

An Efficient B-Spline Lagrangian/Eulerian Method for Compressible Flow, Shock Waves, and Fracturing Solids

YADI CAO, YUNUO CHEN, and MINCHEN LI, UCLA
 YIN YANG, Clemson University
 XINXIN ZHANG, Zenus Technology
 MRIDUL AANJANEYA, Rutgers University
 CHENFANFU JIANG, UCLA



Fig. 1. A car drives through a triggered landmine. The body and the wheels are made of different materials, leading to their distinctive interactions with the blowing exhaust. Our method naturally captures the corresponding effects and solves the solid-gas coupling problem in a monolithic style.

This study presents a new method for modeling the interaction between compressible flow, shock waves, and deformable structures, emphasizing destructive dynamics. Extending advances in time-splitting compressible flow and the Material Point Methods (MPM), we develop a hybrid Eulerian and Lagrangian/Eulerian scheme for monolithic flow-structure interactions. We adopt the second-order WENO scheme to advance the continuity equation. To stably resolve deforming boundaries with sub-cell particles, we propose a blending treatment of reflective and passable boundary conditions inspired by the theory of porous media. The strongly coupled velocity-pressure system is discretized with a new mixed-order finite element formulation employing B-spline shape functions. Shock

wave propagation, temperature/density-induced buoyancy effects, and topology changes in solids are unitedly captured.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; • **Mathematics of computing** → *Discretization*;

Additional Key Words and Phrases: Compressible fluids, gas, FEM, MPM, FSI

ACM Reference format:

Yadi Cao, Yunuo Chen, Minchen Li, Yin Yang, Xinxin Zhang, Mridul Aanjaneya, and Chenfanfu Jiang. 2022. An Efficient B-Spline Lagrangian/Eulerian Method for Compressible Flow, Shock Waves, and Fracturing Solids. *ACM Trans. Graph.* 41, 5, Article 169 (May 2022), 13 pages. <https://doi.org/10.1145/3519595>

1 INTRODUCTION

Supersonic motions and the detonation of explosive devices lead to shock waves passing through the air. With violent changes in pressure, shock waves can transport a tremendous amount of energy, leading to destructive effects on structures such as fragmentation of rocks, rupture of organic tissues, and the blast of soil.

Reliably simulating these phenomena imposes significant challenges in the efficient treatment of both the compressible flow and its two-way interaction with solids under extreme deformation and topological changes. Igniting explosive materials in fluids produces drastic motion exceeding the speed of sound, destroying surrounding matters with a high energy shock wavefront. Such phenomena challenge available numerical schemes in multiple ways.

Many existing schemes in computer graphics focus on simulating incompressible fluids. Among them, hybrid

This work has been supported in part by NSF CAREER 2153851, CCF-2153863, ECCS-2023780, DOE ORNL contract 4000171342. M. A. was supported in part by the Rutgers University startup grant, NSF CCF-2110861 and IIS-2132972. Y. Y. was supported in part by NSF 2016414 and 2011471. We also appreciate the supports from Zenus Technology, Taichi Graphics, and Adobe Research.

Authors' addresses: Y. Cao, Y. Chen, M. Li, and C. Jiang, Department of Computer Science, UCLA, 404 Westwood Plaza, Los Angeles, CA 90095; emails: yadi-cao95@ucla.edu, {yunuo, minchen, chenfanfu.jiang}@gmail.com; Y. Yang, School of Computing, Clemson University, 821 McMillan Rd, Clemson, SC 29631; email: yin5@clemson.edu; X. Zhang, Zenus Technology, Yuehai Rd Nanshan District, Shenzhen Guangdong 518000, China; email: zhangshinshin@gmail.com; M. Aanjaneya, Department of Computer Science, Rutgers University, 110 Frelinghuysen Rd, Piscataway, NJ 08854; email: mridul.aanjaneya@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0730-0301/2022/05-ART169 \$15.00

<https://doi.org/10.1145/3519595>

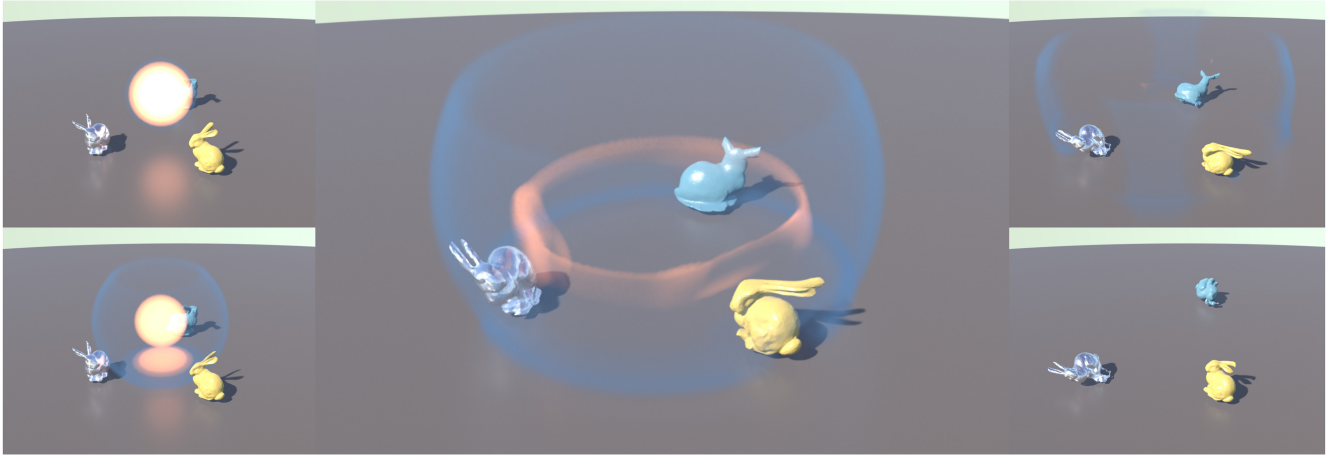


Fig. 2. “Tribunny”: three bunny models surrounding an explosion. The bunnies are made of iron, (yellow) rubber with a large density, and (blue) rubber with a small density. The iron bunny suffers from some permanent damage on its feet; the heavier elastic bunny almost restores its original shape at the end of the animation; the lighter elastic bunny is blown away, spinning in the air.

Lagrangian/Eulerian methods such as the **Particle-in-Cell (PIC)** methods [Brackbill et al. 1988; Bridson 2015; Jiang et al. 2015, 2017; Fu et al. 2017] are widely adopted. PIC methods track fluid’s motion via particles, allowing practitioners to create initial fluid volumes easily or emit fluids from sources. In addition, the time-splitting scheme [Chorin 1967; Bridson 2015] separates the nonlinear advection step from other steps that are linear so that every step can be solved efficiently. Various Eulerian advection schemes have been developed to handle large time steps [Stam 1999; Kim et al. 2005; Qu et al. 2019]. In particular, the advection process automatically conserves mass if the fluid is represented with particles. On the other hand, enforcing incompressibility is often through pressure projection, which corresponds to solving a pressure Poisson’s equation. The pressure projection system is usually **Symmetric Positive Definite (SPD)** [Batty et al. 2007; Bridson 2015] with a uniform grid representation of a constant fluid density. For liquids with a free surface, it is also common to utilize the sparsity of the domain [Setaluri et al. 2014; Wu et al. 2018] to further reduce the memory usage and the computational cost.

Modifying incompressible flow solvers, researchers in computer graphics have developed methods for modeling explosion effects by artificially introducing volume changes. A popular approach is to apply “divergence control” to force the pressure divergence to be a specified source value as if the density is nonuniform. This approach has been used to mimic the appearance of an expanding smoke plume [Feldman et al. 2003; Takeshita et al. 2003]. The Boussinesq approximation [Spiegel and Veronis 1960] is another approximation that mimics the buoyancy-like effects. Some work also devised a procedural explosion model based on a grid-based incompressible flow simulation [Kawada and Kanai 2011]. However, these approaches ignore many physical quantities except for velocity when solving for compressible flow fields, potentially resulting in misleading effects. In the Boussinesq approximation-based methods, relying on a buoyancy model from the temperature field to allow smoke to rise even falls out of its reliable regime and can

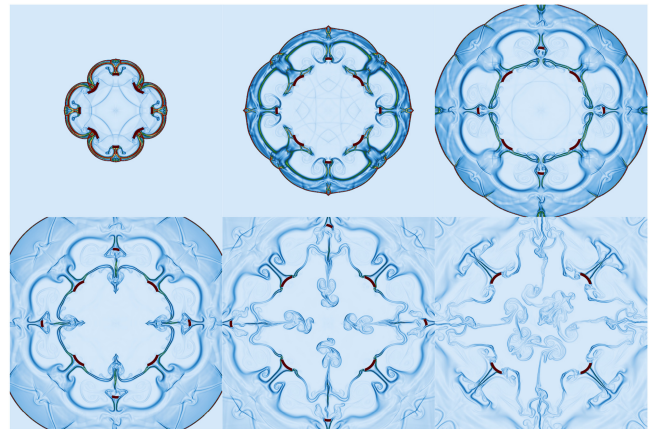


Fig. 3. Exploding elastic shell. The first three snapshots (top) show the expansion of the primary shock. The subsequent three snapshots (bottom) show the mixing of the back-flowing exhaust.

suffer from non-convergence issues. Overall, despite their sometimes plausible visual quality, these methods lack physical accuracy, and the quality of their results may rely on tuning artificial parameters.

Unfortunately, a more rigorous extension from incompressible flow to compressible flow is quite nontrivial. The following challenges arise:

- It is no longer feasible to use particles and sparse grids to represent the fluid domain because the ambient is not a vacuum. This leads to significantly more **degrees of freedom (DOFs)**.
- The direct one-step approach is often preferred as opposed to operator splitting since there exist internal relations (the Equation of State) between the pressure and other conserved variables [Forrer and Jeltsch 1998; Forrer and Berger 1999; Monasse et al. 2012]. This also introduces extra complexity in

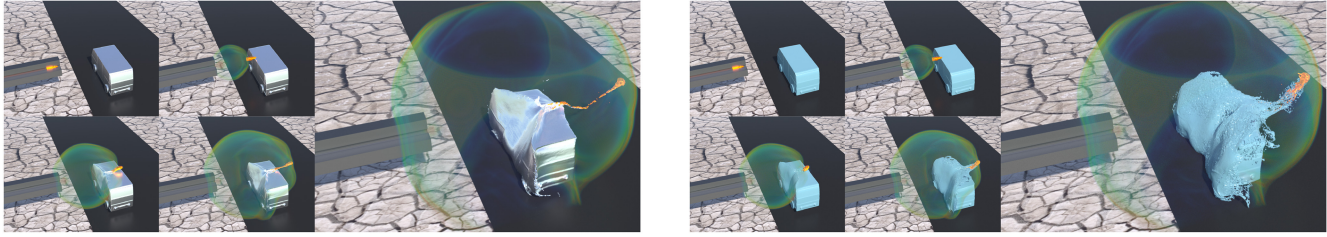


Fig. 4. Firing at toy trucks. The trucks are made of different materials: (left, strong) high Young's modulus and von Mises yield stress; (right, weak) low Young's modulus and von Mises yield stress. The stronger truck "feels" the shock and gets partially squashed. The weaker truck is drastically destroyed.

handling the nonlinear terms with various methods, such as the characteristic decomposition [Deconinck et al. 1993; Fey 1998].

