $h = 1.563$ m analyses. Although a variation of 15% in the normalised stress error seems significant, it is only a very small variation in the actual error, which is magnified by normalising by the error associated with the non-stabilised analysis.

5.6 | Implicit *quasi*-static: Elastic beam

The final example in this article presents the implicit *quasi*-static analysis of a large deformation beam as a challenging problem for the MPM due numerous small overlaps between the material points and the background grid. The elastic cantilever beam was subjected to a point load at its free end and modelled using the GIMPM. 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 Figure 16 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. The stiffness stabilisation parameter was taken to be the same as the Young's modulus of the material, $\gamma_K = 12$ MPa.

Tables 6 and 7 provide information on the stability of the *quasi*-static analysis with different numbers of generalised interpolation and standard material points per element with and without ghost stabilisation. The tables report different information depending on if an analysis was able to complete all load steps:

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

The normalised convergence tolerance[###] was set to 1×10^{-6} for all analyses. All of the ghost-stabilised analyses were able to complete all of the load steps, however several of the standard GIMPM and MPM analyses failed to converge (or reached the maximum number of Newton iterations, set to 10 for all analyses) before the full load had been applied. For the standard GIMPM there is a general trend of increasing stability with increasing numbers of material points but this is not always the case, for example the $h = 0.5$ m, 6^2 material point failed at load step 10 whereas the 5^2 material point analysis was able to apply the full load for the same background mesh size. The is no clear trend for the sMPM results shown in Table 7. In essence a user does not know if the analysis will converge or not before running a simulation when using the standard GIMPM or MPM, even if *similar* analyses have converged. Reducing the mesh size also increases the average total number of Newton–Raphson iterations required to complete the analysis for the standard (non-stabilised) GIMPM whereas the total and maximum number of iterations for the stabilised GIMPM is relatively insensitive to mesh size variations. In fact, there is trend of reducing total numbers of iterations as the mesh is refined. Although the mean total number of iterations for the stabilised MPM analyses increases with mesh refinement, the total number of iterations

TABLE 6 Elastic beam (GIMP): 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.

	Standard GIMPM			**Ghost stabilised GIMPM**		
MPs/elem.	$\boldsymbol{h} = 0.500$	$\boldsymbol{h} = 0.250$	$\boldsymbol{h} = 0.125$	$\boldsymbol{h} = 0.500$	$\boldsymbol{h} = 0.250$	$\boldsymbol{h} = 0.125$
2^2	✗ 5	✗ 11	✗ 1	✓ 205 (5)	✓ 203 (5)	✓ 205 (5)
3^2	✓ 215 (9)	✓ 215 (8)	✗ 5	✓ 204 (5)	✓ 204 (5)	✓ 203 (5)
4^2	✗ 40	✗ 26	✓ 215 (5)	✓ 204 (5)	✓ 204 (5)	✓ 201 (5)
5^2	✓ 207 (6)	✓ 207 (5)	✓ 214 (5)	✓ 204 (5)	✓ 203 (5)	✓ 202 (5)
6^2	✗ 9	✓ 207 (5)	✓ 216 (6)	✓ 204 (5)	✓ 204 (5)	✓ 202 (5)
Mean	211 (7.5)	209.7 (6)	215 (5.3)	204.2 (5)	203.6 (5)	202.6 (5)

TABLE 7 Elastic beam (MPM): 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.

	Standard MPM			**Ghost stabilised MPM**		
MPs/elem.	$\boldsymbol{h} = 0.500$	$\boldsymbol{h} = 0.250$	$\boldsymbol{h} = 0.125$	$\boldsymbol{h} = 0.500$	$\boldsymbol{h} = 0.250$	$\boldsymbol{h} = 0.125$
2^2	✓ 216 (6)	✗ 14	✗ 13	✓ 214 (5)	✓ 218 (5)	✓ 224 (6)
3^2	✗ 40	✗ 37	✓ 224 (9)	✓ 211 (5)	✓ 216 (5)	✓ 218 (5)
4^2	✓ 211 (5)	✓ 213 (5)	✓ 222 (6)	✓ 206 (5)	✓ 213 (5)	✓ 218 (5)
5^2	✓ 208 (6)	✓ 215 (5)	✓ 218 (5)	✓ 202 (5)	✓ 210 (5)	✓ 217 (5)
6^2	✓ 211 (6)	✗ 44	✗ 5	✓ 202 (5)	✓ 206 (5)	✓ 214 (5)
Mean	211.5 (5.8)	214 (5)	221.3 (6.7)	207 (5)	212.7 (5)	218.2 (5.2)

reduces with increasing numbers of material points and the maximum number of iterations within a step is 5 for all but one of the 15 analyses.

