Computer Animation and Virtual Worlds

WILEY

SPECIAL ISSUE PAPER

An Adaptive Boundary Material Point Method With Surface Particle Reconstruction

Haokai Zeng[1] | Dongyu Yang[2] | Yanrui Xu[1] | Yalan Zhang[1,7] | Zhongmin Wang[2] | Feng Tian[3] | Xiaokun Wang[1,7] | Xiaojuan Ban[1,4,5,6]

[1]School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing, China | [2]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China | [3]Division of Natural and Applied Sciences, Duke Kunshan University, Jiangsu, China | [4]Beijing Key Laboratory of Big Data Innovation and Application for Skeletal Health Medical Care, University of Science and Technology Beijing, Beijing, China | [5]Key Laboratory of Advanced Materials and Devices for Post-Moore Chips, Ministry of Education, University of Science and Technology Beijing, Beijing, China | [6]Institute of Materials Intelligent Technology, Liaoning Academy of Materials, Shenyang, China | [7]Shunde Innovation School, University of Science and Technology Beijing, Guangdong, China

Correspondence: Xiaokun Wang (wangxiaokun@ustb.edu.cn) | Xiaojuan Ban (banxj@ustb.edu.cn)

Received: 24 April 2025 | **Accepted:** 6 May 2025

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 62332017 and 62376025), and Basic and Applied Basic Research Foundation of Guangdong Province (Grant No. 2023A1515030177).

Keywords: adaptive grids | computer animation | material point method | surface reconstruction

ABSTRACT

The expression of fine details such as fluid flowing through narrow pipes or split by thin plates poses a significant challenge in simulations involving complex boundary conditions. As a hybrid method, the material point method (MPM), which is widely used for simulating various materials, combines the advantages of Lagrangian particles and Eulerian grids. To achieve accurate simulations of fluid flow through narrow pipes, high-resolution uniform grid cells are necessary, but this often leads to inefficient simulation performance. In this article, we present an adaptive boundary material point method that facilitates adaptive subdivision within regions of interest and conducts collision detection across grids of varying sizes. Within this framework, particles interact through grids of differing resolutions. To tackle the challenge of unevenly distributed subdivided particles, we propose a surface reconstruction approach grounded in the color distance field (CDF), which accurately defines the relationship between the particles and the reconstructed surface. Furthermore, we incorporate a mesh refinement technique to enrich the detail of the mesh utilized to mark the grids during subdivision. We demonstrate the effectiveness of our approach in simulating various materials and boundary conditions, and contrast it with existing methods, underscoring its distinctive advantages.

1 | Introduction

The material point method (MPM) has been widely applied to the simulation of various materials since its introduction into computer graphics by Stomakhin et al. [1]. MPM utilizes a background grid for force calculation and particle-based simulation, making it particularly effective for modeling elastic and plastic materials undergoing large deformations [2]. While MPM provides the convenience of a unified framework, the efficiency and accuracy of simulations remain constrained by the grid size and particle count. As grid resolution increases, more particles are required within each grid cell to ensure a uniform and stable force distribution. However, this results in significant particle-grid transmission overhead, as seen in simulations of sand [3] and water [4]. To prevent numerical fractures, additional particles must be introduced into the grid.

https://doi.org/10.1002/cav.70024

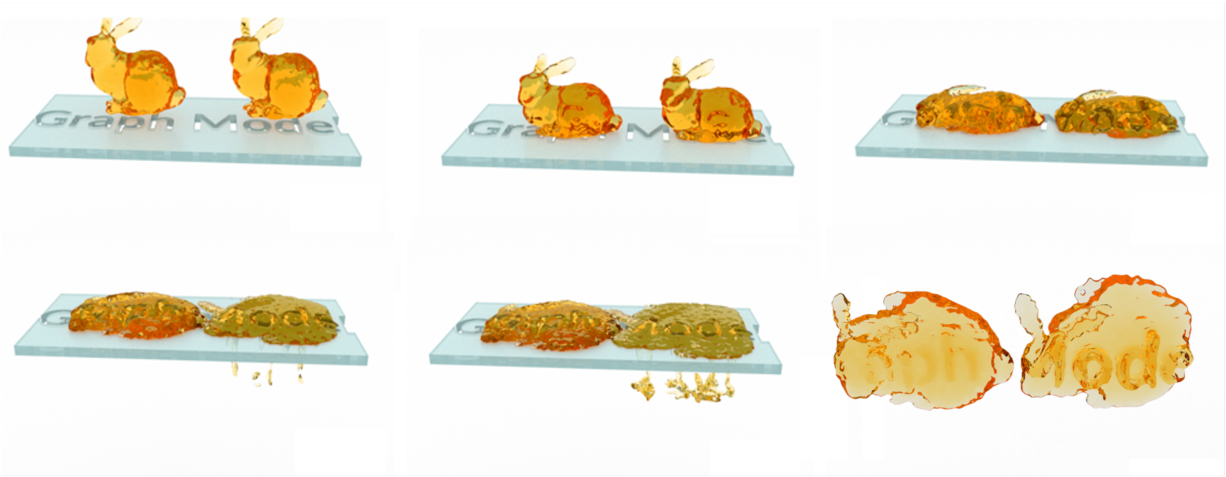

FIGURE 1 | Bunny penetrates thin alphabet panel. Comparing the results of the original grid and the adaptive grid, bunny with original grid (left in the scene) is difficult to penetrate through the pores. Our adaptive method (right in the scene) allows penetration, and produces a more refined and detailed effect.

These challenges hinder the improvement of simulation efficiency as grid accuracy increases. In addition, the grid size influences the phenomena of material self-collision and boundary coupling. When using a uniform grid for force calculation, particles separated by a distance smaller than grid size do not interact with each other. This can lead to unrealistic phenomena, such as excessively wide cuts, missed collisions and slow-moving streams in pipes.

To maintain efficiency while addressing issues arising from grid resolution, we propose an adaptive boundary material point method. By adaptively subdividing particles in regions of interest, we can simulate particles of varying sizes using grids at different scales. All particles interact through background grid transmission. Our main contributions can be summarized as follows:

1. A novel multigrid adaptive interpolation MPM that enables the interaction of multiscale particles across different grids. It enables the simulation of fine details (see Figure 1) and improves the efficiency of the simulation meanwhile consuming less computational resources.
2. A new surface reconstruction scheme based on the signed distance field, which mitigates the issue of uneven distribution of newly generated subdivision particles on the surface. It ensures a smooth and consistent particle distribution for high-quality surface reconstruction.
3. An extension of the MPM solver to support multiple complex boundary conditions with corresponding subdivision strategies. It improves the versatility and applicability of MPM solver to handle a wider range of simulation scenes.

