



This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Tran, Quoc Anh; Sołowski, Wojciech Tomasz

Temporal and Null-space filter for the Material Point Method

Published in: International Journal for Numerical Methods in Engineering

DOI: 10.1002/nme.6138

Published: 19/10/2019

Document Version Publisher's PDF, also known as Version of record

Published under the following license: CC BY

Please cite the original version:

Tran, Q. A., & Sołowski, W. T. (2019). Temporal and Null-space filter for the Material Point Method. *International Journal for Numerical Methods in Engineering*, *120*(3), 328-360. https://doi.org/10.1002/nme.6138

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

RESEARCH ARTICLE

WILEY

Temporal and null-space filter for the material point method

Quoc-Anh Tran¹⁰ | Wojciech Sołowski

Department of Civil Engineering, Aalto University, Espoo, Finland

Correspondence

Quoc-Anh Tran, Department of Civil Engineering, Aalto University 10, Rakentajanaukio 4 A, 02150 Espoo, Finland. Email: tran.quocanh@aalto.fi

Funding information Academy of Finland, Grant/Award Number: 286628

Summary

This paper presents algorithm improvements that reduce the numerical noise and increase the numerical stability of the material point method formulation. Because of the linear mapping required in each time step of the material point method algorithm, a possible mismatch between the number of material points and grid nodes leads to a loss of information. These null-space related errors may accumulate and affect the numerical solution. To remove the null-space errors, the presented algorithm utilizes a null-space filter. The null-space filter shown removes the null-space errors, resulting from the rank deficient mapping and the difference between the number of material points and the number of nodes. The presented algorithm enhancements also include the use of the explicit generalized- α integration method, which helps optimizing the numerical algorithmic dissipation. This paper demonstrates the performance of the improved algorithms in several numerical examples.

KEYWORDS

dual domain material point method, generalized interpolation material point method, hydro-coupling, material point method, null-space

1 | INTRODUCTION

Since the proposal of the material point method (MPM) by Sulsky et al,¹ as an extension of a particle-in-cell method² and a fluid implicit particle (FLIP) method³ for solids, MPM has been used to simulate dynamic problems involving large deformations.⁴⁻⁷ However, in the simulations made with the original MPM, there are numerical noises when material points crossing the cell boundaries. These numerical inaccuracies led to the development of other MPM variants, including the generalized interpolation material point method (GIMP),⁸ the dual domain material point method (DDMP),⁹ and the B-spline MPM.¹⁰⁻¹³ Those implementations improved the errors directly related to cell-crossing, yet some high frequency noises are still present in the solutions. A simple approach to reduce those spurious noises is to add a nonviscous damping to the linear momentum balance equations such as described in the implicit¹⁴ or two-phase formulations.^{15,16} However, the local damping leads to energy dissipation, inaccuracies in the time-dependent simulations, such as in the consolidation process¹⁵ and, in extreme cases, to an over-damped system with qualitatively different characteristics. To avoid these severe limitations, for example, Lu et al¹⁷ and Renaud et al¹⁸ combined the time-discontinuous Galerkin method to the MPM to control the spurious noises. Furthermore, Jiang et al^{19,20} have shown that the use of the PIC scheme can filter the spurious noises in the velocity but leads to an excessive dissipation. It leads to the development of the PIC variants such as affine PIC,^{19,20} which employs locally affine velocity gradients, and extended PIC,²¹ which replaces the transpose mapping by the Moore-Penrose inverse mapping. These PIC variants retain the filter properties of

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2019 The Authors. International Journal for Numerical Methods in Engineering Published by John Wiley & Sons Ltd.

PIC and attempt to preserve the information, somewhat reducing energy dissipation. In this paper, a generalized- α dissipative scheme²² is adopted in the framework of the MPM. The generalized- α scheme simply damps out the high-frequency noises only. Therefore, it avoids the dissipation present in the PIC schemes without cumbersome coupling with other numerical methods.

Furthermore, the MPM formulation requires transfer of the state variables between the material points and the background grid. This frequent mapping between material points and grid nodes can cause significant numerical errors and lead to instabilities, such as ringing instability (presented in detail by Brackbill³ for PIC methods), which relates to a null-space problem.²³ Similar to the PIC methods, the null-space in the mapping is also presented in MPM, which leads to spurious numerical oscillations in the solution. In general, the null-space related errors may grow continuously over multiple time steps, and, consequently, the numerical solutions may diverge.^{23,24} Tran et al²⁴ have shown that, if there are more material points than nodes, null-space errors exist despite the choice of the shape function. Therefore, as the null-space existence is due to the larger number of the material points than the grid nodes, the null-space error is present in all the variants of the MPM, including GIMP,⁸ DDMP,⁹ and B-spline MPM.¹⁰⁻¹³ To remove the null-space errors, Gritton et al^{23,25} proposed a null-space filter using the single-value-decomposition (SVD) operator in the 1D formulation of the original MPM. This paper shows how to apply a null-space filter to higher-order interpolation function, for example, in the GIMP or DDMP formulations of the MPM. The presented general formulation also removes the null-space instability at lower computational cost, when compared with the SVD.^{23,25} Furthermore, the paper extends the null-space filter to a two-phase hydro-mechanically coupled formulation of MPM. The potential of the proposed formulation is demonstrated in several numerical examples in both one-phase and two-phase simulations.

2 | GENERALIZED- α INTEGRATION SCHEME FOR MPM

The MPM algorithm has been described in detail by many researchers^{1,4} and summarized in Figure 1. In the MPM framework, the velocity update scheme is based on the acceleration, stemming from the FLIP scheme²⁶ as follows:

$$\mathbf{a}_{p}^{t} = \sum_{i} S_{ip} \mathbf{a}_{i}^{t} dt$$
(1)

$$\mathbf{v}_{p}^{t+1} = \mathbf{v}_{p}^{t} + \mathbf{a}_{p}^{t}$$
⁽²⁾

The other approach is to overwrite the velocity at material points by the nodal velocity using the PIC scheme²

$$\mathbf{v}_{i}^{t+1} = \mathbf{v}_{i}^{t} + \mathbf{a}_{i}^{t} dt$$
(3)

$$\mathbf{v}_{p}^{t+1} = \sum_{i} S_{ip} \mathbf{v}_{i}^{t+1}.$$
(4)



FIGURE 1 Material point method formulation for a single time step [t; t+1]

WILEV

Jiang et al^{19,20} showed that the PIC scheme can filter the spurious modes but exhibits an excessive energy dissipation. In contrast, the FLIP scheme used in MPM is energy conservative, but the spurious modes are preserved and accumulated. Stomakhin et al²⁷ and Nairn²⁸ updated the velocity by combining the PIC and FLIP scheme velocity update, with PIC component as a damping factor. In this paper, we apply the explicit generalized- α time integration scheme proposed by Hulbert and Chung,²² a pure FLIP scheme, but the acceleration is evaluated both in the beginning and the end of the time step. This approach is shown to be able to damp out the high frequency modes only without causing excessive numerical dissipation for structural engineering.²⁹ Kontoe³⁰ has shown that the generalized- α algorithm is more accurate and has better numerical dissipation characteristics than other dissipated time integration schemes.^{31,32} Moreover, the generalized- α algorithm can mitigate the spurious high frequency mode but does not affect the lower modes strongly. Additionally, it allows controlling the numerical dissipation with a single parameter, which is more appropriate than damping related to the numerical time step size.²⁹ In the generalized- α algorithm,³³ the acceleration is updated as follows:

$$\mathbf{m}^{t}\mathbf{a}^{t+1-\alpha_{m}} = \mathbf{f}^{t+1-\alpha_{f}} \tag{5}$$

$$\mathbf{a}^{t+1-\alpha_{m}} = (1-\alpha_{m})\,\mathbf{a}^{t+1} + \alpha_{m}\mathbf{a}^{t} \tag{6}$$

$$\mathbf{f}^{t+1-\alpha_{f}} = (1-\alpha_{f})\,\mathbf{f}^{t+1} + \alpha_{f}\mathbf{f}^{t},\tag{7}$$

where α_m and α_f are time integration parameters, $\mathbf{a}^{t+1-\alpha_m}$ is the nodal acceleration evaluated at $t = (1 - \alpha_m)t^{t+1} + \alpha_m t^t$, and $\mathbf{f} = \mathbf{f}^{int} + \mathbf{f}^{ext}$ is the sum of internal and external forces and evaluated at $t = (1 - \alpha_f)t^{t+1} + \alpha_f t^t$. For the update of displacements and velocities, the original Newmark's equations are

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \left[(1 - \gamma) \, \mathbf{a}^t + \gamma \mathbf{a}^{t+1} \right] \, \mathrm{d}t \tag{8}$$

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{v}^t dt + \left[\left(\frac{1}{2} - \beta \right) \mathbf{a}^t + \beta \mathbf{a}^{t+1} \right] dt^2,$$
(9)

where γ and β are time integration parameters. Chung and Hulbert³³ showed that the generalized- α method is second-order accurate, provided

$$\gamma = \frac{1}{2} - \alpha_{\rm m} - \alpha_{\rm f}. \tag{10}$$

Moreover, the generalized- α method is unconditionally stable, provided

$$\alpha_{\rm m} \le \alpha_{\rm f} \le \frac{1}{2}; \quad \beta \ge \frac{1}{4} + \frac{1}{2} \left(\alpha_{\rm f} - \alpha_{\rm m} \right). \tag{11}$$

To derive the explicit generalized- α method, Hulbert and Chung²² selected $\alpha_f = 1$ and rederived the numerical algorithmic parameters. Because $\alpha_f = 1$, the explicit scheme becomes conditionally stable and Equation (5) becomes

$$\mathbf{a}^{t+1-\alpha_m} = \mathbf{f}^t / \mathbf{m}^t. \tag{12}$$

Hilber and Hughes²⁹ showed that the stability of a time integration scheme depends on the natural frequency $\omega = \sqrt{K/M}$, where K and M are stiffness and mass, respectively, and time step is dt. Therefore, $\Omega = \omega^*$ dt is considered to evaluate the stability of the scheme. The explicit generalized- α integration²² is stable with $\Omega \in [0, \Omega_s]$, where Ω_s is the critical limit. The maximum dissipation is obtained at a bifurcation limit Ω_b . The spectral radius at the bifurcation limit ρ_b is a user-specified parameter to control the algorithmic dissipation. If ρ_b is equal to 1, there is no dissipation. When the spectral radius ρ_b reduces, the numerical damping increases and gets to maximum values at $\rho_b = 0$ (see Figure 2). The bifurcation limit Ω_b and the critical limit Ω_s are the function of ρ_b as follows:

$$\Omega_{\rm b} = (1+\rho_{\rm b})\sqrt{2-\rho_{\rm b}} \tag{13}$$

$$\Omega_{\rm s} = \sqrt{\frac{12(1+\rho_{\rm b})^3 (2-\rho_{\rm b})}{10+15\rho_{\rm b}-\rho_{\rm b}^2+\rho_{\rm b}^3-\rho_{\rm b}^4}}.$$
(14)

