A Unified Approach to Monolithic Solid-Fluid Coupling of Sub-Grid and More Resolved Solids

David A. B. Hyde**, Ronald Fedkiw*

Stanford University, 353 Serra Mall, Gates Computer Science Room 207, Stanford, CA 94305

Abstract

We present a novel numerical approach to monolithic two-way fluid-structure interaction. Our formulation applies to both sub-grid solids that may be significantly smaller than a single computational grid cell as well as more resolved solid bodies. Importantly, we remove the non-physical ansatz of velocity equilibration for sub-grid bodies, allowing for physically accurate behavior even though the boundary layer is drastically under-resolved or typically even completely absent. Instead, we obtain closure by incorporating a physical model for drag. Our coupling system has numerically advantageous properties, including symmetric positive-definiteness, and is extensible to various interpolation schemes, enabling it to capture the exact solution in the case of neutral buoyancy. We provide a number of numerical examples illuminating the advantages and limitations of our approach. We close with remarks about future possibilities for pursuing advanced sub-grid modeling as well as incorporating machine learning techniques into drag modeling.

Keywords: Solid-Fluid Coupling, Sub-Grid Modeling, Drag, Incompressible Flow

1. Introduction

Solid-fluid coupling is an increasingly important topic in physical simulation as recent advances in computer hardware, networking, and numerical methods enable the computational physics community to pursue progressively more challenging simulation problems. Accurate two-way fluid-structure interaction, which captures the mutual interactions between fluid and solid components in a simulation, is key to better understanding a wide variety of phenomena. Such phenomena may occur under significantly different physical conditions. For instance, one may wish to accurately couple near-laminar water flow with a buoyant object. Or, in a very different physical regime, one might consider simulating the hypervelocity impact of solid bodies immersed in a turbulent flow field. In general, considerations such as the amount of structural deformation, the turbulence of the flow, the size and shape of solids, and the velocity of solids and fluids guide the development and selection of appropriate simulation techniques.

In this paper, our particular focus is on sub-grid solid-fluid coupling, for example, as occurs in underbody blasts and blast-structure interaction. In the underbody blast problem, an explosive device is detonated underneath a vehicle, and the resulting behavior of the vehicle body as well as the flow field that rapidly develops is studied; in particular, care must be given to small, high-velocity shrapnel that emanates from both the vehicle and the explosive. Blast-structure interaction considers the effects, such as spallation, of an explosive blast impinging on a building or other structure. Experiments have demonstrated the importance of small solid particles in these scenarios [31, 46]. However, even with recent advances in hardware, networking, algorithms, etc., resolving the interface between these small solid particles and the fluid is intractable; in fact, for problems with disparate scales, it may be impractical to have *any* grid or sample points in the solid-fluid boundary layer. Sub-grid solids exacerbate these issues, especially when the solids are so small that there are multiple solids within a single computational cell of the fluid grid.

A standard approach to two-way solid-fluid coupling is to consider the solid velocities as boundary conditions on the fluid and to integrate the fluid pressure force along the boundary of the solid [44, 14,

*{dabh,fedkiw}@cs.stanford.edu, Stanford University
**Corresponding author

5, 20, 29]. Explicitly applying these forces, as opposed to using an implicitly coupled system, results in a so-called partitioned scheme, which is straightforward to implement. However, disadvantages of partitioned approaches include stability issues, such as the added-mass instability [11, 2], as well as possibly costly performance due to repeated iteration of the solid and fluid solves with no guarantees on convergence. Monolithic schemes, where coupled solid and fluid updates are solved for simultaneously using an implicitly coupled system, offer remedies to these aspects of partitioned schemes.

Monolithic two-way solid-fluid coupling schemes have been considered in a number of previous works [38, 37, 18, 16, 33]. In [38], the implicit, monolithic scheme was indefinite, which was remedied by the symmetric positive-definite (SPD) system of [37], which was in turn extended in [18] to handle compressible flow. Recently, [33] integrated this scheme with a framework for positivity preservation in compressible flow, and also opened the investigation into incorporating sub-grid solids into the coupling scheme. The implicit, monolithic, two-way solid-fluid coupling scheme we develop in this paper remedies certain deficiencies of and extends upon these previous works. Two recent interesting papers use a variational formulation [3] in order to simulate fluid in cut cells, including thin gaps [1] and two-way solid-fluid coupling [45]; however, it is unclear how to extend their topological approach to sub-grid bodies where the volume integral rather than the surface integral seems more appropriate. We also highlight [34], another treatment of thin gaps wherein the fluid degrees of freedom are discretized on the solid surface mesh, allowing for sub-fluid grid modeling.

Consider a small sub-grid body moving to the right, as in Figure 1a. A velocity equilibration assumption between the solid and fluid would transfer an inordinate amount of momentum from this solid to the fluid that co-occupies the grid cell containing it, when in reality there may be only a small boundary layer around the solid transferring a much smaller amount of momentum such that most of the fluid in that computational cell remains largely unaffected by the solid. Moreover, this non-physical transfer of momentum happens in a single time step, even when that time step is infinitesimally small. For more resolved solids, such as those depicted in Figures 1b, 1c, and 1d, it makes more sense to consider velocity equilibration. We note that [33] discussed non-physical velocity equilibration for sub-grid bodies and suggested an ad-hoc parameter that allows one to alleviate the equilibration by controlling the amount of momentum transfer between the fluid and solid, albeit with no strong connection to underlying physical equations. In contrast, we show that a physical drag law can be introduced into the equations, providing a physically-grounded alternative closure to velocity equilibration. Notably, this physical model for drag can be incorporated into the implicit coupling solve while still achieving a symmetric positive-definite system.