- The more accurate advection schemes such as the **Weighted Essentially Non-Oscillatory (WENO)** scheme [Shu and Osher 1988; Liu et al. 1994] need to be conservative and often require tiny time step sizes strictly bounded by a small CFL number and the sound speed.
- The linear system associated with implicit integration is not guaranteed to be SPD because of different densities and the resulting coefficients in the mass matrix.

Under the circumstance that the increase of the DOF count is unavoidable, a lot of existing research in the compressible flow literature has focused on dealing with the last three challenges above.

1.1 Related Work

An extended amount of mechanical literature exists on the traditional partitioned methods for coupling compressible flow with solids. The most well-known categories include the arbitrary Lagrangian Eulerian method [Banks et al. 2016], the immersed boundary method [Pasquariello et al. 2016; Wang et al. 2017; Monasse et al. 2012], the discontinuous Galerkin method [Kosik et al. 2015], the ghost fluid method [Bailoor et al. 2017], and the Lattice Boltzmann method [Li and Favier 2017]. Many such methods only support small deformation or vibration. Some of them support large deformation without fracture. In computer graphics, Sewall et al. [2009] also adopted the partitioned method and weakly coupled shockwaves with rigid bodies.

A characteristic visual feature of explosions and shockwaves is their destructive effects on structures. Without modeling fracture mechanics, the above works cannot simulate destructive visual effects. Furthermore, these methods' partitioned (instead of monolithic) approaches often suffer from numerical instabilities when large deformation occurs.

To enable fracturing of solids, we adopt the recently advanced **Material Point Method (MPM)** [Sulsky et al. 1995; Stomakhin et al. 2013; Zhang et al. 2016; Jiang et al. 2016] for the solid domain. Nowadays, MPM is widely used in computer graphics and computational engineering for simulating elastoplastic materials with topology change. Due to the hybrid usage of meshless Lagrangian particles and Eulerian grids, MPM automatically captures material separation and contact.

MPM also naturally simulates mixed materials [Jiang et al. 2016; Hu et al. 2018] without the need for modeling the explicit

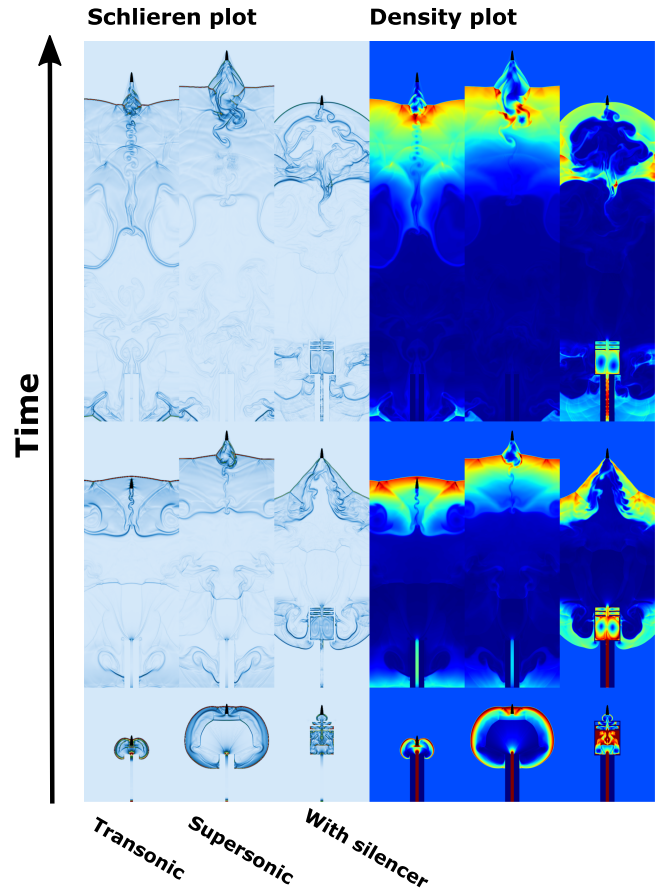


Fig. 5. Bullets in 2D. The schlieren & density plots for firing a heavy bullet (left, $\rho = 1250kg/m^3$), a light bullet (middle, $\rho = 500kg/m^3$), and a light bullet with a silencer at the outlet of the chamber (right, $\rho = 500kg/m^3$). The snapshots are at (from bottom to top) 0.028s, 0.0935s, and 0.17s.

coupling among them. However, due to the usage of a single grid, MPM suffers from artificial adhesion artifacts at material interfaces. Furthermore, it is computationally expensive and memory intensive to represent gas with particles. Therefore, we only model the solid phase with MPM and adopt an Eulerian grid-based discretization for the fluid phase. Existing work has coupled MPM sediments with Eulerian incompressible fluids [Gao et al. 2018a; Baumgarten et al. 2021]. Ma et al. [2009]'s method switches from MPM to FDM during the detonation process based on whether

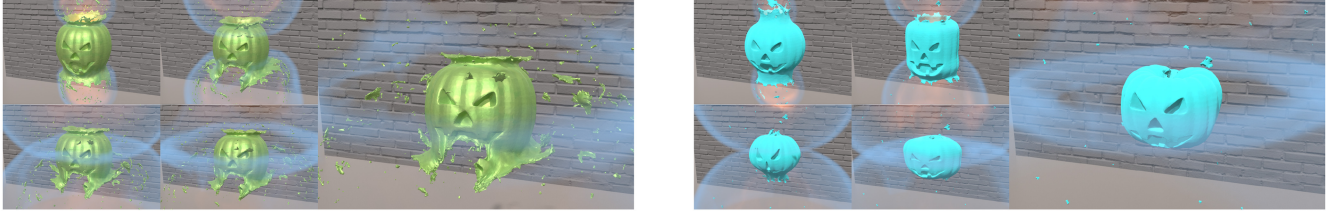


Fig. 6. Explosion of plastic (left) and elastic (right) pumpkins. The high internal pressure causes the pumpkin shells to fracture and blow away their fragments. The plastic pumpkin's deformation remains permanent; its fragments remain stretched and thin. The elastic pumpkin jiggles in the air, feeling the back-flow.

the material interaction or the gas pressure evolution is dominant. However, they do not model the coupling between the two phases.

In summary, most existing schemes for coupling compressible flow with deformable solids do not simultaneously support strong coupling, solid fracture, and efficient integration of the compressible flow. The motivation of our work is thus to develop a new computationally efficient, monolithic, and fracture-supporting strong coupling scheme for destructive effects due to shockwaves and high-speed flows.

A highly related work is the **Interface Quadrature Material Point Method (IQ-MPM)** by Fang et al. [2020]. IQ-MPM uses MPM particles to represent both solid and incompressible liquid phases. Strong coupling is achieved by modeling the interfacial interaction with a layer of imaginary quadrature points. Their approach is computationally efficient, only requiring the solution of an SPD system per time step. We extend some of their ideas from incompressible liquids to compressible flow.

The last piece of our framework concerns the performance of the compressible flow solver. We follow Kwatra et al. [2009, 2010]'s work, which successfully performed the time-splitting method in the compressible flow regime, removed the strict restriction of the sound speed and built an SPD pressure-only system—a system that can be integrated into the coupling phase from [Fang et al. 2020]. In recent years, the semi-implicit solver in [Kwatra et al. 2009, 2010] was further improved in its stability [Grétarsson et al. 2011], mass conservation [Grétarsson and Fedkiw 2013], and subgrid resolution [Hyde and Fedkiw 2019]. These works couple gas with sediments and simple elastic bodies. They also capture simple fracture events by plugging pressure-induced forces into a rigid body fracture tool. On the other hand, our work supports arbitrarily complex fracture events of nonlinear elastic solids under extreme deformation, as well as splitting and merging dynamics of plastic materials, without extra efforts beyond having an MPM solver for modeling these processes.

1.2 Contributions

In summary, we present a new method for simulating potentially destructive interactions between compressible flow (shock waves) and nonlinear elastoplastic solids. Our technical contributions include

- A hybrid Eulerian Finite Element and Lagrangian/Eulerian Material Point scheme for monolithic compressible flow and nonlinear structure interactions;

- A mixed treatment of reflective and passable interfaces inspired by the porous media theory, enabling a stable *sub-grid* resolution—a critical component for modeling the interaction between gas and fragments; and
- A new mixed-order finite element discretization using B-spline kernels with *mismatching* interface pressure orders and (thus) *non-staggered* solid/fluid degrees of freedom, avoiding extra interpolation steps and leading to a diagonally dominant SPD system for the velocity-pressure unknowns.

In contrast to Kwatra et al. [2010]'s method, which couples staggered **Marker-and-Cell (MAC)** grid Finite Difference-based fluids with purely Lagrangian solids, our method is based on the *non-staggered* grid, which avoids extrapolation in the advection step for the compressible flow. Our framework is based on operator splitting, and it remains stable under large time step sizes determined by the pressure gradient and the maximum velocity. All of our examples ran with a CFL number 0.5.

