Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 352 (2019) 85–109

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

An efficient staggered grid material point method[☆]

Yong Liang, Xiong Zhang[*], Yan Liu

School of Aerospace Engineering, Tsinghua University, Beijing 100084, PR China

Received 27 October 2018; received in revised form 31 March 2019; accepted 19 April 2019
Available online 26 April 2019

Highlights

- The SGMP is proposed to eliminate the cell crossing noise in the MPM efficiently.
- The cell center quantities are reconstructed efficiently based on an auxiliary grid.
- A continuous strain rate/vorticity field is established based on the auxiliary grid.

Abstract

The material point method (MPM) has demonstrated itself as an effective numerical method to simulate extreme events with large deformations. However, the original MPM suffers the cell crossing noise because it takes the material points as integration points and employs the piecewise linear grid nodal shape functions whose gradient is discontinuous on the cell boundary. A number of techniques have been developed to alleviate the cell crossing noise. In this paper, a new staggered grid material point method (SGMP) is proposed to eliminate the cell crossing noise very efficiently. The volume integrals in the weak form are evaluated by cell center quadrature instead of particle quadrature as the sum of value of the integrand at each cell center of the background grid multiplied by the corresponding quadrature weight. The physical quantities and the quadrature weights at the cell centers are reconstructed efficiently based on an auxiliary grid, which is obtained by shifting the background grid half the side length of its cell in each direction. Similar to the original MPM, both grids carry no permanent information and can be reset after each time step. In addition, the SGMP evaluates the constitutive equations at the particles, just like the original MPM, to readily model the history-dependent materials. To further reduce the cell crossing noise, a continuous strain rate/vorticity field is established based on the auxiliary grid, whose values are determined by the background grid velocity gradient. The strain rate/vorticity at each particle is interpolated from the auxiliary grid nodal values. Due to the overlap of the cell centers and the corresponding auxiliary grid nodes, a very efficient implementation is established in the SGMP. Numerical studies illustrate that the SGMP is capable of eliminating the cell crossing noise with little extra computational effort and the extra cost ratio reduces as the number of the grid cells or the particles increases.

Keywords: Material point method; Staggered grid; Cell crossing noise; Large deformation

[☆] Supported by the National Natural Science Foundation of China (11672154) and Science Challenge Project (TZ2018002).

[*] Corresponding author.

E-mail address: xzhang@tsinghua.edu.cn (X. Zhang).

https://doi.org/10.1016/j.cma.2019.04.024

1. Introduction

Numerical simulation has been playing an increasingly important role in modern science and engineering [1]. Following theoretical analysis and experimental methods, numerical methods have become another powerful tool for solving practical engineering problems [2,3]. Quite a number of numerical methods have been developed to tackle different kinds of problems. According to the way how the object is described, the methods can be mainly classified into Lagrangian [4], Eulerian [5] and hybrid methods [6,7]. Both the Lagrangian and Eulerian methods have the significant advantages and disadvantages for large deformation problems, so that the hybrid method that combines the advantages of both Lagrangian and Eulerian methods is developed for such problems. One representative is the particle-in-cell [8,9] (PIC) method, which discretizes the fluid into a set of Lagrangian particles and uses a uniform Eulerian mesh as the computational grid. Based on the PIC, various variations have been developed [10–12].

The material point method (MPM) is an extension of the FLIP PIC method from fluid mechanics to solid mechanics by Sulsky et al. [7,13]. The MPM discretizes the material domain by a set of particles, each of which represents a material volume carrying all its state variables including mass, momentum, stress, etc. Besides, a predefined Eulerian background grid is used to solve the equations of momentum, and the grid nodal solution is mapped back to the particles to update their state variables. The background grid carries no permanent information and can be discarded at the end of each time step, so that the mesh distortion is avoided for problems with large deformation. The advantages make the MPM be applied to various kinds of extreme events such as hyper velocity impact [14–17], penetration [18–20], explosion [21,22], fracture evolution [23–26], incompressible flow [27,28], fluid–structure interaction [29–31], multiphase flow [32], etc.

The linear shape function used in the conventional MPM has a discontinuous gradient on the cell boundary, which leads to the so called cell crossing noise while the particle crosses the cell boundary [33]. As a result, the numerical solution loses its accuracy and becomes unphysical. Bardenhagen et al. proposed a generalized interpolation material point (GIMP) method [33] in the Petrov–Galerkin framework to suppress the numerical noise. The GIMP introduces a characteristic function to represent the spatial volume of the material point domain, which leads to a C^1 continuous shape function to map the information between the particles and the grid nodes and calculate the nodal internal force. Due to the continuity of the gradient of the GIMP shape function on the cell boundary, the cell crossing noise is reduced and the solution accuracy is improved significantly. According to the characteristic function used, the GIMP has several variants. The original MPM can be viewed as a special case of the GIMP by taking the Dirac delta function as the characteristic function. In the contiguous particle GIMP (cpGIMP), the particle characteristic function has a contiguous and non-overlapping support. Therefore, the cpGIMP requires tracking the particle shape, which is difficult and has heavy computing burden for the three-dimensions. To avoid the difficulties, several approximations have been proposed. For example, the particle shape can be assumed to be rectangular or cuboid during the simulation [34]. The simplest one is to assume the particle shape remains unchanged, which is known as the uniform GIMP [33,35] (uGIMP). In the uGIMP, the particle characteristic function is not a partition of unity, as the particle shape overlaps or leaves gaps. The GIMP has been used in several studies [36–38].

Besides, Sadeghirad et al. proposed a convected particle domain interpolation (CPDI) method [39], in which the particle domain is always parallelogram and an alternative grid nodal shape function is employed for the computational efficiency. The particle domain in the current configuration is obtained by the particle initial domain multiplied by the deformation gradient, which means that the overlaps and gaps still exist in the CPDI. The enhanced CPDI [40] is further proposed by Sadeghirad et al. to track the particle domains by updating the corner position of the particle. The particle shape in the enhanced CPDI is hexahedrons for three-dimensions and the overlaps and gaps are avoided. Zhang [41] et al. modified the gradient of the shape function to make it continuous on the cell boundary, so that the cell crossing noise is eliminated. The method named as the dual domain material point method (DDMP) does not require the particle shape. Steffen [42] et al. employed the smoother grid basis function including the quadratic and cubic B-spline in the MPM framework and demonstrated that the strategy could decrease the integration errors caused by cell crossing. Kafaji [43] employed a mixed integration in the MPM to reduce the cell crossing noise. The Gauss integration was used in the fully filled elements, whereas the particle integration was adopted in the partially filled elements. Sulsky and Gong proposed an improved material point method (IMPM) [44,45] by using the Gauss quadrature to evaluate the integrals in the weak form and employing the MLS to reconstruct the velocity, density and stress. The IMPM improved the convergence of the MPM significantly for the problems whose boundary is aligned with the background grid boundary. Tielen [46] et al. recently employed the two-point Gauss quadrature and the function reconstruction technique in the B-spline

MPM. The one dimensional numeral tests showed the cell crossing noise is reduced. Gan et al. [47] investigated the accuracy and convergence of the B-spline MPM schemes using quadratic, cubic, and quartic B-spline basis functions, respectively, and a Gauss quadrature method is proposed with respect to the particle domain for reducing the quadrature error. All these methods are able to reduce the cell crossing noise, but few of them examined the extra cost quantitatively.

The aim of the present work is to propose an efficient computational framework, which can eliminate the cell crossing noise and retains the high efficiency of the conventional MPM. In the proposed method, an auxiliary grid is employed, which is obtained by shifting the background grid by half the side length of its cell in each direction. Thus, the proposed method is named as the staggered grid material point (SGMP) method. Instead of using particle quadrature as the MPM, the volume integrals in the weak form are evaluated by cell center quadrature. The physical quantities and quadrature weights at cell centers are reconstructed by the Shepard approximation based on the auxiliary grid. To further reduce the cell crossing noise, a continuous strain rate/vorticity field is established based on the auxiliary grid with its grid nodal strain/vorticity rate obtained from the background grid. The particle strain rate/vorticity is interpolated from the auxiliary grid nodal values. Numerical studies show that the SGMP is capable of eliminating the cell crossing noise with a little extra computational effort.

The rest of this paper is organized as follows: First, the MPM scheme and the cell crossing noise are briefly reviewed in Section 2, while the proposed SGMP method is described in detail in Section 3. The detailed computer implementation is presented in Section 4, and several numerical tests are carried out in Section 5 to verify and validate the SGMP method. Finally, some concluding remarks are drawn in Section 6.

2. Material point method

The weak form equivalent to the momentum equation and the traction boundary equation in the updated Lagrangian formulation is given by

$$\int_{\Omega} \rho \ddot{u}_i \delta u_i \mathrm{d}\Omega + \int_{\Omega} \sigma_{ij} \delta u_{i,j} \mathrm{d}\Omega - \int_{\Omega} \rho b_i \delta u_i \mathrm{d}\Omega - \int_{\Gamma_t} \bar{t}_i \delta u_i \mathrm{d}\Gamma = 0 \tag{1}$$

where Γ_t denotes the traction boundary of the material domain Ω, ρ is the current density, the subscripts i and j indicate the components of the spatial variables following the Einstein convention, u_i is the displacement, σ_{ij} is the Cauchy stress, b_i is the body force per unit mass, $\bar{t}_i$ is the traction.

2.1. Standard MPM

In the MPM, the material domain Ω is discretized as a set of particles moving through an Eulerian background grid, as shown in Fig. 1. Therefore, the displacement field $u_i(\boldsymbol{x})$ can be approximated using the background grid as

$$u_i(\boldsymbol{x}) = \sum_{I=1}^{n_g} N_I(\boldsymbol{x}) u_{iI} \tag{2}$$

where the subscript I denotes the variables associated with the grid node I, n_g is the total number of background grid nodes, and $N_I(\boldsymbol{x})$ is the global shape function of node I evaluated at the position $\boldsymbol{x}$ as used in the FEM.

Substituting the displacement approximation Eq. (2) into the weak form Eq. (1) and invoking the arbitrariness of the virtual displacement δu_{iI} lead to the grid nodal momentum equation

$$\dot{p}_{iI} = f_{iI}^{\text{int}} + f_{iI}^{\text{ext}}, \quad \forall I \notin \Gamma_u \tag{3}$$

where Γ_u denotes the displacement boundary of the material domain,

$$p_{iI} = m_I \dot{u}_{iI} \tag{4}$$

is the grid nodal momentum,

$$m_I = \int_{\Omega} \rho N_I(\boldsymbol{x}) \mathrm{d}\Omega \tag{5}$$

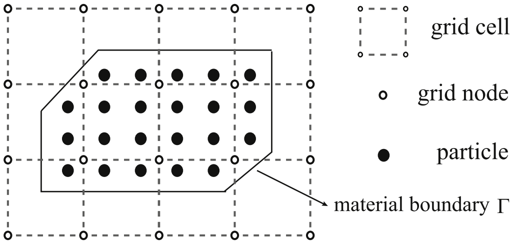


Fig. 1. Typical MPM discretization.

is the lumped grid nodal mass by row summation to improve the computational efficiency,

$$f_{iI}^{\text{int}} = -\int_{\Omega} N_{I,j}\left(\boldsymbol{x}\right)\sigma_{ij}\mathrm{d}\Omega \tag{6}$$

is the internal nodal force,

$$f_{iI}^{\text{ext}} = \int_{\Omega} \rho b_i N_I\left(\boldsymbol{x}\right)\mathrm{d}\Omega + \int_{\Gamma_t} N_I\left(\boldsymbol{x}\right)\bar{t}_i \mathrm{d}\Gamma \tag{7}$$

is the external nodal force.

