A transport point method for complex flow problems with free surface

Yan Song, Yan Liu & Xiong Zhang

Computational Particle Mechanics

ISSN 2196-4378
Volume 7
Number 2

Comp. Part. Mech. (2020) 7:377-391
DOI 10.1007/s40571-019-00282-9

Springer

Computational Particle Mechanics (2020) 7:377–391
https://doi.org/10.1007/s40571-019-00282-9

A transport point method for complex flow problems with free surface

Yan Song[1] · Yan Liu[1] · Xiong Zhang[1]

Received: 16 April 2019 / Revised: 19 August 2019 / Accepted: 31 August 2019 / Published online: 9 September 2019

Abstract

The material point method (MPM) has been widely used in a broad area of engineering. However, it still suffers from the accuracy problem. One of the most important sources of the accuracy problem is the error of particle quadrature. The discontinuity of the gradient of the shape function causes severe crossing-mesh error. This paper proposes a new transport point method (TPM) which employs a mixed quadrature scheme combining Gaussian quadrature in the internal cells and particle quadrature in the boundary cells. The main distinction between the TPM and MPM is the quantities carried by particles. The transport points in TPM do not have volume and only carry intensive quantities and have dual properties of both Euler and Lagrange. The MLS method is used to reconstruct quantities at cell centers and nodes, and the integral weight is the cell volume. A point rearrangement algorithm is proposed so that the transport points can be added, moved or deleted arbitrarily as long as the accuracy of the reconstructed flow field is maintained, which can be used to eliminate the numerical fracture and impose inlet condition.

Keywords Transport point method · Material point method · Mixed quadrature scheme · Inlet condition

1 Introduction

The material point method (MPM) [1,2] is a meshfree particle method which treats the object as a group of particles moving through a regular background grid. It combines the benefits of both the Eulerian method and the Lagrangian method. On the one hand, since the momentum equation is solved on a regular background every step, the mesh distortion problem in the Lagrangian method is avoided naturally. On the other hand, as the Lagrangian particles carry the mass, momentum and energy and move with the background grid, there is no need to compute the convective term which is necessary in the Eulerian method. Therefore, the MPM does not suffer from the numerical difficulties caused by the convective term such as the asymmetric stiffness matrix. Tracing the boundary of the object is much easier than the Eulerian method as well because of the Lagrangian particles. The MPM has been widely used in a great many problems such as flyer impact [3–8], dynamic fracture [9–15], armor piercing [8,16,17], dam break [8], just to name a few.

However, there still exist a few problems in getting broader use, one of which is the accuracy. The accuracy problems mainly come from three sources: reconstructing quantities on nodes, particle quadrature in cells and boundary representation. Bardenhagen et al. [18] figured out that the discontinuity of the gradient of shape function between adjacent cells leads to the severe crossing-mesh error. Hence, the generalized interpolation material point method (GIMP) was proposed to smooth the shape function. After that, a series of methods with smooth shape functions were proposed such as the uniform GIMP (uGIMP), the contiguous particle GIMP (cpGIMP) [11], the convected particle domain interpolation (CPDI) [19], the affine particle-in-cell (APIC) [20], the dual domain material point (DDMP) [3] and the B-splines material point method (BSMPM) [21]. The other way to eliminate the crossing-method error is to replace the particle quadrature with Gaussian quadrature. Sulsky and Gong [22] constructed the quantities on cell centers with moving least square (MLS) method and employed Gaussian quadrature in cells. Although the rate of velocity convergence was raised to second order, the method did not deal with problems with free surfaces. However, the reconstructing quadrature scheme is still a good way to improve the accuracy of the MPM.

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

✉ Xiong Zhang
xzhang@tsinghua.edu.cn

[1] School of Aerospace Engineering, Tsinghua University, Beijing 100084, People's Republic of China

As a particle method, the MPM also has to face on the numerical fracture problem like other methods such as SPH. The numerical fracture in MPM appears when particles are separated by a cell, which is a common situation in explosion simulation and hypervelocity impact. Ma et al. [23] has proposed a particle adaptive splitting scheme to eliminate the numerical fracture which performs well in explosive driven flyer problem and shaped charge jet problem. However, an artificial parameter α which controls the maximum particle size is necessary and this method cannot ensure the elimination of numerical fracture theoretically. Furthermore, under some circumstances with large deformation and small strain, such as the weakly compressible flow problems, the particle adaptive splitting scheme does not work.

The inlet condition is of great importance in fluid simulation. Obviously, the key point of inlet condition is generating new particles in the computational domain. Monteleone et al. [24] and Zhao et al. [25] have proposed ways to deal with the inlet condition in SPH and MPM, respectively, but when, where and how many particles should be generated have to be artificially specified.

In this paper, a transport point method (TPM) is proposed. Particles only carry intense quantities such as density ρ, momentum density ρv, internal energy in unit initial volume E, stress σ_{ij} and strain ε_{ij}, so that they are referred as transport points. To some degree, those transport points have dual properties of both Lagrange and Euler. On the one hand, they move through the background grid and transport quantities as Lagrangian particles, but on the other hand the transport points take the local value of the field quantities. The dual properties provide us great convenience to arrange the particles without deteriorating the accuracy of the field.

This paper is structured as follows. Section 2 presents the formulation of the transport method. Section 3 introduces the point adaptive algorithm, including the point rearrangement algorithm in Sect. 3.1 and the inlet condition in Sect. 3.3. Section 4 presents several numerical examples to verify the proposed method, and Sect. 5 draws a brief conclusion.

2 Transport point method

The governing equations in the updated Lagrangian frame are discretized on a regular background grid similar to the MPM. To establish the nodal momentum equations, the Gaussian quadrature is employed and the quantities on the Gauss point are reconstructed by MLS method.

2.1 Governing equations

Considering the fluid domain Ω, the basic governing equations in the updated Lagrangian frame are

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho v_{i,i} = 0 \tag{1}$$

$$\sigma_{ij,j} + \rho b_i = \rho \ddot{u}_i \tag{2}$$

$$\rho \dot{e} = \dot{\varepsilon}_{ij}\sigma_{ij}, \tag{3}$$

where ρ is the current density, t is the time, v_i is the velocity, $\ddot{u}_i$ is the acceleration, b_i is the specific body force, $\dot{e}$ is the rate of change of the specific internal energy, σ_{ij} is the Cauchy stress tensor and ε_{ij} is the strain tensor. Einstein summation convention is employed in this paper such that

$$v_{i,i} = \sum_{i=1}^{n_\mathrm{d}} \frac{\partial v_i}{\partial x_i}, \tag{4}$$

where n_d indicates the number of space dimensions.

The boundary conditions are given by

$$\begin{aligned} (n_j\sigma_{ij})|_{\Gamma_\mathrm{t}} &= \bar{t}_i \\ v_i|_{\Gamma_\mathrm{u}} &= \bar{v}_i, \end{aligned} \tag{5}$$

where Γ_t and Γ_u denote the prescribed traction boundary and velocity boundary, respectively. The initial conditions are given by

$$v_i(\boldsymbol{X}, 0) = v_{0i}(\boldsymbol{X}), \quad u_i(\boldsymbol{X}, 0) = u_{0i}(\boldsymbol{X}). \tag{6}$$

The weak form of the momentum Eq. (2) and the boundary conditions Eq. (5) are formulated as

$$\int_\Omega \rho \ddot{u}_i \delta u_i \mathrm{d}V + \int_\Omega \sigma_{ij}\delta u_{i,j}\mathrm{d}V - \int_\Omega \rho b_i \delta u_i \mathrm{d}V - \int_{\Gamma_\mathrm{t}} \bar{t}_i \delta u_i \mathrm{d}A = 0, \tag{7}$$

where δu_i is the virtual displacement.

2.2 Nodal momentum equations

Considering the fluid domain Ω on a regular background grid, the coordinates of an arbitrary point can be approximated as the linear combination of grid nodal coordinates by

$$x_i = \sum_{I=1}^{n_\mathrm{g}} N_I x_{iI}, \tag{8}$$

where the subscript I denotes the quantities associated with the grid node I and n_g is the total number of grid nodes. N_I is the global shape function of node I. Consequently, the displacement and virtual displacement of a point can be approximated as

$$u_i = \sum_{I=1}^{n_\mathrm{g}} N_I u_{iI} \quad \delta u_i = \sum_{I=1}^{n_\mathrm{g}} N_I \delta u_{iI}. \tag{9}$$

Substituting Eq. (9) into the weak form Eq. (7) gives the nodal momentum equation of a grid node I as

$$\dot{p}_{iI} = f_{iI}^{\mathrm{int}} + f_{iI}^{\mathrm{ext}}, \tag{10}$$

where

$$p_{iI} = \int_\Omega \rho N_I N_J \mathrm{d}V \dot{u}_{iJ} = m_{IJ}\dot{u}_{iJ} \tag{11}$$

is the nodal momentum, where

$$m_{IJ} = \int_\Omega \rho N_I N_J \mathrm{d}V \tag{12}$$

is the consistent mass,

$$f_{iI}^{\mathrm{int}} = -\int_\Omega \sigma_{ij} N_{I,j} \mathrm{d}V \tag{13}$$

is the nodal internal force and

$$f_{iI}^{\mathrm{ext}} = \int_\Omega \rho b_i N_I \mathrm{d}V + \int_{\Gamma_\mathrm{t}} \bar{t}_i N_I \mathrm{d}A \tag{14}$$

is the nodal external force.

