A Review of The Affine Particle-In-Cell Method

Seminar: Current Topics in Physically-Based Animation

Pin-Hua Ho

June 23, 2025

Abstract

In computer graphics, hybrid grid/particle methods, especially the Particle-In-Cell (PIC) and Fluid-Implicit-Particle (FLIP), have been the standard solution for simulating free surface fluids. However, these methods force simulation engineers or content creators to trade between stability and the details of the simulation. Traditional PIC, while being stable, comes with severe numerical dissipation that results in loss of energy and angular momentum; On the other hand, FLIP was designed to solve the dissipation from PIC, introduce noise, and become unstable under certain conditions.

This paper aims to introduce and explain the Affine Particle-In-Cell method (APIC). APIC was developed to solve the aforementioned issues from PIC and FLIP. The core idea of APIC is to compensate for the information loss when performing particle and grid transfer by adding a locally affine description of the velocity for each particle. In this way, APIC not only inherited stability from PIC but also preserved energy and angular momentum during simulation, providing a vivid and detailed result.

Contents

1 Introduction

Fluid representation has been a core challenge in the domain of computer graphics and physics simulation. To simulate fluids, two paradigms have been deployed: Lagrangian and Eulerian methods. The Lagrangian method, which tracks individual material points, is excellent at handling advection and catching the details of free surface flows, but faces difficulties with maintaining incompressibility and with scalability of parallel computing. On the other hand, the Eulerian method, which performs calculations on a fixed grid, is suited for handling dynamics such as pressure but struggles at capturing high-frequency details and advection. To combine the advantages of two paradigms, hybrid methods have emerged and quickly become the standard solution in the film and gaming industries [ZB05][JSS+15].

The core idea of the hybrid approach is using Lagrangian particles to carry physical properties such as mass, position, and velocity, which are responsible for kinematics and advection; At the same time, a fixed Eulerian grid is used to solve for dynamics terms within the Navier-Stokes equations, such as pressure and viscosity.

The PIC method was the first hybrid method that could handle large deformation of flow and quickly became prominent. However, it introduces severe numerical dissipation during the data transfer between particles and the grid, leading to a loss of information and detail. The grid inherently acts as a lossy low-pass filter [BKR88][JSS+15]. To address the dissipation, the FLIP method was proposed by Brackbill et al. in 1988 [BKR88]. The main difference between FLIP to PIC is to compute an increment to the particle's velocity, rather than completely overwriting it with an interpolated velocity from grid cells in each time step as PIC does [BKR88][JSS+15].

The increment approach used by FLIP, however, exposes a problem inherent in hybrid methods that was hidden by PIC's filtering characteristic: the so-called "Ringing Instability". FLIP is particularly affected by this, as its incremental update preserves the particle's previous velocity, allowing numerical errors or unidentified modes to accumulate over time. This results in noise and simulation instability. To mitigate this instability, it is often necessary to use a linear blend of PIC and FLIP, which unfortunately re-introduces the dissipation from PIC and adds the extra burden of tuning the blend weights [ZB05][JSS+15].

To resolve this trade-off between stability and detail, Jiang et al. [JSS+15] proposed the Affine Particle-In-Cell (APIC) method, which will be reported in-depth in this paper.

The core idea of APIC is based on PIC's stable grid-filtering characteristic, but to augment each particle with a 3×3 locally affine matrix that describes the flow. The authors also designed new particle and grid transfer functions to utilize the extra information on the particle [JSS+15].

As a result, APIC minimized the dissipation from PIC, and while being stable, was able to preserve angular momentum and capture the high-frequency details on the surface. The APIC was designed to be a general particle and grid transfer scheme, that is, it's not limited to the use of fluid simulation, but also other hybrid particle/grid methods such as Material Point Method (MPM) [JSS+15].

2 Background

2.1 Simulation Paradigms: Eulerian and Lagrangian

2.1.1 Lagrangian Method

The Lagrangian method describes a fluid as a set of particles moving through space. In this paradigm, every particle carries its own physical properties such as mass, position, and velocity. The primary advantage of this method is the handling of advection. Since the flowing material is the particles themselves, advection is handled naturally, by kinematics, without numerical dissipation, and mass can be perfectly conserved. However, in general, the particle method is less accurate compared to the grid (Eulerian) method due to the difficulty of computing the spatial derivatives on an unstructured particle cloud; this applies to solving dynamics such as pressure in the Navier-Stokes equation [BS10].

