Solid-Fluid Interaction on Particle Flow Maps

DUOWEN CHEN, Georgia Institute of Technology, USA
ZHIQI LI, Georgia Institute of Technology, USA
JUNWEI ZHOU, Purdue University, USA and University of Michigan, USA
FAN FENG, Dartmouth College, USA
TAO DU, Tsinghua University, China and Shanghai Qi Zhi Institute, China
BO ZHU, Georgia Institute of Technology, USA

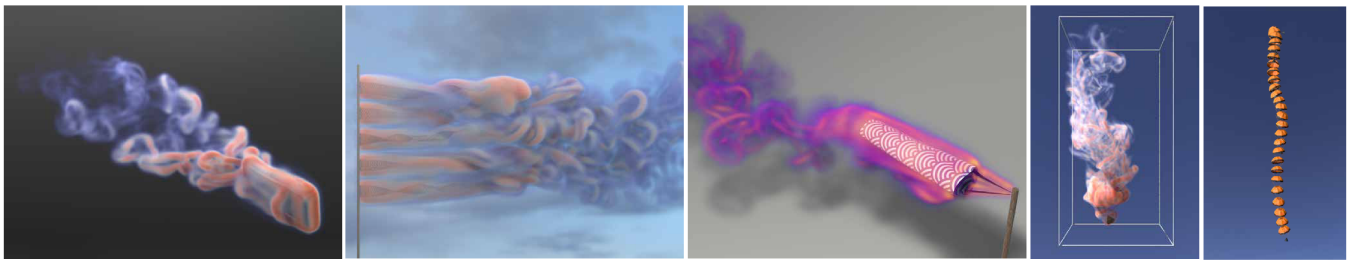

Fig. 1. We demonstrate our method's efficacy with various examples of fluid-solid interaction, including a swimming fish, long silk flags, a Koinobori, and a falling parachute along with its trajectory (from left to right), all exhibiting strong vortex dynamics and solid-vortex interactions.

We propose a novel solid-fluid interaction method for coupling elastic solids with impulse flow maps. Our key idea is to unify the representation of fluid and solid components as particle flow maps with different lengths and dynamics. The solid-fluid coupling is enabled by implementing two novel mechanisms: first, we developed an impulse-to-velocity transfer mechanism to unify the exchanged physical quantities; second, we devised a particle path integral mechanism to accumulate coupling forces along each flow-map trajectory. Our framework integrates these two mechanisms into an Eulerian-Lagrangian impulse fluid simulator to accommodate traditional coupling models, exemplified by the Material Point Method (MPM) and Immersed Boundary Method (IBM), within a particle flow map framework. We demonstrate our method's efficacy by simulating solid-fluid interactions exhibiting strong vortical dynamics, including various vortex shedding and interaction examples across swimming, falling, breezing, and combustion.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Solid-Fluid Interaction, Particle Flow Map, Impulse Gauge Fluid, Vortex Dynamics, Path Integral

ACM Reference Format:
Duowen Chen, Zhiqi Li, Junwei Zhou, Fan Feng, Tao Du, and Bo Zhu. 2024. Solid-Fluid Interaction on Particle Flow Maps. *ACM Trans. Graph.* 43, 6, Article 267 (December 2024), 20 pages. https://doi.org/10.1145/3687959

Authors' Contact Information: Duowen Chen, dchen322@gatech.edu, Georgia Institute of Technology, USA; Zhiqi Li, zli3167@gatech.edu, Georgia Institute of Technology, USA; Junwei Zhou, zjw330501@gmail.com, Purdue University, USA and University of Michigan, USA; Fan Feng, Fan.Feng.GR@dartmouth.edu, Dartmouth College, USA; Tao Du, taodu.eecs@gmail.com, Tsinghua University, China and Shanghai Qi Zhi Institute, China; Bo Zhu, bo.zhu@gatech.edu, Georgia Institute of Technology, USA.

ACM 1557-7368/2024/12-ART267
https://doi.org/10.1145/3687959

1 Introduction

In recent years, flow map methods have attracted increasing attention in computer graphics [Nabizadeh et al. 2022; Qu et al. 2019] and computational physics [Rycroft et al. 2020]. Unlike traditional approaches that advect physical quantities with flow velocity at each timestep, flow maps establish a (typically bidirectional) point-to-point correspondence between the initial frame and the current frame to transport quantities, achieving impressive long-range advection accuracy through the use of additional data structures such as neural buffers [Deng et al. 2023] or particles [Li et al. 2024; Zhou et al. 2024]. One of the most significant advantages of using a flow map is its ability to preserve vortical structures during their creation and evolution, which makes flow maps particularly well-suited for modeling solid-fluid interactions, especially those dominated by vorticity produced around moving boundaries of solids immersed in turbulent fluid environments. Such vortex-solid phenomena are ubiquitous and can be observed in examples like fish swimming, birds gliding, parachutes descending, and wind blowing through thin cloth and hair.

However, despite the burgeoning literature on using flow map methods to solve fluid flow problems, the study of solid-fluid interactions on flow maps remains sparse. In computer graphics, there has been no previous study on this problem. In computational physics, several prior works (e.g., see [Rycroft et al. 2020; Wang et al. 2022]) have developed frameworks to solve solid-fluid interactions based on reference maps within an Eulerian setting, focusing on devising an Eulerian solid framework to fit into the fluid flow map. The primary coupling mechanism in these approaches predominantly relies on creating a narrow band around the solid boundary and mixing stresses between solid and fluid using a Heaviside blending

function. This method is highly parameter-sensitive and could produce non-physical results if the narrow band size and the blending function are improperly chosen.

Why do the various classical solid-fluid coupling strategies (e.g., MPM [Jiang et al. 2016], IBM [Peskin 2002], variational [Batty et al. 2007], monolithic [Robinson-Mosher et al. 2008], etc.) not fit within the flow-map framework? We speculate three potential reasons: (1) *Flow-map coupling requires a unified representation of both solid and fluid.* In other words, the solid representation and discretization must be identical to their fluid counterparts, such as a flow map with the same initial and final time stamps defined on the same Eulerian grid, which significantly limits the scope of solid models that can be chosen to accommodate complex solid-fluid interactions. In particular, the significance of devising a long-range flow map model for solid simulation remains unclear due to the less connected nature between flow advection and solid dynamics. (2) *Flow-map coupling requires exchangeable physical quantities between solid and fluid.* Though this was not a problem in conventional solid-fluid interaction frameworks (e.g., exchanging velocity or momentum via G2P or P2G operations in traditional MPM), modern flow map methods typically evolve gauge variables (e.g., impulse [Cortez 1996], vorticity [Cottet et al. 2000], and other gauges [Saye 2016, 2017]) instead of fluid velocities, which cannot be directly operated with solid velocities. For instance, we cannot naively conduct a P2G operation across the solid-fluid interface with fluid particles carrying impulses and solid particles carrying velocities. (3) *Adding an external force to a flow-map model remains an open problem.* Although simple forces such as gravity can be incorporated into existing flow-map models, their physical accuracy is less grounded. Local forces, such as momentum exchange, remain unclear regarding how they should be transferred from solid to fluid in a flow-map system.

We propose a novel solid-fluid interaction framework based on flow map models by addressing the abovementioned challenges. Our key idea is to model both solid and fluid as a unified forward flow map on particles: each fluid particle represents a long flow map governed by impulse fluid dynamics, while each solid particle represents a short flow map governed by elastic solid dynamics. Specifically, we restrict the solid flow map to a single time step to adapt an arbitrary conventional solid simulation model (e.g., MPM or XPBD). The fluid and solid flow maps are coupled based on two key mechanisms: (1) we implement an impulse-to-velocity transfer mechanism to unify the physical quantities exchanged between solid and fluid particles; (2) we implement a particle path integral mechanism to accurately accumulate both pressure and coupling forces along each flow-map trajectory. The combination of these two mechanisms, in conjunction with the standard particle-grid operations and incompressibility projections, synergistically enables a versatile coupling framework to exchange information between particle flow maps with different lengths and governing physics, which further accommodates the adaptation of various traditional coupling models into flow map methods. In our implementation, we demonstrate two examples of MPM and IBM coupling by integrating both into a hybrid Eulerian-Lagrangian fluid simulator on particle flow maps. Thanks to the inherent advantage of preserving vortical structures in our flow map model, these flow-map-enhanced coupling systems produce vortex-solid interaction simulations that outperform traditional methods in terms of both physical accuracy and visual complexity. In our experiments, we implemented a diverse set of benchmark tests and simulation examples, ranging from leaves falling and fish swimming to complex vortex shedding behind cloth, hair, and combustion processes, demonstrating our framework's versatility and efficacy in tackling complex vortex-object interaction simulations.

We summarize our main contributions as follows:

- A unified particle flow-map representation with different lengths and governing equations for fluid and solid;
- A reformulated impulse gauge fluid model to enable solid-fluid momentum exchange on particles;
- A path integral approach on particle flow maps to accumulate coupling forces;
- A versatile framework to accommodate traditional solid-fluid coupling mechanisms on flow-map models.

2 Related Work

2.1 Flow Map & Impulse Fluid

Initially known as the method of characteristic mapping (MCM), the concept of flow maps was first introduced by Wiggert and Wylie [1976]. By reducing the diffusion error caused by semi-Lagrangian advection, various attempts in the graphics community were made to adapt this method in fluid simulation [Hachisuka 2005; Qu et al. 2019; Sato et al. 2018, 2017; Tessendorf 2015]. In Covector Fluid (CF) [Nabizadeh et al. 2022], flow maps were first introduced to aid the advection of the covector variable, achieving state-of-the-art vorticity preservation effects. The impulse variable, a form of covector, was first introduced by Buttke [1992]. By rewriting the incompressible Navier-Stokes Equations through the use of a gauge variable and gauge transformation [Buttke 1992, 1993; Oseledets 1989; Roberts 1972], it allows for the gauge freedom to be designed for specific applications [Buttke 1993; Buttke and Chorin 1993; Cortez 1996; Saye 2016, 2017; Summers 2000; Weinan and Liu 2003]. This concept was revisited in computer graphics by Feng et al. [2022] and Yang et al. [2021]. Neural Flow Map (NFM) [Deng et al. 2023] further enhances the flow-map accuracy with a neural buffer. Recently, Particle Flow Map (PFM) [Zhou et al. 2024] and Impulse PIC (IPIC) [Sancho et al. 2024] used a hybrid method with particles, and [Li et al. 2024] extended the covector to pure Lagrangian representations.

In [Cortez 1996; Saye 2016, 2017; Summers 2000], attempts to couple impulse with solids were made. However, these methods were limited by the need to redesign the gauge variable for different solids [Saye 2016, 2017] and could not be adapted to the advection scheme using flow maps [Cortez 1996; Summers 2000]. Our method aims to use PFM to design a general solid-fluid coupling scheme for impulse fluids within the flow map framework.

2.2 Full Eulerian Coupling

For full Eulerian methods, computation time benefits arise from both solid and fluid being treated on a single fixed background grid. Such methods include the deformation gradient-based method [Liu and Walkington 2001] and initial point set (IPS) [Dunne 2006]. Recently, the reference map technique (RMT) [Kamrin and Nave 2009; Kamrin et al. 2012; Rycroft et al. 2020] has attracted wide

attention and was later extended to couple rigid bodies with fluid [Wang et al. 2022]. As for pure Eulerian treatment in graphics, Teng et al. [2016] allows for larger time steps in pure Eulerian solid-fluid coupling by setting up a semi-implicit coupling system. Concerning flow maps, by noting the correspondence between the flow map and the deformation gradient used in the elastic solid simulation, RMT tries to incorporate them together on Eulerian mesh and bridge the two with a Heaviside function but is limited by the requirement of a narrowband blending scheme.

2.3 Full Lagrangian Coupling

Full Lagrangian methods use Lagrangian elements in both the fluid and solid domains. Representatives of this method in computational physics include particle FEM methods [Becker et al. 2015; Cremonesi et al. 2020; Idelsohn et al. 2008] and, later, the combination of MPM and FEM [Lian et al. 2011a, 2012, 2011b, 2014]. As for research in graphics, it was initially explored in [Keiser et al. 2005; Müller et al. 2004]. Later, Klingner et al. [2006] proposed using a body-confronting mesh for coupling. Subsequent works proposed a unified framework representing both solid and fluid with Lagrangian elements [Clausen et al. 2013]. Coupling SPH with deformable has also been explored [Akinci et al. 2013; Solenthaler et al. 2007] but is limited to relatively simple simulation settings. In [Akbay et al. 2018], authors proposed an extended partition method (XPM) and demonstrate this using Lagrangian solid coupling with an Eulerian fluid solver on the grid and a Lagrangian fluid solver like SPH.

2.4 Mixed Lagrangian-Eulerian Coupling

In mixed Lagrangian-Eulerian mesh methods, solids are represented by Lagrangian markers coupled with fixed Eulerian background meshes. Representatives of such methods include the immersed boundary method (IBM) [Huang and Sung 2009; Mori and Peskin 2008; Peskin 1972, 2002]. Based on IBM, the immersed finite element method [Liu et al. 2007, 2006; Shimada et al. 2022], immersed interface method [Zhao et al. 2008], and immersed continuum method [Wang 2006, 2007] were proposed. Other improvements on IBM methods have been made, such as monolithic projection methods [Wang et al. 2020] and combining marker particles with finite volume expression [Shimada et al. 2022]. See [Huang and Tian 2019] for more details. In the graphics community, since the pioneering work of Carlson et al. [2004], Génevaux et al. [2003], and Guendelman et al. [2005], various works in this direction have been explored. Batty et al. [2007] and Ng et al. [2009] treated coupling as an energy minimization form, Robinson-Mosher et al. [2008], Robinson-Mosher et al. [2009] and Robinson-Mosher et al. [2011] perform implicit coupling and eables the free-slip boundary conditions. Zarifi and Batty [2017], Takahashi and Batty [2020] and Takahashi and Batty [2022] use the cut-cell technique combined with a properly constrained global system for further improvement. As for hybrid methods like MPM [Stomakhin et al. 2013], it comes naturally with non-slip boundary conditions for coupling, and further research extends it to enable rigid body coupling [Hu et al. 2018] and the free-slip boundary conditions [Fang et al. 2020]. Exploration of hybrid methods in incorporating flow maps was present in [Shimada et al. 2021]. Authors improved RMT to a hybrid method with marker particles to better track the solid interface. However, the main problem of such a method is that fluid does not directly benefit from the advection using a flow map and, therefore, cannot achieve simulation quality as shown in impulse-based fluid methods.

Table 1. Summary of important notations used in the paper.

Notation	Type	Definition
$*^f$	vector/matrix	fluid property
$*^s$	vector/matrix	solid property
$*^n$	vector/matrix	fluid properties near solid (only used in MPM coupling)
$*_t$	scalar/vector/matrix	quantities evaluated at time t
$\boldsymbol{x}$	vector	particle / mesh vertices location
$\mathcal{F}_{[c,a]}$	matrix	Forward Jacobian from c to a
$\mathcal{T}_{[a,c]}$	matrix	Backward Jacobian from a to c
$\mathbb{C}$	function	constraint in XPBD simulation
$\boldsymbol{u}$	vector	velocity
$\nabla \boldsymbol{u}$	matrix	velocity gradient
$\boldsymbol{m}$	vector	impulse
$\boldsymbol{f}$	vector	force
p	scalar	pressure
Λ	vector	pressure correction buffer
Υ	vector	external force buffer
n	scalar	reinitialization steps

2.5 Coupling with Thin Structures

Using the coupling techniques mentioned above, different scales of coupling phenomena have also been studied. In particular, coupling with thin structures is of interest and examples include coupling hair-fluid coupling [Fei et al. 2017], fabric-fluid coupling [Fei et al. 2018], coupling fabric with non-Newtonian fluid [Fei et al. 2019], coupling parachute/cloth with fluid [Wang et al. 2020], insect flying and fish swimming [Borazjani and Sotiropoulos 2010; Cui et al. 2018; Tian et al. 2014], coupling uniform flow with flags [Uddin et al. 2013; Wang and Tian 2019], coupling free-surface water with thin shells [Robinson-Mosher et al. 2008] and also combustion between fire and paper or cloth [Losasso et al. 2006]

3 Physical Model

Naming Convention. We will adhere to the naming conventions in Table 1. Specifically, we will use superscripts for the type a quantity belongs to. For example, $*^f$ denotes fluid-related quantities, and $*^s$ denotes solids-related ones. We will use subscripts to indicate the evaluation time of a quantity, such as $*_t$ for values evaluated at time t. Similarly, $*_{[a,c]}$ represents a time interval from a to c and is used in flow map notations to indicate the duration over which the mapping occurs, with a and c as starting and ending time.