2 | Related Work

2.1 | Material Point Method

The MPM, was introduced into computer graphics and applied to snow simulation by Stomakhin et al. [1]. Since then, the MPM has continued to attract interest and has undergone extensive development. Tampubolon et al. [5] used the continuous mixing theory to deal with fluid–sand interactions to realize realistic physical phenomena such as mudslides. The affine particle-in-cell (APIC) method proposed by Jiang et al. [6] reduces dissipation and preserves the lost angular momentum from the particle-grid transition compared to particle-in-cell (PIC) methods [7] and fluid implicit particle (FLIP) methods [8] and has become the dominant interpolation scheme for the MPM. On this basis, Klár et al. [9] introduced the Drucker-Prager model to describe the elastic and plastic behavior of materials. Hu et al. [10] proposed the compatible particle-in-cell (CPIC) method and the moving-least-squares MPM (MLS-MPM) for dealing with two-way rigid body coupling, significantly improving the efficiency of particle-grid information transfer. Fang et al. [11] implemented a return mapping algorithm in the prediction correction stage to simulate viscoelastic materials. The ghost matrix operator-splitting scheme proposed by Fang et al. [12] effectively solves the viscous numerical artifacts at the boundary of the MPM.

2.2 | Adaptive Method

Adaptive methods dynamically adjust the resolution to fit the accuracy, mainly including temporal and spatial adaptation, where the spatial method determines the underlying discretization scheme. An octree-based adaptive method is a classic example of a spatial adaptive technique. Seiler et al. [13] proposed an octree-based deformable object cutting scheme that achieves high-detail resolution surfaces through refinement and node splitting. Setaluri et al. [14] utilized a sparse grid to store the octree, enhancing the efficiency of data access. Xiao et al. [15]

introduced a tilted mesh based on the octree to avoid T-junctions, thereby improving the effectiveness of adaptive simulations.

Regarding particles, Adams et al. [16] were the pioneers in introducing a size function based on geometric local feature dimensions, enabling adaptive particle resolution. To mitigate instability arising from excessive differences in neighboring particle sizes, Winchenbach et al. [17, 18] implemented continuous transitions in particle resolution by adjusting splitting and merging modes. Building upon this, Zhai et al. [19] enhanced particle distribution stability through improved interpolation functions. Liu et al. [20] considered factors such as vorticity and splashing, optimizing adaptive simulations for high-dynamic scenarios. To further reduce computational costs, Chentanez et al. [21] proposed a coupled pure Lagrangian and Eulerian approach for simulating single-fluid bulk materials. Nakanishi et al. [22] focused on constructing a suitable data structure to realize the octree grids adaptation of FLIP, adjusting the background grid size only near the fluid surface.

In terms of temporal adaptation, Sun et al. [23] solved the problem of the global time step that affects the stability of MPM. Fang et al. [24] used a chunked time-step approach in the matter-point method to achieve time acceleration for asynchronous matter-point methods, using asynchronous computation locally and synchronization globally. Spatially, Lian et al. [25] proposed a grid-based hierarchical MPM by modifying the shape function. Gao et al. [26] further modified the shape function to ensure its non-negativity of the shape function, which ensures the stability of the simulation.

Compared to most previous work, we implement a multigrid adaptive scheme that avoids direct modification of the mesh geometry, thereby ensuring interpolation stability and non-negativity while maintaining compatibility with existing computational frameworks.

2.3 | Surface Reconstruction

Meshless surface reconstruction is widely used in SPH-based fluid simulation. Blinn [27] expressed the positional relationship between the sampling points and object surface through implicit surface equations. Zhu et al. [3] obtained a smoother quicksand surface by smoothing the color distance field (CDF). Adams et al. [16] proposed time continuity to improve the accuracy of distance field calculation. Wang et al. [28] used an anisotropic kernel method to accurately reconstruct the fine effects of multiphase interfaces.

Compared with meshless methods, mesh schemes are more suitable for dealing with regular data distributions. Yu et al. [29] avoided reconstructing the surface in every frame by constructing object surface at the beginning moment and tracking the segmentation and integration of the surface at each time step, which ensured efficiency. Akinci et al. [30] efficiently reconstructed a high-quality mesh by constructing scalar fields in a narrow band around the surface in parallel. Xing et al. [31] constructed a localized mesh by detecting fluid surface particles through the shading method to achieve the surface tension constraints.

In this article, our surface reconstruction framework demonstrates seamless integration with the grid-based topology of the MPM, using color-distance-field encoding to quantitatively determine particle–surface positional relationships. This hybrid approach significantly improves the accuracy of reconstruction while maintaining computational stability across complex material boundaries.

3 | Adaptive MPM

3.1 | MPM Basics

The material point method uses the APIC method for particle-grid transfer. We use subscripts i to denote grid nodes, p to denote particles, and superscripts n to denote discrete time steps. Particles carry information such as mass m, position $\boldsymbol{x}$, velocity $\boldsymbol{v}$, deformation gradient $\mathbf{F}$, and affine matrix $\mathbf{C}$. At the beginning of each time step, the particle information is first passed to the grid.

$$m_i^n = \sum_p w_{ip}^n m_p \tag{1}$$

$$m_i^n \mathbf{v}_i^n = \sum_p w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{C}_p^n(\mathbf{x}_i - \mathbf{x}_p^n))s \tag{2}$$

where w represents the interpolation weight. After collecting the particle information, the velocity of the grid $\mathbf{v_i^n}$ is calculated and the force and boundary treatment are performed on the grid.

$$\mathbf{v}_i^n = m_i^n \mathbf{v}_i^n / m_i^n \tag{3}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \left(m_i^n \mathbf{g} - \sum_p V_p \frac{\partial \psi}{\partial \mathbf{F}} (\mathbf{F}_p^n)^T \nabla w_{ip}^n \right) \Delta t / m_i^n \tag{4}$$

where ψ is the energy density function. After the grid is updated, the velocity and affine matrix for the next time step will be interpolated back to the particles.

$$\mathbf{v}_p^{n+1} = \sum_i w_{ip}^n \mathbf{v}_i^{n+1} \tag{5}$$

$$\mathbf{C}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} \left(\frac{\partial w_{ip}}{\partial x} \right)^T \tag{6}$$

3.2 | Multigrid Adaptive Method

We use multiple grids to solve the motions of particles inside the object and near the boundary, while computing their interactions in the overlapping region of multisized particles.

The traditional material point method incorporates nearby particle interactions by gathering information from neighboring particles at a common grid node, as shown in Figure 2a.

In the original MPM, particles are only interpolated to a single grid. To ensure that particles of different sizes can still influence each other through the nodes, we achieve interaction by interpolating particles in overlapping regions onto the opposite grid, as shown in Figure 1b. At the boundary of the collision, the original grid with a length of dx is subdivided into a finer grid with a length of $dx' = dx/b(b > 1)$. Where b is the subdivision base, by adjusting the value of b, our approach can be adapted to different scenarios and enable more detailed simulations.