In the MPM, the continuum Ω is discretized with a set of particles, each of which represents a material volume Ω_p and carries all its information such as mass, momentum, stress, etc. Thus, the integrals in Eqs. (5)–(7) can be evaluated as

$$m_I = \sum_{p=1}^{n_\mathrm{p}} N_{Ip} m_p \tag{8}$$

$$f_{iI}^{\text{int}} = -\sum_{p=1}^{n_\mathrm{p}} V_p N_{Ip,j}\sigma_{ijp} \tag{9}$$

$$f_{iI}^{\text{ext}} = \sum_{p=1}^{n_\mathrm{p}} m_p N_{Ip} b_{ip} \tag{10}$$

where the subscript p denotes the variables associated with particle p, $m_p = \rho_p V_p$, V_p, σ_{ijp} and b_{ip} are the mass, volume, stress and body force per unit mass of particle p, respectively; $N_{Ip} = N_I\left(\boldsymbol{x}_p\right)$ is the shape function of node I evaluated at the position of particle p. In Eq. (10), the traction term is omitted for simplicity and the imposition of the Neumann boundary conditions could be referred to the literature [48,49].

2.2. Cell crossing noise

Taking one dimension for example, the linear shape function used in the standard MPM is given as

$$N_I\left(\xi\right) = \frac{1}{2}\left(1+\xi_I\xi\right) \tag{11}$$

whose gradient ∇N_I is discontinuous between cells, as shown in Fig. 2.

Suppose the particle p crosses the cell boundary in the time step $[t^n, t^{n+1}]$, as shown in Fig. 2. It leads to a non-physical difference in the internal forces. At time t^n, the internal nodal forces are given as

$$f_{I-1}^{\text{int},n} = \frac{V_p\sigma_p}{L} \qquad f_{I}^{\text{int},n} = -\frac{V_p\sigma_p}{L} \qquad f_{I+1}^{\text{int},n} = 0 \tag{12}$$

while at time t^{n+1} they become

$$f_{I-1}^{\text{int},n+1} = 0 \qquad f_{I}^{\text{int},n+1} = \frac{V_p\sigma_p}{L} \qquad f_{I+1}^{\text{int},n+1} = -\frac{V_p\sigma_p}{L} \tag{13}$$

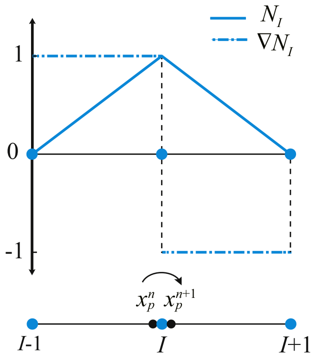


Fig. 2. The linear shape function and its gradient with the assumption of $L = 1$.

where L is the length of the cell. All the internal nodal forces have a non-physical jump while the particle crosses the cell boundary, which would produce numerical noise, called cell crossing noise. It leads to an unsatisfactory and even unphysical results.

Besides, the strain rate and vorticity used to update the particle stress are also discontinuous on the cell boundary due to the discontinuous gradient of the linear shape function. In the GIMP, the C^1 continuous shape function is employed. Therefore, the jump in both the internal force and the strain rate/vorticity are avoided.

3. Staggered grid material point method

The cell crossing noise is caused by the cell crossing of the particles (integration points) because the gradient of the shape function is discontinuous between cells. Hence, the cell crossing noise could be eliminated either by employing a shape function with continuous gradient between cells, as that in the GIMP and B-spline MPM, or by employing an integration scheme whose integration points do not move across cells. In this paper, a cell center quadrature is proposed to efficiently eliminate the cell crossing noise.

3.1. Discretization of the momentum equation

To eliminate the cell crossing noise, the volume integrals in Eqs. (5)–(7) can be evaluated as the sum of value of the integrand at each cell center multiplied by the quadrature weight of the cell, namely,

$$m_I = \sum_{c=1}^{n_c} N_{Ic} m_c \tag{14}$$

$$f_{iI}^{\text{int}} = -\sum_{c=1}^{n_c} V_c N_{Ic,j} \sigma_{ijc} \tag{15}$$

$$f_{iI}^{\text{ext}} = \sum_{c=1}^{n_c} m_c N_{Ic} b_{ic} \tag{16}$$

where the subscript c denotes the variables associated with cell c, n_c is the total number of background grid cells, $m_c = \rho_c V_c$ is the cell mass, V_c is the cell quadrature weight, which represents the cell volume determined by the particle volume, and $N_{Ic} = N_I(\boldsymbol{x}_c)$ is the shape function of grid node I evaluated at cell center $\boldsymbol{x}_c$. For three dimensional problems, N_{Ic} always equals $1/8$ because $\xi_c = \eta_c = \zeta_c = 0$.

3.2. Reconstruction of the physical quantities at cell centers

Similar to the MPM, all state variables are also carried by the particles in the SGMP. Thus, the values of physical quantities at the cell centers in Eqs. (14)–(16) including m_c, $\boldsymbol{\sigma}_c V_c$ and $m_c \boldsymbol{b}_c$ need to be reconstructed from those of the particles. The moving least squares (MLS), which is commonly used in meshfree methods, can be employed

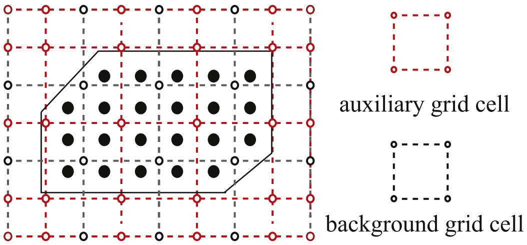


Fig. 3. The background grid and the auxiliary grid in the SGMP.

to reconstruct these quantities at the cell centers. However, to avoid the time consuming neighbor search and make the proposed SGMP as efficient as the conventional MPM, a cell based reconstruction scheme is employed in this paper. An auxiliary grid is defined for the cell center quantities reconstruction by shifting the background grid half the side length of its cell in each direction and resetting the position of the outermost auxiliary grid nodes to match the boundary of the background grid, as shown in Fig. 3. For clarity, the term "grid" just represents the conventional background grid while the term "auxiliary grid" represents the newly generated auxiliary grid in the following.

The cell centers of the background grid are the auxiliary grid nodes, so that the cell center quantities can be reconstructed in the auxiliary grid in the same way as the grid node quantities reconstruction in the background grid used in the conventional MPM. Namely, a physical quantity φ can be reconstructed in the auxiliary grid as

$$\varphi_c = \sum_{p=1}^{n_\mathrm{p}} N_{cp}\varphi_p \tag{17}$$

where φ_c is the value of φ at cell center c, φ_p is the value of φ at particle p, and N_{cp} is the nodal shape function N_c of the auxiliary grid node c evaluated at particle p. Therefore, the cell center quantities m_c, $\boldsymbol{\sigma}_c V_c$ and $m_c \boldsymbol{b}_c$ used in Eqs. (14)–(16) can be reconstructed in the auxiliary grid as

$$m_c = \sum_{p=1}^{n_\mathrm{p}} N_{cp} m_p \tag{18}$$

$$\boldsymbol{\sigma}_c V_c = \sum_{p=1}^{n_\mathrm{p}} N_{cp} \boldsymbol{\sigma}_p V_p \tag{19}$$

$$m_c \boldsymbol{b}_c = \sum_{p=1}^{n_\mathrm{p}} N_{cp} m_p \boldsymbol{b}_p \tag{20}$$

Thus, the grid nodal mass m_I can be obtained by Eq. (14). Similarly, the grid nodal momentum $\boldsymbol{p}_I$ can be given as

$$\boldsymbol{p}_I = \sum_{c=1}^{n_\mathrm{c}} N_{Ic} \boldsymbol{p}_c \tag{21}$$

3.3. Correction of the auxiliary grid nodal shape functions

It can be shown that the conservation of the mass, momentum and external force can be guaranteed as long as both the shape functions N_I and N_c are partitions of unity, which requires

$$\sum_{I=1}^{n_\mathrm{g}} N_I(\boldsymbol{x}) = 1 \text{ and } \sum_{c=1}^{n_\mathrm{c}} N_c(\boldsymbol{x}) = 1 \tag{22}$$

The first equation always holds, but the second equation may not. As shown in Fig. 4, the auxiliary grid can be divided into two regions: a single layer of boundary region (in cyan) and an inner region (in yellow). In the inner

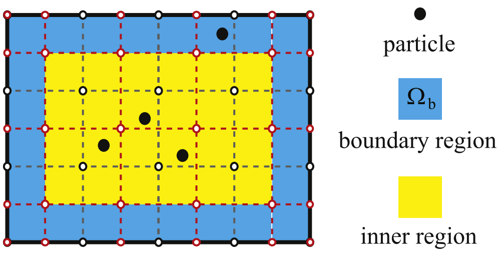


Fig. 4. The auxiliary grid, which consists of a single layer of boundary region and an inner region.

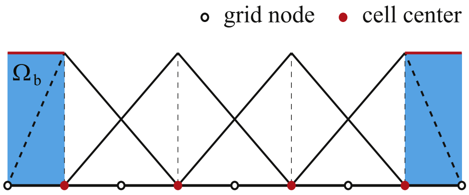


Fig. 5. The corrected shape function for the boundary cell.

region, each auxiliary grid node is the cell center of the background grid. For the particles located in the inner region, the second equation holds. However, for the particles located in the boundary region, the second equation does not hold.

Therefore, the shape function N_c has to be corrected in the boundary region Ω_b. Take one-dimension as an example, the corrected shape function is taken as

$$N_c = \begin{cases} \frac{1}{2}\left(1 + \xi_c \xi\right) & \boldsymbol{x} \in \Omega - \Omega_\mathrm{b} \\ 1 & \boldsymbol{x} \in \Omega_\mathrm{b} \end{cases} \tag{23}$$

which is plotted in Fig. 5. In the boundary region Ω_b, the shape function is replaced by $N_c = 1$ shown by the red line, while the rest remains unchanged. The three-dimensional shape function can be obtained by the tensor product of the three one-dimensional shape functions.

After the minor correction, the second equation in Eq. (22) always holds, which leads to

$$\sum_{I=1}^{n_\mathrm{g}} m_I = \sum_{c=1}^{n_\mathrm{c}} \left(\sum_{I=1}^{n_\mathrm{g}} N_{Ic} \right) m_c = \sum_{p=1}^{n_\mathrm{p}} \left(\sum_{c=1}^{n_\mathrm{c}} N_{cp} \right) m_p = \sum_{p=1}^{n_\mathrm{p}} m_p \tag{24}$$

$$\sum_{I=1}^{n_\mathrm{g}} \boldsymbol{p}_I = \sum_{c=1}^{n_\mathrm{c}} \left(\sum_{I=1}^{n_\mathrm{g}} N_{Ic} \right) \boldsymbol{p}_c = \sum_{p=1}^{n_\mathrm{p}} \left(\sum_{c=1}^{n_\mathrm{c}} N_{cp} \right) \boldsymbol{p}_p = \sum_{p=1}^{n_\mathrm{p}} \boldsymbol{p}_p \tag{25}$$

Therefore, the mass and momentum are conserved during the reconstruction stage. After these reconstructions, the momentum equation can be solved at each grid node.