Figure 20 shows the normalised force-displacement response of the beam for the ghost stabilised GIMPM with different background mesh sizes and 3^2 material points per initially populated element. The squares and circles show the analytical solution of Molstad[66] for the horizontal and vertical displacements, respectively, and the ghost GIMPM results are shown by the black lines. All of the ghost GIMPM results are in good agreement with the analytical solution. The simulations do over estimate the vertical displacement of the beam compared to the analytical solution but this is consistent with other numerical results in the literature[29,30,39] and is linked to the analytical assumption that the beam does not change in length. The red line in Figure 20 shows the non-stabilised GIMPM result with $h = 0.5$ m, which deviates from the expected (analytical) response for the horizontal displacement, showing under-stiff behaviour.

The vertical normal stress, σ_{yy}, distributions at the end of the analysis for the standard, non-stabilised (left) and ghost stabilised (right) GIMPMs with $h = 0.25$ m and 3^2 material point per initially populated background grid cell are shown in Figure 21. The key difference in the stress distributions between the two methods is on the top and bottom surfaces of the beams, where the standard GIMPM predicts spurious oscillations in the stress field. This variation in stress is particularly evident around region A within the top inset figure, where a drop in the tensile stress on the top surface of the beam is due to the poor distribution of material points within the background grid at the edge of the physical domain. This poor distribution of material relative to the background grid results in nodal degrees of freedom with very little stiffness and therefore their displacements can deform in a way that is not consistent with the deformation, and therefore stress field, of the well-conditioned part of the linear system of equations. These stress oscillations are removed by the ghost stabilisation due to the penalisation of variation in the gradient of the solution field over the boundary element edges.

5.7 | Observations

This section has presented five numerical investigations for dynamic and *quasi*-static large deformation stress analysis problems. The following key observations can be drawn from these numerical examples:

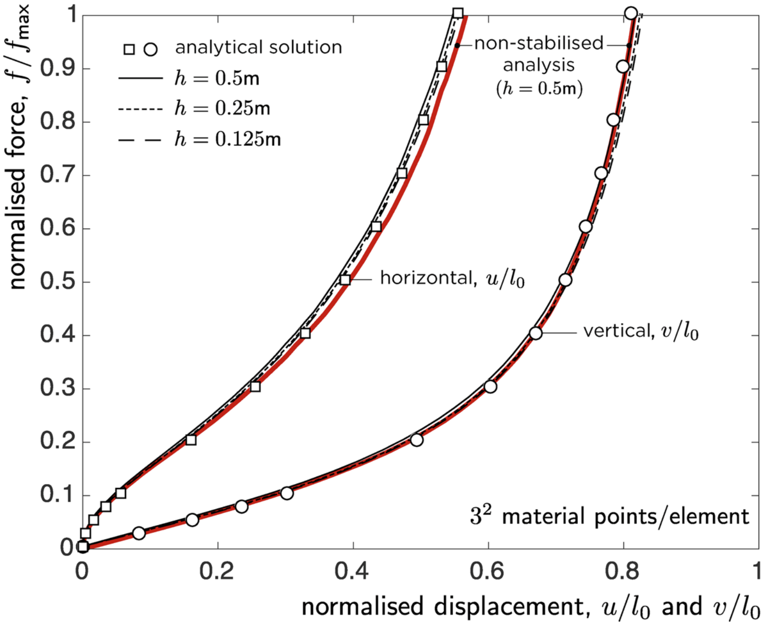

FIGURE 20 Elastic beam: Normalised force versus displacement (where the squares and circles show the analytical solution of Molstad[66] for the horizontal and vertical displacements, respectively) with 3^2 material points per initially populated element. The red line shows the non-stabilised result with $h = 0.5$ m.