The consistent mass matrix m_{IJ} can be replaced by the lumped mass matrix as the explicit dynamics algorithm does. As a result, the nodal momentum p_{iI} can be simplified as

$$p_{iI} = m_I \dot{u}_{iI}, \tag{15}$$

where

$$m_I = \sum_{J=1}^{n_\mathrm{g}} m_{IJ} = \int_\Omega \rho N_I \mathrm{d}V \tag{16}$$

is the nodal lumped mass.

2.3 Reconstruction

In the TPM, the flow field is discretized as a set of transport points which carry the field quantities, such as density ρ, velocity $\dot{u}_i$ and the internal energy in unit initial volume E. Particularly, points carry the specific volume v instead of the volume. The quantities of an arbitrary point $\boldsymbol{x}$ can be reconstructed with the moving least square (MLS) method from the transport point quantities as

$$\rho(\boldsymbol{x}) = \sum_{p=1}^{n_\mathrm{p}} \Phi(\boldsymbol{x}_p; \boldsymbol{x})\rho_p, \tag{17}$$

where $\Phi(\boldsymbol{x}_p; \boldsymbol{x})$ is the MLS shape function associated with transport point p and n_p is the number of transport points in the support domain. In this method, the MLS weight function is defined on a square (cubic in 3D space) support domain to reduce the computational cost on bucket searching, so that

$$\Phi(\boldsymbol{x}_p; \boldsymbol{x}) = \phi(x_p; x)\phi(y_p; y), \tag{18}$$

where $\phi(x_p; x)$ is a 1D MLS weight function.

The conservation of mass and momentum must be considered when reconstructing the node quantities. The MPM maintains laws of conservation by the unit decomposition of the shape function. However, because the points in the TPM do not carry volume, the laws of conservation must be further discussed. The conservation laws require the discretization and reconstruction method satisfying

$$\begin{aligned} \int_\Omega \rho \mathrm{d}V &= \sum_{I=1}^{n_\mathrm{g}} m_I \\ \int_\Omega \rho \dot{u}_i \mathrm{d}V &= \sum_{I=1}^{n_\mathrm{g}} p_{iI}. \end{aligned} \tag{19}$$

Note that the Gaussian quadrature and the MLS method can guarantee the conversation laws Eq. (19). Because the nodal mass m_I and momentum p_{iI} are computed by Gaussian quadrature, the quantities of Gauss points are reconstructed by the MLS method which can guarantee the rate of convergence at a proper order. Therefore, the quantities which are constructed are the density ρ and the momentum density $\rho\dot{u}_i$ instead of the density ρ and the velocity $\dot{u}_i$, namely

$$\begin{aligned} \rho_I &= \sum_{p=1}^{n_\mathrm{p}} \Phi_I(\boldsymbol{x}_p)\rho_p \\ (\rho\dot{u}_i)_I &= \sum_{p=1}^{n_\mathrm{p}} \Phi_I(\boldsymbol{x}_p)[(\rho\dot{u}_i)_p] \\ \dot{u}_{iI} &= \frac{(\rho\dot{u}_i)_I}{\rho_I}. \end{aligned} \tag{20}$$

Employing Gaussian quadrature in Eq. (16) gives

$$m_I = \sum_{c=1}^{n_\mathrm{c}} \rho(\boldsymbol{x}_c) N_I(\boldsymbol{x}_c) V_c, \tag{21}$$

where $n_\mathrm{c} = 2^{n_\mathrm{d}}$ is the number of cells connecting to node I, V_c is the cell volume and $\boldsymbol{x}_\mathrm{c}$ is the coordinates of the cell center. In the regular background grid, V_c is a constant denoted as V_e and $N_I(\boldsymbol{x}_\mathrm{c}) = 1/n_\mathrm{c}$. Thus, the nodal mass can be simplified as

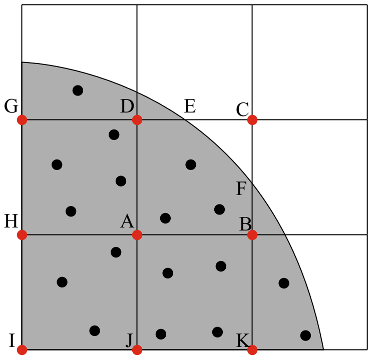


Fig. 1 Fluid boundary cutting a cell ABCD

$$m_I = \left[\frac{1}{n_c} \sum_{c=1}^{n_c} \rho(\boldsymbol{x}_c) \right] V_e = \tilde{\rho}_I V_e, \tag{22}$$

where $\tilde{\rho}_I$ is the average density of node I. If the linear basis is used, the average density can be obtained directly by reconstructing on node I, namely

$$\tilde{\rho}_I = \sum_{p=1}^{n_p} \Phi_I(\boldsymbol{x}_p)\rho_p. \tag{23}$$

The internal force f_{iI}^{int} can be computed as

$$f_{iI}^{\text{int}} = -\sum_{c=1}^{n_c} \sigma_{ij}(\boldsymbol{x}_c) N_{I,j}(\boldsymbol{x}_c) V_e. \tag{24}$$

2.4 Mixed quadrature scheme

In Eqs. (21) and (24), the Gaussian quadrature is evaluated assuming that the cells connecting to the node are full of material, which is not always the case.

The case where the boundary of the fluid cuts a cell is shown in Fig. 1, where the capital letters indicate points in the grid and the cell is indicated by its connecting nodes. For example, the cell at the lower left corner can be expressed as 'IJAH.' The domain of integration associated with node A in Eq. (16) is no longer the surrounding four cells. If a cell is not fully filled like cell ABCD in Fig. 1, the volume of the fluid in it (the shaded area ABFED) should be used as the integral weight instead of the cell volume.

There are many effective methods to estimate the volume of ABFED. A level-set function can be employed to reconstruct the boundary and cuts the cell ABCD so that the area of the polygon ABFED can be computed accurately. However, the polygon cutting in 3D case is a time-consuming task. A mixed quadrature scheme [26] is employed in this paper. As shown in Fig. 1, there are three kinds of cells: internal cell which is full of fluid (HIJA), boundary cell which is cut by the fluid boundary (ABCD) and empty cell. The quadrature method in cells differs as well: The Gaussian quadrature is used in internal cells, while particle quadrature is used on boundary cells. As a result, the mass of the node A has the form as

$$m_{\text{A}} = \frac{3}{4}\rho_{\text{A}} V_e + \sum_{p=1}^{n_{\text{p}}^{\text{c}}} \rho_p V_p N_p, \tag{25}$$

where n_{p}^{c} is the number of points located in the cell ABCD, ρ_p and V_p are the density and volume of transport point p, respectively. Unlike the material point method, the transport point does not possess volume. As a result, the volume V_p must be determined. A simple but efficient way is proposed here.

Considering the transport points in cell ABCD shown in Fig. 1, the volume of each transport point should be the space it occupies. Assuming that the shape of transport points is circular (sphere in 3D space) with radius r, the volume can be obtained by $V = \pi r^2$ $(V = \frac{4}{3}\pi r^3 \text{in3Dspace})$. It is reasonable to assume that the points are close-packed in the flow field. So a proper estimation of the radius is half of the nearest distance between the point p and others, namely

$$r_p = \frac{1}{2} \min_{p \neq q} |\boldsymbol{x}_p - \boldsymbol{x}_q|. \tag{26}$$

As the loss of the boundary information, the quadrature accuracy in boundary cells is lower than that in internal cells. However, the rate of convergence of this method is not lower than the material point method for they share the same quadrature scheme in boundary cells.

Another significant part of the mixed quadrature scheme is the method to distinguish the cell type. The traditional MPM method takes the particles volume as a criterion [26]. If the particles in a cell satisfy

$$\sum_{p=1}^{n_{ep}} V_p < \beta V_e, \tag{27}$$

where n_{ep} is the number of particles in a cell and β is a factor which is usually set around 0.9 [27], the criterion cannot be used in the TPM because the transport points do not carry volume. In Sect. 3, a method is proposed to detect the cell type and numerical fracture.

2.5 Stress and energy updating

Because the transport points do not carry volume, the stress and energy updating progress is different from the MPM. The increment of the strain and spinor of a transport point p can be calculated by

$$\Delta\varepsilon_{ijp} = \frac{1}{2}(N_{Ip,j}v_{iI} + N_{ip,i}v_{jI})\Delta t \tag{28}$$

$$\Delta\Omega_{ijp} = \frac{1}{2}(N_{Ip,j}v_{iI} - N_{Ip,i}v_{jI})\Delta t, \tag{29}$$

where v_{iI} is the velocity of cell node I.