2 METHOD

2.1 Governing Equations

Following standard literature, here, we describe our governing equations: a conservative form of the inviscid compressible Euler equations for fluids and the conservation of mass and momentum for elastoplastic solids. Introducing density ρ , velocity \mathbf{u} , and pressure p , the compressible flow (\bullet^f) region is governed by

$$\frac{\partial \rho^f}{\partial t} + \nabla \cdot (\rho^f \mathbf{u}^f) = 0, \quad (1)$$

$$\frac{\partial \rho^f \mathbf{u}^f}{\partial t} + \nabla \cdot (\rho^f \mathbf{u}^f \otimes \mathbf{u}^f) + \nabla p^f = 0, \quad (2)$$

$$\frac{\partial \rho^f E}{\partial t} + \nabla \cdot [(\rho^f E + p^f) \mathbf{u}^f] = 0, \quad (3)$$

and the solid region (\bullet^s) is governed by

$$\frac{\partial \rho^s}{\partial t} + \nabla \cdot (\rho^s \mathbf{u}^s) = 0, \quad (4)$$

$$\rho^s \frac{\partial \mathbf{u}^s}{\partial t} + \mathbf{u}^s \cdot \nabla \mathbf{u}^s - \nabla \cdot \boldsymbol{\sigma} - \rho^s \mathbf{g} = 0. \quad (5)$$

At the solid-gas interface we enforce the slip velocity condition and pressure continuity:

$$(\mathbf{u}^s - \mathbf{u}^f) \cdot \mathbf{n} = 0, \quad (6)$$

$$p^f - p^s = 0. \quad (7)$$

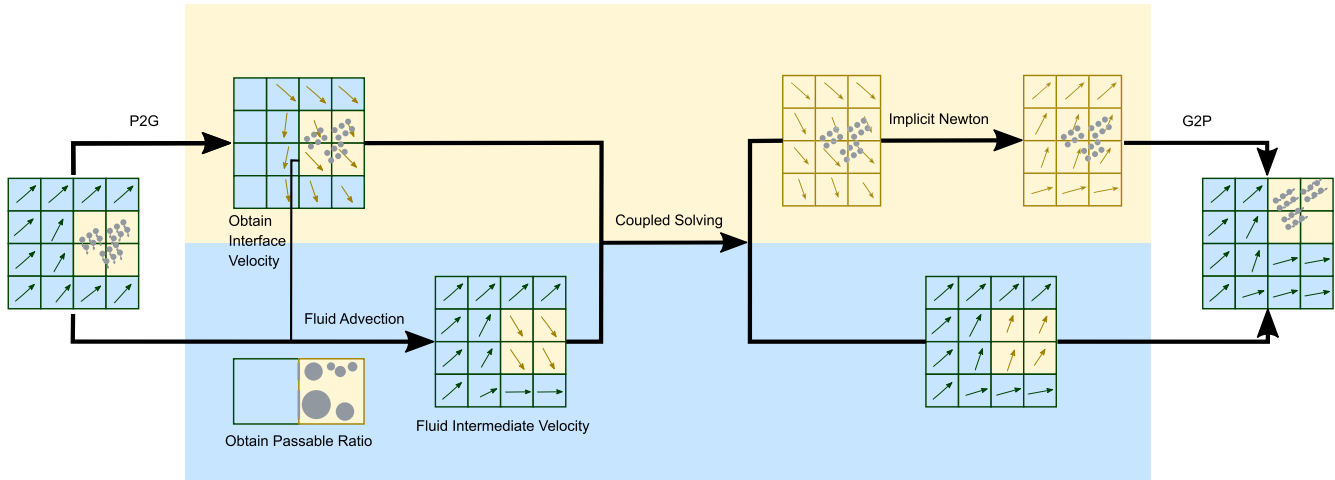


Fig. 7. Algorithm overview. Here is a schematic illustration of our method. The pipeline starts with the previously advected particles transferring their physical properties to a background grid through a P2G process. The grid is used for: (a) an implicit integration of the solids (upper row), and (b) estimating a passable ratio for more accurate flow advection (bottom row). Based on DOFs on the grid, we assemble and solve a coupled velocity-pressure system to advance these quantities to the next step. Section 2.2 presents an overview of our full algorithm in more detail.

In Equation (3), E is the fluid's enthalpy (total energy per unit volume). It relates to the internal energy e through

$$E = e + \frac{1}{2} |\mathbf{u}^f|^2 = e + \frac{1}{2} \mathbf{u}^f \cdot \mathbf{u}^f. \quad (8)$$

The system is closed by the **equation of state (EOS)** for an ideal gas [Kwatra et al. 2010]

$$p^f = (\gamma - 1) \rho^f e, \quad (9)$$

where we pick $\gamma = 1.4$.

In the remaining subsections, we split these coupled equations into advection and non-advection parts. For the reasons mentioned earlier in the introduction, we discretize the gas with an Eulerian grid and the solid with the hybrid Lagrangian/Eulerian MPM.

2.2 Overview of the Pipeline

We summarize the overall pipeline in Figure 7. Our method consists of the following procedures in each time step.

- (1) **Particle-to-grid transfer.** Solid particles transfer their physical properties \mathbf{u}^s , m^s , and J^g to the grid [Jiang et al. 2015].
- (2) **Moving interface identification.** The proper velocity and the passable ratio are obtained at each cell interface between the two phases following the technical details in Section 2.3.
- (3) **Flux based advection for the fluid.** The conserved fluid variables are updated according to Equations (10)–(12) using a second order WENO-LLF solver.
- (4) **Intermediate pressure update.** The intermediate pressures for fluid and solid phases are obtained by Equations (9) and (17), respectively.
- (5) **Coupled solve.** A coupled linear system Equation (37) is built and solved. The solid's velocity is updated using Equation (36). The fluid's conserved variables are updated using the local gradient calculation as in [Kwatra et al. 2009].

- (6) **MPM grid update.** A standard MPM-based time integration updates the solid's grid velocity values, incorporating elasticity, and wall collisions.

- (7) **Grid-to-particle transfer.** The information on the solid grid is transferred back to the solid particles. We update the particles' deformation gradients and advect their positions.

2.3 Advection

The solid advection takes place during the grid-to-particle transfer at the end of the previous time step [Jiang et al. 2016]. At the beginning of the current step, the solid particles transfer their quantities to the Eulerian grid via a standard **particle-to-grid (P2G)** operation. Subsequently, the location and the velocity of the interface between the two phases are obtained.

The fluid advection refers to the process of updating the conserved fluid variables. They are treated as independent variables, temporally ignoring the influence of the pressure. The fluid is updated to an intermediate state (\bullet^*) after the advection. Here, we use the finite difference method and the second-order WENO-LLF flux reconstruction to solve the advection equations:

$$\rho^{f,*} = \rho^{f,n} - \Delta t \nabla \cdot (\rho^f \mathbf{u}^f), \quad (10)$$

$$(\rho \mathbf{u})^{f,*} = (\rho \mathbf{u})^{f,n} - \Delta t \nabla \cdot (\rho^f \mathbf{u}^f \otimes \mathbf{u}^f), \quad (11)$$

$$(\rho E)^{f,*} = (\rho E)^{f,n} - \Delta t \nabla \cdot [(\rho E)^f \mathbf{u}^f]. \quad (12)$$

In simulations involving massive destruction, many solid fragments are scattered all over. The resulting moving interface configurations between the two phases are complex, and require specific attention to avoid errors. We discuss our approaches below.

Mixed reflective and passable interface. The P2G operation may result in cells with small solid volume fractions. The advection at the interface between such a cell and a gas cell is neither reflective nor passable. Hence, we take the following three steps to blend between purely reflective and passable interfaces. Firstly, we model



Fig. 8. Supernova. A supernova destroys multiple planets into massive amount of fragments in a mini solar system. This example contains many complex solid-fluid interfaces with a wide range of solid volume fractions. Our partially passable interface treatment robustly handles them.

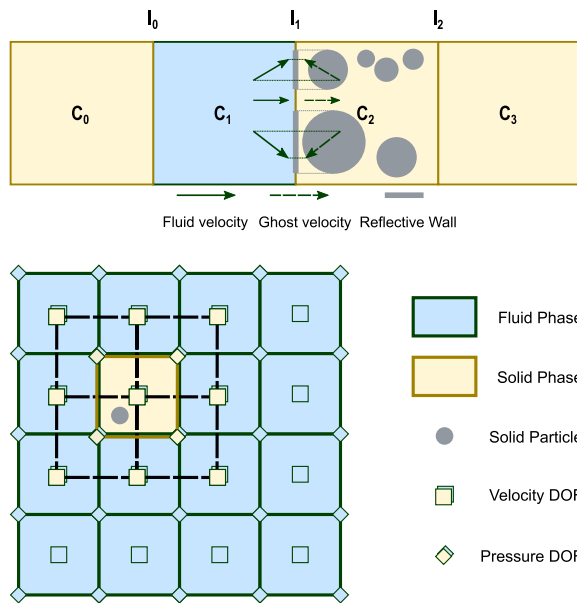


Fig. 9. (Top) An illustration of the flux blending method. The overall flux at the interface is treated as the mixture of that on a passable wall and on a reflective wall. For the passable part, the ghost velocity and the ghost conserved variables are extrapolated. For the reflective part, the ghost velocity is reflected, while the ghost conserved variables are extrapolated. (Bottom) The grid configuration. To avoid staggered velocity DOFs between the two phases, we stick to B2B1 (quadratic velocity and linear pressure) shape functions for the solid phase. The grid configuration is compatible with both B2B1 and B1B0 (linear velocity and constant pressure) shape functions for the fluid phase.

every interface as a mixture of a reflective wall and a passable passage (see Figure 3 in [Hyde and Fedkiw 2019]). The passable ratio R is derived from the cross-section of the solid's volume by treating it as a ball. Thus, we have

$$R = \min(2(V^s/\pi)^{1/2}/dx^2, 1) \text{ in 2D,}$$

$$R = \min(\pi(3V^s/(4\pi))^{2/3}/dx^2, 1) \text{ in 3D,}$$

where V^s is the solid's volume [Baumgarten and Kamrin 2019; Gao et al. 2018a]. Secondly, we extrapolate both the conserved

variables and the velocity for the passable flux stencil, and extrapolate only the conserved variables, but mirror reflect the velocity for the reflective flux stencil [Forrer and Jeltsch 1998; Forrer and Berger 1999]. Lastly, the obtained two fluxes are blended using the passable ratio. We illustrate this mixed interface I_1 in Figure 9.

Avoiding using reflection at any interface but the one being evaluated flux at. Consider an interface such as I_1 in Figure 9, there are scenarios where the adjacent gas cell C_1 is touching another solid cell C_0 . In such cases, the ghost values of the higher-order terms in the flux stencil (those in C_0) become ambiguous. In particular, it becomes a question of whether to treat the interface I_0 as a reflective interface. If we choose to do so, this ambiguity becomes more severe in multi-dimensional cases since there may exist other solid-gas interfaces in other dimensions (for example, at C_0 's top face). In such scenarios, reflecting the ghost value at different interfaces yields inconsistent results and causes simulation artifacts. To avoid this issue, we do not perform reflection at any solid-gas interfaces other than the one we are evaluating the flux at. We use the extrapolated value from the nearest gas cell corresponding to a lower-order term in the stencil for the additional required higher-order terms from solid cells.

