A Hybrid Lagrangian/Eulerian Collocated Velocity Advection and Projection Method for Fluid Simulation

S. Gagniere[1], D. Hyde[1], A. Marquez-Razon[1], C. Jiang[2], Z. Ge[1], X. Han[1], Q. Guo[1] and J. Teran[1]

[1]UCLA
[2]University of Pennsylvania
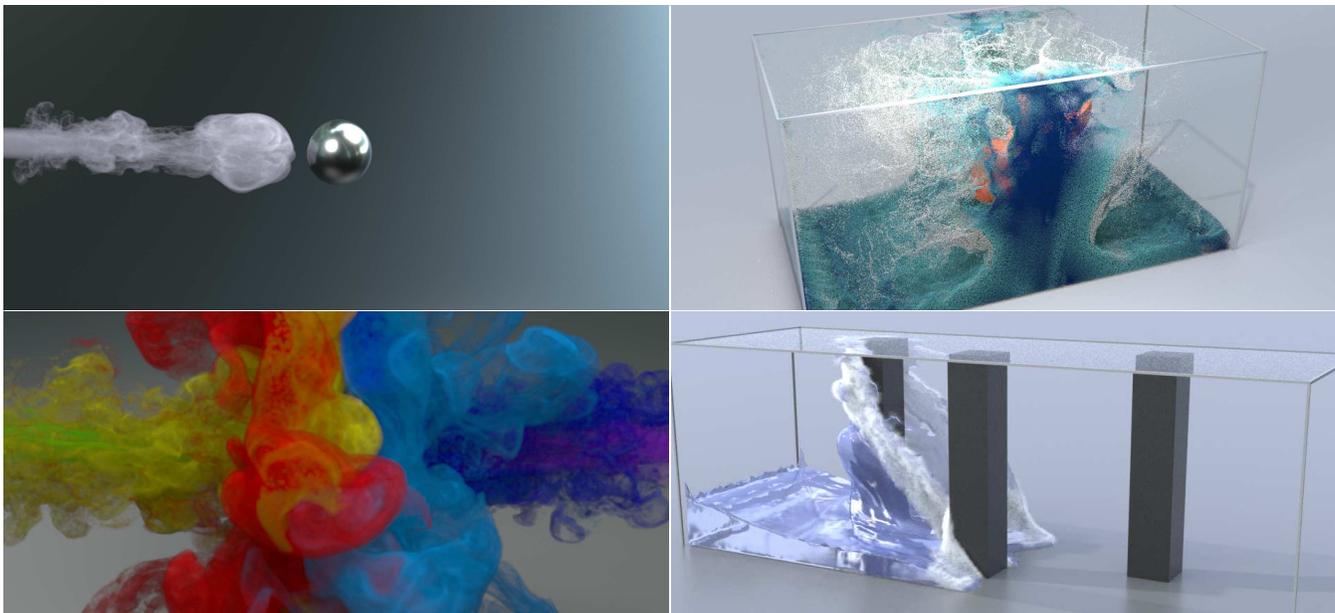
Figure 1: *We simulate detailed incompressible flows with free surfaces and irregular domains using our novel hybrid particle/grid simulation approach. Our numerical method yields intricate flow details with little dissipation, even at modest spatial resolution. Furthermore, we use collocated velocity grids rather than staggered MAC grids.*

Abstract

We present a hybrid particle/grid approach for simulating incompressible fluids on collocated velocity grids. Our approach supports both particle-based Lagrangian advection in very detailed regions of the flow and efficient Eulerian grid-based advection in other regions of the flow. A novel Backward Semi-Lagrangian method is derived to improve accuracy of grid based advection. Our approach utilizes the implicit formula associated with solutions of the inviscid Burgers' equation. We solve this equation using Newton's method enabled by C^1 continuous grid interpolation. We enforce incompressibility over collocated, rather than staggered grids. Our projection technique is variational and designed for B-spline interpolation over regular grids where multiquadratic interpolation is used for velocity and multilinear interpolation for pressure. Despite our use of regular grids, we extend the variational technique to allow for cut-cell definition of irregular flow domains for both Dirichlet and free surface boundary conditions.

CCS Concepts

*• **Mathematics of computing** → **Discretization; Partial differential equations;** Solvers; • **Applied computing** → Physics;*
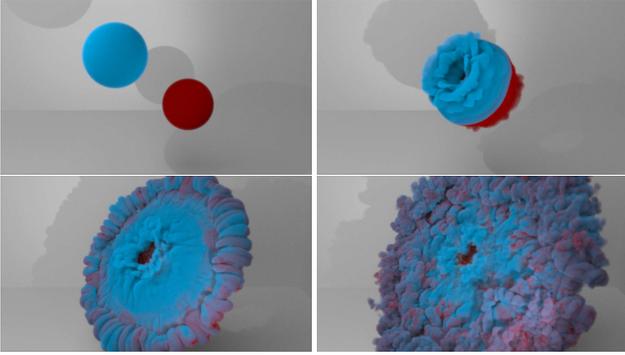
Figure 2: *High-resolution smoke: Two spheres of smoke collide in a high-resolution 3D simulation ($\Delta x = 1/255$). BSQLB accurately resolves vorticial flow detail.*

1. Introduction

Whether it be billowing smoke, energetic explosions, or breaking waves, simulation of incompressible flow is an indispensable tool for modern visual effects. Ever since the pioneering works of Foster and Metaxas [FM96], Stam [Sta99] and Fedkiw et al. [FSJ01; FF01], the Chorin [Cho67] splitting of advective and pressure projection terms has been the standard in computer graphics applications [Bri08]. Most techniques use regular grids of Marker-And-Cell (MAC) [HW65] type with pressure and velocity components staggered at cell centers and faces respectively. Furthermore, advection is most often discretized using semi-Lagrangian techniques originally developed in the atmospheric sciences [Sta99; Rob81]. Although well-established, these techniques are not without their drawbacks. For example, the staggering utilized in the MAC grids is cumbersome since variables effectively live on four different grids. This can complicate many algorithms related to incompressible flow. E.g. Particle-In-Cell (PIC) [Har64] techniques like FLIP [BR86; ZB05], Affine/Polynomial Particle-In-Cell (APIC/PolyPIC) [JSS*15; FGG*17] and the Material Point Method (MPM) [SCS94; SSJ*14] must transfer information separately to and from each individual grid. Similarly, semi-Lagrangian techniques must separately solve for upwind locations at points on each of the velocity component grids. Moreover, while semi-Lagrangian techniques are renowned for the large time steps they admit (notably larger than the Courant-Friedrichs-Lewy (CFL) condition), their inherent stability is plagued by dissipation that must be removed for most visual effects phenomena. Another limitation of the MAC grid arises with free-surface water simulation. In this case, the staggering prevents many velocity components near the fluid free surface from receiving a correction during projection (see e.g. [Bri08]). Each of these velocity components must then be separately extrapolated to from the interior to receive a pressure correction.

MAC grids are useful because the staggering prevents pressure null modes while allowing for accurate second order central differencing in discrete grad/div operators. However, there are alternatives in the computational physics literature. Many mixed Finite Element Method (FEM) techniques use collocated velocities [Hug00] without suffering from pressure mode instabilities. For example,

Taylor-Hood elements [TH73] use collocated multi-quadratic velocity interpolation and multilinear pressure interpolation to enforce incompressiblity. Recently, B-spline interpolation [dBoo78] has been used with Taylor-Hood [Bre10]. We build on this work and develop an approach based on multiquadratic B-spline interpolation. As in Nielsen et al. [NSB*18], we use collocated velocities with pressures staggered at cell centers. This choice is motivated by the simplicity of collocated grids compared to staggering, but also because of the ease of attaining continuous derivatives with B-spline interpolation. For example, this interpolation is often chosen with MPM applications since C^1 interpolation is essential for stability [SKB08]. In the context of fluids, we show that this allows for extremely stable and accurate advection.

We develop a new approach for Chorin splitting [Cho67] based on the collocated multiquadratic B-spline velocity, multilinear pressure Taylor-Hood element [Bre10]. However, unlike the fully collocated technique of Bressan [Bre10], we stagger pressures on the nodes of the grid and velocities at cell centers as in [ATW13; NSB*18], since it reduces coupling in the pressure projection system and naturally accommodates particle-based definition of the flow domain for free-surface simulation of water. Our collocated velocity approach removes many complications associated with MAC grids. For example, with our approach all grid velocities receive a correction by design so no extrapolation of this type is needed. We use regular grids, but as in [BBB07; BB08; LBB17], we allow for irregular domains in a variational way using cut cells. However, rather than a weighted finite difference approach, we use an FEM approach as in XFEM [BGV09; KBT17] and virtual node (VNA) [SSHT14] techniques. In VNA and XFEM approaches, integrals arising in the variational formulation are carried out over the intersection of the grid with the domain geometry.

We leverage C^1 continuity guaranteed by our quadratic B-spline velocity interpolation to develop BSLQB, a novel Backward Semi-Lagrangian (BSL) [Rob81] technique that achieves second order accuracy in space and time. BSL techniques utilize the implicit form of semi-Lagrangian advection. We show that our novel BSL method for quadratic B-splines dramatically reduces numerical dissipation with only a small modification to the widely-adopted explicit semi-Lagrangian formulations typically used in graphics applications. Semi-Lagrangian techniques for velocity advection utilize the implicit relation associated with solution of the invsicid Burgers' equation

$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}(\mathbf{x} - (t-s)\mathbf{u}(\mathbf{x},t),s) \iff \frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{u} = \mathbf{0} \quad (1)$$

Figure 3: *Dam break. A block of water falls in a rectangular domain with obstacles. Dynamic splashing behavior is followed by settling of the water in the tank. White water rendering effects are added based on [IAAT12].*

for $s \leq t$ [Eva10]. We henceforth refer to the inviscid Burgers' equation as Burgers' equation for brevity. Traditionally, graphics applications have preferred the explicit variant of semi-Lagrangian advection whereby grid velocities are updated through the expression

$$\mathbf{u}_{\mathbf{i}}^{n+1} = \mathbf{u}(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{u}_{\mathbf{i}}^{n}, t^{n}) \tag{2}$$

where $\mathbf{x}_{\mathbf{i}}$ is the location of grid node $\mathbf{i}$, $\mathbf{u}_{\mathbf{i}}^{n}, \mathbf{u}_{\mathbf{i}}^{n+1}$ are velocities at the node at times t^{n} and t^{n+1} respectively and interpolation over the velocity grid is used to estimate $\mathbf{u}(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{u}_{\mathbf{i}}^{n}, t^{n})$ at non-grid node locations [Saw63; Sta99]. In contrast, BSL techniques leverage Equation (1) directly

$$\mathbf{u}_{\mathbf{i}}^{n+1} = \mathbf{u}(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{u}_{\mathbf{i}}^{n+1}, t^{n}) \tag{3}$$

which requires the solution of an implicit equation for $\mathbf{u}_{\mathbf{i}}^{n+1}$ [Rob81]. Since our grid interpolation is naturally C^1, we show that this can be done very efficiently using a few steps of Newton's method. While this is more expensive than the explicit semi-Lagrangian formulations, we note that each node can still be updated in parallel since the implicit equations for $\mathbf{u}_{\mathbf{i}}^{n+1}$ are decoupled in $\mathbf{i}$. We show that solution of the implicit Equation (3), rather than the traditionally used explicit Equation (2) improves the order of convergence from first to second (in space and time). Notably, this does not require use of multiple time steps for backward/forward estimations of error, as is commonly done [KLLR06; KLLR05; SFK*08; XK01; SSHT14]. Furthermore, our method allows for larger-than-CFL time steps and is as stable or more so than explicit semi-Lagrangian formulations.