-Wiifv

WILEY 331



FIGURE 2 Evolution of spectral radius for explicit generalized- α method²²

Other algorithm parameters could be linked to ρ_b as follows:

$$\alpha_{\rm m} = \frac{2\rho_{\rm b} - 1}{1 + \rho_{\rm b}} \tag{15}$$

$$\beta = \frac{5 - 3\rho_b}{\left(1 + \rho_b\right)^2 \left(2 - \rho_b\right)}$$
(16)

$$\gamma = \frac{3}{2} - \alpha_{\rm m}.\tag{17}$$

Overall, the generalized- α method requires to compute the nodal accelerations twice, first at the beginning \mathbf{a}_i^t and then at the end of the time step \mathbf{a}_i^{t+1} . This scheme doubles the amount of computations needed, hence we apply another approach for MPM based on tracking the acceleration \mathbf{a}_p^t at material points. Therefore, each material points have two kinetic variables: velocity \mathbf{v}_p^t and acceleration \mathbf{a}_p^t , which are mapped to the grid as follows:

$$\mathbf{v}_{i}^{t} = \left(\sum_{i} S_{ip} m_{p} \mathbf{v}_{p}^{t}\right) / m_{i}^{t}$$
(18)

$$\mathbf{a}_{i}^{t} = \left(\sum_{i} S_{ip} m_{p} \mathbf{a}_{p}^{t}\right) / m_{i}^{t}.$$
(19)

The nodal accelerations $\mathbf{a}_{i}^{t+1-\alpha_{m}}$ and \mathbf{a}_{i}^{t+1} and nodal velocities \mathbf{v}_{i}^{t+1} are now computed explicitly as follows:

$$\mathbf{a}_{i}^{t+1-\alpha_{m}} = \mathbf{f}_{i}^{t}/m_{i}^{t}$$
(20)

$$\mathbf{a}_{i}^{t+1} = \left(\mathbf{a}_{i}^{t+1-\alpha_{m}} - \alpha_{m}\mathbf{a}_{i}^{t}\right) / (1-\alpha_{m})$$
⁽²¹⁾

$$\mathbf{v}_{i}^{t+1} = \mathbf{v}_{i}^{t} + \left[(1 - \gamma) \, \mathbf{a}_{i}^{t} + \gamma \mathbf{a}_{i}^{t+1} \right] \mathrm{dt}.$$

$$(22)$$

Finally, the velocity, position, and acceleration of material points are updated, ie,

$$\mathbf{v}_{p}^{t+1} = \mathbf{v}_{p}^{t} + \sum_{i} S_{ip} \left[(1-\gamma) \, \mathbf{a}_{i}^{t} + \gamma \mathbf{a}_{i}^{t+1} \right] dt$$
⁽²³⁾

$$x_{p}^{t+1} = x_{p}^{t} + \sum_{i} S_{ip} \mathbf{v}_{i}^{t+1} dt + \sum_{i} S_{ip} \left[\left(\frac{1}{2} - \beta \right) \mathbf{a}_{i}^{t} + \beta \mathbf{a}_{i}^{t+1} \right] dt^{2}$$
(24)

$$\mathbf{a}_{p}^{t+1} = \sum_{i} S_{ip} \mathbf{a}_{i}^{t+1}.$$
(25)

The algorithm is controlled by only a single parameter ρ_b , as all the other parameters α_m , β , γ are automatically updated (see Equations 15-17). In the numerical examples, ρ_b was set to be 0.818, as recommended by Kontoe et al.³²



FIGURE 3 Schematic of different mapping scheme in material point method (MPM). DDMP, dual domain material point method; GIMP, generalized interpolation material point method

3 | NULL-SPACE FILTER FOR MPM

3.1 | Choice of shape functions gradient ∇S_{ip}

The original MPM¹ selects the gradient of the shape function ∇S_{ip} such that the sign of the internal force changes when material points cross to a new cell. Consequently, cell-crossing creates unphysical noise and possibly leads to numerical instability. The cell-crossing can be mitigated by using characteristic functions in the GIMP.⁸ The characteristic functions in GIMP give the material points domains (red square in Figure 3) and continuous gradients. Another approach, which reduces the cell-crossing errors, is DDMP.⁹ The DDMP replaces the gradient ∇S_{ip} with a continuous function between the grid cells while leaving the shape functions intact. The shape function gradient in the cell, which contains the material points, is the shape function gradient of the original MPM, denoted ∇S_i , whereas the shape function gradient in the neighbor-node cells is the node-based function, denoted $\overline{\nabla S_i}$. The DDMP enlarges the influence domain of material points through dual domains, a cell contains the material points (gray cell in Figure 3) and neighbor-node cells (blue dash cell in Figure 3), without using the characteristic functions to track the domain of the material points.

In DDMP, the gradient of the shape function is modified to

$$\overline{\nabla S_{i}}(\mathbf{x}) = \alpha(\mathbf{x}) \nabla S_{i} + [1 - \alpha(\mathbf{x})] \widetilde{\nabla S_{i}}, \qquad (26)$$

with the node-based function $\widetilde{\nabla S_i}$ being

$$\widetilde{\nabla S_{i}}(\mathbf{x}) = \sum_{j=1}^{N_{n}} \frac{S_{j}(\mathbf{x})}{V_{j}} \int_{\Omega} S_{j} \nabla S_{i} dv, \qquad (27)$$

where $V_j = \int_{\Omega} S_j dv$ is the nodal volume at node *j*. In Equation (27), the integral $\int_{\Omega} S_j \nabla S_i dv$ is calculated as a convolution of the gradient of the shape function (∇S_i) and the shape function of neighbor nodes *j* (S_j). Therefore, it can be solved analytically, as the background grid remains unchanged during the calculation.

Figure 4 shows the gradient of the shape function in a 1D case. On the right-hand side in Equation (26), the first term represents the mapping of material points to particle-based nodes (defined as particle-based mapping) and the second term represents the mapping of material points to node-based nodes (defined as node-based mapping). Both mappings are weighted by α , a weighting function. The gradient of the shape function is continuous only if α is continuous and $\alpha = 0$ at cell boundaries. The choice of α can retrieve the gradient of the shape function of other versions of MPM as follows:

$$\nabla S_{i} = \begin{cases} \nabla S_{i} \text{ for MPM, } & \alpha = 1\\ \overline{\nabla S_{i}} \text{ for DDMP, } & \alpha = \alpha_{1}\\ \overline{\nabla S_{i}} \text{ for cpGIMP, } & \alpha = \alpha_{2}. \end{cases}$$
(28)

The Appendix gives more details related to function α . In this paper, the dual domain technique is used to switch the shape function gradient between MPM, cpGIMP, and DDMP by the choice of weighted function α in Equation (28).



FIGURE 4 Gradient of the shape function in 1D. DDMP, dual domain material point method; GIMP, generalized interpolation material point method; MPM, material point method

3.2 | Null-space instability in the gradient mapping matrix ∇S

Considering a mapping from a material point stress *p* to a grid node force *i*, the mapping can be written as follows:

$$\begin{bmatrix} f_{i,x} & f_{i,y} & f_{i,z} \end{bmatrix} = \begin{bmatrix} \nabla S_{i,x} (\mathbf{x}_p) & \nabla S_{i,y} (\mathbf{x}_p) & \nabla S_{i,z} (\mathbf{x}_p) \end{bmatrix} \cdot \mathbf{V}_p \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}.$$
(29)

For a global mapping from all N_p material point's stress to all N_n grid node's internal force, the global mapping can be written as follows:

$$\begin{bmatrix} \mathbf{f}_{1} \\ \mathbf{f}_{2} \\ \vdots \\ \mathbf{f}_{N_{n}} \end{bmatrix} = \begin{bmatrix} \nabla \mathbf{S}_{1}(\mathbf{x}_{1}) & \nabla \mathbf{S}_{1}(\mathbf{x}_{2}) & \cdots & \nabla \mathbf{S}_{1}(\mathbf{x}_{N_{p}}) \\ \nabla \mathbf{S}_{2}(\mathbf{x}_{1}) & \nabla \mathbf{S}_{2}(\mathbf{x}_{2}) & \cdots & \nabla \mathbf{S}_{2}(\mathbf{x}_{N_{p}}) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla \mathbf{S}_{N_{n}}(\mathbf{x}_{1}) & \nabla \mathbf{S}_{N_{n}}(\mathbf{x}_{2}) & \cdots & \nabla \mathbf{S}_{1}(\mathbf{x}_{N_{p}}) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_{1}\mathbf{\sigma}_{1} \\ \mathbf{V}_{2}\mathbf{\sigma}_{2} \\ \vdots \\ \mathbf{V}_{N_{p}}\mathbf{\sigma}_{N_{p}} \end{bmatrix},$$
(30)

where N_n and N_p are the total number of the nodes and particles, respectively. Equation (30) can be written in the matrix form

$$\mathbf{F}^{\text{int}} = \nabla \mathbf{S} \cdot \left(\boldsymbol{\sigma}_{\text{p}} \mathbf{V}_{\text{p}} \right), \tag{31}$$

where \mathbf{F}^{int} is the vector of all nodal internal forces and $(\boldsymbol{\sigma}_p V_p)$ is the product of the material point stress tensor and the corresponding material point volume, for all material points. The gradient mapping matrix $\nabla \mathbf{S}$ depends on the value of the shape function gradient $\nabla S_i(\mathbf{x}_p)$, with the size of the matrix $\nabla \mathbf{S}$ being $N_n \times N_p$ in 1D case. In reverse, the velocity gradient vector at material points is computed by the transpose of $\nabla \mathbf{S}$ as follows:

$$\nabla \mathbf{v}_{\mathrm{p}} = \nabla \mathbf{S}^{\mathrm{T}} \cdot \mathbf{v}_{\mathrm{i}},\tag{32}$$

where $\nabla \mathbf{v}_p$ is the velocity gradient vector of all material points and \mathbf{v}_i is the nodal velocity vectors of all grid nodes. The null-space of the gradient mapping matrix $\nabla \mathbf{S}$, written as null($\nabla \mathbf{S}$), is the set of all nonzero independent vectors $\boldsymbol{\sigma}_p^{null}$, which satisfy that $\nabla \mathbf{S} \cdot \boldsymbol{\sigma}_p^{null} = 0$, ie,

$$\operatorname{null}\left(\nabla \mathbf{S}\right) = \left\{ \boldsymbol{\sigma}_{p}^{\operatorname{null}} \in \mathbb{R}^{N_{p}}, \boldsymbol{\sigma}_{p}^{\operatorname{null}} \neq 0 | \nabla \mathbf{S} \cdot \boldsymbol{\sigma}_{p}^{\operatorname{null}} = 0 \right\}.$$
(33)

The presence of the null-space may lead to infinite solutions. Hence, the mapping in Equation (31) may result in a nonunique solution. In other words, there possibly exists a set of stress vectors $\boldsymbol{\sigma}_p^{null}$ at material points, which leads to zero internal forces after mapping with Equation (31). Because the balance equations are solved at the grid, the $\boldsymbol{\sigma}_p^{null}$ in the null-space of ∇S cannot be detected on the background grid and represents the null-space noise. This noise can accumulate and leads to instability in the solution.