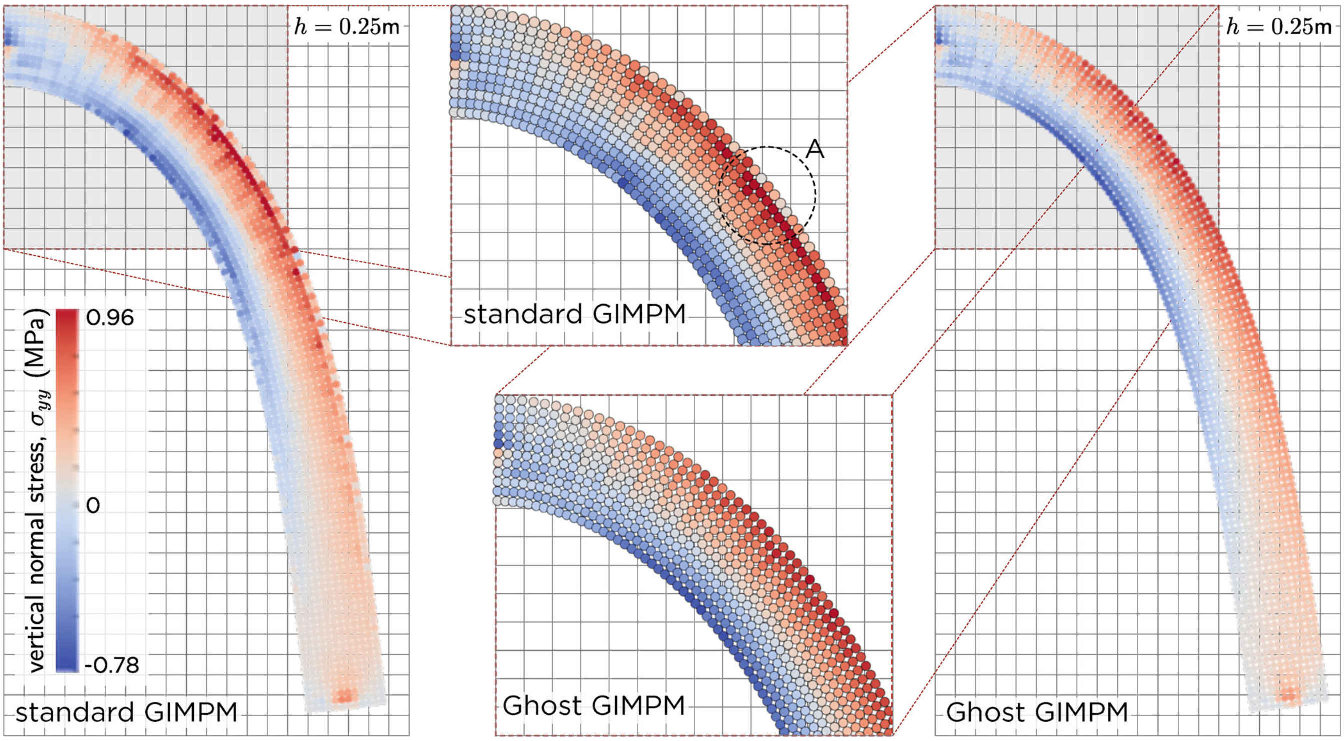

FIGURE 21 Elastic beam: Vertical normal stress, σ_{yy}, distribution at the end of the analysis for the non-stabilised and ghost stabilised GIMPMs with $h = 0.25$ m and 3^2 material point per initially populated background grid cell.

(i) for explicit dynamics the ghost stabilisation technique:
  (a) opens the door to the consistent mass matrix being used for practical simulations with arbitrary positioning of the physical domain (material points) on the background grid;
  (b) unlike the use of the lumped mass matrix, adopting the stabilised consistent mass matrix maintains the material point velocity field for rigid body motions and linear deformation fields; and
  (c) corrects the excessive dissipation seen when combining USL approaches with a lumped mass matrix, due to consistent mapping of the nodal velocity field to the material points when determining changes in the deformation field.

(ii) for *quasi*-static implicit analysis the ghost stabilisation technique:
  (a) maintains the ability of the underlying numerical algorithm to converge towards problems with analytical solutions as the stabilisation only influences a local region of the physical body near the problem domain boundary;
  (b) removes the stability uncertainty when modelling large deformation problems—without stabilisation it is difficult to say if a analysis will converge or not even if similar analyses are stable;
  (c) permits the use of lower numbers of material points whilst maintaining stability of the overall method; and
  (d) significantly reduces the stress oscillations seen near the boundary of the physical domain, which are due to small overlaps rather than cell crossing errors in non-stabilised methods.

6 | CONCLUSION

This article has proposed a new stabilisation technique for explicit dynamic and implicit *quasi*-static MPMs that resolves the vast majority of the stability issues encountered by the method, whilst not damaging their underlying properties. The technique introduces a bound on the condition number of the linear system of equations being solved independent of the position of the physical body being modelled at the expense of additional computations over the boundary of the body and the introduction of a non-physical penalty parameter. However, unlike other methods available in the literature, the stabilisation technique proposed here is relatively insensitive to the penalty parameter, which can be linked to the density or stiffness of the bulk material being modelled. The method has the additional benefit of reducing stress oscillations at the boundary of the body due to small interactions between the material points and the background grid for both explicit and implicit analysis. For implicit analyses the method permits the use of lower numbers of material points whilst maintaining a stable simulation. For explicit analysis it means that the consistent mass matrix can be adopted with USF, USL, or MUSL algorithms without excessive energy dissipation, albeit with increased computational cost compared to using a lumped mass matrix. Although this article has focused on explicit dynamic and implicit *quasi-static* analyses, the approach can be applied to all MPM variants, including implicit dynamics, without requiring an explicit description of the boundary of the problem. The technique offered in this article is one step on the road to the MPM becoming a usable tool for practical engineering analyses.