After the advection step, we update the fluid's pressure with the EOS using the intermediate quantities (instead of using semi-Lagrangian methods as in [Kwatra et al. 2009]) to avoid the numerical drifting caused by non-conservative advection methods [Aanjanya et al. 2013].

2.4 Post-advection Equations

Projection is a common term in the simulation of incompressible flow [Bridson 2015] and refers to projecting the intermediate post-advection velocities onto a divergence-free subspace, using pressure to enforce the divergence-free constraints. In the compressible flow regime, velocity and pressure variables do not appear in the advection step as we use the conserved variables. Nonetheless, we treat velocity and pressure variables as primary states and describe the equations they must obey in the following.

2.4.1 Fluid Mass and Momentum. Since the mass equation does not explicitly contain any pressure-related terms for the

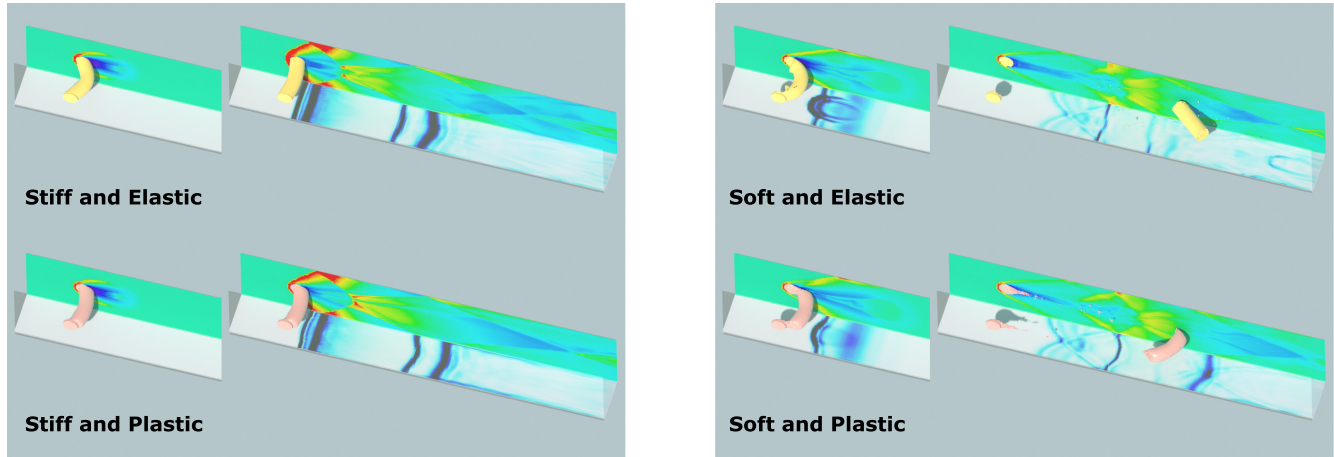


Fig. 10. Flow passing through cylinders. The schlieren (bottom) and density (side) plots for Mach 3 flow passing through (left-top) a stiff elastic cylinder, (left-bottom) a stiff plastic cylinder, (right-top) a soft elastic cylinder, and (right-bottom) a soft plastic cylinder. The stiff elastic cylinder deforms significantly near its center initially and then bounces back to exhibit a more uniform deformation. The stiff plastic cylinder deforms permanently near its two ends. The soft cylinders are quickly torn apart from the walls. The elastic fragments restore their shapes, while the plastic ones are permanently bent.

compressible flow, we assume that the intermediate state for ρ^f does not need any further correction after advection. Thus,

$$\rho^{f,n+1} = \rho^{f,*}. \quad (13)$$

With a constant density, the post-splitting momentum equation is then the same as that for incompressible fluids during the advancement from (\bullet^*) to (\bullet^{n+1}) :

$$\frac{\partial(\rho^f \mathbf{u}^f)}{\partial t} + \nabla p^f = \rho^{f,n+1} \frac{\partial \mathbf{u}^f}{\partial t} + \nabla p^f = 0. \quad (14)$$

2.4.2 Fluid Energy/pressure. The post-splitting energy equation can be transformed into a pressure diffusion equation using the primitive form of the Euler equation and by linking the pressure with the internal energy using EOS:

$$\frac{\partial p^f}{\partial t} + \rho^f c^2 \nabla \cdot \mathbf{u}^f = 0; \quad (15)$$

we refer to Fedkiw et al. [2002] for a detailed derivation.

2.4.3 Solid Ghost Pressure. Following Fang et al. [2020], we augment the hyperelastic energy density function of the solid with a quadratic volume penalization term $\Psi^g(J^g) = \frac{1}{2} \lambda^g (J^g - 1)^2$, where λ^g is Lamé's first parameter and J^g is an independent strain measure of the infinitesimal volume change ratio. Conceptually, the solid is embedded in a background ghost matrix that only resists volume change. The total energy density in the solid domain is then

$$\Psi(\mathbf{F}^s, J^g) = \Psi^s(\mathbf{F}^s) + \Psi^g(J^g), \quad (16)$$

where $\Psi^s(\mathbf{F}^s)$ is a hyperelastic energy density function such as neo-Hookean. We resolve the response of $\Psi^s(\mathbf{F}^s)$ in the MPM grid update step after the coupled solve. During the coupling step, we only consider the mechanical response of $\Psi^g(J^g)$, or equivalently, the effect of its corresponding pressure [Stomakhin et al. 2014; Fang et al. 2020]

$$p^g = -\frac{\partial \Psi^g(J^g)}{\partial J^g} = -\lambda^g (J^g - 1). \quad (17)$$

Note that p^g is the key to our coupling mechanism; it links to the solid's interface velocity and plays a vital role in enforcing the continuity requirement of the interfacial velocities. The effect of p^g on the solid's momentum is then governed by

$$\rho^{s,*} \frac{\partial \mathbf{u}^s}{\partial t} + \nabla p^g = 0. \quad (18)$$

A pressure evolution equation for p^g can be shown to be [Stomakhin et al. 2014; Gonzalez and Stuart 2008]

$$\frac{\partial p^g}{\partial t} + \lambda^g J^g \nabla \cdot \mathbf{u}^s = 0. \quad (19)$$

Equivalently, we can track J^g through

$$\frac{\partial J^g}{\partial t} = (\nabla \cdot \mathbf{u}^s) J^g \quad (20)$$

and update p^g using Equation (17). Notice the similarity in the form between Equations (14) and (18), as well as between Equations (15) and (19).

2.5 Summary of Time-discretized Equations

In summary, through operator splitting, we can conduct a post-advection coupled solve for solid-fluid coupling. The coupling step occurs after the solid's particle-to-grid transfer and the fluid's advection, but before the MPM grid update (see Figure 7). More specifically, the time-discretized equations are

(1) Fluid momentum update ($\mathbf{x} \in \Omega^f$):

$$\frac{1}{\Delta t} \rho^f \mathbf{u}^{f,n+1} (\mathbf{u}^{f,n+1} - \mathbf{u}^{f,*}) + \nabla p^{f,n+1} = 0; \quad (21)$$

(2) Fluid pressure evolution ($\mathbf{x} \in \Omega^f$):

$$\nabla \cdot \mathbf{u}^{f,n+1} + \frac{1}{\rho^{f,n} (c^2)^n \Delta t} (p^{f,n+1} - p^{f,*}) = 0; \quad (22)$$

(3) Solid momentum update ($\mathbf{x} \in \Omega^s$):

$$\frac{1}{\Delta t} \rho^{s,*} (\mathbf{u}^{s,n+1} - \mathbf{u}^{s,*}) + \nabla p^{g,n+1} = 0; \quad (23)$$

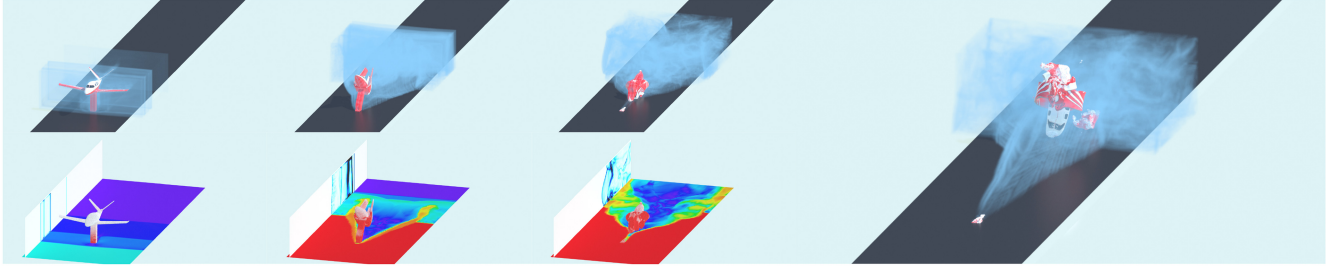


Fig. 11. Airplane in a wind tunnel. We visualize a slice of (bottom) the flow's velocity and (side) the schlieren, with the MPM particles colored based on their deviatoric stress norms. A huge lifting force deforms the wings and then the body. Later on, the airplane is torn apart, with a strong flow initiated and shooting away from its small residual fraction.

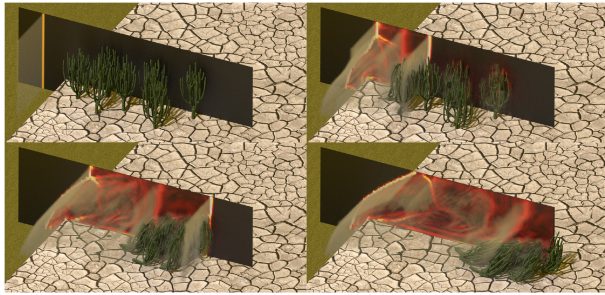


Fig. 12. Cactus forest. The cactus plants are destroyed by the high speed air flow from an explosion. The wavefront of the Mach 9 is first reflected by the staircase between the highland and the desert. It then hits the forest and breaks the plants near their roots.

(4) Solid pressure evolution ($\mathbf{x} \in \Omega^s$):

$$\nabla \cdot \mathbf{u}^{s,n+1} + \frac{1}{\lambda^g J^g, n \Delta t} (p^{g,n+1} - p^{g,*}) = 0. \quad (24)$$

Combined with the interfacial continuity Equations (6) and (7) at the solid-fluid interface Γ and the Dirichlet wall boundary conditions for \mathbf{u}^\bullet at $\partial\Omega^\bullet$, these equations form a first-order system in terms of the unknown fields $\{\mathbf{u}^{f,n+1}, p^{f,n+1}, \mathbf{u}^{s,n+1}, p^{g,n+1}\}$ and can be further spatially discretized with the finite element method.