3.1 Governing Equations

We lay out the physical model of the solid-fluid coupling system on impulse variable:

$$\begin{cases} \dfrac{D\boldsymbol{m}}{Dt} = -\left(\boldsymbol{\nabla u}\right)^T \boldsymbol{m}, \\ \dfrac{1}{\rho^f}\nabla^2\varphi = -\boldsymbol{\nabla}\cdot\boldsymbol{m}, \\ \boldsymbol{u} = \boldsymbol{m} - \dfrac{1}{\rho^f}\boldsymbol{\nabla}\varphi, \\ \rho^s\dfrac{D\boldsymbol{v}}{Dt} = \nabla\cdot\sigma, \\ \boldsymbol{u} = \boldsymbol{v}, \quad \mathbf{x}\in\partial\Omega^{s,f}. \end{cases} \tag{1}$$

where $\boldsymbol{m}$ and $\boldsymbol{u}$ being fluid impulse and fluid velocity, and φ an intermediate variable used only for projecting $\boldsymbol{m}$ to the divergence-free $\boldsymbol{u}$. Here, $\boldsymbol{v}$ represents solid velocity, and σ is the elastic stress tensor. Here we use $\partial\Omega^{s,f}$ to represent the interface between fluid and solid. The first three equations describe the fluid momentum (first row) and incompressibility (second and third rows), the fourth equation describes the solid, and the fifth equation features the non-slip boundary conditions on the solid-fluid interface.

3.2 Flow Map

Flow map defines a bidirectional mapping ϕ and ψ between material space $\boldsymbol{X}$ and world space $\boldsymbol{x}$. The quantity defining infinitesimal change in each space resulting in changes in another can also be described using their Jacobians $\mathcal{F}$ and $\mathcal{T}$.

Specifically, we define the forward flow map $\phi(\cdot,t)$ as a function of space and time, mapping the initial position of a particle at time 0 to its position at a subsequent time t, and backward flow map $\psi(\cdot,t)$, as the mapping from time t back to time 0. We can define the forward flow map as:

$$\begin{cases} \dfrac{\partial\phi\left(\boldsymbol{X},\tau\right)}{\partial\tau} = \boldsymbol{u}\left(\phi\left(\boldsymbol{X},\tau\right),\tau\right), \\ \phi\left(\boldsymbol{X},0\right) = \boldsymbol{X}, \\ \phi\left(\boldsymbol{X},t\right) = \boldsymbol{x}, \end{cases} \tag{2}$$

Similarly, we can define the backward flow map as:

$$\begin{cases} \dfrac{\partial\psi\left(\boldsymbol{x},\tau\right)}{\partial\tau} = \boldsymbol{u}\left(\psi\left(\boldsymbol{x},\tau\right),\tau\right), \\ \psi\left(\boldsymbol{x},t\right) = \boldsymbol{x}, \\ \psi\left(\boldsymbol{x},0\right) = \boldsymbol{X}, \end{cases} \tag{3}$$

Subsequently, $\mathcal{F}$ and $\mathcal{T}$, the Jacobians of ϕ and ψ, can be calulated by $\mathcal{F} := \frac{\partial\phi}{\partial\boldsymbol{X}}$, $\mathcal{T} := \frac{\partial\psi}{\partial\boldsymbol{x}}$ and satisfies the following advection equations with a velocity field $\boldsymbol{u}$:

$$\begin{cases} \dfrac{D\mathcal{F}}{Dt} = \boldsymbol{\nabla u}\mathcal{F}, \\ \dfrac{D\mathcal{T}}{Dt} = -\mathcal{T}\boldsymbol{\nabla u}. \end{cases} \tag{4}$$

Here, $\frac{D(\cdot)}{Dt}$ represents the material derivative, which describes the rate of change of the Jacobians moving with the particle along its flow map's trajectory. As shown in previous works [Cortez 1996; Deng et al. 2023; Nabizadeh et al. 2022], fluid impulse can be transported via a bidirectional flow map with the defined Jacobians as:

$$\begin{cases} \boldsymbol{m}(x,t) &= \mathcal{T}^T\,\boldsymbol{m}(\psi(\boldsymbol{x}),0), \\ \boldsymbol{m}(X,0) &= \mathcal{F}^T\,\boldsymbol{m}(\phi(\boldsymbol{X}),t). \end{cases} \tag{5}$$

Particle Flow Map. By observing that a particle trajectory can naturally characterize a bidirectional flow map, a particle flow map method was proposed in [Zhou et al. 2024] by combining affine Particle-in-Cell [Jiang et al. 2015], covector/impulse fluid [Nabizadeh et al. 2022], and bidirectional flow map [Deng et al. 2023] as depicted in Figure 2. Under a particle perspective, fluid quantities such as $\boldsymbol{m}$, $\mathcal{T}$, and $\mathcal{F}$ are carried on Lagrangian particles. Its trajectory, originating from time a and culminating at time c, characterizes both the forward map $\phi_{[c,a]}$ and the backward map $\psi_{[a,c]}$. The Jacobians $\mathcal{F}$ and $\mathcal{T}$ are evolved along the trajectory and facilitate the mapping between the initial and current time frames.

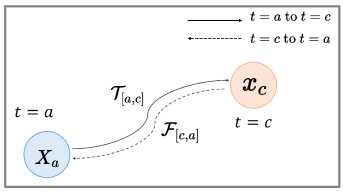

Fig. 2. Particle Flow Map starting from time a and ending at time c.

4 Coupling System

4.1 Solid-Fluid Coupling on Flow Maps

With the flow map theory in hand, we next consider a simple solid-fluid interaction problem. Suppose we have an immersed solid Ω^s in fluid Ω^f and their interface is denoted as $\partial\Omega^{s,f}$. We denote the force acting on fluid from solid as $\boldsymbol{f}^{s\to f}$ and the force on solid from the fluid as $\boldsymbol{f}^{f\to s} = -\boldsymbol{f}^{s\to f}$. From Euler's equation, the advection equation of impulse variable and the definition of fluid impulse, we can describe the evolution of fluid velocity and impulse in a coupling system as:

$$\begin{cases} \dfrac{D\boldsymbol{u}}{Dt} = -\nabla p + \boldsymbol{f}^{s\to f}, \\ \dfrac{D\boldsymbol{m}}{Dt} = -(\nabla\boldsymbol{u})^T\boldsymbol{m} \\ \boldsymbol{u} = \boldsymbol{m} - \boldsymbol{q}. \end{cases} \tag{6}$$

In the original impulse method, since there were no external forces, $\boldsymbol{q} = \nabla\varphi$ for $\boldsymbol{q}$ only needed to include pressure. In fluid-structure interaction problems, $\boldsymbol{q}$ needs to include external forces such as those exerted by the solid, hence the expression for $\boldsymbol{q}$ needs to be rederived. For simplicity in notation, we use $\boldsymbol{f} := f^{s\to f}$. From Eq. 6, we can obtain:

$$\begin{aligned} &\frac{D\boldsymbol{u}}{Dt} = \frac{D\boldsymbol{m}}{Dt} - \frac{D\boldsymbol{q}}{Dt}, \\ &-\nabla p + \boldsymbol{f} + \frac{D\boldsymbol{q}}{Dt} + (\nabla\boldsymbol{u})^T\boldsymbol{m} = 0, \\ &-\nabla p + \boldsymbol{f} + \frac{D\boldsymbol{q}}{Dt} + (\nabla\boldsymbol{u})^T\boldsymbol{u} + (\nabla\boldsymbol{u})^T\boldsymbol{q} = 0, \\ &\frac{D\boldsymbol{q}}{Dt} + (\nabla\boldsymbol{u})^T\boldsymbol{q} + \nabla(\frac{1}{2}|\boldsymbol{u}|^2) - \nabla p + \boldsymbol{f} = 0, \end{aligned} \tag{7}$$

Here, we used the vector identity $(\nabla \boldsymbol{u})^T \boldsymbol{u} = \frac{1}{2}|\boldsymbol{u}|^2$. By applying Eq. 4 and flow map identity $\mathcal{F}\mathcal{T} = \mathcal{I}$, we have:

$$\begin{aligned}
&\frac{D\boldsymbol{q}}{Dt} + \mathcal{T}^T \frac{D\mathcal{F}^T}{Dt}\boldsymbol{q} + [\nabla(\frac{1}{2}|\boldsymbol{u}|^2) - \nabla p + \boldsymbol{f}] = 0,\\
&\frac{\mathcal{F}^T D\boldsymbol{q}}{Dt} + \frac{D\mathcal{F}^T}{Dt}\boldsymbol{q} + \mathcal{F}^T[\nabla(\frac{1}{2}|\boldsymbol{u}|^2) - \nabla p + \boldsymbol{f}] = 0,\\
&\frac{D\mathcal{F}^T\boldsymbol{q}}{Dt} = \mathcal{F}^T[\nabla p - \nabla(\frac{1}{2}|\boldsymbol{u}|^2)] - \mathcal{F}^T\boldsymbol{f}.
\end{aligned} \tag{8}$$

Assume the flow map starts from time a and ends at time c, by integrating both sides of the above equation and multiplying with $\mathcal{T}^T$, we have the following equation at time c:

$$\boldsymbol{q}_c = \mathcal{T}^T_{[a,c]}[\int_a^c \mathcal{F}^T_{[\tau,a]}(\nabla p_\tau - \nabla\frac{1}{2}|u_\tau|^2)d\tau - \int_a^c \mathcal{F}^T_{[\tau,a]}\boldsymbol{f}_\tau d\tau]. \tag{9}$$

Now we have rederived an expression for $\boldsymbol{q}$ at time c with the presence of external forces and by applying the equation $\boldsymbol{u} = \boldsymbol{m} - \boldsymbol{q}$, we obtain the expression of fluid velocity at the end of the flow map as:

$$\begin{aligned}
\boldsymbol{u}_c = \quad & \underbrace{\mathcal{T}^T_{[a,c]}\boldsymbol{m}_a}_{\text{Impulse Mapping}} \quad \underbrace{-\mathcal{T}^T_{[a,c]}\int_a^c \mathcal{F}^T_{[\tau,a]}(\nabla p_\tau - \nabla\frac{1}{2}|u_\tau|^2)d\tau}_{\text{Projection}}\\
& + \boxed{\underbrace{\mathcal{T}^T_{[a,c]}\int_a^c \mathcal{F}^T_{[\tau,a]}\boldsymbol{f}_\tau d\tau)}_{\text{Coupling Force Integral}}},
\end{aligned} \tag{10}$$

where τ denotes the intermediate time instant along the trajectory of the flow map.

The three terms on the right-hand side of Eq. 10 indicate the flow map of fluid impulse, the projection, and the coupling force integral. For an impulse flow-map system without solid-fluid coupling, only the first two terms contribute to calculating velocity in each time step. For a solid-fluid coupling system, the third term is essential for evaluating the coupling effects from solid to fluid. As shown in Figure 3, the calculation of the coupling force integral can be comprehended as integrating the coupling force with flow maps along the entire trajectory. For each time instant, the integral is calculated by conducting a backward map from the current to the initial time and then conducting a forward map from the initial to the end.

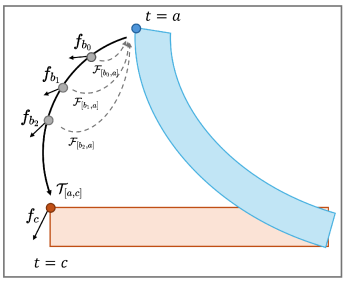

Fig. 3. Illustration for mapping coupling force from current to initial frame.

For the solid part, we will have:

$$\boldsymbol{v}_c = \boldsymbol{v}_a + \int_a^c \nabla\cdot\sigma_\tau(\mathcal{F}_\tau) - \boldsymbol{f}_\tau d\tau. \tag{11}$$

The solid velocity at time c is the time integral of both its elastic force and the fluid force from time a to time c. The velocity on the solid-fluid boundary at time c must satisfy $\boldsymbol{u}_c = \boldsymbol{v}_c$. In order to utilize existing solid simulation methods, in practice we use one-step advection to simulate solids in place of using a long-range mapping.

4.2 Impulse-To-Velocity Coupling

In this section, we introduce how we solve Eq. 8 and 10 with a set of reformulated flow-map equations. Our key idea is to convert the impulse-form flow map to the velocity-form flow map to solve the coupled system, allowing a direct coupling implementation based on the force exchange within a single time step. The mathematical idea of our impulse-to-velocity transfer was motivated by [Li et al. 2024], with our particular focus on deriving the formula in an Eulerian-Lagrangian setting for tackling solid-fluid coupling.

Particle Buffer. Our coupling scheme performs in the velocity domain as shown in Eq. 10. The calculation of two integrals is needed to convert the flow map advected impulse to velocity. We name the two buffers for computing integrals in Eq. 10 as pressure correction buffer $\boldsymbol{\Lambda}$ and coupling force buffer $\boldsymbol{\Upsilon}$:

$$\begin{cases}
\boldsymbol{\Lambda}_c &= \int_a^c \mathcal{F}^T_{[\tau,a]}\nabla(p_\tau - \frac{1}{2}|\boldsymbol{u}_\tau|^2)d\tau,\\
\boldsymbol{\Upsilon}_c &= \int_a^c \mathcal{F}^T_{[\tau,a]}\boldsymbol{f}_\tau d\tau.
\end{cases} \tag{12}$$

Grid-based flow map methods cannot easily calculate path integrals starting from a random point within the computational domain, as seen in [Deng et al. 2023; Sato et al. 2017] due to the need for a storage buffer for velocity fields. Meanwhile, as shown in [Li et al. 2024; Zhou et al. 2024], particles can easily accumulate over their trajectories. Therefore, we employed a particle flow map framework to integrate the force model into impulse fluid simulation. Specifically, we carry individual flow map gradients $\mathcal{F}$ and $\mathcal{T}$ and initial impulse variables $\boldsymbol{m}_a$ on fluid particles. Such design allows the particle buffers $\boldsymbol{\Lambda}$ and $\boldsymbol{\Upsilon}$ to be directly carried and calculated on particles at each time step, making using the particle buffers possible.