ACKNOWLEDGMENTS

The author 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, Charles Augarde, Robert Bird, Nathan Gavin, Ted O'Hare, and Giuliano Pretti. This work was supported by the Engineering and Physical Sciences Research Council (Grant numbers EP/W000970/1, EP/R004900/1, and EP/N006054/1).

DATA AVAILABILITY STATEMENT

All data created during this research are openly available at https://doi.org/10.15128/r1x920fw91n. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any author accepted manuscript version arising.

ENDNOTES

*Note, there are several reasons why MPM implementations adopt a lumped mass matrix, such as computational efficiency as the inversion of a lumped (diagonal) matrix is trivial compared to inversion of the consistent mass matrix, even if it is well posed.

[†] It is worth mentioning that some of the issues identified by Ma et al.[13] are linked to the used of linear shape functions as identified by Steffen et al.[16] It should be noted that the analysis of Steffen et al.[16] was restricted to one-dimension, which removes many of the practical issues faced in general two and three-dimensional analysis.

[‡] In the MPM literature this stage is often referred to as the *convection* step, or that the material points are *convected* to their new positions. *Convected* is an erroneous choice of word to describe this stage of the analysis as it has nothing to do with convection (especially in solid materials), it is simply saying that the material point positions are updated based on their displacements within the current time/load step.

[§] AMPLE adopts the same formulation as implemented in the generalised interpolation approach of Charlton et al.[30]

[¶] $[\nabla_x S_{vp}]$ is essentially the same as the conventional strain-displacement $[B]$ matrix found in finite element literature and $[S_{vp}]$ is the equivalent of the shape function matrix, often denoted $[N]$.

[#] In this case, the assembly operator, $\mathcal{A}$, is simply stating to perform this calculation at for each material point and sum their contributions to the background grid nodes.

[‖] FLIP[41] increments the material point velocity field using the change in the vertex velocities over the time step, rather than using the total vertex velocities to overwrite the material point velocity field as with a Particle In Cell (PIC)[42] update.

[**] The analysis presented in this section raises a question regarding what is *reasonable* in terms of the condition number of a linear system of equations. The answer to this question is related to: the precision of the computational framework being used and the method being used to solve the system of equations. In all cases, a condition number approaching the reciprocal of the precision of the machine being used will cause accuracy problems.

[††] The term *reduced* is referring to a matrix where the rows and columns associated with constrained degrees of freedom have been removed.

[‡‡] Adopting the lumped mass matrix removes the need to invert the consistent mass matrix in explicit time stepping approaches, replacing it with the inversion of a diagonal matrix, which is trivial. However, the use of a lumped mass matrix introduces additional numerical dissipation.[21,23] It also does not remove the possibility of spuriously large acceleration values due to very small mass matrix entries.

[§§] The definition of the positive element, K^+, and the negative element, K^-, associated with a face is arbitrary and swapping the the positive and negative elements will not change of the nature of the stabilisation. Simply one of the the elements connected to the face is labelled as *positive* and the other as *negative*.

[¶¶] Note that the dark grey shaded elements and the boundary element edges are different in Figures 4C and 5 as Figure 4C describes elements *cut* by the boundary rather than the MPM approach described in this section.

[##] The value of this penalty parameter will be discussed later in the section.

[‖‖] The term *skeleton*, often used in discontinuous Galerkin finite element methods, refers to the boundaries between elements.

[***] Determining the active/inactive elements is required for MPMs without stabilisation.

[†††] The positions, volume, mass and so forth of all of the material points for all of the analysis presented in the article are provided as VTK files within the supplementary data associated with the article. See https://doi.org/10.15128/r1x920fw91n for details.

[‡‡‡] The PIC approach[42] uses the total nodal velocities to overwrite the material point velocity field, assuming that the velocity through the domain varies according to the vertex values and their associated basis functions.

[§§§] These properties are similar to those used by Pretti et al.[25] in order to focus on the bulk response of the beam rather than high frequency stress oscillations.

[¶¶¶] The distribution coefficient algorithm for small mass nodes of Ma et al.[13] redistributes the residual force associated with nodes with *small* mass to surrounding nodes in order to avoid spurious large acceleration predictions. The force is distributed to the surrounding nodes proportional to the square root of the nodal mass. The 4% threshold is arbitrary and has been selected based on the value used in Ma et al.[13] See Ma et al.[13] for details.