Lastly, we develop a hybrid particle/BSLQB advection technique that utilizes PolyPIC [FGG*17] in portions of the domain covered by particles and BSLQB in portions without particles. Our formulation naturally leverages the strengths of both approaches. Dense concentrations of particles can be added to regions of the domain where more detail is desired. Also, if particle coverage becomes too sparse because of turbulent flows, BSLQB can be used in the gaps. We demonstrate the efficacy of this technique with smoke simu-
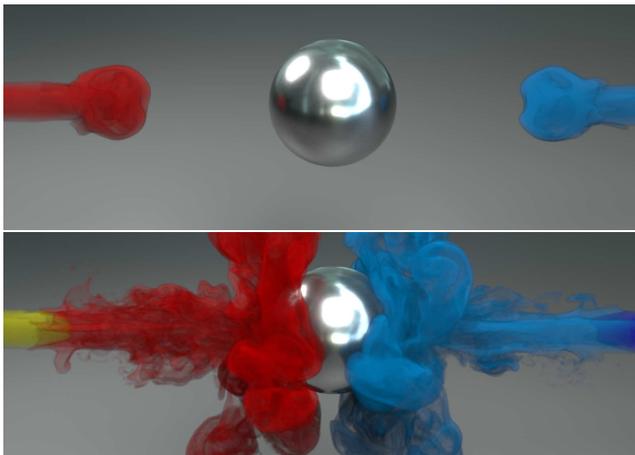
Figure 4: *Colorful smoke jets. Multicolored jets of smoke are simulated with BSLQB. Intricate mixing is induced as the flows collide at the spherical boundary.*

lation and narrow banding of particles near the fluid surface with water simulations as in [CMK15; FAW*16; SWT*18]. In this case, level set advection naturally enabled with our BSLQB formulation is preferred in deeper water regions. We summarize our contributions as:

- A novel cut-cell collocated velocity B-spline mixed FEM method for Chorin [Cho67] splitting discretization of the incompressible Euler equations.
- BSLQB: a novel BSL technique designed for collocated multiquadratic B-spline velocity interpolation that achieves second order accuracy in space and time for the advection step.
- A hybrid BSLQB/PolyPIC method for narrow band free-surface flow simulations and concentrated-detail smoke simulations.
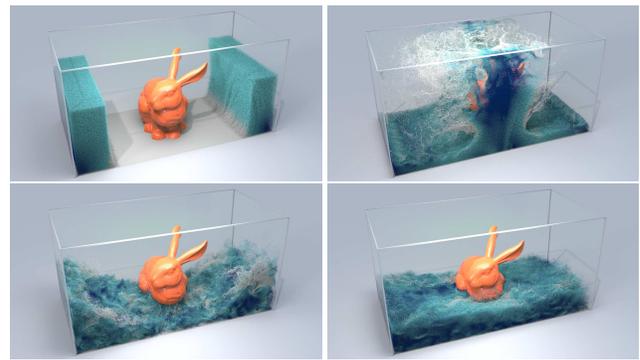
Figure 5: *Dam break with bunny: Opposing blocks of water collapse in a tank and flow around the irregular domain boundary placed in the middle of the tank. Particles are colored from slow (blue) to fast (white) speed.*

2. Previous work

2.1. Advection

Stam [Sta99] first demonstrated the efficacy of semi-Lagrangian techniques for graphics applications and they have since become the standard, largely due to the large time steps they engender and their simple interpolatory nature. Many modifications to the original approach of Stam [Sta99] have been developed, often inspired by approaches in the engineering literature. Fedkiw et al. [FSJ01] use vorticity confinement [SU94] to counterbalance vorticity lost to dissipation and cubic grid interpolation. Kim et al. [KLLR06; KLLR05] and Selle et al. [SFK*08] combine forward and backward semi-Lagrangian steps to estimate and remove dissipative errors. Constrained Interpolation Profile [KSK08; YXU01; SKK09] techniques additionally advect function derivatives to reduce dissipation. Molemaker et al. [MCPN08] use the QUICK technique of Leonhard [Leo79] which is essentially upwinding with quadratic interpolation and Adams-Bashforth temporal discretization, although this does not have the favorable stability properties of semi-Lagrangian. Backward Difference Formula techniques are useful because they use an implicit multistep formulation for higher-order semi-Lagrangian advection yet still only require one projection per time step [XK01; SSHT14].

The main idea in semi-Lagrangian techniques is to interpolate
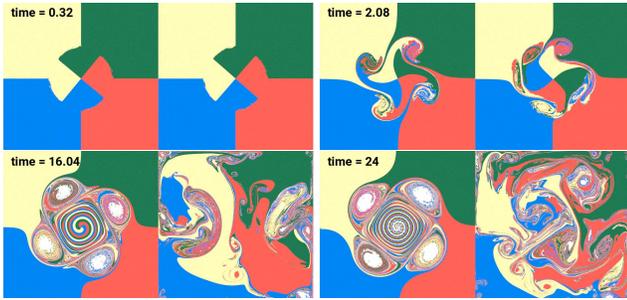
Figure 6: *SL vs. BSLQB. We compare semi-Lagrangian (left) and BSLQB (right) in a vorticity-intensive example. BSLQB breaks symmetry and exhibits a more turbulent flow pattern. Note we only use particles for flow visualization and not for PolyPIC advection in this example.*

data from a characteristic point. This idea goes back to the Courant-Isaacson-Rees [CIR52] method. However, as noted in [FSJ01] semi-Lagrangian advection is very popular in atmospheric science simulation and the variants used in graphics that account for characteristics traveling beyond the local cell in one time step go back to Sawyer [Saw63]. The first BSL approach utilizing Equation (3) was done by Robert [Rob81] in which they use fixed point iteration to solve the nonlinear equation. They fit a bicubic function to their data over 4×4 grid patches, then use that function in the fixed point iteration. If the upwind point leaves the grid, they clamp it to the boundary of the 4×4 patch. This clamping will degrade accuracy for larger time steps. In this case, more general interpolation is typically used (see [SC91; FF98] for useful reviews). Pudykiewicz and Staniforth [PS84] investigate the effects of BSL versus explicit semi-Lagrangian. Specifically, they compare Bates and McDonald [BM82] (explicit) versus Robert [Rob81] (BSL). They show that keeping all things equal, the choice of Equation (2) (explicit) instead of Equation (3) (BSL) leads to more dissipation and mass loss. This is consistent with our observations with BSLQB.

Interestingly, multiquadratic B-splines have not been adopted by the semi-Lagrangian community, despite their natural regularity. Hermite splines, multicubic splines and even Lagrange polynomials are commonly used [SC91]. Preference for Hermite splines and Lagrange polynomials is likely due to their local nature (they do not require solution of a global system for coefficients) and preference for multicubic splines (over multi-quadratic) is possibly due to the requirement of odd degree for natural splines (odd degree splines behave like low pass filters and tend to be smoother than even degree splines [CWB01; CK12]). Cubic splines are considered to be more accurate than Hermite splines and Lagrange interpolation [SC91; MK96]. Interestingly, Riishøjgaard et al. [RCLM98] found that cubic spline interpolation gave rise to a noisier solution than cubic Lagrange interpolation with a technique analogous to that of Makar and Karpik [MK96]. However, they also note that addition of a selective scale diffusion term helps reduce noise associated with cubic splines. Wang and Layton [WL10] use linear B-splines with BSL but only consider one space dimension which makes Equation (3) linear and easily solvable.

Dissipation with explicit semi-Lagrangian advection is so severe

that many graphics researchers have resorted to alternative methods to avoid it. Mullen et al. [MCP*09] develop energy preserving integration to prevent the need for correcting dissipative behavior. Some authors [QZG*19; TP11; SIBA17; SBIA18] resolve the flow map characteristics for periods longer than a single time step (as opposed to one step with semi-Lagrangian) to reduce dissipation. Hybrid Lagrange/Eulerian techniques like PIC (and related approaches) [Bri08; JSS*15; FGG*17; ZB05] explicitly track motion of particles in the fluid, which is nearly dissipation-free, but can suffer from distortion in particle sampling quality. Vorticity formulations are also typically less dissipative, but can have issues with boundary conditions enforcement [SRF05; AN05; CKP*16; STK*07; PK05; WP10]. Zehnder et al., Zhang et al. and Mullen et al. [MCP*09; ZNT18; NZT19; ZBG15] have noted that the Chorin projection itself causes dissipation. Zhang et al. [ZBG15] reduced artificial dissipation caused by the projection step by estimating lost vorticity and adding it back into the fluid. Zehnder et al. [ZNT18; NZT19] propose a simple, but very effective modification to the splitting scheme that is similar to midpoint rule integration to reduce the projection error.

2.2. Pressure projection

Graphics techniques utilizing pressure projection typically use voxelized MAC grids with boundary conditions enforced at cell centers and faces, however many methods improve this by taking into account sub-cell geometric detail. Enright et al. [ENGF03] showed that enforcing the pressure free surface boundary condition at MAC grid edge crossings (rather than at cell centers) dramatically improved the look of water surface waves and ripples. Batty, Bridson and colleagues developed variational weighted finite difference approaches to enforce velocity boundary conditions with MAC grids on edge crossings and improved pressure boundary conditions at the free surface in the case of viscous stress [BBB07; BB08; LBB17]. XFEM [BGV09; KBT17] and virtual node (VNA) [SSHT14] techniques also use cut cell geometry with variational techniques. Schroeder et al. [SSHT14] use cut cells with MAC grids, but their technique is limited to moderate Reynolds numbers. Recently, Nielsen et al. [NSB*18] have shown that collocated velocities with only staggered pressures can be used effectively for projection with turbulent detailed flow simulations.

There is a vast literature on enforcing incompressibility in the FEM community [Hug00]. Our approach is most similar to the B-spline Taylor-Hood element of Bressan [Bre10]. Adoption of B-spline interpolation in FEM is part of the isogeometric movement [HCB05; RC12]. Originally motivated by the desire to streamline the transition from computer-aided design (CAD) to FEM simulation, isogeometric analysis explores the use of CAD-based interpolation (e.g. B-splines and nonuniform rational B-splines (NURBS)) with FEM methodologies. Hughes et al. [HCB05] show that in addition to simplifying the transition from CAD to simulation, the higher regularity and spectral-like properties exhibited by these splines makes them more accurate than traditionally used interpolation. We enforce Dirichlet boundary conditions weakly as in XFEM and VNA approaches [BGV09; KBT17; SSHT14]. Bazilevs et al. [BH07] show that weak Dirichlet enforcement with isogeometric analysis can be more accurate than strong enforcement.
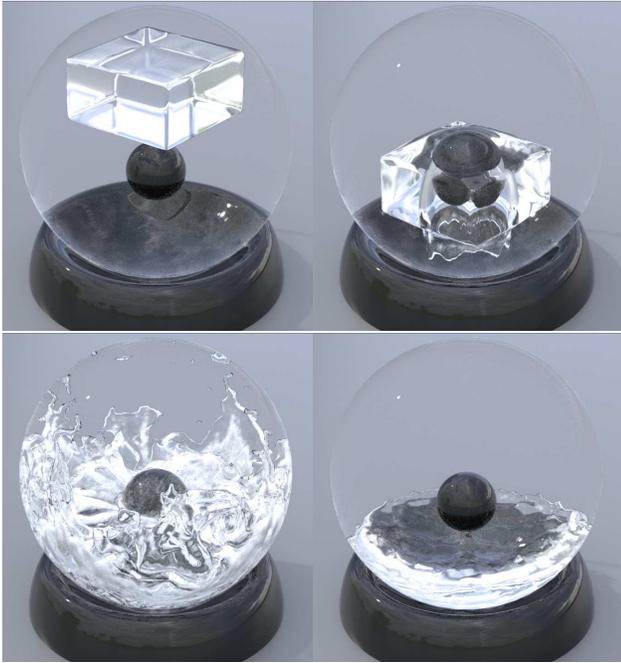
Figure 7: *Water in a globe. A block of water splashes and naturally slides along cut cell boundaries in an irregular domain interior to one large sphere and exterior to one small sphere.*

Figure 8: *Smoke in an irregular domain. Multicolored spheres of smoke with non-zero initial velocity conditions flow and collide inside the Stanford bunny. Zero normal velocity is enforced with our cut cell formulation.*

Edwards and Bridson [EB14] use a discontinuous Galerkin FEM approach to simulate free surface flows over adaptive grids. Ferstl et al. [FWD14] also use an FEM based approach for discretization of pressure projections over adaptive hexahedral grids. Schneider et al. [SDG*19] recently developed a third order accurate FEM approach for solving Poisson and other problems on predominantly hexahedral meshes.