Impulse-to-Velocity Conversion. To use the particle buffers getting $\boldsymbol{u}_c$ from $\boldsymbol{m}_c$ in simulation, we need to discretize Eq. 10 and Eq. 12 that forms a single step update equation from time $b = c - \Delta t$ to c. However, in the definition of $\boldsymbol{m}$, $\boldsymbol{u}$ is a divergence-free velocity field. Therefore, Eq. 10 contains pressure at time c that's unknown and waiting to be solved by Poisson projection. Similarly, $\frac{1}{2}\boldsymbol{u}_c^2$ is unknown and waiting to be solved. Hence, the actual divergent velocity field $\boldsymbol{u}_c^*$ at time c needs to eliminate the pressure component from Eq. 10 and we use $\frac{1}{2}\boldsymbol{u}_b^2$ to approximate the original expression for $\frac{1}{2}\boldsymbol{u}_c^2$. And, in practice, $\frac{1}{2}\boldsymbol{u}_{\text{mid}}^2$ is used because of our adaptation of mid-point velocity approximation. Hence, Eq. 10 can now be written as:

$$\boldsymbol{u}_c^* = \mathcal{T}^T_{[a,c]}(\boldsymbol{m}_a) - \mathcal{T}^T_{[a,c]}(\mathcal{F}^T_{[b,a]}\boldsymbol{q}_b) + \Delta t\boldsymbol{f}_c + \Delta t(\nabla\frac{1}{2}|\boldsymbol{u}_b|^2) \tag{13}$$

for a single timestep update from $t = b$ to $t = c$. For deriving update equation for $\boldsymbol{\Lambda}$ and $\boldsymbol{\Upsilon}$, we perform discretization on Eq. 12 and derive:

$$\begin{aligned}
\boldsymbol{\Lambda}_c &= \boldsymbol{\Lambda}_b + \mathcal{F}^T_{[c,a]}\Delta t(\nabla p_c - \nabla\frac{1}{2}|\boldsymbol{u}_c|^2)\\
\boldsymbol{\Upsilon}_c &= \boldsymbol{\Upsilon}_b + \mathcal{F}^T_{[c,a]}\Delta t\boldsymbol{f}_c
\end{aligned} \tag{14}$$

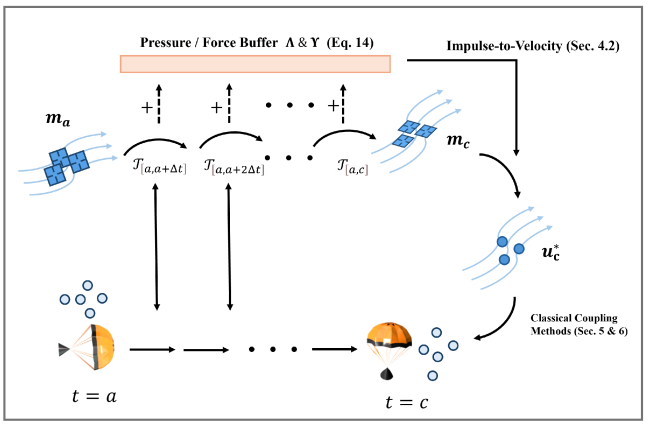

Fig. 4. Illustration of the key idea of our method where we convert impulse to velocity using our particle buffers Λ and Υ. The problem of direct coupling between impulse and velocity is shown in Section 7. Our whole pipeline is shown in Section 5 and the adaptation of our method to classical coupling methods is shown in Section 6.2 and 6.1.

Reformulated Flow-Map Equations. Now we have the system of equations for updating $\boldsymbol{m}_c$, $\boldsymbol{u}_c^*$, Λ_c, and Υ_c and we summerize them as follows:

$$\begin{cases} \boldsymbol{m}_c = \mathcal{T}^T_{[a,c]}\boldsymbol{m}_a, \\ \boldsymbol{u}_c^* = \boldsymbol{m}_c - \mathcal{T}^T_{[a,c]}(\Lambda_b - \Upsilon_b) + \nabla\frac{1}{2}|\boldsymbol{u}_b|^2\Delta t + \boldsymbol{f}_c\Delta t, \\ \Lambda_c = \Lambda_b + \mathcal{F}^T_{[c,a]}\Delta t(\nabla p_c - \nabla\frac{1}{2}|\boldsymbol{u}_c|^2), \\ \Upsilon_c = \Upsilon_b + \mathcal{F}^T_{[c,a]}\Delta t\boldsymbol{f}_c. \end{cases} \tag{15}$$

Notice we use $\boldsymbol{f}_c$ denote the coupling force between fluid and solid at time step c, but external forces like gravity $\boldsymbol{f}_c = \rho\boldsymbol{g}$ and viscosity $\boldsymbol{f}_c = \upsilon\Delta\boldsymbol{m}_c$ can also be incorporated to Υ in the same way. Furthermore, in scenarios where non-uniform density is present, Λ_c becomes:

$$\Lambda_c = \Lambda_b + \mathcal{F}^T_{[c,a]}\Delta t(\frac{1}{\rho}\nabla p_c - \nabla\frac{1}{2}|\boldsymbol{u}_c|^2). \tag{16}$$

Calulation of Λ_c. For evaluating Λ_c, it directly follows the update equation in Eq. 15. After solving the Poisson projection for time c, we can obtain p_c and $\boldsymbol{u}_c$ on the grid. We then calculate $\nabla(p_c - \frac{1}{2}|\boldsymbol{u}_c|^2)$ on the grid and transfer this quantity to particles using a G2P process and accumulate the value on Λ_c by mapping it back to the initial frame a using $\mathcal{F}_{[c,a]}$. The gradient operator can be evaluated as the gradient of the weight kernel used in PFM. Notice that ρ is a material property carried by particles. Therefore, the transition from uniform density to non-uniform density calculation is straightforward.

Calulation of Υ_c. As for evaluating Υ_c, same procedure is used. At time c, we first calculate the coupling force $\boldsymbol{f}_c$ along with optional external forces. Then, using the advected $\mathcal{F}_{[c,a]}$ on particles, we accumulate the forces to Υ by Eq. 15.

Calulation of $\boldsymbol{u}_c^*$. Having the two buffers in hand, converting from impulse $\boldsymbol{m}_c$ to divergent velocity $\boldsymbol{u}_c^*$ become straight forward. We use $\mathcal{T}^T_{[a,c]}\boldsymbol{m}_a$ to get $\boldsymbol{m}_c$. Then, by using $\boldsymbol{u}_{\text{mid}}$, we calculate $\nabla\frac{1}{2}\boldsymbol{u}^2_{\text{mid}}$ on particles. Now, all quantities are on particles, together with Λ and Υ calculated from the previous frame, we get $\boldsymbol{u}_c^*$ following Eq. 15.

Coupling with $\boldsymbol{u}_c^*$ and projection. Following the calculation of $\boldsymbol{u}_c^*$, the coupling method can be performed using this advected divergent velocity field because both solid and fluid are now described using velocity, which is a short-range physical property. Depending on the chosen coupling schemes, either Particle-to-Grid (P2G) or force spreading is performed, as detailed in Section 5. After this, coupling force $\boldsymbol{f}_c$ with external forces for a single frame is added to the grid. The velocity on the grid can then be used for Poisson projection to enforce a divergence-free condition.

Summary. In sum, by accumulating Λ and Υ on each particle, we achieve a divergent velocity field from impulse using flow map advection, which substitutes the semi-Lagrangian step in typical fluid solvers without requiring the typical projection operation. Now we have (1) solid and fluid using the same variable for representation and (2) forces are correctly handled through maintaining the force buffer Υ on particles, coupling between solid and fluid can fall back to utilizing methods used in typical velocity domain without the presence of a flow map.

5 Time Integration

We outline our time integration scheme in Alg. 1. Our pipeline allows any coupling method to be integrated as long as the computation of Eq. 10 especially the coupling force between solid and fluid can be accumulated to our coupling force buffer Υ. Examples with pseudo code of using this pipeline for coupling under MPM or IBM framework are shown in Appendix A.1.3 and Appendix A.2.2.

(1) **Reinitialization**. After every n steps, quantities carried on fluid particles are reinitialized in the manner adopted from PFM. Specifically, particles are uniformly redistributed, and both $\mathcal{F}^f_{[c,a]}$ and $\mathcal{T}^f_{[a,c]}$ are reset to $\mathcal{I}$, with $\boldsymbol{m}_a$ reset to the initial state using $\boldsymbol{u}_b$. In addition, Λ_b, Υ_b are emptied too.
(2) **CFL Condition for Fluid**. Δt is computed based on the fluid velocity field and the CFL number.
(3) **CFL Condition for Solid**. A separate sound CFL condition and velocity CFL condition are used for solid to determine Δt^s and the number of solid substeps.
(4) **Midpoint Method**. We utilize a leapfrog-style energy preservation temporal integration scheme to enhance vorticity preservation. Examples of the midpoint method for MPM are provided in Alg. 5 and IBM in Alg. 10.

Fluid. We predict $\boldsymbol{u}_{mid}$ by using $\boldsymbol{u}_b$.

Solid. We simulate half the number of solid substeps, with initial velocities sampled from $\boldsymbol{u}_b$, to synchronize with the fluid state.

Coupling. We utilize the P2G scheme from MPM or IBM force spreading followed by Poisson solving for incompressibility.

(5) **Fluid Advection**. We march $\boldsymbol{x}^f$, $\mathcal{T}_{[a,c]}$ and $\mathcal{F}_{[c,a]}$, according to Eq. 4.
(6) **Compute $\boldsymbol{m}_c$**. We update $\boldsymbol{m}_c$ with $\boldsymbol{m}_a$ and $\mathcal{T}^f_{[a,c]}$, according to Eq. 15.

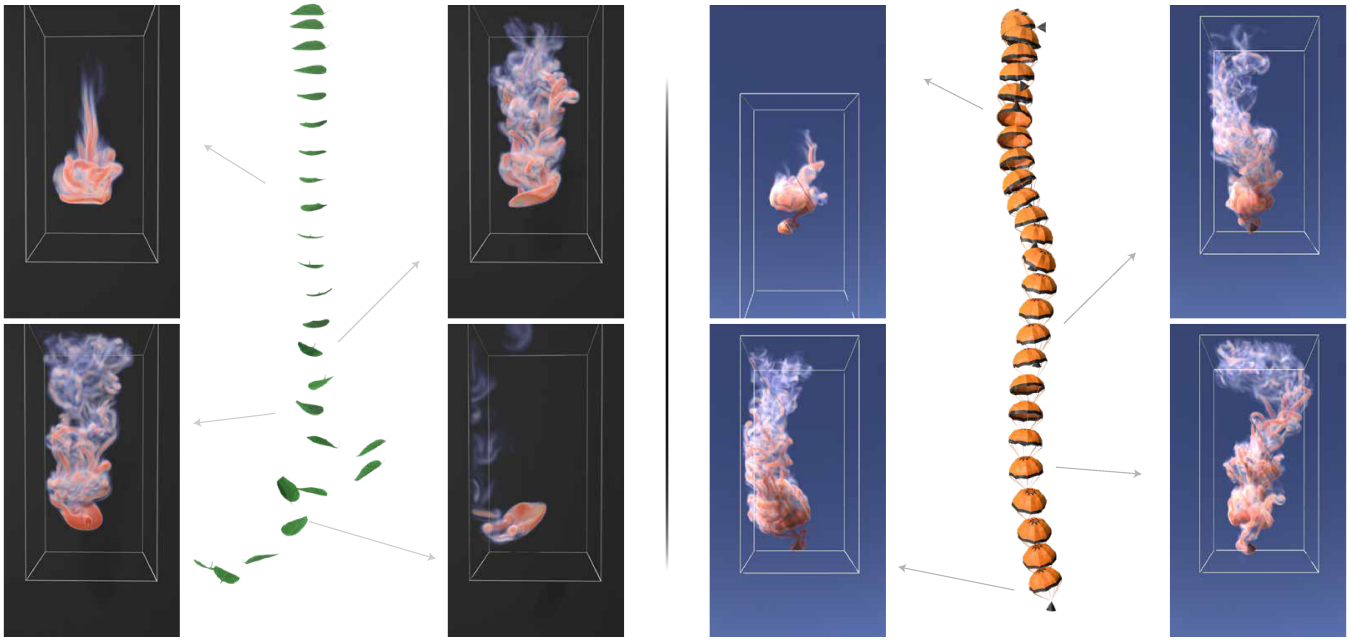

Fig. 5. We show the path and vorticity snapshots of free falling leaf and parachute. The falling path of the leaf is shown on the left. We can clearly observe the falling path experienced three stages: acceleration, converging, and chaos which accords with real-life observation. On the right, we show a parachute falling and we observe velocity convergence after the acceleration stage as expected.

(7) **Compute** ∇u_c^f. We compute ∇u_c^f with $\boldsymbol{u}_{mid}$ using

$$\nabla \boldsymbol{u}_c^f = \sum_i \nabla w_{ip} \boldsymbol{u}_{mid}, \tag{17}$$

where w_{ip} is the quadratic weight function [Jiang et al. 2016].

(8) **Solid Marching**. We simulate the remaining half number of solid substeps with initial velocity sampled from $\boldsymbol{u}_{mid}$ to synchronize with the fluid state.

(9) **Impulse to Velocity Conversion**. We compute $\boldsymbol{u}_c^{f*}$ with $\boldsymbol{\Lambda}_b$, $\boldsymbol{\Upsilon}_b$, $\boldsymbol{u}_{\text{mid}}$ and $\boldsymbol{m}_c$, according to Eq. 15.

(10) **P2G and Coupling**. Compute $\boldsymbol{u}_c^*$ by a P2G process using quantities carried on particles. When using MPM, we perform the P2G process for both fluid and solid using $\boldsymbol{u}_c^f$ and $\boldsymbol{u}_c^n$ (details provided in Section 6.1 and Appendix A.1) on fluid particles and $\boldsymbol{u}_c^s$ on solid particles. When using IBM, force is distributed to the velocity field after the P2G process.

(11) **Add External and Coupling Forces**. Coupling forces $\boldsymbol{f}_c$ and external forces like gravity $\rho \boldsymbol{g}$ and viscosity $\upsilon \Delta \boldsymbol{m}_c$ are added to the grid.

(12) **Poisson Solving**. We solve Poisson equation to ensure divergence-free condition.

(13) **Buffer Update**. We update $\boldsymbol{\Upsilon}_c$ and $\boldsymbol{\Lambda}_c$, according to Eq. 15.

Algorithm 1 Time Integration

for k in total steps **do**
  Reinitialization every n steps;
  Compute Δt with $\boldsymbol{u}_b$ and the CFL number;
  Determine Δt^s and number of substeps for solid;
  Estimate midpoint velocity $\boldsymbol{u}_{\text{mid}}$; ▷ Alg. 5/10
  March $\boldsymbol{x}^f$, $\mathcal{T}^f_{[a,c]}$ $\mathcal{F}^f_{[c,a]}$ with $\boldsymbol{u}_{\text{mid}}$ and Δt; ▷ Eq. 4
  Compute $\boldsymbol{m}_c$ with $\boldsymbol{m}_a$ and $\mathcal{T}^f_{[a,c]}$; ▷ Eq. 15
  Compute $\nabla \boldsymbol{u}_c^f$ using $\boldsymbol{u}_{\text{mid}}$; ▷ Eq. 17
  March solid based on solid simulation methods;
  Compute $\boldsymbol{u}_c^{f*}$ with $\boldsymbol{\Lambda}_b$, $\boldsymbol{\Upsilon}_b$, $\boldsymbol{u}_{\text{mid}}$, $\boldsymbol{m}_c$; ▷ Eq. 15
  Compute $\boldsymbol{u}_c^*$ by transferring $\boldsymbol{u}_c^{f*}(, \boldsymbol{u}_c^n, \boldsymbol{u}_c^s)$ and $\nabla \boldsymbol{u}_c^f$ to grid;
  ▷ Eq. 18
  Add single step coupling forces $\boldsymbol{f}_c$ and external forces;
  $\boldsymbol{u}_c, p_c \leftarrow$ **Poisson**$(\boldsymbol{u}_c^*)$;
  Update $\boldsymbol{\Upsilon}_c$ and $\boldsymbol{\Lambda}_c$ with $\boldsymbol{u}_c$, $\mathcal{F}^f_{[c,a]}$, $\boldsymbol{f}_c$ and p_c; ▷ Eq. 15
end for

6 Coupling Framework Examples

This section demonstrates two coupling frameworks on MPM and IBM using our flow-map coupling method. Each framework calculates the coupling forces that can be exchanged between solid and

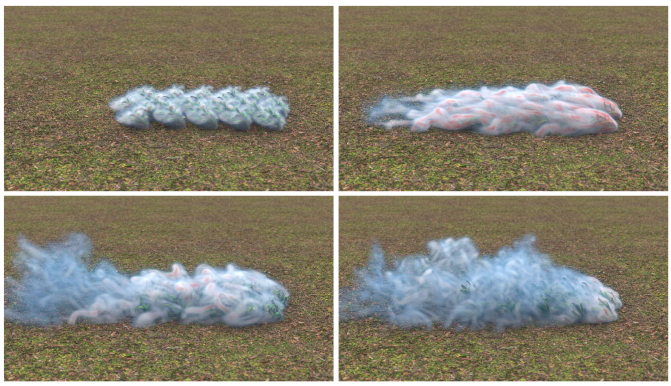

Fig. 6. Under incoming flow with variant velocity, turbulent surface flow is created from the grass geometry. Vorticity is being visualized here for illustration of the turbulent flow.