2.1.2 Eulerian Method

In contrast to the Lagrangian paradigm, the Eulerian method uses a fixed spatial grid to represent the fluid. In this paradigm, physical properties are assigned to the grid cells, such as velocity, pressure, and density. The flow of the fluid is achieved by updating physical property values between the grid cells. The structured nature of this method provides a natural advantage in computing spatial derivatives, using finite differences, making it well-suited for tasks like the pressure solve. The main drawback of the Eulerian approach is handling advection. As physical properties are transferred from one grid cell to the other, they are inevitably interpolated, which introduces numerical errors, and thus maintaining energy and mass conservation is one of the challenges in grid-based methods [BS10].

2.1.3 Hybrid Method

The natural complementarity of the Lagrangian method and the Eulerian method makes them perfect to work together. The core strategy is to leverage Lagrangian particle kinematics to carry out dissipation-free and highly detailed advection, while simultaneously utilizing the efficiency of dynamic solving of Eulerian grids. To achieve this, the transfer

function is introduced, which is used to transfer physical properties between particles and the grid. A diagram of a simulation step is shown in Figure 2 [ZB05][JSS+15].

Ideally, for a simulation to be physically accurate, these transfer functions should conserve physical quantities such as mass, linear momentum, and angular momentum. However, achieving all these conservation properties simultaneously has been a challenge. As the next chapter will detail, traditional methods like PIC fail to conserve angular momentum, leading to dissipation. Therefore, designing transfer schemes that better preserve these conservation properties is one of the main focuses of the research, and it is exactly the problem that APIC was designed to solve [JSS+15].

2.2 Marker-and-Cell

An intuitive way to store physical properties on the grid cells may be to store at the center of the cell (collocated grid). However, this can lead to a decoupling of the pressure and velocity fields of neighboring cells, resulting in checkerboard instability. The mathematical root of this issue, as explained by Braley and Sandu [BS10], is that when calculating finite differences, a standard central difference scheme on a collocated grid fails to consider the value at the point of interest itself.

To resolve this instability, the Marker-and-Cell (MAC) scheme was introduced by Harlow and Welch [HW+65]. In a MAC grid, scalar quantities are stored at the center of the cell, while vector quantities are split into Cartesian components and stored at the center of the normal face of the corresponding Cartesian axis, see Figure 1. For instance, the x-component of velocity is stored on the vertical faces (facing $+x$) of a cell, and the y-component on the horizontal faces (facing $+y$) [BMF07].

The key advantage of this staggered arrangement is how it simplifies the finite difference calculation. To compute the gradient of horizontal velocity with interest of cell *(i, j)*, for example, we can now directly use the velocity values stored on its faces: the one on the right face, $u_{i+1/2,j}$, and the one on the left face, $u_{i-1/2,j}$, which in practice, is stored by the left neighbor cell *(i-1, j)*.

This prevents the checkerboard instability that happens on collocated grids. This stability also allows the incompressibility constraint to be enforced accurately; the MAC grid has become the standard method for simulating incompressible fluids in computer graphics [BMF07].

3 Notation

In accordance with the APIC paper [JSS+15], we use subscripts p and q to denote particle indices, and subscripts i and j refer to grid cell indices. The superscript n (v_p^n) indicates a variable at the beginning of the current time step, while n+1 indicates a variable at the beginning of the next time step (v_p^{n+1}). Lowercase bold letters ($\mathbf{v}$) represent vector quantities, and uppercase bold letters ($\mathbf{B}$) represent matrix quantities.

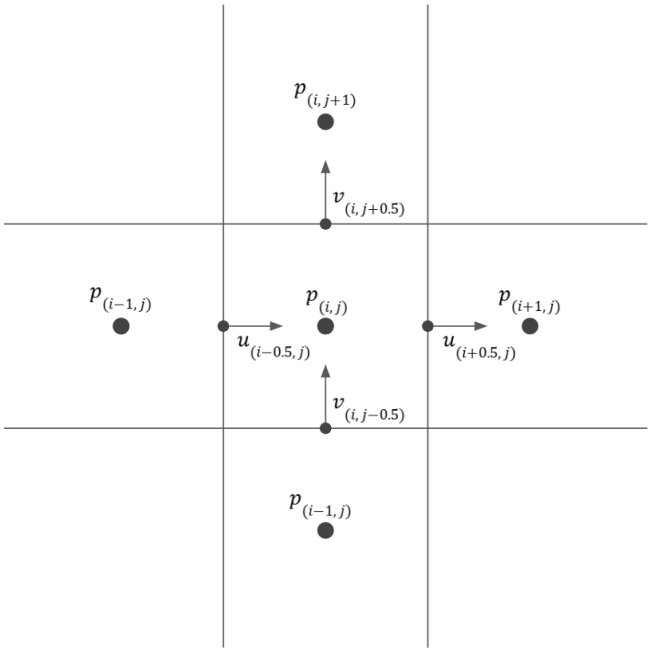

Figure 1: The two-dimensional MAC grid. The image is based on Fig. 2.1 of [BMF07]