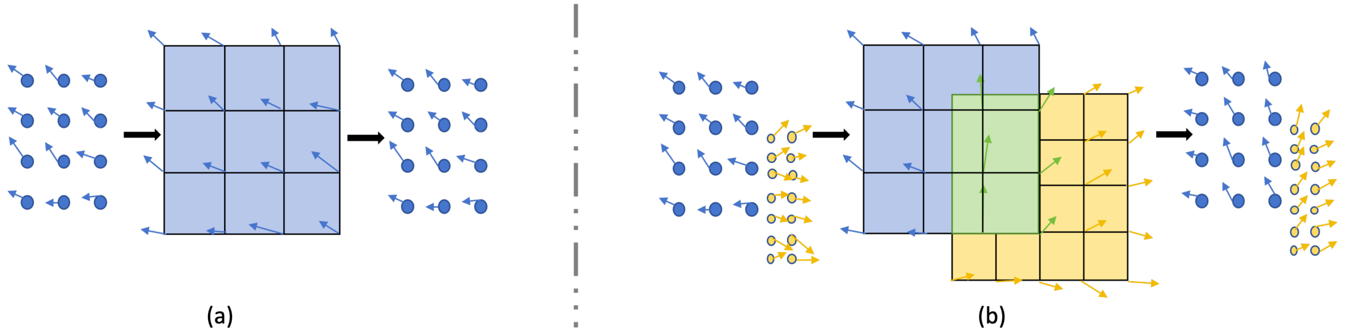

FIGURE 2 | Interpolation scheme. (a) The particles are only affected by a single grid under the traditional way. (b) The adaptive interpolation of the original particles (blue) and subdivided particles (yellow) to the grids, and the green area represents where different grids affect each other.

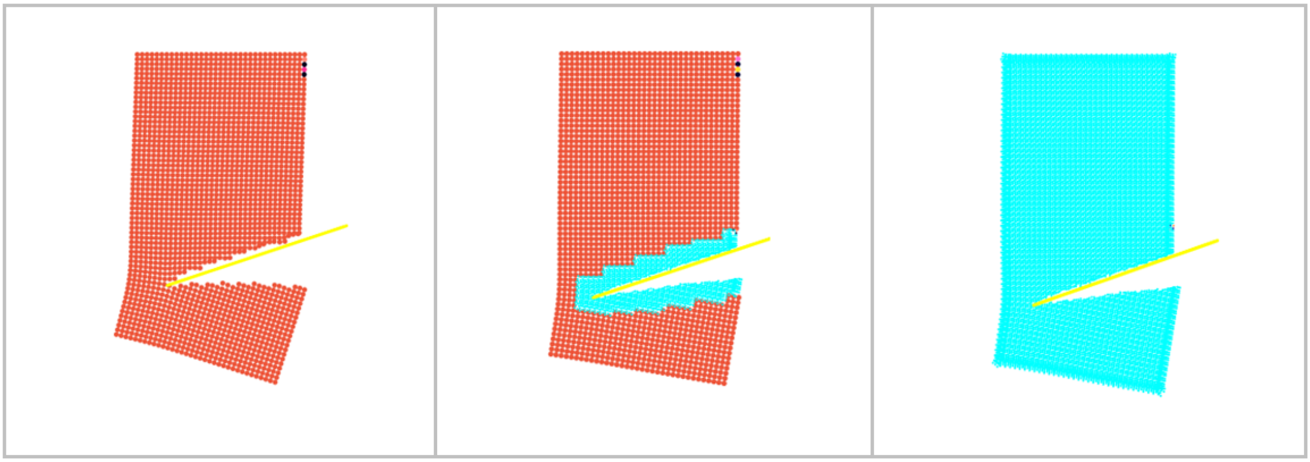

FIGURE 3 | Cut elastic object. Left: original grid, which produces rough boundaries. Right: fully subdivided grid, which increases the computational overhead. Center: adaptive grid, which generates subdivided particles only at the boundary, while balancing accuracy and efficiency.

The motion of the original and subdivided particles is simulated independently using two sets of grids, and a bidirectional interpolation scheme is used for the particles in the overlapping regions of the grids. In this article, we use i to refer to the original grid node and i' to refer to the subdivided grid node, subscripts p and p' denote the original particle and the subdivided particle, while r represents the rigid body. We use the MLS-MPM scheme [10] to update the stress and compute the forces.

$$\mathbf{F}_p = (\mathbf{I} + \Delta t \mathbf{C}_p)\mathbf{F}_p \tag{7}$$

For the original grids, the particle-grid transfer process is as follows:

$$m_i = \sum_p w_{ip} m_p + \sum_{p'} w_{ip'} m_{p'} \tag{8}$$

$$\mathbf{f}_{ip} = -\frac{\partial \Psi}{\partial \mathbf{x}_i} = -\frac{4}{\Delta x^2} V_p^0 \frac{\partial \Psi}{\partial \mathbf{F}} \mathbf{F}_p^T w_{ip}(\mathbf{x}_i - \mathbf{x}_p) \tag{9}$$

$$\begin{aligned} m_i \mathbf{v}_i = {} & \sum_p \mathbf{f}_{ip} \Delta t + w_{ip} m_p (\mathbf{v}_p + \mathbf{C}_p(\mathbf{x}_i - \mathbf{x}_p)) \\ & + \sum_{p'} \mathbf{f}_{ip'} \Delta t + w_{ip'} m_{p'} (\mathbf{v}_{p'} + \mathbf{C}_{p'}(\mathbf{x}_i - \mathbf{x}_{p'})) \end{aligned} \tag{10}$$

For subdivided meshes, the transfer process remains analogous to that of the original mesh. During this process, all the occurrences of the index i pertaining to the original mesh points in Equations (8–10) should be replaced with i' for the subdivided mesh points. Additionally, the corresponding grid size dx should be replaced with dx' for the subdivided mesh.

After the grid force is solved, the grid transfers the velocity only to its own particles. The grid–particle transfer form is similar to that in the APIC. The grid–particle transfer equation is as follows:

$$\mathbf{v}_p = \sum_i w_{ip} \mathbf{v}_i \tag{11}$$

$$\mathbf{v}_{p'} = \sum_{i'} w_{i'p'} \mathbf{v}_{i'} \tag{12}$$

4 | Split and Reconstruction

4.1 | Elastoplasticity

For elastic materials (see Figure 3), the deformation $\mathbf{F}_E$ is composed of both rotational and nonrotational deformations [2]. The elastic potential energy function X arises to penalize the nonrotational part. The first Piola-Kirchhoff stress is determined by $\mathbf{P} = \partial \Psi / \partial \mathbf{F}_E$. The Cauchy stress in the elastic material is $\boldsymbol{\sigma} = \mathbf{P}\mathbf{F}_E^T / det(\mathbf{F})$, where Ψ is the energy function of the elastic potential. We use the fixed-corotated model to deal with the elastic response, Ψ is defined as $\Psi = \mu \sum_k (\boldsymbol{\sigma}_k - 1)^2 + \frac{\lambda}{2}(J - 1)^2$, and

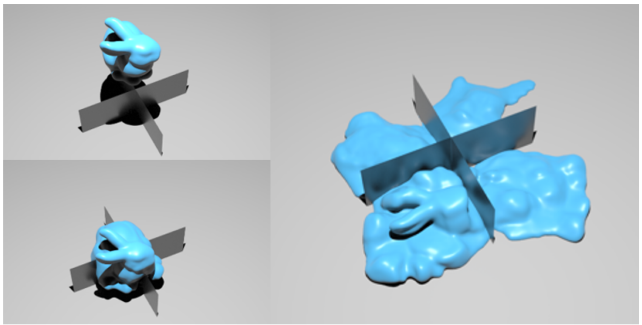

FIGURE 4 | Bunny split. A viscoelastic bunny separated by a thin plate.