Because the transport points do not carry volume, we update the specific volume V and the internal energy in unit initial volume E directly. Assuming that the quantities in step n are known, the specific volume can be updated by

$$V^{n+1} = V^n\left(1 + \Delta\varepsilon_{ii}^{n+1/2}\right) \tag{30}$$

and the increment of volumetric strain $\Delta\varepsilon_{ii}^{n+1/2}$ can be calculated by Eq. (28). So the density is updated by

$$\rho^{n+1} = \frac{\rho^n}{1 + \Delta\varepsilon_{ii}^{n+1/2}}. \tag{31}$$

Assuming that the initial volume of a transport point is V_0, the internal energy is $e^n = E^n V_0$. Consequently, the internal is updated by

$$e^{n+1} = e^n + \dot{e}^{n+1/2}\Delta t^{n+1/2}, \tag{32}$$

where the rate of internal energy can be calculated by

$$\dot{e}^{n+1/2} = \frac{1}{2}V_0(V^n + V^{n+1})(s_{ij}^{n+1/2}\dot{\varepsilon}_{ij}^{n+1/2} - p^{n+1/2}\dot{\varepsilon}_{ii}^{n+1/2}), \tag{33}$$

where s_{ij} is the deviatoric stress. By substituting Eq. (33) into Eq. (32), we can get the internal energy updating formulation

$$E^{n+1} = E^n + \frac{1}{2}\left(V^n + V^{n+1}\right)\left(s_{ij}^{n+1/2}\Delta\varepsilon_{ij}^{n+1/2} - p^{n+1/2}\Delta\varepsilon_{ii}^{n+1/2}\right). \tag{34}$$

With the internal energy updated, the stress can be updated directly by the EOS and strength model directly.

3 Point rearrangement algorithm

The numerical fracture would cause severe accuracy problem in MPM simulation. Ma et al. [23] proposed the adaptive particle splitting scheme to deal with the problem. The basic idea is to split a specific particle with extreme expansion. The simulation result is significantly improved under some circumstances. However, numerical fracture may occur without extreme expansion and stretching, where the adaptive particle splitting scheme cannot work.

The key point of TMP is regarding the object as a field which is marked and transported by points. Transport points only carry intensive quantities. Therefore, the points are flexible and free to move as long as the accuracy of the field is ensured. The flexibility makes it possible to rearrange the transport point.

In this section, we propose the point arrangement algorithm which can rearrange points in a grid cell. The second part introduces the numerical fracture detection which can detect the cell type. At last, a method to impose inlet condition based on the point rearrangement algorithm is proposed.

3.1 Point rearrangement algorithm

In the transport point method, the property that points only carry intensive quantities brings a lot of benefits. Because the quantities in the nodal momentum equation Eq. (2) are reconstructed with the MLS method, the accuracy of the simulation would be maintained as long as the MLS reconstruction ensures required accuracy. In another perspective, the accuracy would not be deteriorated as long as the points are able to represent the flow field.

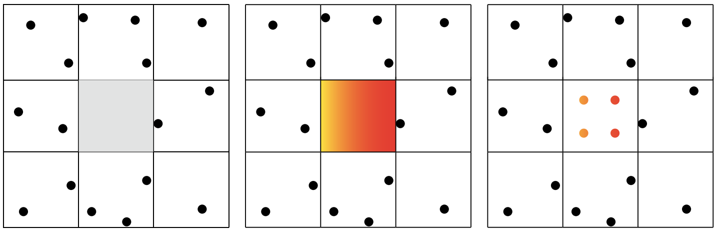
As a result of the employment of the mixed quadrature scheme, the numerical fracture would influence the accuracy severely. On the other hand, the points aggregation would waste computational resources. Because of the flexibility of transport points in this method, it is possible to rearrange the transport points without deteriorating the accuracy of the simulation.

The point rearrangement algorithm is a particle adaptive algorithm similar to the mesh adaptive method used in FEM. The flow chart of the algorithm is shown in Fig. 2.

The first step is to detect the numerical fracture as shown in Fig. 2a which will be discussed in Sect. 3.2. The cell where the numerical fracture occurs must be surrounded by full cells so that we can reconstruct a flow field in it with MLS as shown in Fig. 2b. (Linear basis functions are used.) The last step is to generate new points in the cell. The quantities on the new points are set as the local value of the flow field as shown in Fig. 2c. Because the accuracy is controlled by the MLS approximation, the positions of the new transport points are flexible. Usually, we put one transport point at the cell center or put four transport points (eight in 3D space) at the Gauss quarter points.

We can also reconstruct a fluid field in the cell where there are too many transport points and generate several new points to reduce the computational burden. The original points are deleted to free the memory they occupied.

The point rearrangement algorithm can balance the points distribution and eliminate the numerical fracture. It is necessary for problem with large deformation.



a Numerical fracture **b** Linear field **c** Generate points

Fig. 2 Point arrangement algorithm

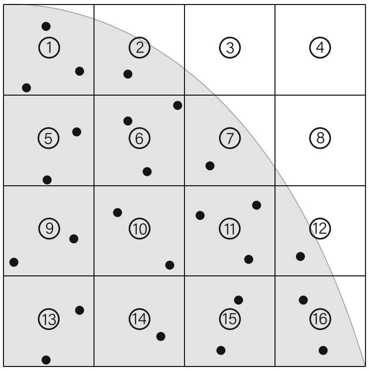


Fig. 3 Boundary criteria

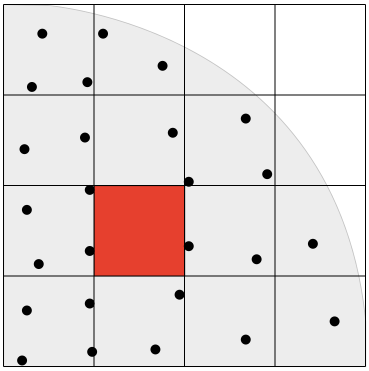


Fig. 4 Numerical fracture

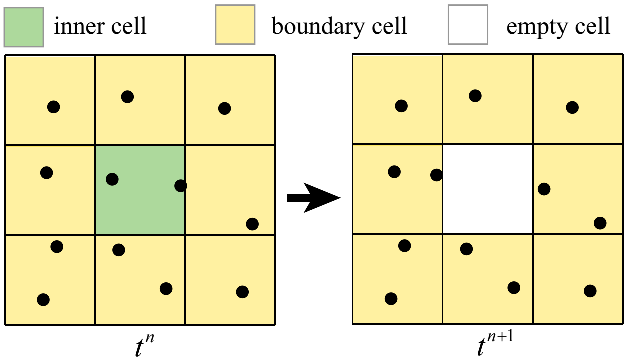


Fig. 5 Internal fracture

3.2 Numerical fracture detection

In the above discussion, a significant procedure is to judge whether a cell is a boundary cell to determine the quadrature scheme. We need an efficient method to distinguish the different type of cells rather than to reconstruct the boundary accurately.

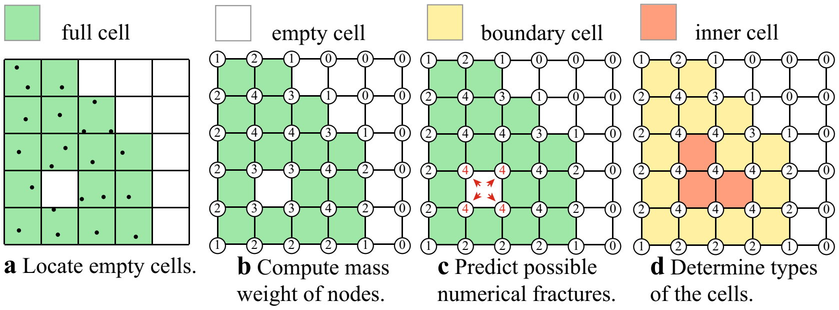
First of all, a judging criteria should be proposed. As mentioned in Sect. 2.4, the criteria in essence should judge whether a cell is full of fluid or not. Therefore, the information of the cell and its adjacent cells must be used. An example shown in Fig. 3 may help us to figure out the criteria where the cells are numbered from 1 to 16. The gray region represents the flow field which is discretized into transport points shown as dark dots. Obviously, a cell without transport points located in it is the 'empty cell' defined in Sect. 2.4, such as cells 3, 4 and 8. The other cells should be distinguished according to their surrounding cells. A boundary cell must be adjacent to at least one empty cell. For example, the empty cell 8 is adjacent to cell 7 so that cell 7 is a boundary cell. The cells whose neighbors are not empty are internal cells such as cell 10.