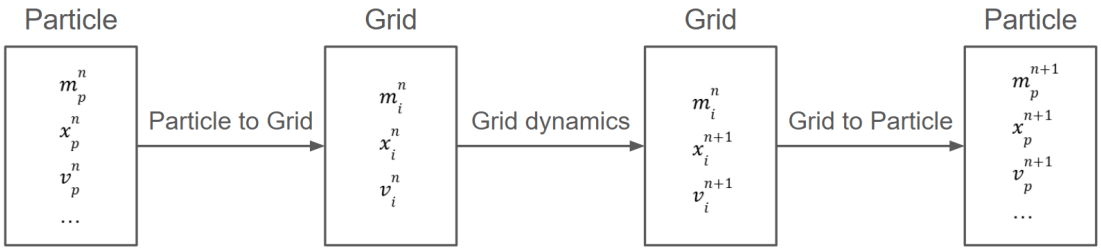

Figure 2: A diagram of a simulation step in a hybrid particle/grid setting. The image is based on the tech report of APIC [JSS+15]

4 Related Work

4.1 The PIC Method

The PIC method was developed by Harlow in 1964 and was an early method for simulating compressible fluids that handles advection with particles and solves dynamics on a grid [ZB05]. At each time step, particles update their kinematics by simply moving their positions by their velocities. Then a weighted average of the particles' variables is rasterized onto the grid cells by the transfer function:

$$\mathbf{m}_i^n \mathbf{v}_i^n = \sum_p w_{ip} m_p^n \mathbf{v}_p^n.$$

After performing grid dynamics, that is the non-advection part of the equation, the particle values are then **overridden** by the interpolated values from the grid cells:

$$\mathbf{v}_p^{n+1} \leftarrow \sum_i w_{ip} \mathbf{v}_i^{n+1}.$$

The main advantage of PIC is that it is stable, as the grid inherently serves as a filter. The mechanism of overriding particle values from grid interpolation at each time step also supported this stability [ZB05].

The major problem with PIC was that it was highly dissipative. In the APIC paper, the authors stated that the root cause is the mismatch of degrees of freedom between the particles and the grid. Since usually the number of particles is way larger than the number of grid cells, some sub-grid scale modes of particles couldn't be seen by the grid, getting no physical response, see Figure 3 . This could result in so-called "ringing instability". This wasn't a severe problem compared to the dissipation in PIC, since every time step the particle values get overridden by interpolated values from the grid, effectively resetting the error and noise. However, it gets amplified over time in FLIP, which we will talk about in the next subsection [JSS+15].

4.2 The FLIP Method

In order to solve the dissipation problem in PIC, Brackbill et al. [BKR88] introduced FLIP. The core change is that instead of completely overriding the particle value in G2P step, in FLIP, the grid computes the value differences and then adds to the old values:

$$\mathbf{v}_p^{n+1} \leftarrow \mathbf{v}_p^n + \sum_i w_{ip} \mathbf{\Delta v}_i^{n+1}.$$

The solution was effective, FLIP massively minimized dissipation and improved angular momentum conservation from PIC, if not perfectly conserved, since the condition for FLIP to achieve exact conservation of angular momentum is the use of a non-diagonal (consistent) mass matrix, which in practice is not possible [JSS+15].

However, the scheme of computing value changes and reusing the old values from particles reveals another problem that lies in the hybrid method, which was previously hidden by PIC's dissipation.