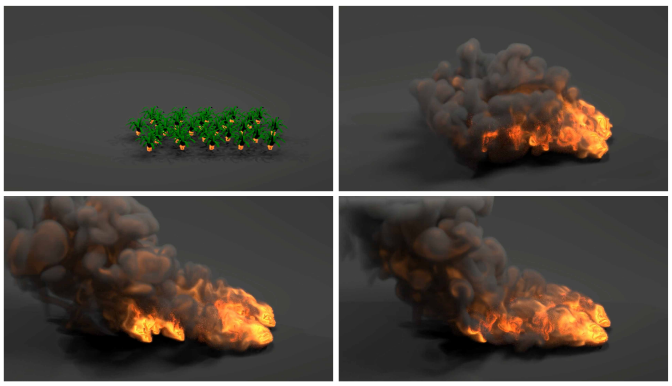

Fig. 7. We show the fire ignition on grassland leading to intense smoke and fire production. We see interesting vortical structures formed within the visualized smoke and fire regions.

fluid particles (implicitly for MPM and explicitly for IBM) and then accumulates these forces on our particle force buffers.

6.1 Coupling with MPM

This section demonstrates how to adapt our coupling method to the MPM framework. Couplings within the MPM framework are realized through P2G and G2P procedures and pressure projection for momentum exchange and coupling force calculation. Because the coupling force for MPM implicitly inherits from the pressure gradient in the projection step, our accumulation for $\mathbf{\Lambda}$ provides the buffer for accumulating coupling force under the MPM framework in flow map settings. Using our scheme provided in Section 5, fluid particles carrying impulse variables described under a long-range flow map can be converted to a short-range velocity in the same physical model as MPM particles. With this unified presentation, we perform P2G using typical P2G procedures for solid particles $\boldsymbol{u}_c^s$ and fluid particles $\boldsymbol{u}_c^f$. For simplicity, we use $\boldsymbol{u}_c^p$ to represent velocity for both kinds of particles:

$$\boldsymbol{u}_i \leftarrow \sum_p w_{ip}(\boldsymbol{u}_c^p + \nabla \boldsymbol{u}_c^p(\boldsymbol{x}_i - \boldsymbol{x}_p)) \,/\, \sum_p w_{ip}, \tag{18}$$

where weight kernel is chosen to be quadratic kernel in [Jiang et al. 2016] and $\boldsymbol{x}_i$ denotes MAC grid face center location.

The G2P process is performed for fluid and solid particles in the advection step. One difference from classical MPM simulation is that we perform RK4 time integration for advection for better energy conservation. The G2P process is performed with

$$\boldsymbol{u}^p \leftarrow \sum_i w_{ip}\boldsymbol{u}_i. \tag{19}$$

6.2 Coupling with IBM

The IBM coupling happens between solid and fluid through the exchange of elastic force spread through a smoothed Dirac-Delta kernel $\delta_h(\boldsymbol{x}) = \frac{1}{h^3}\phi(\frac{x}{h})\phi(\frac{y}{h})\phi(\frac{z}{h})$, where $\phi(r)$ is defined as:

$$\phi(r) = \begin{cases} \frac{1}{4}\left(1 + \cos\left(\frac{\pi r}{2}\right)\right) & |r| \leqslant 2 \\ 0 & \text{otherwise} \end{cases}, \tag{20}$$

Due to this explicit definition of coupling force, we can easily integrate the coupling force using our particle buffer Υ at each step. After we convert from the long-range mapped impulse variable to the short-range velocity for fluid particles and perform P2G for fluid particles, solid coupling forces are spread to the grid using the smoothed Dirac-Delta kernel as:

$$\boldsymbol{f}^{\text{grid}} = \int \boldsymbol{f}^{\text{solid}}\delta(\boldsymbol{x}^{\text{grid}} - \boldsymbol{X}^{\text{solid}}) \tag{21}$$

This force is accumulated to Υ at the end of this time step. The divergence-free velocity field interpolated to solid particles through the kernel function is used to march the solid particles following:

$$\boldsymbol{u}^{\text{solid}} = \int \boldsymbol{u}^{\text{grid}}\delta(\boldsymbol{x}^{\text{grid}} - \boldsymbol{X}^{\text{solid}}) \tag{22}$$

Here, $\boldsymbol{f}^{\text{grid}}$ corresponds to the force that is added to the velocity field after the impulse-to-velocity conversion, and $\boldsymbol{u}^{\text{grid}}$ represents the divergence-free velocity field after Poisson projection.

For implementation details, all advections are performed using the 4th order of the Runge-Kutta (RK4) method in [Deng et al. 2023; Zhou et al. 2024] to track flow maps on particles. Specifically, we follow the common practice in APIC that uses the gradient of kernel function to calculate $\nabla \boldsymbol{u}$. We provide more details for implementing MPM and IBM in Appendix A.1 and Appendix A.2.

7 Ablation Studies

In this experiment, we implemented six different alternatives in addition to our proposed method, including coupling on grid-based flow map methods with/without RMT [Rycroft et al. 2020] and force accumulation, hybrid flow map methods directly using P2G and with a thin layer of velocity particles. As shown in Figure 10, we tested these seven implementations in the example of a falling sphere immersed in the fluid and analyzed the results of their simulations. We use a simulation domain size of 128x384 and set gravity to 3. We use 0.03 as the radius for the sphere with a density ratio of 15:1 and $E = 5 \times 10^3$ and $\mu = 0.3$ throughout all the tests.

For grid-based flow map methods, we implemented the reference map method described in [Rycroft et al. 2020], which includes flow

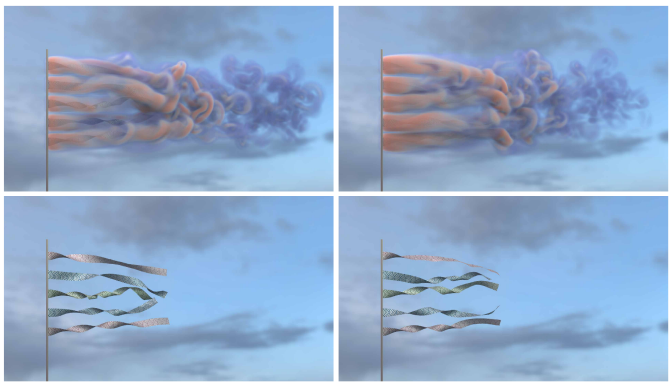

Fig. 8. In this experiment, we show a long silk flag affected by an incoming flow with variant velocity. Interesting spiral-shaped vortices are formed. We refer readers to our supplementary video for the dynamic motion of the vortices.

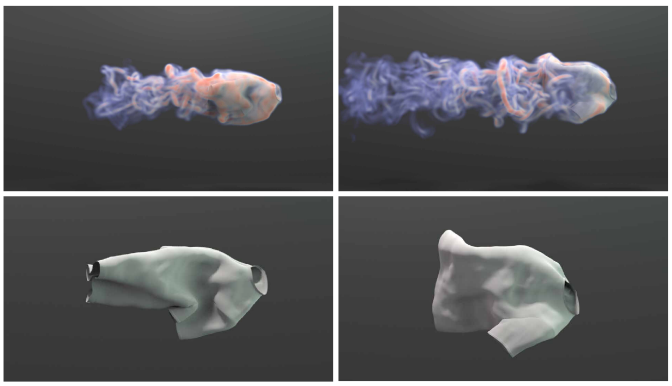

Fig. 9. T-shirt in wind field: we show high-frequency oscillation on the T-shirt geometry can be clearly observed. Such wrinkles of the T-shirt geometry are driven by the vorticity shredding on the surface which is automatically formed due to our coupling mechanism.

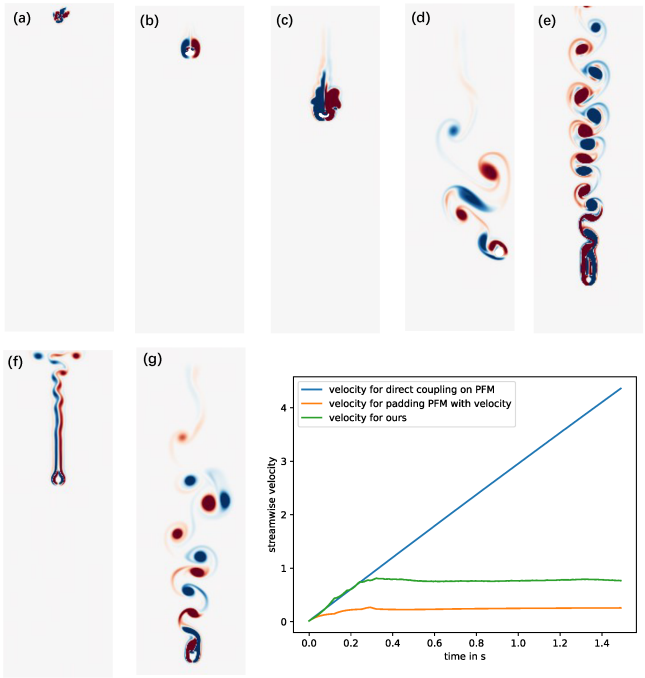

Fig. 10. On the left figure, we illustrate several approaches: (a) Direct GFMC, (b) Single-Step Heaviside GFMC,(c) Accumulative Heaviside GFMC, (d) Tuned Accumulative Heaviside GFMC, (e) Direct HFMC, (f) Padded Velocity HFMC, and (g) our method. In the right figure, we demonstrate that direct hybrid flow map coupling does not converge to a velocity, which is expected due to the fluid-to-solid forces.

map extrapolation and Heaviside blending for (1) fluid/solid density and (2) fluid stress τ^f and solid τ^s on grid-based flow map. At each step, a level set is created using MPM particles, and a Heaviside function (H) is calculated based on this level set on the grid. This Heaviside function is utilized to blend MPM solid stress τ^s with fluid viscosity stress τ^f, which is defined as $\mu^f(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$, and to blend solid density ρ^s with fluid density ρ^f. The blended value is defined as:

$$\begin{aligned} \tau &= \tau^f + H_\varepsilon(\tau^s - \tau^f) \\ \rho &= \rho^f + H_\varepsilon(\rho^s - \rho^f) \end{aligned} \tag{23}$$

and

$$H_\varepsilon(\phi) = \begin{cases} 0 & \text{if } \phi \leqslant -\varepsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin\frac{\pi\phi}{\varepsilon}\right) & \text{if } |\phi| < \varepsilon, \\ 1 & \text{if } \phi \geqslant \varepsilon, \end{cases} \tag{24}$$

where ϕ is the level set value and ε denotes the width of the blurring zone and is typically set at $2.5\Delta x$, as noted in [Rycroft et al. 2020]. The blurred τ is then used for solid elastic force calculation and ρ for P2G calculation. To utilize this blending scheme, specifically to calculate τ^s outside solid, we sample a narrowband of blur-zone particles in MPM simulation, and sampling follows the same way as described in Section A.1.1. Complicatedly, using this blending requires extrapolation of $\mathcal{F}^{\text{particle}}$ in the blur zone to calculate τ^s outside solid. This task lacks clear physical meaning, and the way we do this is to directly copy the $\mathcal{F}$ from the nearest surface particle to the particle in the blur zone. Forces calculated from P2G using blurred $\mathcal{F}$ are applied to impulse field advected by flow map.

(a) Direct Grid Flow Map Coupling (Direct GFMC). We directly use the deformation gradient on solids to calculate force, without blending with the forward map gradient $\mathcal{F}$ on fluids stored on the grid. We show in Figure 10(a) that numerical error dominates.

(b) Single-Step Heaviside GFMC. We add force for just a single frame within the grid-based flow map framework using the RMT method. We show in Figure 10(b) that such a method results in a slower falling speed than expected but also causes the solid to behave much softer than it should.

(c) Accumulative Heaviside GFMC. We add the accumulated force without mapping the force to the first frame within the flow map framework together with the RMT method. In Figure 10(c), we illustrate that this leads to numerical instability

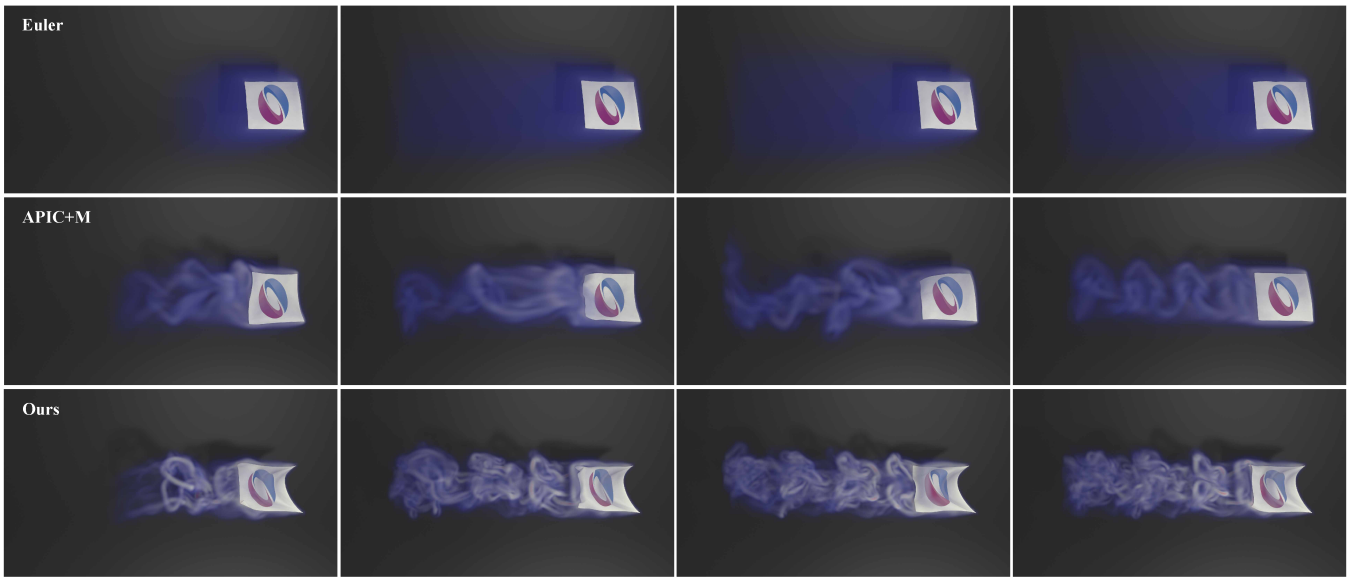

Fig. 11. We compare our method, Euler's method, and APIC+M. We observed no vorticities in Euler. APIC+M forms blurred vortices, and ours shows the clearest structure of vortices.

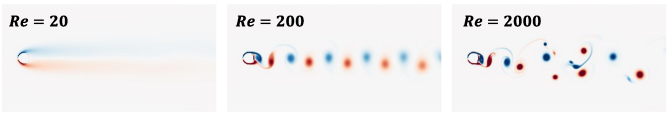

Fig. 12. Karman Vortex street under different Re number which is calculated from viscosity $\mu = 4 \times 10^{-4}, 4 \times 10^{-5}, 4 \times 10^{-6}$ with geometry radius being 0.05 and dominate velocity being 0.16.

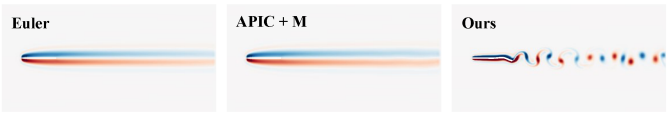

Fig. 13. This experiment compares our method with traditional fluid models within the IBM framework. Using a 2D mass-spring system with an incoming velocity of 0.16, our method demonstrates the formation of a vorticity street.