The rank of the gradient mapping matrix ∇S , written as rank(∇S), is the set of linearly independent column vector spaces. Equation (31) and Equation (32) give the gradient mapping based on the ∇S and its transpose ∇S^T , which transfer

the data between material points and grid nodes and vice versa. Therefore, the following four fundamental subspaces are considered:

- the row space, rank(∇S^{T}), a subspace of R^{N_n} ;
- the column space, rank(∇ **S**), a subspace of R^{N_p};
- the right null-space, null($\nabla \mathbf{S}$), a subspace of $\mathbb{R}^{N_{p}}$;
- the left null-space, null($\nabla \mathbf{S}^{\mathbf{T}}$), a subspace of \mathbb{R}^{N_n} .

Since the column rank is equal to the row rank,³⁴ we have rank(∇ **S**) = rank (∇ **S**^T). According to the rank-nullity theorem,³⁴ the sum of column space and null-space is equal to the column numbers of the matrix as follows:

$$\operatorname{rank}\left(\nabla \mathbf{S}\right) + \operatorname{null}\left(\nabla \mathbf{S}\right) = N_{p} \tag{34}$$

$$\operatorname{rank}\left(\nabla \mathbf{S}\right) + \operatorname{null}\left(\nabla \mathbf{S}^{\mathbf{T}}\right) = N_{n}.$$
(35)

In case there are more grid nodes than material points, the linear equations may lead to rank-deficiency problem. A similar problem may be encountered in the finite element method using a low order of quadrature rules (see rank-deficiency, page 239, in the work of Hughes³⁵). This issue can be mitigated in finite element method (and, in theory, in MPM) by using a high-order interpolation. Commonly, there are more material points than grid nodes, therefore, we only consider that $N_p > N_n$ in the analysis. Considering $N_p > N_n$, as can be seen from Equation (34) and Equation (35), when ∇S is full rank, rank(∇S) is maximum and is equal to the number of grid nodes N_n . In contrast, if ∇S is rank-deficient, that is, rank(∇S) = r < N_n , the left null-space exists and null(∇S^T) = $N_n - r$.

In the original MPM, the gradient mapping is rank-deficient²⁴ when there are more material points than nodes $N_p > N_n$. In such case, ∇S could have rank(∇S) = N_n (ie, full rank) by using higher-order shape function (for example cubic B-spline) so that the left null-space is equal to zero null(∇S^T) = 0. For example, MPM, DDMP, and GIMP contain rank-deficient mappings, whereas cubic B-spline MPM²⁴ is full-rank mapping. However, right null-space still exists null(∇S) > 0 despite the choice of the shape function even for cubic B-spline MPM because it depends only on the difference between the number of material points and the number of nodes rather than on the order of the interpolating function. As such, a higher number of material points than the number of grid nodes in a cell leads to a right null-space and associated instability. That may be the reason why using one material point per cell could give less errors than a higher number of material points per cell⁸ because N_p < N_n. As the null-space instability exists when N_p > N_n, one can enhance the stability of the solution with an application of a null-space filter, which removes the null-space error. The next section presents how to detect the null-space components of velocity gradients.

3.3 | Detect null-space components of velocity gradients

To detect the null-space component, the transpose of the gradient mapping matrix needs to be decomposed in the QR form first, ie,

$$\begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_{N_p} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_{N_p} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \cdots & \mathbf{r}_{1N_n} \\ \mathbf{r}_{21} & \mathbf{r}_{22} & & \mathbf{r}_{2N_n} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & & \mathbf{r}_{N_nN_n} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\nabla \mathbf{S}_{N_p \times N_n}^{\mathrm{T}} = \mathbf{Q}_{N_p \times N_p} \cdot \mathbf{R}_{N_p \times N_n}$$
(36)

where **R** is the upper triangular matrix with r_{ij} are nonzero if ∇S is full rank. If ∇S is rank-deficient, the zero rows will develop from the bottom of matrix **R** and contribute to the left null-space (see Equation (36)). **Q** is the unitary matrix, in which the columns \mathbf{q}_i are orthonormal vectors that satisfy $\mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}$ and span in the space \mathbb{R}^{N_p} . Figure 5 shows the QR decomposition for the gradient mapping matrix. The unitary matrix **Q** is divided into the column space $\mathbf{Q}^*_{N_n \times r}$ and the null-space. The null-space includes the left null-space, null(∇S^T) = $N_n - r$, and the right null-space null(∇S) = $N_p - N_n$. The velocity gradient vector of all the material points $\nabla \mathbf{v}_p \in \mathbb{R}^{N_p}$ can be expressed as a linear combination of the orthonormal vector \mathbf{q}_i with the coefficients a_i , ie,

$$\nabla \mathbf{v}_{p} = \underbrace{\mathbf{a}_{1}\mathbf{q}_{1} + \dots + \mathbf{a}_{r}\mathbf{q}_{r}}_{\operatorname{rank}(\nabla \mathbf{S})=r} + \underbrace{\mathbf{a}_{r+1}\mathbf{q}_{r+1} + \dots + \mathbf{a}_{N_{n}}\mathbf{q}_{N_{n}}}_{\operatorname{null}(\nabla \mathbf{S}^{T})=N_{n}-r} + \underbrace{\mathbf{a}_{N_{n}+1}\mathbf{q}_{N_{n}+1} + \dots + \mathbf{a}_{N_{p}}\mathbf{q}_{N_{p}}}_{\operatorname{null}(\nabla \mathbf{S})=N_{p}-N_{n}}.$$
(37)

334



FIGURE 5 QR decomposition of gradient mapping $\nabla S(N_p > N_n)$

In Equation (37), the solution of the velocity gradient vector of the material points consists of a linear combination of column space and null-space. The left null-space, null(∇S^T), represents the loss of information due to rank-deficient mapping, whereas the remaining components, null(∇S) – null(∇S^T) = N_p – N_n, represent the loss of information due to the difference between the number of material points and the number of nodes. The nonnull-space component, after removing the null-space errors, is

$$\nabla \mathbf{v}_{\mathrm{p}}^{*} = a_{1}\mathbf{q}_{1} + \dots + a_{\mathrm{r}}\mathbf{q}_{\mathrm{r}}.$$
(38)

3.4 | Null-space filter algorithm for MPM

This section presents the null-space filter algorithm in greater detail. The null-space filter algorithm requires two steps. First, the algorithm computes the orthonormal basis of column space $\{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_r\}$, as described in section A. Afterwards, it projects the velocity gradients to the orthonormal basis of the column space, as described in section B.

3.4.1 | Compute the orthonormal basis of column space $\{q_1, q_2, \dots, q_r\}$

Although Equation (36) is written for the global case, ie, for $\nabla S_{N_n \times N_p}$, where N_p and N_n are the total number of material points and nodes, respectively, in practice, the null-space filter can be done locally with the use of the local QR algorithm gradient mapping for each cell is ∇S_c . The size of ∇S_c is n_i by n_p , with n_i being the number of the nodes of the cell and n_p the number of the material points inside this cell.

Algorithm 1 Modified Householder algorithm for 1D

Input:

 ∇S_c , dSc - matrix of 1D shape function derivatives computed at material points

r - rank of $\nabla \mathbf{S}_c$

n_p- number of material points in cell

Output: $\mathbf{Q}_{\mathbf{c}}^{1D}$, Qc - orthonormal basis of the column space of $\nabla \mathbf{S}_{c}$, size $n_{p} \ge r$

Pseudocode for 1D case	Matlab 1D code
$\mathbf{Q} = \mathbf{I}[\mathbf{n}_{\mathrm{p}}]$	r = rank (dSc); Qc = eye(np, r);
$\mathbf{if} \mathbf{n}_{\mathrm{p}} > \mathbf{r}$	if $np > r$
for $k = 1$ to r	for $k = 1 : r$
$\vec{\mathbf{s}} = \nabla \mathbf{S}_{c}[:, K]$	s = dSc(:, k);
$\mathbf{v} = \ \vec{\mathbf{s}}\ $	v = norm(s);
$\vec{\mathbf{s}}[k] = \vec{\mathbf{s}}[k] + v$	$\mathbf{s}(\mathbf{k}) = \mathbf{s}(\mathbf{k}) + \mathbf{v};$
$ec{\mathbf{w}} = ec{\mathbf{s}}/\ ec{\mathbf{s}}\ $	w = s/norm(s);
$\mathbf{Q}_{\mathbf{c}}^{\mathrm{1D}}[:,k] = \mathbf{Q}[:,k] - 2 \vec{\mathbf{w}}(\vec{\mathbf{w}}^{\mathrm{T}} \cdot \mathbf{Q}[:,k])$	Qc(:, k) = Qc(:, k) - 2 * w * (w'*Qc(:, k));
end	end
end	end

335

WII FY-



FIGURE 6 Computational efficiency of null-space filter. DDMP, dual domain material point method; SVD, single-value decomposition [Colour figure can be viewed at wileyonlinelibrary.com]

Because the grid is not deformed during the calculation, the value of the gradient of the linear basis function is constant within the grid cell and independent from the material point coordinates in one dimension. Therefore, the rank of the local ∇S_c can be computed analytically. As such, we can optimize the computational efficiency, with given $r = \operatorname{rank}(\nabla S_c)$, by modified householder algorithm (see Algorithm 1). Instead of computing all the n_p columns of matrix Q and n_i columns of matrix R, the modified householder only computes the column space of Q up to order of r. Therefore, the modified householder algorithm reduces the computational cost compared to the SVD²³ or QR decomposition²⁴ (see Figure 6 for numerical example 6.1). In a 3D case, with a structured grid, the rank r is computed, and then Algorithm 1 calculates three sets of orthonormal basis that are calculated separately for each dimension, leading to Q_c^x , Q_c^y , and Q_c^z . Note that, in this case, the values of the gradient ∇S_c depend on the position of the material points.