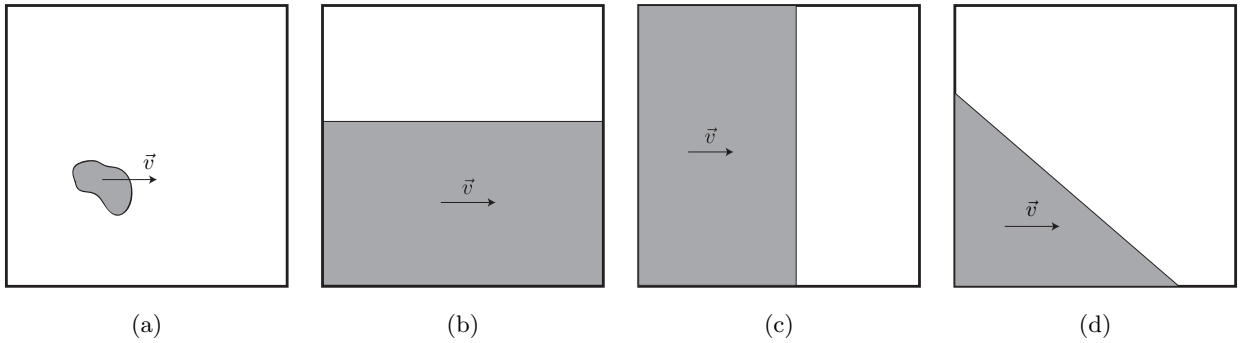


Figure 1: One has to consider a number of scenarios when modeling sub-grid and under-resolved solids. (a) A small sub-grid solid should not lose an inordinate amount of momentum by fully equilibrating with the fluid in its computational cell. (b) A solid moving tangentially similarly should not equilibrate, but rather only transfer a fixed amount of momentum tangentially via a viscous boundary layer. (c) Here, the fluid has no choice but to equilibrate its normal velocity as the solid pushes through it. (d) Here, the normal velocity should equilibrate but fluid may still flow tangentially unequilibrated up and to the left.

2. Governing Equations

The present work focuses on inviscid incompressible flow and rigid bodies for sake of exposition, though we note that our methodology is generalizable to other scenarios. Accordingly, we model the fluid's motion

using the inviscid incompressible Navier-Stokes equations,

$$\begin{cases} \rho\left(\frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\,\vec{u}\right) = -\nabla p + \vec{f} \\ \nabla\cdot\vec{u} = 0, \end{cases} \tag{1}$$

which are affected by the presence of structures. Here, ρ is (a constant) density, $\vec{u}$ is the fluid velocity, p is pressure, and $\vec{f}$ represents external body forces such as gravity. We discretize the fluid using a standard Marker and Cell (MAC) grid [21] and denote by $\vec{u}_f$ the vector of well-defined fluid velocity samples on that grid. We solve the discretized form of Equation 1 based on the projection method of [6], making a first-order time approximation. The projection method divides the time integration into two steps. First, an intermediate fluid velocity is computed, ignoring pressure terms:

$$\vec{u}_f^* = \vec{u}_f^n - \Delta t\left(\vec{u}_f^n\cdot\nabla\right)\vec{u}_f^n + \Delta t\rho^{-1}\vec{f}. \tag{2}$$

Subsequently, incompressibility is enforced via the implicit pressure projection

$$\vec{u}_f^{n+1} = \vec{u}_f^* - \Delta t\rho^{-1}\nabla p. \tag{3}$$

A semi-Lagrangian advection scheme [35] is applied to compute the convective terms in Equation 2, filling ghost cells inside solids and outside the domain as appropriate. Advection is performed using the divergence-free $\vec{u}_f^n$. For a purely fluid domain, incompressibility is enforced as usual by substituting Equation 3 into the discretized incompressibility condition of Equation 1 to obtain

$$\nabla\cdot\left(\Delta t\rho^{-1}\nabla p\right) = \nabla\cdot\vec{u}_f^*, \tag{4}$$

and solving for p.

We only consider rigid bodies, noting that works such as [37, 30, 45] demonstrate several techniques for extending monolithic coupling systems to handle deformable or reduced deformable bodies. Rigid body dynamics are governed by the classical Newton-Euler equations. The force and torque exerted on an immersed rigid body are evaluated around the body's boundary Γ:

$$F = \int_\Gamma (-p\hat{n})\,ds + f_e, \quad T = \int_\Gamma (\vec{x} - \vec{x}_{\text{com}})\times(-p\hat{n})\,ds + \tau_e, \tag{5}$$

where $\hat{n}$ is a unit surface normal, $\vec{x}$ is a point on Γ, $\vec{x}_{\text{com}}$ is the body's center of mass, f_e and τ_e are any external body forces or torques, and p is fluid pressure as above. Solid positions and orientations are updated explicitly in parallel with the semi-Lagrangian advection of the fluid, noting this also changes world space inertia tensors (and thus angular velocities in order to keep angular momentum unchanged). Then body forces such as gravity are applied to obtain time t^* solid quantities. After time t^* fluid and solid quantities are obtained, a monolithic coupled system may be solved for the solid and fluid together, which we formulate to give solutions for the linear and angular solid velocities as well as the fluid pressure. This coupled system replaces Equation 4, and the resulting pressure is used to update Equation 3. Optionally, for a more Newmark-oriented approach, a similar coupled system would be solved in order to obtain temporary fluid and solid velocities for use in fluid advection and the solid position and orientation update (see e.g. [37]).