(d) Tuned Accumulative Heaviside GFMC. We use the same method as (c) but adjust the parameters to use smaller timestep, density ratio and gravity. However, distortions in flow map extrapolation become predominant, culminating in incorrect behavior as depicted in Figure 10(d).

(e) Direct Hybrid Flow Map Coupling (Direct HFMC). We use the impulse for fluid particles and directly P2G with solid particles carrying velocity and solve Poisson together. This method does not result in velocity convergence, as seen in Figure 10(e) and the right figure of Figure 10. This issue arises due to the differing representations of fluid and solid, specifically impulse versus velocity.

(f) Padded Velocity HFMC. We pad solid particles with a layer of fluid particles carrying velocity instead of impulse but didn't deal with fluid particles distant from the solid. Velocity convergence is observed. However, this approach makes fluid behavior anomalous due to the different models used for describing fluid dynamics as observed in Figure 10(f).

(g) Ours. We show our method correctly couples fluid and MPM solid without the issues presented above in Figure 10(g).

Based on the observations above, to correctly couple solid and fluid under the flow map framework, a method must ensure that (1) fluid and solid have the same physical representation when performing coupling and (2) forces are correctly managed within the flow map framework. Our framework successfully satisfies these requirements.

8 Validation and Comparison

In this section, we first verify the correctness of our external force treatment by validating it through the Karman Vortex Street experiment, which is run under different viscosity parameters leading to various Reynolds numbers and then on a cylinder with different density sedimentation that converges to different velocities. We then compare our method against Euler's method and affine particle-in-cell (APIC) as shown in Fig. 11. As highlighted in [Deng et al. 2023; Nabizadeh et al. 2022], a symmetric energy conservation scheme like the Leapfrog/Verlet method is crucial for vorticity conservation. We enhance the APIC method by adding a midpoint prediction, denoted as APIC+M, for a fair comparison.

8.1 2D Viscosity Test

We demonstrate that we can correctly handle viscosity with our treatment for external force, as discussed in Section 4.2. In Figure 12, we present the Karman Vortex Street phenomenon under different Reynolds numbers, specifically 20, 200, and 2000.

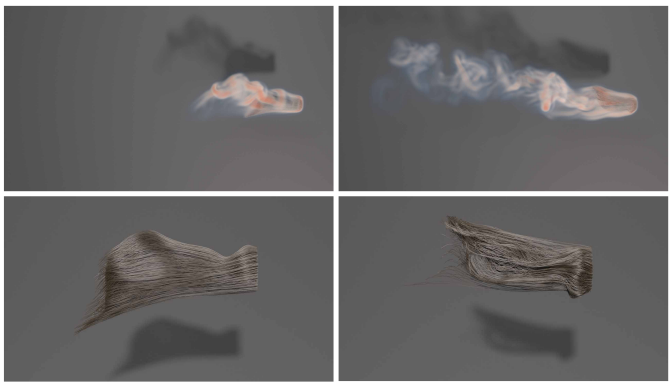

Fig. 14. We simulate hair using the model described in [Bender et al. 2014], employing XPBD and coupling it with our particle flow map fluid solver. Complex vortex structures emerge under the influence of a constant incoming flow, causing the hair to be lifted due to the flow's speed.

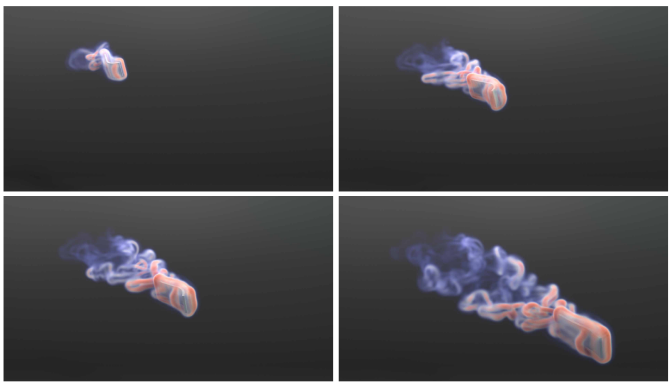

Fig. 15. We show the result of our method combining the active strain method to generate a self-contraction soft body simulated by MPM, creating a fish-like movement to move forward. Vortices behind the fishtail form a similar pattern as observed in [Lin et al. 2019].

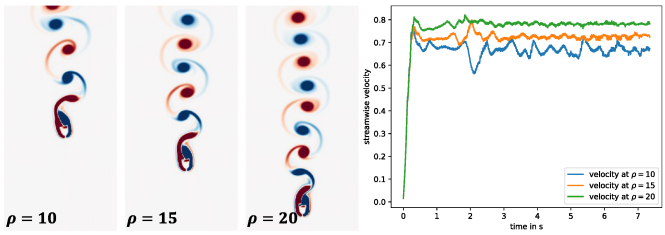

Fig. 16. The image on the left illustrates a comparison of the cylinder positions at time $t = 7.5$, highlighting the effects of different density settings on their behavior. All other parameters remain constant except for the density. The right image shows the convergences of the terminal velocity of the cylinders.

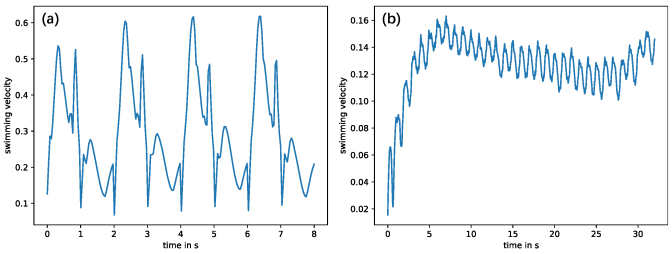

Fig. 17. This figure shows the solid velocity of our flapping swimmer and fish model shown in 2D swimmer experiment in Sec. 8.4 and fish experiment in Sec. 9.1. The velocity of this plot is calculated with finite difference using positions of two consecutive frames.

8.2 3D Bouyancy Test

We demonstrate that our algorithm can correctly handle external forces such as buoyancy in a 3D plume example. As shown in Figure 18, a turbulent smoke plume rises due to buoyancy and produces complex flow details.

Fig. 18. Turbulent smoke plume driven by buoyancy.

8.3 2D Cylinder Sediment

We demonstrate that with proper density treatment in handling impulse-to-velocity conversion and Poisson projection with Eq. 16, cylinders with different densities will converge to different velocities. The converged velocity is proportional to the cylinder density, as shown in Figure 16.

8.4 2D Swimmer

We compare our method against Euler's and APIC+M under the setting of a periodic flapping swimmer. We first introduce the method we used to create such motion, which is utilized not only for this comparison but also for 2D and 3D fish experiments.

Active strain contraction. We show our method adopting the active-strain method [Lin et al. 2019] to create periodic self-contraction, which moves solid through interaction with the fluid. The active strain approach imposes contractive strains to drive elastic deformation following a multiplicative decomposition of the deformation gradient tensor. This method defines a ,F_a as an active deformation tensor. To apply actuation on deformation gradient $\mathcal{F}$ is by assuming $\mathcal{F}_{\text{total}} = \mathcal{F} \cdot \mathcal{F}_a$ following multiplicative decomposition [Lee and Liu 1967; Lee 1969]. Therefore the deformation gradient $\mathcal{F}$ used in stress calculation is given by:

$$\mathcal{F} = \mathcal{F}_{\text{total}} \cdot (\mathcal{F}_a)^{-1}, \tag{25}$$

where $\mathcal{F}_{\text{total}}$ is the deformation gradient advected from previous timestep and $\mathcal{F}_a$ is the actuation applied.

We use the force model in [Lin et al. 2019] for $\mathcal{F}_a$. By assuming principle contraction in first dimension, $\mathcal{F}_a$ is defined as $\mathcal{F}_a = \text{diag}[\lambda, \lambda^{-1}]$ in 2D and $\mathcal{F}_a = \text{diag}[\lambda, \sqrt{\lambda^{-1}}, \sqrt{\lambda^{-1}}]$ in 3D where $\lambda < 1$. Contraction in other principle axes can also be applied similarly.

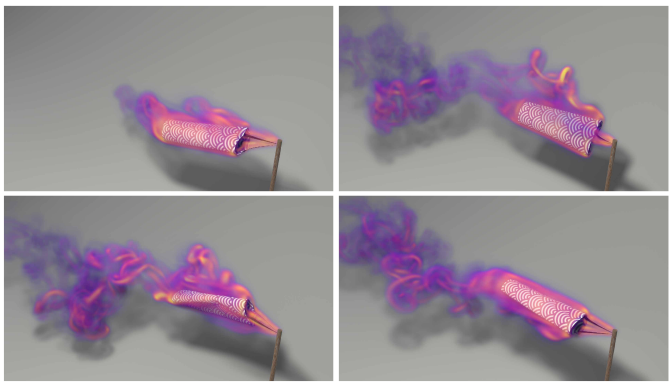

Fig. 19. We simulate a Koinobori, represented by a cylindrical shape adorned with a traditional Japanese cloud pattern, being lifted by a constant incoming flow. The waving motion of the Koinobori generates intricate vortex structures along its surface and tail, with the vorticity being visualized to highlight the turbulent flow dynamics.

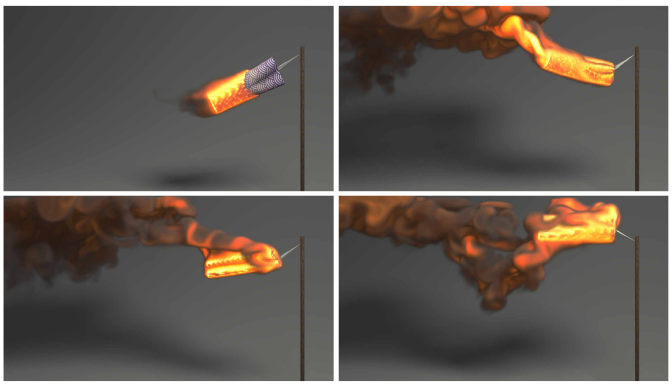

Fig. 20. In this experiment, we incorporate combustion into the Koinobori simulation, leading to the vivid visualization of fire and smoke colored with black-body radiation. As the flag waves under fiery conditions, it generates intricate vortex patterns and dynamic interactions between the flames, smoke, and the flag itself.

In calculation of λ, we use:

$$\lambda = \begin{cases} 1 - \alpha \sin\left(\frac{2\pi t}{T}\right) \exp\left(-\frac{h-y}{d_0}\right), & 0 \le \mathrm{t} \le \mathrm{T}/2, \\ 1 - \alpha \sin\left(\frac{2\pi t}{T}\right) \exp\left(-\frac{y}{d_0}\right), & \mathrm{T}/2 < \mathrm{t} \le \mathrm{T}, \end{cases} \tag{26}$$

for flapping swimmer experiment and use:

$$\lambda = \begin{cases} 1 - \alpha \| (\sin\left(\frac{2\pi t}{T}\right)) \| \exp\left(-\frac{h-y}{d_0}\right), & 0 \le \mathrm{t} \le \mathrm{T}/2, \\ 1 - \alpha \| \sin\left(\frac{2\pi t}{T}\right) \| \exp\left(-\frac{y}{d_0}\right), & \mathrm{T}/2 < \mathrm{t} \le \mathrm{T}, \end{cases} \tag{27}$$

for 2D and 3D fish experiment. Here, d_0 controls the steepness of the decay, and we use $h/3$ in all experiments, T denotes the period of contraction, y denotes the material space coordinate in the principle axis of contraction, and α controls the strength of contraction.

Comparison. In this experiment, the active contraction area is set between 0.3 and 0.7 of the long axis of the solid. We use $\alpha = 0.3$ and $T = 2$, with a dynamic viscosity of 8×10^{-6}. Our method preserves the clearest vortical structures compared to the other two methods, as shown in Figure 24. We also show the swimming velocity of our swimer in Figure 17(a).

8.5 2D / 3D Flag

In 2D flag comparison, we compare our method with Euler and APIC+M by applying the IBM method combined with a mass-spring system for the flag. As shown in Figure 13, numerical viscosity dominates the fluid behavior in the Euler and APIC+M methods, resulting in a lack of interesting fluid vorticity. In contrast, our method encourages the formation of interesting vorticity structures. In the 3D flag experiment in Figure 11, no vortical structures are observed in the simulation produced by the Euler solver. Compared to APIC+M, our method displays clear and regular vorticity structures and coupled flapping motions.

9 Examples

In this section, we show various 2D/3D experiments to validate our coupling simulator. Table 2 shows the comprehensive list of examples. It is assumed that the shortest edge of the simulation domain is of unit length. All simulations were implemented with Taichi [Hu et al. 2019] and performed on a laptop with an Intel i7-11800 H CPU and NVIDIA RTX 3080 GPU or a workstation equipped with AMD Ryzen Threadripper 5990X CPU and NVIDIA RTX 4090/A6000 GPUs. Depending on the GPU we used, our simulation takes an average of 1.7 seconds (E.g. 3D parachute or 3D Koinobori) to 5.2 seconds (E.g. 3D Fish or 3D Grass) for each step in 3D simulations.

9.1 MPM Coupling

The experiments below set the flow map reinitialization steps to 20 for 2D and 12 for 3D. The reinitialization step for narrowband fluid particles is set to 2 across all MPM coupling simulations.

2D cylinder sediment. In the experiment shown in Figure 21, we show the result of a 2D cylinder falling from rest and converging to a constant speed due to the force from fluid acting on solid. The ratio between fluid and solid is set to 15:1, and gravity is set to 3. The radius of the cylinder is 0.03, and the center is initially placed at [0.16, 0.5]. Fluid viscosity is set to be 8×10^{-5}. We see Karman Vortex Street patterns created in this falling process as discussed in [Gazzola et al. 2011].

2D multi-cylinder. As shown in Figure 22, cylinders are placed at a distance of $3r$ between their centers, and their radius is $r = 0.02$. The dynamic viscosity is set to be 6×10^{-4}, the density ratio is 30:1, and gravity is 9.8. We observe symmetric patterns created for both vorticity and particles' paths.[Gazzola et al. 2011]

2D Fish. As discussed in 8.4, we use the active strain method to create self-contraction simulating muscle movement. In the experiment shown in Figur 23, we set the activation area between 0.6 and 0.9 in the x direction in the solid body's material space and principal contraction in the y direction. Dynamic viscosity is set

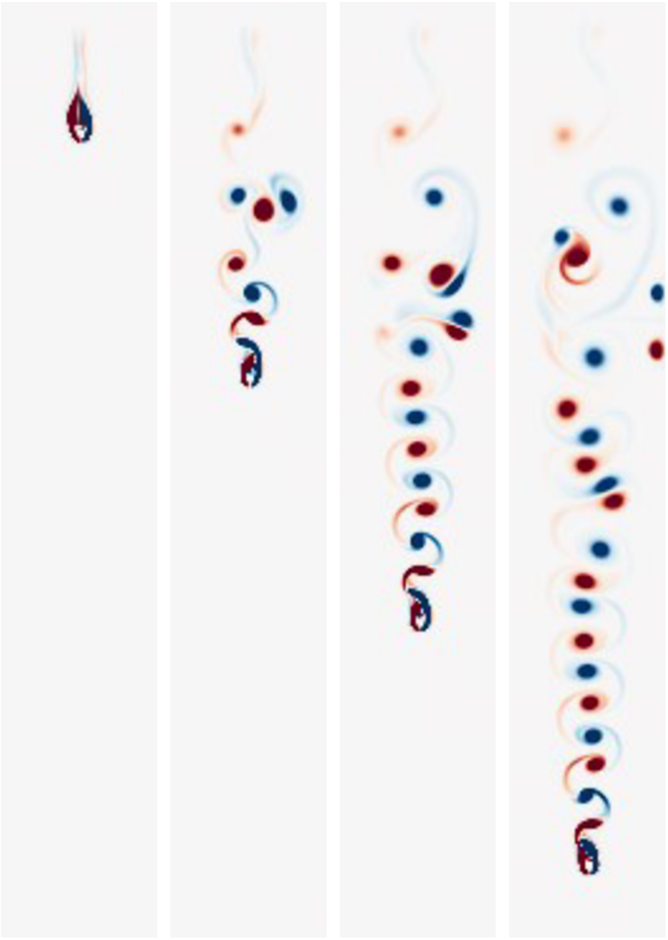

Fig. 21. Vorticity plot of 2D cylinder sediment at time $t = 1, 3, 5, 7$

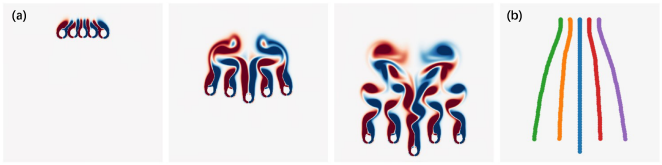

Fig. 22. Figure (a) shows multi-cylinder sediment at time $t = 0.15, 0.45, 0.65$. Figure (b) shows the trace for the center of mass of cylinders.