3.4. Central difference method

The momentum equation (3) can be solved by using the explicit time integration to accurately capture the high-frequency response in transient problems like impact and blast. The central difference method is a common choice and its leapfrog variant is used. As shown in Fig. 6, after obtaining the acceleration $\ddot{u}_{iI}^k$ at time t^k by solving the momentum equation (3), the velocity $\dot{u}_{iI}^{k+1/2}$ at time $t^{k+1/2}$ can be updated as

$$\dot{u}_{iI}^{k+1/2} = \dot{u}_{iI}^{k-1/2} + \ddot{u}_{iI}^k \Delta t^k \tag{26}$$

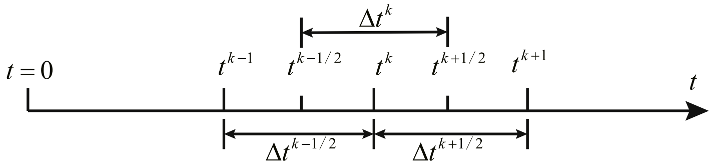


Fig. 6. Central difference scheme.

and then the displacement u_{iI}^{k+1} at time t^{k+1} can be updated as

$$u_{iI}^{k+1} = u_{iI}^{k} + \dot{u}_{iI}^{k+1/2} \Delta t^{k+1/2} \tag{27}$$

where $\Delta t^k = \frac{1}{2}(\Delta t^{k-1/2} + \Delta t^{k+1/2})$.

3.5. *Update of the particle information*

The next mission is to update the particle information, including the position, velocity and stress. To update the stress, the strain rate $\dot{\varepsilon}_{ij}$ and vorticity Ω_{ij} are required. In the conventional MPM, the strain rate/vorticity is evaluated at particles by the grid nodal velocity gradient so that they are discontinuous across the cell boundary.

To inherit the Lagrangian properties from MPM, the SGMP also updates the stress at particles and the rate form of the constitutive models is used for nonlinear analyses. However, the SGMP constructs a smooth strain rate/vorticity field based on the auxiliary grid to interpolate the strain rate/vorticity at the particles. The strain rate/vorticity of all auxiliary grid nodes can be obtained from the velocity gradient of the background grid cell as

$$\dot{\varepsilon}_{ija} = \sum_{I=1}^{n_{\mathrm{g}}} \frac{1}{2} \left(N_{Ia,j} v_{iI} + N_{Ia,i} v_{jI} \right) \tag{28}$$

$$\Omega_{ija} = \sum_{I=1}^{n_{\mathrm{g}}} \frac{1}{2} \left(N_{Ia,j} v_{iI} - N_{Ia,i} v_{jI} \right) \tag{29}$$

where the subscript a denotes the variables associated with the auxiliary grid nodes which are shown as red circles in Fig. 3. The strain rate/vorticity of particles can be interpolated from those of the auxiliary grid nodes by

$$\dot{\varepsilon}_{ijp} = \sum_{a=1}^{n_{\mathrm{a}}} N_{ap} \dot{\varepsilon}_{ija} \tag{30}$$

$$\Omega_{ijp} = \sum_{a=1}^{n_{\mathrm{a}}} N_{ap} \Omega_{ija} \tag{31}$$

Furthermore, the velocity and position of particles are updated in the auxiliary grid as

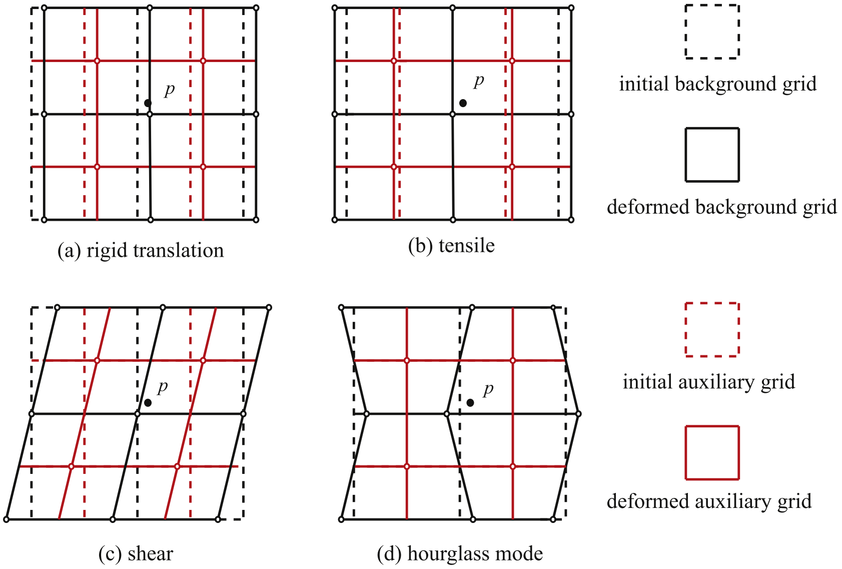
$$v_{ip}^{k+1/2} = v_{ip}^{k-1/2} + \sum_{a=1}^{n_{\mathrm{a}}} N_{ap}^{k} a_{ia}^{k} \Delta t \tag{32}$$

$$x_{ip}^{k+1} = x_{ip}^{k} + \sum_{a=1}^{n_{\mathrm{a}}} N_{ap}^{k} v_{ia}^{k+1/2} \Delta t \tag{33}$$

where the auxiliary grid nodal velocity $v_{ia}^{k+1/2}$ and acceleration a_{ia}^{k} are determined by

$$v_{ia}^{k+1/2} = \sum_{I=1}^{n_{\mathrm{g}}} \frac{N_{Ia}^{k} p_{iI}^{k+1/2}}{m_{I}^{k}} \tag{34}$$

$$a_{ia}^{k} = \sum_{I=1}^{n_{\mathrm{g}}} \frac{N_{Ia}^{k} f_{iI}^{k}}{m_{I}^{k}} \tag{35}$$



(a) rigid translation (b) tensile

(c) shear (d) hourglass mode

Fig. 7. Four deformation modes along x-axis direction in 2D.

3.6. Hourglass modes

Using a single quadrature point at the cell center could result in spurious hourglass modes. However, the SGMP does not suffer from hourglassing because the particle velocity and position are updated on the auxiliary grid instead of the background grid.

Let us take the 2D problem shown in Fig. 7 as an example, in which the black/red solid lines denote the deformed background/auxiliary grids, while the black/red dashed lines denote the initial background/auxiliary grids. There exist four deformation modes along x-axis direction in the background grid: the rigid translation, tensile or compression, shear and hourglass mode. For the first three modes, the auxiliary grid is capable of moving with the background grid, as shown in Fig. 7(a), (b) and (c), respectively. However, for the hourglass mode, the auxiliary grid does not move with the background grid, as shown in Fig. 7(d). In other words, the auxiliary grid filters out the hourglass modes of the background grid so that the SGMP does not suffer hourglassing.

4. Computer implementation

In this section, the computer implementation of the SGMP is described in detail.

4.1. Reconstruction of the cell center quantities

The cell center quantities m_c, $\boldsymbol{\sigma}_c V_c$ and $m_c \boldsymbol{b}_c$ are reconstructed based on the auxiliary grid. To make the proposed SGMP as efficient as possible, only a single boundary layer of the auxiliary grid is generated in our implementation, instead of generating the full auxiliary grid, as shown in Fig. 8. The data of all auxiliary grid nodes in the inner region are stored in the corresponding cell centers of the background grid. Note that the auxiliary grid only has two layers of nodes, whose inner nodes coincide with the cell centers of the outermost layer of the background grid.

Thus, the cell center quantities reconstruction equation (17) can be implemented in two steps without explicitly correcting the auxiliary grid nodal shape function. In the first step, the physical quantities of particles located in the boundary region Ω_{b} (boundary particles) are distributed to the auxiliary grid nodes, while those of particles located in the inner region (inner particles) are distributed to the cell centers, namely

$$\varphi_a = \sum_{p=1}^{n_{\mathrm{bp}}} N_{ap} \varphi_p \tag{36}$$

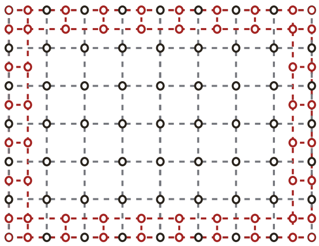


Fig. 8. The auxiliary grid generated in our implementation, which is only a single boundary layer of cells.

$$\varphi_c = \sum_{p=1}^{n_{\mathrm{ip}}} N_{cp} m_p \tag{37}$$

where n_{bp} and n_{ip} are the total number of boundary particles and inner particles, respectively. In the second step, the cell center quantities are obtained by summing up those from both the auxiliary grid nodes and the cell centers of the background grid as

$$\varphi_c = \varphi_c + \sum_{a=1}^{n_{\mathrm{a}}} \delta_{ac} \varphi_a \tag{38}$$

where δ_{ac} equals 1 if the auxiliary grid node a is located in the cell c; otherwise, it equals 0. Note that Eq. (38) simply assigns the cell center quantities obtained from Eq. (37) to the final cell center quantities, and accumulates the auxiliary grid nodal quantities to the cell centers of the outermost layer of the background grid in which they are located. This makes the quantities of all boundary particles are summed to the cell centers in which they are located, which is equivalent to correcting the auxiliary grid nodal shape function in the boundary region Ω_{b} as Eq. (23).

4.2. USL scheme in SGMP

Similar to the conventional MPM, the SGMP could update the particle stress at the beginning of each time step or at the end of each time step, which are referred to the update stress first (USF) scheme, and the update stress last (USL) scheme, respectively. In addition, the modified update stress last (MUSL) scheme, which is an improvement over the USL, can also be used in the SGMP. In this paper, the SGMP is implemented with the USL scheme, which is described in detail as follows.

1. Calculate the physical quantities at the cell center based on the Eqs. (36)–(38), as follows:

a. Distribute the boundary particle quantities m_p, $\boldsymbol{p}_p$, $\boldsymbol{\sigma}_p V_p$ and $\boldsymbol{b}_p m_p$ to the corresponding auxiliary grid nodes using Eq. (36), namely

$$m_a^k = \sum_{p=1}^{n_{\mathrm{bp}}} N_{ap}^k m_p \tag{39}$$

$$\boldsymbol{p}_a^{k-1/2} = \sum_{p=1}^{n_{\mathrm{bp}}} N_{ap}^k m_p \boldsymbol{v}_p^{k-1/2} \tag{40}$$

$$\boldsymbol{\sigma}_a^k V_a^k = \sum_{p=1}^{n_{\mathrm{bp}}} N_{ap}^k \boldsymbol{\sigma}_p^k V_p^k \tag{41}$$

$$\boldsymbol{b}_a^k m_a^k = \sum_{p=1}^{n_{\mathrm{bp}}} N_{ap}^k \boldsymbol{b}_p^k m_p \tag{42}$$

b. Similarly, distribute the inner particle quantities m_p, $\boldsymbol{p}_p$, $\boldsymbol{\sigma}_p V_p$ and $\boldsymbol{b}_p m_p$ to the corresponding cells using Eq. (37).

c. Calculate the cell center quantities using Eq. (38).

2. Determine the grid nodal mass and momentum by Eqs. (14) and (21) as

$$m_I^k = \sum_{c=1}^{n_\mathrm{c}} N_{Ic}^k m_c^k \tag{43}$$

$$p_{iI}^{k-1/2} = \sum_{c=1}^{n_\mathrm{c}} N_{Ic}^k p_{ic}^{k-1/2} \tag{44}$$

where n_c is the total number of background grid cells.