3. Sub-Grid Solid-Fluid Coupling

Given a domain Ω, the fluid pressure gradient gives rise to a net force $\int_\Omega \nabla p dV$, where V is volume and ∇p is the pointwise pressure gradient. On a MAC grid, one discretizes the momentum in each Cartesian direction independently. For example, for the x-direction dual cells, the momentum associated with the scalar velocity u is influenced by $\int_\Omega p_x dV$. Thus, under suitable assumptions, the net pressure force associated with an x-direction dual cell D_i of a MAC grid is $A_i\left(p_{R,i} - p_{L,i}\right)$, where $A_i = V_i/\Delta x$ is the lateral face area of D_i, and $p_{R,i}$ and $p_{L,i}$ are the average pressures on the right and left faces of D_i. When a dual cell contains both fluid and solid components, this total force should be distributed between the components. For example, one may write

$$A_{F,i}\left(p_{R,i} - p_{L,i}\right) + A_{S,i}\left(p_{R,i} - p_{L,i}\right), \tag{6}$$

for some $A_{F,i}$ and $A_{S,i}$ associated with the fluid and the solid, respectively, with $A_{F,i}+A_{S,i}=A_i$. Moreover, if there are multiple distinct solids within D_i, $A_{S,i}$ can be partitioned even further[1], e.g.

$$A_{F,i}\left(p_{R,i}-p_{L,i}\right)+\sum_j A_{S,i}^j\left(p_{R,i}-p_{L,i}\right). \tag{7}$$

Typically, on a MAC grid, one assumes a linear pressure profile within each dual cell, which is consistent with a hydrostatic solution where a fluid at rest stays at rest with the pressure gradient exactly canceling gravity. Thus, given the average pressures on two opposing faces of a dual cell, one typically computes a pressure gradient that is assumed to be constant throughout that dual cell. If one instead chooses to model sub-cell variations in the pressure gradient, then care needs to be taken in order to ensure that these sub-cell gradients are properly balanced by gravity in order to obtain the correct hydrostatic solution. For example, [8] illustrated that spurious waves develop if one does not correctly treat MAC grid pressure gradients in the correct sub-cell fashion near the free surface.

Given the assumption that the pressure gradient is constant in a dual cell, one can partition the net force in a dual cell amongst different fluid sub-regions by simply integrating the pressure gradient separately in each sub-region. Moreover, if one of those fluid sub-regions were replaced with a different type of fluid and/or solid, the net force on that different material would need to be identical to the previously-computed net force using the original fluid; otherwise, using a different force would fail to correctly obtain neutral buoyancy in the case where the new material has the same density as the displaced fluid—this is Archimedes' principle. In the case of our sub-grid solids, this means we can compute $A_{F,i}$ and the $A_{S,i}^j$ for Equation 7 by integrating over the appropriate volumetric regions and multiplying by the assumed-to-be-constant pressure gradient. The $A_{F,i}$ and the $A_{S,i}^j$ simply partition the pressure gradient force amongst the various materials in the cell.

As discussed above, integrating p_x over D_i gives $A_i\left(p_{R,i}-p_{L,i}\right)=V_i\left(p_{R,i}-p_{L,i}\right)/\Delta x$, illustrating that $\left(p_{R,i}-p_{L,i}\right)/\Delta x$ is the assumed constant pressure gradient in D_i. Multiplying and dividing Equations 6 and 7 by Δx illustrates that $\Delta x A_{F,i}$ and $\Delta x A_{S,i}^j$ are the appropriate terms for partitioning this pressure gradient. In order to correctly obtain neutral buoyancy, these should be the volumes of the associated regions, i.e. $V_{F,i}=\Delta x A_{F,i}$ and $V_{S,i}^j=\Delta x A_{S,i}^j$. It is constructive to realize that this definition does not differentiate between the actual shape of a solid and a rectangular prism with the same volume (see Figure 2). This emphasizes the importance of effective cross-sectional area in the MAC grid formulation. With this choice of $A_{F,i}$ and $A_{S,i}^j$,

$$\frac{V_{F,i}}{V_i}=\frac{\Delta x A_{F,i}}{\Delta x A_i}=\frac{A_{F,i}}{A_i}, \qquad \frac{V_{S,i}^j}{V_i}=\frac{\Delta x A_{S,i}^j}{\Delta x A_i}=\frac{A_{S,i}^j}{A_i}, \tag{8}$$

i.e. the volume fractions and face area fractions are equivalent. Throughout the paper, we will refer to the conceptual equivalence of between Figure 2 Left and Figure 2 Middle as the *rectangular prism construction*, stressing that it is a natural consequence of a few assumptions on the MAC grid, including the partitioning of momentum between different types of dual cells, the ability for a linear pressure profile to cancel gravity and achieve hydrostatic equilibrium, the goals of obtaining neutral buoyancy, etc., as discussed above. Nonetheless, more sophisticated constructions are possible based on careful cut cell modeling; for example, using an effective solid area based on the average distance between a cell face and the unoccluded portion of a solid-fluid interface in a cell, the authors in [4, 28] are able to define a stable, conservative flux that yields accurate solutions in various compressible solid-fluid problems, including neutrally buoyant tests.

3.1. Fluid Momentum Update

Consider the discretization of an x-direction dual cell D_i of a uniform MAC grid along the lines of Equation 3. The fluid momentum update is given by

$$\left(V_{F,i}\rho_{f,i}u_{f,i}\right)_t=-A_{F,i}\left(p_{R,i}-p_{L,i}\right), \tag{9}$$

which may be discretized in time to obtain

$$u_{f,i}^{n+1}=u_{f,i}^*-\Delta t M_{F,i}^{-1}A_{F,i}\left(p_{R,i}-p_{L,i}\right), \tag{10}$$

[1] In fact, one might also partition the fluid $A_{F,i}$ under various scenarios, such as when a thin shell fully separates different fluid components.

