DC-APIC: A Decomposed Compatible Affine Particle in Cell Transfer Scheme for Non-sticky Solid-Fluid Interactions in MPM

Chenhui Wang
School of Computer Science and Technology, East China Normal University
3663N. Zhongshan Road, Shanghai, China

Jianyang Zhang
School of Computer Science and Technology, East China Normal University
3663N. Zhongshan Road, Shanghai, China

Chen Li
School of Computer Science and Engineering, Tianjin University of Technology
No.391 Bin Shui Xi Dao Road, Tianjin, China

Changbo Wang
School of Data Science and Engineering, East China Normal University
3663N. Zhongshan Road, Shanghai, China

Abstract

Despite the material point method (MPM) provides a unified particle simulation framework for coupling of different materials, MPM suffers from sticky numerical artifacts, which is inherently restricted to sticky and no-slip interactions. In this paper, we propose a novel transfer scheme called Decomposed Compatible Affine Particle in Cell (DC-APIC) within the MPM framework for simulating the two-way coupled interaction between elastic solids and incompressible fluids under free-slip boundary conditions on a unified background grid. Firstly, we adopt particle-grid compatibility to describe the relationship between grid nodes and particles at the fluid-solid interface, which serves as the guideline for subsequent particle-grid-particle transfers. Then we develop a phase-field gradient method to track the compatibility and normal directions at the interface. Secondly, to facilitate automatic MPM collision resolution during solid-fluid coupling, in the proposed DC-APIC integrator, only the tangential component is transferred between incompatible grid nodes to prevent velocity smoothing in other phases, while the normal component is transferred without limitations. Finally, our comprehensive results confirm that our approach effectively reduces diffusion and unphysical viscosity compared to traditional MPM.

Keywords: material point method, solid-fluid coupling

1. Introduction

Simulations of fluid flow and solid deformation are crucial in the visual effects industry. In computer graphics, complex governing equations are typically discretized onto particles or grids. Particle methods, like smoothed particle hydrodynamics (SPH), are popular for simulating liquids and deformable solids due to their simplicity in tracking topological changes. However, SPH suffers from instability issues, density fluctuations, complex boundary handling and high computational cost in neighbour searching step. Grid-based methods show better stability and convenience on gradient calculation which is suitable for simulating smoke and elastic solids. However, grid-based methods experience significant numerical dissipation and have difficulty in accurately tracking geometric boundaries.

The need for combining advantages of both particle-based and grid-based methods has led deep research and application toward hybrid Lagrangian/Eulerian methods to achieve better topology tracking and numerical stability. In this paper, we focus on the Material Point Method (MPM), which is a generalized framework from Particle In Cell (PIC) and Fluid Implicit Particle Method (FLIP) to solid mechanics [26]. MPM employs a B-Spline kernel to facilitate the transfer of physical attributes between particles and grids. Particles are tasked with carrying essential physical quantities, including mass, velocity, position, and deformation gradient. Meanwhile, the grid is utilized for performing gradient computations and managing collision processing. This division of responsibilities allows MPM to leverage the strengths of both particles and grids, resulting in more accurate and stable simulations. PIC and FLIP can also be seen as MPM transfer scheme, which controls the procedure of transferring between particles and grids. The most important feature of these transfer schemes is that it ensures the conservation of mass and momentum during the transfer process. Several transfer schemes have been proposed in recent years. Jiang et al. [17] proposed the affine particle-in-cell (APIC) transfer scheme to overcome the dissipation problem in PIC and the stability issue in FLIP. Although APIC does make particle more energetic in most cases, it still suffers from severe numerical viscosity problem which results in the phenomenon that fluids and solids are difficult to separate. Fei et al. [8] revisited the intrinsic dissipation problem in MPM and designed a new transfer scheme ASFLIP to release particles adhering within the same scope of impact. Although ASFLIP is easy to implement, it suffers from instability for positional adjustment term modifying positions of particles in an Lagrangian way which should be governed by the energy equation. Meanwhile, Fang et al. [6] addressed MPM dissipation in another manner. They designed a monolithic solid-fluid governing equations solver and discretize the equations based on MPM pipeline to enforce free-slip condition in solid-fluid interface. However, this approach demands two different background grids and suffers from additional computation consumption.

In this paper, our ambitious goal is to design a novel and stable transfer scheme that tackles unphysical viscosity problem in traditional MPM on just one unified background grid. We call it Decomposed Compatible Affine Particle in Cell method(DC-APIC). It is important for our new transfer scheme to get precise geometry information like interface normals of both particles and grid nodes. The level-set method has a significant advantage in handling the complex geometry boundary and topology change in solid-fluid coupling animation, but its time cost is unacceptable. As a result, we develop a high-performance and easily-implemented phase-field gradient method for calculating compatibility, along with a precise normal estimation technique. Moreover, to maximize the utilization efficiency of our MPM method, we make a further enhanced workflow by advanced GPU parallelization. In particular, our salient contributions in this paper include:

(1) Decomposed Compatible APIC transfer scheme that overcomes the limitation of numerical viscosity in traditional MPM methods on a unified background grid. We modify the traditional MPM pipeline with Decomposed Compatible APIC (DC-APIC) integrator to enforce the free-slip boundary condition in tangential directions as well as separation condition in normal directions, which is further verified by various complex simulations with interactions between fluid and elastic solids.

(2) Phase-field gradient method for compatibility calculation. In order to find the solid-fluid boundary and compatibility between particles and grid nodes, we design a novel phase-field partition method with high precision and computational efficiency.

2. Related Work

2.1. Material Point Method

MPM is a hybrid method that unifies the treatment of solids and fluids by using Lagrangian particles to carry material information and an Eulerian grid for force computations. First introduced to computer graphics for the simulation of snow [24], MPM was later extended to handle phase transitions in multi-material simulations [25]. MPM can manage a variety of materials, including elastic-viscoplastic bodies and non-Newtonian fluids. Jiang et al. [17] proposed the affine particle-in-cell transfer scheme, reducing dissipation and offering greater stability than FLIP. Klar et al. [18] discretized the Drucker-Prager plastic model within MPM to simulate sand flow, which was further extended to sand-water coupling [27]. Jiang et al. [16] also modified the MPM workflow to support cloth simulation. Gao et al. [10] developed an adaptive MPM framework for sub-grid boundary processing and improved collision handling. One of the notable strengths of MPM is its ability to effectively handle fracture phenomena, making it particularly useful in simulations involving material breakage and failure. Wolpher et al. [31] presented continuum damage material point methods for animating dynamic fractures involving large elastoplastic deformations.

2.2. Two-way Fluid-Solid Coupling

Various approaches have been proposed in previous work for two-way coupling between solid bodies with incompressible fluids.

Batty et al. [3] developed a method for two-way coupling of Cartesian Eulerian fluids with Lagrangian rigid bodies having irregular geometry by casting the pressure solve as an energy minimization problem. Wolper et al. [19] proposed a method for animating fluid using unstructured dynamic tetrahedral meshes which can be coupled with lagrangian rigid bodies. Chentanez [5] formulated a monolithic non-symmetric linear system for simulating interaction between fluids and deformable solids; it was further extended to solving a symmetric positive definite (SPD) system by Zarifi [33]. Lagrangian particle based methods can also achieve stable solid-fluid coupling. Akinci et al. [2] proposed a coupling method of SPH fluids and arbitrary rigid objects based on hydrodynamic forces, which was later extended to support elastic solids [1].

After FLIP was introduced to the graphics community by Zhu [34], considerable research has been dedicated to hybrid particle-grid methods. Gao et al. [12] present a hybrid algorithm for simulating gaseous fluids that efficiently captures both low-speed and high-speed behaviors by combining grid and particle approaches. Losasso et al. [20] proposed a two-way coupled simulation framework combining the particle level set method for dense liquids and a novel SPH method for diffuse regions, enabling efficient modeling of both dense and diffuse water volumes with seamless mixing and interaction. Raveendran et al. [21] presented a hybrid method for enforcing incompressibility in SPH.