3.4.2 | Project the velocity gradients to the orthonormal basis of column space

To implement the projection for local null-space filter, we consider a cell that contains n_i nodes and n_p material points with the gradient mapping matrix ∇S_c . Then, the velocity gradient is computed by the particle-based mapping, ie,

$$\nabla \mathbf{v}_{\mathrm{p}} = \nabla \mathbf{S}_{\mathrm{c}}^{\mathrm{T}} \cdot \mathbf{v}_{\mathrm{i}}.\tag{39}$$

Afterwards, the nonnull-space components of velocity gradients, $\nabla \mathbf{v}_p^*$, could be extracted by a projection with the orthonormal basis of column spaces $\{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_r\}^{34}$ by

$$\nabla \mathbf{v}_{p}^{*} = \operatorname{Proj}_{\mathbf{Q}^{*}} \left(\nabla \mathbf{v}_{p} \right) = \mathbf{Q}^{*} \cdot \left(\mathbf{Q}^{*T} \cdot \nabla \mathbf{v}_{p} \right).$$

$$(40)$$

In the 3D case, with a structured grid, the projection of the velocity gradient $\nabla \mathbf{v}_p = [\nabla \mathbf{v}_p^x; \nabla \mathbf{v}_p^y; \nabla \mathbf{v}_p^z]$ is done for each dimension as follows:

$$\nabla \mathbf{v}_{p}^{*,x} = \mathbf{Q}_{c}^{x} \cdot \left(\mathbf{Q}_{c}^{x*T} \cdot \nabla \mathbf{v}_{p}^{x}\right); \nabla \mathbf{v}_{p}^{*,y} \mathbf{Q}_{c}^{x} \cdot \left(\mathbf{Q}_{c}^{y*T} \cdot \nabla \mathbf{v}_{p}^{y}\right); \nabla \mathbf{v}_{p}^{*,z} = \mathbf{Q}_{c}^{z} \cdot \left(\mathbf{Q}_{c}^{z*T} \cdot \nabla \mathbf{v}_{p}^{z}\right), \tag{41}$$

where \mathbf{Q}_c^x , \mathbf{Q}_c^x , \mathbf{Q}_c^x are the sets of orthonormal basis in each dimension and $\nabla \mathbf{v}_p^* = [\nabla \mathbf{v}_p^{*,x}; \nabla \mathbf{v}_p^{*,y}; \nabla \mathbf{v}_p^{*,z}]$ is the filtered value of the velocity gradient. Finally, the velocity gradient is weighted between $\nabla \mathbf{v}_p^*$ and the velocity gradient using the node-based mapping $\widetilde{\nabla \mathbf{v}_p}$, based on Equation (26), as

$$\overline{\nabla \mathbf{v}_{\mathbf{p}}}\left(\mathbf{x}\right) = \alpha\left(\mathbf{x}\right)\nabla\mathbf{v}_{\mathbf{p}}^{*}\left(\mathbf{x}\right) + \left[1 - \alpha\left(\mathbf{x}\right)\right]\overline{\nabla \mathbf{v}_{\mathbf{p}}}\left(\mathbf{x}\right).$$
(42)

The selection of α in Equation (49) leads to replication of the shape function gradient of different MPM formulations. Setting ($\alpha = 1$) recovers the original formulation of MPM, GIMP is obtained with ($\alpha = \alpha_2$), and DDMP with ($\alpha = \alpha_1$) with the choice of α described in the Appendix.



FIGURE 7 Schematic of double-point material point method

4 | DOUBLE-POINT MPM FORMULATION WITH NULL-SPACE FILTER

In this paper, the two-phase formulation bases on a double-point formulation of MPM,^{24,36} in which each phase is represented by a set of two material points, one for the liquid phase and one for the solid phase. The information from the material points is projected to the grid separately for the solid and liquid phases (see Figure 7). The algorithms mapping the data from the material points to the grid and back are the same for both liquid and solid phases. The solid and liquid phases are interacting with each other via dragging forces. The governing and time discretization equations are presented in detail by Bandara and Soga³⁶ and Tran and Sołowski³⁷ and summarized in Figure 8. Note that the gradient mapping matrix ∇S is used to compute the nodal internal forces and the material point's strain. Therefore, to implement the null-space filter into the two-phase double-point formulation, the calculation of the nodal internal forces and the material point's strain is modified using the dual domain technique and the null-space filter and presented in the next section.

4.1 | Calculation of nodal internal forces

The vectors of the internal forces for the solid and liquid phase $\mathbf{f}_{\alpha i}^{\text{int,t}}$ (with $\alpha = s, w$) are calculated using the dual domain technique, which, when used with specific values of coefficients, can replicate DDMP, GIMP, and the original MPM formulation, ie,

$$\mathbf{f}_{si}^{\text{int,t}} = \sum_{sp=1}^{N_{sp}} \nabla S_i \left(\mathbf{x}_{sp}^t \right) \alpha \left(\mathbf{x}_{sp}^t \right) \mathbf{\sigma}_{sp}^t V_{sp}^t - \sum_{j=1}^{N_n} \mathbf{\sigma}_{sj}^t \cdot \int_{\Omega_s} S_j \nabla S_i dv$$
(43)

$$\mathbf{f}_{wi}^{int,t} = \sum_{wp=1}^{N_{wp}} \nabla S_i\left(\mathbf{x}_{wp}^t\right) \alpha\left(\mathbf{x}_{wp}^t\right) \mathbf{p}_{wp}^t V_{wp}^t - \sum_{j=1}^{N_n} \mathbf{p}_{wj}^t \cdot \int_{\Omega_w} S_j \nabla S_i dv,$$
(44)

where $\mathbf{\sigma}_{sp}^{t}$ and $\mathbf{p}_{wp}^{t} = (p_{wp}^{t}\mathbf{I})$ are the effective stress and pore water pressure tensor, V_{sp}^{t} and V_{wp}^{t} are the solid and liquid volume. $\alpha(\mathbf{x}_{\alpha p}^{t})$ is a weighted function evaluated at the material point $\mathbf{x}_{\alpha p}^{t}$, and the choice of weighted function is described in the Appendix to switch different version of MPM including MPM, GIMP, and DDMP. The first term in the right-hand side of Equation (43) and Equation (44) is the MPM particle-based mapping for the node *i* and the second term, in the right-hand side of Equation (44), corresponds to the node-based mapping from the neighbor nodes *j*. The nodal effective stresses for the solid phase $\mathbf{\sigma}_{sj}^{t}$ or nodal pore water pressures for the liquid phase \mathbf{p}_{wj}^{t} at the neighbor nodes *j* are calculated as follows:

$$\boldsymbol{\sigma}_{sj}^{t} = \frac{1}{V_{sj}} \sum_{sp=1}^{N_{sp}} S_{j} \left(\boldsymbol{x}_{sp}^{t} \right) \left(1 - \alpha \left(\boldsymbol{x}_{sp}^{t} \right) \right) \boldsymbol{\sigma}_{sp}^{t} V_{sp}^{t}$$
(45)

$$\boldsymbol{p}_{wj}^{t} = \frac{1}{V_{wj}} \sum_{wp=1}^{N_{wp}} S_{j} \left(\boldsymbol{x}_{wp}^{t} \right) \left(1 - \alpha \left(\boldsymbol{x}_{wp}^{t} \right) \right) \boldsymbol{p}_{wp}^{t} V_{wp}^{t}.$$
(46)

4.2 | Calculation of the material points strain with null-space filter

The strain increment vectors $\Delta \varepsilon_{\alpha p,1}^{k+1}$ using MPM particle-based mapping at the solid and liquid material points ($\mathbf{x}_{\alpha p}^{k}$) are calculated as follows:

$$\Delta \varepsilon_{\alpha p,1}^{t+1} = \sum_{i=1}^{N_n} \left(\nabla S_i \left(\mathbf{x}_{\alpha p}^t \right) \mathbf{v}_{\alpha i}^{L} + \left(\nabla S_i \left(\mathbf{x}_{\alpha p}^t \right) \mathbf{v}_{\alpha i}^{L} \right)^{T} \right).$$
(47)

338 WILEY



FIGURE 8 Double-point material point method formulation for a sing time step

The strain increment is filtered with null-space filter, using Equation (40). The filtered components using MPM particle-based mapping $\Delta \varepsilon_{\alpha p,1}^{*k+1}$ is

$$\Delta \boldsymbol{\varepsilon}_{\alpha p,1}^{*t+1} = \mathbf{Q}^* \cdot \left(\mathbf{Q}^{*\mathbf{T}} \cdot (\Delta \boldsymbol{\varepsilon}_{\alpha p,1}^{t+1}) \right).$$
(48)

The strain increment using node-based mapping $\Delta \varepsilon_{\alpha p,2}^{k+1}$ is

$$\Delta \boldsymbol{\varepsilon}_{\alpha p,2}^{t+1} = \sum_{i=1}^{N_n} S_i \left(\mathbf{x}_{\alpha p}^t \right) \Delta \boldsymbol{\varepsilon}_{\alpha j}^t, \tag{49}$$

where nodal strain increments $\Delta \boldsymbol{\epsilon}_{\alpha i}^{t}$ calculated as follows:

$$\Delta \boldsymbol{\varepsilon}_{\alpha j}^{t} = \frac{1}{V_{\alpha j}} \sum_{i=1}^{N_{n}} \mathbf{v}_{\alpha i}^{L} \int_{\Omega_{\alpha}} S_{j} \nabla S_{i} dv.$$
(50)

The final strain increment vectors $\Delta \epsilon_{\alpha p}^{k+1}$ is calculated with the weighted function $\alpha(\mathbf{x})$, ie,

$$\Delta \boldsymbol{\varepsilon}_{\alpha p}^{t+1} = \alpha \left(\mathbf{x}_{\alpha p}^{t} \right) \cdot \Delta \boldsymbol{\varepsilon}_{\alpha p, 1}^{t+1} + \left(1 - \alpha \left(\mathbf{x}_{\alpha p}^{t} \right) \right) \Delta \boldsymbol{\varepsilon}_{\alpha p, 2}^{t+1}.$$
(51)