3. Calculate the grid nodal force by Eqs. (15) and (16) as

$$f_{iI}^{\mathrm{int,k}} = -\sum_{c=1}^{n_\mathrm{c}} N_{Ic,j}^k \sigma_{ijc}^k V_c^k \tag{45}$$

$$f_{iI}^{\mathrm{ext,k}} = \sum_{c=1}^{n_\mathrm{c}} N_{Ic}^k m_c^k b_{ic}^k \tag{46}$$

$$f_{iI}^k = f_{iI}^{\mathrm{int,k}} + f_{iI}^{\mathrm{ext,k}} \tag{47}$$

4. Integrate the grid nodal momentum equation by

$$p_{iI}^{k+1/2} = p_{iI}^{k-1/2} + f_{iI}^k \Delta t^k \tag{48}$$

5. Impose the essential boundary conditions on the background grid. At the fixed boundary, set $p_{iI}^{k+1/2} = 0$.

6. Calculate the incremental strain and incremental vorticity of the cell centers and the auxiliary grid nodes using Eqs. (28) and (29), taking the auxiliary grid nodes for example, which is given as

$$\Delta\varepsilon_{ija}^{k+1/2} = \sum_{I=1}^{n_\mathrm{g}} \frac{1}{2}\left(N_{Ia,j}^k v_{iI}^{k+1/2} + N_{Ia,i}^k v_{jI}^{k+1/2}\right)\Delta t^{k+1/2} \tag{49}$$

$$\Delta\Omega_{ija}^{k+1/2} = \sum_{I=1}^{n_\mathrm{g}} \frac{1}{2}\left(N_{Ia,j}^k v_{iI}^{k+1/2} - N_{Ia,i}^k v_{jI}^{k+1/2}\right)\Delta t^{k+1/2} \tag{50}$$

7. Calculate the velocity and acceleration of the cell centers and the auxiliary grid nodes using Eqs. (34) and (35), taking the auxiliary grid nodes for example, which is given as

$$v_{ia}^{k+1/2} = \sum_{I=1}^{n_\mathrm{g}} \frac{N_{Ia}^k p_{iI}^{k+1/2}}{m_I^k} \tag{51}$$

$$a_{ia}^k = \sum_{I=1}^{n_\mathrm{g}} \frac{N_{Ia}^k f_{iI}^k}{m_I^k} \tag{52}$$

8. The particle incremental strain $\Delta\varepsilon_{ijp}^{k+1/2}$ and incremental vorticity $\Delta\Omega_{ijp}^{k+1/2}$ are obtained by interpolating the auxiliary grid nodal values for the boundary particles, and by interpolating the cell center values for the inner particles. Take the boundary particle for example

$$\Delta\varepsilon_{ijp}^{k+1/2} = \sum_{a=1}^{n_\mathrm{a}} N_{ap}^k \Delta\varepsilon_{ija}^{k+1/2} \tag{53}$$

$$\Delta\Omega_{ijp}^{k+1/2} = \sum_{a=1}^{n_\mathrm{a}} N_{ap}^k \Delta\Omega_{ija}^{k+1/2} \tag{54}$$

9. Update the particle information, as follows:

a. The velocity and position of the boundary particles are updated by the acceleration and velocity of the auxiliary grid nodes by Eqs. (32) and (33) as

$$v_{ip}^{k+1/2} = v_{ip}^{k-1/2} + \sum_{a=1}^{n_\mathrm{a}} N_{ap}^{k} a_{ia}^{k} \Delta t^{k} \tag{55}$$

$$x_{ip}^{k+1} = x_{ip}^{k} + \sum_{a=1}^{n_\mathrm{a}} N_{ap}^{k} v_{ia}^{k+1/2} \Delta t^{k+1/2} \tag{56}$$

The velocity and position of the inner particles are updated in the same way by the acceleration and velocity of the cell centers.

b. The particle density is obtained by

$$\rho_{p}^{k+1} = \rho_{p}^{k} / \left(1 + \Delta\varepsilon_{iip}^{k+1/2}\right) \tag{57}$$

c. Update the particle stress state based on $\Delta\varepsilon_{ijp}^{k+1/2}$ and $\Delta\Omega_{ijp}^{k+1/2}$ by a constitute model.

10. Store all the information in the particles and discard the deformed grid. Then reset a new staggered grid in the next time step.

4.3. The stability of USL scheme

The USL scheme in the conventional MPM is numerically unstable. Assume that the grid node I is only related to one particle p. In the conventional MPM, the grid nodal velocity used to update the stress state in the USL is given as [7]:

$$v_{iI}^{k+1/2} = v_{ip}^{k-1/2} + \left(-\frac{N_{Ip,j}^{k} \sigma_{ijp}}{N_{Ip}^{k} \rho_{p}} + b_{ip}^{k} \right) \Delta t^{k} \tag{58}$$

When this particle moves close to the opposite side of the grid node I, we have $N_{Ip}^{k} \to 0$, but $N_{Ip,j}^{k} \neq 0$, which leads to an infinite velocity and makes the USL scheme unstable.

In the SGMP, taking an inner particle as an example, the mass and force of the grid node I are given as

$$m_{I}^{k} = \sum_{c=1}^{n_\mathrm{c}} N_{Ic}^{k} N_{cp}^{k} m_{p} \tag{59}$$

$$f_{iI}^{k} = -\sum_{c=1}^{n_\mathrm{c}} N_{Ic,j}^{k} N_{cp}^{k} \sigma_{ijp}^{k} V_{p}^{k} + \sum_{c=1}^{n_\mathrm{c}} N_{Ic}^{k} N_{cp}^{k} m_{p} b_{ip}^{k} \tag{60}$$

Substituting Eqs. (59) and (60) into Eq. (48), the grid nodal velocity used to update the stress is obtained as

$$v_{iI}^{k+1/2} = v_{ip}^{k-1/2} + \left(-\frac{\sigma_{ijp}^{k} V_{p}^{k} \sum_{c=1}^{n_\mathrm{c}} N_{Ic,j}^{k} N_{cp}^{k}}{m_{p} \sum_{c=1}^{n_\mathrm{c}} N_{Ic}^{k} N_{cp}^{k}} + b_{ip}^{k} \right) \Delta t^{k} \tag{61}$$

In the above equation, the term N_{Ic}^{k} equals 1/8 in three-dimensions and the term N_{cp}^{k} appears in both the numerator and denominator, so that $v_{iI}^{k+1/2}$ will not become infinity as $N_{cp}^{k} \to 0$.

5. Numerical examples and discussion

The proposed SGMP has been implemented in our MPM3D code [7,19], and several numerical examples are studied in this section to verify and validate the method. The MUSL scheme is used in all MPM simulations.

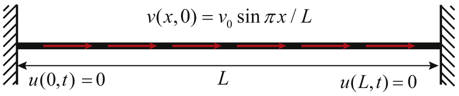


Fig. 9. A linear-elastic bar fixed at both ends.

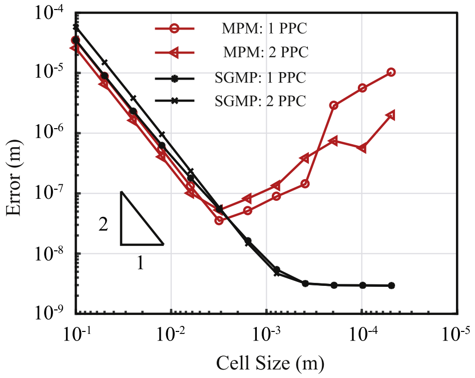


Fig. 10. Error versus cell size for the MPM and SGMP.

5.1. Vibrating bar with fixed ends

The first benchmark is the vibration of a linear-elastic bar studied by Tielen in his master thesis [46]. As shown in Fig. 9, the bar has a length of $L = 1$ m and is fixed at both ends. A prescribed velocity field, $v(x, 0) = v_0 \sin(\pi x/L)$ with $v_0 = 0.1$ m/s, is given at $t = 0$ s. The material has a density of $\rho = 25$ kg/m^3, Young modulus of $E = 50$ Pa and Poisson ratio of $\nu = 0$. The analytical displacement for the case of small deformations is

$$u(x,t) = \frac{v_0}{w} \sin(wt) \sin\left(\frac{\pi x}{L}\right) \tag{62}$$

where

$$w = \frac{\pi}{L}\sqrt{\frac{E}{\rho}} \tag{63}$$

The accuracy of the MPM solution mainly depends on the time and space discretization. This paper focuses on the latter. Therefore, the dependence of solution quality on grid spacing is investigated to evaluate the performance of the MPM and SGMP. The L^2 error norm can be calculated using particle quadrature as

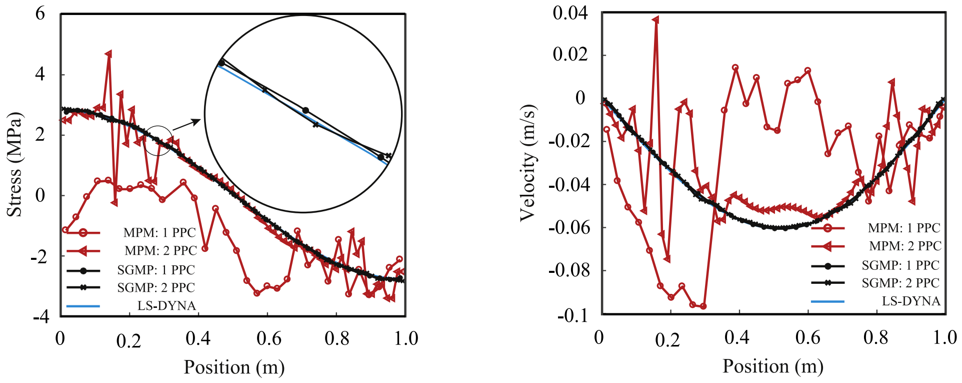
$$\text{Error} = \sqrt{\frac{\sum_{p=1}^{n_p}\left(u_p^{\mathrm{s}}(t) - u^{\mathrm{a}}\left(x_p, t\right)\right)^2}{n_p}} \tag{64}$$

where $u_p^{\mathrm{s}}(t)$ is the simulation solution of the particle p and $u^{\mathrm{a}}\left(x_p, t\right)$ is the analytical solution at position x_p. The analytical solution is applicable for the small deformation case so we examine the error at time $t = 0.02$ s with a maximum strain ϵ of 0.628%. In addition, a small time step of $\Delta t = 10^{-5}$ s is used for all the simulations of this benchmark to ensure that the time discretization error is negligible compared to the spatial discretization error.

Fig. 10 plots the error versus the cell size for the MPM and SGMP. At the beginning stage, the error decreases as the cell size decreases in both methods and the slope of the line, which is the space convergence rate, is very close to 2, whose values are listed in Table 1. But if we continue to refine the cells, the space convergence rate of both methods drops. For the SGMP, the error keeps at a very small value and remains almost unchanged. However, the MPM error increases quickly, which can lead to unreliable results. To study the influence of the number of particles per cell (PPC) on the accuracy, two cases including one particle per cell and two particles per cell are studied in Fig. 10. It shows that the number of particles per cell has a little influence on the solution accuracy in the SGMP.

Table 1
The space convergence rate: $\log_2\left(\frac{\text{Error}(h)}{\text{Error}(h/2)}\right)$ ($h = 1/N$: the cell size).