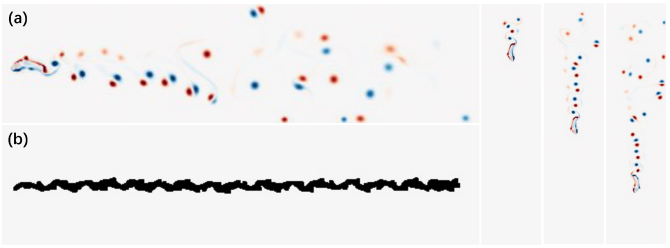

Fig. 23. Figure (a) shows fish movement with vorticity at time $t = 5, 16, 25, 32$. Figure (b) shows the trace for fish movement.

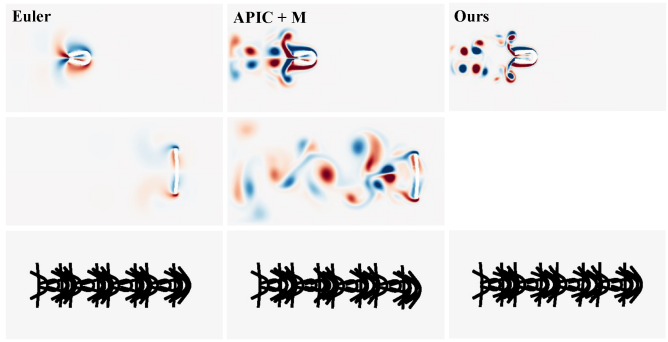

Fig. 24. This experiment shows comparison of coupling between fluid and MPM using active strain method for self contractoin

to be 8×10^{-6}. The solid body has a dimension of $x = 0.25$ and $y = 0.01$. $T = 2$ and $\alpha = 0.25$ is used for contraction. The vorticity plot is shown in Figure 23 and the swimming speed is shown in Figure 17(b).

3D Fish. We use a setting similar to the 2D fish experiment. The solid has dimension $x = 0.25$, $y = 0.01$ and $z = 0.1$. The principal contraction axis is the y direction and the activation area is between 0.6 and 0.9 in the x direction in the solid body's material space. $T = 1.8$ and $\alpha = 0.2$. The result is shown in Figure 15. We see a similar vortex structure created at the end of the tail compared to [Lin et al. 2019]

9.2 IBM Coupling

In all the experiments below, we use reinitialization steps of 12 for fluid, and extended position-based dynamics (XPBD) is used to solve solid behaviors. In the 3D leaf and 3D parachute experiments, we adopt a suiting grid for tracking the solid center of mass to allow a longer falling path.

3D Grass. We use multiple rectangular patches to form grass and fix their ending vertices at $y = 0$. An incoming flow with variant velocity $v = 0.2sin(2\pi t)+0.1$ is used. Details of vortices are observed created from the surface of the grass during simulation. The results are shown in Figure 6.

3D Long Silk Flag. An incoming flow of $v = 0.1sin(2\pi t) + 0.2$ is applied with a random velocity between $[0, 0.05]$. Interestingly, tweaking and torsion of the flag are observed, and spiral patterns of the vortical structure are formed during simulation. See results in Figure 8.

3D Koinobori. In Figure 19, we create a cylinder mesh with a bigger open-up radius at the beginning compared to the end and connect it to a fixed point. We aim to use this shape in correspondence to Japanese Koinobori. A constant incoming flow is applied.

3D Hair. In this experiment, we fixed the hair ends and initialized it at a 45-degree angle. We applied the model in [Bender et al. 2014] for PBD hair simulation. We see hair flowing up by incoming flow, forming clear vortex structures as shown in Figure 14.

3D T-shirt. Figure 9 shows a shirt in a turbulent flow field to demonstrate the solid-fluid coupling with a complicated mesh topology. Surface wrinkles and turbulent wake vortices can be observed in the simulation.

3D Leaf & 3D Parachute. In these two examples shown in Figure 5, we use a body-suiting computation domain to allow a longer falling path to be tracked and simulated. We show the path of the falling object and the vorticity snapshots. For vorticity snapshots, we change the camera position by the center of mass of geometry for rendering. For more details, we refer readers to our supplemented video for world space rendering of the vorticity and falling process.

3D Combustion. In Figure 20 and 7, we introduce combustion to the cylinder mesh depicted in Figure 19 and a wildfire scene on grass mesh in Figure 6. The combustion source is initiated at a vertex of the mesh and spread across the entire mesh. We also impose a constant temperature and density field at the source, which is advected using the same fashion as PFM and visualized as fire and smoke, respectively. An incoming flow with a varying velocity is applied. Additionally, buoyancy is incorporated through $\beta(T - T_{\text{ambient}})$ where β, T, and T_{ambient} represent the thermal expansion coefficient, temperature, and ambient temperature, respectively.

10 Discussion

We discuss the relation between our method and the previous impulse methods in the literature (e.g., [Deng et al. 2023; Feng et al. 2022; Nabizadeh et al. 2022; Sancho et al. 2024; Zhou et al. 2024]).

Impulse Stretching Stability. The current methods of simulating impulse fluid can be categorized into two categories: (1) evolving $\mathcal{F}$ and then solving $\mathcal{T}$ as its inverse [Feng et al. 2022; Sancho et al. 2024]; (2) evolving $\mathcal{F}$ and $\mathcal{T}$ directly (without solving inverse) along a long-range flow map (e.g., on grid [Deng et al. 2023; Nabizadeh et al. 2022] or on particles [Li et al. 2024; Zhou et al. 2024]). Impulse instability (i.e., a particle impulse magnifies itself due to a small numerical error, typically due to the stretching term, as reported in [Feng et al. 2022; Sancho et al. 2024]) was observed when $\mathcal{T}$ was calculated as the $\mathcal{F}$ inverse. This instability was not observed when forward evolving $\mathcal{T}$ on a particle flow map (e.g. [Deng et al. 2023; Zhou et al. 2024]). Our implementation follows the technical pathway of the second category by evolving $\mathcal{T}$ on particles. Therefore, we didn't employ extra treatment such as a Jacobian Limiter [Sancho et al. 2024] to ensure the impulse numerical stability.

Open Boundary Stability (CFL). We observed instability on the open boundary of the grid because $\mathcal{T}$ and $\mathcal{F}$ are undefined outside the domain. Unexpected fluctuating flow may enter the domain, limiting the CFL to a small number. This instability issue is closely related to the unsolved issue remaining in impulse simulations where the free surface / open boundary is present (e.g., [Li et al. 2024; Sancho et al. 2024]) and remains an open problem to solve in future works. Such boundary instability does not occur for a wall boundary.

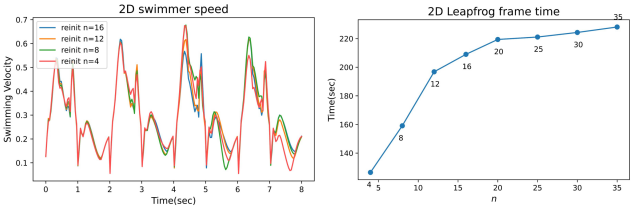

Fig. 25. We study the influence of initialization frequency and the coupled solid swimming speed in the right picture and the influence of its vorticity preservation in a 2D leapfrog scenario. We n to represent how many steps between reinitialization happens. We can see that solid speed is not largely influenced by this parameter. On the other hand, the vorticity preservation effect is not significantly improved after reinitialization frequency reaches 20.

Hessian Term in Flow Map. As noted in recent literatures like [Sancho et al. 2024], the accurate advection of $\boldsymbol{m}$ involving particle-to-grid transfer requires the calculation of $\nabla \boldsymbol{m}$ whose transport equation depends on $\nabla \mathcal{T}$ and can be written as:

$$\nabla \boldsymbol{m}(\boldsymbol{x}, t) = \mathcal{T}_t^T \, \nabla_{\psi} \boldsymbol{m}(\psi(\boldsymbol{x}), 0) \, \mathcal{T}_t + \nabla \mathcal{T}_t^T \, \boldsymbol{m}(\psi(\boldsymbol{x}), 0). \tag{28}$$

Such calculation (i.e., the calculation of $\nabla \mathcal{T}$) is challenging due to its high-order nature. However, we do not need to worry about this term because of the following reasons. Firstly and most importantly, in our method, by converting $\boldsymbol{m}$ to $\boldsymbol{u}$ first, the P2G transfer only requires $\nabla \boldsymbol{u}$, eliminating the problematic calculation of $\nabla \mathcal{T}$ from consideration. Secondly, in concurrent literature such as [Zhou et al. 2024], the authors propose to leave this term out of the calculation due to the observation that $\nabla \boldsymbol{m}$ mostly depends on the first term in Eq. 28.

Reinitialization Frequency. In Figure 25(a), we present the results of a parameter study investigating the influence of reinitialization frequency on solid velocity in solid-fluid coupling. We observe that the frequency of flow map reinitialization does not significantly impact the swimmer's speed. Nevertheless, reinitialization frequency is crucial for vorticity preservation, which is essential for smoke and soot visualization, as well as for managing multi-object and thin-shell object interactions (see Figure 22 and Figure 7). Through experimenting with different reinitialization frequency parameters, we observe that vorticity preservation ability converges once a certain threshold is reached (See Figure 25(b)). As noted in [Nabizadeh et al. 2022], excessively high reinitialization frequencies can lead to instability, particularly under conditions of substantial external force accumulation. Based on this observation, we have chosen to use a reinitialization frequency of $n = 20$ for 2D and $n = 12$ for 3D simulations instead of larger steps. For the number of particles per cell, we follow the standard practice in PFM, setting it to 16 for 2D and 8 for 3D, to obtain a balance between visual quality and GPU memory constraints.

11 Conclustion, Limitation, and Future Work

In conclusion, we present a unified representation of solid and fluid dynamics using particle flow maps, where a single-step flow map is applied for solid simulation to integrate existing elastic body simulations and a longer flow map for fluids to preserve vortex structures. By coupling these two maps through an impulse-to-velocity transfer mechanism and managing force accumulation via particle path integrals, we have developed a robust coupling framework that adapts traditional solid-fluid coupling techniques to flow map models, as demonstrated in our MPM and IBM examples.

The main limitation of our method is that the current framework relies on a weak coupling mechanism that depends on explicit force exchange between solid and fluid, rather than utilizing more sophisticated implicit momentum-conserving coupling schemes, such as the variational form [Batty et al. 2007] and monolithic projection [Robinson-Mosher et al. 2008]. These methods require a large system that implicitly includes the force exchange when formulating the system matrix. Integrating our method with these approaches would necessitate reformulating the two integrals in Eq. 10 into an implicit system over the flow map process to accommodate monolithic coupling. This system reformulation presents an intriguing avenue for future work, particularly for enabling the coupling between complex vortical structures and stiff systems such as rigid bodies and articulated bodies within flow map methods. Specifically, we are motivated to further adapt our flow-map framework to projection-based immersed boundary methods (e.g., see [Guermond et al. 2006; Taira and Colonius 2007; Wang and Eldredge 2015; Wang et al. 2020]) to address the interactions between flow maps and solids with hard constraints.

Another limitation of our method is that our pipeline is confined to the framework of particle-based flow-map representation for the fluid component, which requires maintaining a large set of flow-map particles to buffer the coupling forces and transfer flow-map information between particles and the background grid for the Poisson solve. Reducing the cost of maintaining these particles could potentially decrease the computational costs for long-range flow maps.

On another front, it would be interesting to explore purely Eulerian solid-fluid coupling methods based on flow maps (e.g., by extending the Eulerian solid-fluid interaction work [Teng et al. 2016] to a flow map framework), given the progress made with their Lagrangian counterparts. Additionally, our current system is focused on non-slip boundary conditions. It is unclear how to enforce other solid-fluid boundary conditions, such as non-penetration conditions, within a flow-map perspective. Motivated by progress in neighboring areas such as MPM ([Fang et al. 2020]), it would be valuable to explore different types of boundary conditions for solid-fluid coupling systems using flow maps.

Furthermore, our current approach does not support free-boundary flow. It would be intriguing to develop solid-fluid interaction frameworks that can simulate large-scale open-water phenomena (e.g., [Huang et al. 2021; Wretborn et al. 2022; Xiong et al. 2022]) with the flow-map-created vortical details around solid boundaries. The main challenge for the current approach is the difficulty of solving the impulse stretching terms for particles near the free surface (e.g., as pointed out by [Sancho et al. 2024]). This issue remains a significant gap for flow-map methods due to the difficulty of handling impulse stretching near a free surface.

Overall, our future work aims to tackle these problems by facilitating an implicit formulation for two-way coupling between solids and free surfaces, and multi-phase flow map fluids with different types of boundary conditions.

Acknowledgements

We express our gratitude to the anonymous reviewers for their insightful feedback. We also thank Yuchen Sun, Taiyuan Zhang, and Shiying Xiong for their insightful discussion. Georgia Tech authors also acknowledge NSF IIS #2433322, ECCS #2318814, CAREER #2433307, IIS #2106733, OISE #2433313, and CNS #1919647 for funding support. Tao Du thanks Tsinghua University and Shanghai Qi Zhi Institute for their support. We credit the Houdini education license for video animations.

References

Muzaffer Akbay, Nicholas Nobles, Victor Zordan, and Tamar Shinar. 2018. An extended partitioned method for conservative solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Nadir Akinci, Jens Cornelis, Gizem Akinci, and Matthias Teschner. 2013. Coupling elastic solids with smoothed particle hydrodynamics fluids. *Computer Animation and Virtual Worlds* 24, 3-4 (2013), 195–203.

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 100–es.

P Becker, SR Idelsohn, and E Oñate. 2015. A unified monolithic approach for multi-fluid flows and fluid–structure interaction using the particle finite element method with fixed mesh. *Computational Mechanics* 55 (2015), 1091–1104.

Jan Bender, Matthias Müller, Miguel A Otaduy, Matthias Teschner, and Miles Macklin. 2014. A survey on position-based simulation methods in computer graphics. In *Computer graphics forum*, Vol. 33. Wiley Online Library, 228–251.

I Borazjani and Fotis Sotiropoulos. 2010. On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming. *Journal of Experimental Biology* 213, 1 (2010), 89–107.

Landon Boyd and Robert Bridson. 2012. MultiFLIP for energetic two-phase fluid simulation. *ACM Transactions on Graphics (TOG)* 31, 2 (2012), 1–12.

TF Buttke. 1992. Lagrangian numerical methods which preserve the Hamiltonian structure of incompressible fluid flow. (1992).

Tomas F Buttke. 1993. Velicity methods: Lagrangian numerical methods which preserve the Hamiltonian structure of incompressible fluid flow. In *Vortex flows and related numerical methods*. Springer, 39–57.

Thomas F Buttke and Alexandre J Chorin. 1993. Turbulence calculations in magnetization variables. *Applied numerical mathematics* 12, 1-3 (1993), 47–54.

Mark Carlson, Peter J Mucha, and Greg Turk. 2004. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 377–384.

Pascal Clausen, Martin Wicke, Jonathan R Shewchuk, and James F O'brien. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Transactions on Graphics (TOG)* 32, 2 (2013), 1–15.