5 | NUMERICAL EXAMPLES: APPLICATION OF THE TEMPORAL FILTER

5.1 | Square wave in an elastic bar

The first numerical example is a simulation of an elastic bar under compression. Figure 9 presents the boundary and loading condition of the numerical model. The length of the bar L = 1 m, the Young's modulus E = $10e^6$ Pa, and the density $\rho = 1000 \text{ kg/m}^3$. The elastic wave velocity is calculated as $c = \sqrt{E/\rho} = 100 \text{ m/s}$. The compression force f is applied to the material points at the right boundary to replicate the square stress wave in the elastic bar as follows:

$$f = \begin{cases} 1, & t < 0.5L/c \ (s) \\ 0, & t \ge 0.5L/c \ (s) . \end{cases}$$
(52)





FIGURE 12 Stress wave along the bar using generalized interpolation material point method (GIMP) with fluid implicit particle scheme [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 13 Stress wave along the bar using dual domain material point method (DDMP) with fluid implicit particle scheme [Colour figure can be viewed at wileyonlinelibrary.com]

The 1-m long bar is discretized by 1000 equally distributed material points located in 500 equal-sized grid cells. That leads to the cell size of 0.002 m ($l_i = 0.002$ m). The time step is set to be equal to 5% of the Courant-Friedrichs-Lewy (CFL) critical time step ($\Delta t = 0.05 l_i/c = 10e^{-6}$ s). In all shown numerical results, the numerical models are compared with the analytical solutions at t = 0.75 L/c = 0.0075 s. Figure 10 shows that MPM, using PIC scheme (Equation (4)), can filter the velocity oscillations, hence, filter the stress/strain oscillations, whereas MPM, GIMP, and DDMP show stress oscillations (see Figure 11, Figure 12, and Figure 13). Similar oscillations were reported in the literature.^{17,38} The generalized- α scheme can remove the oscillations for all methods including MPM, GIMP, and DDMP. In this example, the null-space filter does not change the results significantly because the errors are mainly stemming from the spurious noises from the high-frequency mode. To check the energy conservation, the total energy E is the sum of the strain U and the kinetic energy K are computed in each time step as follows:

$$\mathbf{K} = \frac{1}{2} \sum_{\mathbf{p}} \mathbf{m}_{\mathbf{p}} \left\| \mathbf{v}_{\mathbf{p}} \right\|^2 \tag{53}$$

$$U = \frac{1}{2} \sum_{p} \sigma_{p,ij} \varepsilon_{p,ij} V_{p}$$
(54)

$$E = U + K.$$
⁽⁵⁵⁾

Figure 14 shows the total energy after releasing the compression forces at 0.005 s. The oscillations in MPM using FLIP scheme induces the increase of the total energy, whereas both MPM using PIC scheme and generalized- α scheme demonstrate approximately constant total, energy after releasing the compression force.

5.2 | Colliding disks

The second example is the colliding impact of two elastic disks, replicated the simulation by Sulsky et al¹ with given geometry and parameters in Figure 15. The background grid is discretized with the cell size of 0.05 m in plane strain condition and there are total 896 material points and no contact law is applied. The time step is 0.005 s to be equal to 10%



of CFL critical time step. Figure 16, Figure 17, and Figure 18 show the visualization of the disk impact with the velocity colorbar at initial condition, 1.05 s and 2.5 s. Figure 19, Figure 20, and Figure 21 present the evolution of the kinetic, strain, and total energy for the cpGIMP using PIC scheme, FLIP scheme, and generalized-a scheme, respectively. As can be seen,



FIGURE 18 Releasing disks (t = 3.5 s)

FIGURE 19 cpGIMP with particle-in-cell scheme. GIMP, generalized interpolation material point method [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 20 cpGIMP with α -generalized scheme. GIMP, generalized interpolation material point method [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 21 cpGIMP with fluid implicit particle scheme. GIMP, generalized interpolation material point method [Colour figure can be viewed at wileyonlinelibrary.com]







cpGIMP using PIC update scheme shows a strong energy dissipation although it can filter the velocity oscillations in the previous example. In contrast, cpGIMP (with the originally used FLIP update scheme) improves significantly the energy conservation but leads to oscillations under compression impact in the previous example. Some other PIC variants ($aPIC^{19}$ and $xPIC(m)^{21}$) are developed to retain the filter property and improve the energy conservation but the total energy are lower compared with MPM (with the originally used FLIP update scheme). The cpGIMP using the α -generalized scheme can filter the velocity and stress noise due to high-frequency oscillations in the previous example and preserve similar total energy conservation with the cpGIMP without the excessive dissipation (see Figure 22).

5.3 | Gaussian explosion

In the next example, we simulate a 2D wave propagation in an elastic domain with Gaussian-type explosion.^{39,40} An explosive source was placed at $[x_c; y_c]$ the center of a homogeneous, isotropic elastic square (see Figure 23), replicated in the simulation by an exponential force applying to material points with the position [x; y] as follows:

$$f(t, x, y) = \begin{cases} f_1(t) f_2(x, y), & \begin{cases} 0 < t < 2t_o \\ (x - x_c)^2 + (y - y_c)^2 < R^2 \\ 0, & \text{otherwise}, \end{cases}$$
(56)

where

$$f_1(t) = -0.01\pi^2 f_0^2 (t - t_0) e^{-\pi^2 f_0^2 (t - t_0)^2}$$
(57)

$$f_{2}(x,y) = \left[1 - \left(x^{2} + y^{2}\right)/R^{2}\right]^{3} \left[\frac{x - x_{c}}{R}; \frac{y - y_{c}}{R}\right]$$
(58)

$$f_o = \frac{c}{Nl_e}.$$
(59)

In the equation earlier, R is the radius of the explosion source, f_0 is the frequency of the source, $t_0 = 1/f_0$, c is the wave velocity $c = \sqrt{E/\gamma}$, and N is the number of points per wave length. In this example, the parameters used are R = 0.1 m and N = 20.



FIGURE 23 Schematic of the numerical model



FIGURE 24 Velocity wave (original material point method)

³⁴⁴ WILEY

FIGURE 25 Velocity wave (material point method with generalized- α scheme)

FIGURE 26 Velocity wave in the 2D Gaussian explosion. MPM, material point method

This example simulates an elastic 2 x 2 m square and the explosion is located at point [1 m; 1 m]. The elastic square is discretized into 200 x 200 grid cells with 4 material points per square element. The size of the square grid cell is 0.01 m ($l_e = 0.01$ m). The elastic square has Young's modulus $E = 10e^7$ Pa and density $\rho = 1000$ kg/m³. The forces in Equation (56) were applied to the material points as external forces. To avoid any reflection from the boundary, the wave propagation is computed only for the $8e^{-3}$ s with the time step being as $\Delta t = 10e^{-4}$ s.

Figure 24 and Figure 25 show the snapshot of the velocity wave for the original MPM and MPM enriched with the generalized- α scheme at t = 8e⁻³ s. When the wave propagates in the elastic domain, it leaves behind some velocity oscillations near the explosive source. This noise is reduced when the simulation is repeated with the generalized- α scheme for the MPM formulation (see Figure 26 at t = 8e⁻³ s). In such impact problems, the generalized- α scheme can reduce considerable the noises, whereas null-space filter again does not change significantly the results, similarly as in the previous example.

6 | NUMERICAL EXAMPLES: APPLICATION OF THE NULL-SPACE FILTER

6.1 | 1D wave propagation

The following example is the wave propagation of an elastic bar due to the nonlinear predescribed displacement. The length of the bar is 2 m, Young's modulus is $E = 10e^6$ Pa, and the density $\rho = 1000$ kg/m³. Assuming the magnitude of displacement is A = 0.001m, the initial displacement and strain are generated as follows:

$$d(x, t = 0) = Ae^{-50(x-L)^2}$$
(60)

$$\varepsilon(x, t = 0) = -100 \operatorname{Axe}^{-50(x-L)^2}.$$
(61)



Figure 27 presents the schematic of the numerical model. Because each MPM variants require different boundary treatment, the initial condition (strain) was put in the middle of the bar to avoid possible errors associating with the boundary conditions. The initial strain induces 2 identical wave propagating in the opposite directions. Because of the symmetry condition, only half of the bar is considered. The dimension of the considered domain is L = 1 m.

The given initial condition leads to the analytical solution

$$d_{\text{exact}}(x,t) = 0.5A \left[e^{-50(x-L-ct)^2} + e^{-50(x-L+ct)^2} \right]$$
(62)

$$\varepsilon_{\text{exact}}(\mathbf{x}, \mathbf{t}) = -50 \mathbf{A} \left[(\mathbf{x} - \mathbf{L} - \mathbf{ct}) \, \mathrm{e}^{-50(\mathbf{x} - \mathbf{L} - \mathbf{ct})^2} + (\mathbf{x} - \mathbf{L} + \mathbf{ct}) \, \mathrm{e}^{-50(\mathbf{x} - \mathbf{L} + \mathbf{ct})^2} \right]. \tag{63}$$

The cell length is 0.01 m ($l_e = 0.01$ m). The time step is fixed at $\Delta t = 0.05 l_e/c = 5e^{-6}$ s. The final time of all numerical simulations is t = 0.005 s corresponding to 1000 time steps. With those parameters, the compression wave amplitude is small, the cell-crossing errors do not occur and the original MPM can solve this problem without any cell-crossing errors. To demonstrate the null-space oscillations, the bar is discretized into 200 grid cells with 2 material points per cell. It corresponds to a total of 201 nodes and 400 material points ($N_p = 400$). Because the gradient of the shape function in original MPM is constant in the cell, the rank of this mapping, rank(∇S), can be determined analytically and it is equal to number of grid cells (rank(∇S) = 200). Then, the null-space is null(∇S) = $N_p - \text{rank}(\nabla S) = 200$. Figure 28 and Figure 29 present the numerical model using PIC scheme and FLIP scheme (MPM), respectively. The PIC shows a diffusive solution, as reported by Hammerquist and Nairn,²¹ and both PIC and MPM show numerical noises in the middle of the bar due to the null-space errors. This noise can be filtered by using the null-space filter (see Figure 30 and Figure 31). In case there is only 1 material point per cell, leading to 200 material points in total ($N_p = 200$), the null-space errors in material points will be eliminated as $N_p = \text{rank}(\nabla S)$. Figure 32 demonstrates that there is no oscillation for 1 particle per cell because there are no null-space errors.