2.6 Weak form and B-spline-Based Spatial Discretization

Using B-spline shape functions for finite elements has been recently shown effective for *incompressible* flow [Fang et al. 2020; Gagniere et al. 2020]. Here, we extend the formulation to our *compressible* fluid-structure coupled system.

For continuity of elastic forces in MPM, the solid's velocity field must be C^1 continuous [Hughes 2012; Bardenhagen and Kober 2004] (requiring the shape function to be quadratic polynomials), while the pressure can be either piecewise linear or piecewise constant. Fang et al. [2020] described two choices for discretizing coupled incompressible fluids and solids. The first scheme adopts quadratic B-splines for velocities and linear functions for pressures on both phases, resulting in a B2B1-B1-B2B1 scheme, where the B1 in the middle refers to the discretization order for the interfacial pressure. To reduce memory bandwidth and computational cost, they also described a second scheme B2B0-B0-B1B0, switching to piecewise constant pressures everywhere and piecewise linear velocities for the fluid phase.

In our case, however, if B2B0-B0-B1B0 is applied, the velocity nodes will be staggered for the two phases, and the interface will cut through the inner volume of the fluid cell. This “cut-cell” will require extra tedious treatments to the advection step in the fluid's boundary cell due to our usage of the flux-based solver.

To mitigate this problem, we stick to B2B1 for the solids and add offsets to the fluid coordinates, taking advantage of the Eulerian nature of the fluid. In most of our examples, the solid phase only occupies a small portion of the whole domain; thus, the use of B2B1 does not severely affect the performance. We refer to our discretization choice as B2B1-B1/B0-B1B0 and use it in large-scale 3D simulations. We use B2B1 for the fluid phase in our 1D and 2D benchmarks.

The weak form of our coupled system in Section 2.5 can then be obtained by properly applying test functions $\mathbf{q}^{f,s}, r^{f,s}$:

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega^f} \rho^{f,n+1} (\mathbf{u}^{f,n+1} - \mathbf{u}^{f,*}) \cdot \mathbf{q}^f d\Omega^f \\ + \int_{\Omega^f} \nabla p^{f,n+1} \cdot \mathbf{q}^f d\Omega^f = 0, \end{aligned} \quad (25)$$

$$\begin{aligned} \int_{\Omega^f} \nabla \cdot \mathbf{u}^{f,n+1} r^f d\Omega^f + \frac{1}{\rho^{f,n} (c^2)^n \Delta t} \\ \int_{\Omega^f} (p^{f,n+1} - p^{f,*}) r^f d\Omega^f = 0 \end{aligned} \quad (26)$$

$$\frac{1}{\Delta t} \int_{\Omega^s} \rho^{s,*} (\mathbf{u}^{s,n+1} - \mathbf{u}^{s,*}) \cdot \mathbf{q}_\alpha^s d\Omega^s + \int_{\Omega^s} \nabla p^{g,n+1} \cdot \mathbf{q}_\alpha^s d\Omega^s = 0, \quad (27)$$

$$\int_{\Omega^s} \nabla \cdot \mathbf{u}^{s,n+1} r^s d\Omega^s + \frac{1}{\lambda^g J^g, n \Delta t} \int_{\Omega^s} (p^{g,n+1} - p^{g,*}) r^s d\Omega^s = 0. \quad (28)$$

The weak form of the velocity conditions at the solid-fluid interface and the domain boundaries are

$$\int_{\partial\Omega^f} \mathbf{u}^{f,n+1} \cdot \mathbf{n}^f r^f dA = \int_{\partial\Omega^f} \mathbf{u}_b r^f dA, \quad (29)$$

$$\int_{\Gamma} \mathbf{u}^{f,n+1} \cdot \mathbf{n}^i r^f dA = \int_{\Gamma} \mathbf{u}^{s,n+1} \cdot \mathbf{n}^i r^s dA, \quad (30)$$

$$\int_{\partial\Omega^s} \mathbf{u}^{s,n+1} \cdot \mathbf{n}^s r^s dA = \int_{\partial\Omega^s} \mathbf{u}_b r^s dA. \quad (31)$$

Here, \mathbf{q}_α collocates with the velocity nodes and is nonzero only on its α component, Γ is the solid-fluid interface, p is the pressure, \mathbf{u}_b is the normal velocity at the slip boundaries of the solid or fluid

domains, and r is the test function collocated with the pressure nodes. For B2B1, we would have

$$\mathbf{q}_{\alpha,i}(x, \mathbf{x}_u) = \delta_{\alpha,i} N^2(x - x_u), \quad (32)$$

$$r(x, x_p) = N^1(x - x_p), \quad (33)$$

where N^k is the k th order B-spline kernel and δ is the Kronecker delta. A discrete and compact form of the above equations can be obtained through the standard FEM assembling process, which yields

$$\begin{bmatrix} \frac{\mathbf{M}^f}{\Delta t} & \mathbf{G}^f & \mathbf{B}^f & \mathbf{H}^f & 0 & 0 & 0 \\ \mathbf{G}^{f,T} & -\frac{\mathbf{S}^{f,-1}}{\Delta t} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{B}^{f,T} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{H}^{f,T} & 0 & 0 & 0 & \mathbf{H}^{s,T} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{H}^s & \frac{\mathbf{M}^s}{\Delta t} & \mathbf{G}^s & \mathbf{B}^s \\ 0 & 0 & 0 & 0 & \mathbf{G}^{s,T} & -\frac{\mathbf{S}^{s,-1}}{\Delta t} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{B}^{s,T} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\alpha}^{f,n+1} \\ \mathbf{P}^{f,n+1} \\ \mathbf{Y}^{f,n+1} \\ \mathbf{h}^{n+1} \\ \mathbf{U}_{\alpha}^{s,n+1} \\ \mathbf{P}^{g,n+1} \\ \mathbf{Y}^{s,n+1} \end{bmatrix} = rhs, \quad (34)$$

where the \mathbf{h} and $\mathbf{Y}^{f,s}$ are the stacked vectors of the pressure DOFs at the solid-fluid interface and the domain boundaries, respectively, which appear because of the integration by parts of $\int_{\Omega} \nabla p^{n+1} \cdot \mathbf{q}_{\alpha} d\Omega$. The right hand side is

$$rhs = \left(\frac{\mathbf{M}^f}{\Delta t} \mathbf{U}_{\alpha}^{f,*}, -\frac{\mathbf{S}^{f,-1}}{\Delta t} \mathbf{P}^{f,*}, b^f, 0, \frac{\mathbf{M}^s}{\Delta t} \mathbf{U}_{\alpha}^{s,*}, -\frac{\mathbf{S}^{s,-1}}{\Delta t} \mathbf{P}^{s,*}, b^s \right)^T.$$

2.7 Building an SPD System

Equation (34) as a KKT system is challenging to solve. We transform it into an SPD system by mass lumping and eliminating the velocity DOFs. The mass matrix $\mathbf{M}^{f,s}$ and the stiffness matrix $\mathbf{S}^{f,s}$ are not ill-conditioned. Nevertheless, the lumping makes them diagonal-scaling matrices and easily invertible. We eliminate velocity DOFs using

$$\mathbf{U}_{\alpha}^{f,n+1} = \mathbf{U}_{\alpha}^{f,*} - \Delta t \mathbf{M}^{f,-1} (\mathbf{G}^f \mathbf{P}^{f,n+1} + \mathbf{B}^f \mathbf{Y}^{f,n+1} + \mathbf{H}^f \mathbf{h}), \quad (35)$$

$$\mathbf{U}_{\alpha}^{s,n+1} = \mathbf{U}_{\alpha}^{s,*} - \Delta t \mathbf{M}^{s,-1} (\mathbf{G}^s \mathbf{P}^{s,n+1} + \mathbf{B}^s \mathbf{Y}^{s,n+1} + \mathbf{H}^s \mathbf{h}). \quad (36)$$

This yields an SPD and well-conditioned pressure-only system:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & 0 & 0 \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} & \mathbf{A}_{23} & 0 & 0 \\ \mathbf{A}_{13}^T & \mathbf{A}_{23}^T & \mathbf{A}_{33} & \mathbf{A}_{34} & \mathbf{A}_{35} \\ 0 & 0 & \mathbf{A}_{34}^T & \mathbf{A}_{44} & \mathbf{A}_{45} \\ 0 & 0 & \mathbf{A}_{35}^T & \mathbf{A}_{45}^T & \mathbf{A}_{55} \end{bmatrix} \begin{bmatrix} \mathbf{P}^{f,n+1} \\ \mathbf{Y}^{f,n+1} \\ \mathbf{h}^{n+1} \\ \mathbf{Y}^{s,n+1} \\ \mathbf{P}^{s,n+1} \end{bmatrix} = rhs. \quad (37)$$

Here,

$$\begin{aligned} \mathbf{A}_{11} &= \mathbf{S}^{f,-1} + \Delta t^2 \mathbf{G}^{f,T} \mathbf{M}^{f,-1} \mathbf{G}^f, \\ \mathbf{A}_{12} &= \Delta t^2 \mathbf{G}^{f,T} \mathbf{M}^{f,-1} \mathbf{B}^f, \quad \mathbf{A}_{13} = \Delta t^2 \mathbf{G}^{f,T} \mathbf{M}^{f,-1} \mathbf{H}^f, \\ \mathbf{A}_{22} &= \Delta t^2 \mathbf{B}^{f,T} \mathbf{M}^{f,-1} \mathbf{B}^f, \quad \mathbf{A}_{23} = \Delta t^2 \mathbf{B}^{f,T} \mathbf{M}^{f,-1} \mathbf{H}^f, \\ \mathbf{A}_{33} &= \Delta t^2 \mathbf{H}^{f,T} \mathbf{M}^{f,-1} \mathbf{H}^f + \Delta t^2 \mathbf{H}^{s,T} \mathbf{M}^{s,-1} \mathbf{H}^s, \\ \mathbf{A}_{34} &= \Delta t^2 \mathbf{H}^{s,T} \mathbf{M}^{s,-1} \mathbf{B}^s, \quad \mathbf{A}_{35} = \Delta t^2 \mathbf{H}^{s,T} \mathbf{M}^{s,-1} \mathbf{G}^s, \\ \mathbf{A}_{44} &= \Delta t^2 \mathbf{B}^{s,T} \mathbf{M}^{s,-1} \mathbf{B}^s, \quad \mathbf{A}_{45} = \Delta t^2 \mathbf{B}^{s,T} \mathbf{M}^{s,-1} \mathbf{G}^s, \\ \mathbf{A}_{55} &= \mathbf{S}^{s,-1} + \Delta t^2 \mathbf{G}^{s,T} \mathbf{M}^{s,-1} \mathbf{G}^s, \end{aligned}$$

$$\text{and } rhs = (\mathbf{S}^{f,-1} \mathbf{P}^{f,*} + \Delta t \mathbf{G}^{f,T} \mathbf{U}_{\alpha}^{f,*}, \Delta t \mathbf{B}^{f,T} \mathbf{U}_{\alpha}^{f,*} - \Delta t b^f, \Delta t \mathbf{H}^{f,T} \mathbf{U}_{\alpha}^{f,*} + \Delta t \mathbf{H}^{s,T} \mathbf{U}_{\alpha}^{s,*}, \Delta t \mathbf{B}^{s,T} \mathbf{U}_{\alpha}^{s,*} - \Delta t b^s, \mathbf{S}^{s,-1} \mathbf{P}^{s,*} +$$