Although there are various methods for fluid-solid coupling, our focus is on MPM methods. MPM inherently supports the coupling of various materials and handles collisions on a grid. However, this also leads to issues with numerical viscosity. It's difficult for particles in the same B-Spline kernel width to separate from each other. As a consequence, solid-fluid interface in traditional MPM has a no-slip boundary condition. Han et al. [13] divide collision resolution into two aspects: the hyperelastic energy model, which provides a penalty for collisions, and the particle-grid-particle transfer scheme, which functions like a smooth kernel. Many people have previously dedicated efforts to solving this problem. Different grids for different material is the most common approach to solving this problem. After tracking the solid-fluid interface, the collision force is calculated based on grid velocity difference between solid phase and two phase, which is used to prevent particle collision between two phases. [32, 13]. However, this method is limited to explicit time integration and suffers from instability problem. Fang et al.[6] proposed a novel scheme for simulating strongly two-way coupled interactions between elastic solids and fluids using two different background grids, which uses ghost matrix operator-splitting and interface quadrature (IQ) discretization for free-slip boundary conditions. There are also solutions using a single grid in MPM. Hu et al. [15] proposed CPIC transfer scheme to enforce velocity discontinuity which supports better simulation of soft-body thin-rigid-body coupling and fluid thin-rigid-body coupling. However, it only supports rigid bodies with explicit geometric representation, making it unable to simulate interactions between fluids and elastic solids while enforcing a free-slip condition. Fei et al. [8] used an improved transfer scheme based on NFLIP idea to untrap the particles in MPM. It reduces diffusion and unphysical viscosity compared to traditional MPM integrators but it becomes unstable and prone to particle intersection issues if the positional adjustment parameter is not properly selected.

In particular, none of these prior works address the unphysical viscosity problem in two-way coupling between elastic solids and fluids using only one unified grid and achieve enough stability. Inspired by CPIC [15], we propose a novel transfer scheme called Decomposed Compatible Affine Particle in Cell(DC-APIC) within the MPM framework which is able to stably simulate the two-way coupled interaction between elastic solids and incompressible fluids under free-slip boundary conditions on a unified background grid. Compared to other methods [6, 13], DC-APIC does not significantly alter the MPM workflow, allowing for the extension of other MPM-based algorithms, such as elastic-viscoplastic phase transition simulation [29].

3. MPM Background

3.1. Elastoplasticity

In MPM, the material is represented by a collection of particles that carry information about the material's properties, such as mass, velocity and position. Similar to other PIC/FLIP solvers, MPM solves the governing equations on a background Eulerian grid. Stress-based forces are computed on the grid and then transferred back to the particles, which are moved in a Lagrangian manner.

The deformation theory of MPM is from the continuum mechanics. Material particles in the undeformed space with position $\mathbf{X}$ is mapped to the deformed space with position $\mathbf{x}$ as $\mathbf{x} = \phi(\mathbf{X})$, where ϕ is the deformation mapping. The deformation gradient $\mathbf{F}$ is the the Jacobian of ϕ as $\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}}(\mathbf{X}, t)$. The governing function consists of mass conservation and momentum conservation:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0, \quad \rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \sigma + \rho \mathbf{f}_{ext}, \qquad (1)$$

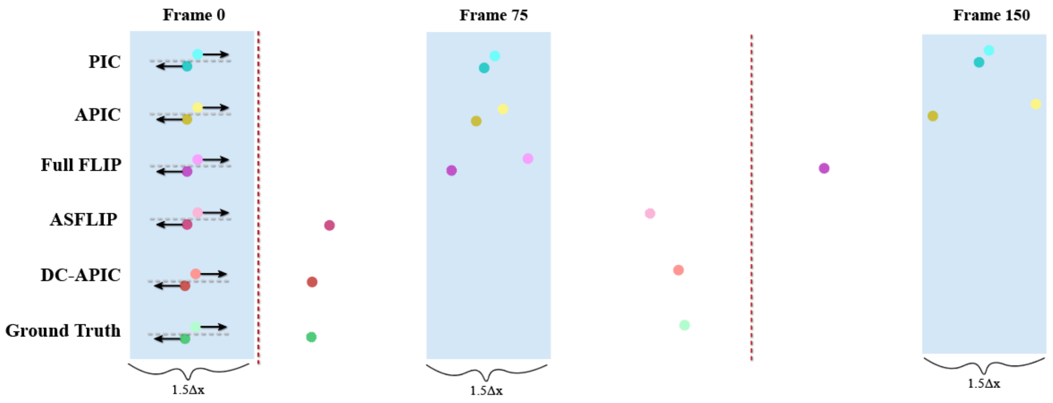

Figure 1. Advection of two particles for different integrators. Two 2D particles are moving away with same velocity magnitude. The grid size is Δx and the initial distance between them is $0.2\Delta x$ for both direction(x and y). No external force is exerted and inner force generated by constitutive model is set to zero. We adopt quadratic B-Spline kernel, width of which is $1.5\Delta x$ (Blue colored scope in figure). The ground truth is calculated based on uniform linear motion.

where ρ is the density, $\mathbf{v}$ is the velocity, σ is the Cauchy stress tensor, $\mathbf{f}_{ext}$ is the external force such as gravity. The deformation gradient can be decomposed as $\mathbf{F} = \mathbf{F^E F^P}$, where $\mathbf{F^E}$ and $\mathbf{F^P}$ are elastic and plastic parts of $\mathbf{F}$ respectively. The stress and strains can be written as

$$\sigma = \frac{1}{J}\frac{\partial \psi}{\partial \mathbf{F}}, \ J = det(F), \tag{2}$$

where J is the determinant of $\mathbf{F}$ representing the volume change of material particles. MPM can be adopted to simulate a wide variety of materials through the use of constitutive relations, which describe the relationship between stress and strain. In this paper, we use the Neo-Hookean model [30] to formulate the derivation of plastic and fluid materials. The elastic energy ψ is split into the volumetric part ψ_{vol} and deviatoric part ψ_{dev}:

$$\begin{aligned} \psi_{dev} &= \frac{\mu}{2}(J^{-2/d} tr(\mathbf{FF}^T) - d), \\ \psi_{vol} &= \frac{\kappa}{2}(\frac{J^2-1}{2} - log(J)), \end{aligned} \tag{3}$$

where d is the dimension, μ is the shear modulus and κ is the bulk modulus. Then the stress can be written as:

$$\begin{aligned} \tau &= \frac{1}{J}\sigma, \\ \tau &= \tau_{vol} + \tau_{dev}, \\ \tau_{vol} &= \frac{\kappa}{2}(J^2-1)\mathbf{I} + \mu dev(J^{-2/d}\mathbf{FF}^T), \end{aligned} \tag{4}$$

where τ is the Kirchhoff stress, $dev(\mathbf{A}) = \mathbf{A} - \frac{1}{d}tr(\mathbf{A})\mathbf{I}$ for any tensor A denotes the deviatoric part of the tensor.

Most materials also exhibit plastic deformation when shear stress exceeds specific yield conditions, which can be described as a yield function y. Any stress violating the condition(i.e., $y > 0$) should be projected to the yield surface using the process called return mapping. In our paper, we adopt Drucker-Prager model [18] for granular materials and fluid. To avoid volume gain artifact, we use the improved return mapping algorithm [28] by tracking and recovering volume changes.

3.2. Transfer Scheme

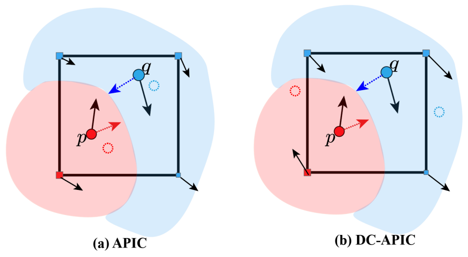

Figure 2. Analysis of dissipation in MPM. Area are colored to distinguish different phase domains, where red stands for solid area and blue stands for fluid. (a) Black arrow of particles represents velocity. Red arrow is the normal of solid particle while blue one is fluid normal vector. Traditional MPM transfer scheme APIC has difficulty in separating particles in same B-spline kernel scope since their tangential component are averaged in the P2G step. Although solid particle (the red one) shows potential to the up direction, it can only be advected to the right-down corner direction to the dashed red circle place. (d) Our novel DC-APIC can help calculate the interface geometry and give instruction for particles to the correct position.