The above criteria are based on an implicit assumption: An internal cell must contain points in it which is not always the case. As Fig. 4 shows, the red cell is inside of the flow field but has no points in it because of the numerical fracture. Moreover, the numerical fracture makes the surrounding cells be considered as boundary cells, and thereby the particle quadrature is employed. Therefore, the numerical fracture leads to the false hole in the domain of integral.

Ma et al. [23] proposed an adaptive particle splitting scheme in the MPM to eliminate the numerical fracture. In the adaptive particle splitting scheme, a particle is split into two particles when its accumulative strain in one direction exceeds a specified value. Its mass, volume and internal energy are halved to each new particle, while other variables such as stress, strain and temperature are copied to the new particles directly. This method suffers from the accuracy loss in the numerical fracture area and is unable to completely eliminate the numerical fracture.

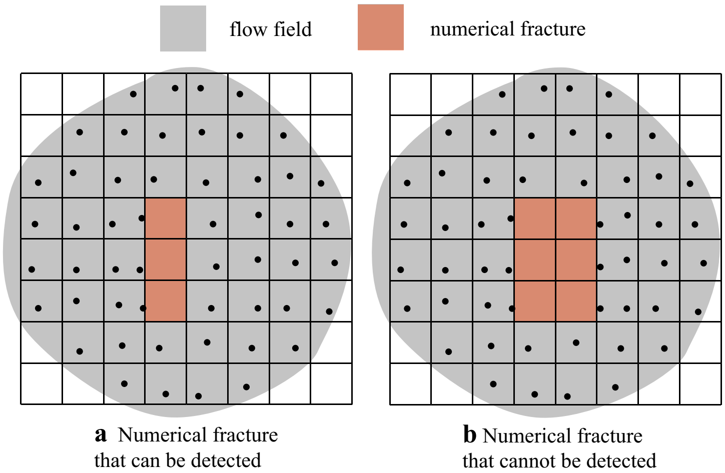
It is difficult to distinguish the numerical fracture from the physical fracture. However, we can conclude several simple rules to recognize the numerical fracture inside of the flow field. An obvious rule is that an internal cell should not

Fig. 6 Boundary detection algorithm



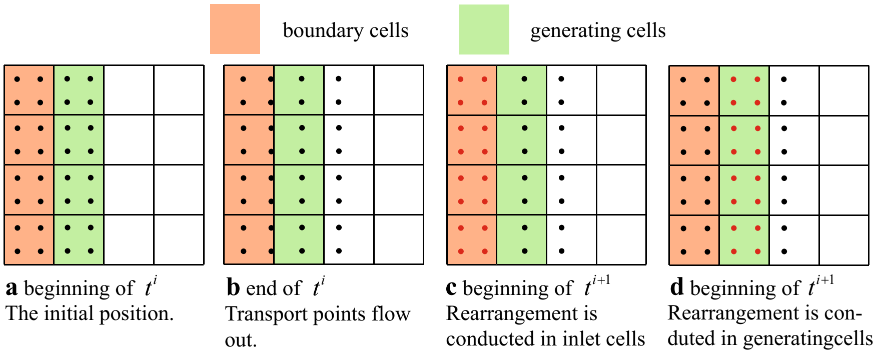
a Locate empty cells. **b** Compute mass weight of nodes. **c** Predict possible numerical fractures. **d** Determine types of the cells.

Fig. 7 Some numerical fracture cases



a Numerical fracture that can be detected **b** Numerical fracture that cannot be detected

Fig. 8 Inlet conditions



a beginning of t^i The initial position. **b** end of t^i Transport points flow out. **c** beginning of t^{i+1} Rearrangement is conducted in inlet cells **d** beginning of t^{i+1} Rearrangement is conduted in generatingcells

become an empty cell in the next step as the displacement of each point is limited to a cell length by the time step. As a result, if an internal cell turns into empty cell in the next time step, the numerical fracture occurs as is shown in Fig. 5. Then, the point rearrangement process shall be conducted in those cells.

In essence, the numerical fracture occurs in empty cells where there should exist transport points. So we have to decide which cells should possess transport points. A reasonable idea is that a cell tends to possess transport points in it if its nodes have positive mass. The algorithm is shown as follows where a node's mass weight is defined as the number of connecting cells which possess transport points.

1. Locate the position of empty cells. Loop through all the transport points and put them in cells. Then, loop through all the cells to find out empty cells (Fig. 6a).
2. Loop through all the nodes to compute their mass weight. Figure 6b shows the mass weight of all the nodes.
3. Predict the numerical fracture. Loop through empty cells. If all of an empty cell's nodes have nonzero mass weight, this cell is possible to be full of fluid. Thus, the mass contribution of this cell should be considered (Fig. 6c) as if it is an internal cell.
4. Determine the numerical fracture. Loop through empty cells. If an empty cell's nodes have full mass weight (equal to 4 in 2D and 8 in 3D), the numerical fracture occurs. Then, the points arrangement shall be conducted in this cell.

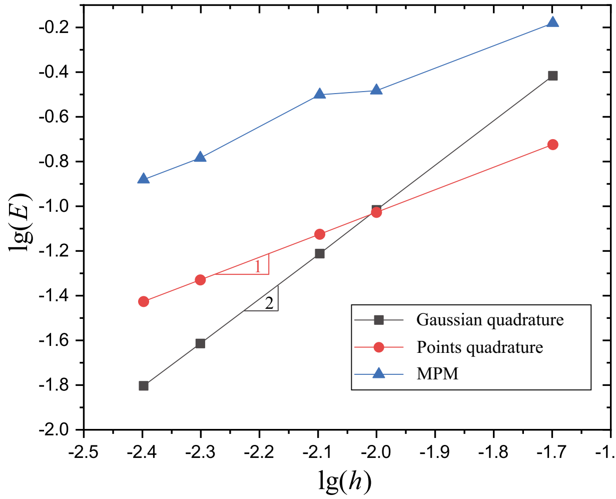


Fig. 9 Rate of convergence of pressure

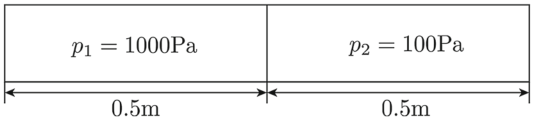


Fig. 10 Riemann problem

5. Determine the cells' type as shown in Fig. 6d.

The above method can detect numerical fracture not only in single cell but also in a narrow band of connecting cells as shown in Fig. 7a. Although it fails in a large area of numerical fracture (Fig. 7b), the case rarely happens because of the limit of the time step. The time complexity is $O(n)$, where n is the total number of cells. Considering the first step is putting points into cells which is also needed in the reconstructing process, the extra time cost is acceptable.

3.3 Inlet conditions

The points arrangement algorithm brings many advantages. For example, it can be adopted to impose the inlet boundary conditions.

Inlet condition is significant in channel flow problems and flow around a solid body. Many meshless methods impose the boundary conditions by adding particles into the domain at intervals and generate some ghost particles to maintain the inlet pressure. When and where to generate a new particle will influence the result, and it is determined by the inlet velocity. The point rearrangement algorithm makes us able to control the flow field directly without relying on the inlet particles.

The basic idea to impose the inlet boundary conditions is maintaining a steady flow field in the inlet region. We can reconstruct the points in the inlet region as a specified flow field so that a steady boundary conditions are imposed on transport points. Figure 8 shows that two columns of cells are set at the inlet boundary. Boundary cells maintain the inlet flow field, and generating cells put new transport points in the computational domain. As shown in Fig. 8a, the computational domain extends two columns of cells to simulate the inlet flow. Transport points are initialized in the cells and involved in the computational progress. At the end of each time step, the points are updated as shown in Fig. 8b and some additional operations will be performed at the beginning of the next time step. Firstly, we fix the inlet conditions. As shown in Fig. 8c, we delete the transport points in the boundary cells and put some new transport points at the initial position whose quantities are set as the local value of the inlet flow. This operation maintains a steady inlet condition for the computational domain. Figure 8d represents the points generating process. After the transport points in boundary cells are reset, we examine the number of transport points in each generating cell. If the transport points number is less than the initial number which means some transport points have flowed into the computational domain, the point rearrangement algorithm will be performed in the cell. This operation ensures the continuity of the inlet flow. New transport points are generated through the point arrangement progress automatically.