$\mathbf{P} = 2\mu(\mathbf{F}_E - \mathbf{R} + \lambda(J-1)J\mathbf{F}_E^{-T})$, where $\boldsymbol{\sigma}_\mathbf{k}$ is singular values of $\mathbf{F}_E$, μ and λ are Lamé parameters, and $J = det(\mathbf{F})$.

Plasticity is achieved by projecting the gradient of elastic deformation onto its yield surface [9], defining the yield function of the elastic deformation as **Z**. The plastic deformation is updated as:

$$\mathbf{F}_E = \mathbf{Z}(\tilde{\mathbf{F}_E}, c_f) \tag{13}$$

$$\mathbf{F}_P = \mathbf{F}_E^{-1}\tilde{\mathbf{F}_E}\tilde{\mathbf{F}_P} \tag{14}$$

where **Z** projects the deformation gradient $\mathbf{F}_E$ onto the yield surface defined by the parameter c_f. The Drucker-Prager plasticity model is based on Coulomb frictional interactions and sets the shear stresses to be no greater than the coefficient of friction multiplied by the normal stress. If the plastic deformation $\Delta\sigma = c_f tr(\epsilon) + ||\epsilon - \frac{tr(\epsilon)}{d}\mathbf{I}||_\mathbf{F}$ is less than 0, it is returned directly; otherwise, it is projected onto the yield surface.

4.2 | Split and Merge

When the original particle enters the subdivision grid, it will be split to generate subdivision particles. To split the particles uniformly into the surrounding subdivision meshes, we split the particles into N_s ones, which avoids potential numerical breaks. To ensure the conservation of mass and momentum, the subdivided particles need to satisfy the following constraints:

$$m_p = \sum_{p'} m_{p'}, m_p\mathbf{v}_p = \sum_{p'} m_{p'}\mathbf{v}_{p'} \tag{15}$$

After collision, our method maintains the stability of the adapted particles, while the merging strategy of Gao et al. [26] remains optional. The scene of the viscoelastic rabbit being cut in Figure 4 demonstrates the effectiveness of our method in handling complex collisions.

4.3 | Surface Reconstruction

When elastoplastic particles are split into subdivided grids, the relationship between the subdivided particles and the original surface is not considered. This can lead to a series of unrealistic issues, such as protrusions and swellings caused by particle aggregation. To address this, we identify surface particles by measuring the particle mass gradient within the grid (see Figure 5).

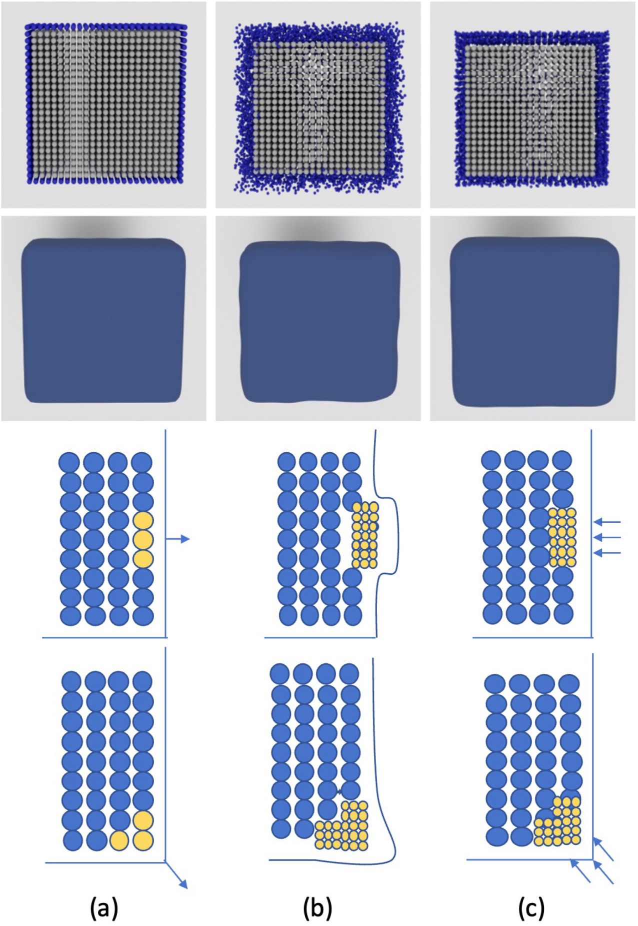

FIGURE 5 | Surface reconstruction. (a) The original state. (b) The subdivided surface without reconstruction, which results in protrusions and swelling. (c) These problems can be effectively avoided by our surface reconstruction method.

$$G_i = \sum_p \nabla w_{ip} m_p + \sum_{p'} \nabla w_{ip'} m_{p'} \tag{16}$$

The distance from the grid nodes to the particle surface is computed using the color distance field (CDF) construction method introduced in Section 5.3. The distance d_s and the direction vector $\mathbf{N}_s$ from the subdivision grid nodes i' to the particle surface are given by the following:

$$d_s = \underset{\mathbf{y}\in\partial\Omega_p}{\arg\min} ||\mathbf{x}_{i'} - \mathbf{y}||_2, \partial\Omega_p = \{p|\mathcal{P}(x_p) = 1\} \tag{17}$$

$$\mathcal{P}(x) = \begin{cases} 1, \text{if} x \in \Omega_i \\ 0, \text{otherwise} \end{cases} \tag{18}$$

$$\mathbf{N}_s = -\nabla d_s \tag{19}$$

where Ω_i denotes the axis-aligned box centered on x_i and $G_i > t_p$ (t_p is a user-specified threshold constant).