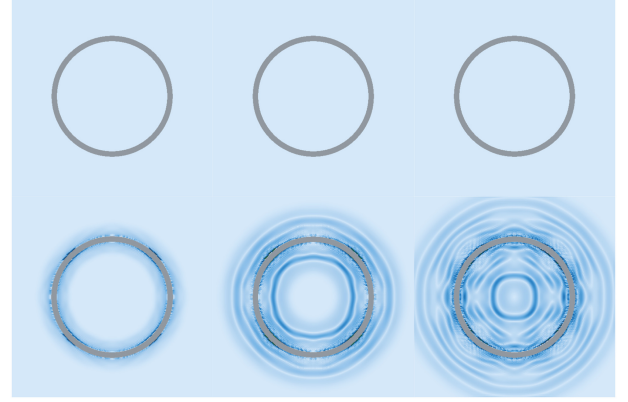


Fig. 13. Comparison between two methods for updating the solid's velocities with pressure. We put a circular shell in the quiescent air with uniform pressure. We follow the method in [Kwatra et al. 2009] for updating the fluid states. For the solid, (top) adopting Equation (36) yields a stable solution; (bottom) using the local pressure gradient causes instability.

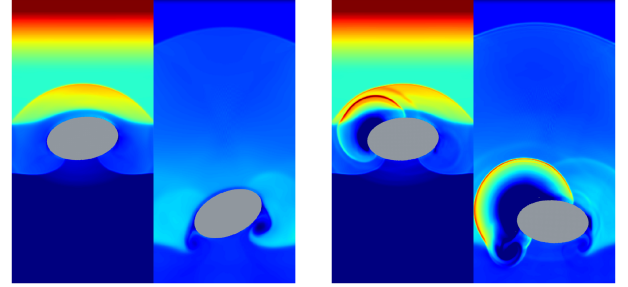


Fig. 14. Blasting an ellipse. We blast an ellipse with a biased mass center with two different ordering choices in our pipeline: putting the coupling step before (left) or after (right) the MPM grid update step. Putting the MPM step before the coupling step causes (incorrect) artificial shocks.

$\Delta t \mathbf{G}^{s,T} \mathbf{U}_{\alpha}^{s,*})^T$. This system is diagonal dominant and can be solved efficiently with Jacobi-preconditioned **Conjugate Gradient (CG)**. It usually takes less than 80 CG iterations to converge for our examples.

After solving the linear system, we use Equation (36) to update the solid's velocities and use local gradients to update the conserved variables of the fluid [Kwatra et al. 2009]. We found in our experiments that updating the solid's velocities with local gradients sometimes causes instability, possibly owing to the nature of the quadratic order of the velocity discretization (see Figure 13 for a visual comparison). On the other hand, we cannot use Equation (35) to update the fluid's velocities because otherwise we cannot update the total energy properly.

2.8 MPM Step

The velocities on the solid cells after the coupling step have taken the contributions from the fluid and require a final standard MPM grid update step to incorporate hyperelasticity. Note that Fang et al. [2020] adopted a different order and put the MPM step before the coupling step. Doing that, however, results in instabilities in our framework when a strong shock interacts with stiff materials (see Figure 14 for a visual comparison).

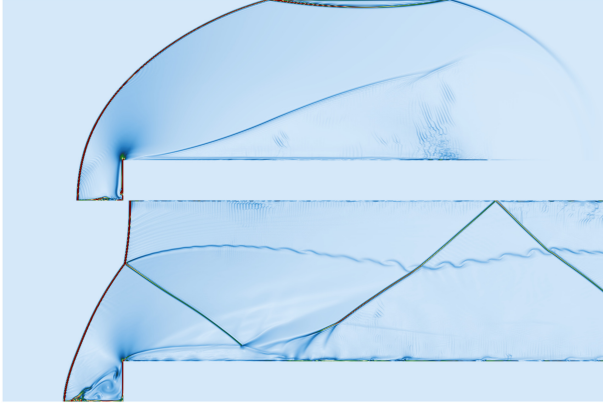


Fig. 15. Stair flow. We show the schlieren plot at (top-to-bottom) 0.2s and 4.0s. The left boundary is set to be a constant Mach 3 flow; the right boundary is set to be the outlet; and the top and the bottom boundaries are set as inviscid walls. The domain is initially filled with the Mach 3 flow. Our solver accurately captures wall reflections and correctly models the inlet/outlet flow conditions.

2.9 Inlet and Outlet Treatment

Traditional one-step approaches for simulating compressible flow use the characteristic decomposition to handle inlet and outlet boundary conditions. In time-splitting approaches, Kwatra et al. [2009] elaborated on the stencil modification for calculating boundary fluxes in the advection step. We need to properly model the Neumann boundary condition in our coupled system specifying the pressure gradient in Equation (14). Equivalently we specify the final t^{n+1} velocities at the boundary faces and modify the right hand side of Equation (37). The user specifies the inlet’s velocity values directly, and the outlet takes the intermediate velocity values after advection.

In summary, to model inlets and outlets, we

- (1) extrapolate and attenuate the conserved variables in ghost cells as in [Kwatra et al. 2009], and calculate the fluxes for the advection;
- (2) obtain the intermediate velocity for the interior cells and one layer of the ghost cells from their conserved variables after the advection, and then interpolate and store the intermediate transport velocity at the boundary faces for the outlets while directly storing the desired value at the inlet; and
- (3) enforce velocity boundary conditions in the coupled system through contributing to boundary integrals in Equation (29).

3 RESULTS

3.1 Validations

1D validation. We validate the convergence behavior of our method following the 1D cases from [Kwatra et al. 2009]. We generate high resolution (4,096 cells) results using their solver and measure the squared density deviation as

$$e_{\Delta m} = \frac{\sum(\rho - \rho_b)^2}{(\sum \rho_b)^2}. \quad (38)$$

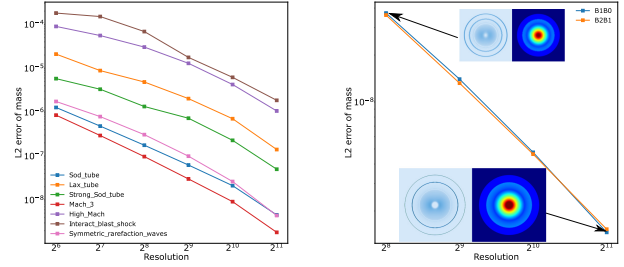


Fig. 16. (Left) The L2 mass error plot of the 1D validations. (Right) The L2 mass error plot of the 2D central circular shock test comparing B2B1 and B1B0 schemes.

The results are plotted in Figure 16, showing that our method has excellent convergence behaviors for relatively higher speed flows, while at lower speeds the convergence order is 1.

2D validation. We use the central circular shock test from [Kwatra et al. 2009] as a 2D validation. The results at varying resolutions (ranging from 256 to 2048) are compared against a 4,096-resolution benchmark obtained by their method. We also use this test to perform an ablation study comparing the accuracy between B2B1 and B1B0 velocity/pressure orders. The results (see Figure 16) indicate that B2B1 and B1B0 both converge under refinement, showing unnoticeable differences in the visual results. We thus adopt B1B0 in our large-scale examples for its lower computational cost.

Boundary treatment. We validate our treatment for inlet and outlet boundaries using the classic stair flow test from computational fluid dynamics. The result in Figure 15 is visually plausible.

Fluid-solid interaction. We validate the qualitative accuracy of our fluid-solid coupling scheme using the classic lifting cylinder test in 2D. The cylinder is sampled with uniformly distributed material points with a high Young’s modulus. Our result is shown in Figure 17. The cylinder is lifted to the top of the channel toward the outlet, reproducing the result in [Forrer and Berger 1999].

3.2 Other Examples

We show more examples “(refer to Table 1 for detailed parameters)” to demonstrate the efficacy of our method with an emphasis on the natural advantage of MPM’s handling of large deformation and topology change.

Car driving through a land mine. We simulate a car driving through a triggered landmine in Figure 1. The car’s main body is plastic, while its wheels are made of much lighter elastic materials. The intense explosion tears apart and blows away the rear wheels and causes permanent distortion to the body.

“Tribunny”. We simulate three bunny toys surrounding an explosion in Figure 2. The heavy elastic bunny and the plastic bunny are damaged without being pushed far away. On the other hand, the lighter elastic bunny is fully blown away by the shockwave, as expected.