Ricardo Cortez. 1996. An impulse-based approximation of fluid motion due to boundary forces. *J. Comput. Phys.* 123, 2 (1996), 341–353.

Georges-Henri Cottet, Petros D Koumoutsakos, et al. 2000. *Vortex methods: theory and practice*. Vol. 313. Cambridge university press Cambridge.

Massimiliano Cremonesi, Alessandro Franci, Sergio Idelsohn, and Eugenio Oñate. 2020. A state of the art review of the particle finite element method (PFEM). *Archives of Computational Methods in Engineering* 27, 5 (2020), 1709–1735.

Zuo Cui, Zixuan Yang, Hong-Zhou Jiang, Wei-Xi Huang, and Lian Shen. 2018. A sharp-interface immersed boundary method for simulating incompressible flows with arbitrarily deforming smooth boundaries. *International Journal of Computational Methods* 15, 01 (2018), 1750080.

Yitong Deng, Hong-Xing Yu, Diyang Zhang, Jiajun Wu, and Bo Zhu. 2023. Fluid Simulation on Neural Flow Maps. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–21.

Th Dunne. 2006. An Eulerian approach to fluid–structure interaction and goal-oriented mesh adaptation. *International journal for numerical methods in fluids* 51, 9-10 (2006), 1017–1039.

Table 2. The catalog of all our 2D and 3D simulation examples.

Name	Figure	Resolution	CFL	Reinit. Steps of Flow Map	Particle Count Per Cell at Reinit. Step
2D Cylinder Sediment	Figure 21	764×128	0.5	20	16
2D Flag	Figure 13	256×128	0.5	20	16
2D Multi-Cylinder Sediment	Figure 22	256×256	0.5	20	16
2D Fish	Figure 23	512×128	0.5	20	16
2D Flapping Swimmer	Figure 24	256×128	0.5	20	16
3D Smoke Plume	Figure 18	$256 \times 128 \times 128$	0.5	12	8
3D Fish	Figure 15	$256 \times 128 \times 128$	0.5	12	8
3D Flag	Figure 11	$256 \times 128 \times 128$	0.25	12	8
3D Parachute	Figure 5	$256 \times 128 \times 128$	0.25	12	8
3D Leaf	Figure 5	$256 \times 128 \times 128$	0.25	12	8
3D Koinobori	Figure 19	$256 \times 128 \times 128$	0.25	12	8
3D Grass	Figure 6	$256 \times 128 \times 128$	0.25	12	8
3D Hair	Figure 14	$256 \times 128 \times 128$	0.25	12	8
3D Long Silk Flag	Figure 8	$256 \times 128 \times 128$	0.25	12	8
3D Koinobori Combustion	Figure 20	$256 \times 128 \times 128$	0.25	12	8
3D Grass Combustion	Figure 7	$256 \times 128 \times 128$	0.25	12	8
3D T-shirt	Figure 9	$256 \times 128 \times 128$	0.25	12	8

Yu Fang, Ziyin Qu, Minchen Li, Xinxin Zhang, Yixin Zhu, Mridul Aanjaneya, and Chenfanfu Jiang. 2020. 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 (2020), 51–1.

Yun Fei, Christopher Batty, Eitan Grinspun, and Changxi Zheng. 2018. A multi-scale model for simulating liquid-fabric interactions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–16.

Yun Fei, Christopher Batty, Eitan Grinspun, and Changxi Zheng. 2019. A multi-scale model for coupling strands with shear-dependent liquid. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–20.

Yun Fei, Henrique Teles Maia, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2017. A multi-scale model for simulating liquid-hair interactions. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–17.

Fan Feng, Jinyuan Liu, Shiying Xiong, Shuqi Yang, Yaorui Zhang, and Bo Zhu. 2022. Impulse fluid simulation. *IEEE Transactions on Visualization and Computer Graphics* (2022).

Mattia Gazzola, Philippe Chatelain, Wim M Van Rees, and Petros Koumoutsakos. 2011. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *J. Comput. Phys.* 230, 19 (2011), 7093–7114.

Olivier Génevaux, Arash Habibi, and Jean-Michel Dischler. 2003. Simulating Fluid-Solid Interaction.. In *Graphics Interface*, Vol. 2003. 31–38.

Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. 2005. Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 973–981.

Jean-Luc Guermond, Peter Minev, and Jie Shen. 2006. An overview of projection methods for incompressible flows. *Computer methods in applied mechanics and engineering* 195, 44-47 (2006), 6011–6045.

Toshiya Hachisuka. 2005. Combined Lagrangian-Eulerian approach for accurate advection. In *ACM SIGGRAPH 2005 Posters*. 114–es.

Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. 2019. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 201.

Libo Huang, Ziyin Qu, Xun Tan, Xinxin Zhang, Dominik L Michels, and Chenfanfu Jiang. 2021. Ships, splashes, and waves on a vast ocean. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–15.

Wei-Xi Huang and Hyung Jin Sung. 2009. An immersed boundary method for fluid–flexible structure interaction. *Computer methods in applied mechanics and engineering* 198, 33-36 (2009), 2650–2661.

Wei-Xi Huang and Fang-Bao Tian. 2019. Recent trends and progress in the immersed boundary method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 233, 23-24 (2019), 7617–7636.

Sergio R Idelsohn, Julio Marti, A Limache, and Eugenio Oñate. 2008. Unified Lagrangian formulation for elastic solids and incompressible fluids: application to fluid–structure interaction problems via the PFEM. *Computer Methods in Applied Mechanics and Engineering* 197, 19-20 (2008), 1762–1776.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*. 1–52.

Ken Kamrin and Jean-Christophe Nave. 2009. An Eulerian approach to the simulation of deformable solids: Application to finite-strain elasticity. *arXiv preprint arXiv:0901.3799* (2009).

Ken Kamrin, Chris H Rycroft, and Jean-Christophe Nave. 2012. Reference map technique for finite-strain elasticity and fluid–solid interaction. *Journal of the Mechanics and Physics of Solids* 60, 11 (2012), 1952–1969.

Richard Keiser, Bart Adams, Dominique Gasser, Paolo Bazzi, Philip Dutré, and Markus Gross. 2005. A unified lagrangian approach to solid-fluid animation. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005.* IEEE, 125–148.

Bryan M Klingner, Bryan E Feldman, Nuttapong Chentanez, and James F O'brien. 2006. Fluid animation with dynamic meshes. In *ACM SIGGRAPH 2006 Papers*. 820–825.

EH Lee and DT Liu. 1967. Finite stain elastic-plastic theory particularly for plane wave analysis. *J. Appl. Phys* 38 (1967), t9.

Erastus H Lee. 1969. Elastic-plastic deformation at finite strains. (1969).

Zhiqi Li, Barnabás Börcsök, Duowen Chen, Yutong Sun, Bo Zhu, and Greg Turk. 2024. Lagrangian Covector Fluid with Free Surface. In *ACM SIGGRAPH 2024 Conference Papers*. 1–10.

YP Lian, X Zhang, and Y Liu. 2011a. Coupling of finite element method with material point method by local multi-mesh contact method. *Computer Methods in Applied Mechanics and Engineering* 200, 47-48 (2011), 3482–3494.

YP Lian, X Zhang, and Y Liu. 2012. An adaptive finite element material point method and its application in extreme deformation problems. *Computer methods in applied mechanics and engineering* 241 (2012), 275–285.

YP Lian, X Zhang, X Zhou, and ZT Ma. 2011b. A FEMP method and its application in modeling dynamic response of reinforced concrete subjected to impact loading. *Computer Methods in Applied Mechanics and Engineering* 200, 17-20 (2011), 1659–1670.

Yan-Ping Lian, Yan Liu, and Xiong Zhang. 2014. Coupling of membrane element with material point method for fluid–membrane interaction problems. *International Journal of Mechanics and Materials in Design* 10 (2014), 199–211.

Zhaowu Lin, Andrew Hess, Zhaosheng Yu, Shengqiang Cai, and Tong Gao. 2019. A fluid–structure interaction study of soft robotic swimmer using a fictitious domain/active-strain method. *J. Comput. Phys.* 376 (2019), 1138–1155.

Chun Liu and Noel J Walkington. 2001. An Eulerian description of fluids containing visco-elastic particles. *Archive for rational mechanics and analysis* 159 (2001), 229–252.

Wing Kam Liu, Do Wan Kim, and Shaoqiang Tang. 2007. Mathematical foundations of the immersed finite element method. *Computational Mechanics* 39 (2007), 211–222.

Wing Kam Liu, Yaling Liu, David Farrell, Lucy Zhang, X Sheldon Wang, Yoshio Fukui, Neelesh Patankar, Yongjie Zhang, Chandrajit Bajaj, Junghoon Lee, et al. 2006. Immersed finite element method and its applications to biological systems. *Computer methods in applied mechanics and engineering* 195, 13-16 (2006), 1722–1749.

Frank Losasso, Geoffrey Irving, Eran Guendelman, and Ronald Fedkiw. 2006. Melting and burning solids into liquids and gases. *IEEE Transactions on Visualization and Computer Graphics* 12, 3 (2006), 343–352.

Yoichiro Mori and Charles S Peskin. 2008. Implicit second-order immersed boundary methods with boundary mass. *Computer methods in applied mechanics and engineering* 197, 25-28 (2008), 2049–2067.

Matthias Müller, Richard Keiser, Andrew Nealen, Mark Pauly, Markus Gross, and Marc Alexa. 2004. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 141–151.

Mohammad Sina Nabizadeh, Stephanie Wang, Ravi Ramamoorthi, and Albert Chern. 2022. Covector fluids. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.

Yen Ting Ng, Chohong Min, and Frédéric Gibou. 2009. An efficient fluid–solid coupling algorithm for single-phase flows. *J. Comput. Phys.* 228, 23 (2009), 8807–8829.

Valery Iustinovich Oseledets. 1989. On a new way of writing the Navier-Stokes equation. The Hamiltonian formalism. *Russ. Math. Surveys* 44 (1989), 210–211.

Charles S Peskin. 1972. Flow patterns around heart valves: a numerical method. *Journal of computational physics* 10, 2 (1972), 252–271.

Charles S Peskin. 2002. The immersed boundary method. *Acta numerica* 11 (2002), 479–517.

Ziyin Qu, Xinxin Zhang, Ming Gao, Chenfanfu Jiang, and Baoquan Chen. 2019. Efficient and conservative fluids using bidirectional mapping. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

PH Roberts. 1972. A Hamiltonian theory for weakly interacting vortices. *Mathematika* 19, 2 (1972), 169–179.

Avi Robinson-Mosher, R Elliot English, and Ronald Fedkiw. 2009. Accurate tangential velocities for solid fluid coupling. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 227–236.

Avi Robinson-Mosher, Craig Schroeder, and Ronald Fedkiw. 2011. A symmetric positive definite formulation for monolithic fluid structure interaction. *J. Comput. Phys.* 230, 4 (2011), 1547–1566.

Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 1–9.

Chris H Rycroft, Chen-Hung Wu, Yue Yu, and Ken Kamrin. 2020. Reference map technique for incompressible fluid–structure interaction. *Journal of Fluid Mechanics* 898 (2020), A9.

Sergio Sancho, Jingwei Tang, Christopher Batty, and Vinicius C Azevedo. 2024. The Impulse Particle-In-Cell Method. In *Computer Graphics Forum*. Wiley Online Library, e15022.

Takahiro Sato, Christopher Batty, Takeo Igarashi, and Ryoichi Ando. 2018. Spatially adaptive long-term semi-Lagrangian method for accurate velocity advection. *Computational Visual Media* 4, 3 (2018), 6.

Takahiro Sato, Takeo Igarashi, Christopher Batty, and Ryoichi Ando. 2017. A long-term semi-lagrangian method for accurate velocity advection. In *SIGGRAPH Asia 2017 Technical Briefs*. 1–4.

Robert Saye. 2016. Interfacial gauge methods for incompressible fluid dynamics. *Science advances* 2, 6 (2016), e1501869.

Robert Saye. 2017. Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: Part I. *J. Comput. Phys.* 344 (2017), 647–682.

Tokimasa Shimada, Koji Nishiguchi, Rahul Bale, Shigenobu Okazawa, and Makoto Tsubokura. 2022. Eulerian finite volume formulation using Lagrangian marker particles for incompressible fluid–structure interaction problems. *Internat. J. Numer. Methods Engrg.* 123, 5 (2022), 1294–1328.

TOKIMASA Shimada, KOJI Nishiguchi, Christian Peco, Shigenobu Okazawa, and Makoto Tsubokura. 2021. Eulerian formulation using lagrangian marker particles with reference map technique for fluid-structure interaction problem. In *of the International Conference on Computational Methods for Coupled Problems in Science and Engineering*. 1–7.

Barbara Solenthaler, Jürg Schläfli, and Renato Pajarola. 2007. A unified particle model for fluid–solid interactions. *Computer Animation and Virtual Worlds* 18, 1 (2007), 69–82.

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.

DM Summers. 2000. A representation of bounded viscous flow based on Hodge decomposition of wall impulse. *J. Comput. Phys.* 158, 1 (2000), 28–50.

Kunihiko Taira and Tim Colonius. 2007. The immersed boundary method: a projection approach. *J. Comput. Phys.* 225, 2 (2007), 2118–2137.

Tetsuya Takahashi and Christopher Batty. 2020. Monolith: a monolithic pressure-viscosity-contact solver for strong two-way rigid-rigid rigid-fluid coupling. (2020).

Tetsuya Takahashi and Christopher Batty. 2022. ElastoMonolith: A monolithic optimization-based liquid solver for contact-aware elastic-solid coupling. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–19.

Yun Teng, David IW Levin, and Theodore Kim. 2016. Eulerian solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–8.

Jerry Tessendorf. 2015. Advection Solver Performance with Long Time Steps, and Strategies for Fast and Accurate Numerical Implementation. (2015).

Fang-Bao Tian, Hu Dai, Haoxiang Luo, James F Doyle, and Bernard Rousseau. 2014. Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems. *Journal of computational physics* 258 (2014), 451–469.

Emad Uddin, Wei-Xi Huang, and Hyung Jin Sung. 2013. Interaction modes of multiple flexible flags in a uniform flow. *Journal of Fluid Mechanics* 729 (2013), 563–583.

Chengjie Wang and Jeff D Eldredge. 2015. Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method. *J. Comput. Phys.* 295 (2015), 87–113.

Li Wang and Fang-bao Tian. 2019. Numerical simulation of flow over a parallel cantilevered flag in the vicinity of a rigid wall. *Physical Review E* 99, 5 (2019), 053111.

Luohao Wang, Chunmei Xie, and Weixi Huang. 2020. A monolithic projection framework for constrained FSI problems with the immersed boundary method. *Computer methods in applied mechanics and engineering* 371 (2020), 113332.

Sheldon Wang. 2006. From immersed boundary method to immersed continuum methods. *International Journal for Multiscale Computational Engineering* 4, 1 (2006).

Xiaolin Wang, Ken Kamrin, and Chris H Rycroft. 2022. An incompressible Eulerian method for fluid–structure interaction with mixed soft and rigid solids. *Physics of Fluids* 34, 3 (2022).

X Sheldon Wang. 2007. An iterative matrix-free method in implicit immersed boundary/continuum methods. *Computers & structures* 85, 11-14 (2007), 739–748.

E Weinan and Jian-Guo Liu. 2003. Gauge method for viscous incompressible flows. *Communications in Mathematical Sciences* 1, 2 (2003), 317–332.

DC Wiggert and EB Wylie. 1976. Numerical predictions of two-dimensional transient groundwater flow by the method of characteristics. *Water Resources Research* 12, 5 (1976), 971–977.

Joel Wretborn, Sean Flynn, and Alexey Stomakhin. 2022. Guided bubbles and wet foam for realistic whitewater simulation. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.

S. Xiong, Z. Wang, M. Wang, and B. Zhu. 2022. A Clebsch method for free-surface vortical flow simulation. *ACM Trans. Graph.* 41, 4 (2022).