Number of cells N	MPM		SGMP	
	1 PPC	2 PPC	1 PPC	2 PPC
10	1.992	1.992	1.966	1.955
$10 * 2^1$	2.000	2.000	1.946	1.978
$10 * 2^2$	2.008	2.006	1.894	1.995
$10 * 2^3$	2.029	2.010	1.806	2.028
$10 * 2^4$	1.920	0.928	1.717	2.018
$10 * 2^5$	−0.552	−0.638	1.733	1.975
$10 * 2^6$	−0.795	−0.718	1.596	1.654
$10 * 2^7$	−0.688	−1.517	0.773	0.550
$10 * 2^8$	−4.326	−0.947	0.094	0.103
$10 * 2^9$	−0.958	0.394	0.002	0.014
$10 * 2^{10}$	−0.874	−1.814	0.013	0.008



(a) The stress distribution along the bar

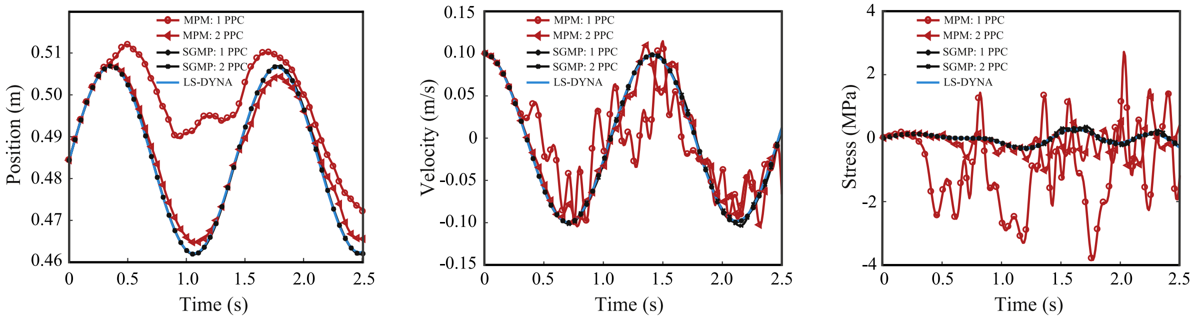
(b) The velocity distribution along the bar

Fig. 11. The stress and velocity distribution at time $t = 0.5$ s obtained by the MPM, SGMP and LS-DYNA.

To further investigate the performance of the SGMP for large deformation problems, the distributions of the stress and velocity along the bar at time $t = 0.5$ s obtained by the MPM and SGMP using 32 cells are compared with those obtained by LS-DYNA in Fig. 11. The maximum strain in this case is 7.07%, so that the solution (62) is no longer valid. Hence, the result obtained by LS-DYNA is taken as the reference instead. In such a large deformation problem, the grid crossing occurs frequently. For the MPM, the grid crossing leads to severe oscillations in the stress and velocity. It shows that increasing the number of particles per cell, which increase the number of integration points, improves the solution accuracy in the MPM. As shown in Fig. 11, the stress and velocity obtained by two particles per cell oscillate around the LS-DYNA solution while the result obtained by one particle per cell has a large and unacceptable error. However, the results obtained by the SGMP agree very well with those obtained by LS-DYNA for both cases. Increasing the number of particles per cell, which does not increase the number of integration points, has little influence on the solution accuracy in the SGMP.

We continue the above simulation to time $t = 2.5$ s, which is more than one cycle. The time histories of the position, velocity and stress of a particle obtained by the MPM and SGMP are compared with those obtained by LS-DYNA in Fig. 12. For the case of one particle per cell, the particle is the one just left to the center of the bar. For the case of two particles per cell, the average results of the two particles just left to the center of the bar are plotted. Similarly, the results of the SGMP agree well with the LS-DYNA solution and the number of particles per cell has little influence on the quality of the solution. However, the results of the MPM oscillate severely due to the grid cell crossing, and increasing the number of particles per cell can improve the quality of the solution.

To study the numerical dissipation of the proposed method, the internal energy ratio, kinetic energy ratio and total energy ratio, relative to the initial total energy, obtained by the MPM and SGMP with 1 PPC and 2 PPC



(a) The position of the particle over the time (b) The velocity of the particle over the time (c) The stress of the particle over the time

Fig. 12. The time history of the position, velocity and stress of a particle just left to the center of the bar.

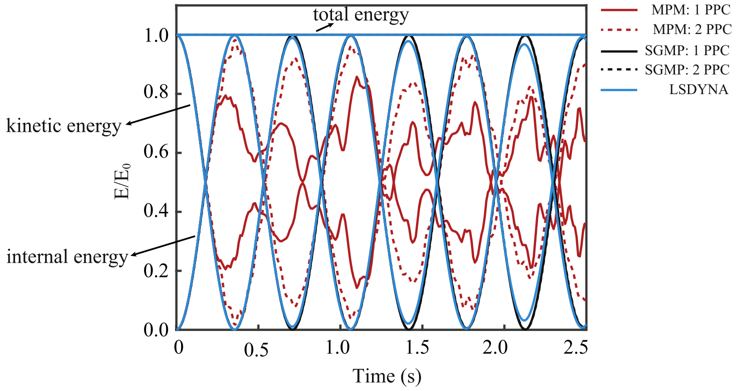


Fig. 13. Comparison of the internal energy, kinetic energy and total energy obtained by the MPM, SGMP and LS-DYNA.

are compared with those obtained by LS-DYNA in Fig. 13. It shows that the SGMP does not introduce numerical dissipation for this example. For the MPM, the results initially agree well with the SGMP and LS-DYNA, but are quite noisy and different with the others later on.

This example shows that the quality of the solution drops while grid crossing occurs in the MPM. Increasing the particle per cell improves the accuracy but is unable to eliminate the oscillations of the solution. In the SGMP, the solution does not oscillate while grid crossing occurs. To improve the accuracy of solution, refining the grid cells is an effective way, and one particle per cell is recommended in the case of no numerical fracture.

5.2. Quasi-static compaction under self-weight

To further study the space convergence of the proposed SGMP, one dimensional quasi-static compaction under self-weight is repeated in this section, which has been studied by Bardenhagen et al. [33]. As shown in Fig. 14, the bar has an initial length of $L = 50$ fixed at the bottom and deforms due to the self-weight of g. The total column weight is given as $W = \rho_0 g L$. The material has a density of $\rho_0 = 1$ and Young modulus of $E = 1.0 \times 10^6$. Any consistent set of units suffices. Material response is determined by a one-dimensional hyperelastic model as [33]

$$\sigma = E\,(F - 1) \tag{65}$$

where F is the deformation gradient in one dimension. The quasi-static analytical solutions under self-weight are given as [36,41]

$$\sigma\,(X) = \rho_0 g\,(L - X) \tag{66}$$

where X is the coordinate in the initial configuration. We discretize the bar with grid size $d = 1$ as used by Bardenhagen et al. [33]. To obtain the quasi-static result, the gravity is linearly increased in 2 s, which is 40 times wave transit time in the column.

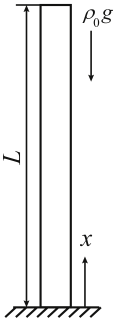


Fig. 14. One-dimensional column under self-weight.

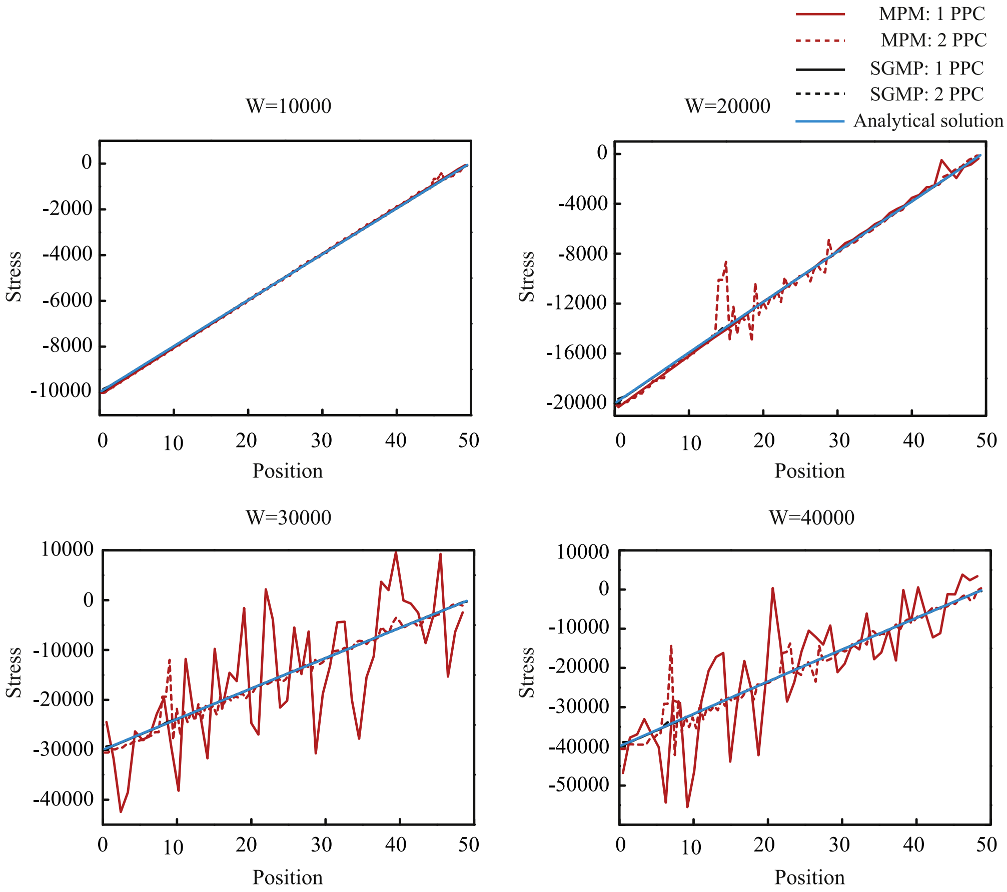


Fig. 15. The stress distribution at different gravity from $W = 10\,000$ to $W = 40\,000$.

Fig. 15 plots the stress distribution for different cases from $W = 10\,000$ to $W = 40\,000$, calculated by the MPM and SGMP. In all cases, the SGMP results obtained with both one and two particles per cell agree well with the analytical solution. However, the MPM can only give satisfactory results for the case $W = 10\,000$ for which no cell crossing occurs. Even for $W = 20\,000$, there is noise in the MPM results due to cell crossing noise. With the weight increasing, the MPM solution loses its accuracy.

To further compare the particle stresses to the analytical solution, an error measure is defined as

$$\text{Error} = \frac{\sum_{p=1}^{n_p} \left|\sigma_p^{\mathrm{s}} - \sigma^{\mathrm{a}}\left(X_p\right)\right| l_{p0}}{WL} \tag{67}$$

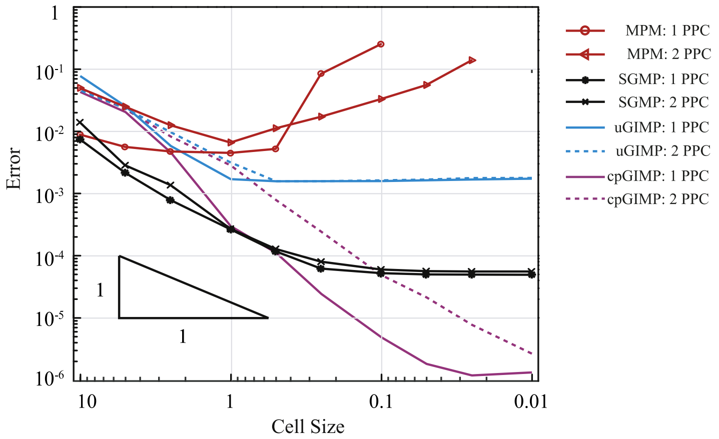


Fig. 16. Error versus cell size for the MPM, SGMP, uGIMP [33] and cpGIMP [33].

where σ_p^s is the simulation solution of the particle p, $\sigma^a(X_p)$ is the analytical solution with initial coordinate X_p and l_{p0} is the initial size of the particle p. Errors are plotted in Fig. 16 for different grid sizes under $W = 10\,000$. The same small time step size of $\Delta t = 5 \times 10^{-6}$ is used for all simulations to ensure that the time discretization error is negligible compared to the spatial discretization error. It shows that with the decrease of cell size, the error of the SGMP decreases at the optimal space convergence rate of 1 when the cell size is greater than 0.25 while the MPM error increases quickly, which can lead to unreliable results. The SGMP gives much better results than the uGIMP. Among these methods, the cpGIMP gives the highest accuracy when the cell size is smaller than 1, but it needs to track the particle shape during the simulation. Even though the gravity is applied slowly, there still exist small oscillations in the SGMP results which leads to a small error. Therefore, the error of the SGMP keeps at a very small value when the cell size is less than 0.25.