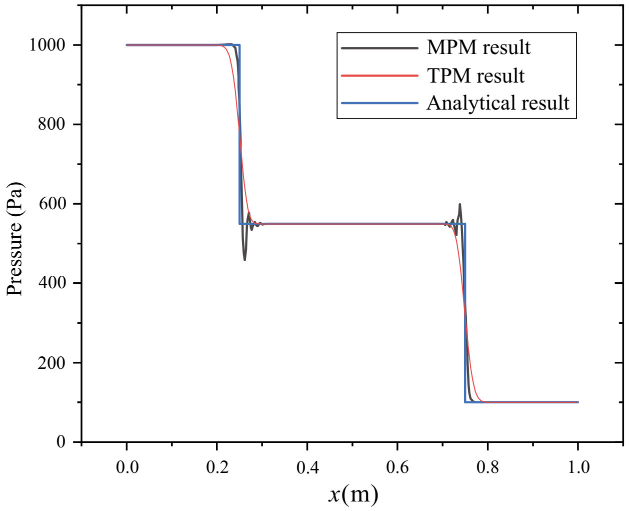
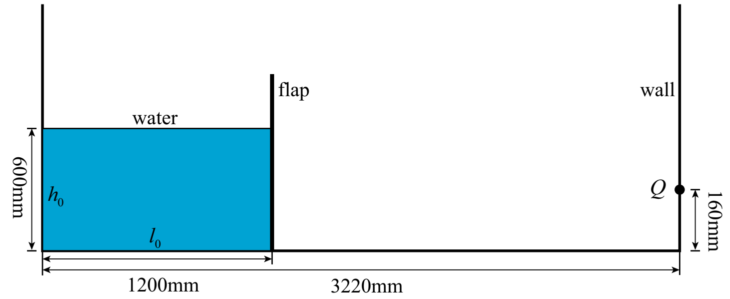


Fig. 11 Pressure distribution at $t = 5.0$ms

4 Numerical results

To verify the proposed transport point method, several examples are presented in this section. The first two examples are investigated to test the accuracy and the rate of convergence. Then, a dam break example is given to verify the accuracy and the ability to eliminate the numerical fracture. And the last example demonstrates the correctness of inlet conditions.

Fig. 12 Dam break problem



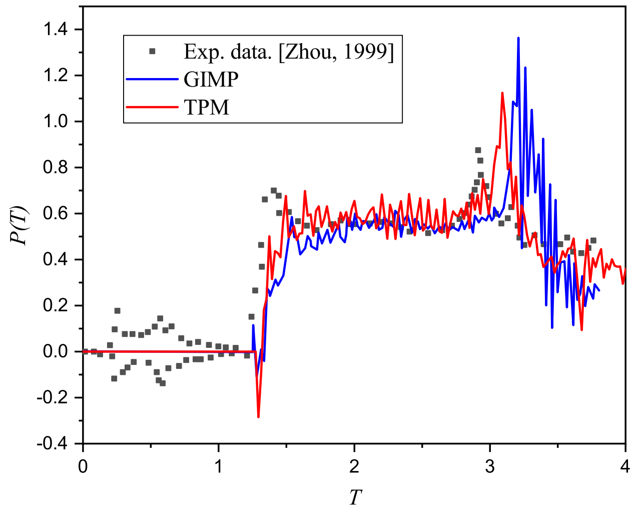


Fig. 13 Pressure at point Q

4.1 Rate of convergence

A convergence test is investigated in this part. The following example shows the pressure rate of convergence is 1 and reaches 2 with superconvergent patch recovery (SPR), which is equal to the results of FEM.

A sine wave example is investigated here. A computational domain with size of 0.1 × 1.0 is full of water. The weakly compressible EOS

$$p = (\rho - \rho_0)c_0^2 \tag{35}$$

is used with the initial density $\rho_0 = 1000$ and the artificial sound speed $c_0 = 50$. The initial pressure is set as 0, and a sinusoidal initial velocity is given as

$$v(x) = \frac{2\pi A}{T} \sin\left(\frac{2\pi x}{L}\right), \tag{36}$$

where $L = 1.0, \quad A = 0.001, \quad T = 0.02$. The analytical pressure and velocity can be given as

$$\tilde{v}(x,t) = \frac{2\pi A}{T} \sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi t}{T}\right) \tag{37}$$

$$\tilde{p}(x,t) = -\rho c_0^2 \frac{2\pi A}{L} \cos\left(\frac{2\pi x}{L}\right) \sin\left(\frac{2\pi t}{T}\right). \tag{38}$$

To measure the error of the numerical result, the $L2$-norm of error is computed as

$$E(t) = \sqrt{\int_{\Omega} \left[a(x,t) - \tilde{a}(x,t)\right]^2, \mathrm{d}V} \tag{39}$$

where $\tilde{a}(x,t)$ is the analytical solution and $a(x,t)$ is the numerical result. Equation (39) can be evaluated by using the point quadrature as

$$E(t) = \sqrt{\sum_{p=1}^{n_\mathrm{p}} \left[a(x_p,t) - \tilde{a}(x_p,t)\right]^2 V_p} \tag{40}$$

or Gaussian quadrature based on cell centers as

$$E(t) = \sqrt{\sum_{c=1}^{n_\mathrm{c}} \left[a_c(t) - \tilde{a}(x_c,t)\right]^2 V_c}. \tag{41}$$

In Eq. (40), n_p indicates the number of transport points, x_p and V_p denote the coordinates and volume of transport point p. In Eq. (41), a_c is the quantity reconstructed from the numerical result on points with Eq. (17). The latter virtually is a SPR scheme which would increase the rate of convergence of stress by one.

The error of a dynamic numerical method mainly comes from two parts: time integral and numerical quadrature scheme in each step. The former increases with time and would deteriorate the rate of convergence which is determined by the latter. Therefore, the time step must be fixed and all cases must share the same and small enough simulation time.

In this paper, five cases with different cell sizes are tested. The cell sizes are chosen as 0.004, 0.005, 0.008, 0.010 and 0.020, respectively. Their time steps are set to 0.0005 and 10 time steps are simulated. The error of pressure is computed with Gaussian quadrature and points quadrature. The result is shown in Fig. 9. The horizontal axis is the logarithm of the cell size h, and the vertical axis is the logarithm of the

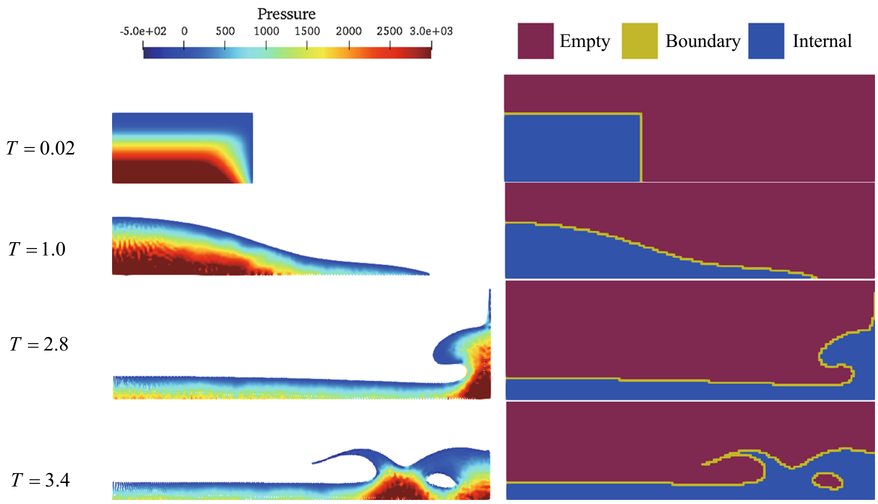


Fig. 14 Pressure nephogram and cell type nephogram

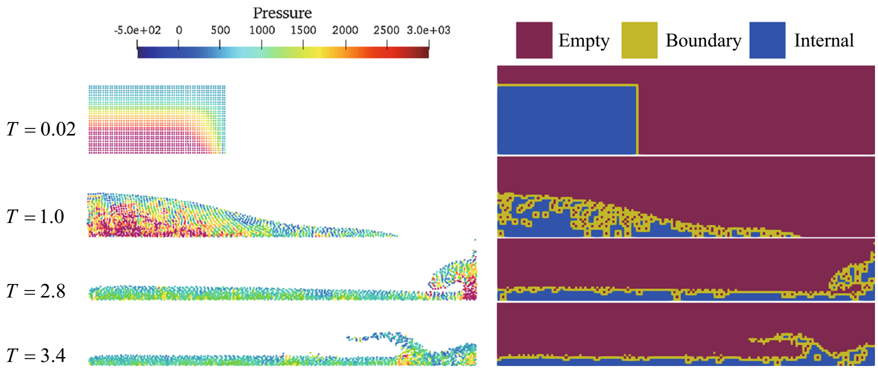


Fig. 15 Configurations with grid size 20mm × 20mm

pressure error. The slope of the error curve represents the rate of convergence. The result shows that the MPM does not represent a uniform rate of convergence, and its error is much higher than the TPM. The error of TMP computed by points quadrature has a first-order rate of convergence, and the Gaussian quadrature reaches the second order.