In MPM, the transfer scheme refers to the process of transferring physical quantities between the material points and the Eulerian grid at each simulation step. Since MPM combines the advantages of both Lagrangian and Eulerian frameworks, the transfer between the two domains is crucial for accurately modeling material behavior. In the following context, superscripts n, $n+1$ represent quantities at time t^n and t^{n+1}, and for brevity, we set $\Delta t = t^{n+1} - t^n$, $\Delta \mathbf{x}_{ip}^n = \mathbf{x}_i - \mathbf{x}_p^n$, and $w_{ip} = N_i(\mathbf{x}_p)$ for short, where $N_i(\mathbf{x}_p)$ denotes the quadratic B-spline interpolation function between node i and particle p, used in transfer steps.

The standard PIC scheme transfers mass m_p, position $\mathbf{x}_p$, and velocity $\mathbf{v}_p$ from particles to Eulerian grids according to:

$$\begin{aligned} P2G:\ & m_i \mathbf{v}_i^n = \sum_p w_{ip} m_p \mathbf{v}_p^n, \\ G2P:\ & \mathbf{v}_p^{n+1} = \sum_i w_{ip} \mathbf{v}_i^*, \\ & \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_i w_{ip} \mathbf{v}^* \end{aligned} \tag{5}$$

where $\mathbf{v}_i^*$ denotes grid velocity after updating by stress and collision resolving. PIC transfer scheme is stable but suffers from excessive dissipation.

FLIP transfer scheme [34] adds an additional particle velocity preserving term to mitigate the dissipation in PIC as:

$$\begin{aligned} P2G:\ & m_i \mathbf{v}_i^n = \sum_p w_{ip} m_p \mathbf{v}_p^n, \\ G2P:\ & \mathbf{v}_p^{n+1} = \sum_i w_{ip} \mathbf{v}_i^* + \alpha(\mathbf{v}_p^n - \sum_i w_{ip} \mathbf{v}_i^n), \\ & \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_i w_{ip} \mathbf{v}^*, \end{aligned} \tag{6}$$

where α is the PIC-FLIP blending ratio [4]. With $\alpha = 0$, Eq. (6) turns back to PIC, and with $\alpha = 1$, it is full FLIP. Although FLIP transfer scheme makes simulation more energetic by additional particle velocity preserving term, it suffers from instability.

The APIC [17] scheme preserves angular momentum and prevents instabilities, which also reduces part of dissipation. We summarize the P2G and G2P transfers of APIC as:

$$\begin{aligned} P2G:\ & m_i \mathbf{v}_i^n = \sum_p w_{ip} m_p (\mathbf{v}_p^n + \mathbf{B}_p^n (\mathbf{D}_p^n)^{-1} \Delta \mathbf{x}_{ip}), \\ G2P:\ & \mathbf{v}_p^{n+1} = \sum_i w_{ip} \mathbf{v}_i^*, \\ & \mathbf{B}_p^{n+1} = \sum_i w_{ip} \mathbf{v}_i^* \Delta \mathbf{x}_{ip}, \\ & \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_i w_{ip} \mathbf{v}^*, \end{aligned} \tag{7}$$

where $\mathbf{D}_p^n$ is analogous to an inertia tensor and is given by

$$\mathbf{D}_p^n = \sum_i w_{ip} \Delta \mathbf{x}_{ip} (\Delta \mathbf{x}_{ip})^T \tag{8}$$

The locally velocity affine term $\mathbf{B}_p \mathbf{D}_p^{-1}$ helps preserve energy that would otherwise be lost by transfer schemes such as PIC due to numerical viscosity. However, APIC is still based on the assumption of material continuity, and therefore, it cannot effectively handle particle separation.

The Affine-augmented Separable FLIP(ASFLIP) scheme[8] is designed to address cases where the material continuity assumption may not hold, such as for highly dispersed sand. It adds the particle velocity preserving term in FLIP and an additional positional adjustment term to APIC for easier particle separation:

$$\begin{aligned} P2G:\ & m_i \mathbf{v}_i^n = \sum_p w_{ip} m_p (\mathbf{v}_p^n + \mathbf{B}_p^n (\mathbf{D}_p^n)^{-1} \Delta \mathbf{x}_{ip}), \\ G2P:\ & \mathbf{v}_p^{n+1} = \sum_i w_{ip} \mathbf{v}_i^* + \alpha(\mathbf{v}_p^n - \sum_i w_{ip} \mathbf{v}_i^n), \\ & \mathbf{B}_p^{n+1} = \sum_i w_{ip} \mathbf{v}_i^* \Delta \mathbf{x}_{ip}, \\ & \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t (\sum_i w_{ip} \mathbf{v}^* + \beta_p \alpha (\mathbf{v}_p^n - \sum_i w_{ip} \mathbf{v}_i^n)), \end{aligned} \tag{9}$$

where β_p is the ratio to control the magnitude of the positional adjustment. However, it still suffers from numerical viscosity. This issue becomes particularly pronounced in fluid-solid coupling problems, where both solids and fluids are discretized into MPM particles, making it challenging for the two phases to separate smoothly. In addition, the effectiveness of ASFLIP is influenced by parameters, making it more prone to penetration artifacts, as illustrated in Fig. 12.

For hybrid particle-grid methods, numerical viscosity arises from the transfer between the particles and the grid. Typically, the number of particles is much greater than the number of grid nodes. As a result, the P2G step can cause information loss. As illustrated in Fig 2(a), when two particles with equal mass and opposite velocities transfer momentum to the grid, the velocity will be averaged. From another perspective, MPM is based on the assumption of material continuity, making it difficult to physically describe material separation, especially in fluid-solid coupling cases. The velocity field at the interface of two phases can easily be smoothed by P2G.

4. Method

4.1. Particle Grid Compatibility

We adopt the concept of compatibility from CPIC [15] to mark the relationship between particle and grid node. In CPIC, a particle p and a node i are compatible when they are within the same domain that is not separated by a rigid body collision. Since multiple thin rigid bodies can divide an elastic body into two or more regions, two particles compatible with the same grid node may belong to different regions.

In our work, an additional property s to each particle is assigned representing the material state from solid($s = 1$) to fluid ($s = 0$). We define particle p and node i as compatible when they share the same solid-fluid phase. Compatibility c_{ip} for particle p and node i is equal to 1 only if $s_p = s_i$. Given that only two phases (solid and fluid) exist, two particles compatible with the same grid node must belong to the same region. Therefore, there is no need to use the CDF data structure in [15] to compute the compatibility.

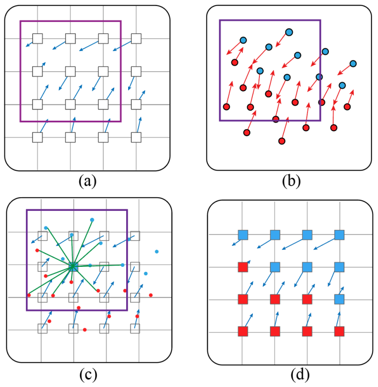

Figure 3. Phase-field gradient method for particle grid compatibility. The quadratic B-Spline kernel scope is annotated with purple square. (a) The background grid along with the scope of B-spline kernel, in the center of which lies the grid node to be calculated. The blue arrow is the gradient of grid node phase value ∇s_i (b) The particle color indicates the solid-fluid phase. The blue color represents fluid particles, while the red color represents solid particles. The red arrow is material weighted particle phase gradient value $\mathbf{Q}_p$. (c) Particles and background grid nodes. The green arrows indicate the estimated grid node phase contribution gained by center node from nearby particles calculated by $\nabla s_i \cdot \mathbf{Q}_p$. Finally the grid node phase is valued by signal of total contribution it achieved. (d) Phase value of all grid nodes are calculated.