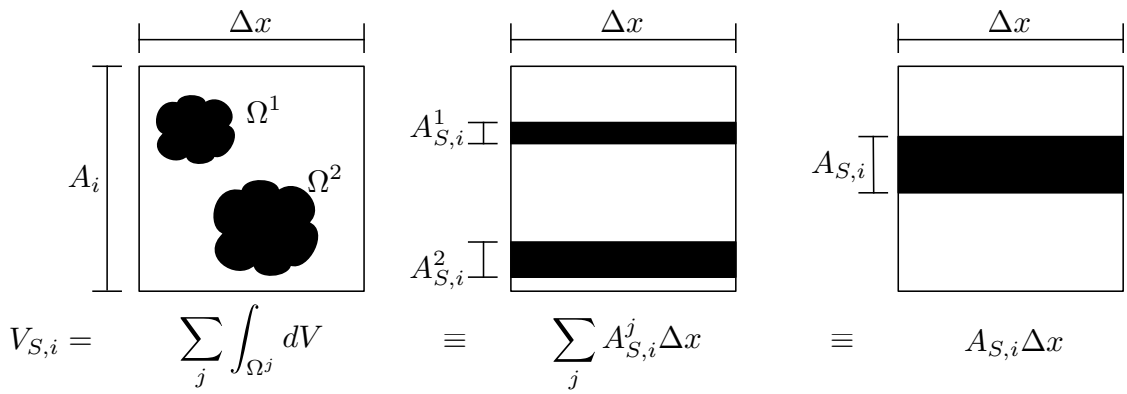


Figure 2: Two sub-grid solids are depicted as occupying regions Ω^1 and Ω^2 of a dual cell. *(Left)* Their total volume can be computed by summing their individually-computed volumes. *(Middle)* Defining $A^j_{S,i} = V^j_{S,i}/\Delta x$ allows one to conceptualize their effect as rectangular prisms, emphasizing the importance of their effective cross-sectional areas in a MAC grid dual cell. *(Right)* Defining $A_{S,i} = A^1_{S,i} + A^2_{S,i}$ emphasizes their net effective cross-sectional area.

where $M_{F,i} = \rho_{f,i} V_{F,i}$. Defining $\tilde{p}_i = \Delta t p$ for each face-averaged pressure p and $G_i = \Delta x^{-1} \begin{bmatrix} -1 & 1 \end{bmatrix}$, the above equation can be written as

$$u^{n+1}_{f,i} = u^*_{f,i} - M^{-1}_{F,i} A_{F,i} \Delta x G_i \begin{bmatrix} \tilde{p}_{L,i} & \tilde{p}_{R,i} \end{bmatrix}^T, \tag{11}$$

and similar equations can be written for the y and z-direction dual cells. Then, stacking all such equations for any dual cell that contains some fluid results in

$$\vec{u}^{n+1}_f = \vec{u}^*_f - M^{-1}_F A_F H_F G_F \tilde{p}, \tag{12}$$

where M_F is a diagonal matrix of the $M_{F,i}$, A_F is a diagonal matrix of the $A_{F,i}$, and H_F is a diagonal matrix with entries of either Δx, Δy, and Δz. $\tilde{p}$ contains one face-averaged pressure for each unique MAC grid dual cell face adjacent to any dual cell that contains fluid; note, the same face-averaged pressure is used for both dual cells adjacent to a face. The gradient matrix G_F has a row for every dual cell containing some fluid and a column for every unique face-averaged pressure. For ease of notation, we define the unitless difference operator $\hat{G}_F = H_F G_F$.

3.2. Solid Momentum Update

We define $\vec{v}$ as a stacked vector of solid generalized velocity degrees of freedom, e.g. translational and rotational velocities for rigid bodies, all the solid particle velocities for deformable bodies, etc. Each rigid body j has a block of entries in $\vec{v}$ of the form $\begin{pmatrix} \vec{v}_j & \vec{\omega}_j \end{pmatrix}^T$ for its associated linear and angular velocity. For each dual cell D_i that body j overlaps, we define a row vector J^j_i that maps from $\vec{v}$ to the scalar velocity due to body j in D_i. In the particular case of rigid bodies, the non-zero elements of J^j_i can be composed from two terms. The first term is a row vector chosen from the standard basis (e.g. $\hat{e}^T_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}$, $\hat{e}^T_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}$ in 2D) which selects the correct component of velocity for D_i. The second term maps $\vec{v}$ to an appropriate vector velocity in D_i. Thus,

$$J^j_i \vec{v} = \hat{e}^T_k \begin{pmatrix} I_{3\times 3} & \vec{r}^{j*T}_i \end{pmatrix} \begin{pmatrix} \vec{v}_j & \vec{\omega}_j \end{pmatrix}^T, \tag{13}$$

where $k = 1, 2, 3$ depends on the direction of the dual cell, $I_{3\times 3}$ is the 3×3 identity matrix, and $\vec{r}^{j*}_i$ is the skew-symmetric matrix such that $\vec{r}^{j*}_i \vec{a} = \vec{r}^j_i \times \vec{a}$ for all $\vec{a}$. A dual cell D_i may contain multiple solid components, and we construct the block matrix J_i by stacking all relevant J^j_i. In addition, we define a matrix J as a stack of the J_i. Finally, we define $\hat{Z}_i$ as a row vector that sums over the rows in J corresponding to D_i (i.e. the rows of J_i) using weights $A^j_{S,i}/A_{S,i}$, in order to define $\hat{J}_i = \hat{Z}_i J$. This allows us to write a scalar average solid velocity for D_i as $\hat{J}_i \vec{v} = \hat{Z}_i J \vec{v}$.

Equation 12 applies pressure impulses $-A_F\hat{G}_F\tilde{p}$ to the fluid, and thus pressure impulses $-A_S\hat{G}_S\tilde{p}$ must be applied to the solid in order to exactly conserve linear momentum with every average pressure at an interior dual cell face applied in an equal and opposite fashion. These impulses are distributed to the rigid bodies using $\hat{J}^T = J^T\hat{Z}^T$ where $\hat{Z}$ and $\hat{J}$ are stacked versions of $\hat{Z}_i$ and $\hat{J}_i$. The action of $\hat{Z}^T$ on $-A_S\hat{G}_S\tilde{p}$ is to divide the total impulse in each dual cell into individual impulses for each solid in that dual cell via area weighting $A^j_{S,i}/A_{S,i}$. Then J^T turns this scalar impulse into a vector impulse before applying it to the rigid body as a force and torque. Thus, the solid momentum update is

$$M_S\vec{v}^{n+1} = M_S\vec{v}^* - \hat{J}^T A_S\hat{G}_S\tilde{p}, \tag{14}$$

where M_S is a block diagonal mass matrix with each block corresponding to a rigid body's mass and inertia tensor.