4.2 Riemann problem

A Riemann problem is investigated in this part to investigate the accuracy of the transport point method. A computational domain with a length of 5m is shown in Fig. 10. The initial condition is prescribed as a piecewise field by

$$p = \begin{cases} 1000 & x < 0.5 \\ 100 & x \geq 0.5. \end{cases} \tag{42}$$

The weakly compressible EOS with the artificial sound speed $c_0 = 50\text{m/s}$ and the initial density $\rho_0 = 1000\text{kg/m}^3$ is employed here to simulate water in the computational domain. An elastic wave will be excited on the interface and spread toward both ends of the domain at the speed of c_0. The analytical solution of pressure distribution at time $t < 10\text{ms}$ is given by

$$p = \begin{cases} 1000 & x < 0.5 - c_0 t \\ 100 & x > 0.5 + c_0 t \\ 550 & \text{else.} \end{cases} \tag{43}$$

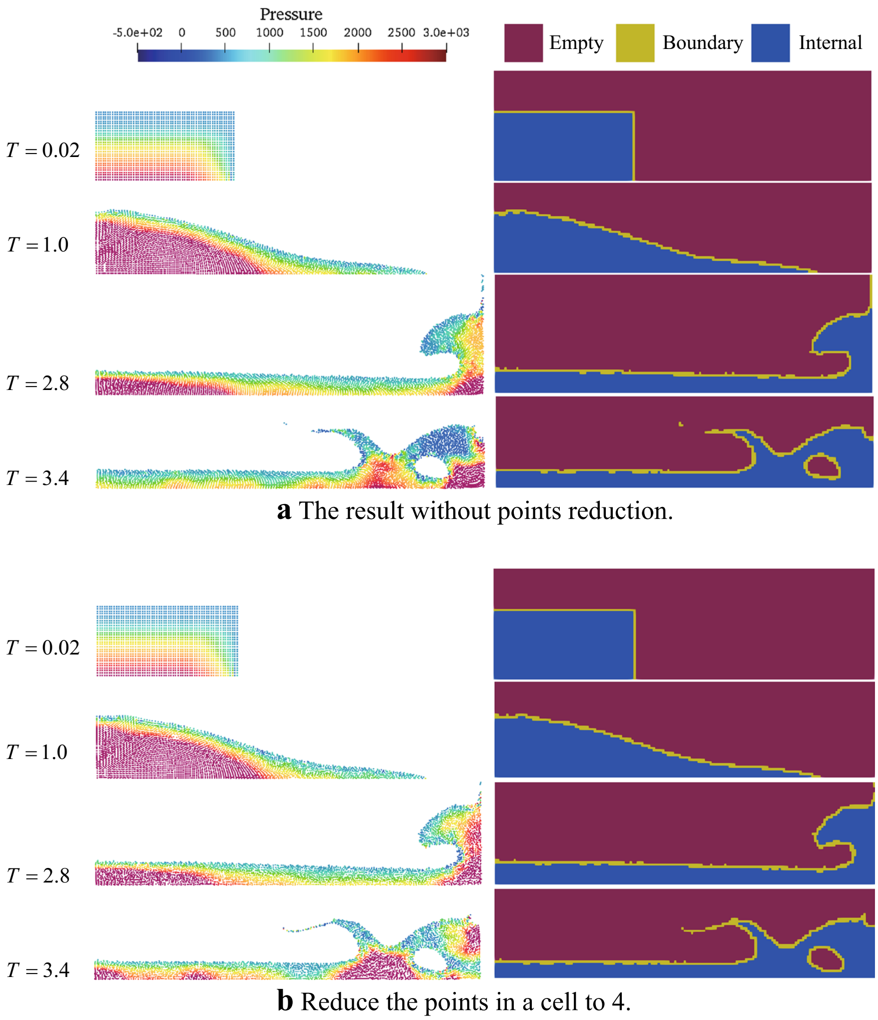
The MPM suffers from the volumetric locking problem when simulating the weakly compressible material. Chen et al. [28] proposed a v-p formulation which smoothens the pressure and avoids the volumetric locking. A slope limiter is required in the MPM to limit the pressure oscillation.

We make a comparison between the MPM and TPM in Riemann problem. The size of the background grid cell is set as $0.001\text{m} \times 0.001\text{m}$, and 2×2 transport points are put in each cell. The v-p formulation without the slope limiter is employed in both MPM and TPM. The pressure distribution at $t = 5.0\text{ms}$ is shown in Fig. 11. The result of the MPM shows a severe pressure oscillation, but the TPM does not. Because the TPM has a smoother pressure field than the MPM, the slope limiter is unnecessary for TPM.

4.3 Dam break

Dam break problem is a typical weakly compressible flow problem with free surface which is usually used to test a new numerical method. A schematic drawing of the prob-

Fig. 16 Pressure and cell type nephogram with point rearrangement algorithm activated



a The result without points reduction.

b Reduce the points in a cell to 4.

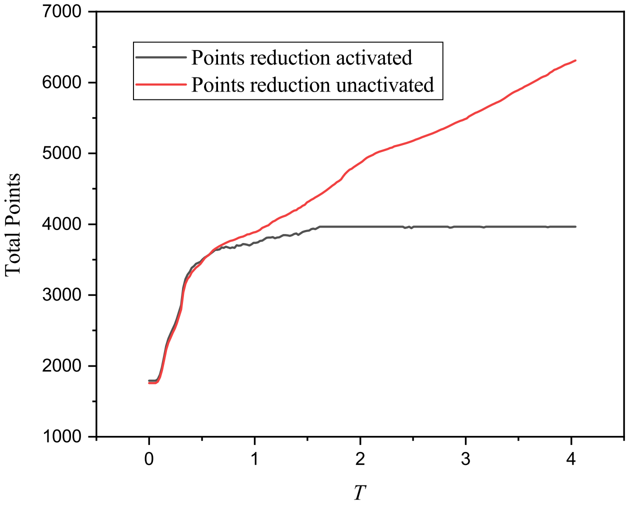


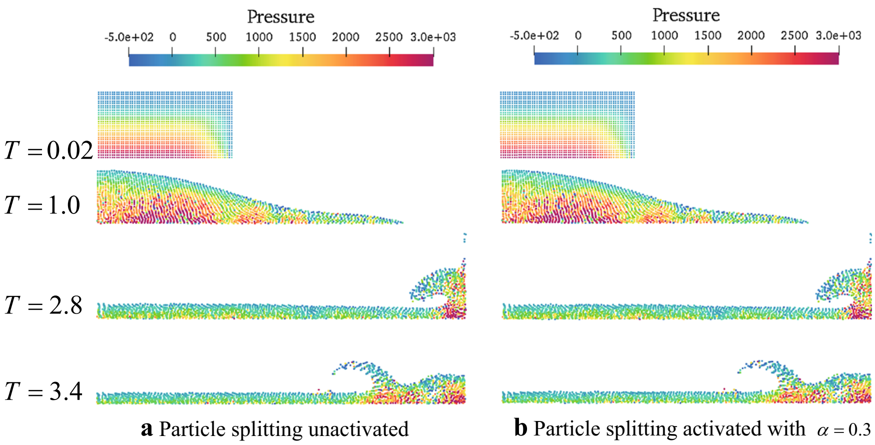
Fig. 17 Points number curve

lem is shown in Fig. 12. The water with depth $h_0 = 600\text{mm}$ and width $l_0 = 1200\text{mm}$ is blocked at the left of the computational domain. The domain is 3220mm in width, and a rigid wall is set at the right side. At the time $t = 0\text{ms}$, the flap is lifted quickly and the water flows to the right side because of the gravity. The initial density of the water is $\rho_0 = 1000\text{kg/m}^2$, and the gravity is $g = 9.8\text{m/s}^2$. The weakly compressible EOS Eq. (35) is employed with the artificial sound speed $c_0 = 50\text{m/s}$. To smoothen the free surface, the strength of extension of water is set to 5000Pa, namely the pressure $p \geq -5000\text{Pa}$. The $v - p$ formulation is also employed here to resist the volumetric locking.

A sampling point Q is set on the wall at the height of 160mm. In order to compare the result with experiments [29, 30] and other numerical methods [28], the following non-dimensional parameters are defined:

$$T = t\sqrt{\frac{h_0 g}{l_0^2}}, \qquad P(T) = \frac{p(T)}{\rho g h_0}. \tag{44}$$

First of all, a dense particle distribution is tested to provide a benchmark for the next several cases. The grid cell size is 20mm × 20mm. The water is discretized by 28800 particles



a Particle splitting unactivated **b** Particle splitting activated with $\alpha = 0.3$