[###] The convergence criteria used in the implicit *quasi*-static MPM implementation adopted in this article is the same as that used by,[29] where the residual out of balance force (the difference between the internal force and the external actions), normalised by the magnitude of the external actions is checked until it converges below a given tolerance. The only change for this article is that the residual force equation now includes a contribution from the ghost stabilisation, as explained in Section 3.4.

ORCID

William M. Coombs https://orcid.org/0000-0003-2099-1676

REFERENCES

1. Sulsky D, Chen Z, Schreyer HL. A particle method for history-dependent materials. *Comput Methods Appl Mech Eng*. 1994;118(1):179-196.
2. de Vaucorbeil A, Nguyen VP, Sinaie S, Wu JY. Material point method after 25 years: theory, implementation, and applications. In: Bordas SPA, Balint DS, eds. *Advances in Applied Mechanics*. Vol 53. Elsevier; 2020:185-398.
3. Solowski WT, Berzins M, Coombs WM, et al. Material point method: overview and challenges ahead. In: Bordas SPA, Balint DS, eds. *Advances in Applied Mechanics*. Vol 54. Elsevier; 2021:113-204.
4. Liu K, Wang Y, Huang M, Yuan WH. Postfailure analysis of slopes by random generalized interpolation material point method. *Int J Geomech*. 2021;21(3):04021015.
5. Sticko S, Ludvigsson G, Kreiss G. High-order cut finite elements for the elastic wave equation. *Adv Comput Math*. 2020;46(45):1-28.
6. Hansbo P, Larson M, Larsson K. Cut finite element methods for linear elasticity problems. In: Bordas S, Burman E, Larson M, Olshanskii M, eds. *Geometrically Unfitted Finite Element Methods and Applications*. Lecture Notes in Computational Science and Engineering. Springer; 2017:25-63.

7. Burman E, Elfverson D, Hansbo P, Larson M, Larsson K. Shape optimization using the cut finite element method. *Comput Methods Appl Mech Eng*. 2018;328:242-261.
8. Terada K, Asai M, Yamagishi M. Finite cover method for linear and non-linear analyses of heterogeneous solids. *Int J Numer Methods Eng*. 2003;58(9):1321-1346.
9. Parvizian J, Düster A, Rank E. Finite cell method. *Comput Mech*. 2007;41:121-133.
10. Düster A, Parvizian J, Yang Z, Rank E. The finite cell method for three-dimensional problems of solid mechanics. *Comput Methods Appl Mech Eng*. 2008;197(45):3768-3782.
11. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *Int J Numer Methods Eng*. 1999;46(1):131-150.
12. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *Int J Numer Methods Eng*. 1999;45(5):601-620.
13. Ma X, Giguere PT, Jayaraman B, Zhang DZ. Distribution coefficient algorithm for small mass nodes in material point method. *J Comput Phys*. 2010;229(20):7819-7833.
14. Bardenhagen S, Kober E. The generalized interpolation material point method. *Comput Model Eng Sci*. 2004;5(6):477-496.
15. Gan Y, Sun Z, Chen Z, Zhang X, Liu Y. Enhancement of the material point method using B-spline basis functions. *Int J Numer Methods Eng*. 2018;113(3):411-431.
16. Steffen M, Kirby RM, Berzins M. Analysis and reduction of quadrature errors in the material point method. *Int J Numer Methods Eng*. 2008;76(6):922-948.
17. Yamaguchi Y, Moriguchi S, Terada K. Extended B-spline-based implicit material point method. *Int J Numer Methods Eng*. 2021;122(7):1746-1769.
18. Sulsky D, Zhou SJ, Schreyer HL. Application of a particle-in-cell method to solid mechanics. *Comput Phys Commun*. 1995;87(1):236-252.
19. Nairn JA. Material point method calculations with explicit cracks. *Comput Model Eng Sci*. 2003;4(6):649-663.
20. de Vaucorbeil A, Nguyen VP, Hutchinson CR. A total-Lagrangian material point method for solid mechanics problems involving large deformations. *Comput Methods Appl Mech Eng*. 2020;360:112783.
21. Burgess D, Sulsky D, Brackbill J. Mass matrix formulation of the FLIP particle-in-cell method. *J Comput Phys*. 1992;103(1):1-15.
22. Bardenhagen S. Energy conservation error in the material point method for solid mechanics. *J Comput Phys*. 2002;180(1):383-403.
23. Love E, Sulsky DL. An energy-consistent material-point method for dynamic finite deformation plasticity. *Int J Numer Methods Eng*. 2006;65(10):1608-1638.
24. Nairn JA, Hammerquist CC. Material point method simulations using an approximate full mass matrix inverse. *Comput Methods Appl Mech Eng*. 2021;377:113667.
25. Pretti G, Coombs WM, Augarde CE, Sims B, Puigvert MM, Gutiérrez JAR. A conservation law consistent updated Lagrangian material point method for dynamic analysis. *J Comput Phys*. 2023;485:112075.
26. Hammerquist CC, Nairn JA. A new method for material point method particle updates that reduces noise and enhances stability. *Comput Methods Appl Mech Eng*. 2017;318:724-738.
27. Wang B, Vardon PJ, Hicks MA, Chen Z. Development of an implicit material point method for geotechnical applications. *Comput Geotech*. 2016;71:159-167.
28. Burman E. Ghost penalty. *C R Math*. 2010;348(21):1217-1220.
29. Coombs WM, Augarde CE. AMPLE: a material point learning environment. *Adv Eng Softw*. 2020;139:102748.
30. Charlton TJ, Coombs WM, Augarde CE. iGIMP: an implicit generalised interpolation material point method for large deformations. *Comput Struct*. 2017;190:108-125.
31. Bing Y, Cortis M, Charlton T, Coombs W, Augarde C. B-spline based boundary conditions in the material point method. *Comput Struct*. 2019;212:257-274.
32. Remmerswaal G. *Development and Implementation of Moving Boundary Conditions in the Material Point Method*. Master's Thesis. TU Delft; 2017.
33. Chandra B, Singer V, Teschemacher T, Wücheer R, Larese A. Nonconforming Dirichlet boundary conditions in implicit material point method by means of penalty augmentation. *Acta Geotech*. 2021;16:2315-2335.
34. Cortis M, Coombs WM, Augarde CE, Brown MJ, Brennan A, Robinson S. Imposition of essential boundary conditions in the material point method. *Int J Numer Methods Eng*. 2018;113(1):130-152.
35. Simo J. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Comput Methods Appl Mech Eng*. 1992;99:61-112.
36. de Souza Neto EA, Peric D, Owen DRJ. *Computational Methods for Plasticity: Theory and Applications*. John Wiley & Sons, Ltd; 2008.
37. Coombs WM, Charlton TJ, Cortis M, Augarde CE. Overcoming volumetric locking in material point methods. *Comput Methods Appl Mech Eng*. 2018;333:1-21.
38. Wang L, Coombs W, Augarde C, et al. On the use of domain-based material point methods for problems involving large distortion. *Comput Methods Appl Mech Eng*. 2019;355:1003-1025.
39. Coombs WM, Augarde CE, Brennan AJ, et al. On Lagrangian mechanics and the implicit material point method for large deformation elasto-plasticity. *Comput Methods Appl Mech Eng*. 2020;358:112622.
40. Wang L, Coombs WM, Augarde CE, et al. An efficient and locking-free material point method for three-dimensional analysis with simplex elements. *Int J Numer Methods Eng*. 2021;122(15):3876-3899.
41. Brackbill J, Ruppel H. FLIP: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J Comput Phys*. 1986;65(2):314-343.