The subdivided particle reconstruction positions are constructed as follows:

$$\mathbf{x}_{p'} = \mathbf{x}_{p'} + \sum_{i'} w_{i'p'}\mathbf{N}_s d_s \tag{20}$$

5 | Boundary Processing

5.1 | Grid Force Responses

We use the CPIC [10] method to evaluate the rigid body response. The color distance field (CDF) is used to establish the relationship between particles and grids with respect to the rigid body. Here, T_{ir} and T_{pr} indicate whether the mesh and particles lie in the positive direction of the normal vector of the rigid body, while d_{ir} and d_{pr} represent the minimum distances from the rigid body. The term T_{pr} is defined as:

$$T_{pr} = \operatorname{sign}\left(\sum_i w_{ip} T_{ir} d_i\right) \tag{21}$$

Particles and grids are considered compatible if T_{ir} and T_{pr} are equal. In such cases, the particles are labeled as p^{i+} and i^{p+}. For incompatible particles, the grid transform is disabled and the boundary response velocity $\overline{\mathbf{v}}_p$ is computed, where c is a non-zero constant.

$$\mathbf{v}_{ir} = \mathbf{v}_r + \mathbf{w}_r \times (\mathbf{x}_i - \mathbf{x}_r) \tag{22}$$

$$\overline{\mathbf{v}}_p \ = \mathbf{v}_{ir} + (\mathbf{v}_p - \mathbf{v}_{ir}) - ((\mathbf{v}_p - \mathbf{v}_{ir}) \cdot \mathbf{n}_p)\mathbf{n}_p + c\mathbf{n}_p \tag{23}$$

For original particles, the grid-to-particle transfer under the corresponding adaptive scheme is

$$\mathbf{v}_p = \sum_{j \in i^{p-}} \mathbf{w}_{jp} \overline{\mathbf{v}}_p + \sum_{j \in i^{p+}} \mathbf{w}_{jp} \mathbf{v}_j \tag{24}$$

$$\mathbf{C}_p = \mathbf{D}_p^{-1} \left(\sum_{j \in i^{p-}} \mathbf{w}_{jp} \overline{\mathbf{v}}_p (\mathbf{x}_j - \mathbf{x}_p)^T + \sum_{j \in i^{p+}} \mathbf{w}_{jp} \mathbf{v}_j (\mathbf{x}_j - \mathbf{x}_p)^T \right) \tag{25}$$

For subdivided particles, the transfer process is similar, and the index p in Equations (24, 25) should be replaced with p'.

5.2 | Dynamic Boundary Detection

Our adaptive method is applicable to handle collisions under complex boundaries, thin rigid bodies, and different models, such as the fluid flow in the hourglass shown in Figure 6. We perform two types of boundary detection. The first is the subdivided boundary B_s detection. When the particle enters B_s, the particle will be subdivided and its surface will be optionally reconstructed. The second is the constraints boundary B_c, where we apply the boundary constraint to the particles.

To detect the subdivided boundary B_s, such as bottlenecks, we employ the Catmull et al. [32] subdivision algorithm to enhance the details near the boundary.

$$\mathbf{x}_r = \frac{F + 2E + (n_r - 3)\mathbf{x}_r}{n_r} \tag{26}$$

where F is the average of the face points of all surrounding faces, E is the average of all neighboring edge points, and n_r is the number of connected edges of the vertex. We utilize the Gaussian kernel function $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$ to compute the spatial distribution density λ_i of rigid-body nodes around grid cells, where the bandwidth parameter h governs the smoothing intensity of the kernel through the density estimator,

$$\lambda_i = \frac{1}{n_r h^3} \sum_r K\left(\frac{||\mathbf{x}_i - \mathbf{x}_r||}{h}\right) \tag{27}$$

Detecting the value of λ_i, we mark regions with a dense distribution of points as the subdivided boundary B_s.

5.3 | CDF Build

To get the correct rigid body response, the distance from each grid node x_i to the rigid body needs to be calculated to construct the signed distance field. To avoid numerical errors, we use the triangle center of gravity interpolation formula to calculate the projection point $\overline{x_i}$ of the grid node x_i toward the triangle (mesh)

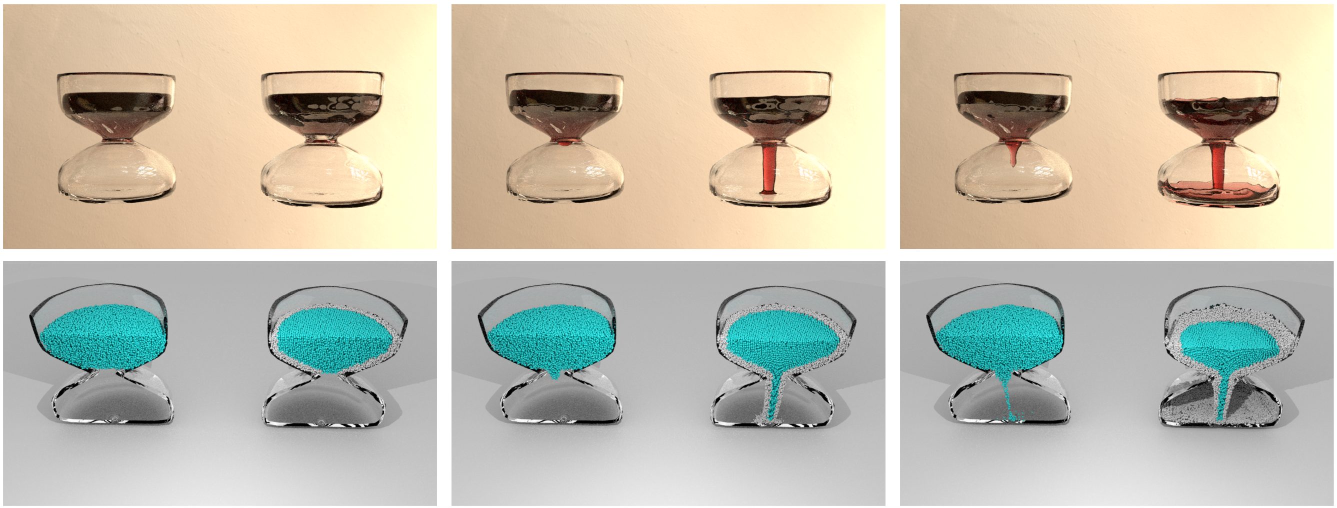

FIGURE 6 | Fluid flow in an hourglass. We use blue color for original particles and white color for subdivided ones. Comparing with the results of the original grid (left), our adaptive method (right) allows the fluid to pass through fine apertures smoother.

and the shortest directed distance d_i. Assume that the vertices of the triangle are A, B, C, and $\overline{x_i}$ satisfy the following constraints:

$$\overline{\mathbf{x}_i} = \alpha A + \beta B + \gamma C \tag{28}$$

$$\begin{aligned} \left(\overline{\mathbf{x_i}} - \mathbf{x_i}\right) \cdot AB &= 0 \\ \left(\overline{\mathbf{x_i}} - \mathbf{x_i}\right) \cdot BC &= 0 \end{aligned} \tag{29}$$

Equations (28) and (29) give:

$$\gamma = -\frac{\left(\left(A - x_i\right) \cdot AB\right)(AB \cdot AC)}{D} + \frac{\left(\left(A - x_i\right) \cdot BC\right)(AB \cdot AB)}{D} \tag{30}$$

$$\beta = -\frac{\left(A - x_i\right) \cdot (AB) - \gamma(AC \cdot AB)}{(AB \cdot AB)} \tag{31}$$

where $D = (AB \cdot AC)(AB \cdot BC) - (AC \cdot BC)(AB \cdot AB)$, and the value of the gravity interpolation coefficients α, β, and γ determines whether the projected point $\overline{x_i}$ falls inside the triangle. If the projected point falls outside, calculate the closest point of x_i with respect to the three sides. In summary, we get $d_i = |x_i - \overline{x_i}|$, and the normal grid $\mathbf{n}_i = \nabla d_i / |\nabla d_i|$. We use the robust MLS reconstruction technique to rebuild d_p and $\mathbf{n}_p = \nabla d_p / |\nabla d_p|$.