The moment arm $\vec{r}^j_i$ points from the center of mass of rigid body j to a point in dual cell D_i. Considering a small sub-grid solid that is neutrally buoyant in a quiescent flow, we note that any non-zero $\vec{r}^j_i$ would apply torque to the solid, and that this torque would then create a non-zero velocity contribution to the net dual cell velocity. Thus, we choose the center of mass of body j as the other endpoint of the moment arm $\vec{r}^j_i$, making $\vec{r}^j_i$ identically zero in this situation. This means that a sub-grid solid contained entirely within a single dual cell experiences no pressure-induced torque. When the sub-grid solid happens to span more than one dual cell, we would like to obtain the same solution, especially since a slight coarsening of the grid or small perturbation of the grid would cause the sub-grid solid to once again be contained within a single dual cell. *Our definition of a sub-grid solid is one that is small enough to be wholly contained in a single dual cell either directly or upon a slight coarsening or perturbation of the grid.* In order to achieve the desired solution even when the sub-grid solid overlaps multiple dual cells, we always define $\vec{r}^j_i$ as a vector that points from the center of mass of body j to the center of mass of the subset of body j which is contained in D_i. This definition gives an identically zero $\vec{r}^j_i$ when a sub-grid solid is wholly contained in D_i, but also gives the same desirable solution to the neutrally buoyant quiescent flow case when the sub-grid solid happens to overlap multiple dual cells. To see this, note that the center of mass can be computed using the masses of sub-components, i.e. $m\vec{x}_{\text{com}} = \sum_i m_i\vec{x}_i$, where m is the mass of the rigid body, and the m_i and $\vec{x}_i$ are the masses and center of masses of the subset of the rigid body in each dual cell D_i. For a constant density solid, one can divide through by the constant density to obtain a similar statement in terms of volume, i.e. $V\vec{x}_{\text{com}} = \sum_i V_i\vec{x}_i$. Assuming a constant ∇p that contributes a force of $V_i\nabla p$ in D_i, and applying this force at the center of mass $\vec{x}_i$ of the subset of the body in D_i gives a torque on the body of $(\vec{x}_i - \vec{x}_{\text{com}}) \times V_i\nabla p$. Summing this over all sub-bodies gives $\sum_i (\vec{x}_i - \vec{x}_{\text{com}}) \times V_i\nabla p$, which is identically zero using the previous volume identity. In addition, the net force is $\sum_i V_i\nabla p = V\nabla p$, as desired.

3.3. Composite Velocity

The aforementioned discussion highlights the fact that sub-grid solids as shown in Figure 2 Left have an *effective* cross-sectional area which, as illustrated in Figure 2 Middle, is independent of their actual profile (unlike a more resolved solid) but rather dictated by their volume and the size of the dual cell. The net pressure gradient in the dual cell is partitioned amongst the materials within it using this effective cross-sectional area; thus, the velocity flux (i.e. swept volumes) should be partitioned in a similar manner in order to obtain consistency in the conservation of momentum. Figure 3 illustrates an example where the fluid and solid velocities differ, and that difference is pictorialized in a manner similar to Figure 2 Middle using the rectangular prism construction. As a consequence of these considerations, we define the *composite velocity* $w_{f,i}$ of a dual cell D_i in accordance with Figure 2 Middle and Figure 3:

$$A_i w_{f,i} = A_{F,i}u_{f,i} + \sum_j A^j_{S,i}J^j_i\vec{v}. \tag{15}$$

Dividing through by A_i and multiplying and dividing by $A_{S,i}$ gives

$$w_{f,i} = \frac{A_{F,i}}{A_i}u_{f,i} + \frac{A_{S,i}}{A_i}\left(\sum_j \frac{A^j_{S,i}}{A_{S,i}}J^j_i\vec{v}\right) = \frac{A_{F,i}}{A_i}u_{f,i} + \frac{A_{S,i}}{A_i}\hat{Z}_iJ\vec{v} = \frac{A_{F,i}}{A_i}u_{f,i} + \frac{A_{S,i}}{A_i}\hat{J}_i\vec{v}. \tag{16}$$

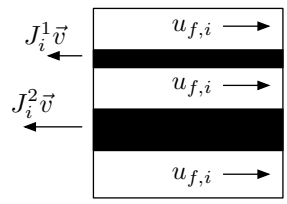


Figure 3: The rectangular prism construction depiction of solid and fluid velocities in a dual cell.

In Equation 12, $\vec{u}_f$ is defined for any dual cell that contains some fluid, and thus the face-averaged pressures $\tilde{p}$ need to be defined for every MAC grid primal cell that overlaps any dual cell that contains some fluid. Since Equation 4 will be discretized for each of these primal cells, $w_{f,i}$ in Equation 16 is defined for every dual cell that overlaps any such primal cell. Then, stacking the far left and far right of Equation 16 for every such dual cell that overlaps a primal cell where $\tilde{p}$ is defined yields

$$\vec{w}_f = A^{-1}\hat{A}_F\vec{u}_f + A^{-1}\hat{A}_S\hat{J}\vec{v}, \tag{17}$$

where A is a diagonal matrix of A_i. $\hat{A}_F$ is a tall matrix consisting of rows that either multiply a single $u_{f,i}$ by $A_{F,i}$ for dual cells containing some fluid or are completely zero otherwise. Similarly, each row of $\hat{A}_S$ either multiplies a single entry of $\hat{J}\vec{v}$ by $A_{S,i}$ for dual cells containing some solid or is a row of zeros otherwise.