We can clearly use signed distance functions (SDFs) to calculate the grid node phase s_i, but its time complexity is very high which takes almost half of each time step in our implementation. As a result, we design a phase-field gradient method which is initially used for fracture simulation [14]. First, we estimate grid phase gradient by particle phase state as $\nabla s_i = \sum_p s_p \nabla w_{ip}$. Second we use $\mathbf{Q}_p$ to denote material weighted particle phase gradient and $\mathbf{Q}_p = (-1)^{s_p} \sum_i \nabla s_i w_{ip}$. Then we partition the grid node into different phase based on the sign of the dot product of the material weighted phase gradient at the particle and phase gradient at the node as

$$s_i = \begin{cases} s_p, & \operatorname{sgn}(\nabla s_i \cdot \mathbf{Q}_p) > 0 \\ 1 - s_p. & \text{otherwise.} \end{cases} \tag{10}$$

However, two different phase particles may classify one grid node into their own phase at the same time, which means $\exists p, q, i, \textbf{s.t.} \operatorname{sgn}(\nabla s_i \cdot \mathbf{Q}_p) > 0 \wedge \operatorname{sgn}(\nabla s_i \cdot \mathbf{Q}_p) > 0$, where s_p is different from s_q. This pathological case can occur quite frequently at the sharp border of solid region. We employ a heuristic algorithm to solve this problem (as shown in Fig. 3). At the boundary area, the greater the length of the projection of the particle phase gradient onto the grid phase gradient direction $\|\nabla s_i \cdot \mathbf{Q}_p\|$, the more significant its impact on determining the grid node phase. Therefore, we use the dot product of the gradients as the weight for determining the phase of the grid node. We define $I_i = \sum_p (\nabla s_i \cdot \mathbf{Q}_p)$. If $I_i > 0$, we classify node i as a solid node and assign $s_i = 1$. Conversely, if $I_i < 0$, we classify node i as a fluid node and assign $s_i = 0$.

We also implement the SDF-based compatibility calculation method, which reconstructs solid particle level set and queries each grid node to get its phase. This is a bit more correct than phase field gradient method but suffers from time consumption in building level set. We compare the result of these two methods in Fig. 6.

4.2. Particle-grid Transfer

4.2.1 DC-APIC Particle-to-Grid Transfer

We use subscripts p, q for particle quantities and i, j for grid quantities. i^{p+} denotes nodes that are compatible with particle p, and i^{p-} denotes the incompatible nodes. Similarly, p^{i+} are the particles that are compatible with grid node i, and p^{i-} are the incompatible particles. We will still use APIC transfer scheme in most parts of simulation objects. However at the interface, particles only transfer the normal component of momentum to incompatible grid nodes to avoid smoothing velocity in another phase and transfer both tangential and normal component of momentum to compatible grid nodes to support automatic MPM collision solving during fluid-solid coupling:

$$\begin{aligned}
m_i &= \sum_{q\in\{p^{i+},p^{i-}\}} w_{iq} m_q, \\
m_{i+} &= \sum_{q\in\{p^{i+}\}} w_{iq} m_q, \\
(m_i\mathbf{v}_i)^{norm} &= \sum_{q\in\{p^{i+},p^{i-}\}} w_{ip} m_q(\mathbf{v}_q\mathbf{n}_q + \mathbf{B}_q\mathbf{D}_q^{-1}\Delta\mathbf{x}_{iq}), \\
(\mathbf{m}_i\mathbf{v}_i)^{tan} &= \sum_{q\in\{p^{i+}\}} w_{iq} m_q(\mathbf{v}_q - \mathbf{v}_q\mathbf{n}_q), \\
\mathbf{v}_i &= (m_i\mathbf{v}_i)^{norm}/m_i + (m_i\mathbf{v}_i)^{tan}/m_{i+},
\end{aligned} \tag{11}$$

where m_i is the total mass transferred to a node by mapped particles and m_{i+} is only the mass transferred by compatible particles, the locally velocity affine term $\mathbf{B}_q\mathbf{D}_q^{-1}$ helps preserve energy that would otherwise be lost by transfer schemes such as PIC due to numerical viscosity. We divide momentum into normal component and tangential component because they need to be divided by different mass in velocity calculation step. To be detailed, if the tangential velocity is $(m_i\mathbf{v}_i)^{tan}/m_i$, the velocity will be dissipated.

As illustrated in Fig. 2, it is relatively difficult for APIC scheme to separate particles moving in opposite directions within the same grid node (within the BSpline kernel range), because the particle-to-grid operation averages the momentum of particles within the kernel range, resulting in numerical viscosity. In contrast, DC-APIC decomposes the momentum into tangential and normal components, transmitting only the tangential momentum to the compatible grid node. This avoids the averaging influence of particles from another phase. Additionally, since the normal velocity is still transferred normally and not directly adding an position-adjust term to $\mathbf{x}_p$, it effectively resolves the issue of particle interpenetration found in NFLIP [24] and can be more stable than ASFLIP [8].

4.2.2 DC-APIC Grid-to-Particle Transfer

For each particle, the tangential velocities on incompatible grid nodes are non-associated with the particle due to the enforcement of discontinuity. We take a ghost velocity approach, where we assume for any node $j \in i^{p-}$, its velocity is simply $v_j = v_p^n$ through a constant extrapolation from particle p. Thus the DC-APIC transfer from grid to particle which gathers contributions from both incompatible and compatible nodes is given by

$$\begin{aligned}
\mathbf{v}_p &= \sum_{j\in i^{p+}} w_{jp}\mathbf{v}_j^* + \sum_{j\in i^{p-}} w_{jp}\mathbf{v}_j^*\mathbf{n}_j + \sum_{j\in i^{p-}} w_{jp}(\mathbf{v}_p - \mathbf{v}_p\mathbf{n}_j), \\
\mathbf{B}_p &= \sum_{j\in i^{p+}} w_{jp}\mathbf{v}_j^*\mathbf{x}_{jp} + \sum_{j\in i^{p-}} w_{jp}(\mathbf{v}_p - \mathbf{v}_p\mathbf{n}_j)\Delta\mathbf{x}_{ip}.
\end{aligned} \tag{12}$$

Particles will gain velocity from compatible nodes as well as the normal component of velocity from incompatible nodes to achieve the automatic MPM coupling without self intersection. We only use the tangential component of ghost particles velocity in Eq. (12) for stability. To be noticed, the normal vector here has to be grid node normal n_i for the sake of momentum conservation. Momentum conservation can be rigorously proven in APIC, where the total particle momentum before and after G2P and P2G is equal to the total grid momentum. However DC-APIC uses CPIC techniques, where P2G selectively transfers partial momentum to the grid based on the geometry of the fluid-solid boundary, and during G2P, the lost momentum is transferred back to the particles using $\mathbf{v}_p^n$ as an approximation. From another perspective, our transfer scheme can be viewed as a composite of two operations: orthogonal decomposition and CPIC. Orthogonal decomposition obviously preserves total momentum. During P2G of CPIC, velocities transfer to compatible nodes. Then incompatible nodes transmit ghost velocities back to particles during G2P. These ghost velocities are the same as particle velocities from the previous timestep. Therefore, the entire P2G-G2P pipeline seems like making an interpolation between APIC Filter and Ground Truth, where Ground Truth refers to velocities keep the same as previous timestep, ensuring momentum conservation. A more detailed analysis of the effect of energy conservation will be presented in Section 6.

4.3. MPM Workflow with DC-APIC Scheme

Compared with traditional MPM workflow, we employ an additional step to obtain information about particle grid compatibility to regulate the grid-to-particle and particle-to-grid processes for material discontinuity in solid-fluid coupling. Fig. 5 showcases the main data flow of our numerical solver and the main workflow is listed below.

(1) Interface information calculation. Calculate both particle normal vectors n_p and grid node normal vectors n_i along the interface as well as grid-wise colored solid-fluid phase field state s_i to control DC-APIC particle-to-grid and grid-to-particle steps.