Graphics applications are typically concerned with turbulent, high-Reynolds numbers flows. Interestingly, B-splines have proven effective for these flows by researchers in the Large Eddy Simulation (LES) community [Kim98; KMS99]. Kravchenko et al. [KMS99] use a variational weighted residuals approach with B-splines for turbulent LES and show that the increased regularity significantly reduces computational costs. Botella [Bot02] use a similar approach, but apply a collocation technique where the strong form of the div-grad formulation of incompressibility is enforced pointwise. They show that their B-spline approach attains optimal order of accuracy with accurate resolution of quadratic flow invariants. Botella [Bot02] also introduce a notion of sparse approximation to the inverse mass matrix to avoid dense systems of equations in the pressure solve.

3. Governing equations and operator splitting

We solve the incompressible Euler equations that describe the evolution of a fluid in terms of its mass density ρ, velocity $\mathbf{u}$, pressure p and gravitational constant $\mathbf{g}$ as

$$\rho \frac{D\mathbf{u}}{Dt} = \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \mathbf{u} \right) = -\nabla p + \rho \mathbf{g}, \ \mathbf{x} \in \Omega \quad (4)$$

$$\nabla \cdot \mathbf{u} = 0, \ \mathbf{x} \in \Omega \quad (5)$$

$$\mathbf{u} \cdot \mathbf{n} = a, \ \mathbf{x} \in \partial\Omega_S \quad (6)$$

$$p = 0, \ \mathbf{x} \in \partial\Omega_{FS} \quad (7)$$

where Equation (4) is balance of linear momentum, Equation (5) is the incompressibility constraint, Equation (6) is the boundary condition for the normal component of the velocity and Equation (7) is the free surface boundary condition. We use Ω to denote the region occupied by the fluid, $\partial\Omega_S$ to denote the portion of the boundary of the fluid domain on which velocity is prescribed to be a (which may vary over the boundary) and $\partial\Omega_{FS}$ is the surface of the water where the pressure is zero (see Figure 9).

In a Chorin [Cho67] operator splitting of the advective and pressure terms, velocity is first updated to an intermediate field $\mathbf{w}$ under the convective $\rho \frac{D\mathbf{u}}{Dt} = \mathbf{0}$, followed by an update from the pressure and gravitational body forcing under $\rho \frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \rho \mathbf{g}$ where the pressure is determined to enforce $\nabla \cdot \mathbf{u} = 0$. Dividing by the mass density, the convective step is seen to be an update under Burgers' equation (1). Burgers' equation governs temporally constant Lagrangian velocity (zero Lagrangian acceleration). The characteristic curves for flows of this type are straight lines (since the Lagrangian acceleration is zero), on which the velocity is constant (see Figure 10). This gives rise to the implicit relation $\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x} - (t - s)\mathbf{u}(\mathbf{x}, t), s)$ for $s \leq t$. Intuitively, if we want to
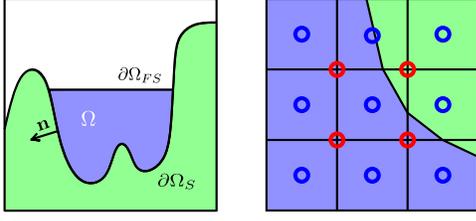
Figure 9: *Flow domain and grid. Left: we use Ω to denote the fluid domain, with $\partial\Omega_S$ used to indicate the portion of the fluid domain subject to velocity boundary conditions and $\partial\Omega_{FS}$ to indicate the free-surface portion of the boundary with pressure condition $p = 0$. Right: We use multiquadratic interpolation for velocity ($\bar{\mathbf{u}}_{\mathbf{i}}$ at cell centers, blue) and multilinear for pressure ($p_{\mathbf{c}}$ at nodes, red). The fluid domain is defined with sub-grid-cell accuracy.*

know the velocity $\mathbf{u}(\mathbf{x}, t)$ at point $\mathbf{x}$ at time t, we look back along the characteristic passing through $\mathbf{x}$ at time t to any previous time s; however, the characteristic is the straight line defined by the velocity $\mathbf{u}(\mathbf{x}, t)$ that we want to know. Hence we take an implicit approach to the solution of this equation, which when combined with the operator splitting amounts to

$$\frac{\mathbf{w} - \tilde{\mathbf{u}}^n}{\Delta t} = \mathbf{0} \tag{8}$$

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{w}}{\Delta t} = -\nabla p^{n+1} + \rho \mathbf{g} \tag{9}$$

$$\nabla \cdot \mathbf{u}^{n+1} = \mathbf{0} \tag{10}$$

where we use the notation $\mathbf{u}^{n+\alpha}(\mathbf{x}) = \mathbf{u}(\mathbf{x}, t^{n+\alpha})$, $\alpha = 0, 1$ to denote the time $t^{n+\alpha}$ velocities. Furthermore, the intermediate velocity $\mathbf{w}$ is related to $\tilde{\mathbf{u}}^n$ through $\tilde{\mathbf{u}}^n(\mathbf{x}) = \mathbf{u}(\mathbf{x} - \Delta t \mathbf{w}(\mathbf{x}), t^n)$.

4. Spatial discretization

We discretize in space by first representing velocity and pressure in terms of multiquadratic and multilinear B-splines for velocity and pressure respectively. We use a regular grid with spacing Δx and define pressure degrees of freedom at grid vertices and velocity degrees of freedom at grid cell centers as in [ATW13] (see Figure 9). This efficiently aligns the support of the multiquadratic and multilinear interpolating functions which naturally allows for a grid-cell-wise definition of the flow domain (see Figure 11). We use $N_{\mathbf{i}}(\mathbf{x})$ to represent the multiquadratic B-spline basis function associated with velocity degree of freedom $\bar{\mathbf{u}}_{\mathbf{i}}$ at grid cell center $\mathbf{x}_{\mathbf{i}}$ and $\chi_{\mathbf{c}}(\mathbf{x})$ for the multilinear basis function associated with pressure $p_{\mathbf{c}}$ at grid node $\mathbf{x}_{\mathbf{c}}$. These are defined as

$$N_{\mathbf{i}}(\mathbf{x}) = \prod_\alpha \hat{N}\left(\frac{x_\alpha - x_{\alpha \mathbf{i}}}{\Delta x}\right), \; \chi_{\mathbf{c}}(\mathbf{x}) = \prod_\alpha \hat{\chi}\left(\frac{x_\alpha - x_{\alpha \mathbf{c}}}{\Delta x}\right) \tag{11}$$

$$\hat{N}(\eta) = \begin{cases} \frac{1}{2}\left(\frac{3}{2} - |\eta|\right)^2, & |\eta| \in (\frac{1}{2}, \frac{3}{2}) \\ -\eta^2 + \frac{3}{4}, & |\eta| \in [0, \frac{1}{2}] \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$\hat{\chi}(\nu) = \begin{cases} 1 - |\nu|, & |\nu| \in (0, 1) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where we use Greek indices α to indicate components of the vectors $\mathbf{x}$, $\mathbf{x}_{\mathbf{i}}$ and $\mathbf{x}_{\mathbf{c}}$. With this convention we interpolate to define velocity and pressure fields

$$\mathbf{u}(\mathbf{x}) = \sum_{\mathbf{i}} \bar{\mathbf{u}}_{\mathbf{i}} N_{\mathbf{i}}(\mathbf{x}), \; p(\mathbf{x}) = \sum_{\mathbf{c}} p_{\mathbf{c}} \chi_{\mathbf{c}}(\mathbf{x}). \tag{14}$$

We use the notation $\bar{\mathbf{u}}_{\mathbf{i}}$ to distinguish it from the velocity at the grid node $\mathbf{u}(\mathbf{x}_{\mathbf{i}}) = \sum_{\mathbf{j}} \bar{\mathbf{u}}_{\mathbf{j}} N_{\mathbf{j}}(\mathbf{x}_{\mathbf{i}})$ since the multiquadratic B-splines are not interpolatory and these will in general be different. Note that multilinear interpolation is interpolatory and $p_{\mathbf{c}} = \sum_{\mathbf{d}} p_{\mathbf{d}} \chi_{\mathbf{d}}(\mathbf{x}_{\mathbf{c}})$.

4.1. BSLQB advection

With this interpolation choice, we first solve for intermediate grid node velocity values $\mathbf{w}(\mathbf{x}_{\mathbf{i}})$ from Equation (8) as

$$\mathbf{w}(\mathbf{x}_{\mathbf{i}}) = \sum_{\mathbf{j}} \bar{\mathbf{u}}_{\mathbf{j}}^n N_{\mathbf{j}}\left(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{w}(\mathbf{x}_{\mathbf{i}})\right). \tag{15}$$

We can solve this equation using Newton's method since the multiquadratic B-splines are C^1. We use $\mathbf{w}_{\mathbf{i}}^k$ to denote the kth Newton approximation to $\mathbf{w}(\mathbf{x}_{\mathbf{i}})$. Explicit semi-Lagrangian is used as an initial guess with $\mathbf{w}_{\mathbf{i}}^0 = \sum_{\mathbf{j}} \bar{\mathbf{u}}_{\mathbf{j}}^n N_{\mathbf{j}}(\mathbf{x}_{\mathbf{i}} - \Delta t \sum_{\mathbf{l}} \bar{\mathbf{u}}_{\mathbf{l}}^n N_{\mathbf{l}}(\mathbf{x}_{\mathbf{i}}))$ and then we update iteratively via $\mathbf{w}_{\mathbf{i}}^k \mathrel{+}= \delta \mathbf{u}^k$ with Newton increment $\delta \mathbf{u}^k$ satisfying

$$\delta \mathbf{u}^k = \left(\mathbf{I} + \Delta t \frac{\partial \mathbf{u}^n}{\partial \mathbf{x}}\left(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{w}_{\mathbf{i}}^k\right)\right)^{-1} \left(\sum_{\mathbf{j}} \bar{\mathbf{u}}_{\mathbf{j}}^n N_{\mathbf{j}}\left(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{w}_{\mathbf{i}}^k\right) - \mathbf{w}_{\mathbf{i}}^k\right)$$

where $\frac{\partial \mathbf{u}^n}{\partial \mathbf{x}}\left(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{w}_{\mathbf{i}}^k\right) = \sum_{\mathbf{j}} \bar{\mathbf{u}}_{\mathbf{j}}^n \frac{\partial N_{\mathbf{j}}}{\partial \mathbf{x}}\left(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{w}_{\mathbf{i}}^k\right)$. It is generally observed [KW90; PS84] that with BSL approaches of this type, this iteration will converge as long as $\mathbf{I} + \Delta t \sum_{\mathbf{j}} \bar{\mathbf{u}}_{\mathbf{j}}^n \frac{\partial N_{\mathbf{j}}}{\partial \mathbf{x}}\left(\mathbf{x}_{\mathbf{i}} - \Delta t \mathbf{w}_{\mathbf{i}}^k\right)$ is non-singular. We note that this condition holds as long as no shocks form under Burgers' equation [Eva10] (forward from time t^n). This is a safe assumption since we are modeling incompressible flow with which shock formation does not occur, but it may be a problem for compressible flows. In practice, this iteration converges in 3 or 4 iterations, even with CFL numbers larger than 4 (see Section 7.1). When it does fail (which occurs less than one percent of the time in the examples we run), it is usually for points near the boundary with characteristics that leave the domain (since we cannot estimate $\frac{\partial \mathbf{u}^n}{\partial \mathbf{x}}$ using grid interpolation if the upwind estimate leaves the grid). In this case we use explicit semi-Lagrangian and interpolate from the boundary conditions if the characteristic point is off the domain.