Fig. 18 Pressure nephogram of MPM method with adaptive particle splitting scheme

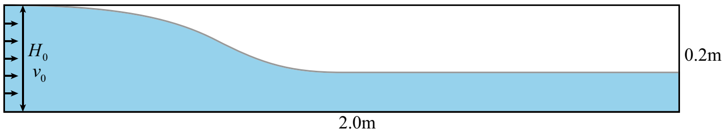


Fig. 19 Steady flow with gravity

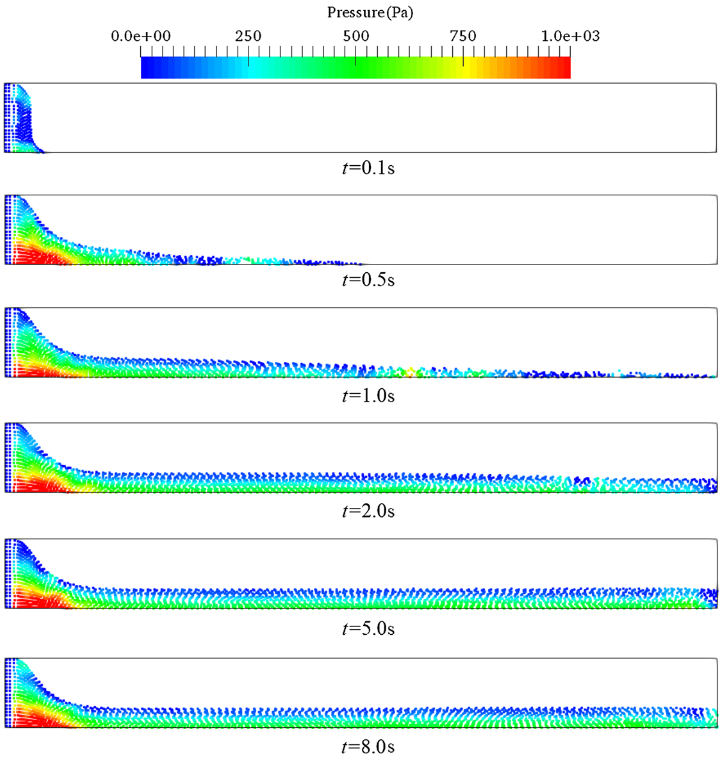
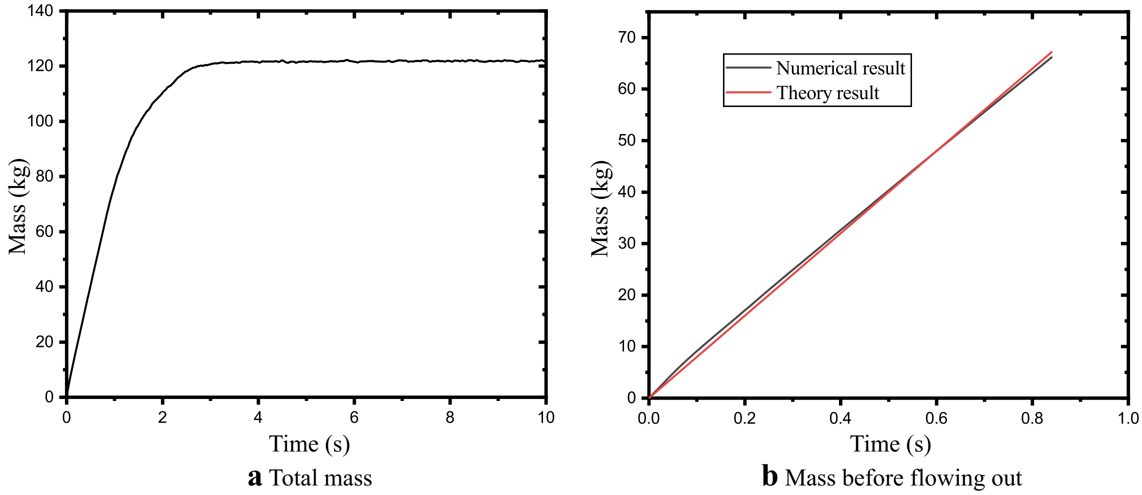


Fig. 20 Numerical result of steady flow

with a size of 5mm × 5mm. At time $t = 0$ms, 16 particles are placed in each grid cell in the water domain. Figure 13 shows the time evolution of the pressure at point Q. The result of the TPM as well as experiment [29] and GIMP [28] are compared. At $T = 1.24$, the water in the TPM and GIMP reaches the point Q nearly at the same time, but the pressure peak at $T = 1.5$ of the TPM is closer to the experimental result. The pressure of the TPM reaches the second peak at $T = 3.1$, which is also closer to the experiment than the GIMP result.

Figure 14 shows the pressure nephogram at $t = 0.02,\ 1.0$, 2.8 and 3.4, respectively, and the cell type where red, yellow and blue represent empty cell, boundary cell and internal cell, respectively. Owing to the dense distribution of the initial water points, there is no numerical fracture in the dam break process.

If we reduce the number of points and cells, numerical fracture may causes severe noise in the result. In the next case, the size of grid cells is 20mm × 20mm and one point



a Total mass

b Mass before flowing out

Fig. 21 Total mass curve

is set in the cell center in each grid cell in the initial water domain. Figure 15 shows the configurations at different time steps. The cell type nephogram shows severe numerical fracture. Because of the numerical fracture, the height of the rebounded wave is lower than the result shown in Fig. 14.

The point rearrangement algorithm is able to eliminate the numerical fracture. Figure 16a shows the result with the point rearrangement algorithm activated. It is obviously that the rebounded wave is higher than the result shown in Fig. 15, and the outline is similar to Fig. 14. The cell type nephogram shows that the point rearrangement algorithm can thoroughly eliminate the numerical fracture in this case.

The point arrangement algorithm can also reduce the points number to save the memory and computational cost. A maximum point number N_m can be set which indicates the number of points in a cell should not exceed it. If a cell is detected to have many points, the arrangement algorithm will assign a new point in the cell center, while the original points will be deleted. Figure 16b is the configurations with $N_\mathrm{m} = 4$, and Fig. 17 shows the points number curve with and without points reduction method. The memory cost stops growing at $T = 1.64$, and the total time cost is reduced by 15% (from 172.6s to 146.3s) with the points reduction method.

Furthermore, a similar numerical model is tested with MPM. The water domain is discretized with particles spacing 20 mm, and the grid cell size is set as 20mm × 20mm. A case in which the adaptive particle splitting scheme [23] is activated is compared. Figure 18 shows the pressure nephogram at $T = 0.02$, $T = 1.0$, 2.8 and 3.4. In comparison with Figs. 14 and 15, the numerical fracture dissipates the energy and lowers the rebounded wave. However, the situation does not improve with the adaptive particle splitting scheme activated. For the reason that the pressure $p \geq 5000\mathrm{Pa}$ is subjected to the tensile strength, according to the weakly compressible EOS Eq. (35), the density $\rho \geq 1.002\mathrm{kg/m}^3$. Then, the volume of particle is

$$V = \frac{\rho_0}{\rho} V_0 \leq 1.002 V_0, \tag{45}$$

where V_0 and ρ_0 are the initial volume and density, respectively. Hence, the particle length in the ith direction

$$L_i \leq \sqrt[3]{1.002} L_0 \tag{46}$$

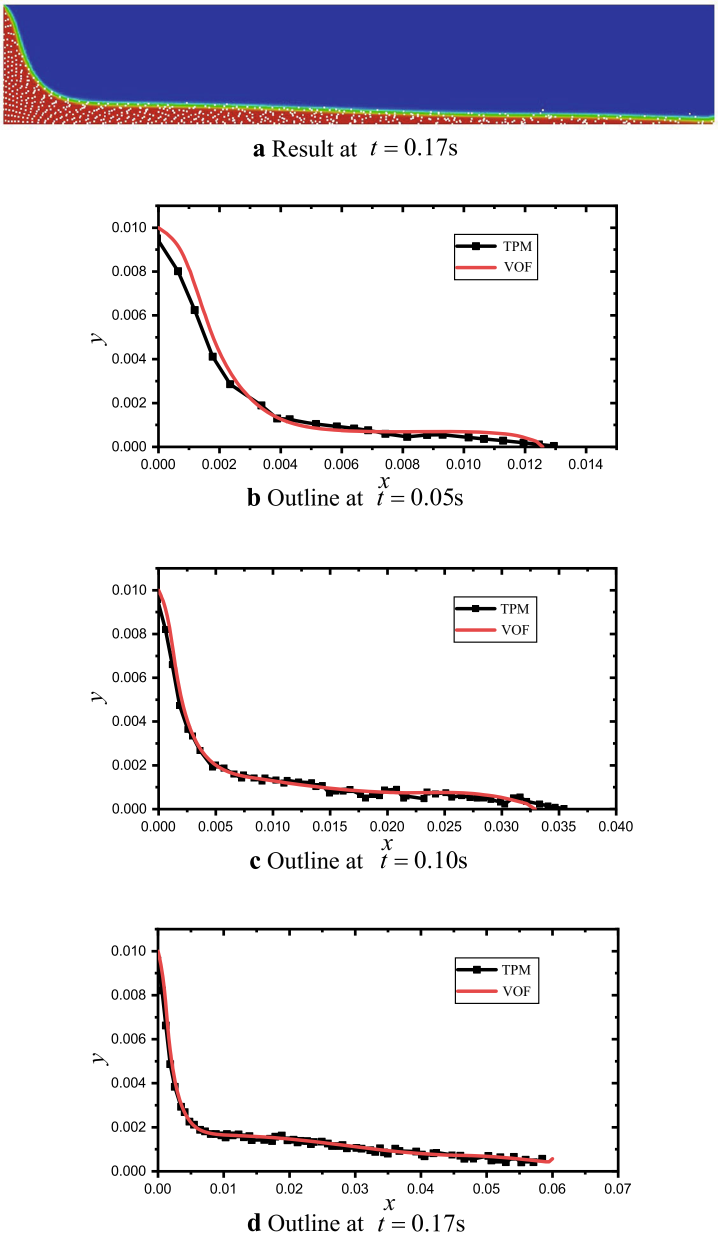
signifies that the particle is hardly stretched. The numerical fracture in flow problems is caused mainly because the particle resolution is too low to represent the flow field accurately. Without extreme particle stretch, the adaptive particle scheme cannot eliminate the numerical fracture.