6.2 | Convergence rate study

In the example earlier, cell-crossing does not occur in the simulation. Therefore, the original MPM could solve the problem accurately. This section compares spatial convergence for MPM, GIMP, DDMP, and DDMP with null-space filter in two scenarios (no cell-crossing and cell-crossing as described in Figure 33). The same example as in the previous section, with

1.00

1.00

1.00



3 6 1 0 1 -1 -2 0.00 0.25 0.50 0.75 1.00 Position (m)

FIGURE 29 Strain wave in material point method (MPM) (2 material points per cell, no cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 30 Strain wave in particle-in-cell (PIC) scheme (2 material points per cell, no cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 31 Strain wave in material point method (MPM) with null-space filter (2 material points per cell, no cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 32 Strain wave in material point method (MPM) (1 material point per cell, no cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

³⁴⁶ WILEY





WILEY

347

FIGURE 33 Schematic of two cases (no cell-crossing and cell crossing)

FIGURE 34 Strain wave along the bar in material point method (MPM) (cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 35 Strain wave along the bar in generalized interpolation material point method (GIMP) (cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

the initial displacement amplitude A increased from 0.001 to 0.002 m, leads to particles crossing the grid cell boundaries. To minimize the time errors, the time step is constantly set to be $\Delta t = 10e^{-6}$ s corresponding to the CFL number of 0.05 for the finest grid density in this study. The final time of all numerical simulations is t = 0.005 s corresponding to 1000 time steps.

To compare between MPM, GIMP, and DDMP, the cell length of 0.004 m is selected as the reference case. It corresponds to 500 cells and 1000 material points in the entire 2-m bar. Cell-crossing causes significant errors in the right boundary of the bar when using MPM formulation (Figure 34). These errors increase dramatically with the increase of the grid cell number, leading to cell-crossing instability. This instability can be reduced by using GIMP or DDMP formulation (Figure 35 and Figure 36). However, in GIMP and DDMP, the strain still slightly oscillates in the middle of the bar and this oscillation can be reduced using the null-space filter (Figure 37 and Figure 38).

To study the convergence rate of the proposed formulation, the spatial error is defined as follows:

$$RMS = \log\left(\sqrt{\frac{\sum_{p=1}^{N_{p}} \|\varepsilon_{exact}\left(\mathbf{x},t\right) - \varepsilon\left(\mathbf{x},t\right)\|^{2}}{N_{p}}}\right),$$
(64)

where $\varepsilon_{exact}(\mathbf{x}, \mathbf{t})$ is the exact strain from Equation (63) and $\varepsilon(\mathbf{x}, \mathbf{t})$ is the strain of the numerical solution. N_p is the total number of material points. Figure 39 and Figure 40 show the spatial convergence rate of strain at the $\mathbf{t} = 0.005 \text{ s}$ for the



FIGURE 36 Strain wave along the bar in dual domain material point method (DDMP) (cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 37 Strain wave along the bar in dual domain material point method (DDMP) with null-space filter (cell-crossing) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 38 Strain oscillation in the middle of the bar. DDMP, dual domain material point method; GIMP, generalized interpolation material point method

"no cell-crossing" cases and the "cell-crossing" case, respectively. The MPM can converge in the "no cell-crossing" case but fail in the "cell-crossing" case. Although the cell-crossing errors is dominant, in both cases, the null-space filter can reduce the errors for GIMP and DDMP.

6.3 | Two-phase gravity loading and consolidation

In the previous section, the results of the simulation of wave propagation have indicated that the null-space filter can help in removing the numerical oscillations. This section shows that, in a two-phase quasi-static problem under large deformations, the null-space errors can grow significantly in time and result in a nonconvergence.

6.3.1 | Numerical parameters

The simulations of the gravity loading and large-strain consolidation use double-point MPM with DDMP interpolation. The calculation considers a 1D 1-m high fully saturated elastic soil column, discretized into 50 grid cells. Each cell contains 2 solid material points and 2 water material points. In total, 100 solid material points and 100 water material points are

348 WILEY



	Parameter	Symbol	Values
Model	Column height	H (m)	1
	Number of solid particles	N_{sp}	100
	Number of water particles	N_{wp}	100
	Particle interval	2l _p (m)	0.01
	Grid interval	l (m)	0.02
	Time step	$\Delta t(s)$	10^{-6}
Solid	Young's modulus	E _o (mPa)	10
	Solid density	$\rho_{s} (kg/m^{3})$	2143
	Initial Porosity	n _o	0.3
	Initial Permeability	$k_o (m/s)$	10^{-3}
Water	Density	$\rho_w (kg/m^3)$	1000
	Bulk modulus	K _w (mPa)	2200

FIGURE 39 Spatial convergence of wave propagation (no cell-crossing). DDMP, dual domain material point method; GIMP, generalized interpolation material point method; MPM, material point method

FIGURE 40 Spatial convergence of wave propagation (cell-crossing). DDMP, dual domain material point method; GIMP, generalized interpolation material point method; MPM, material point method

TABLE 1	Input	parameters	for the	geostatic	problem
	mput	parameters	ior the	geostatie	problem

distributed equally in the column. Soil is modeled as a linear elastic material with Young's modulus of 10 MPa, the solid density equal to 2143 kg/m³ and the initial porosity and permeability equal to 0.3 and 10^{-3} m/s, respectively. For the liquid phase, the water density is taken as 1000 kg/m³ and the bulk modulus is equal to 2.2 GPa. The time step is set to be $\Delta t = 10e^{-6}$ s, which satisfies the CFL condition and the permeability-dependent criterion.⁴¹ Table 1 summarizes the parameters of the numerical model.

6.3.2 | Numerical solutions of gravity loading

The first example simulates the application of the gravity loading. Initially, the pore pressure and the effective stress are equal to zero. Then, the gravity loading is applied immediately after the first time step. As the dragging force, proportional to gravity acceleration, acts as a natural damping to counter the dynamic effect to reach the steady state, the gravity does not need to increase gradually with time, even though no numerical damping is applied. The final time of the simulation is 2 seconds, which is high enough to reach the steady state solution.





for small deformation (gravity 10 m/s2). MPM, material point method [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 42 Pore water pressure using cubic B-spline for small deformation (gravity 10 m/s2). MPM, material point method [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 43 Pore water pressure using dual domain material point method (DDMP) for large deformation (gravity 1500 m/s2) [Colour figure can be viewed at wileyonlinelibrary.com]

For small strain, with the gravitational acceleration of 10 m/s², both DDMP (rank-deficient mapping) and B-spline (full rank mapping) replicate the analytical solution (see Figure 41 and Figure 42). To obtain large displacements, the gravity is increased to 1500 m/s², leading to approximately 10% of engineering strain. Figure 43 and Figure 44 present the numerical results of DDMP and cubic B-spline MPM without the null-space filter. Clearly, the pore water pressure oscillates under the large deformation without the filter, whereas the null-space filter corrects the water pressure so it just increases linearly with depth (see Figure 45).

6.3.3 | Analytical solutions of large strain oedometric consolidation

The second example is simulations of the large strain oedometric consolidation validated with an analytical solution given by Xie and Leo.⁴² In Terzaghi small strain consolidation theory, the soil thickness H, the permeability k, and Young's modulus of soil E assume to be constant. These assumptions are not valid for large strain consolidation, and therefore Xie and Leo⁴² considered the soil thickness H(t), permeability k_t , and modulus of soil E_t being time dependent for the derivation of the consolidation equations. The MPM naturally considers the change of soil thickness and updates the



FIGURE 44 Pore water pressure using cubic B-spline for large deformation (gravity 1500 m/s2). MPM, material point method [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 45 Pore water pressure using dual domain material point method (DDMP) with null-space filter for large deformation (gravity 1500 m/s2) [Colour figure can be viewed at wileyonlinelibrary.com]

porosity n every time step, therefore, we update the permeability and Young's modulus from porosity as follows:

$$\frac{\mathbf{k}_{\mathrm{t}}\left(\mathbf{n}\right)}{\mathbf{k}_{\mathrm{o}}} = \left(\frac{1-\mathbf{n}_{\mathrm{o}}}{1-\mathbf{n}}\right)^{2} \tag{65}$$

$$\frac{E_{t}(n)}{E_{o}} = \frac{1-n}{1-n_{o}},$$
(66)

where n_o , k_o , and E_o are the initial porosity, permeability, and modulus, respectively. The governing equations for the excess pore water pressure is

$$c_{v}\frac{\delta^{2}u}{\delta z^{2}}(H,t) = \frac{\delta u}{\delta t},$$
(67)

where c_v is the coefficient of consolidation. The coefficient of consolidation can be written as follows:

$$c_{\rm v} = \frac{k}{m_{\rm vl}\rho_{\rm w}},\tag{68}$$

where k is the permeability and $m_{\nu l}$ is the compressibility in the large strain related to void ratio as

$$m_{\rm vl} = -\frac{\delta \varepsilon_z}{\delta \sigma'} = -\frac{1}{1+e} \frac{\delta e}{\delta \sigma'}.$$
 (69)

The excess pore water pressure along the depth z = [0, H] can be written as follows:

$$u = \frac{1}{m_{vl}} \ln\left(1 + (\exp\left(m_{vl}q\right) - 1) \sum_{m=1}^{\infty} \frac{2}{M} \sin\left(\frac{Mz}{H}\right) \exp\left(-M^2 T_v\right)\right),\tag{70}$$

where q is the traction forces and the time factor is given by the following:

352

Wh fv

$$T_{v} = \frac{c_{v}t}{H}$$
(71)

$$M = (m - 0.5)\pi.$$
 (72)

The average degree of consolidation defined by strain (U_s) is given by the following:

$$U_{s} = 1 - \sum_{m=1}^{\infty} \frac{2}{M^{2}} \exp\left(-M^{2}T_{v}\right).$$
(73)

The settlement at the top surface (S_t) is given by the following:

$$S_{t} = H \left(1 - \exp\left(-m_{vl}q\right)\right) \left(1 - \sum_{m=1}^{2} \frac{2}{M^{2}} \exp\left(-M^{2}T_{v}\right)\right).$$
(74)

The final top surface settlement, when $U_s \rightarrow 1$, is equal to

$$S_{\infty} = H_0 (1 - \exp(-m_{vl}q)).$$
 (75)