Once we have obtained the grid node values of the intermediate velocity $\mathbf{w}(\mathbf{x}_{\mathbf{i}})$, we must determine interpolation coefficients $\bar{\mathbf{w}}_{\mathbf{j}}$ such that $\mathbf{w}(\mathbf{x}_{\mathbf{i}}) = \sum_{\mathbf{j}} \bar{\mathbf{w}}_{\mathbf{j}} N_{\mathbf{j}}(\mathbf{x}_{\mathbf{i}})$. On the boundary of the grid, we set $\bar{\mathbf{w}}_{\mathbf{j}} = \mathbf{w}(\mathbf{x}_{\mathbf{j}})$ since we can only interpolate to $\mathbf{x}_{\mathbf{i}}$ if all of its neighbors have data. This yields a square, symmetric positive definite system of equations for the remaining $\bar{\mathbf{w}}_{\mathbf{j}}$. The system is very well conditioned with sparse, symmetric matrix $N_{\mathbf{j}}(\mathbf{x}_{\mathbf{i}})$ consisting of non-negative entries and rows that sum to one. The sparsity and symmetry of the system arises from the compact support and geometric symmetry, respectively, of the B-spline basis functions $N_{\mathbf{j}}$. In practice, the system can be solved to machine precision in tens of unpreconditioned CG iterations. We have noticed that for some flows, determining the coefficients $\bar{\mathbf{w}}_{\mathbf{j}}$ can lead to increasingly oscillatory
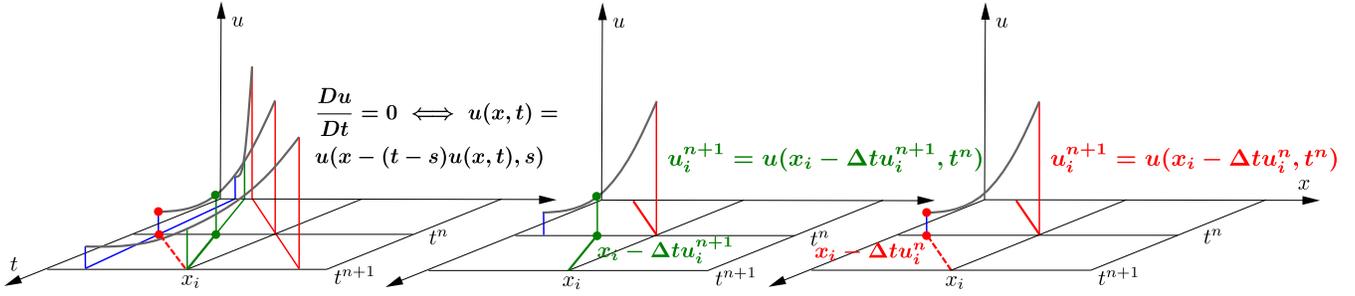
Figure 10: *BSL versus SL. We illustrate the difference between explicit semi-Lagrangian and BSL in 1D. Left: The exact solution of Burgers' equation has straight line characteristics shown in blue, green and red on which velocity (plotted above the plane in gray) is constant. Center: BSL (green) uses Newton's method to solve for the exact characteristic going through x_i at time t^{n+1} to determine u_i^{n+1}. Right: explicit semi-Lagrangian (red) uses a stale, time t^n approximation of the characteristic which overshoots, resulting in an underestimate of the velocity and energy loss.*

velocity fields. This is perhaps due to the unfavorable filtering properties of even order B-splines [CWB01; CK12]. However, we found that a simple stabilization strategy can be obtained as

$$\sum_{\mathbf{j}} \left(\lambda N_{\mathbf{j}}(\mathbf{x_i}) + (1-\lambda)\delta_{\mathbf{ij}} \right) \bar{\mathbf{w}}_{\mathbf{j}} = \mathbf{w}(\mathbf{x_i}) \tag{16}$$

where $\lambda \in [0,1]$ and $\delta_{\mathbf{ij}}$ is the Kronecker delta. A value of $\lambda = 0$ is very stable, but extremely dissipative. Stable yet energetic behavior is achieved by decreasing the value of λ under grid refinement. In practice we found that $\lambda \in (.95,1]$ with $\lambda = c\Delta x$ for constant c provided a good balance without compromising second order accuracy of the method (see Section 7.1). We note that Riishøjgaard et al. [RCLM98] also added diffusion to cubic spline interpolation based semi-Lagrangian to reduce noise.

4.2. Hybrid BSLQB-PolyPIC advection

In some portions of the domain, we store particles with positions $\mathbf{x}_p^n$ and PolyPIC [FGG*17] velocity coefficients $\mathbf{c}_p^n$. In the vicinity of the particles, we use PolyPIC [FGG*17] to update the intermediate velocity field $\bar{\mathbf{w}}_{\mathbf{j}}$. First we update particle positions as $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^n$ (where the velocity $\mathbf{v}_p^n$ is determined from $\mathbf{c}_p^n$ following [FGG*17]). Then the components $\bar{w}_{\mathbf{j}\alpha}$ of the coefficients $\bar{\mathbf{w}}_{\mathbf{j}}$ are determined as

$$\bar{w}_{\mathbf{j}\alpha} = \frac{\sum_p m_p N_{\mathbf{j}}(\mathbf{x}_p^{n+1}) \left(\sum_{r=1}^{N_r} s_r(\mathbf{x_j} - \mathbf{x}_p^{n+1}) c_{pr\alpha}^n \right)}{\sum_p m_p N_{\mathbf{j}}(\mathbf{x}_p^{n+1})} \tag{17}$$

where N_r is the number of polynomial modes $s_r(\mathbf{x})$, as in Fu et al. [FGG*17]. To create our hybrid approach, we update $\bar{w}_{\mathbf{j}\alpha}$ from Equation (17) whenever the denominator is greater than a threshold $\sum_p m_p N_{\mathbf{j}}(\mathbf{x}_p^{n+1}) > \tau^m$, otherwise we use the BSLQB update from Equation (16). We use this threshold because the grid node update in Equation(17) loses accuracy when the denominator is near zero and in this case the BSLQB approximation is likely more accurate. Note that the polynomial mode coefficients for the next time step $\mathbf{c}_p^{n+1}$ are determined from the grid velocities at the end of the time step (using particle positions $\mathbf{x}_p^{n+1}$ and after pressure projection).

5. Pressure projection

We solve Equations (9)-(10) and boundary condition Equations (6)-(7) in a variational way. To do this, we require that the dot products of Equations (9), (10) and Equations (6) with arbitrary test functions $\mathbf{r}$, q and μ respectively integrated over the domain are always equal to zero. The free surface boundary condition in Equation (7) is naturally satisfied by our treatment of Equation (9). We summarize this as

$$\int_{\Omega} \mathbf{r} \cdot \rho \left(\frac{\mathbf{u}^{n+1} - \mathbf{w}}{\Delta t} \right) d\mathbf{x} = \int_{\Omega} p^{n+1} \nabla \cdot \mathbf{r} + \rho \mathbf{r} \cdot \mathbf{g} d\mathbf{x} \tag{18}$$

$$- \int_{\partial\Omega} p^{n+1} \mathbf{r} \cdot \mathbf{n} ds(\mathbf{x})$$

$$\int_{\Omega} q \nabla \cdot \mathbf{u}^{n+1} d\mathbf{x} = 0 \tag{19}$$

$$\int_{\partial\Omega_S} \mu \left(\mathbf{u}^{n+1} \cdot \mathbf{n} - a \right) ds(\mathbf{x}) = 0. \tag{20}$$

Here we integrate by parts in the integral associated with Equation (9). Furthermore, we modify the expression $\int_{\partial\Omega} p^{n+1} \mathbf{r} \cdot \mathbf{n} ds(\mathbf{x})$ in Equation (18) in accordance with the boundary conditions. We know that the pressure is zero on $\partial\Omega_{FS}$, however we do not know its value on $\partial\Omega_S$. We introduce the pressure on this portion of the domain as a Lagrange multiplier λ^{n+1} associated with satisfaction of the velocity boundary condition in Equation (20). Physically, this is the external pressure we would need to apply on $\partial\Omega_S$ to ensure that $\mathbf{u}^{n+1} \cdot \mathbf{n} = a$. With this convention, we have $\int_{\partial\Omega} p^{n+1} \mathbf{r} \cdot \mathbf{n} ds(\mathbf{x}) = \int_{\partial\Omega_S} \lambda^{n+1} \mathbf{r} \cdot \mathbf{n} ds(\mathbf{x})$. We note that unlike Equation (20) (and its strong form counterpart (6)) that requires introduction of a Lagrange multiplier, Equation (7) is naturally enforced through the weak form simply by setting $p^{n+1} = 0$ in the integral over $\partial\Omega_{FS}$ in Equation (18).

To discretize in space, we introduce interpolation for the test functions $\mathbf{r}$, q and μ. We use the same spaces as in Equation (14) for velocity and pressure for $\mathbf{r} = \sum_{\mathbf{i}} \bar{\mathbf{r}}_{\mathbf{i}} N_{\mathbf{i}}$ and $q = \sum_{\mathbf{d}} q_{\mathbf{d}} \chi_{\mathbf{d}}$. For the test functions μ, we choose the same space as q, p, but with functions restricted to $\partial\Omega_S$, $\mu = \sum_{\mathbf{b}} \mu_{\mathbf{b}} \chi_{\mathbf{b}}$ for $\mathbf{b}$ with grid cell $\Omega_{\mathbf{b}} \cap \partial\Omega_S \neq \emptyset$ (see Figure 11). We choose the same space for $\lambda^{n+1} = \sum_{\mathbf{b}} \lambda_{\mathbf{b}}^{n+1} \chi_{\mathbf{b}}$ to close the system. We note that this choice of interpolating functions is necessary for preserving a standing pool since the pressure

and λ interpolation functions need to cancel out to prevent artificial currents (see proof in [GHM*20]). With these choices for the test functions, the variational problem is projected to a finite dimensional problem defined by the interpolation degrees of freedom. This is expressed as a linear system for velocities $\bar{\mathbf{u}}_{\mathbf{j}}^{n+1}$, internal pressures $p_{\mathbf{c}}^{n+1}$, and external pressures $\lambda_{\mathbf{b}}^{n+1}$ that is equivalent to

$$
\begin{pmatrix} \mathbf{M} & -\mathbf{D}^T & \mathbf{B}^T \\ -\mathbf{D} & & \\ \mathbf{B} & & \end{pmatrix} \begin{pmatrix} \mathbf{U}^{n+1} \\ \mathbf{P}^{n+1} \\ \Lambda^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{MW} + \hat{\mathbf{g}} \\ \mathbf{0} \\ \mathbf{A} \end{pmatrix}. \quad (21)
$$