Elastic shell explosion. We simulate an exploding elastic shell in Figure 3. The shell is filled with highly compressed hot gas. The gas expands and breaks the shell. The process first results in two layers

Table 1. Simulation Parameters

Example	Grid resolution	MPM particles(K)	Young's modulus	Possion's ratio	Yeild stress
(Figure 1) Car driving through a land mine	128*256*256	1300	body/tire: $2 \times 10^4/1.5 \times 10^3$	0.22	body: 600
(Figure 2) Tribunny	128*64*128	600	SE/WE/iron: $1600/200/2 \times 10^4$	0.22	iron: 600
(Figure 3) Elastic Shell Explosion 2D	1024*1024	15.8	100	0.3	N/A
(Figure 4) Firing Cannon at a truck	256*128*128	450	S/W: $1 \times 10^8/5 \times 10^7$	S/W: 0.27/0.334	S/W: $1.7 \times 10^6/5.2 \times 10^5$
(Figure 5) Firing Bullet 2D	1600*400	1.3	1×10^6	0.25	N/A
(Figure 6) Pumpkin explosion	128*128*128	120	E/P: $1.5 \times 10^3/2 \times 10^4$	0.25	P: 600
(Figure 8) Supernova	400*80*80	900	1×10^6	0.25	2.6×10^3
(Figure 10) Cylinder flow and von Kármán vortices	200*40*24	80	SE/WE/SP/WP: $1 \times 10^3/2 \times 10^2/1 \times 10^4/5 \times 10^3$	0.35	SP/WP: 150/75
(Figure 11) Airplane in a wind tunnel	400*60*160	98	1×10^8	0.25	2.6×10^5
(Figure 12) Cactus near explosion	300*100*75	430	1×10^6	0.25	6×10^4
(Figure 19) Mach Diamond and Koalas	256*128*64	174	1×10^3	0.22	N/A

S: strong / W: weak / E: elastic / P: plastic / SE: strong elastic / WE: weak elastic / SP: strong plastic / WP: weak plastic

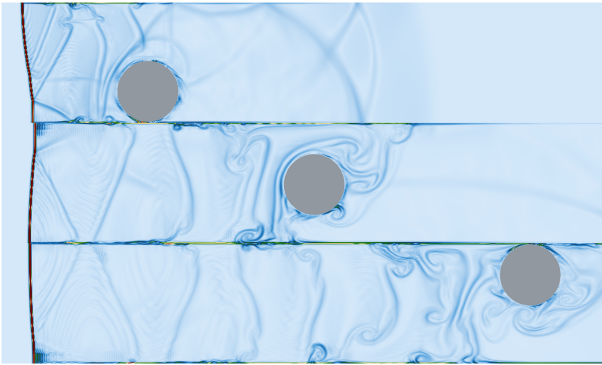


Fig. 17. Lifting cylinder. We show the schlieren plot at (top-to-bottom) 0.18s, 0.53s, and 0.89s. The leftmost 8% of the domain is set as a constant Mach 3 flow; the right side of the domain is set as the outlet with a denser but slower flow; and the top and the bottom are set as inviscid walls. Our method accurately captures the lifting of the cylinder to the top of the channel near the outlet.

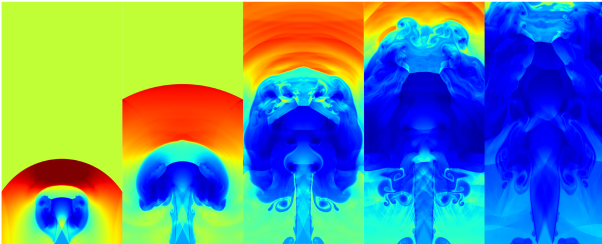


Fig. 18. Mach diamond. We show the density plot of a Mach diamond in 2D at 0.22s, 0.44s, 0.85s, 1.11s, and 1.39s. The central 1/8 of the bottom is set to be the inlet of Mach 3 with 50% of the ambient pressure.

of primary shockwaves spreading outwards. Later on, it triggers an inward injection of the gas and the mixing of the post-explosion exhaust. The elastic fragments naturally interact with the complex flow pattern.

Firing Cannon at a truck. We simulate projectiles hitting truck toys in Figure 4. The trucks are first damaged by the projectiles and then deformed more by the shockwaves. The material parameters of each truck determine the severity of its final damage.

Firing bullets. We use high-temperature high-pressure gas to fire bullets at varying conditions in Figure 5. A heavy bullet reaches the transonic state. It does not break the wavefront until near the end of the simulation and creates uniform vortices behind its tail. The lighter bullet reaches the supersonic state. It breaks the wavefront early in the simulation and produces massive chaotic vortices. In another test, we add a silencer at the end of the firing chamber to observe its successful suppression of the primary shock. This bullet has its speed slightly decreased, and it moves forward, passing the much slower shockwave and slicing out a secondary shock.

Pumpkin explosion. We simulate the explosions of a plastic pumpkin and an elastic pumpkin by setting up ellipsoidal high-pressure gas inside them (Figure 6). They are both severely torn apart. The deformation of the plastic pumpkin is permanent, while the elastic pumpkin jiggles vividly in the back-flow.

Supernova. We use the planar wave to mimic a far-field supernova explosion, destroying a mini solar system in Figure 8. Due to the plastic nature of the solids in this example, they are easily blown into dust.

Cylinder flow and von Kármán vortices. We simulate the interactions between March 3 flow and cylinders made of different elastoplastic materials. Varying the Young's modulus and switching plasticity on and off, we observe intricate fluid-solid coupling behaviours and distinct patterns of von Kármán vortices; see Figure 10.

Airplane in a wind tunnel. We put a toy airplane in a wind tunnel and gradually increase the flow speed until fractures occur; see Figure 11. The loss of balance causes the plane to spin around in the tunnel. The plane is severely deformed and fractured during its dynamic interaction with the supersonic flow.

Cactus near explosion. We simulate the interaction between a cactus forest and a supersonic wind blow caused by a far-field explosion in Figure 12. The cactus plants are cut off near their roots as the shock approaches.

Mach Diamond and Koalas. The Mach diamond appears at the exhaust of a slightly over-expanded jet with an outlet pressure a bit lower than the ambient. The jet is compressed and expands periodically to form the shape of diamonds (shown in Figure 18). In a

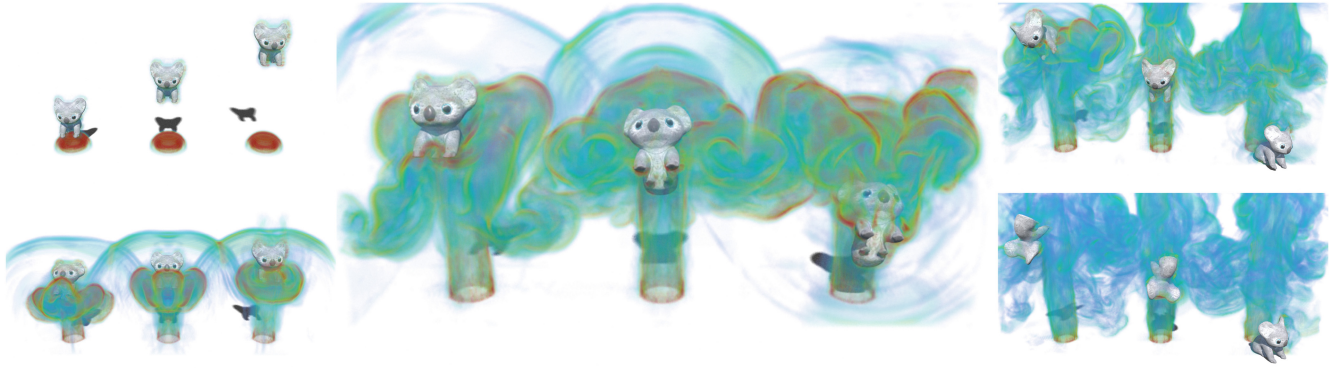


Fig. 19. Three koala toys hit by Mach diamond jets. The inlet setting mimics the 2D version in Figure 18. The density ratio (from left to right) of the koalas is 1:4:9. The lightest koala is easily lifted up; the middle one stays balanced under the effects of the pumping jet and gravity; and the heaviest koala easily breaks the wavefront of the jet.

3D simulation (Figure 19), we put elastic koala toys with different densities above air jets. We pick the densities of the koalas so that the lightest koala is easily blown up, the heaviest one breaks the wavefront, and the middle one stays balanced for a while before slipping off the exhaust area.

4 DISCUSSION

Extending the time-splitting scheme for compressible flow and the Material Point Method for solids, we have developed a hybrid Eulerian and Lagrangian/Eulerian method for strongly coupled flow-structure interaction. Our unified framework robustly and efficiently simulates shock wave propagation, fluid-solid coupling, solid fracturing, and temperature/density-induced buoyancy effects.

4.1 Limitations

Our method contains several limitations. Firstly, even though our mixed grid treatment increases subgrid resolution to some extent, it cannot account for detailed sub-grid solid geometry. For example, our method cannot capture the accurate reflection at a sub-grid non-axis-aligned solid surface. Cut cell methods may improve the accuracy here but will also bring in more challenges.

Secondly, the computational cost of our method imposes restrictions on the practical resolution we can achieve. The performance bottleneck of our method is the construction of the linear system due to its requirement of performing many sparse matrix multiplications. In our implementation, we accelerate this part based on pre-obtained knowledge of the sparsity pattern. Our optimization strategy cannot be easily generalized to more sophisticated future extensions using cut-cells or adaptive meshes.

A third limitation comes from the shortcomings of operator splitting. In fluid-structure interactions, the velocity at the interface greatly impacts the flow pattern. Like almost every other method using operator splitting, our method does not maintain the same interface velocity at different stages of the algorithm. Our work does not cover the impact on the accuracy of such inconsistency.

Lastly, we simplify the treatment of the inlet/outlet boundary conditions by directly specifying the desired velocities. A more accurate treatment of them in a time-splitting framework neces-

itates more rigorous mathematical descriptions of the coupling phase from the characteristic decomposition of the flux terms.

4.2 Future Work

In this work, we rely on MPM's artificial numerical fracture for simulating the fragmentation of solids. It is an exciting future work to incorporate continuum damage mechanics [Wolper et al. 2019, 2020] into the MPM component of our framework to model more versatile brittle and ductile fracture mechanics.

It would also be interesting to couple shockwaves with incompressible liquids, granular media, and soil sediments. For powder-like solid particles, our mixed reflective and passable interface scheme in the advection step may no longer fully capture the turbulent sub-grid details; thus, new schemes are required.

From the performance perspective, we observe that while the **Adaptive Mesh Refinement (AMR)** schemes are well developed in the traditional characteristic-based compressible flow solvers, they are not broadly studied in time-splitting methods. It is thus meaningful to design a spatially adaptive treatment for the coupled phases, extending Gao et al. [2017].

As mentioned in limitations, the performance of our solver is severely bottlenecked by the linear system's construction. It is a promising future work to devise matrix-free solvers with multi-grid preconditioners [Wang et al. 2020a] and mitigate the solver to modern computational platforms following [Gao et al. 2018; Wang et al. 2020b].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments.

REFERENCES

- M. Aanjaneya, S. Patkar, and R. Fedkiw. 2013. A monolithic mass tracking formulation for bubbles in incompressible flow. *Journal of Computational Physics* 247 (2013), 17–61.
- S. Bailoor, A. Annangi, J. H. Seo, and R. Bhardwaj. 2017. Fluid-structure interaction solver for compressible flows with applications to blast loading on thin elastic structures. *Applied Mathematical Modelling* 52 (2017), 470–492. DOI: <https://doi.org/10.1016/j.apm.2017.05.038>
- J. W. Banks, W. D. Henshaw, A. K. Kapila, and D. W. Schwendeman. 2016. An added-mass partition algorithm for fluid-structure interactions of compressible fluids and nonlinear solids. *Journal of Computational Physics* 305 (2016), 1037–1064.
- S. G. Bardenhagen and E. M. Kober. 2004. The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences* 5, 6 (2004), 477–496.