6 | Algorithmic Process

The adaptive method we introduced can be easily integrated with the classical material point method framework. We use APIC transfers to ensure stability and the MLS-MPM scheme to improve particle-grid transfer efficiency. The workflow is shown in Figure 7 and the main steps are as follows (Algorithm 1):

1. Loop preparation: at the beginning of each step, the signed distance field of the grid is constructed based on the rigid body and particle positions. The rigid body and the original grid around the model are detected and marked as subdivided.

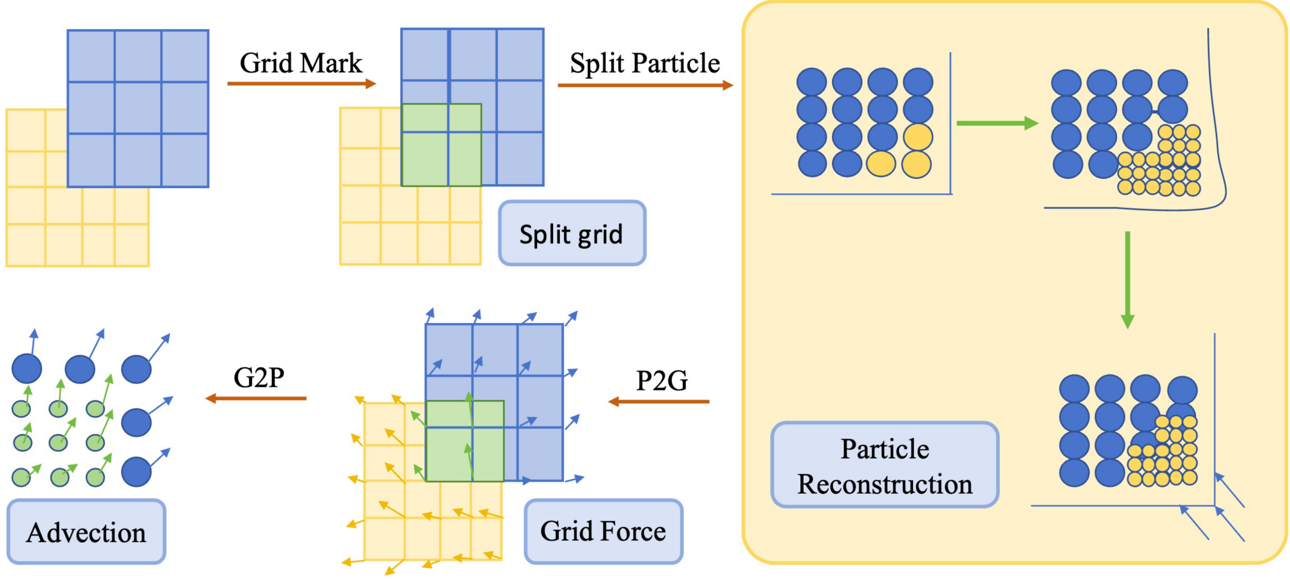

FIGURE 7 | The workflow of our adaptive boundary MPM simulation method includes three steps : boundary mesh refinement, smooth particle surface reconstruction, and grid-particle hybrid calculation.

ALGORITHM 1 | MPM Transmission Scheme.

```
procedure INITIALIZATION
    Mark the Subdivision Grid. Update $\lambda_i$ using Equations (26-27).
    Update grid SDF for reconstruction. Update $d_i, N_i$ by Equations (28-29).
end procedure
procedure LOOP STEP
    Particle Split. $m_{p'}, v_{p'}$ should obeys conservation of mass and momentum.
    Reconstruct. Reconstruct surface position $x_{p'}$ by Equations (17-20).
    Update deformation gradient $F_p, F_{p'}$ by Equation (7).
    P2G. Using grid nodes to introduce interaction of different particles. Update grid state
  $m_i, m_{i'}, v_i, v_{i'}$ by Equations (8-10).
    Grid Force. Update external force gravity and boundary force by Equations (22-23).
    G2P. Update particle state $v_p$, $v_{p'}$, $C_p$, $C_{p'}$ by Equations (24-25).
    Advect. $x_p = x_p + v_p \Delta t$, $x_{p'} = x_{p'} + v_{p'} \Delta t$.
end procedure
```

2. Split and reconstruction: the mass $m_{p'}$ and velocity $v_{p'}$ of the split particle follow Equation (15), and the position $x_{p'}$ of the split particle is reconstructed according to Equation (20).
3. Particles-to-grid: calculate the mass m_i, $m_{i'}$ and velocity v_i, $v_{i'}$ of the grid nodes using the P2G transfer.
4. Grid force: explicit forces f_i are computed using MLS-MPM. Different boundary responses are imposed through the grid-rigid distance d_i.
5. Grid-to-particles: consistent with the G2P process of the APIC scheme, Equation (24) are used to update the particle velocities v_p, $v_{p'}$.
6. Particle advection.

7 | Experiments and Results

To validate the performance of our method, we set several fluid and elastic simulation scenarios and perform the simulation of the same scene multiple times using low resolution, high resolution, and adaptive resolution, respectively, to compare the performance and capabilities of our method. The basis of our comparison is the APIC and MLS-MPM methods. Table 1 presents the experimental parameters, computation time, and comparisons between different approaches. These experiments are conducted on a workstation with an 11th Gen Intel(R) Core(TM) i5-11260H, and a Nvidia GTX 1650 GPU.

7.1 | Fine Detail Boundaries in Simulation

Surface reconstruction. Our method effectively restores the subdivided object surface, reducing surface irregularities. In Figure 5, our method is able to achieve multiscale surface reconstruction. After particle subdivision, the distribution becomes uneven, causing surface roughness and potential volume expansion or shrinkage. After reconstruction, the particle distribution is more uniform, resulting in a smoother and more even surface, thereby enhancing the overall visual quality.