Considering the motion of the soil skeleton (vs), the velocities of soil skeleton are

$$v_{s} = \frac{2c_{v}}{H} \left(1 - \exp\left(-m_{vl}q\right)\right) \sum_{m=1} \cos\left(\frac{Mz}{H}\right) \exp\left(-M^{2}T_{v}\right).$$
(76)

6.3.4 | Numerical solutions of large strain oedometric consolidation

In the numerical model, the compression force on top of the layer is equal to 2 GPa, which is enough to create approximately 20% of engineering strain. Therefore, initial pore pressure is set to 2 GPa, leading to a zero effective stress (no gravity is considered). The top boundary layer is drained, whereas the bottom boundary layer is undrained. Therefore, the water material points can freely move outside the solid domain through the top of the solid column. Indeed, it was observed during the analysis that some water particles flew over the top soil surface. The transition from a 2-phase porous medium to a single liquid phase led to oscillations in the top boundary layer. However, as there was no gravitational loading, these water particles were not important in the analysis. As such, the water particles that moved from soil body to a free water zone were simply removed, which reduced pressure and velocity oscillations. In the further research, the interface of free water-porous media transition will be investigated in greater detail. The numerical results (see Figure 46), obtained with DDMP using null-space filter, are compared with the analytical solution for the average degrees of consolidation defined by strain in the Equation (73), U_s = 0.4, 0.5, 0.6, 0.75, 0.95, and 1.0 respectively. The numerical solutions show a very good agreement with the analytical solution (see Equation (70)) and the calculated settlements (see Figure 47) are almost



FIGURE 46 Excess pore water pressure for large strain consolidation [Colour figure can be viewed at wileyonlinelibrary.com]



FIGURE 47 Settlement for large strain consolidation [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 48 Velocity for large strain consolidation [Colour figure can be viewed at wileyonlinelibrary.com]

identical as the analytical solution (see Equation (74)). The dynamic motion of the soil phase is captured in the numerical model and compared with the analytical model in Figure 48. It may be that the constant time step is somewhat too large at the beginning of the analysis under a high value of velocity, leading to some velocity oscillation. However, overall, the results show an acceptable correlation between the numerical and analytical solutions for approximately 20% of engineering strain.

6.3.5 | Influence of the interpolation in the null-space instability

Both DDMP and cubic B-spline MPM can replicate the small strain consolidation (see Figure 49 and Figure 50), but not for large strain consolidation. Figure 51, Figure 52, and Figure 53 give the comparison between the numerical model and the analytical solution for the average consolidation degree defined by strain of 0.5 for DDMP (rank-deficient mapping), MPM with cubic B-spline (full rank mapping), and DDMP with null-space filter, respectively. The result shows that null-space filter removes the pore water pressure instability and leads to a good agreement between the numerical and analytical solutions.



FIGURE 49 Excess pore water pressure using dual domain material point method (DDMP) (traction 1 Pa, 50 cells, 100 material points) [Colour figure can be viewed at wileyonlinelibrary.com]

0.0

-0.2



FIGURE 51 Excess pore water pressure using dual domain material point method (DDMP) (traction 2 GPa, 50 cells, 2 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 52 Excess pore water pressure using cubic B-spline (traction 2 GPa, 50 cells, 2 material points per cell). MPM, material point method [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 53 Excess pore water pressure using dual domain material point method (DDMP) with null-space filter (traction 2 GPa, 50 cells, 2 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]



6.3.6 | Influence of the grid cell size on the null-space instability

To investigate the influence of grid size, the consolidation problem is solved with three different grids, consisting of 20 cells, 50 cells, and 100 cells. Figure 54, Figure 51, and Figure 55 show the DDMP solution of the large strain consolidation with a different number of cells (20 cells, 50 cells, and 100 cells, respectively). The figures show that grid refinement reduces the magnitude of the oscillations but does not remove them fully. The null-space errors, however, can be mitigated significantly by using the null-space filter (see Figure 53, Figure 56, and Figure 57). Note that the slight discrepancy between numerical and analytical solution in Figure 57 may be influenced by the removal of the liquid particles when moving outside the solid domain.



FIGURE 54 Excess pore water pressure using dual domain material point method (DDMP)(traction 2 GPa, 20 cells, 2 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 55 Excess pore water pressure using dual domain material point method (DDMP) (traction 2 GPa, 100 cells, 2 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 56 Excess pore water pressure using dual domain material point method (DDMP) with null-space filter (traction 2 GPa, 20 cells, 2 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]





FIGURE 57 Excess pore water pressure using dual domain material point method (DDMP) with null-space filter (traction 2 GPa, 100 cells, 2 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]

6.3.7 | Influence of the material points per cell in the null-space instability

Next, the problem is solved with 2, 3, and 4 material points per grid cell, with the problem discretized with the total of 20 cells. Figure 54, Figure 58, and Figure 59 show the DDMP solution of the large strain consolidation with a different number of material points per cell (2 material points per cell, 3 material points per cell, and 4 material points per cell, respectively). Increasing the number of material points per cell does not reduce the oscillations and generates more null modes, which is also observed in the PIC method.³ The null-space errors, again, can be mitigated significantly by using the null-space filter (see Figure 56, Figure 60, and Figure 61).

In calculations without the null-space filter, the null-space errors accumulate rapidly in time and lead to a nonconvergent solution (the simulation collapse when pore water pressure approaches infinity). During the large deformation consolidation, the grid remains undeformed, but the number of nodes involved in the calculation reduces in time (see Figure 62). Therefore, the number of the column spaces reduces and the number of the null-space increases, which may be the reasons for the rapid increase of the null-space errors.

-0.2 Depth (m) -0.4 -0.6 Analytical solution -0.8 DDMP -1.0 2.0 0.0 0.5 1.0 1.5 Excess pore water pressure (GPa) 0.0 -0.2 -0.4 Depth (m) -0.6 Analytical solutio -0.8 DDMP -1.0 1.5 2.0

FIGURE 58 Excess pore water pressure using dual domain material point method (DDMP) (traction 2 GPa, 20 cells, 3 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]

figure can be viewed at wileyonlinelibrary.com]



0.0



FIGURE 60 Excess pore water pressure using dual domain material point method (DDMP) with null-space filter (traction 2 GPa, 20 cells, 3 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 61 Excess pore water pressure using dual domain material point method (DDMP) with null-space filter (traction 2 GPa, 20 cells, 4 material points per cell) [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 62 De-rank of degrees of freedom (from 10 nodes to 8 nodes) [Colour figure can be viewed at wileyonlinelibrary.com]

7 | CONCLUSION

This paper shows how to implement the explicit generalized- α integration scheme in the MPM. The generalized- α integration scheme suppresses the high-frequency spurious oscillations and controls the dissipation with just a single parameter. Numerical examples show that the scheme can reduce the oscillations in the impact and wave propagation problems for MPM formulation. Furthermore, this paper gives a null-space filter algorithm, which improves the simulation accuracy and stability. The algorithm is suitable for different interpolations, including original MPM, GIMP, or DDMP. The proposed algorithm is much less expensive, when compared with the original method,²³ which used the SVD or QR decomposition. Several simulations demonstrated the improvement in stability in calculations with the null-space filter, for both single phase and two-phase material.

Apart from the null-space related errors, MPM simulation results are also affected by the cell-crossing errors and quadrature errors. While the former occurs when a material point crosses to a new cell, the latter results from the first-order integration of the field quantities,⁴³ leading to a reduction of the convergence rate in the numerical solution. The null-space error, on the other hand, occurs due to a null-space existing due to the linear mapping between material point and grid nodes. It explains why, in the vibration example (no cell-crossing), null-space error does not exist when there is only 1 particle per cell. Moreover, in the quasi-static hydro-coupling problems, the null-space errors increase

358 WILEY

dramatically because the number of nodes involved in the calculation reduces in time, leading to the expansion of the null-space.

ACKNOWLEDGEMENTS

This work has been funded by the Academy of Finland under the project "Progressive failure and post-failure modeling of slopes with Generalized Interpolation Material Point Method (GIMP)" under decision number 286628. The authors thank the reviewers for their constructive comments to improve the paper.

ORCID

Quoc-Anh Tran^D https://orcid.org/0000-0001-5987-435X

REFERENCES

- 1. Sulsky D, Chen Z, Schreyer HL. A particle method for history-dependent materials. *Comput Methods Appl Mech Eng.* 1994; 118(1-2):179-196.
- 2. Harlow FH. The particle-in-cell computing method in fluid dynamics. Methods Comput Phys. 1964;3:319-343.
- 3. Brackbill JU. The ringing instability in particle-in-cell calculations of low-speed flow. J Comput Phys. 1988;75(2):469-492.
- 4. Sołowski W, Sloan S. Evaluation of material point method for use in geotechnics. Int J Numer Anal Methods Geomech. 2015;39:685-701.
- 5. Tran QA, Sołowski W, Karstunnen M, Korkiala-Tanttu L. Modelling of fall-cone tests with strain-rate effects. *Procedia Engineering*. 2017;175:293-301.
- Tran QA, Sołowski W, Thakur V, Karstunnen M. Modelling of the quickness test of sensitive clays using the generalized interpolation material point method. In: *Landslides in Sensitive Clays: From Research to Implementation*. Cham, Switzerland: Springer International Publishing AG; 2017:323-336.
- 7. Tran QA, Sołowski W. Generalized interpolation material point method modelling of large deformation problems including strain-rate effects–application to penetration and progressive failure problems. *Comput Geotech*. 2019;106:249-265.
- 8. Bardenhagen SG, Kober EM. The generalized interpolation material point method. *Comput Model Eng Sci.* 2004;5:477-496.
- 9. Zhang DZ, Ma X, Giguere P. Material point method enhanced by modified gradient of shape function. J Comput Phys. 2011;230:6379-6398.
- 10. Steffen M, Wallstedt PC, Guilkey JE, Kirby RM, Berzins M. Examination and analysis of implementation choices within the material point method (MPM). *Comput Model Eng Sci.* 2008;32(2):107-127.
- 11. Tielen R, Wobbes E, Möller M, Beuth L. A high order material point method. Procedia Engineering, 2017;175:265-272.
- 12. 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.* 2017;113(3):411-431.
- Tran QA, Wobbes E, Sołowski W, Möller M, Vuik C. Moving least squares reconstruction for B-spline material point method. In: Proceedings of the 2nd International Conference on the Material Point Method for Modelling Soil-Water-Structure Interaction (MPM); 2019; Cambridge UK.
- 14. Wang B, Vardon PJ, Hicks MA, Chen Z. Development of an implicit material point method for geotechnical applications. *Comput Geotech*. 2016;71:159-167.
- 15. Yerro A, Alonso E, Pinyol N. Modelling large deformation problems in unsaturated soils. Paper presented at: 3rd European Conference on Unsaturated Soils; 2016; Paris, France.
- 16. Ceccato F, Beuth L, Vermeer P, Simonini P. Two-phase material point method applied to the study of cone penetration. *Comput Geotech*. 2016;80:440-452.
- 17. Lu M, Zhang J, Zhang H, Zheng Y, Chen Z. Time-discontinuous material point method for transient problems. *Comput Methods Appl Mech Eng.* 2018;328:663-685.
- 18. Renaud A, Heuzé T, Stainier L. A discontinuous Galerkin material point method for the solution of impact problems in solid dynamics. *J Comput Phys.* 2018;369:80-102.
- 19. Jiang C, Schroeder C, Selle A, Teran J, Stomakhin A. The affine particle-in-cell method. ACM Trans Graph. 2015;34(4). Article No. 51.
- 20. Jiang C, Schroeder C, Teran J. An angular momentum conserving affine-particle-in-cell method. J Comput Phys. 2017;338:137-164.
- 21. 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.
- 22. Hulbert GM, Chung J. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Comput Methods Appl Mech Eng.* 1996;137:175-188.
- 23. Gritton C, Berzins M, Kirby R. Improving accuracy in particle methods using null spaces and filters. In: Proceedings of the IV International Conference on Particle-based Methods. Fundamentals and Applications; 2015; Barcelona, Spain.
- 24. Tran QA, Sołowski W, Cummings E. Hydro-mechanical coupled dual domain material point method stabilized with a null-space filter. Paper presented at: V International Conference on Particle-based Methods. Fundamentals and Applications; 2017; Hannover, Germany.
- 25. Gritton C, Berzins M. Improving accuracy in the MPM method using a null space filter. Comput. Part Mech. 2016;3(12):1-12.
- 26. Brackbill JU, Kothe DB, Ruppel HM. FLIP: a low-dissipation, particle-in-cell method for fluid flow. *Comput Phys Commun.* 1988;48(1):25-38.