- C. Batty, F. Bertails, and R. Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics* 26, 3 (2007), 100–es.
- A. Baumgarten and K. Kamrin. 2019. A general fluid–sediment mixture model and constitutive theory validated in many flow regimes. *Journal of Fluid Mechanics* 861 (2019), 721–764.
- A. S. Baumgarten, B. L. S. Couchman, and K. Kamrin. 2021. A coupled finite volume and material point method for two-phase simulation of liquid–sediment and gas–sediment flows. *Computer Methods in Applied Mechanics and Engineering* 384, 1 (2021), 113940. DOI: <https://doi.org/10.1016/j.cma.2021.113940>
- J. Brackbill, D. Kothe, and H. Ruppel. 1988. FLIP: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications* 48, 1 (1988), 25–38.
- R. Bridson. 2015. *Fluid Simulation for Computer Graphics*. CRC press.
- Alexandre Joel Chorin. 1967. The numerical solution of the Navier-Stokes equations for an incompressible fluid. *Bulletin of the American Mathematical Society* 73, 6 (1967), 928–931.
- H. Deconinck, P. L. Roe, and R. Struijs. 1993. A multidimensional generalization of Roe’s flux difference splitter for the Euler equations. *Computers & Fluids* 22, 2–3 (1993), 215–222.
- Y. Fang, Z. Qu, M. Li, X. Zhang, Y. Zhu, M. Aanjaneya, and C. 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), 51:1–51:16.
- R. Fedkiw, X. Liu, and S. Osher. 2002. A general technique for eliminating spurious oscillations in conservative schemes for multiphase and multispecies Euler equations. *International Journal of Nonlinear Sciences and Numerical Simulation* 3, 2 (2002), 99–106.
- B. E. Feldman, J. F. O’Brien, and O. Arikan. 2003. Animating suspended particle explosions. In *Proceedings of the ACM SIGGRAPH 2003 Papers*. 708–715.
- M. Fey. 1998. Multidimensional upwinding. Part II. decomposition of the euler equations into advection equations. *Journal of Computational Physics* 143, 1 (1998), 181–199.
- H. Forrer and M. Berger. 1999. Flow simulations on Cartesian grids involving complex moving geometries. In *Proceedings of the Hyperbolic Problems: Theory, Numerics, Applications*. Springer, 315–324.
- H. Forrer and R. Jeltsch. 1998. A higher-order boundary treatment for Cartesian-grid methods. *Journal of Computational Physics* 140, 2 (1998), 259–277.
- C. Fu, Q. Guo, T. Gast, C. Jiang, and J. Teran. 2017. A polynomial particle-in-cell method. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–12.
- S. Gagniere, D. Hyde, A. Marquez-Razon, C. Jiang, Z. Ge, X. Han, Q. Guo, and J. Teran. 2020. A hybrid lagrangian/eulerian collocated velocity advection and projection method for fluid simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2020*, Vol. 39.
- M. Gao, A. Pradhana, X. Han, Q. Guo, G. Kot, E. Sifakis, and C. Jiang. 2018a. Animating fluid sediment mixture in particle-laden flows. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.
- Ming Gao, Andre Pradhana Tampubolon, Chenfanfu Jiang, and Eftychios Sifakis. 2017. An adaptive generalized interpolation material point method for simulating elastoplastic materials. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–12.
- Ming Gao, Xinlei Wang, Kui Wu, Andre Pradhana, Eftychios Sifakis, Cem Yuksel, and Chenfanfu Jiang. 2018. GPU optimization of material point methods. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12.
- O. Gonzalez and A. M. Stuart. 2008. *A First Course in Continuum Mechanics*. Cambridge University Press.
- J. T. Grétarsson and R. Fedkiw. 2013. Fully conservative leak-proof treatment of thin solid structures immersed in compressible fluids. *Journal of Computational Physics* 245, 8 (2013), 160–204.
- J. T. Grétarsson, N. Kwatra, and R. Fedkiw. 2011. Numerically stable fluid–structure interactions between compressible flow and solid structures. *Journal of Computational Physics* 230, 8 (2011), 3062–3084.
- Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics* 37, 4, Article 150 (2018), 14 pages. DOI: <https://doi.org/10.1145/3197517.3201293>
- T. J. Hughes. 2012. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation.
- D. A. B. Hyde and R. Fedkiw. 2019. A unified approach to monolithic solid-fluid coupling of sub-grid and more resolved solids. *ACM Transactions on Graphics* 390 (2019), 490–526. DOI: <https://doi.org/10.1016/j.jcp.2019.03.049>
- C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.
- C. Jiang, C. Schroeder, and J. Teran. 2017. An angular momentum conserving affine-particle-in-cell method. *Journal of Computational Physics* 338 (2017), 137–164.
- C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle. 2016. The material point method for simulating continuum materials. In *Proceedings of the ACM SIGGRAPH 2016 Course*. 24:1–24:52.
- G. Kawada and T. Kanai. 2011. Procedural fluid modeling of explosion phenomena based on physical properties. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 167–176.
- B. Kim, Y. Liu, I. Llamas, and J. R. Rossignac. 2005. *Flowfixer: Using Bfccc for Fluid Simulation*. Technical Report. Georgia Institute of Technology.
- A. Kosik, M. Feistauer, M. Hadrava, and J. Horáček. 2015. Numerical simulation of the interaction between a nonlinear elastic structure and compressible flow by the discontinuous Galerkin method. *Applied Mathematics and Computation* 267 (2015), 382–396.
- N. Kwatra, J. Grétarsson, and R. Fedkiw. 2010. Practical animation of compressible flow for ShockWaves and related phenomena. In *Proceedings of the Symposium on Computer Animation*. 207–215.
- N. Kwatra, J. Su, J. T. Grétarsson, and R. Fedkiw. 2009. A method for avoiding the acoustic time step restriction in compressible flow. *Journal of Computational Physics* 228, 11 (2009), 4146–4161.
- Z. Li and J. Favier. 2017. A non-staggered coupling of finite element and lattice Boltzmann methods via an immersed boundary scheme for fluid-structure interaction. *Computers and Fluids* 143 (2017), 90–102.
- X. Liu, S. Osher, and T. Chan. 1994. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics* 115, 1 (1994), 200–212.
- S. Ma, X. Zhang, Y. Lian, and X. Zhou. 2009. Simulation of high explosive explosion using adaptive material point method. *Cmes-computer Modeling in Engineering and Sciences* 39, 2 (2009), 101–124.
- L. Monasse, V. Daru, C. Mariotti, S. Piperno, and C. Tenaud. 2012. A conservative coupling algorithm between a compressible flow and a rigid body using an Embedded Boundary method. *Journal of Computational Physics* 231, 7 (2012), 2977–2994. DOI: <https://doi.org/10.1016/j.jcp.2012.01.002>
- V. Pasquariello, G. Hammerl, F. Örley, S. Hickel, C. Danowski, A. Popp, W. A. Wall, and N. A. Adams. 2016. A cut-cell finite volume – finite element coupling approach for fluid–structure interaction in compressible flow. *Journal of Computational Physics* 307 (2016), 670–695. DOI: <https://doi.org/10.1016/j.jcp.2015.12.013>
- Z. Qu, X. Zhang, M. Gao, C. Jiang, and B. Chen. 2019. Efficient and conservative fluids using bidirectional mapping. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.
- R. Setaluri, M. Aanjaneya, S. Bauer, and E. Sifakis. 2014. SPGrid: A sparse paged grid structure applied to adaptive smoke simulation. *ACM Transactions on Graphics* 33, 6, Article 205 (2014), 12 pages. DOI: <https://doi.org/10.1145/2661229.2661269>
- J. Sewall, N. Galoppo, G. Tsankov, and M. Lin. 2009. Visual simulation of shockwaves. *Graphical Models* 71, 4 (2009), 126–138.
- C. Shu and S. Osher. 1988. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics* 77, 2 (1988), 439–471.
- E. A. Spiegel and G. Veronis. 1960. On the Boussinesq approximation for a compressible fluid. *The Astrophysical Journal* 131 (1960), 442.
- J. Stam. 1999. Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. 121–128.
- A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. 2013. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.
- A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.
- D. Sulsky, S. Zhou, and H. L. Schreyer. 1995. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 87, 1–2 (1995), 236–252.
- D. Takeshita, S. Ota, M. Tamura, T. Fujimoto, K. Muraoka, and N. Chiba. 2003. Particle-based visual simulation of explosive flames. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*. IEEE, 482–486.
- L. Wang, G. M. D. Currao, F. Han, A. J. Neely, J. Young, and F. Tian. 2017. An immersed boundary method for fluid–structure interaction with compressible multiphase flows. *Journal of Computational Physics* 346 (2017), 131–151. DOI: <https://doi.org/10.1016/j.jcp.2017.06.008>
- X. Wang, M. Li, Y. Fang, X. Zhang, M. Gao, M. Tang, D. M. Kaufman, and C. Jiang. 2020a. Hierarchical optimization time integration for cfl-rate mpm stepping. *ACM Transactions on Graphics (TOG)* 39, 3 (2020), 1–16.
- Xinlei Wang, Yuxing Qiu, Stuart R. Slatery, Yu Fang, Minchen Li, Song-Chun Zhu, Yixin Zhu, Min Tang, Dinesh Manocha, and Chenfanfu Jiang. 2020b. A massively parallel and scalable multi-GPU material point method. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 30–1.
- Joshua Wolper, Yu Fang, Minchen Li, Jiecong Lu, Ming Gao, and Chenfanfu Jiang. 2019. CD-MPM: Continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.
- Joshua Wolper, Yunuo Chen, Minchen Li, Yu Fang, Ziyin Qu, Jiecong Lu, Meggie Cheng, and Chenfanfu Jiang. 2020. AnisoMPM: Animating anisotropic damage mechanics. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 37–1.
- K. Wu, N. Truong, C. Yuksel, and R. Hoetzlein. 2018. Fast fluid simulations with sparse volumes on the GPU. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2018)* 37, 2 (2018), 157–167. DOI: <https://doi.org/10.1111/cgf.13350>
- X. Zhang, Z. Chen, and Y. Liu. 2016. *The Material Point Method: A Continuum-based Particle Method for Extreme Loading Cases*. Academic Press.

Received October 2021; revised February 2022; accepted February 2022