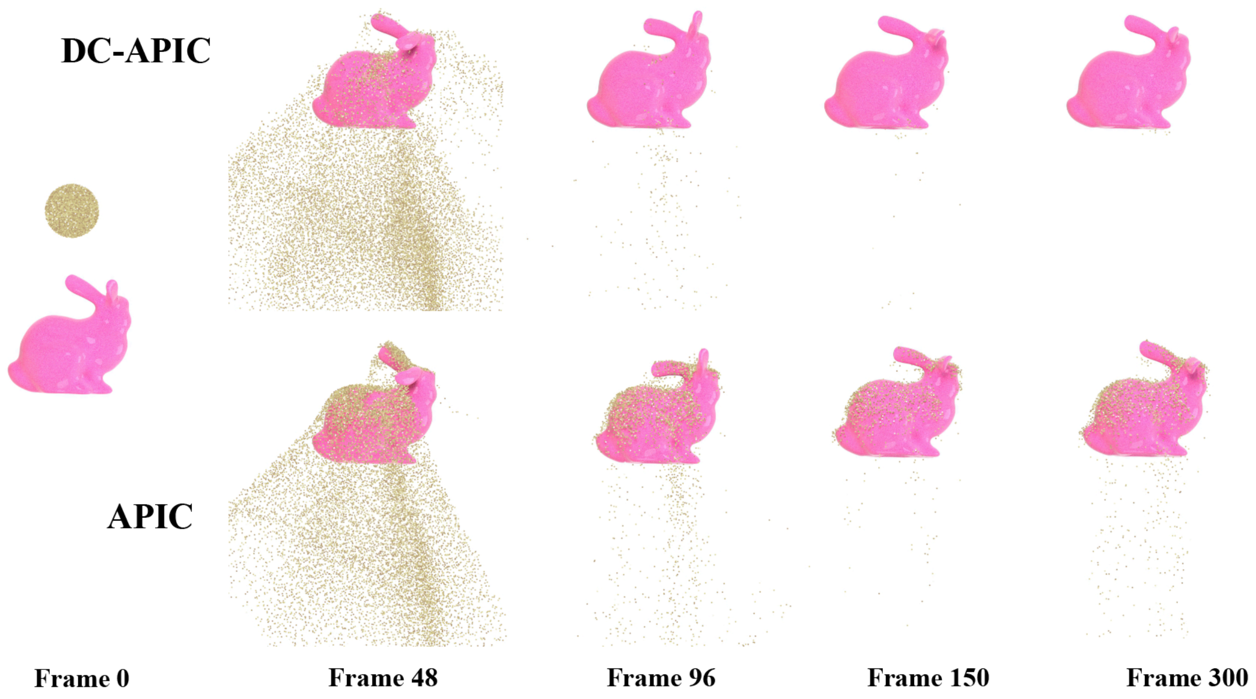

Figure 4. Pinned Bunny with Sand Dropped. Our DC-APIC method can effectively solve the solid-fluid coupling with free-slip condition while traditional MPM is hard to separate water and elastic solid bunny.

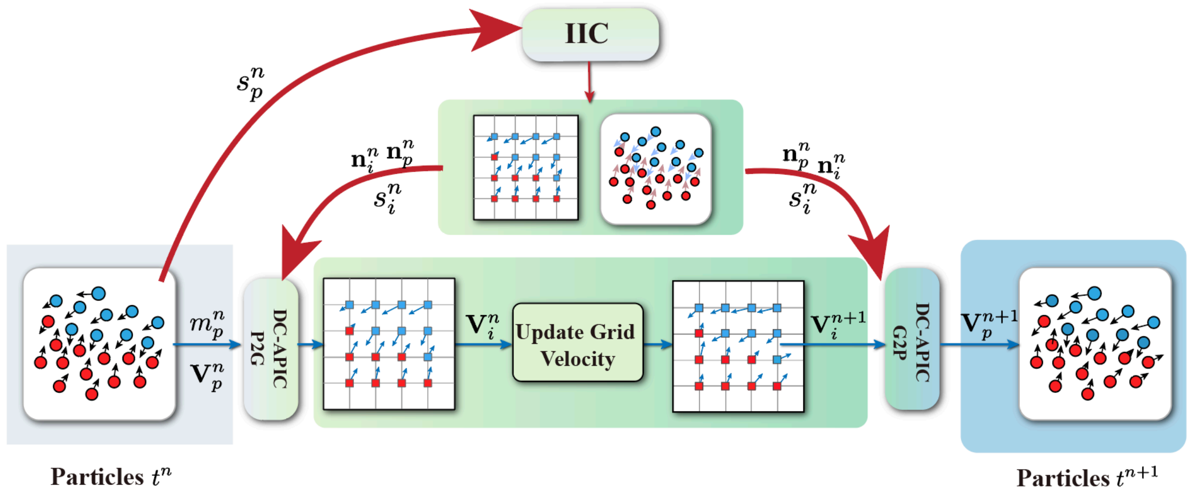

Figure 5. The data flow of our algorithm. The figure illustrates the data flow between particles and grid in one time step. The solid red lines denote the additional steps for DC-APIC scheme. Interface Information Calculation(IIC) step calculate the particle normal n_p, grid normal n_i and grid node solid-fluid phase state s_i at the beginning of each time step, which will be used for DC-APIC particle-to-grid (P2G) and DC-APIC grid-to-particle (G2P) step.

(2) Particles to grid. Transfer mass and momentum from particles to the grid according to DC-APIC scheme, which is described in detail in Section 4.2.1.

(3) Grid velocity update. Calculate the force applied to node i according to neighbor particles' deformation gradient $\mathbf{F}_p^n$, formulated as $\mathbf{f}_i^n = -\sum_p V_p^0 \frac{\partial \Psi}{\partial F} \mathbf{F}_p^{nT} \nabla w_{ip}^n$, and then update the velocity in a symplectic Euler manner as $\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{f}_i^n / m_i$.

(4) Grid to particles. Update the velocity $\mathbf{v}_\mathbf{p}^{\mathbf{n+1}}$, spatial velocity gradient of particle $(\nabla\mathbf{v})_p^{n+1}$ and affine state $\mathbf{B}_p^{n+1}$ in APIC according to Section 4.2.2, and then perform particle advection with collisions handling.

(5) Strain update. Update particle trial deformation gradient as $\mathbf{F}_p^{n+1} = (I + \Delta t \nabla \mathbf{v}_p^{n+1})\mathbf{F}_p^n$.

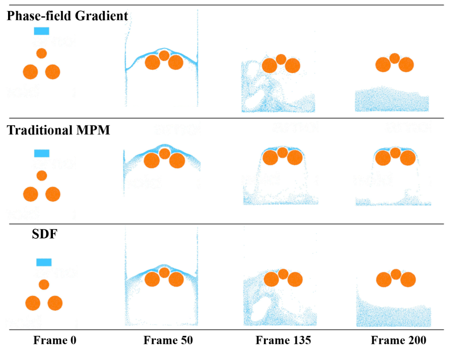

Figure 6. Water drops on elastic balls. (Top) Our novel phase-field gradient compatibility calculation based DC-APIC MPM. Water separate easily from free-slip elastic balls. (Middle) Coupling simulated by traditional MPM suffers from unphysical viscosity. It's hard for water to leave elastic body. (Bottom) SDF based DC-APIC which reconstruct level set to classify grid node phase, result of which is apparently the same as phase-field gradient but takes more time to compute.

4.3.1 Interface Information Calculation

At the first of IIC step, we use an additional P2G which only transfers phase field. Grid nodes that receive both fluid and solid phase are marked as boundary nodes, and particles mapped to these nodes are marked as boundary particles.

4.3.2 Normal Estimation

Traditional signed distance functions(SDFs) are convenient for performing inside and outside queries and normal estimations. However, it is expensive to reconstruct the particle level set at each time step of simulation. We use the same method as [6] by picking solid normals as the negative mass gradient field of solid particles.

$$\begin{aligned} m_i^s &= \sum_{p^s} m_p w_{ip}, \\ \mathbf{n}_p^s &= -\sum_i m_i^s \nabla w_{ip} / \| \sum_i m_i^s \nabla w_{ip} \|, \end{aligned} \tag{13}$$

where m_i^s is the solid mass mapped to one grid node, p^s denotes the solid particles and $\mathbf{n}_p^s$ denotes the solid particle normal. For grid normals and fluid particle normals we design a different strategy as:

$$\begin{aligned} \mathbf{n}_i &= \sum_{p^s} \mathbf{n}_p^s w_{ip} / \| \sum_{p^s} \mathbf{n}_p^s w_{ip} \|, \\ \mathbf{n}_p^f &= -\sum_i \mathbf{n}_i w_{ip} / \| \sum_i \mathbf{n}_i w_{ip} \|, \end{aligned} \tag{14}$$

where $\mathbf{n}_i$ is the grid node normal which can cover all interface grid nodes, $\mathbf{n}_p^f$ denotes the fluid particle normal. As we can see from Eq. (14), first we apply the B-spline smooth kernel on solid particle normals $\mathbf{n}_p^s$ to get grid node normals. The smoothing kernel can make DC-APIC grid-to-particle (Section 4.2.1) and particle-to-grid (Section 4.2.2) more stable. Then we calculate the fluid particle normals using grid node normals. This strategy is both stable and efficient, avoiding the need to find the nearest solid particles through spatial data structures when computing fluid normals.