Here $\mathbf{U}^{n+1}$, $\mathbf{P}^{n+1}$ and Λ^{n+1} are the vectors of all unknown $\bar{\mathbf{u}}_{\mathbf{j}}^{n+1}$, $p_{\mathbf{c}}^{n+1}$ and $\lambda_{\mathbf{b}}^{n+1}$ respectively. Furthermore $\mathbf{M}$ is the mass matrix, $\mathbf{B}$ defines the velocity boundary conditions and $\mathbf{D}$ defines the discrete divergence condition. Lastly, $\mathbf{W}$ is the vector of all $\bar{\mathbf{w}}_{\mathbf{i}}$ that define the intermediate velocity, $\hat{\mathbf{g}}$ is from gravity and $\mathbf{A}$ is the variational boundary condition. Using the convention that Greek indices α, β range from 1–3, these matrices and vectors have entries

$$
M_{\alpha \mathbf{i} \beta \mathbf{j}} = \delta_{\alpha\beta} \int_{\Omega} \frac{\rho}{\Delta t} N_{\mathbf{i}} N_{\mathbf{j}} d\mathbf{x}, \quad (22)
$$

$$
D_{\mathbf{d}\beta \mathbf{j}} = \int_{\Omega} \chi_{\mathbf{d}} \frac{\partial N_{\mathbf{j}}}{\partial x_{\beta}} d\mathbf{x}, \ \hat{g}_{\alpha \mathbf{i}} = \int_{\Omega} \rho g_{\alpha} N_{\mathbf{i}} d\mathbf{x} \quad (23)
$$

$$
B_{\mathbf{b}\beta \mathbf{j}} = \int_{\Omega_S} \chi_{\mathbf{b}} N_{\mathbf{j}} n_{\beta} ds(\mathbf{x}), \ A_{\mathbf{b}} = \int_{\Omega} a \chi_{\mathbf{b}} ds(\mathbf{x}). \quad (24)
$$

If we define $\mathbf{G} = [-\mathbf{D}^T, \mathbf{B}^T]$, we can convert this system into a symmetric positive definite one for $\mathbf{P}^{n+1}$ and Λ^{n+1} followed by a velocity correction for $\mathbf{U}^{n+1}$

$$
\begin{pmatrix} \mathbf{P}^{n+1} \\ \Lambda^{n+1} \end{pmatrix} = \left(\mathbf{G}^T \mathbf{M}^{-1} \mathbf{G} \right)^{-1} \left(\mathbf{G}^T \left(\mathbf{W} + \mathbf{M}^{-1} \hat{\mathbf{g}} \right) - \begin{pmatrix} \mathbf{0} \\ \mathbf{A} \end{pmatrix} \right)
$$
$$(25)$$

$$
\mathbf{U}^{n+1} = -\mathbf{M}^{-1} \mathbf{G} \begin{pmatrix} \mathbf{P}^{n+1} \\ \Lambda^{n+1} \end{pmatrix} + \mathbf{W} + \mathbf{M}^{-1} \hat{\mathbf{g}}. \quad (26)
$$

Unfortunately, this system will be dense in the current formulation since the full mass matrix $M_{\alpha \mathbf{i} \beta \mathbf{j}}$ is non-diagonal with dense inverse [Bot02]. However, a simple lumped mass approximation

$$
M_{\alpha \mathbf{i} \beta \mathbf{j}}^{l} = \begin{cases} \delta_{\alpha\beta} \int_{\Omega} \frac{\rho}{\Delta t} N_{\mathbf{i}} d\mathbf{x}, & \mathbf{i} = \mathbf{j} \\ 0, & \text{otherwise} \end{cases} \quad (27)
$$

gives rise to a sparse matrix in Equation (25).
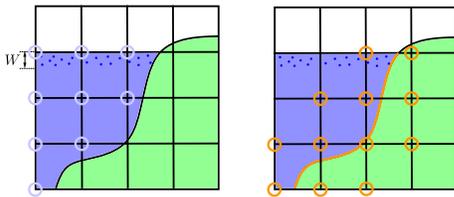
Figure 11: *Discrete free surface fluid domain.* **Left:** *We define the fluid domain to consist of cells that either have (1) a particle (dark blue) in it or (2) a node with non-positive level set value (light blue).* **Right:** *Boundary Lagrange multiplier external pressure $\lambda_{\mathbf{b}}$ (orange circles) are like the interior pressures $p_{\mathbf{c}}$ except only defined on fluid domain cells that intersect $\partial \Omega_S$.*
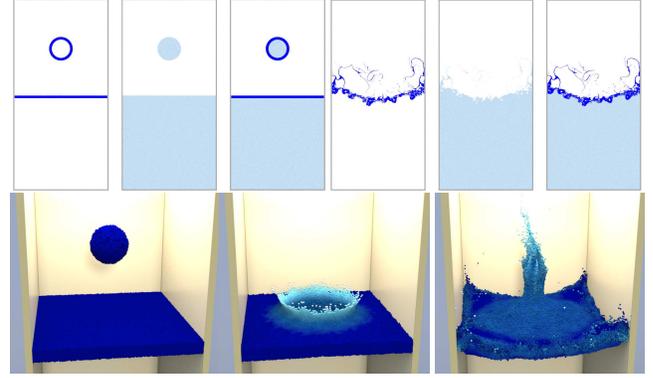
Figure 12: *Narrow band free surface. A circle/sphere falls in a tank of water under gravity. Using only a narrow band of particles saves computational cost and enables increased resolution of the free surface. Top: In 2D we illustrate the hybrid particle (dark blue)/level set (light blue) representation. Bottom: Particles are colored based on velocity magnitude.*

5.1. Cut cells

As in XFEM and VNA approaches [BGV09; KBT17; SSHT14], we resolve sub-grid-cell geometry by simply performing the integrations in Equations (23)-(24) over the geometry of the fluid domain. We use a level set to define solid boundaries (green in Figure 11) on which velocity boundary conditions are defined. We triangulate the zero isocontour using marching cubes [Che95] (see Figure 13). The integrals in Equations (23)-(24) all involve polynomials over volumetric polyhedra (Equations (23), blue in Figure 13) or surface polygons (Equations (24), green in Figure 13) and we use Gauss quadrature of order adapted to compute the integrals with no error (see [GHM*20]). For free surface flows, we use particles (and additionally a level set function in the case of narrow banding, see Section (6)) to denote grid cells with fluid in them. Cells near the solid boundary are clipped by the marching cubes geometry, but otherwise the free surface boundary is voxelized. The fluid domain Ω is defined as the union of all clipped and full fluid cells (see Figure 11).

Notably, taking a cut cell approach with our variational formulation allows us to prove that our method can resolve a standing pool of water exactly without producing numerical currents. We know that with gravitational force $\rho \mathbf{g}$ (e.g. with $\mathbf{g}$ pointing in the y direction with magnitude g), steady state is maintained if the pressure increases with depth as $p = \rho g (y_0 - y)$ where y_0 is the height of the water surface at rest, since $-\nabla p + \rho \mathbf{g} = \mathbf{0}$. Since we use multilinear interpolating functions for p, the exact solution is representable in our discrete space and with a short proof we show (see [GHM*20]) that this means our method will choose it to maintain a standing pool of water, independent of fluid domain boundary geometry.

6. Narrow band free surface

For free surface flows, we develop a narrow band approach as in [CMK15; FAW*16; SWT*18]. We represent the fluid domain with a level set and seed particles in a band of width W from the zero
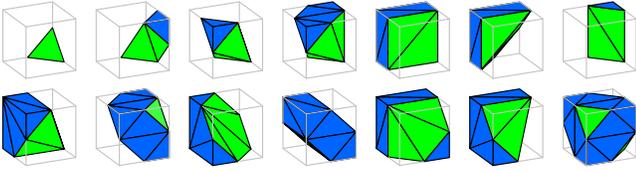
Figure 13: *Cut cells. We show the 14 essential cases used in determining the cut cell fluid domain geometry. Blue faces indicate the intersection of the grid cell with the fluid domain. Green faces indicate the velocity boundary condition faces on $\partial\Omega_S$.*

isocontour (see Figure 11). Particles are advected and used to augment BSLQB advection as detailed in Section 4.2. We also advect the level set by interpolating its value at the previous step from the upwind location $\mathbf{x_i} - \Delta t \mathbf{w}(\mathbf{x_i})$ determined in Equation (15). We then use the updated particle locations to compute a narrow band level set from the particles based on the method of Boyd and Bridson [BB12]. We update the level set to be the union of that defined by the narrow band and that from advection. This is done by taking the minimum of the two level set values and then redistancing with the method of Zhao [Zha05].

7. Examples
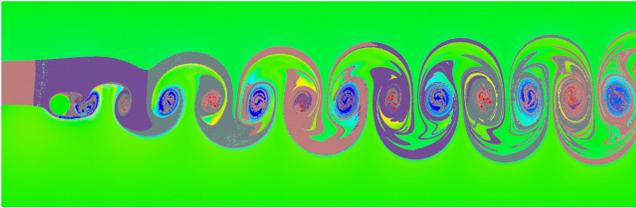
Figure 14: *Von Karman vortex shedding. We demonstrate the accuracy of our Hybrid BSLQB/PolyPIC with vortex shedding past a disk in 2D. Note the smooth transition between regions with particles (PolyPIC) and those without (BSLQB).*

7.1. Hybrid BSLQB/PolyPIC

We demonstrate our hybrid BSLQB/PolyPIC advection with water simulation. We prevent excessive run times by utilizing a narrow band of particles near the free surface and a level set (with BSLQB advection) in deeper levels. Figure 12 Top shows a disc of water splashing in a rectangular tank with dimension 1×2 and grid cell size $\Delta x = 1/255$. The time step is restricted to be in the range $\Delta t \in [0.005, 0.01]$. 20 particles are initialized in every cell that is initially in a narrow band of $7\Delta x$ below the zero isocontour of the level set. Figure 12 Bottom shows an analogous 3D example where a sphere of water splashes in a tank. A cell size of $\Delta x = \frac{1}{63}$ is used in a domain with dimensions $1 \times 2 \times 1$. We take a fixed time step of $\Delta t = 0.01$ and demonstrate that narrow banding does not prevent larger-than-CFL time steps. 1,008,187 particles are used to resolve the free surface in a narrow band of width $5\Delta x$. As in 2D, the particles capture highly-dynamic behavior of the free surface while the level

set is sufficient to represent the bulk fluid in the bottom half of the domain.

We also demonstrate our hybrid advection with a vortex shedding example (see Figure 14). The flow domain Ω is a 3×1 rectangle with circle of radius 0.05. We seed a band of particles of width .2 above the midline $y = .5$ for PolyPIC advection. Advection in the rest of the domain is done with BSLQB. The vorticity plot illustrates a seamless transition between the two advection schemes. The simulation was run with a grid resolution of $\Delta x = \frac{1}{255}$, CFL number of 4 (i.e. $\Delta t = \frac{4\Delta x}{v_{\max}}$), and inlet speed of 1.5.