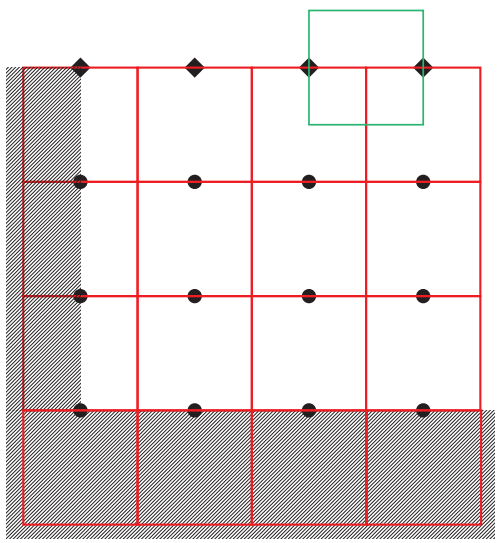


Figure 4: A solid wall (shaded) acts as a boundary. The y-direction dual cell grid (red) has dual cells that are fully occluded by the wall at the bottom of the domain and half occluded on the left-hand side of the domain. When Dirichlet pressure boundary conditions are set along the top of the domain (shown as black diamonds), x-direction dual cells (one of which is shown in green) between two such pressure samples do not appear in Equation 4.

Both $\hat{A}_F$ and $\hat{A}_S$ have a row of zeros where boundary conditions are required, such as the bottom row of dual cells in Figure 4. Thus, for completeness in handling boundary conditions, we augment Equation 17:

$$\vec{w}_f = A^{-1}\hat{A}_F\vec{u}_f + A^{-1}\hat{A}_S\hat{J}\vec{v} + A^{-1}\hat{A}_B\vec{w}_B, \tag{18}$$

where $\vec{w}_B$ is a vector of prescribed boundary $w_{f,i}$ values, with $A_{F,i} + A_{S,i} + A_{B,i} = 1$ for all D_i. The y-direction dual cells in the bottom row of Figure 4 need to be defined as boundary conditions in order to calculate the pressure degrees of freedom on the bottom wall. However, note that Equation 17 already

correctly handles the case where the wall is stationary since $w_{B,i} = 0$ in that case. For dual cells on the left-hand boundary, Equation 17 is also sufficient as long as one correctly sets $A_{F,i} = A_i/2$. This makes $w_{f,i}$ half as large as one might expect and could lead to unwanted drag forces; however, our formulation allows for straightforward slippage of the fluid past the solid, as illustrated in Figure 3. Thus, one can use the correct geometric and physical formulation near the wall boundary without artificially making half ghost cells or devising other ad-hoc procedures in order to compensate for the staggered dual cell arrangements of the MAC grid. We will continue to point out how these dual cells are handled as we further discuss our method. For now, in particular, these half-occluded dual cells have their $A_{F,i}$ and $M_{F,i}$ values correctly halved in Equation 12, although the 2's cancel, leaving Equation 12 unchanged even though $w_{f,i}$ is half its usual value.

Finally, note that one may choose to specify a known average pressure boundary condition such as $p = 0$ on the face of any dual cell containing fluid. This pressure can then be used in Equation 3 without including it as an unknown in Equation 4. In turn, this pressure is not included in $\tilde{p}$, and dual cells overlapping the associated primal cell are not necessarily included in $\vec{w}_f$, etc. For example, specifying $p = 0$ on the top boundary of Figure 4 removes the need to specify $w_{f,i}$ for dual cells above those pressures. In addition, dual cells contained between two such specified pressures, such as the one shown in green in Figure 4, also do not appear in $\vec{w}_f$.

3.4. Coupled System

We enforce a volume-weighted divergence-free condition on the composite velocity via $-G^T V \vec{w}_f$, where V is a diagonal matrix of dual cell volumes V_i and G^T is the full divergence matrix mapping to all primal cells where $\tilde{p}$ is defined. We note that previous works such as [12] and [43] have also enforced the zero divergence of a volume-fraction-weighted sum of solid and fluid velocities, in the context of particle-laden flows. The weights are generally different than in our scheme given that such previous works are based on a material point method and a coupled molecular dynamics-colocated fluid grid scheme, respectively; nonetheless, these works also demonstrate the appropriateness of using a blended solid-fluid velocity for coupling the two phases, under the assumptions of incompressible fluid and impermeable rigid solids.

Substituting $\vec{w}_f$ from Equation 17 into $-G^T V \vec{w}_f$ gives

$$-G^T H \hat{A}_F \vec{u}_f - G^T H \hat{A}_S \hat{J} \vec{v}, \tag{19}$$

where H is a diagonal matrix of dual cell sidelengths. For the dual cells with no fluid, the corresponding row of $\hat{A}_F$ is zero, meaning the corresponding column of G^T does not contribute to $G^T V \vec{w}_f$. Eliminating all such columns from G^T results in G_F^T, allowing us to write $G^T H \hat{A}_F = G_F^T H_F A_F$. Similarly, $G^T H \hat{A}_S = G_S^T H_S A_S$. Thus,

$$-G^T V \vec{w}_f = -G_F^T H_F A_F \vec{u}_f - G_S^T H_S A_S \hat{J} \vec{v}. \tag{20}$$

Note that our G_F and G_S are similar to terms in previous solid-fluid coupling works, such as [18]. However, in prior works, the columns of G_F and G_S were mutually exclusive; if a dual cell D_i had an associated column in G_F^T, it would not have an associated column in G_S^T, and vice versa. In the present work, mixed solid-fluid dual cells are not treated in a binary fashion, and as such, columns corresponding to these dual cells appear in both G_F^T and G_S^T.