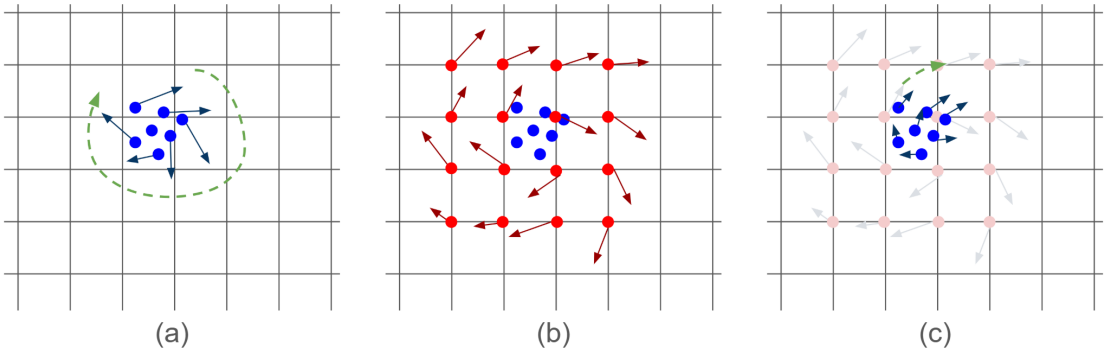

Figure 3: An illustration of mode mismatch and dissipation in the traditional PIC method. (a) Initially, a group of particles (blue dots) carrying sub-grid scale vortex motion, with angular momentum denoted by the green arrow. (b) The Particle-to-Grid (P2G) transfer step. The particle velocities are rasterized onto the surrounding grid cells (red dots). (c) The Grid-to-Particle (G2P) transfer step. The velocities of the particles are completely overridden by new values interpolated from the grid's velocity field. As a result, the original, high-frequency rotational mode has been filtered out, resulting in a loss of angular momentum.

As mentioned previously, the mismatch of degrees of freedom between the particles and the grid created sub-grid scale modes that the grid couldn't resolve. In PIC, these unresolved modes are filtered out at every step. In FLIP, however, the direct data path from the previous particle state preserves the error. Since these modes receive no physical response from the grid, their numerical errors are allowed to persist and accumulate over time [JSS+15].

This accumulation of error leads to the "ringing Instability," which is far more pronounced in FLIP than in PIC. This is why FLIP is often noisy, and has unnatural particle clumps on the surface, and in some cases, might break the simulation. Therefore, while FLIP successfully solves the problem of dissipation, it does so at the cost of stability.

To mitigate FLIP's instability in practice, Zhu and Bridson [ZB05] proposed using a linear blending of FLIP and PIC. Although this prevents the severe instability issue that FLIP might get, and adds high-frequency details that PIC doesn't have, it reintroduces unwanted dissipation. Moreover, the blending weight varies among use cases, adding another complexity for tuning the simulation [JSS+15].

5 Method

The APIC is based on PIC. The only difference between PIC, FLIP, and APIC is the transfer function between the particles and the grid. The authors aimed to show that APIC can minimize dissipation without relying on the unsafe direct data path in FLIP, maintaining the stability in PIC.

The key observation is that, when performing G2P transfers, the influences of multiple grid cells are reduced to a single value on the particle, resulting in information loss. The authors progressively introduced two strategies to increase the amount of information on the particles: the Rigid Particle-In-Cell (RPIC) method and finally, the Affine Particle-In-Cell (APIC) method [JSS+15].

5.1 Rigid Particle-In-Cell (RPIC)

As angular momentum loss was the primary observation from PIC, an intuitive approach is to compensate for that rotation by storing a local angular momentum $\mathbf{L}_p^n$ on the particle.

In the PIC framework, a single particle's influence across several nearby grid cells with weighting functions inherently creates a "soft boundary." The RPIC method combines this concept with the particle's stored local angular momentum, effectively treating the particle as a rotating rigid body with volume. In order to preserve the angular momentum throughout the particle/grid transfers, the transfer functions are modified to take this rotational motion into consideration [JSS+15].

In G2P transfer, a term to update the local angular momentum on the particle is added:

$$\mathbf{L}_p^{n+1} = \sum_i w_{ip}^n (\mathbf{x}_i - \mathbf{x}_p^n) \times m_p \mathbf{v}_i^{n+1}.$$

The term $(\mathbf{x}_i - \mathbf{x}_p^n)$ represents the lever arm from the particle's center to a nearby grid cell i. The term $(m_p \mathbf{v}_i^{n+1})$ represents the linear momentum on that grid cell. Their cross product thus gives the angular momentum of this "piece" of the particle rigid body's motion relative to the center of mass.

The full equation then calculates the weighted sum of angular momentum over all influencing grid cells, where w_{ip}^n is the interpolation weight. This can be interpreted as a discrete integration of the angular momentum of the particle rigid body.