7.2. BSLQB comparisons

We demonstrate improved resolution of flow detail with BSLQB compared to explicit semi-Lagrangian in a 2D example of smoke flowing past a circle (see Figure 16) and with a 2D spinning circle example (see Figure 6). Note that particles are only used for flow visualization and not for PolyPIC advection in these examples. BSLQB exhibits more energetic, turbulent flows than semi-Lagrangian advection. Notably, the BSLQB result breaks symmetry sooner. In Figure 16 we also examine the effect of extremal values of the λ parameter described in Equation (16). A zero value of λ is quite dissipative compared to a full value of $λ = 1$ for both semi-Lagrangian and BSLQB. As mentioned in Section 4.1, we generally found that keeping λ close to 1 provided the least dissipative behavior, while setting the value slightly less than 1 helped restore stability when necessary (one can also dynamically adjust this value over the course of a simulation, e.g. setting λ closer to 1 when vorticity is high to better resolve desirable details.). In Table 1 we examine the efficiency of semi-Lagrangian and BSLQB for various grid resolutions and values of λ. We see that BSLQB takes more time to run than semi-Lagrangian, and that time also increases slightly with higher values of λ. Similarly, in Table 2 we look at the stability of semi-Lagrangian and BSLQB for different values of λ and Δt. We observe that for $λ = 1$, both semi-Lagrangian and BSLQB are unstable when the time step is sufficiently small, though the instability vanishes when λ is reduced to 0.9. We illustrate this instability in Figure 20. In Figure 6, we initially set the angular velocity to 4 radians per second in a circle of radius .2 (with $\Omega = [0,1] \times [0,1]$). The simulation is run with $\Delta x = \frac{1}{511}$ and a $\Delta t = .02$ (CFL number of 3).

We also compare BSLQB with APIC and advection-reflection [ZNT18] in Figure 19. We again set the angular velocity of each circle to 4 radians per second in a circle of radius .2, and the simulation is run with $\Delta x = \frac{1}{127}$ and $\Delta t = .02$. Even at a lower resolution, both BSLQB and APIC exhibit more energetic flows than
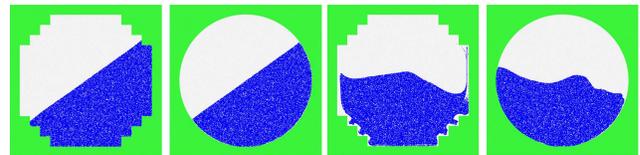
Figure 15: *Cut cell vs. voxelized domain. Using a cut cell domain (right) instead of a voxelized domain (left) yields marked improvements in simulation quality.*

$\Delta x =$	1/31	1/63	1/127	1/255	1/511
SL ($\lambda = 1$)	3.16	11.26	48.25	260.26	1329.88
SL ($\lambda = 0.5$)	2.62	11.05	48.07	252.68	1263.91
SL ($\lambda = 0$)	2.31	10.66	44.35	238.84	1193.98
BSL ($\lambda = 1$)	4.92	19.33	79.86	393.47	1838.78
BSL ($\lambda = 0.5$)	4.64	18.53	77.36	378.95	1777.42
BSL ($\lambda = 0$)	4.49	18.19	74.75	365.98	1707.63

Table 1: *Comparison of run times (in seconds) for SL and BSL for three values of the interpolation parameter λ in 2D. The example is that of Figure 16 with a fixed timestep of $\Delta t = 0.2$ at various grid resolutions out to a total time of 4.*

advection-reflection. BSLQB also shows more turbulent behavior compared to APIC.

In addition, in Figure 21 we compare our approach with Houdini's smoke tool using basic settings. We do this with the smoke past a sphere example from Figureřeffig:smokejets. Both simulations used a grid resolution of $\Delta x = \frac{1}{127}$. Houdini's simulation is much faster per frame (see Table 3 for our Smoke Jet's average time per frame). However, our method exhibits finer detail at similar resolution.

We examine the convergence behavior of BSLQB for the 2D Burgers' equation $\frac{D\mathbf{u}}{Dt} = \mathbf{0}$ with initial data $\mathbf{u}(\mathbf{x}) = \mathbf{x} \cdot (\mathbf{Ax})$ for $\mathbf{A} = \mathbf{R}\Lambda\mathbf{R}^T$ for diagonal Λ with entries 1 and .25 and rotation (of .1 radians) $\mathbf{R}$ (see Figure 17). We examine the convergence behavior under refinement in space and time with $\Delta t = \Delta x$. We compute the best fit line to the plot of the logarithm of the L^∞ norm of the error versus the logarithm of Δx for a number of grid resolutions. We observe slopes of approximately 2 for BSLQB with interpolation parameter $\lambda = 1$ and $\lambda = 1 - c\Delta x$ (with $c = 2.95$), indicating second order accuracy in space and time under refinement. We observe slopes of approximately 1 for explicit semi-Lagrangian, indicating first order.
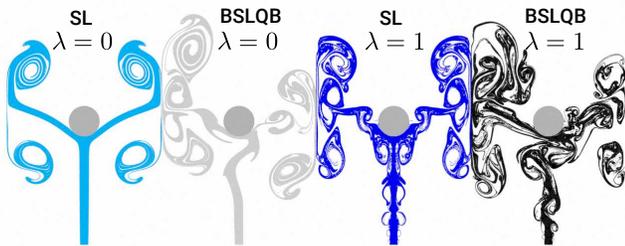
Figure 16: *Interpolation correction. BSLQB exhibits more fine-scale flow detail and vorticity than semi-Lagrangian for extremal values of interpolation parameter λ (Equation (16)). From left to right: semi-Lagrangian with $\lambda = 0$, BSLQB with $\lambda = 0$, semi-Lagrangian with $\lambda = 1$, BSLQB with $\lambda = 1$.*
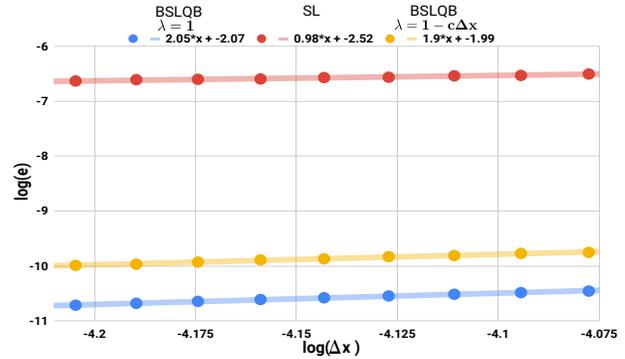
Figure 17: *Convergence. We compare explicit semi-Lagrangian (SL, red), with BSLQB (blue) and interpolation coefficient $\lambda = 1$ (Equation (16)) and BSLQB with interpolation coefficient $\lambda = 1 - c\Delta x$ (orange) to a final time of 1. We plot $\log(\Delta x)$ versus $\log(e)$ (where e is the infinity norm of the error) for a variety of grid resolutions Δx and compute the best fit lines. The slope of the line provides empirical evidence for the convergence rate of the method.*
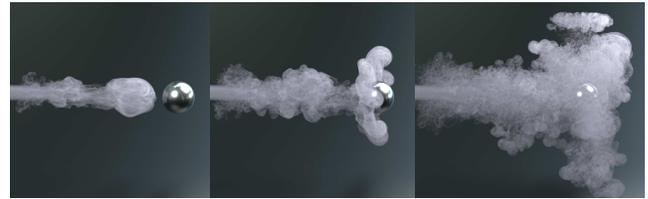
Figure 18: *Smoke jet. A plume of smoke is simulated with BSLQB. Zero normal velocity boundary conditions are enforced on the irregular boundary of the sphere inducing intricate flow patterns as the smoke approaches it.*

7.3. Cut cell examples

We demonstrate the ability of our cut cell method to produce detailed flows in complicated irregular domains for smoke and free surface water examples. Figure 4 demonstrates the subtle and visually interesting behavior that arises as two plumes of multicolored smoke flow to the center of a cubic domain colliding with a spherical boundary. We use $\Delta x = 1/63$ and $\Delta t = .02$. We demonstrate a more complex domain in Figure 8. Puffs of colored smoke with converging initial velocities are placed in a bunny shaped clear domain. We use a grid cell size of $1/127$ and a fixed time step of
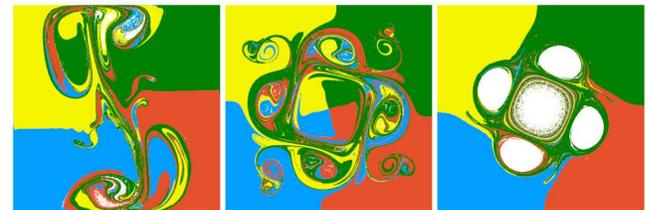
Figure 19: *BSLQB compared to other advection schemes. From left to right: BSLQB, APIC, and Advection-Reflection at time = 6.*

$\Delta t =$	.02	.001	.0005	.00025	.0001
SL ($\lambda = 1$)	✓	✓	✓	✓	✗
SL ($\lambda = 0.9$)	✓	✓	✓	✓	✓
SL ($\lambda = 0$)	✓	✓	✓	✓	✓
BSL ($\lambda = 1$)	✓	✓	✗	✗	✗
BSL ($\lambda = 0.9$)	✓	✓	✓	✓	✓
BSL ($\lambda = 0$)	✓	✓	✓	✓	✓

Table 2: *Comparison of stability for SL and BSL at three values of the interpolation parameter λ in 2D. The example is that of Figure 16 with a fixed resolution of $\Delta x = 1/127$ at various timesteps Δt out to a total time of 2. Stable simulations are marked with a check mark, while unstable simulations are marked with an x.*

$\Delta t = 0.01$ (CFL number > 1). In Figure 7, we demonstrate water splashing, while accurately conforming to the walls of an irregular domain defined as the interior of a large sphere and exterior of a small inner sphere. The spatial resolution of the domain is $\Delta x = 1/127$, and 30 particles per cell are seeded in the initial fluid shape. A minimum time step of $\Delta t = 0.001$ is enforced, which is often larger than the CFL condition. We also consider dam break simulations in rectangular domains with column obstacles (Figure 3) and a bunny obstacle (Figure 3). Both examples use a grid cell size of $\Delta x = 1/127$, 8 particles per cell and a fixed time step of $\Delta t = 0.003$. Lastly, we demonstrate the benefits of our cut cell formulation over a more simplified, voxelized approach in Figure 15. Notice the water naturally sliding in the cut cell domain compared with the jagged flow in the voxelized domain.