42. Harlow F. The particle-in-cell computing method for fluid dynamics. *Methods Comput Phys*. 1964;3:319-343.
43. Berzins M. Energy conservation and accuracy of some MPM formulations. *Comput Part Mech*. 2022;9:1205-1217.
44. Sadeghirad A, Brannon RM, Burghardt J. A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. *Int J Numer Methods Eng*. 2011;86(12):1435-1456.
45. Sadeghirad A, Brannon R, Guilkey J. Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces. *Int J Numer Methods Eng*. 2013;95(11):928-952.
46. Steffen M, Wallstedt PC, Guilkey JE, Kirby RM, Berzins M. Examination and analysis of implementation choices within the material point method. *Comput Model Eng Sci*. 2008;31:107-127.
47. Andersen S, Andersen L. Analysis of spatial interpolation in the material-point method. *Comput Struct*. 2010;88(7):506-518.
48. Wang B, Vardon P, Hicks M. Investigation of retrogressive and progressive slope failure mechanisms using the material point method. *Comput Geotech*. 2016;78:88-98.
49. Sticko S, Kreiss G. Higher order cut finite elements for the wave equation. *J Sci Comput*. 2019;80:1867-1887.
50. Liu Z, Zhang Y, Jiang Y, Yang H, Yang Y. Unfitted finite element method for fully coupled poroelasticity with stabilization. *Comput Methods Appl Mech Eng*. 2022;397:115132.
51. Badia S, Neiva E, Verdugo F. Linking ghost penalty and aggregated unfitted methods. *Comput Methods Appl Mech Eng*. 2022;388: 114232.
52. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Appl Numer Math*. 2012;62(4):328-341.
53. Burman E, Claus S, Hansbo P, Larson MG, Massing A. CutFEM: discretizing geometry and partial differential equations. *Int J Numer Methods Eng*. 2015;104(7):472-501.
54. Divi SC, Zuijlen vPH, Hoang T, et al. Residual-based error estimation and adaptivity for stabilized immersed isogeometric analysis using truncated hierarchical B-splines. *J Mech*. 2022;38:204-237.
55. Noël L, Schmidt M, Doble K, Evans JA, Maute K. XIGA: an eXtended IsoGeometric analysis approach for multi-material problems. *Comput Mech*. 2022;70:1281-1308.
56. Kadapa C, Dettmer W, Perić D. A stabilised immersed boundary method on hierarchical b-spline grids for fluid–rigid body interaction with solid–solid contact. *Comput Methods Appl Mech Eng*. 2017;318:242-269.
57. Coombs W. *Finite Deformation of Particulate Geomaterials: Frictional and Anisotropic Critical State Elasto-Plasticity*. PhD Thesis. Durham University; 2011.
58. González Acosta JL, Vardon PJ, Hicks MA. Development of an implicit contact technique for the material point method. *Comput Geotech*. 2021;130:103859.
59. Bardenhagen SG, Guilkey JE, Roessig KM, Brackbill JU, Witzel WM, Foster JC. An improved contact algorithm for the material point method and application to stress propagation in granular material. *Comput Model Eng Sci*. 2001;2(4):509-522.
60. Ma J, Wang D, Randolph M. A new contact algorithm in the material point method for geotechnical simulations. *Int J Numer Anal Methods Geomech*. 2014;38(11):1197-1210.
61. Chen ZP, Zhang X, Qiu XM, Liu Y. A frictional contact algorithm for implicit material point method. *Comput Methods Appl Mech Eng*. 2017;321:124-144.
62. Huang P, Zhang X, Ma S, Huang X. Contact algorithms for the material point method in impact and penetration simulation. *Int J Numer Methods Eng*. 2011;85(4):498-517.
63. Bardenhagen S, Brackbill J, Sulsky D. The material-point method for granular materials. *Comput Methods Appl Mech Eng*. 2000;187(3):529-541.
64. Homel MA, Herbold EB. Field-gradient partitioning for fracture and frictional contact in the material point method. *Int J Numer Methods Eng*. 2017;109(7):1013-1044.
65. Gao L, Guo N, Yang Z, Jardine R. MPM modeling of pile installation in sand: contact improvement and quantitative analysis. *Comput Geotech*. 2022;151:104943.
66. Molstad T. *Finite Deformation Analysis Using the Finite Element Method*. PhD Thesis. University of British Columbia; 1977.