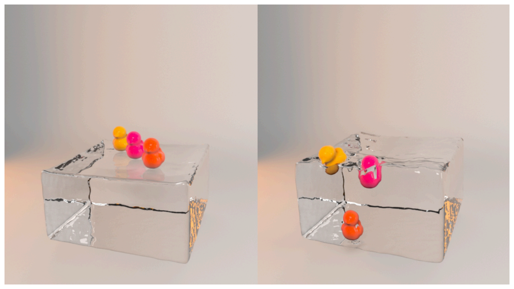

Figure 7. Rubber duck toy with different density dropped into water. From left to right, the duck-water density ratio is 0.5, 1.0 and 2.5.

5. GPU Implementation

In this section, we will briefly introduce the storage, reorder, and parallel scheme of GPUMPM [11], as we implement DC-APIC based on it. We will then present the additional components, including a specialized warp write algorithm.

5.1. GSPGrid Overview

The GSPGrid [11] is the GPU adaptation of the SPGrid [23], which is an alternative sparse data structure. Its particle-reorder algorithm and write-back scheme work effectively on the parallel architecture of CUDA. In particle domain, particle indices are divided into target-pages from different blocks which include $4 \times 4 \times 4$ grids. Particles in one block may easily exceed the maximum number of threads in a CUDA block. To address this issue, target pages are further divided into virtual pages containing up to 512 particles each. Simultaneously, in the grid domain, a 4KB page can store 64 grids that possess 16 single-precision float attributes.

Since the BSpline function serves as the weight kernel, one particle is associated with $3 \times 3 \times 3$ grid nodes. In this situation, some particles may interact with grids belonging to other blocks. Therefore, GPUMPM pre-calculates the mapping between each block and its seven neighbors with off-by-one staggering strategy. Neighbors are the ones with a positive offset alongside at least one axis. If the dimension of simulation is three, there should be seven neighbors. Additionally, it utilizes shared memory for all $2 \times 2 \times 2$ blocks in the MPM transfer kernel, such as during the particle-to-grid step, to resolve cross-block write conflicts and cross-warp write conflicts. This means that every CUDA block handles one virtual page (512 particles) and writes back to eight blocks (one block and its seven neighboring blocks).

Algorithm 1 Specialized Warp Write

```
1:  procedure GatherAndWrite(T* $buffer$, T $mass$, int $tag$, int $iter$)
2:      $stride \leftarrow 1$
3:      $kind \leftarrow tag$                    ▷ // 1 for solid, 0 for fluid
4:      if $kind$ then        ▷ // only reduce solid particle mass
5:          $val \leftarrow mass$
6:      else
7:          $val \leftarrow 0$
8:      while $stride <= iter$ do
9:          $tmp \leftarrow$ shfldown$(val, stride)$
10:         if $stride <= interval$ then
11:             $val \leftarrow val + tmp$
12:         $stride \leftarrow stride << 1$
13:     if boundary then
14:         $*buffer \leftarrow *buffer + val$
```

5.2. Specialized Warp Write for Non-Consecutive Data

For most common serial data, such as velocity and stress, summation can be easily performed using traditional warp write algorithms that include CUDA primitives. However, in our project, solid mass for the grid needs to be calculated based on affected solid particles. The best way to distinguish solid from fluid particles is to assign a tag when they are sampled. As a result, Algorithm 1 must be slightly modified to accommodate new attributes. When the warp computation algorithm [11] is completed, each particle governed by a thread will know whether it is the first particle in the current warp and how many particles remain. Fig. 8 illustrates how this procedure works.

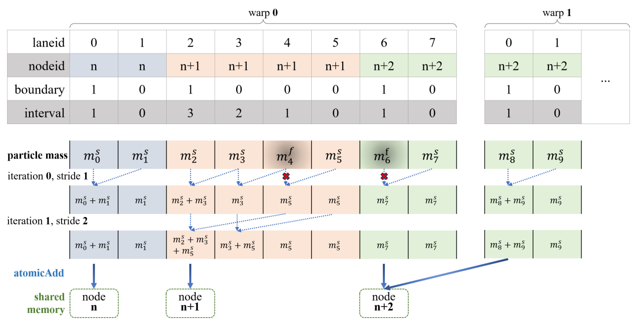

Figure 8. Reduce solid particle masses. It shows a way to reduce non-consecutive data like different kind particle's mass. Besides, it demonstrates solving cross-warp-write-conflict by shared memory.

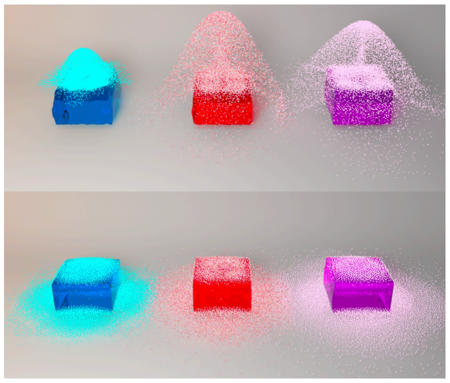
Figure 9. Sand jelly. Elastic Jell-O's with varying Young's modulus are two-way coupled with sand particles.

It is important to note that basic particle attributes have distinct reorder schemes tailored to their intrinsic characteristics, e.g., mass, position, and velocity. For example, the sequence of masses remains unchanged since they are constant; the only requirement is to map the current order of particles back to their original order. The additional particle attributes also require different sorting methods; for instance, tags are treated similarly to masses while normals are treated similarly to velocities.

6. Results

We evaluate the effectiveness of our DC-APIC MPM method by a number of solid-fluid coupling simulations. Our demos were performed on a desktop with Intel Core i7-8700K CPU at 3.70 GHz and NVIDIA GTX 2080TI. The parameter settings of the material and the performance statistics are given in Table 1. As shown in Fig. 14, DC-APIC uses a unified background grid, avoiding extra solid-fluid boundary solver, resulting in a 4.9× improvement in computational effi-

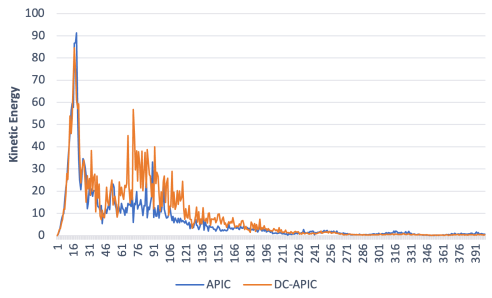

Figure 10. Kinetic Energy Curve. The kinetic energy is plotted over time for the duck case (Fig. 7). The rubber ducks enter the water at frame 50, leading to a subsequent period where the overall particle kinetic energy in DC-APIC is higher than in APIC for less numerical dissipation.

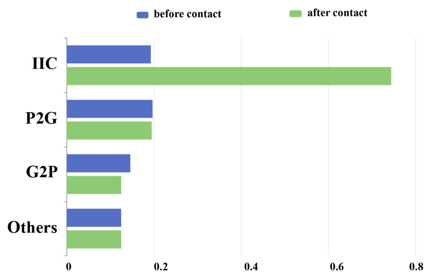

Figure 11. Timing Consumption. In the bear-bath example (Fig. 13), the average timing costs per time step before and after contact are 1.17s and 1.42s respectively. Others step includes reinitialize, grid velocity update, plastic update and particle state update by force.

ciency compared to IQ-MPM. Additionally, it simplifies the implementation, allowing for easy extension of other simulation algorithms based on DC-APIC, reducing numerical instability at fluid-solid interfaces at the same time. Time consumption of each step is shown in Fig. 11. We also present an analysis of the kinetic energy stored in particles in Fig. 10. After the fluid impacts the solid, the kinetic energy in the APIC method dissipates more quickly compared to that in the DC-APIC method.