7.4. Performance considerations

The implementation of our method takes advantage of hybrid parallelism (MPI, OpenMP, and CUDA/OpenCL) on heterogeneous compute architectures in order to achieve practical runtime performance (see Table 3 for 3D example performance numbers). The spatial domain is uniformly divided into subdomains assigned to distinct MPI ranks, which distributes much of the computational load at the expense of synchronization overhead exchanging ghost information across ranks. On each rank, steps of our time integration loop such as BSLQB advection are multithreaded using OpenMP or CUDA when appropriate. The dominant costs per time step are the solution of the pressure projection system and, in the case of free surface simulation, assembly of the pressure system and its preconditioner. We permute Equation (25) so that each rank's degrees of freedom are contiguous in the solution vector then solve the system using AMGCL [Dem19] using the multi-GPU VexCL backend (or the OpenMP CPU backend on more limited machines). Using a strong algebraic multigrid preconditioner with large-degree Chebyshev smoothing allows our system to be solved to desired tolerance in tens of iterations, even at fine spatial resolution. An important step in minimizing the cost of system assembly is to scalably parallelize sparse matrix-matrix multiplication, for which we use the algorithm of Saad [Saa03]. In the future, we are interested in implementing load balancing strategies such as the simple speculative load balancing approach of [SHQL18], particularly for free surface flows. We note that our implementa-

Example	Seconds	# Particles	# Nodes
Smoke Jet (Fig. 18)	1,212	12,502,349	$2*127^3$
Multiple Jets (Fig. 4)	53	25,004,699	63^3
Bunny Smoke (Fig. 8)	160	24,000,000	127^3
Smoke Spheres* (Fig. 2)	428	64,000,000	255^3
Narrow Band (Fig. 12)	396	1,008,187	$2*63^3$
Water Globe (Fig. 7)	242	524,415	127^3
Dam Break (Fig. 3)	870	3,251,409	$2*127^3$
Bunny Dam Break (Fig. 5)	1,171	4,797,535	$2*127^3$

Table 3: *Average time per frame (in seconds) for each of the 3D examples shown in the paper. Examples were run on workstations with 16-core CPUs running at 2.20 GHz, except for the smoke spheres example, which was run on a cluster equipped with CPUs running at 3.07 GHz and Nvidia Tesla V100 GPUs which were used for the linear solves.*

tion enables high-resolution simulations such as that in Figure 2 at relatively modest computational cost (see Table 3).
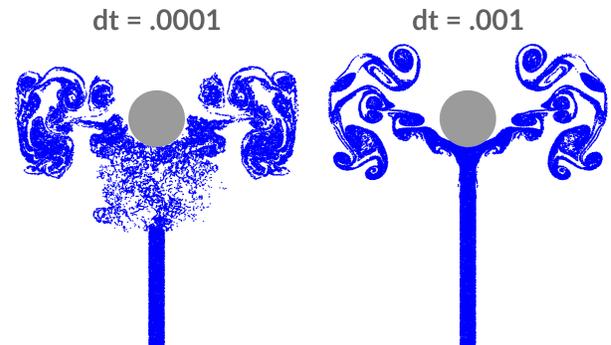
Figure 20: *Instability: When λ is close (or equal) to 1, simulations can become unstable for smaller values of Δt. We show this here with a simulation of smoke against a circle at a spatial resolution of $\Delta x = 1/127$. This is seen for both semi-Lagrangian and BSLQB.*

8. Discussion and limitations

Our approach has several key limitations that could be improved. First, our adoption of collocated multiquadratic velocity and staggered multilinear pressure is a significant departure from most fluid solvers utilized in graphics applications. We note that BSLQB and BSLQB/PolyPIC could be used with a MAC grid; however, each velocity face component would have to be solved for individually. Another drawback for our multiquadratic velocity and multilinear pressure formulation is that it gives rise to a very wide pressure system stencil consisting of 49 non-zero entries per row in 2D and 343 in 3D. Collocated approaches that make use of multilinear velocities and constant pressure give rise to 9 (2D) and 27 (3D) entries per row [ZZS*17], however they do not allow for C^1 continuity and require spurious pressure mode damping. Our wide stencils likely negatively affect the efficacy of preconditioning techniques as well, however we were very pleased with the efficiency of the
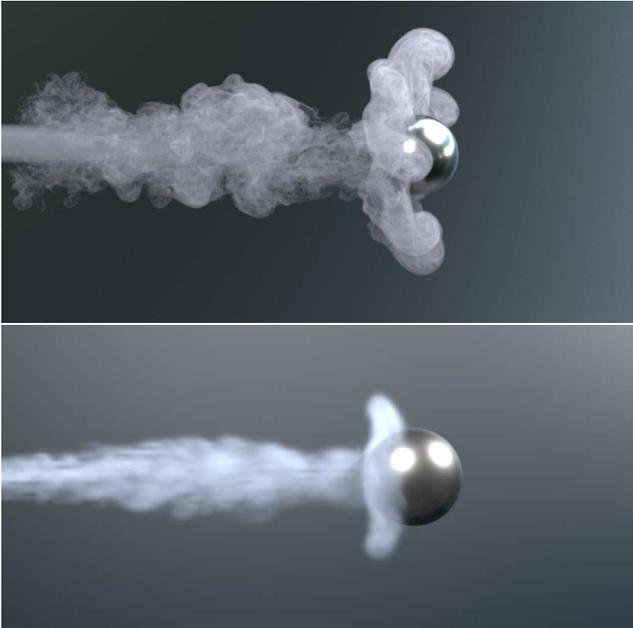
Figure 21: *Comparison with Houdini Smoke. We compare the smoke jet of Figure 18 with Houdini's Billowy Smoke using basic settings at similar grid resolution. Our method displays more detail without additional disturbance or turbulence.*

AMGCL [Dem19] library. Also, while the use of mass lumping in Equation (27) is necessary to ensure a sparse pressure projection system, Botella [Bot02] note that this has been shown to degrade accuracy. In fact, Botella [Bot02] introduce a sparse approximate inverse to the full mass matrix to avoid dense systems of equations in the pressure solve without degrading accuracy. Split cubic interpolation, which approximates similar systems with tridiagonal ones could also possibly be used for this [Hua94]. Adoption of one of these approaches with our formulation would be an interesting area of future work. Also, we note that the more sophisticated transition criteria for narrow banding techniques in Sato et al. [SWT*18] could naturally be used with our method. Additionally, the free surface boundary in our approach is voxelized. In future work, we would like to use non-voxelized cut cell boundaries. Finally, we note that the work of Zehnder et al. [ZNT18; NZT19] could be easily applied to our technique to further reduce dissipation since it is based on the Chorin [Cho67] splitting techniques (Equations (8)-(10)) that we start from.

9. Acknowledgments

References

[AN05] ANGELIDIS, A. and NEYRET, F. "Simulation of smoke based on vortex filament primitives". *Proc 2005 ACM SIGGRAPH/Eurographics Symp Comp Anim*. ACM. 2005, 87–96 4.

[ATW13] ANDO, R., THÜREY, N., and WOJTAN, C. "Highly adaptive liquid simulations on tetrahedral meshes". *ACM Trans Graph* 32.4 (2013), 103:1–103:10. DOI: 10.1145/2461912.2461982 2, 6.

[BB08] BATTY, C. and BRIDSON, R. "Accurate viscous free surfaces for buckling, coiling, and rotating liquids". *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim* (2008), 219–228 2, 4.

[BB12] BOYD, L. and BRIDSON, R. "MultiFLIP for energetic two-phase fluid simulation". *ACM Trans Graph* 31.2 (2012), 16:1–16:12 9.

[BBB07] BATTY, C., BERTAILS, F., and BRIDSON, R. "A fast variational framework for accurate solid-fluid coupling". *ACM Trans Graph* 26.3 (2007), 2, 4.

[BGV09] BELYTSCHKO, T., GRACIE, R., and VENTURA, G. "A review of extended/generalized finite element methods for material modeling". *Mod Sim Mat Sci Eng* 17.4 (2009), 043001 2, 4, 8.

[BH07] BAZILEVS, Y. and HUGHES, T. "Weak imposition of Dirichlet boundary conditions in fluid mechanics". *Comp Fluids* 36.1 (2007), 12–26 4.

[BM82] BATES, J. and MCDONALD, A. "Multiply-upstream, semi-Lagrangian advective schemes: Analysis and application to a multi-level primitive equation model". *Monthly Weather Review* 110.12 (1982), 1831–1842 4.

[Bot02] BOTELLA, O. "On a collocation B-spline method for the solution of the Navier–Stokes equations". *Comp Fl* 31.4-7 (2002), 397–420 5, 8, 12.

[BR86] BRACKBILL, J. and RUPPEL, H. "FLIP: A method for adaptively zoned, Particle-In-Cell calculations of fluid flows in two dimensions". *J Comp Phys* 65 (1986), 314–343 2.

[Bre10] BRESSAN, A. "Isogeometric regular discretization for the Stokes problem". *IMA J Num Anal* 31.4 (2010), 1334–1356 2, 4.

[Bri08] BRIDSON, R. *Fluid simulation for computer graphics*. Taylor & Francis, 2008 2, 4.

[Che95] CHERNYAEV, E. *Marching cubes 33: Construction of topologically correct isosurfaces*. Tech. rep. 1995 8.

[Cho67] CHORIN, A. "A numerical method for solving incompressible viscous flow problems". *J Comp Phys* 2.1 (1967), 12–26 2, 3, 5, 12.

[CIR52] COURANT, R., ISAACSON, E., and REES, M. "On the solution of nonlinear hyperbolic differential equations by finite differences". *Comm Pure App Math* 5.3 (1952), 243–255 4.

[CK12] CHENEY, W. and KINCAID, D. *Numerical mathematics and computing*. Cengage Learning, 2012 4, 7.

[CKP*16] CHERN, A., KNÖPPEL, F., PINKALL, U., et al. "Schrödinger's smoke". *ACM Trans Graph (TOG)* 35.4 (2016), 77 4.

[CMK15] CHENTANEZ, N., MÜLLER, M., and KIM, T. "Coupling 3D eulerian, heightfield and particle methods for interactive simulation of large scale liquid phenomena". *IEEE Trans Vis Comp Graph* 21.10 (2015), 1116–1128 3, 8.

[CWB01] CHENG, F., WANG, X., and BARSKY, B. "Quadratic B-spline curve interpolation". *Comp Math App* 41.1-2 (2001), 39–50 4, 7.

[dBoo78] DE BOOR, C. *A Practical Guide to Splines*. Springer, 1978 2.

[Dem19] DEMIDOV, D. "AMGCL: An Efficient, Flexible, and Extensible Algebraic Multigrid Implementation". *Lobachevskii Journal of Mathematics* 40.5 (May 2019), 535–546. ISSN: 1818-9962. DOI: 10.1134/S1995080219050056. URL: https://doi.org/10.1134/S1995080219050056 11, 12.

[EB14] EDWARDS, E. and BRIDSON, R. "Detailed water with coarse grids: combining surface meshes and adaptive Discontinuous Galerkin". *ACM Trans Graph* 33.4 (2014), 136:1–136:9 5.

[ENGF03] ENRIGHT, D., NGUYEN, D., GIBOU, F., and FEDKIW, R. "Using the particle level set method and a second order accurate pressure boundary condition for free surface flows". *ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference*. ASMEDC. 2003, 337–342 4.

[Eva10] EVANS, L. *Partial differential equations*. Providence, R.I.: American Mathematical Society, 2010 3, 6.

[FAW*16] FERSTL, F., ANDO, R., WOJTAN, C., et al. "Narrow band flip for liquid simulations". *Comp Graph For*. Vol. 35. 2. Wiley Online Library. 2016, 225–232 3, 8.

[FF01] FOSTER, N. and FEDKIW, R. "Practical animation of liquids". *Proc 28th SIGGRAPH*. ACM. 2001, 23–30 2.

[FF98] FALCONE, M. and FERRETTI, R. "Convergence analysis for a class of high-order semi-Lagrangian advection schemes". *SIAM J Num Anal* 35.3 (1998), 909–940 4.

[FGG*17] FU, C., GUO, Q., GAST, T., et al. "A Polynomial Particle-in-cell Method". *ACM Trans Graph* 36.6 (Nov. 2017), 222:1–222:12 2–4, 7.

[FM96] FOSTER, N. and METAXAS, D. "Realistic animation of liquids". *Graph Mod Imag Proc* 58 (1996), 471–483 2.

[FSJ01] FEDKIW, R., STAM, J., and JENSEN, H. "Visual simulation of smoke". *SIGGRAPH*. ACM. 2001, 15–22 2–4.

[FWD14] FERSTL, FLORIAN, WESTERMANN, RÜDIGER, and DICK, CHRISTIAN. "Large-scale liquid simulation on adaptive hexahedral grids". *IEEE transactions on visualization and computer graphics* 20.10 (2014), 1405–1417 5.

[GHM*20] GAGNIERE, S., HYDE, D., MARQUEZ-RAZON, A., et al. *Supplementary Technical Document*. Tech. rep. 2020 8.

[Har64] HARLOW, F. "The particle-in-cell method for numerical solution of problems in fluid dynamics". *Meth Comp Phys* 3 (1964), 319–343 2.