How to cite this article: Coombs WM. Ghost stabilisation of the material point method for stable quasi-static and dynamic analysis of large deformation problems. *Int J Numer Methods Eng*. 2023;1-35. doi: 10.1002/nme.7332

APPENDIX A. GHOST STABILISATION MATRIX FORM

This appendix details the transitioning between the bi-linear form of the ghost stabilisation term, (20), and the matrix expression required for numerical implementation of the method, (22). Starting with the gradient of the positive displacement field projected normal to the boundary (i.e., the directional derivative of the displacement field normal to the positive element boundary) and writing it in terms of matrices/vectors in 2D gives

$$\frac{\partial u_i^+}{\partial x_j}n_j = n_x\left\{\begin{matrix}\frac{\partial u_x^+}{\partial x}\\ \frac{\partial u_y^+}{\partial x}\end{matrix}\right\} + n_y\left\{\begin{matrix}\frac{\partial u_x^+}{\partial y}\\ \frac{\partial u_y^+}{\partial y}\end{matrix}\right\}, \tag{A1}$$

which can also be expressed as

$$\frac{\partial u_i^+}{\partial x_j}n_j = \begin{bmatrix} n_x & 0 & 0 & n_y\\ 0 & n_y & n_x & 0\end{bmatrix}\left\{\begin{matrix}\frac{\partial u_x}{\partial x}\\ \frac{\partial u_y}{\partial y}\\ \frac{\partial u_y}{\partial x}\\ \frac{\partial u_x}{\partial y}\end{matrix}\right\}. \tag{A2}$$