5.3. Euler–Bernoulli Beam

The two dimensional bending beam with a free–free boundary condition is studied next. As shown in Fig. 17, the beam has a length of $L = 0.06$ m and a height of $H = 0.01$ m. The small ratio of H/L satisfies the requirement of the Euler–Bernoulli beam theory [50]. The first flexural mode of the Euler–Bernoulli beam is given as

$$v(x) = Aw\left[C_1 \sin(\beta x) + C_2 \cos(\beta x) + C_3 \sinh(\beta x) + C_4 \cosh(\beta x)\right] \tag{68}$$

where $A = y_{\max}/2C_2$, $y_{\max}$ is the maximum deflection at the ends which is set to 0.005 m here; β is the first non-zero root of the frequency equation (70) and w is the angular frequency, which is related to β by

$$w = \beta^2 \sqrt{EH^2/12\rho\left(1 - \nu^2\right)} \tag{69}$$

The frequency equation and the constant coefficients C_i are determined by the free–free boundary condition as

$$\cos(\beta L)\cosh(\beta L) - 1 = 0 \tag{70}$$

$$C_1 = C_3 = \cosh(\beta L) - \cos(\beta L) \tag{71}$$

$$C_2 = C_4 = \sin(\beta L) - \sinh(\beta L) \tag{72}$$

The initial velocity of the beam is determined by Eq. (68). The material has a density of $\rho = 1845$ kg/m^3, Young modulus of $E = 318$ GPa and Poisson ratio of $\nu = 0.054$.

Figs. 18 and 19 compare the stress σ_{xx} and velocity V_y distribution obtained by the MPM, SGMP and LS-DYNA at time $t = 0.5$ µs, 4 µs, 10 µs and 20 µs, respectively. Both figures show that the SGMP did eliminate the cell crossing noise and performs well. The distribution pattern and the magnitude of both the stress and velocity obtained

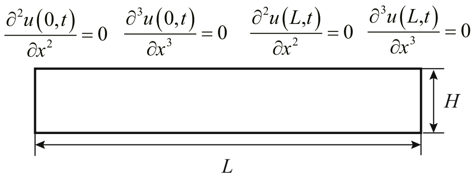
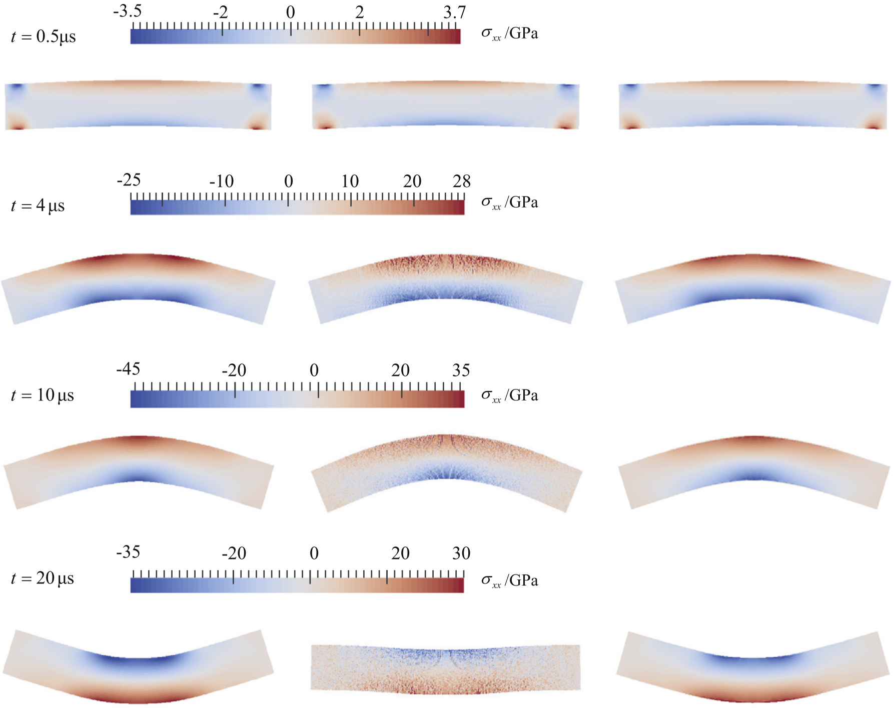

$\frac{\partial^2 u(0,t)}{\partial x^2}=0$
$\frac{\partial^3 u(0,t)}{\partial x^3}=0$
$\frac{\partial^2 u(L,t)}{\partial x^2}=0$
$\frac{\partial^3 u(L,t)}{\partial x^3}=0$
H
L

Fig. 17. The beam with a free–free boundary condition.



(a) LS-DYNA (b) MPM (c) SGMP

Fig. 18. Stress σ_{xx} distribution obtained by the LS-DYNA, MPM and SGMP at time $t = 0.5$ μs, 4 μs, 10 μs and 20 μs, respectively.

by the SGMP agree well with those obtained by LS-DYNA. However, the results of the MPM are quite noisy and the accuracy is seriously affected by the cell crossing noise.

Fig. 20 compares the internal energy ratio, kinetic energy ratio and total energy ratio, relative to the initial total energy, obtained by the MPM, SGMP and LS-DYNA. The MPM gives very poor results whose total energy increases significantly. The refinement of grid could not improve the accuracy. On the contrary, the results obtained by the SGMP agree reasonably with the LS-DYNA solution. In this example, the SGMP introduces numerical dissipation leading to small energy loss. The numerical dissipation will be further reduced when the grid is refined.

5.4. Taylor bar impact

The Taylor bar impact shown in Fig. 21 is studied next to further evaluate the performance of the SGMP. In this problem, a cylindrical bar with an initial length of $L_0 = 25.4$ mm and a diameter of $D_0 = 7.6$ mm impacts on a rigid wall at an initial velocity of $v_0 = 190$ m/s. The Taylor bar impact is often used to validate constitutive models because its experimental data are available. This problem has been simulated in many studies [7,51], and the average error

$$\text{Error} = \frac{1}{3}\left(\frac{|\Delta L|}{L} + \frac{|\Delta D|}{D} + \frac{|\Delta W|}{W}\right) \tag{73}$$

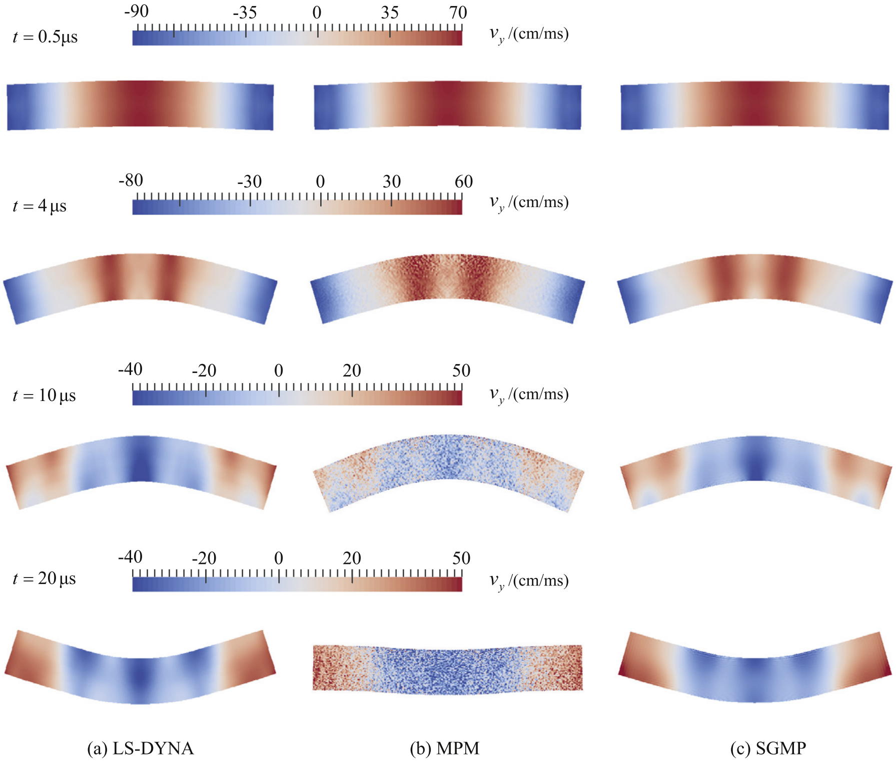


Fig. 19. Velocity v_y distribution obtained by the LS-DYNA, MPM and SGMP at time $t = 0.5$ μs, 4 μs, 10 μs and 20 μs, respectively.

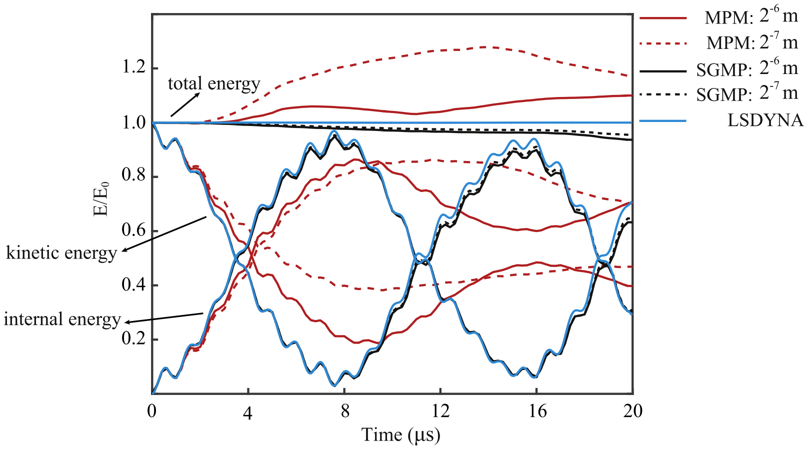


Fig. 20. Comparison of the internal energy, kinetic energy and total energy obtained by the MPM, SGMP and LS-DYNA.

is usually used to compare the final configuration of the bar obtained by the simulation and the experiment. In Eq. (73), L, D and W are the final length, the diameter at the bottom and the diameter at the height of $0.2L_0$, respectively, ΔL, ΔD and ΔW are their errors between the numerical and experimental results.

The bar is made of copper which is modeled by the Johnson–Cook model with material parameters listed in Table 2. This problem is simulated by the MPM, SGMP and LS-DYNA. The MPM and SGMP discretize the bar by 21 172 particles with an initial particle space of 0.38 mm and an initial grid space of 0.76 mm, while LS-DYNA discretizes the bar by 6528 elements with a maximum element length of 0.76 mm.

Table 2
Material parameters.