[HCB05] HUGHES, T.J.R., COTTRELL, J.A., and BAZILEVS, Y. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". *Comp Meth App Mech Eng* 194.39,41 (2005), 4135–4195 4.

[Hua94] HUANG, C. "Semi-Lagrangian advection schemes and Eulerian WKL algorithms". *Monthly Weather Review* 122.7 (1994), 1647–1658 12.

[Hug00] HUGHES, T. *The finite element method : linear static and dynamic finite element analysis*. Dover, 2000 2, 4.

[HW65] HARLOW, F. and WELCH, E. "Numerical Calculation of Time Dependent Viscous Flow of Fluid with a Free Surface". *Phys Fluid* 8.12 (1965), 2182–2189 2.

[IAAT12] IHMSEN, M., AKINCI, N., AKINCI, G., and TESCHNER, M. "Unified spray, foam and air bubbles for particle-based fluids". *Vis Comp* 28.6-8 (2012), 669–677 2.

[JSS*15] JIANG, C., SCHROEDER, C., SELLE, A., et al. "The Affine Particle-In-Cell Method". *ACM Trans Graph* 34.4 (2015), 51:1–51:10 2, 4.

[KBT17] KOSCHIER, D., BENDER, J., and THUEREY, N. "Robust eXtended Finite Elements for complex cutting of deformables". *ACM Trans Graph* 36.4 (2017), 55:1–55:13. DOI: 10 . 1145 / 3072959 . 3073666 2, 4, 8.

[Kim98] KIM, E. "A mixed Galerkin method for computing the flow between eccentric rotating cylinders". *Int J Num Meth Fl* 26.8 (1998), 877–885 5.

[KLLR05] KIM, B., LIU, Y., LLAMAS, I., and ROSSIGNAC, J. "FlowFixer: Using BFECC for Fluid Simulation". *Proc Eurograph Conf Nat Phen*. Eurographics Association, 2005, 51–56 3.

[KLLR06] KIM, B., LIU, Y., LLAMAS, I., and ROSSIGNAC, J. "Advections with significantly reduced dissipation and diffusion". *IEEE Trans Viz Comp Graph* 13.1 (2006), 135–144 3.

[KMS99] KRAVCHENKO, A., MOIN, P., and SHARIFF, K. "B-spline method and zonal grids for simulations of complex turbulent flows". *J Comp Phys* 151.2 (1999), 757–789 5.

[KSK08] KIM, D., SONG, O., and KO, H. "A semi-lagrangian CIP fluid solver without dimensional splitting". *Comp Graph For*. Vol. 27. 2. Wiley Online Library. 2008, 467–475 3.

[KW90] KUO, H. and WILLIAMS, R. "Semi-Lagrangian solutions to the inviscid Burgers equation". *Monthly Weather Review* 118.6 (1990), 1278–1288 6.

[LBB17] LARIONOV, E., BATTY, C., and BRIDSON, R. "Variational stokes: A unified pressure-viscosity solver for accurate viscous liquids". *ACM Trans Graph* 36.4 (2017), 2, 4.

[Leo79] LEONARD, B. "A stable and accurate convective modelling procedure based on quadratic upstream interpolation". *Computer methods in applied mechanics and engineering* 19.1 (1979), 59–98 3.

[MCP*09] MULLEN, P., CRANE, K., PAVLOV, D., et al. "Energy-preserving integrators for fluid animation". *ACM Trans Graph (TOG)*. Vol. 28. 3. ACM. 2009, 38 4.

[MCPN08] MOLEMAKER, J., COHEN, J., PATEL, S., and NOH, J. "Low viscosity flow simulations for animation". *Proc 2008 ACM SIGGRAPH/Eurographics Symp Comp Anim*. Eurographics Association. 2008, 9–18 3.

[MK96] MAKAR, P. and KARPIK, S. "Basis-spline interpolation on the sphere: Applications to semi-Lagrangian advection". *Monthly Weather Review* 124.1 (1996), 182–199 4.

[NSB*18] NIELSEN, MICHAEL B, STAMATELOS, KONSTANTINOS, BOJSEN-HANSEN, MORTEN, et al. "A collocated spatially adaptive approach to smoke simulation in bifrost". *ACM SIGGRAPH 2018 Talks*. 2018, 1–2 2, 4.

[NZT19] NARAIN, R., ZEHNDER, J., and THOMASZEWSKI, B. "A Second-order advection-reflection solver". *Proc ACM Comput Graph Interact Tech* 2.2 (July 2019) 4, 12.

[PK05] PARK, S. and KIM, M. "Vortex fluid for gaseous phenomena". *Proc 2005 ACM SIGGRAPH/Eurograph Symp Comp Anim*. ACM. 2005, 261–270 4.

[PS84] PUDYKIEWICZ, J. and STANIFORTH, A. "Some properties and comparative performance of the semi-Lagrangian method of Robert in the solution of the advection-diffusion equation". *Atmosphere-Ocean* 22.3 (1984), 283–308 4, 6.

[QZG*19] QU, Z., ZHANG, X., GAO, M., et al. "Efficient and conservative fluids using bidirectional mapping". *ACM Trans. Graph.* 38.4 (2019) 4.

[RC12] RÜBERG, T. and CIRAK, F. "Subdivision-stabilised immersed b-spline finite elements for moving boundary flows". *Comp Meth App Mech Eng* 209 (2012), 266–283 4.

[RCLM98] RIISHØJGAARD, L., COHN, S., LI, Y., and MÉNARD, R. "The use of spline interpolation in semi-Lagrangian transport models". *Monthly Weather Review* 126.7 (1998), 2008–2016 4, 7.

[Rob81] ROBERT, A. "A stable numerical integration scheme for the primitive meteorological equations". *Atm Ocean* 19.1 (1981), 35–46 2–4.

[Saa03] SAAD, Y. *Iterative Methods for Sparse Linear Systems*. 2nd. USA: Society for Industrial and Applied Mathematics, 2003. ISBN: 0898715342 11.

[Saw63] SAWYER, J. "A semi-Lagrangian method of solving the vorticity advection equation". *Tellus* 15.4 (1963), 336–342 3, 4.

[SBIA18] SATO, T., BATTY, C., IGARASHI, T., and ANDO, R. "Spatially adaptive long-term semi-Lagrangian method for accurate velocity advection". *Comp Vis Med* 4.3 (2018), 223–230 4.

[SC91] STANIFORTH, A. and CÔTÉ, J. "Semi-Lagrangian integration schemes for atmospheric models-A review". *Monthly weather review* 119.9 (1991), 2206–2223 4.

[SCS94] SULSKY, D., CHEN, Z., and SCHREYER, H. "A particle method for history-dependent materials". *Comp Meth App Mech Eng* 118.1 (1994), 179–196 2.

[SDG*19] SCHNEIDER, TESEO, DUMAS, JÉRÉMIE, GAO, XIFENG, et al. "Poly-Spline Finite-Element Method". *ACM Transactions on Graphics (TOG)* 38.3 (2019), 1–16 5.

[SFK*08] SELLE, A., FEDKIW, R., KIM, B., et al. "An unconditionally stable MacCormack method". *J Sci Comp* 35.2-3 (2008), 350–371 3.

[SHQL18] SHAH, C., HYDE, D., QU, H., and LEVIS, P. "Distributing and Load Balancing Sparse Fluid Simulations". *Computer Graphics Forum* 37.8 (2018): *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2018 (SCA 2018)*. Ed. by BEELER, T. and THUEREY, N., 35–46. DOI: `10.1111/cgf.13510` 11.

[SIBA17] SATO, T., IGARASHI, T., BATTY, C., and ANDO, R. "A long-term semi-lagrangian method for accurate velocity advection". *SIGGRAPH Asia 2017 Tech Briefs*. ACM. 2017, 5 4.

[SKB08] STEFFEN, M., KIRBY, R., and BERZINS, M. "Analysis and reduction of quadrature errors in the material point method (MPM)". *Int J Numer Meth Eng* 76.6 (2008), 922–948 2.

[SKK09] SONG, O., KIM, D., and KO, H. "Derivative particles for simulating detailed movements of fluids". *IEEE Trans Vis Comp Graph* (2009), 247–255 3.

[SRF05] SELLE, A., RASMUSSEN, N., and FEDKIW, R. "A vortex particle method for smoke, water and explosions". *ACM Trans Graph (TOG)*. Vol. 24. 3. ACM. 2005, 910–914 4.

[SSHT14] SCHROEDER, C., STOMAKHIN, A., HOWES, R., and TERAN, J. "A second order virtual node algorithm for Navier-Stokes flow problems with interfacial forces and discontinuous material properties". *J Comp Phys* 265 (2014), 221–245. DOI: `https://doi.org/10.1016/j.jcp.2014.01.051` 2–4, 8.

[SSJ*14] STOMAKHIN, A., SCHROEDER, C., JIANG, C., et al. "Augmented MPM for phase-change and varied materials". *ACM Trans Graph* 33.4 (2014), 138:1–138:11 2.

[Sta99] STAM, J. "Stable Fluids". *Siggraph*. Vol. 99. 1999, 121–128 2, 3.

[STK*07] SHARIF, E., TONG, Y., KANSO, E., et al. "Stable, circulation-preserving, simplicial fluids". *ACM Trans Graph (TOG)* 26.1 (2007), 4 4.

[SU94] STEINHOFF, J. and UNDERHILL, D. "Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings". *Phys Fl* 6.8 (1994), 2738–2744 3.

[SWT*18] SATO, T., WOJTAN, C., THUEREY, N., et al. "Extended narrow band FLIP for liquid simulations". *Comp Graph For* (2018) 3, 8, 12.

[TH73] TAYLOR, C. and HOOD, P. "A numerical solution of the Navier-Stokes equations using the finite element technique". *Comp & Fl* 1.1 (1973), 73–100 2.

[TP11] TESSENDORF, J. and PELFREY, B. "The characteristic map for fast and efficient VFX fluid simulations". 2011 4.

[WL10] WANG, J. and LAYTON, A. "New numerical methods for Burgers' equation based on semi-Lagrangian and modified equation approaches". *App Num Math* 60.6 (2010), 645–657 4.

[WP10] WEISSMANN, S. and PINKALL, U. "Filament-based smoke with vortex shedding and variational reconnection". *ACM Trans Graph (TOG)*. Vol. 29. 4. 2010, 115 4.

[XK01] XIU, D. and KARNIADAKIS, G. "A semi-Lagrangian high-order method for Navier–Stokes equations". *J Comp Phys* 172.2 (2001), 658–684 3.

[YXU01] YABE, T., XIAO, F., and UTSUMI, T. "The constrained interpolation profile method for multiphase analysis". *J Comp Phys* 169 (2001), 556–593 3.

[ZB05] ZHU, Y. and BRIDSON, R. "Animating sand as a fluid". *ACM Trans Graph* 24.3 (2005), 965–972 2, 4.

[ZBG15] ZHANG, X., BRIDSON, R., and GREIF, C. "Restoring the missing vorticity in advection-projection fluid solvers". *ACM Trans Graph (TOG)* 34.4 (2015), 52 4.

[Zha05] ZHAO, H. "A fast sweeping method for eikonal equations". *Math Comp* 74.250 (2005), 603–627 9.

[ZNT18] ZEHNDER, J., NARAIN, R., and THOMASZEWSKI, B. "An advection-reflection solver for detail-preserving fluid simulation". *ACM Trans Graph (TOG)* 37.4 (2018), 85 4, 9, 12.

[ZZS*17] ZHANG, F., ZHANG, X., SZE, Y., et al. "Incompressible material point method for free surface flow". *J Comp Phys* 330.C (2017), 92–110 11.