Flowing through narrow pipes. Figure 1 shows a bunny made of viscous fluid penetrating through a thin alphabet panel with holes. When comparing the results of the original grid and the adaptive grid, our adaptive method is able to allow penetration, which is difficult for the fluid to penetrate under the original grid conditions. The scene in Figure 6 presents the wine flowing through the glass hourglass. It is notable that our adaptive method can pass through the narrow neck of the hourglass easily and smoothly, while the original method is more difficult to pass.

Elastic cutting. Figure 3 showcases an experiment of a thin blade cutting through an elastic object. It can be observed that, under the original grid, the elastic object exhibits a noticeable jaggedness after being cut by the blade, failing to simulate a smooth cutting surface. However, our method achieves effective cutting. Moreover demonstrates significantly higher efficiency compared to fully subdivided grid. Additionally, the scene of a viscoelastic bunny falling on a crossed thin blade and being cut in Figure 4 also shows the effectiveness of our method.

7.2 | Performance Evaluation

Efficiency. According to Table 1, our method is significantly more efficient than high-resolution experiments. As shown in Figure 8, we compare the adaptive method with the material point method (MPM) under different grid resolutions. The MPM with the original coarse grid achieves the highest computational efficiency, but suffers from collision gaps near the boundaries. In contrast, using a refined grid significantly increases the number of particles, leading to a substantial rise in computational cost. Our adaptive method leverages the advantages of both grid resolutions, improving computational efficiency compared to the fully refined approach while achieving comparable visual results.

Comparison of kinetic energy evolution. In Figure 9, we compare the adaptive method with MLS-MPM under different grid

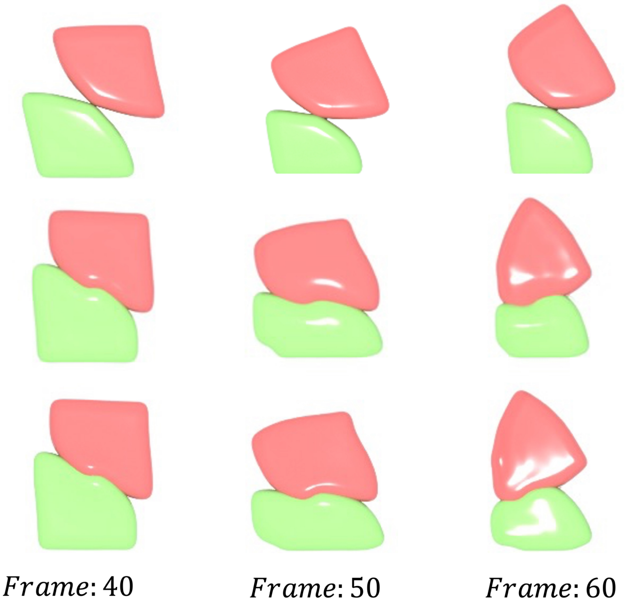

FIGURE 8 | Jelly collision. Compare the simulation results under different mesh sizes. Each row from top to bottom is the effect of low resolution, adaptive resolution and high resolution respectively.

TABLE 1 | Simulation performance and parameters. In this article, various experimental scenarios are simulated on different scales of grids, and the advantages of the adaptive approach over the original grid and the finest grid are demonstrated.

	x	x'	Particle (original)	Particle (adaptive)	Particle (finest)	Step/s (original)	Step/s (adaptive)	Step/s (finest)
Panel (Figure 1)	1/100	1/200	28k	71k	141k	3.9	2.7	2.4
Hourglass (Figure 6)	1/64	1/128	60k	167k	304k	5.9	3.0	1.7
Cut (Figure 3)	1/30	1/90	9.2k	27k	250k	14.5	12.3	7.8
Bunny split (Figure 4)	1/64	1/128	10k	140k	280k	15.1	8.2	5.6
Jelly collision (Figure 8)	1/30	1/90	21k	100k	560k	12.5	6.5	6.0

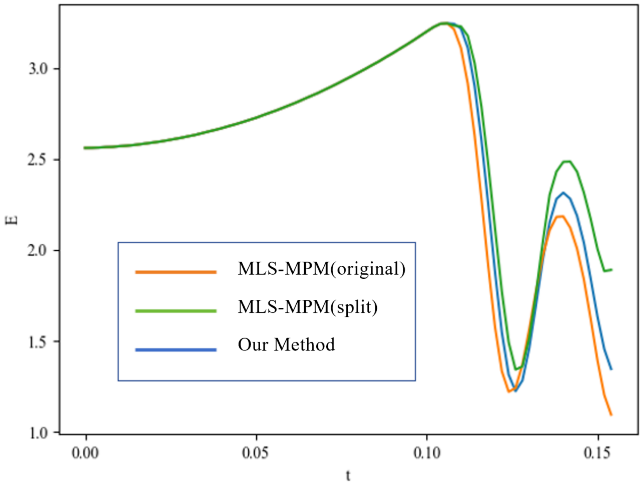

FIGURE 9 | Kinetic energy comparison.

resolutions. The peak kinetic energy of the adaptive method lies between those of the high-resolution and low-resolution grids. This indicates that the elastic potential energy can be converted into kinetic energy. In the APIC framework, our method effectively reduces energy loss during the particle-grid transfers process compared to the original grid, helping to preserve the elastic potential energy of the material, and extending the duration of elastic contact.

8 | Conclusion

We propose a multigrid boundary adaptive MPM to improve the accuracy of boundary processing. This method uses a subdivided grid to solve the contact of boundary particles and adopts the original grid to simulate the internal motion of the original particles, enabling the interaction of multiscale particles through communication between the two grids. While ensuring efficiency, fine boundary collision effects are achieved. Newly generated subdivided particles are reconstructed based on the normal direction of the surface mesh, achieving a smooth surface effect. Finally, the method is coupled with the mainstream material point method framework, and its effectiveness is demonstrated through several sets of experiments.

Although this article achieves a simulation that balances efficiency and realism, there are still some limitations. For example, it cannot be dynamically adjusted during the simulation. Additionally, the stability of the experiment is limited by the distribution of the subdivided particles.

The method presented in this article is applicable to various collision and coupling scenarios within the MPM framework. Furthermore, applying the adaptive method to simulate the injection molding of particulate materials coupled with boundaries presents a promising direction for future research.

Acknowledgments

This research was funded by National Natural Science Foundation of China (Nos. 62376025,62332017), Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515030177).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, "A Material Point Method for Snow Simulation," *ACM Transactions on Graphics (TOG)* 32, no. 4 (2013): 1–10.

2. C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle, "The Material Point Method for Simulating Continuum Materials," in *ACM SIGGRAPH 2016 Courses* (Association for Computing Machinery, 2016), 1–52.

3. Y. Zhu and R. Bridson, "Animating Sand as a Fluid," *ACM Transactions on Graphics (TOG)* 24, no. 3 (2005): 965–972.

4. X. Yan, C.-F. Li, X.-S. Chen, and S.-M. Hu, "Mpm Simulation of Interacting Fluids and Solids," in *Computer Graphics Forum*, vol. 37 (Wiley Online Library, 2018), 183–193.