Next, the authors designed the P2G transfer function by considering the rotating velocity when calculating linear momentum on the grid cell:

$$m_i^n v_i^n = \sum_p w_{ip}^n m_p (\mathbf{v}_p^n + ((\mathbf{K}_p^n)^{-1} \mathbf{L}_p^n) \times (\mathbf{x}_i - \mathbf{x}_p^n).$$

Here, the term $\mathbf{K}_p^n$ is the inertial tensor of the particle rigid body, and thus $(\mathbf{K}_p^{n-1})\mathbf{L}_p^n$ represents the angular velocity of the particle. The cross product gives the tangent velocity at grid location $\mathbf{x}_i$, contributed by the particle's rotation. The full equation then calculates the weighted sum of linear momentum of influencing particles, taking both particles' linear motions and rotational motions into consideration.

The authors stated that these new P2G and G2P transfer functions for RPIC conserve the total angular momentum of the system. The core concept of the proof, as detailed in their supplementary document, is to prove conservation of linear momentum and conservation of angular momentum, respectively, both in P2G and G2P transfers. By setting $\Delta t = 0$, and derive the equality of momentum in transfer functions, thus proving

the conservation of linear momentum and angular momentum in the whole simulation step.

While the RPIC successfully conserves angular momentum and corrects the rotational damping of PIC, it only solves one aspect of the information loss problem; thus, the dissipation persists. As the authors of APIC noted, fluid motion involves more than just rigid body rotation; it also includes non-rigid deformations such as shearing. Therefore, while being a major improvement from PIC, RPIC remains an incomplete solution, which motivates the need for a more general model that can represent the full affine modes of the fluid [JSS+15].

5.2 Affine Particle-In-Cell (APIC)

The APIC method took a more fundamental approach. Instead of explicitly compensating for the momentum loss like RPIC, APIC aims to prevent information loss by enriching each particle with a 3×3 affine matrix to capture the full, local affine motion of the fluid. By doing so, the method can implicitly preserve the sub-grid modes that were previously lost in the traditional PIC transfer.

The authors introduced the 3×3 affine matrix $\mathbf{C}_p^n$ to replace the local angular momentum $\mathbf{L}_p^n$ (3×1 vector) in RPIC. Then the local velocity at grid cell $\mathbf{x}_i$ with the influence of particle $\mathbf{x}_p^n$ is $\mathbf{v}_p^n + \mathbf{C}_p^n(\mathbf{x}_i - \mathbf{x}_p^n)$.

However, to uniquely update all 9 components in matrix $\mathbf{C}_p^n$ from $\mathbf{L}_p^n$, which only has 3 components, is not possible [JSS+15]. The authors thus proposed dividing the matrix $\mathbf{C}_p^n = \mathbf{B}_p^n(\mathbf{D}_p^n)^{-1}$.

Then the P2G transfer function is analogous to the RPIC case:

$$m_i^n \mathbf{v}_i^n = \sum_p w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{B}_p^n (\mathbf{D}_p^n)^{-1} (\mathbf{x}_i - \mathbf{x}_p^n)).$$

The matrix $\mathbf{D}_p^n$ is an inertia-like tensor that only depends on the particle's position relative to the grid, describing the spatial distribution of the particle's influence. It could be calculated at the beginning of each time step, thus no need to update during the G2P transfer:

$$\mathbf{D}_p^n = \sum_i w_{ip}^n (\mathbf{x}_i - \mathbf{x}_p^n)(\mathbf{x}_i - \mathbf{x}_p^n)^T.$$

For the matrix $\mathbf{B}_p^n$, which would be updated during G2P transfer, the update function is designed as:

$$\mathbf{B}_p^{n+1} = \sum_i w_{ip}^n \mathbf{v}_i^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T.$$

The matrix $\mathbf{B}_p^n$ can be seen as an intermediate quantity that combines both local velocity and inertia tensor ($\mathbf{B}_p^n = \mathbf{C}_p^n \mathbf{D}_p^n$), making it a **momentum-like** matrix. Thus, the matrix $\mathbf{B}_p^n$ can be seen as a richer representation of the simple angular momentum vector $\mathbf{L}_p^n$ in RPIC. Its skew-symmetric part contains the angular momentum information, while its symmetric part contains information about shearing and stretching, therefore capturing the full local affine motion of the fluid [JSS+15].

This new transfer scheme, which preserves the full affine motion of the particles, is the core contribution of the APIC method. The fundamental concept is general and powerful. While the authors also expanded an adaptation of this scheme for staggered MAC grids in fluid simulation, the core mechanisms of using an affine state to prevent information loss are the same and can be applied to other simulation domains, such as the Material Point Method (MPM) [JSS+15].