In Fig. 4, we show the comparison between our DC-APIC MPM and traditional MPM. We also provide the 2D comparison between phase-field DC-APIC, SDF DC-APIC and traditional MPM in Fig. 6. As shown in sub-figures, traditional MPM suffers from numerical viscosity problem while our DC-APIC method effectively overcome this shortcoming. At the same time, we prove that our phase-field gradient method to calculate compatibility is precise enough to compare with time-consuming SDF-based method. It can reach 2.7× speedup than SDF-based method in 2D case with no obviously artifacts.

In Fig. 1 we compare our DC-APIC integrator to

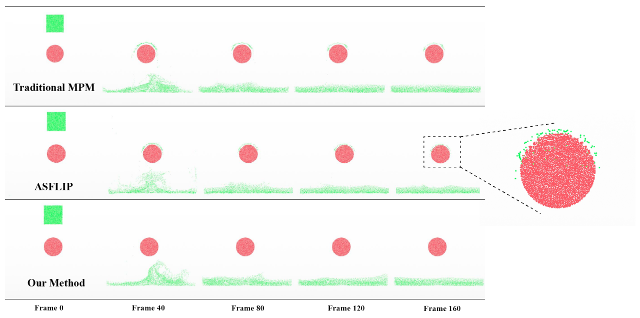

Figure 12. Comparison of the solid-fluid interaction between our methods and ASFLIP [8].

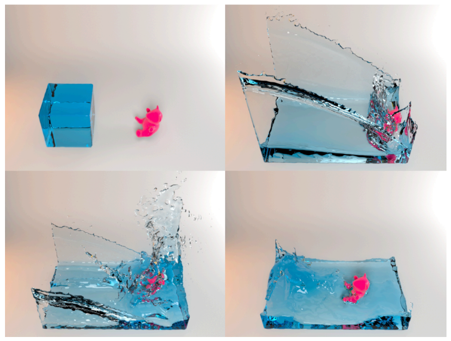

Figure 13. Bear bath. A dam breaks, hitting a hyperelastic bear and sweeping it away. Water particles flow bypass elastic surface correctly with no unphysical viscosity.

other earlier transfer schemes. At the beginning, all test cases have two particles of different phase, the upper one moving right and the lower one moving left. At the end frame, only Full FLIP, ASFLIP and our DC-APIC can untrap the B-Spline kernel smoothing scope. It's obviously seen our result is closest to ground truth due to tangential velocity only transferred to compatible grid nodes. Although ASFLIP can handle scenarios where the material continuity assumption breaks down, such as highly dispersed sand, it still suffers from significant numerical viscosity. This issue becomes especially evident in fluid-solid coupling problems, where both solids and fluids are discretized into MPM particles, making it difficult for the two phases to separate smoothly. Furthermore, the performance of ASFLIP is sensitive to parameter settings, which increases its susceptibility to penetration artifacts. As illustrated in Fig. 12, we present a simple case comparing the performance of DC-APIC and ASFLIP.

Moreover we present several examples with different density of solid and fluid. We can demonstrate that our method correctly handles the buoyancy of solids in Fig. 13 and Fig. 7.

To further validate the robustness of our method, we simulate sand-like fluid with plasticity interacting with

Table 1. Parameters and performance.

Example		sec/frame	Δx	Δt	#Particles	Young's Modulus	Poisson's Ratio	density ratio
(Fig. 4 left)	Pinned bunny	6.33	0.1	0.001	98K	8000	0.4	1.0
(Fig. 4 right)	Pinned bunny	3.60	0.1	0.001	98K	8000	0.4	1.0
(Fig. 13)	Bear bath	24.17	0.1	5×10^{-4}	400K	6000	0.4	1.2
(Fig. 7)	Duck	57.63	0.1	0.001	640K	6000	0.4	0.5/1.0/2.5
(Fig. 9 left)	Jelly Sand	14.72	0.05	5×10^{-4}	1.2M	1500	0.2	1.0
(Fig. 9 middle)	Jelly Sand	17.16	0.05	5×10^{-4}	1.2M	4000	0.3	1.0
(Fig. 9 right)	Jelly Sand	15.91	0.05	5×10^{-4}	1.2M	6000	0.4	1.0
(Fig. 6 top)	Balls(2D)	0.78	0.05	0.001	8K	4000	0.4	1.0
(Fig. 6 middle)	Balls(2D)	0.42	0.05	0.001	8K	4000	0.4	1.0
(Fig. 6 bottom)	Balls(2D)	1.14	0.05	0.001	8K	4000	0.4	1.0

a jelly cube in Fig. 9. The sand uses Drucker-Prager plastic model [18] where friction angle ϕ_f is 30.

7. Conclusion and Future Work

While DC-APIC enables free-slip solid-fluid coupling for MPM at low cost since only a narrow band near solid-fluid interface needs DC-APIC, it still suffers from resolution problem. We can obviously see the one Δx gap at interface area for it cannot handle sub-level topology as sharp corners and narrow gaps. Hence for realistic simulation, we have to decrease Δx which may cause instability problem and more time consumption. Adaptive grid MPM [10] combined with DC-APIC is a promising future direction which can use higher resolution grids in the solid-fluid interface area to support more narrow gap. The coupling between porous materials and fluids is also an issue worth paying attention to [22].

Fully implicit elastic-MPM strong coupling using DC-APIC on one unified background grid is not formulated in this work and we leave that as future work.

Although our DC-APIC transfer scheme is stable in volumetric solids case, particles can still intersection with each other, which has been solved in SDF-based DC-APIC method with a penalty force based on penetration distance in level set. However, there's still no solid resolution in phase-field gradient DC-APIC method for lack of penetration distance information.

Using DC-APIC in co-dimensional solids case remains quite challenging. It would be interesting to extend our work to support fluid interactions with cloth [7, 8] and hair [9, 8]. Another promising direction for future work is to modulate the tangential interaction between incompatible grid nodes and particles to better account for frictional effects.

Acknowledgments

This article is partially supported by Natural Science Foundation of China under Grants 62002121 and 62472178, National Science Foundation of USA (IIS-1715985 and IIS-1812606), Science and Technology Commission of Shanghai Municipality (No.22511104600). The authors would like to express sincere gratitude to these institutions for their support, as well as to all the reviewers for their thoughtful and valuable suggestions.

References

[1] N. Akinci, J. Cornelis, G. Akinci, and M. Teschner. Coupling elastic solids with smoothed particle hydrodynamics fluids. Computer Animation and Virtual Worlds, 24(3-4):195–203, 2013. 3

[2] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner. Versatile rigid-fluid coupling for incompressible sph. ACM Transactions on Graphics (TOG), 31(4):1–8, 2012. 3

[3] C. Batty, F. Bertails, and R. Bridson. A fast variational framework for accurate solid-fluid coupling. ACM Transactions on Graphics (TOG), 26(3):100–es, 2007. 3

[4] R. Bridson. Fluid simulation for computer graphics. AK Peters/CRC Press, 2015. 5

[5] N. Chentanez, T. G. Goktekin, B. E. Feldman, and J. F. O'Brien. Simultaneous coupling of fluids and deformable bodies. 2006. 3

[6] Y. Fang, Z. Qu, M. Li, X. Zhang, Y. Zhu, M. Aanjaneya, and C. Jiang. 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(4):51–1, 2020. 2, 3, 9

[7] Y. Fei, C. Batty, E. Grinspun, and C. Zheng. A multi-scale model for simulating liquid-fabric interactions. ACM Transactions on Graphics (TOG), 37(4):1–16, 2018. 12

[8] Y. Fei, Q. Guo, R. Wu, L. Huang, and M. Gao. Revisiting integration in the material point method: a scheme for easier separation and less dissipation. ACM Transactions on Graphics (TOG), 40(4):1–16, 2021. 2, 3, 5, 7, 11, 12