Taking the negative of Equation 20 at time t^{n+1}, inserting Equation 12, and setting equal to zero leads to

$$-\left(A_F \hat{G}_F\right)^T M_F^{-1} \left(A_F \hat{G}_F\right) \tilde{p} + \hat{G}_S^T A_S \hat{J} \vec{v}^{n+1} = -\hat{G}_F^T A_F \vec{u}_f^*. \tag{21}$$

Then, combining Equations 21 and 14 yields the following monolithic two-way solid-fluid coupling system:

$$\begin{pmatrix} -\left(A_F \hat{G}_F\right)^T M_F^{-1} \left(A_F \hat{G}_F\right) & \hat{G}_S^T A_S \hat{J} \\ \hat{J}^T A_S \hat{G}_S & M_S \end{pmatrix} \begin{pmatrix} \tilde{p} \\ \vec{v}^{n+1} \end{pmatrix} = \begin{pmatrix} -\hat{G}_F^T A_F \vec{u}_f^* \\ M_S \vec{v}^* \end{pmatrix}. \tag{22}$$

Since Equation 22 is not positive-definite, we use MINRES [32] in the numerical examples that use this formulation. Later in the paper, we modify the approach to be positive-definite and use a different approach for solving the linear systems.

If Equation 18 were used in place of Equation 14, $-\hat{G}_B^T A_B \vec{w}_B$ would be added to the upper term on the right-hand side, although this term vanishes when the wall is stationary. In addition, entries in A_F and M_F corresponding to dual cells containing boundary walls would be affected, as discussed above. However, these modifications do not change the structure of the system. Considering a pressure degree of freedom in the leftmost column of Figure 4, these modifications reduce the treatment of a y-direction dual cell from one that is half-filled with a wall boundary to a dual cell half as wide containing only fluid and sub-grid solids. Since both formulations give the same equations, one obtains the standard linear pressure profile along the boundary in the hydrostatic case, matching that of the rest of the domain. This allows us to easily handle sub-grid solids in dual cells along the boundary (as shown below in Figure 5d) without the need for special treatment including half ghost cells, etc. Finally, note that Dirichlet pressure boundary conditions also do not alter the structure of the system. Terms corresponding to homogeneous Dirichlet pressure boundary conditions vanish, while non-zero pressure values result in an extra term on the right-hand side.

3.5. Hydrostatic and Neutral Buoyancy Tests

First, we verify that we obtain the correct hydrostatic solution from Equation 22 in the absence of any solids. We perform a simulation for $1,000$ time steps with a fixed $\Delta t = 1.0 \times 10^{-3}$s. We use a grid with 17×34 cells and of size 32cm$\times$64cm. Solid wall boundary conditions are placed on the side and bottom walls of the domain, as discussed, and homogeneous Dirichlet boundary conditions are set for the fluid pressure on the top of the domain (i.e., an open tank of fluid). These boundary conditions shrink the physical domain by half a grid cell on all sides. The measured L^∞ norm difference between the computed fluid pressure profile and the analytic linear hydrostatic pressure profile $\rho g h$ across all time steps is 7.65×10^{-9}, indicating our treatment of boundary conditions supports the correct hydrostatic solution for the fluid.

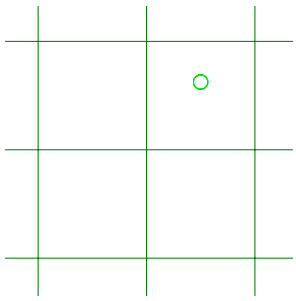
Next, we verify that we obtain the correct static solution with coupled neutrally buoyant sub-grid solids. We add a circle (infinite cylinder) of diameter 0.25cm to the above domain. Both the fluid and solid have unit density. Figure 5 shows that the solid remains stationary over the duration of the simulation, regardless, importantly, of the solid's position with respect to the grid. The figure shows the solid overlapping one, two, or four y-direction dual cells. Also, the results are not affected by the solid being near a solid wall boundary, see Figure 5d. Finally, buoyancy is still maintained for multiple sub-grid solids in the same dual cells, as shown in Figure 5e.

3.6. Under-Resolved Vortices

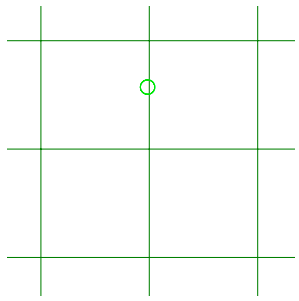
Even if the domain were a single computational cell, our formulation would allow a sub-grid solid denser than the fluid to properly fall under gravity, moving downwards in the cell while creating an upward velocity in the fluid in order to maintain incompressibility. This mix of upward fluid and downward solid velocities in a single cell gives a discrete computational approximation to a small, sub-grid vortex. Note that imposing velocity equilibration on this single cell would instead incorrectly freeze the solid and fluid in place.

For the sake of exposition, we craft a simple illustrative test problem with results simple enough that our code can be verified via pencil and paper. Starting from Equation 22, we remove the horizontal velocity degrees of freedom from both the fluid and solid as well as the angular velocity degrees of freedom from the solid. This yields a pseudo-one-dimensional system that only performs coupling along the y-direction, with no solid rotational motion. We choose a domain composed of a single column of ten y-direction dual cells, with $\Delta x = \Delta y = 1$. Additionally, we consider only one time step with $\Delta t = 1$ and set Dirichlet boundary conditions of 0 at the top of the column and $\rho g h$ at the bottom of the column, where $\rho = 1$, $g = 10$, and $h = 10$. A single sub-grid solid with density 5 is placed in the topmost dual cell and is allowed to fall from rest. In the case of velocity equilibration, the downward buoyant force of the solid would drag all the fluid in the topmost dual cell downwards with it, and via incompressibility, the entire rest of the fluid column would also be dragged downwards. Thus, the solid buoyancy force ends up being averaged over the entire domain. Alternatively, with our formulation, the solid may slip downwards through the fluid, driving the fluid upwards.