$\rho\ (\mathrm{kg/m^3})$	E (GPa)	ν	A (MPa)	B (MPa)	n	c
8930	117	0.35	157	425	1.0	0.0

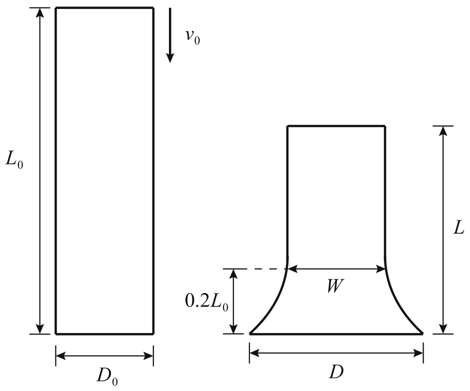
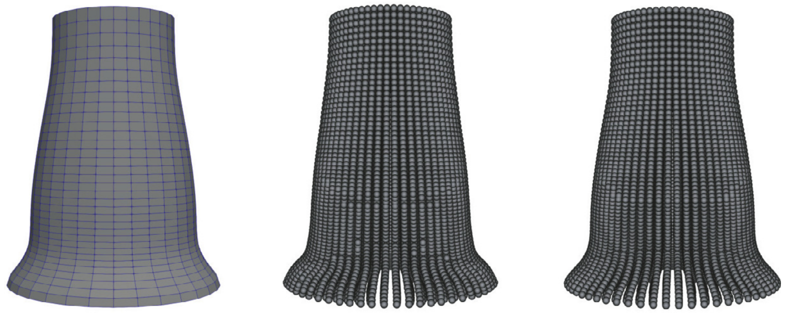
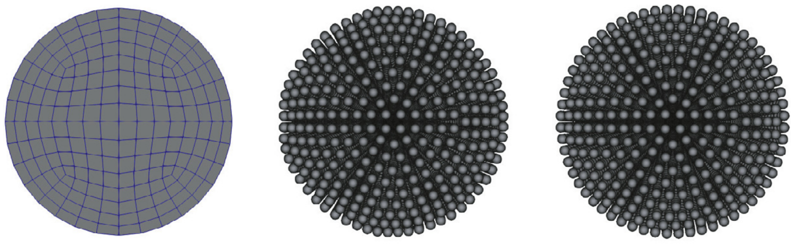


Fig. 21. Taylor bar impact.



(a) LS-DYNA (b) MPM (c) SGMP

Fig. 22. Front view of the Taylor bar at $t = 80$ μs obtained by LS-DYNA, MPM and SGMP.



(a) LS-DYNA (b) MPM (c) SGMP

Fig. 23. Bottom view of the Taylor bar at $t = 80$ μs obtained by the LS-DYNA, MPM and SGMP.

The simulation is continued to time $t = 80$ μs, when the velocity of the bar has reached zero. Figs. 22 and 23 compare the final configurations obtained by the MPM and SGMP with those obtained by LS-DYNA from front and bottom view, respectively. Both the MPM and SGMP results show good agreement with LS-DYNA results.

Fig. 24 plots the effective plastic strain at the end of the simulation. Compared to the MPM, the effective plastic strain obtained by the SGMP are much smoother, especially at the bottom of the bar.

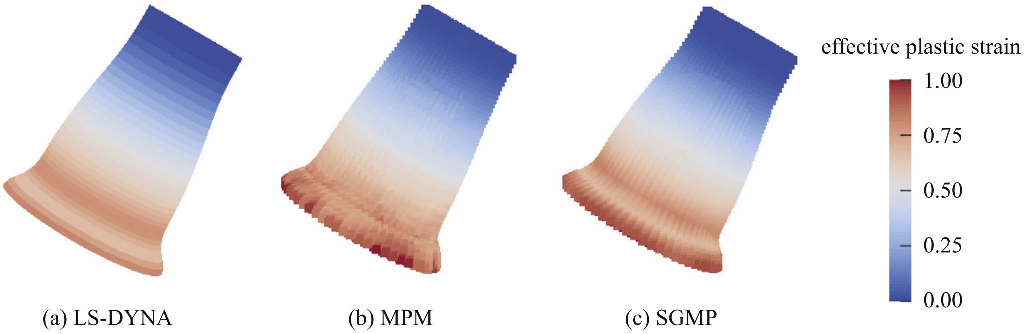


Fig. 24. Effective plastic strain of the Taylor bar at $t = 80$ μs obtained by the LS-DYNA, MPM and SGMP.

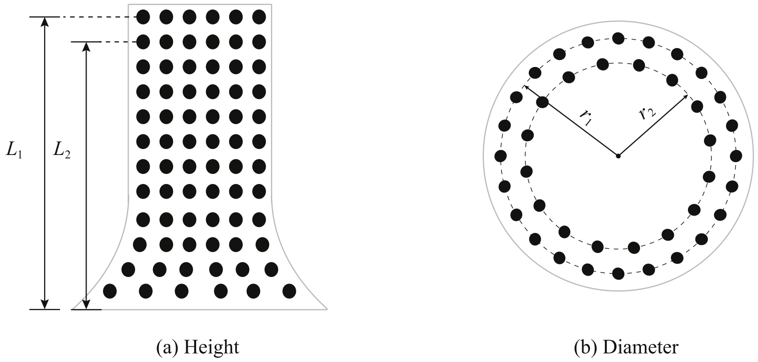


Fig. 25. The measurement method of the height and the diameter.

As each particle represents a material volume in the MPM and SGMP, the outermost particles in the bar are not located on the surface of the bar, as shown in Fig. 25. Let L_1 and L_2 denote the average heights of the particles located at the two uppermost layers, respectively, and r_1 and r_2 denote the average radii of the particles located at the two outermost circles, respectively. The parameters L and D of the bar can be determined as

$$L = L_1 + \frac{L_1 - L_2}{2} \tag{74}$$

$$D = 2\left(r_1 + \frac{r_1 - r_2}{2}\right) \tag{75}$$

The diameter at the height of $0.2L_0$, W, can be determined in the same way with D and the particles located at the height of $0.2L_0 \pm 0.19$ mm (0.19 mm is the initial radius of the particle) are chosen. In addition, the normalized standard deviation of the outermost layer particle radius at the bottom is defined to compare the uniformity of the particle distribution as

$$\bar{s}(r) = \frac{1}{r_1}\sqrt{\frac{1}{n_\mathrm{p}}\sum_{p=1}^{n_\mathrm{p}}\left(r_p - r_1\right)^2} \tag{76}$$

where r_p is the radius of the particle p.

Table 3 compares the numerical results of LS-DYNA, MPM and SGMP with the experimental data. Compared with the conventional MPM, the SGMP is closer to the experiment and has a more even distribution at the bottom of the bar. However, the conventional MPM also gives satisfactory results.

Fig. 26 compares the energy ratios obtained by the MPM, SGMP and LS-DYNA. In this example, both the MPM and SGMP give satisfactory results, which agree well with the LS-DYNA solution. The SGMP introduces a little numerical dissipation in this example which will be further reduced when refining the grid.

Table 3
Accuracy comparison for the Taylor bar impact simulation.

	L (mm)	D (mm)	W (mm)	Error	$\bar{s}\,(r)$ (%)
Experiment	16.2	13.5	10.1	–	–
LS-DYNA	16.22	13.14	10.18	0.012	0.073
MPM	16.34	13.18	9.98	0.015	0.530
SGMP	16.25	13.44	10.20	0.005	0.179

Table 4
Comparison in efficiency for the Taylor bar impact.

d_{cell} (mm)	0.76	0.38	0.19	0.095
t_{MPM} (s)	11.988	201.41	3740.3	63 691
t_{SGMP} (s)	15.273	253.94	4641.6	71 399
t_{r}(%)	27.40	26.08	24.10	12.10

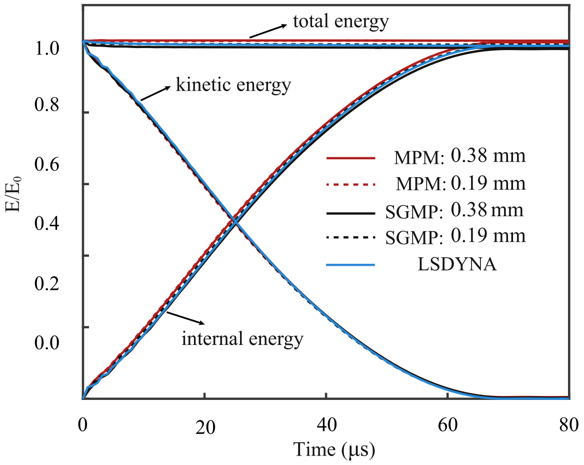


Fig. 26. Comparison of the internal energy, kinetic energy and total energy obtained by the MPM, SGMP and LS-DYNA.

5.5. *Efficiency*

To investigate the efficiency of the SGMP, the relative extra cost of the SGMP defined by

$$t_{\text{r}} = \frac{t_{\text{SGMP}} - t_{\text{MPM}}}{t_{\text{MPM}}} \tag{77}$$

is investigated next. All the simulations in this paper are run in a PC with one Intel(R) core(TM) i7-4790 CPU @ 3.60 GHz.

The Taylor bar impact studied in Section 5.4 is repeated with different grid space d_{cell} as listed in Table 4. The particle space d_{p} is chosen as a half of the grid space d_{cell}, so that there are initially two particles in each direction in each cell in all the cases. For $d_{\text{cell}} = 0.76$ mm and $d_{\text{p}} = 0.38$ mm, which corresponds to the case studied in Section 5.4, the SGMP takes 27.40% extra cost compared with the conventional MPM. As we refine the grid, the extra cost ratio t_{r} further reduces.

The extra cost of the SGMP mainly depends on the number of the auxiliary cells, which can be determined by

$$n_{\text{cell}} = (n_{\text{x}} + 1)\left(n_{\text{y}} + 1\right)(n_{\text{z}} + 1) - (n_{\text{x}} - 1)\left(n_{\text{y}} - 1\right)(n_{\text{z}} - 1) \tag{78}$$

where n_{x}, n_{y} and n_{z} are the number of the background grid cells or the conventional MPM cells along the x, y and z directions, respectively. For the three-dimensions, Eq. (78) shows that each factor of 0.5 in cell size leads to a factor of about 4 increase in the number of auxiliary grid cells but a factor of 8 increase in the number of background grid cells, which is given as $n_{\text{x}}n_{\text{y}}n_{\text{z}}$. Therefore, the ratio of the number of the auxiliary grid cells to

Table 5
Comparison in efficiency for the vibrating bar by refining the grid.

Number of cells	20^3	40^3	80^3	160^3
t_{MPM} (s)	1.7080	27.242	458.27	7815.4
t_{SGMP} (s)	1.9070	29.709	493.07	7837.9
t_{r}(%)	11.65	9.06	7.59	0.29

Table 6
Comparison in efficiency for the vibrating bar by refining the particles.

Particles per cell	1^3	2^3	4^3	8^3
t_{MPM} (s)	4.424	27.242	209.6	1658.3
t_{SGMP} (s)	6.523	29.709	214.96	1677.6
t_{r}(%)	47.45	9.06	2.56	1.16

Table 7
The percentage of CPU time consumed in each stage.

d_{cell} (mm)		0.76	0.38	0.19	0.095
MPM(%)	1	14.33	14.18	13.98	14.83
	2	17.62	18.40	18.23	18.76
	3	64.65	64.10	64.54	63.64
SGMP(%)	1	43.25	44.30	45.66	47.01
	2	4.07	3.91	3.66	3.45
	3	47.59	47.12	46.17	45.52

that of the background grid cells reduces as we refine the grid, which leads to the reduction of the extra cost ratio t_{r}.

To further study the effect of the number of grid cells on efficiency in the SGMP, the vibrating bar studied in Subsection 5.1 is extended to three-dimensions by applying the initial velocity field in each direction. Table 5 compares the computational time used by the MPM and SGMP with different grid sizes. In all cases, two particles are used in each direction in each cell. It shows that the extra cost ratio t_{r} reduces as the number of grid cells increases. When the number of grid cells is increased to 160 in each direction, the SGMP costs almost the same as the conventional MPM.