Shuqi Yang, Shiying Xiong, Yaorui Zhang, Fan Feng, Jinyuan Liu, and Bo Zhu. 2021. Clebsch gauge fluid. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–11.

Omar Zarifi and Christopher Batty. 2017. 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*. 1–11.

Hong Zhao, Jonathan B Freund, and Robert D Moser. 2008. A fixed-mesh method for incompressible flow–structure systems with finite solid deformations. *J. Comput. Phys.* 227, 6 (2008), 3114–3140.

Junwei Zhou, Duowen Chen, Molin Deng, Yitong Deng, Yuchen Sun, Sinan Wang, Shiying Xiong, and Bo Zhu. 2024. Eulerian-Lagrangian Fluid Simulation on Particle Flow Maps. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

A Implementation Details

A.1 Implementation detail for MPM

We explain our implementation for MPM solid substeps and removing fluid particles in solid regions. Solid substeps need to be coupled with fluid, and fluid particles in solid regions must be removed during reinitialization for flow maps and fluid particles. Therefore, we

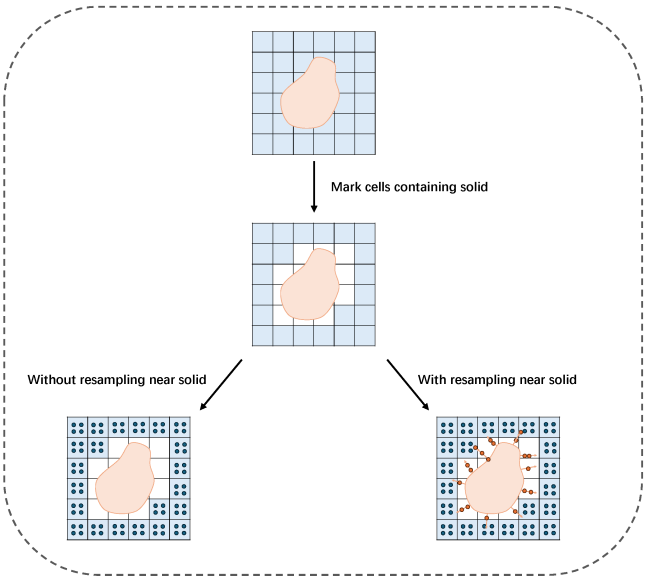

Fig. 26. We show that if we do not perform any resampling in the cells where fluid particles are removed, an empty band that contains no fluid particles will be created. And by using our resampling method, we can fill in this band easily.

choose to use narrowband particles resampled at a higher frequency than the fluid particles in distant regions to facilitate coupling at solid substeps with fluid and to remove fluid particles in solid regions.

A.1.1 Narrowband fluid particles & Resampling. In order to quickly remove particles in the solid region without the need to reconstruct the level set from particles at each step [Boyd and Bridson 2012], we utilize the deformation gradient carried by that are particles that are sampled exactly on a solid surface. Directly removing particles in every cell containing solid particles would create a gap between solid and fluid particles, as illustrated in Figure 26. Therefore, we need to find a method to resample fluid particles near the solid surface to fill in this gap.

Noticing that normal vectors can also be advected through the flow map according to the following relationship:

$$\boldsymbol{n}_c = \mathcal{T}_{[0,c]}^{-T} \boldsymbol{n}_0 \tag{29}$$

we sample on the solid surface before the simulation begins and document the surface points as $\boldsymbol{x}^{\text{surface}}$ and their initial surface normals as $\boldsymbol{n}_0^{\text{surface}}$. A random distance $D \in (0, 1.5\Delta x]$ is recorded for each $\boldsymbol{x}^{\text{surface}}$ before the simulation begins, and the resampling during each reinitialization step of narrowband fluid particles $\boldsymbol{x}^n$ naturally becomes:

$$\boldsymbol{x}^n = \boldsymbol{x}^{\text{surface}} + (\mathcal{T}_{[0,c]}^{\text{surface}})^T \boldsymbol{n}_0^{\text{surface}} D \tag{30}$$

We outline the algorithm for resampling in Alg. 2.

Algorithm 3 Reinit for coupling MPM

$j \leftarrow k \pmod{n^L}$;
$h \leftarrow k \pmod{n^S}$;
if $j = 0$ **then**
    Uniformly distribute particles;
    Mask out particles in solid
    Reinitialize $\boldsymbol{m}^f$ for all fluid particles $\boldsymbol{x}^f$;
    Reinitialize $\mathcal{T}^f$, $\mathcal{F}^f$ to identity
;
    Empty stored pressure correction buffer $\boldsymbol{\Lambda}^f$
    Empty stored external force buffer $\boldsymbol{\Upsilon}^f$
end if
if $h = 0$ **then**
    Resample near solid particles using solid normal
    Reinitialize $\boldsymbol{m}^n$ for all near solid fluid particles $\boldsymbol{x}^n$;
    Reinitialize $\mathcal{T}^n$, $\mathcal{F}^n$ to identity
;
    Empty stored pressure correction buffer $\boldsymbol{\Lambda}^n$
    Empty stored external force buffer $\boldsymbol{\Upsilon}^n$
end if

Algorithm 2 Sampling Method

1: Random sample points on surface and document as $\boldsymbol{x}^{\text{surface}}$ and their normals as $\boldsymbol{n}_0^{\text{surface}}$;
2: Perturb particle on surface along their normal $\boldsymbol{n}_0^{\text{surface}}$ with distance D;
3: Update $\boldsymbol{x}^{\text{surface}}$ and $\mathcal{T}^{\text{surface}}$ in simulation;
4: At t timestep, $\boldsymbol{x}^n = \boldsymbol{x}^{\text{surface}} + (\mathcal{T}_{[0,c]}^{\text{surface}})^T \boldsymbol{n}_0^{\text{surface}} D$;

Subsequently, the resampled narrowband fluid particles will always remain outside the solid, and we can safely remove fluid particles from every cell that contains solid particles at the reinitialization step.

A.1.2 Narrowband fluid particles in solid substeps. Based on the assumption that minimal movement will occur within solid substeps for fluid, we only utilize narrowband fluid particles for P2G and G2P during these substeps. Additionally, we use the $\mathcal{F}^n$ and $\mathcal{T}^n$ advected through solid substeps to update the impulse $\boldsymbol{m}^n$ and calculate $\boldsymbol{\Lambda}^n$ and $\boldsymbol{\Upsilon}^n$ when synchronizing solid and fluid states for coupling. Incompressibility is enforced by solving the Poisson equation during the synchronization step, rather than using the artificial bulk modulus cited in [Hu et al. 2018].

A.1.3 Pseudo Code for Coupling with MPM. Below we show the full pseudo-code for coupling impulse fluid with MPM under the time integration scheme we proposed in Algorithm 1. Limited modifications are required to adapt MPM into the pipeline as shown below.

Algorithm 4 Coupling with MPM

for k in total steps **do**
  Do reinit if needed based on Alg. 7
  Compute Δt with $\boldsymbol{u}_i$ and the CFL number;
  Compute Δt^s with solid parameter and the sound CFL;
  Decide the number of substeps for solid
  Estimate $\boldsymbol{u}_{\text{mid}}$ with Alg. 5
  March $\boldsymbol{x}_c^f$, $\mathcal{T}_{[a,c]}^f$ $\mathcal{F}_{[c,a]}^f$ with $\boldsymbol{u}_{\text{mid}}$ and Δt;
  March the other half MPM substep with Alg. 6
  G2P $\boldsymbol{u}_{sub}$ to get $\boldsymbol{u}_c^s$
  Get $\boldsymbol{m}_c^f$ and $\boldsymbol{m}_c^n$ using Eq. 15
  Compute $\boldsymbol{u}_c^{f*}$ using Λ_b^f, Υ_b^f, $\boldsymbol{u}_{\text{mid}}$, $\boldsymbol{m}_c^f$ with Eq. 15.
  Compute $\boldsymbol{u}_c^{n*}$ using Λ_b^n, Υ_b^n, $\boldsymbol{u}_{\text{mid}}$, $\boldsymbol{m}_c^n$ with Eq. 15.
  Compute $\nabla \boldsymbol{u}_c^f$ $\nabla \boldsymbol{u}_c^n$ using $\boldsymbol{u}_{\text{mid}}$
  Compute $\boldsymbol{u}_i$ by P2G using $\boldsymbol{x}_c^f$, $\boldsymbol{x}_c^n$, $\boldsymbol{x}_c^s$, $\boldsymbol{u}_c^f$, $\boldsymbol{u}_c^n$, $\boldsymbol{u}_c^s$, $\nabla \boldsymbol{u}_c^f$, $\nabla \boldsymbol{u}_c^n$, $\mathcal{F}^s$
  Add gravity on $\boldsymbol{u}_i$ if needed
  Add viscosity on $\boldsymbol{u}_i$ if needed
  P2G density carried by fluid and solid particles
  Solve Poisson
  Update Υ_c^f, Υ_c^n by adding external force to buffer following Eq. 15.
  Update Λ_c^f, Λ_c^n by adding pressure correction to buffer following Eq. 15.
end for

Algorithm 5 Midpoint MPM

March half MPM substep with Alg. 6
March $\boldsymbol{x}^f$ with $\boldsymbol{u}_i$ and $0.5\Delta t$ and get $\boldsymbol{u}^f$, $\nabla \boldsymbol{u}^f$
G2P $\boldsymbol{u}_{sub}$ to $\boldsymbol{u}^n$ and $\boldsymbol{u}^s$ and compute $\nabla \boldsymbol{u}^n$
P2G using $\boldsymbol{x}^f$, $\boldsymbol{x}^n$, $\boldsymbol{x}^s$, $\boldsymbol{u}^f$, $\boldsymbol{u}^n$, $\boldsymbol{u}^s$, $\nabla \boldsymbol{u}^f$, $\nabla \boldsymbol{u}^n$, $\mathcal{F}^s$
P2G density carried by fluid and solid particles
Solve Poisson and get $\boldsymbol{u}_{\text{mid}}$

Algorithm 6 MPM substep

$\boldsymbol{u}_{sub} \leftarrow \boldsymbol{u}$
for substeps **do**
  march $\boldsymbol{x}^n$, $\boldsymbol{x}^s$, $\mathcal{T}^n$, $\mathcal{F}^n$, $\mathcal{T}^s$, $\mathcal{F}^s$ with $\boldsymbol{u}_{sub}$
  G2P u_{sub} to $\boldsymbol{u}^n$ and $\boldsymbol{u}^s$ and compute $\nabla \boldsymbol{u}^n$
  P2G using $\boldsymbol{x}^n$, $\boldsymbol{x}^s$, $\boldsymbol{u}^n$, $\boldsymbol{u}^s$, $\nabla \boldsymbol{u}^n$, $\mathcal{F}^s$
end for

A.2 Implementation detail for IBM

A.2.1 Marching solid by XPBD. Except for the 2D validation tests, where we used a mass-spring system for our solid model, we employ XPBD to simulate solids for better visual effects. We applied the edge length and bending constraints as outlined in [Bender et al. 2014]. We used an iteration count of 50 for solving XPBD constraints in all our simulations, with solid Δt_{solid} set to 0.0005 for stability.

Algorithm 7 Reinit for coupling MPM

$j \leftarrow k \pmod{n^L}$;
$h \leftarrow k \pmod{n^S}$;
if $j = 0$ **then**
  Uniformly distribute particles;
  Mask out particles in solid
  Reinitialize $\boldsymbol{m}^f$ for all fluid particles $\boldsymbol{x}^f$;
  Reinitialize $\mathcal{T}^f$, $\mathcal{F}^f$ to identity
;
  Empty stored pressure correction buffer Λ^f
  Empty stored external force buffer Υ^f
end if
if $h = 0$ **then**
  Resample near solid particles using solid normal
  Reinitialize $\boldsymbol{m}^n$ for all near solid fluid particles $\boldsymbol{x}^n$;
  Reinitialize $\mathcal{T}^n$, $\mathcal{F}^n$ to identity
;
  Empty stored pressure correction buffer Λ^n
  Empty stored external force buffer Υ^n
end if

One caveat to note is that we use substeps for solid calculations. Therefore, the force that is spread to the grid velocity is calculated in an explicit fashion, as follows:

$$\boldsymbol{f} = \rho \frac{\boldsymbol{u}_c - \boldsymbol{u}_b}{\Delta t} \tag{31}$$

Algorithm 8 Reinit for coupling IBM

$j \leftarrow k \pmod{n^L}$;
if $j = 0$ **then**
  Uniformly distribute particles;
  Reinitialize $\boldsymbol{m}_s$ for all fluid particles $\boldsymbol{x}^f$;
  Reinitialize $\mathcal{T}^f$, $\mathcal{F}^f$ to identity
;
  Empty stored pressure correction buffer Λ^f
  Empty stored external force buffer Υ^f
end if

A.2.2 Pseudo Code for Coupling with IBM. Below we show the full pseudo-code for coupling covector/impulse fluid with IBM under the general pipeline we proposed in Algorithm 1. The only modification required is to use XPBD to solve for solid behaviors.

Algorithm 9 Coupling with IBM

for k in total steps **do**
  Do reinitialization if needed based on Alg. 8
  Compute Δt with $\boldsymbol{u}_i$ and the CFL number;
  Determine Δt^s;
  Decide the number of substeps for solid
  Estimate $\boldsymbol{u}_{\text{mid}}$ with Alg. 10
  March $\boldsymbol{x}^f$, $\mathcal{T}^f_{[a,c]}$ $\mathcal{F}^f_{[c,a]}$ with $\boldsymbol{u}_{\text{mid}}$ and Δt;
  March the other half IBM substep with Alg. 11
  Get $\boldsymbol{m}_c$ using Eq. 15
  Compute $\boldsymbol{u}_c^{f*}$ using $\boldsymbol{\Lambda}_b$, $\boldsymbol{\Upsilon}_b$, $\boldsymbol{u}_{\text{mid}}$, $\boldsymbol{m}_c$ with Eq. 15.
  Compute $\nabla \boldsymbol{u}_c^f$ using $\boldsymbol{u}_{\text{mid}}$
  Compute $\boldsymbol{u}_i$ by P2G using $\boldsymbol{x}_c^f$, $\boldsymbol{u}_c^f$, $\nabla \boldsymbol{u}_c^f$
  Add gravity on $\boldsymbol{u}_i$ if needed
  Add viscosity on $\boldsymbol{u}_i$ if needed
  Compute force $\boldsymbol{f}^s$ with Update $\rho \frac{(\boldsymbol{u}_c^s - \boldsymbol{u}_b^s)}{\Delta t}$
  Spread $\boldsymbol{f}^s$ with IBM kernel to $\boldsymbol{u}_i$
  Solve Poisson
  Get $\boldsymbol{f}^f$ on fluid particles using G2P
  Update $\boldsymbol{\Upsilon}_c$ by adding external force to buffer as Eq. 15.
  Update $\boldsymbol{\Lambda}_c$ by adding pressure correction to buffer as Eq. 15.
end for

Algorithm 10 Midpoint IBM

Get solid velocity using IBM kernel and $\boldsymbol{u}_i$
March $\boldsymbol{x}^s$ with Alg. 11
Compute force $\boldsymbol{f}^s$ with Update $\rho \frac{(\boldsymbol{u}_{b+0.5\Delta t} - \boldsymbol{u}_b)}{\Delta t}$
Get $\boldsymbol{u}^*_{\text{mid}}$ with RK4 semi-Lagragian update.
Spread force with IBM kernel to $\boldsymbol{u}^*_{\text{mid}}$
Solve Poisson to get $\boldsymbol{u}_{\text{mid}}$

Algorithm 11 IBM Solid substep

for substeps **do**
  if With XPBD **then**
    Update α based on Δt for each $\mathbb{C}$
    Predict solid location $\boldsymbol{x}^s$ with forward Euler
    Solve with XPBD iterations and get updated $\boldsymbol{x}^s$
    Update $\boldsymbol{u}^s$
  else
    Explicit Euler solving spring-mass system to update $\boldsymbol{x}^s$
  end if
end for