5. A. P. Tampubolon, T. Gast, G. Klár, et al., "Multi-Species Simulation of Porous Sand and Water Mixtures," *ACM Transactions on Graphics (TOG)* 36, no. 4 (2017): 1–11.

6. C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The Affine Particle-In-Cell Method," *ACM Transactions on Graphics (TOG)* 34, no. 4 (2015): 1–10.

7. D. Sulsky, S.-J. Zhou, and H. L. Schreyer, "Application of a Particle-In-Cell Method to Solid Mechanics," *Computer Physics Communications* 87, no. 1–2 (1995): 236–252.

8. J. U. Brackbill and H. M. Ruppel, "Flip: A Method for Adaptively Zoned, Particle-In-Cell Calculations of Fluid Flows in Two Dimensions," *Journal of Computational Physics* 65, no. 2 (1986): 314–343.

9. G. Klár, T. Gast, A. Pradhana, et al., "Drucker-Prager Elastoplasticity for Sand Animation," *ACM Transactions on Graphics (TOG)* 35, no. 4 (2016): 1–12.

10. Y. Hu, Y. Fang, Z. Ge, et al., "A Moving Least Squares Material Point Method With Displacement Discontinuity and Two-Way Rigid Body Coupling," *ACM Transactions on Graphics (TOG)* 37, no. 4 (2018): 1–14.

11. Y. Fang, M. Li, M. Gao, and C. Jiang, "Silly Rubber: An Implicit Material Point Method for Simulating Non-Equilibrated Viscoelastic and Elastoplastic Solids," *ACM Transactions on Graphics (TOG)* 38, no. 4 (2019): 1–13.

12. Y. Fang, Z. Qu, M. Li, et al., "Iq-Mpm: An Interface Quadrature Material Point Method for Non-Sticky Strongly Two-Way Coupled Nonlinear Solids and Fluids," *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 51.

13. M. Seiler, D. Steinemann, J. Spillmann, and M. Harders, "Robust Interactive Cutting Based on an Adaptive Octree Simulation Mesh," *Visual Computer* 27 (2011): 519–529.

14. R. Setaluri, M. Aanjaneya, S. Bauer, and E. Sifakis, "Spgrid: A Sparse Paged Grid Structure Applied to Adaptive Smoke Simulation," *ACM Transactions on Graphics (TOG)* 33, no. 6 (2014): 1–12.

15. Y. Xiao, S. Chan, S. Wang, B. Zhu, and X. Yang, "An Adaptive Staggered-Tilted Grid for Incompressible Flow Simulation," *ACM Transactions on Graphics (TOG)* 39, no. 6 (2020): 1–15.

16. B. Adams, M. Pauly, R. Keiser, and L. J. Guibas, "Adaptively Sampled Particle Fluids," in *ACM SIGGRAPH 2007 Papers* (Association for Computing Machinery, 2007), 48.

17. R. Winchenbach, H. Hochstetter, and A. Kolb, “Infinite Continuous Adaptivity for Incompressible SPH,” *ACM Transactions on Graphics (TOG)* 36, no. 4 (2017): 1–10.

18. R. Winchenbach and A. Kolb, “Optimized Refinement for Spatially Adaptive SPH,” *ACM Transactions on Graphics (TOG)* 40, no. 1 (2021): 1–15.

19. X. Zhai, F. Hou, H. Qin, and A. Hao, “Fluid Simulation With Adaptive Staggered Power Particles on Gpus,” *IEEE Transactions on Visualization and Computer Graphics* 26, no. 6 (2018): 2234–2246.

20. S. Liu, X. Ban, S. Li, et al., “Fluidplaying: Efficient Adaptive Simulation for Highly Dynamic Fluid,” in *Proceedings of the 2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (IEEE, 2023), 831–832.

21. N. Chentanez, M. Müller, and T.-Y. Kim, “Coupling 3d Eulerian, Heightfield and Particle Methods for Interactive Simulation of Large Scale Liquid Phenomena,” *IEEE Transactions on Visualization and Computer Graphics* 21, no. 10 (2015): 1116–1128.

22. R. Nakanishi, F. Nascimento, R. Campos, P. Pagliosa, and A. Paiva, “RBF Liquids: An Adaptive Pic Solver Using RBF-FD,” *ACM Transactions on Graphics (TOG)* 39, no. 6 (2020): 1–13.

23. Y. Sun, T. Shinar, and C. Schroeder, “Effective Time Step Restrictions for Explicit MPM Simulation,” in *Computer Graphics Forum*, vol. 39 (Wiley Online Library, 2020), 55–67.

24. Y. Fang, Y. Hu, S.-M. Hu, and C. Jiang, “A Temporally Adaptive Material Point Method With Regional Time Stepping,” in *Computer Graphics Forum*, vol. 37 (Wiley Online Library, 2018), 195–204.

25. Y. Lian, P. Yang, X. Zhang, F. Zhang, Y. Liu, and P. Huang, “A Mesh-Grading Material Point Method and Its Parallelization for Problems With Localized Extreme Deformation,” *Computer Methods in Applied Mechanics and Engineering* 289 (2015): 291–315.

26. M. Gao, A. P. Tampubolon, C. Jiang, and E. Sifakis, “An Adaptive Generalized Interpolation Material Point Method for Simulating Elastoplastic Materials,” *ACM Transactions on Graphics (TOG)* 36, no. 6 (2017): 1–12.

27. J. F. Blinn, “A Generalization of Algebraic Surface Drawing,” *ACM Transactions on Graphics (TOG)* 1, no. 3 (1982): 235–256.

28. X. Wang, X. Ban, Y. Zhang, Z. Pan, and S. Liu, “Anisotropic Surface Reconstruction for Multiphase Fluids,” in *Proceedings of the 2017 International Conference on Cyberworlds (CW)* (IEEE, 2017), 118–125.

29. J. Yu, C. Wojtan, G. Turk, and C. Yap, “Explicit Mesh Surfaces for Particle Based Fluids,” in *Computer Graphics Forum*, vol. 31 (Wiley Online Library, 2012), 815–824.

30. G. Akinci, M. Ihmsen, N. Akinci, and M. Teschner, “Parallel Surface Reconstruction for Particle-Based Fluids,” in *Computer Graphics Forum. 31* (Wiley Online Library, 2012), 1797–1809.

31. J. Xing, L. Ruan, B. Wang, B. Zhu, and B. Chen, “Position-Based Surface Tension Flow,” *ACM Transactions on Graphics (TOG)* 41, no. 6 (2022): 1–12.

32. E. E. Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces* (University of Utah, 1974).

Supporting Information

Additional supporting information can be found online in the Supporting Information section. **Data S1.** Online version of the article at the publisher’s website.