Similarly, an increase in number of particles also leads to a reduction in the extra cost ratio t_{r}. To investigate this reduction, the number of grid cells is fixed as 40 in each direction while the number of particles per cell is increased in each direction. As listed in Table 6, the extra cost ratio t_{r} is reduced significantly with the increase of the number of particles per cell. When 2 particles per cell are used in each direction, the SGMP costs only about 9% more than the MPM.

Furthermore, the CPU time consumed in different stages of the MPM and SGMP is further studied using the Taylor bar impact with different cell size and two particles per cell. Each time step in the conventional MPM consists of three stages: 1. initialize the grid information, 2. solve the grid nodal momentum equation, and 3. update the particle information by the background grid. Similarly, the SGMP also consists of three stages: 1. initialize the background grid and auxiliary grid information, 2. solve the grid nodal momentum equation, and 3. update the particle information by both sets of grids. Table 7 lists the percentage of CPU time consumed in each stage. It shows that the relative cost of the initialization stage in the SGMP is much higher than that in the MPM as the SGMP needs to initialize both the background and auxiliary grids. The cost ratio of the second stage in the SGMP is much less than that in the MPM due to fewer integration points used. Besides, the most time-consuming stage is the particle update stage, which costs about 60% in the MPM and less than 50% in the SGMP. Table 7 also shows that the relative cost of each stage is almost independent of the cell size.

6. Conclusion

In this paper, the staggered grid material point method, SGMP, is proposed to eliminate the cell crossing noise efficiently. The volume integrals in the weak form are evaluated by cell center quadrature instead of particle quadrature as the sum of value of the integrand at each cell center multiplied by the quadrature weight of the cell. To make the SGMP as efficient as the MPM, the cell center quantities used in the cell center quadrature are reconstructed using an auxiliary grid. The inner nodes of the auxiliary grid coincide with the cell centers of the background grid, so that the inner nodal quantities of the auxiliary grid can be stored at the cell centers of the background grid. Thus, only a single boundary layer of the auxiliary grid needs to be generated. To further reduce the cell crossing noise, the SGMP establishes a continuous strain rate/vorticity field based on the auxiliary grid to interpolate the particle values. The USL version of the SGMP is presented and implemented in our MPM3D code.

Numerical studies show that the SGMP eliminates the cell crossing noise efficiently and gives a much more accurate result than the MPM. Especially for the problems involving frequent cell crossing, the SGMP performs well while the conventional MPM gives unacceptable results. It shows that the extra cost of the SGMP is about 27% in the Taylor bar impact problem, and reduces as the problem scale increases.

References

[1] T. Belytschko, W.K. Liu, B. Moran, K. Elkhodary, Nonlinear Finite Elements for Continua and Structures, John wiley & sons, 2013.

[2] M.R. Benioff, E.D. Lazowska, et al., Computational science: ensuring america's competitiveness, Report to the President, President's Information Technology Advisory Committee, Washington, DC, 2005.

[3] J. Oden, T. Belytschko, J. Fish, T. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, S. Yip, Simulation-based engineering science: Revolutionizing engineering science through simulation report of the national science foundation blue ribbon panel on simulation-based engineering science, february 2006.

[4] J.A. Zukas (Ed.), High Velocity Impact Dynamics, Wiley, New York, 1990.

[5] L.J. Hageman, J. Walsh, HELP, a multi-material Eulerian program for compressible fluid and elasticplastic flows in two space dimensions and time. Volume 2. Fortran listing of HELP, Tech. rep., Systems, Science and Software, La Jolla, California (1971).

[6] W.F. Noh, CEL: A time-dependent, two-space-dimensional, coupled Eulerian-Lagrange code, Tech. rep., Lawrence Radiation Laboratory, University of California, Livermore (1963).

[7] X. Zhang, Z. Chen, Y. Liu, The Material Point Method: A Continuum-Based Particle Method for Extreme Loading Cases, Academic Press, 2016.

[8] F.H. Harlow, The particle-in-cell computing method for fluid dynamics, Methods Comput. Phys. 3 (1964) 319–343.

[9] F.H. Harlow, A machine calculation method for hydrodynamic problems, Tech. Rep. LAMS-1956, Los Alamos Scientific Laboratory (1955).

[10] S. McKee, M.F. Tomé, V.G. Ferreira, J.A. Cuminato, A. Castelo, F. Sousa, N. Mangiavacchi, The MAC method, Comput. & Fluids 37 (8) (2008) 907–930.

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

[12] J.U. Brackbill, D.B. Kothe, H.M. Ruppel, FLIP: a low-dissipation, particle-in-cell method for fluid flow, Comput. Phys. Comm. 48 (1) (1988) 25–38.

[13] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, Comput. Methods Appl. Mech. Eng. 118 (1994) 179–186.

[14] W. Gong, Y. Liu, X. Zhang, H. Ma, Numerical investigation on dynamical response of aluminum foam subject to hypervelocity impact with material point method, CMES - Comput. Model. Eng. Sci. 83 (5) (2012) 527–545.

[15] Y. Liu, H.K. Wang, X. Zhang, A multiscale framework for high-velocity impact process with combined material point method and molecular dynamics, Int. J. Mech. Mater. Des. 9 (2) (2013) 127–139.

[16] P. Huang, X. Zhang, S. Ma, H. Wang, Shared memory openmp parallelization of explicit MPM and its application to hypervelocity impact, CMES - Comput. Model. Eng. Sci. 38 (2) (2008) 119–148.

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

[18] D. Sulsky, S.-J. Zhou, H.L. Schreyer, Application of a particle-in-cell method to solid mechanics, Comput. Phys. Comm. 87 (1) (1995) 236–252.

[19] Z. Ma, X. Zhang, P. Huang, An object-oriented MPM framework for simulation of large deformation and contact of numerous grains, CMES - Comput. Model. Eng. Sci. 55 (1) (2010) 61.

[20] P. Huang, X. Zhang, S. Ma, X. Huang, Contact algorithms for the material point method in impact and penetration simulation, Internat. J. Numer. Methods Engrg. 85 (4) (2011) 498–517.

[21] W. Hu, Z. Chen, Model-based simulation of the synergistic effects of blast and fragmentation on a concrete wall using the MPM, Int. J. Impact Eng. 32 (12) (2006) 2066–2096.

[22] Y. Wang, H. Beom, M. Sun, S. Lin, Numerical simulation of explosive welding using the material point method, Int. J. Impact Eng. 38 (1) (2011) 51–60.

[23] J.A. Nairn, Material point method calculations with explicit cracks, CMES - Comput. Model. Eng. Sci. 4 (6) (2003) 649–663.

[24] Y. Liang, T. Benedek, X. Zhang, Y. Liu, Material point method with enriched shape function for crack problems, Comput. Methods Appl. Mech. Engrg. 322 (2017) 541–562.

[25] H. Tan, J.A. Nairn, Hierarchical, adaptive, material point method for dynamic energy release rate calculations, Comput. Methods Appl. Mech. Engrg. 191 (19–20) (2002) 2123–2137.

[26] H. Schreyer, D. Sulsky, S. Zhou, Modeling delamination as a strong discontinuity with the material point method, Comput. Methods Appl. Mech. Engrg. 191 (23–24) (2002) 2483–2507.

[27] F. Zhang, X. Zhang, K.Y. Sze, Y. Lian, Y. Liu, Incompressible material point method for free surface flow, J. Comput. Phys. 330 (2017) 92–110.

[28] S. Kularathna, K. Soga, Implicit formulation of material point method for analysis of incompressible materials, Comput. Methods Appl. Mech. Engrg. 313 (2017) 673–686.

[29] A.R. York, D. Sulsky, H.L. Schreyer, Fluid–membrane interaction based on the material point method, Internat. J. Numer. Methods Engrg. 48 (6) (2000) 901–924.

[30] A. Gilmanov, S. Acharya, A hybrid immersed boundary and material point method for simulating 3D fluid–structure interaction problems, Internat. J. Numer. Methods Fluids 56 (12) (2008) 2151–2177.

[31] J.G. Li, Y. Hamamoto, Y. Liu, X. Zhang, Sloshing impact simulation with material point method and its experimental validations, Comput. & Fluids 103 (2014) 86–99.

[32] D.Z. Zhang, Q. Zou, W.B. VanderHeyden, X. Ma, Material point method applied to multiphase flows, J. Comput. Phys. 227 (6) (2008) 3159–3173.

[33] S.G. Bardenhagen, E.M. Kober, The generalized interpolation material point method, CMES - Comput. Model. Eng. Sci. 5 (6) (2004) 477–495.

[34] P.C. Wallstedt, J. Guilkey, An evaluation of explicit time integration schemes for use with the generalized interpolation material point method, J. Comput. Phys. 227 (22) (2008) 9628–9642.

[35] S. Ma, Material Point Meshfree Methods for Impact and Explosion Problems (Ph.D. thesis), Tsinghua University, 2006.

[36] T. Charlton, W. Coombs, C. Augarde, Igimp: An implicit generalised interpolation material point method for large deformations, Comput. Struct. 190 (2017) 108–125.

[37] N.P. Daphalapurkar, H. Lu, D. Coker, R. Komanduri, Simulation of dynamic crack growth using the generalized interpolation material point (GIMP) method, Int. J. Fract. 143 (1) (2007) 79–102.

[38] J. Ma, H. Lu, R. Komanduri, Structured mesh refinement in generalized interpolation material point (GIMP) method for simulation of dynamic problems, Comput. Model. Eng. Sci. 12 (3) (2006) 213.

[39] A. Sadeghirad, R.M. Brannon, J. Burghardt, A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations, Internat. J. Numer. Methods Engrg. 86 (12) (2011) 1435–1456.

[40] A. Sadeghirad, R.M. Brannon, J. Guilkey, Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces, Internat. J. Numer. Methods Engrg. 95 (11) (2013) 928–952.

[41] D.Z. Zhang, X. Ma, P.T. Giguere, Material point method enhanced by modified gradient of shape function, J. Comput. Phys. 230 (16) (2011) 6379–6398.

[42] M. Steffen, R.M. Kirby, M. Berzins, Analysis and reduction of quadrature errors in the material point method (MPM), Internat. J. Numer. Methods Engrg. 76 (6) (2008) 922–948.

[43] I.K.J. AL-Kafaji, Formulation of a Dynamic Material Point Method (MPM) for Geomechanical Problems (Ph.D. thesis), University of Stuttgart, 2013.

[44] D. Sulsky, M. Gong, Improving the material-point method, in: K. Weinberg, A. Pandolfi (Eds.), Innovative Numerical Approaches for Multi-Field and Multi-Scale Problems, in: Lecture Notes in Applied and Computational Mechanics, vol. 81, Springer, Cham, 2016.

[45] M. Gong, Improving the Material Point Method (Ph.D. thesis), University of New Mexico, 2015.

[46] R. Tielen, High-order Material Point Method (Master's thesis), Delft University of Technology, 2016.

[47] Y. Gan, Z. Sun, Z. Chen, X. Zhang, Y. Liu, Enhancement of the material point method using B-spline basis functions, Internat. J. Numer. Methods Engrg. 113 (3) (2018) 411–431.

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

[49] G. Remmerswaal, P. Vardon, M. Hicks, J.G. Acosta, Development and implementation of moving boundary conditions in the material point method, ALERT Geomaterials (2017) 28.

[50] S.M. Han, H. Benaroya, T. Wei, Dynamics of transversely vibrating beams using four engineering theories, J. Sound Vib. 225 (5) (1999) 935–988.

[51] G.R. Johnson, T.J. Holmquist, Evaluation of cylinder-impact test data for constitutive model constants, J. Appl. Phys. 64 (8) (1988) 3901–3910.