4.4 Steady flow with gravity

An example is shown in this part to verify the inlet condition. As shown in Fig. 19, the computational domain is 2.0m in length and 0.2m in width. An inlet condition is imposed on the left end of the domain with an initial velocity v_0. The right end of the domain is a free boundary condition which allows the water flow out freely. At the beginning, the computational domain is empty. Water flows into the domain through the inlet condition and flows out through the right end. The flow field is developed and goes steady as shown in Fig. 19.

Figure 20 shows the pressure nephograms at different times with the inlet velocity $v_0 = 0.4$. It is apparent that a steady flow field is reached at time $t = 2.0$ and the height of fluid level becomes consistent at downstream location. The change in the total mass in the computational domain with time is shown in Fig. 21. Figure 21a is the change in mass in a long time, and Fig. 21b shows the mass before the water flows out of the domain through the right end. The mass of water

Fig. 22 Comparison between TPM and VOF



a Result at $t = 0.17$s

b Outline at $t = 0.05$s

c Outline at $t = 0.10$s

d Outline at $t = 0.17$s

before flowing out appears linear with time and matches well with the analytical result. It proves the conservation of mass and the correctness of the imposition of the inlet condition.

To examine the accuracy of TPM, we made a comparison between this method and VOF (volume of fluid) method with a PLIC geometrical reconstruction. The result of VOF is obtained by Fluent. The domain is 0.06m long and 0.01m width. An inlet condition is imposed with an incoming velocity 0.05m and the Reynolds number $Re = 500$. Thus, the laminar flow model is still appropriate. Figure 22 shows the result and the outline of water at different times. In Fig. 22a, the nephogram shows the volume fraction of water and the green part represents the outline of water. The white dots are transport points in the result of TPM which matches the outline very well. The outlines at three time steps are plotted in Fig. 22b–d, where the outlines of TPM results are delineated with the points with the maximum y coordinate. Thus, the outline is unsmooth. In spite of that, the outlines of TPM and VOF match well.

5 Conclusion

The transport point method is an improvement over the MPM. It employs MLS method to reconstruct quantities on nodes and cell centers and Gaussian quadrature to compute integrals in the weak form. Meanwhile, point quadrature has to be used on the boundary cells. Gaussian quadrature uses the cell volume as the integral weight instead of the particle volume; thus, it is unnecessary to carry volume on transport points. As a result, points can be deleted, moved and added easily without deteriorating the accuracy. Therefore, we can rearrange the transport points in a cell by reconstructing a linear field in it. By generating new points in some cells to maintain an inlet flow field, a smooth and steady inlet condition can be imposed.

In comparison with the MPM, the transport points in TPM take the local value of quantities in the flow field. They also indicate where the flow field exists and delineates the boundary. MLS reconstruction and Gaussian quadrature improve the rate of accuracy, making it reach the optimal rate in case the boundary is fixed as previously stated. A Riemann problem is also investigated, and the result matches well with the analytical result. The last example shows the applicability in inlet condition and the complex flow problem.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Zhang X, Chen Z, Lian YP, Liu Y (2017) Material point method. Elsevier, Amsterdam
2. Sulsky D, Chen Z, Schreyer HL (1994) A particle method for history-dependent materials. Comput Methods Appl Mech Eng 118(1–2):179–196
3. Zhang DZ, Ma X, Giguere PT (2011) Material point method enhanced by modified gradient of shape function. J Comput Phys 230(16):6379–6398
4. Zhang DZ, Zou Q, VanderHeyden WB, Ma X (2008) Material point method applied to multiphase flows. J Comput Phys 227(6):3159–3173
5. Huang P, Zhang X, Ma S, Huang X (2011) Contact algorithms for the material point method in impact and penetration simulation. Int J Numer Methods Eng 85(4):498–517
6. Zhang X, Huang P (2010) An object-oriented mpm framework for simulation of large deformation and contact of numerous grains. CMES Comput Model Eng Sci 55(1):61–88
7. Huang P, Zhang X, Ma S, Wang H (2008) Shared memory openmp parallelization of explicit mpm and its application to hypervelocity impact. CMES Comput Model Eng Sci 38(2):119–148
8. Lian Y, Zhang X, Liu Y (2011) Coupling of finite element method with material point method by local multi-mesh contact method. Comput Methods Appl Mech Eng 200(47–48):3482–3494
9. Bardenhagen SG, Nairn JA, Lu H (2011) Simulation of dynamic fracture with the material point method using a mixed j-integral and cohesive law approach. Int J Fract 170(1):49–66
10. Daphalapurkar NP, Lu H, Coker D, Komanduri R (2007) Simulation of dynamic crack growth using the generalized interpolation material point (gimp) method. Int J Fract 143(1):79–102
11. Chen Z, Hu W, Shen L, Xin X, Brannon R (2002) An evaluation of the mpm for simulating dynamic failure with damage diffusion. Eng Fract Mech 69(17):1873–1890
12. Chen Z, Feng R, Xin X, Shen L (2003) A computational model for impact failure with shear-induced dilatancy. Int J Numer Methods Eng 56(14):1979–1997
13. Sulsky D, Schreyer L (2004) Mpm simulation of dynamic material failure with a decohesion constitutive model. Eur J Mech A Solids 23(3):423–445
14. Sulsky D, Peterson K (2011) Toward a new elastic-decohesive model of arctic sea ice. Physica D Nonlinear Phenom 240(20):1674–1683
15. Liang Y, Benedek T, Zhang X, Liu Y (2017) Material point method with enriched shape function for crack problems. Comput Methods Appl Mech Eng 322:541–562
16. Lian Y, Zhang X, Zhou X, Ma Z (2011) A femp method and its application in modeling dynamic response of reinforced concrete subjected to impact loading. Comput Methods Appl Mech Eng 200(17–20):1659–1670
17. Lian Y, Zhang X, Liu Y (2012) An adaptive finite element material point method and its application in extreme deformation problems. Comput Methods Appl Mech Eng 241:275–285
18. Bardenhagen SG, Kober EM (2004) The generalized interpolation material point method. Comput Model Eng Sci 5(6):477–496
19. Sadeghirad A, Brannon RM, Burghardt J (2011) A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. Int J Numer Methods Eng 86(12):1435–1456
20. Jiang C, Schroeder C, Selle A, Teran J, Stomakhin A (2015) The affine particle-in-cell method. ACM Trans Gr 34(4):51
21. Steffen M, Kirby RM, Berzins M (2008) Analysis and reduction of quadrature errors in the material point method (mpm). Int J Numer Methods Eng 76(6):922–948
22. Sulsky D, Gong M (2016) Improving the material-point method. In: Weinberg K, Pandolfi A (eds) Innovative mumerical approaches for multi-field and multi-scale problems, lecture notes in applied and computational mechanics, vol 81. Springer International Publishing, Switzerland, pp 217–240
23. Ma S, Zhang X, Lian Y, Zhou X (2009) Simulation of high explosive explosion using adaptive material point method. Comput Model Eng Sci 39(2):101
24. Monteleone A, Monteforte M, Napoli E (2017) Inflow/outflow pressure boundary conditions for smoothed particle hydrodynamics simulations of incompressible flows. Comput Fluids 159:9–22
25. Zhao X, Bolognin M, Liang D, Rohe A, Vardon PJ (2019) Development of in/outflow boundary conditions for mpm simulation of uniform and non-uniform open channel flows. Comput Fluids 179:27–33
26. Kafaji I (2013) Formulation of a dynamic material point method (MPM) for geomechanical problems. PhD thesis, University of Stuttgart, Stuttgart, Germany
27. Beuth, L (2012) Formulation and application of a quasi-static material point method. PhD thesis, University of Stuttgart, Stuttgart, Germany
28. Chen ZP, Zhang X, Sze KY, Kan L, Qiu XM (2018) vp material point method for weakly compressible problems. Comput Fluids 176:170–181
29. Zhou Z, De Kat J, Buchner B (1999) A nonlinear 3d approach to simulate green water dynamics on deck. In: Proceedings of the seventh international conference on numerical ship hydrodynamics, Nantes, France, pp 1–15
30. Lobovskỳ L, Botia-Vera E, Castellana F, Mas-Soler J, Souto-Iglesias A (2014) Experimental investigation of dynamic pressure loads during dam break. J Fluids Struct 48:407–434