[9] Y. Fei, H. T. Maia, C. Batty, C. Zheng, and E. Grinspun. A multi-scale model for simulating liquid-hair interactions. ACM Transactions on Graphics (TOG), 36(4):1–17, 2017. 12

[10] M. Gao, A. P. Tampubolon, C. Jiang, and E. Sifakis. An adaptive generalized interpolation material point

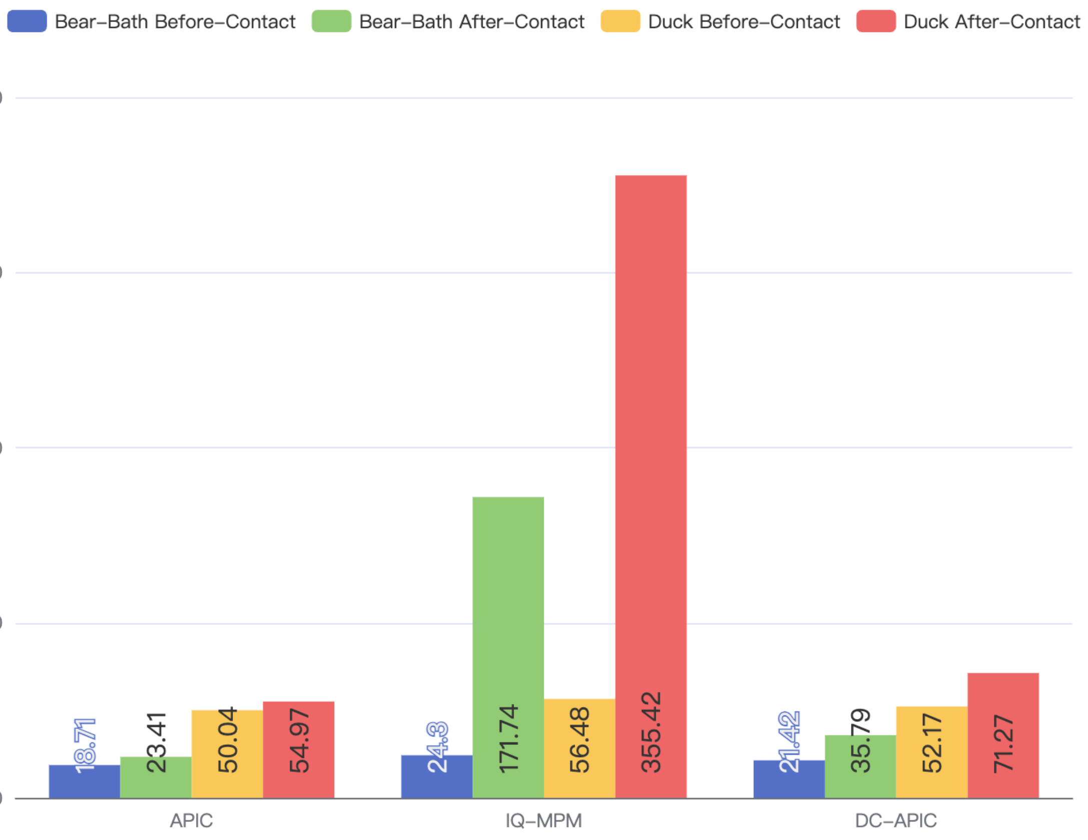

Figure 14. Benchmark experiments. The comparison of the average time consumption(sec/frame) for the three transfer schemes in the bear-bath and duck cases. Before the fluid-solid interaction, the average computational efficiency of the three methods is quite similar. However, after the interaction occurs, IQ-MPM requires significant time to compute the pressure at the fluid-solid boundary on the grid, leading to a much longer processing time compared to traditional MPM. In contrast, DC-APIC uses a unified background grid and does not require an additional solver, resulting in computational speeds comparable to traditional MPM.

method for simulating elastoplastic materials. ACM Transactions on Graphics (TOG), 36(6):1–12, 2017. 2, 12

[11] M. Gao, X. Wang, K. Wu, A. Pradhana, E. Sifakis, C. Yuksel, and C. Jiang. Gpu optimization of material point methods. ACM Trans. Graph., 37(6), Dec. 2018. 9, 10

[12] Y. Gao, C.-F. Li, S.-M. Hu, and B. A. Barsky. Simulating gaseous fluids with low and high speeds. In Computer Graphics Forum, volume 28, pages 1845–1852. Wiley Online Library, 2009. 3

[13] X. Han, T. F. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran. A hybrid material point method for frictional contact with diverse materials. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2(2):1–24, 2019. 3

[14] M. A. Homel and E. B. Herbold. Field-gradient partitioning for fracture and frictional contact in the material point method. International Journal for Numerical Methods in Engineering, 109(7):1013–1044, 2017. 6

[15] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. ACM Transactions on Graphics (TOG), 37(4):1–14, 2018. 3, 6

[16] C. Jiang, T. Gast, and J. Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. ACM Transactions on Graphics (TOG), 36(4):1–14, 2017. 2

[17] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. The affine particle-in-cell method. ACM Transactions on Graphics (TOG), 34(4):1–10, 2015. 2, 5

[18] G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran. Drucker-prager elastoplasticity for sand animation. ACM Transactions on Graphics (TOG), 35(4):1–12, 2016. 2, 4, 12

[19] B. M. Klingner, B. E. Feldman, N. Chentanez, and J. F. O'brien. Fluid animation with dynamic meshes. In ACM SIGGRAPH 2006 Papers, pages 820–825. 2006. 3

[20] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw. Two-way coupled sph and particle level set fluid simulation. IEEE transactions on visualization and computer graphics, 14(4):797–804, 2008. 3

[21] K. Raveendran, C. Wojtan, and G. Turk. Hybrid smoothed particle hydrodynamics. In Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation, pages 33–42, 2011. 3

[22] B. Ren, B. Xu, and C. Li. Unified particle system for multiple-fluid flow and porous material. ACM Trans. Graph., 40(4), July 2021. 12

[23] R. Setaluri, M. Aanjaneya, S. Bauer, and E. Sifakis. Spgrid: a sparse paged grid structure applied to adaptive smoke simulation. ACM Trans. Graph., 33(6), nov 2014. 9

[24] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. A material point method for snow simulation. ACM Transactions on Graphics (TOG), 32(4):1–10, 2013. 2, 7

[25] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle. Augmented mpm for phase-change and varied materials. ACM Transactions on Graphics (TOG), 33(4):1–11, 2014. 2

[26] D. Sulsky, S.-J. Zhou, and H. L. Schreyer. Application of a particle-in-cell method to solid mechanics. Computer physics communications, 87(1-2):236–252, 1995. 2

[27] A. P. Tampubolon, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, and K. Museth. Multi-species simulation of porous sand and water mixtures. ACM Transactions on Graphics (TOG), 36(4):1–11, 2017. 2

[28] A. P. Tampubolon, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, and K. Museth. Multi-species simulation of porous sand and water mixtures. ACM Trans. Graph., 36(4), July 2017. 4

[29] Z. Tu, C. Li, Z. Zhao, L. Liu, C. Wang, C. Wang, and H. Qin. A unified mpm framework supporting phase-field models and elastic-viscoplastic phase transition. ACM Trans. Graph., 43(2), Jan. 2024. 3

[30] Z. Tu, C. Li, Z. Zhao, L. Liu, C. Wang, C. Wang, and H. Qin. A unified mpm framework supporting phase-field models and elastic-viscoplastic phase transition. ACM Trans. Graph., 43(2):19:1–19:19, April 2024. 4

[31] J. Wolper, Y. Fang, M. Li, J. Lu, M. Gao, and C. Jiang. Cd-mpm: continuum damage material point methods for dynamic fracture animation. ACM Transactions on Graphics (TOG), 38(4):1–15, 2019. 2

[32] X. Yan, C.-F. Li, X.-S. Chen, and S.-M. Hu. Mpm simulation of interacting fluids and solids. In Computer Graphics Forum, volume 37, pages 183–193. Wiley Online Library, 2018. 3

[33] O. Zarifi and C. Batty. A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 1–11, 2017. 3

[34] Y. Zhu and R. Bridson. Animating sand as a fluid. ACM Transactions on Graphics (TOG), 24(3):965–972, 2005. 3, 5