- 28. Nairn J. Numerical simulation of orthogonal cutting using material point method. Eng Fract Mech. 2015;149:262-275.
- 29. Hilber HM, Hughes TJR. Collocation, dissipation and [overshoot] for time integration schemes in structural dynamics. *Earthq Eng Struct Dyn.* 1978;6:99-117.
- 30. Kontoe S. Development of Time Integration Schemes and Advanced Boundary Conditions for Dynamic Geotechnical Analysis [PhD thesis]. London, UK: Imperial College London; 2006.
- Tsaparli V, Kontoe S, Taborda DMG, Potts DM. The importance of accurate time-integration in the numerical modelling of P-wave propagation. *Comput Geotech*. 2017;86:203-208.
- 32. Kontoe S, Zdravkovic L, Potts DM. An assessment of time integration schemes for dynamic geotechnical problem. *Comput Geotech*. 2008;35:253-264.
- 33. Chung J, Hulbert GM. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-*α* method. *J Appl Mech*. 1993;60:371-375.
- 34. Trefethen LN, Bau I. Numerical Linear Algebra. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997.
- 35. Hughes TJR. The Finite Element Method Linear Static and Dynamic Finite Element Analysis. Englewood Cliffs, NJ: Prentice-Hall; 1987.
- 36. Bandara S, Soga K. Coupling of soil deformation and pore fluid flow using material point method. Comput Geotech. 2015;63:199-214.
- 37. Tran QA, Sołowski W. Large strain consolidation modelling using Dual domain material point method. Paper presented at: The 15th International Conference of the International Association for Computer Methods and Advances in Geomechanics; 2017; Wuhan, China.
- Chen Z, Hu W, Shen L, Xin X, Brannon R. An evaluation of the MPM for simulating dynamic failure with damage diffusion. *Eng Fract Mech.* 2002;69:1873-1890.
- 39. Collino F, Tsogka C. Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *Geophysics*. 2001;66(1):294-307.
- 40. Yue B, Guddati MN. Dispersion-reducing finite elements for transient acoustics. J Acoust Soc Am. 2005;118:2132-2141.
- 41. Mieremet M, Stolle D, Ceccato F, Vuik C. Numerical stability for modelling of dynamic two-phase interaction. *Int J Numer Anal Methods Geomech.* 2016;40:1284-1294.
- 42. Xie KH, Leo CJ. Analytical solutions of one-dimensional large strain consolidation of saturated and homogeneous clays. *Comput Geotech*. 2004;31:301-314.
- 43. Steffen M, Kirby RM, Berzins M. Analysis and reduction of quadrature errors in the material point method (MPM). *Int J Numer Methods Eng.* 2008;76:922-948.

How to cite this article: Tran Q-A, Sołowski W. Temporal and null-space filter for the material point method. *Int J Numer Methods Eng.* 2019;120:328–360. https://doi.org/10.1002/nme.6138

APPENDIX

The function α , for DDMP α_1 , can be written as follows:

$$\alpha_{1} \left(\mathbf{x} \right) = 0.5 \left[\prod_{i=1}^{n_{i}} n_{i} S_{i} \left(\mathbf{x} \right) \right]^{\frac{3}{2(n_{i}-1)n_{dim}}},$$
(A1)

where n_i is the number of the nodes of the cell, which stores the material point (**x**). $S_i(\mathbf{x})$ is the linear basis shape function of the node "i" and $n_{dim} = 1, 2, 3$ is the number of the dimension of the problem. one dimension, the function α can also retrieve the gradient shape function of MPM with $\alpha = 1$ or of cpGIMP with α_2 set as follows:

$$\alpha_{2}\left(x\right) = \begin{cases} \left(1 - \frac{2|\Delta x|}{l_{p}}\right) \frac{\Delta x}{\Delta x - 2sign(\Delta x)|\Delta x|}, & |\Delta x| \leq l_{p} \\ \left(1 + \frac{|\Delta x|}{l_{p}}\right) \frac{\left(\Delta x - sign(\Delta x)L_{i}\right)}{\left(\Delta x - sign(\Delta x)\left(L_{i} + |\Delta x|\right)\right)}, & L_{i} - l_{p} < |\Delta x| \leq L_{i} \\ 1 - \frac{|\Delta x|}{l_{p}} \cdot \frac{\left(\Delta x - sign(\Delta x)\left(L_{i} + l_{p}\right)\right)}{\left(\Delta x - sign(\Delta x)\left(L_{i} + |\Delta x|\right)\right)}, & L_{i} < |\Delta x| \leq L_{i} + l_{p} \\ 1 & , \text{ else,} \end{cases}$$

$$(A2)$$

where the gradient of the shape function in cpGIMP depends on the interval length $\Delta x = x - x_i$, cell spacing L_i, and the current material point's domain 2l_p, which evolves with the deformation of the material point, corresponding to cpGIMP. Figure A1 shows function α for both GIMP and DDMP.

360 WILEY



FIGURE A1 Weight functions for generalized interpolation material point method (GIMP) and dual domain material point method (DDMP)

LIST OF SYMBOLS

	Ger	neral notations		Μ	aterial point (MP) variables
Variable	Туре	Meaning	Variable	Туре	Meaning
ρ	Scalar	Density	x _p	Vector	MP position
X	Vector	Position	m _p	Scalar	MP mass
u	Vector	Displacement	\mathbf{v}_{p}	Vector	MP velocity
v	Vector	Velocity	b _p	Vector	MP body acceleration
a	Vector	Acceleration	$\sigma_{\rm p}$	Matrix	MP stress
τ	Vector	Traction forces	Vp	Scalar	MP volumn
σ	Matrix	Cauchy stress	$\nabla \mathbf{v}_{\mathrm{p}}$	Vector	MP gradient velociy
b	Vector	Body acceleration	\mathbf{F}_{p}	Matrix	MP deformation gradient
\mathbf{f}^{int}	Vector	Internal force	ρ_p	Scalar	MP density
\mathbf{f}^{b}	Vector	Body force	lp	Scalar	Half of MP size
\mathbf{f}^{ext}	Vector	External force	Interpolati	on varia	bles
n	Scalar	Porosity	α	Scalar	Bound function
k	Scalar	Permeability	S_{ip}	Scalar	Nodal shape function
\mathbf{f}_{d}	Vector	Dragging force	∇S_{ip}	Vector	Nodal shape function gradient
g	Scalar	Gravity acceleration	$\nabla \mathbf{S}$	Matrix	Mapping matrix
с	Scalar	Elastic wave velocity	Np	Scalar	Total number of material points
Е	Scalar	Young modulus	N _n	Scalar	Total number of nodes
K _w	Scalar	Bulk modulus of liquid	r	Scalar	Rank of mapping matrix
c _v	Scalar	consolidation coefficient	R	Matrix	Upper triangular matrix
m _{vl}	Scalar	Compressibility	Q	Matrix	Unitary matrix
q	Scalar	Traction force	Q*	Matrix	Unitary matrix in column space
St	Scalar	Settlement	\mathbf{q}_{i}	Vector	Orthonormal vector
T _v	Scalar	Time factor	$\nabla \mathbf{S}_{c}$	Vector	Local mapping maitrx
Us	Scalar	Consolidation defined by strain	n _i	Scalar	Number of nodes in the cell
Nodal varia	bles		Time integ	ration va	ariables
mi	Scalar	Nodal mass	α _m , γ, β	Scalar	Time integration parameters
\mathbf{v}_{i}	Vector	Nodal velocity	dt, Δt	Scalar	Time step
\mathbf{q}_{i}	Vector	Nodal momentum	ω	Scalar	Natural frequeny
\mathbf{a}_{i}	Vector	Nodal acceleration	ρ_b	Scalar	Secptral radius at bifurcation limit
\mathbf{f}_{i}^{int}	Vector	Nodal internal force	$\Omega_{ m b}$	Scalar	Bifucation limit
\mathbf{f}_{i}^{b}	Vector	Nodal body force	$\Omega_{\rm s}$	Scalar	Critical limit
\mathbf{f}_{i}^{ext}	Vector	Nodal external force	$\nabla \mathbf{v}_{\mathbf{p}}^{*}$	Vector	Filter value of particle-based mapping function
\mathbf{f}_{i}^{t}	Vector	Nodal total force	$\nabla \mathbf{v}_{p}$	Vector	Value of node-based mapping function
Li	Scalar	Cell size			
σ_{i}	Matrix	Nodal stress			
Vi	Scalar	Nodal volume			