Introducing the finite element approximation for the displacement field

$$\{u^+\} = [N^+]\{d^+\} = \begin{bmatrix} N_1^+ & 0 & \dots & N_n^+ & 0\\ 0 & N_1^+ & \dots & 0 & N_n^+\end{bmatrix}\left\{\begin{matrix}(u_x^+)_1\\ (u_y^+)_1\\ \vdots\\ (u_x^+)_n\\ (u_y^+)_n\end{matrix}\right\}, \tag{A3}$$

where $[N]$ is the matrix of shape functions and $\{d\}$ a vector containing the nodal displacement components, where n is the number of nodes associated with the element. Combing (A3) with (A2) gives

$$\frac{\partial u_i^+}{\partial x_j}n_j = \begin{bmatrix} n_x & 0 & 0 & n_y\\ 0 & n_y & n_x & 0\end{bmatrix}\begin{bmatrix} N_1^+,_x & 0 & \dots & N_n^+,_x & 0\\ 0 & N_1^+,_y & \dots & 0 & N_n^+,_y\\ 0 & N_1^+,_x & \dots & 0 & N_n^+,_x\\ N_1^+,_y & 0 & \dots & N_n^+,_y & 0\end{bmatrix}\left\{\begin{matrix}(u_x^+)_1\\ (u_y^+)_1\\ \vdots\\ (u_x^+)_n\\ (u_y^+)_n\end{matrix}\right\}, \tag{A4}$$

or compactly

$$\frac{\partial u_i^+}{\partial x_j}n_j = [n]^T[G^+]\{d^+\}. \tag{A5}$$

Similar equations can be used to express the gradient of displacement associated with the negative element projected in normal to the boundary under consideration. The next step is to realise that the order of the contribution of the test and trial function in (20) is arbitrary as the resultant of the expression is a scalar, therefore (20) can be written as

$$j(u_i, w_i) = \frac{h^3}{3}\int_\Gamma \left(\frac{\partial w_i^+}{\partial x_j}n_j - \frac{\partial w_i^-}{\partial x_j}n_j\right)\left(\frac{\partial u_i^+}{\partial x_j}n_j - \frac{\partial u_i^-}{\partial x_j}n_j\right)d\Gamma, \tag{A6}$$

and introducing (A5), as well as similar expressions for the test function, and their corresponding nodal values, $\{d_w\}$, and negative element components, gives

$$j(u_i, w_i) = \frac{h^3}{3}\int_\Gamma \left\{[n]^T[G^+]\{d_w^+\} - [n]^T[G^-]\{d_w^-\}\right\}^T\left\{[n]^T[G^+]\{d^+\} - [n]^T[G^-]\{d^-\}\right\}\ d\Gamma. \tag{A7}$$

Combining the nodal displacements of the positive and negative elements into a single vector allows us to write

$$j(u_i, w_i) = \frac{h^3}{3}\int_\Gamma \left\{[n]^T[G]\{d_w\}\right\}^T \left\{[n]^T[G]\{d\}\right\}\, d\Gamma, \tag{A8}$$

where the combined $[G] = \left[[G^+] \quad -[G^-]\right]$ is defined in Equation (22) and $\{d\}^T = \left\{\{d^+\}^T \quad \{d^-\}^T\right\}$. The nodal values are constants in terms of the integral, which gives

$$j(u_i, w_i) = \frac{h^3}{3}\{d_w\}^T \int_\Gamma [G]^T \underbrace{[n][n]^T}_{[m]}[G] d\Gamma\{d\}, \tag{A9}$$

and, finally, the arbitrary test functions, $\{d_w\}$, can be eliminated since they are nodal constants and pre-multiply all terms of the weak form, leading to Equation (21).

APPENDIX B. THREE DIMENSIONAL GHOST STABILISATION

When moving from two to three-dimensional analysis the majority of the ghost stabilisation equations presented in this article remain unchanged, albeit moving from line to surface integrals. The key quantities that must be modified for three dimensional analysis are the normal matrix associated with the positive element, $[n]$, and the matrix containing the shape function derivatives, $[G^+]$, which must be changed to

$$[n]^T = \begin{bmatrix} n_x & 0 & 0 & n_y & 0 & n_z \\ 0 & n_y & 0 & n_x & n_z & 0 \\ 0 & 0 & n_z & 0 & n_y & n_x \end{bmatrix},$$

and

$$[G^+] = \begin{bmatrix} N^+_{1,x} & 0 & 0 & \dots & N^+_{n,x} & 0 & 0 \\ 0 & N^+_{1,y} & 0 & \dots & 0 & N^+_{n,y} & 0 \\ 0 & 0 & N^+_{1,z} & \dots & 0 & 0 & N^+_{n,z} \\ N^+_{1,y} & N^+_{1,x} & 0 & \dots & N^+_{n,y} & N^+_{n,x} & 0 \\ 0 & N^+_{1,z} & N^+_{1,y} & \dots & 0 & N^+_{n,z} & N^+_{n,y} \\ N^+_{1,z} & 0 & N^+_{1,x} & \dots & N^+_{n,z} & 0 & N^+_{n,x} \end{bmatrix}.$$

Care should also be taken when implementing the approach into axi-symmetric MPM codes, where it is important to account for the out of plane contribution to the in-plane stabilisation terms.