The authors also formally prove their conservation properties in the supplementary document. Similar to the RPIC proofs, they show that both linear and angular momentum are conserved during the P2G and G2P transfers.

6 Results and Discussion

Free-surface flow, which describes the simulation of a liquid interacting with another medium like air, is the most common application for hybrid methods in computer graphics. The main characteristics are the large deformations and complex surface topologies of the fluid. The authors of APIC provide several comparisons to demonstrate APIC's advantages in this application.

The authors picked pure PIC, FLIP, and the linear blend of PIC/FLIP to compare. APIC visually improves among the traditional methods, see Figure 4. At the beginning of the simulation (the top row), the dissipation of PIC is already apparent, as the natural damping of PIC causes the water cap to be slower than it is in the other methods. While FLIP maintains energy, it appears uneven and noisy at the edge of the water cap. As the simulation proceeds (the bottom row), the water cap from PIC shows a significantly smaller volume due to dissipation, while the water surface in FLIP becomes noisy, with unnatural lumps. The linear blend of PIC/FLIP mitigates but also mixes the dissipation and the noise together. In contrast, APIC produces a result that is both energetic and maintains a smooth, stable surface [JSS+15].

The authors stated that, theoretically, APIC does not improve nor does it worsen the ringing instability, which is inherent in the PIC method. The main difference is how each method responds to this instability. In FLIP, because the particle's old state is preserved by incremental updates, the errors from unresolved sub-grid modes (the source of the ringing instability) are amplified over time. In APIC, however, the method keeps PIC's grid filtering characteristic. This filtering, while causing information loss in standard PIC, effectively resets these errors at each time step, contributing to its stability [JSS+15].

The authors also analyzed the energy conservation capabilities. They found that while APIC performs best in the rotation test, FLIP is often better at preserving total kinetic energy in many cases. This suggests that if FLIP's instability could be resolved by other means, its energy preservation might be an advantage over APIC [JSS+15].

The authors noted that APIC introduces a minor overhead due to the extra storage for the 3×3 matrix and additional computations in the transfer functions. However, practical tests showed that during these two modified steps, which are the P2G rasterization and the G2P update, the cost is not much higher than other methods. Furthermore, these steps are typically dominated by other, more time-consuming stages, where the

Figure 4: A comparison of a fountain simulation. APIC (left) produces a result that is both energetic and maintains a smooth surface, compared to the dissipative PIC and noisy FLIP results. (Image from Jiang et al. [JSS+15])

velocities are the primary factor influencing the time step. Thus, generally, PIC is the fastest due to its damping of motions, while FLIP tends to be the slowest due to instability and therefore larger velocities [JSS+15].

Finally, the authors also pointed out an interesting practical consideration: the noise generated by the linear blend of PIC/FLIP might sometimes be preferred by artists as it can be seen as a more "vivid" or detailed simulation. However, the authors argued that if such artifacts are desired, they should be introduced in a controllable manner, rather than being an uncontrollable side effect of the underlying simulation framework [JSS+15].

The subsequent work following APIC has not focused on solving its remaining issues, such as the ringing instability. As the authors suggest, APIC already resolves most of the artifacts of the pure PIC method and provides a controllable and stable solution compared to FLIP. Instead, research has favored building upon APIC. For example, the Moving Least Squares Material Point Method (MLS-MPM) paper, co-authored by Jiang in 2018 [HFG+18], uses the APIC method as a basis to expand the framework's applications. It enables the simulation of discontinuities, which allows for realistic material cutting and interaction with thin boundaries.

References

[BKR88] J.U. Brackbill, D.B. Kothe, and H.M. Ruppel. Flip: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1):25–38, 1988.

[BMF07] Robert Bridson and Matthias Müller-Fischer. Fluid simulation: Siggraph 2007 course notesvideo files associated with this course are available from the citation page. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, page 1–81, New York, NY, USA, 2007. Association for Computing Machinery.

[BS10] Colin Braley and Adrian Sandu. Fluid simulation for computer graphics: A tutorial in grid based and particle based methods. *Virginia Tech, Blacksburg*, 3, 2010.

[HFG+18] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.*, 37(4), July 2018.

[HW+65] Francis H Harlow, J Eddie Welch, et al. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids*, 8(12):2182, 1965.

[JSS+15] Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. The affine particle-in-cell method. *ACM Trans. Graph.*, 34(4), July 2015.

[ZB05] Yongning Zhu and Robert Bridson. Animating sand as a fluid. *ACM Trans. Graph.*, 24(3):965–972, July 2005.