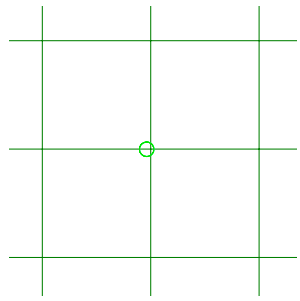
For simplicity, we construct a background pressure p_0 that ranges linearly from 0 to $\rho g h$, matching the domain boundary conditions. Then, letting $\tilde{p}_0 = \Delta t p_0$ allows us to write $\tilde{p} = \tilde{p}_0 + \tilde{p}'$, where $\tilde{p}'$ is the unknown perturbation of $\tilde{p}_0$ that gives $\tilde{p}$. Substituting this expression for $\tilde{p}$ into Equation 12 still leads to Equation 22, except with $\tilde{p}'$ as the unknown (instead of $\tilde{p}$) and $\vec{u}_f^*$ augmented by the effects of $\tilde{p}_0$, i.e. $M_F^{-1} A_F H_F G_F \tilde{p}_0$. This reformulation allows us to consider the effects of the coupling in Equation 22 on a system that already



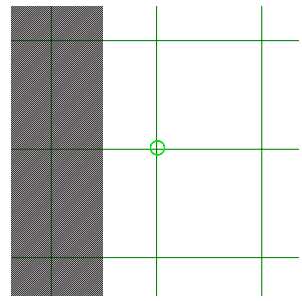
(a) $n = 1$ configuration



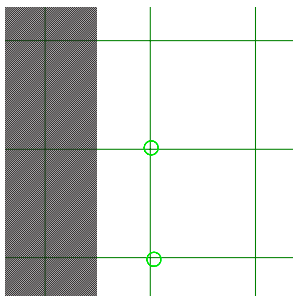
(b) $n = 2$ configuration



(c) $n = 4$ configuration



(d) $n = 4$ near boundary



(e) Two $n = 4$ solids

Ex.	n	$\vec{x}_{\text{com}}$	x-velocity	y-velocity	Angular velocity
(a)	1	$(16.0, 36.0)$	7.28×10^{-13}	2.78×10^{-12}	1.84×10^{-6}
(b)	2	$(15.03, 35.9)$	2.62×10^{-11}	1.34×10^{-11}	2.04×10^{-7}
(c)	4	$(14.99, 34.82)$	4.54×10^{-12}	2.60×10^{-11}	6.46×10^{-7}
(d)	4	$(1.9, 36.75)$	1.26×10^{-12}	3.24×10^{-11}	4.74×10^{-9}
(e1)	4	$(1.9, 36.75)$	2.34×10^{-13}	2.00×10^{-12}	8.78×10^{-10}
(e2)	4	$(1.95, 34.8)$	2.83×10^{-13}	2.36×10^{-12}	3.53×10^{-10}

(f) L^∞ norms of velocity components for different grid placements of a sub-grid solid (centered at $\vec{x}_{\text{com}}$), simulated over $1,000$ time steps. Linear velocity is measured in cm/s and angular velocity in rad/s. (e1) and (e2) are for the upper and lower solids.

Figure 5: One or more neutrally buoyant sub-grid solids are placed in all possible overlapping configurations in two spatial dimensions, overlapping one, two, or four y-direction dual cells. In all cases, the object is observed to remain stationary.

includes the effects of gravity *and* the background pressure $\tilde{p}_0$. Thus, the initial solid and fluid velocities of zero receive a downward perturbation $-g\Delta t$ as well as an upward pressure force which exactly cancels $-g\Delta t$ in the fluid-only cells, so that the time t^* velocities in the fluid-only cells are identically zero. In the topmost cell that contains both solid and fluid, the relative face areas of solid and fluid allow the upward pressure force to also exactly cancel the downward velocity in the fluid portion of the cell. However, the upward pressure force on the solid's area fraction of the cell is too small to entirely cancel the solid's downward velocity. We utilize a solid in the shape of a square with total area 0.5, and thus mass equal to 2.5, resulting in the upward pressure force only cancelling one-fifth of the downward velocity of 10, yielding a downward velocity of 8.

In a velocity equilibration formulation, the downward solid momentum of 20 would be averaged with the fluid in the topmost cell to obtain a downward velocity of 20/3. However, incompressibility further dictates that every cell in the column should have the same velocity, and so the downward momentum of 20 is averaged over the other nine cells in the column as well, resulting in a downward velocity of 20/12 for every cell in the column. It is straightforward to see that adding more cells to the column asymptotes to a net downward velocity of zero since the solid momentum gets averaged over all the fluid in the column. This example illustrates how sub-grid solids are forced not only to drag the fluid in their associated mixed dual cells, but also to perturb a large volume of surrounding fluid in order to move.

Let p_1 be the pressure below the topmost cell such that $p_1 - 0$ gives the net upward force on the topmost cell. The bottom nine cells will all move downward uniformly due to incompressibility, and the net downward force on them is also $p_1 - 0$. Note that both the upper and lower Dirichlet boundary conditions are zero on $\tilde{p}'$ in Equation 22 since we have already added the contributions of $\tilde{p}_0$ into the time t^* velocities. p_1 reduces the downward momentum of 20 in the topmost cell to $20 - p_1$ to obtain a net velocity of $(20 - p_1)/3$, and the velocity in each of the other nine cells is $p_1/9$. Incompressibility dictates that these two velocities are equal, and thus $p_1 = 15$, which asymptotes to $p_1 = 20$ if the column is allowed to extend infinitely downward. Depending on how one partitions $p_1 = 20$ amongst the components of the topmost dual cell, one could stop both the solid and fluid motion (velocity equilibration), or one might try to let the solid fall by driving the fluid upwards. If one used the typical volume-weighted pressure gradient, half of this pressure force would go the fluid and half would go the solid resulting in the solid falling at speed 4 and the fluid moving upward

at speed 10, which seems unhelpful. Therefore, we find the use of the total momentum over the total mass as a dual cell velocity value problematic. This is one of the key observations that led to our approach.

In our approach, incompressibility calculations see the mixed solid-fluid velocity $w_{f,i}$ in the topmost dual cell, resulting in this example in

$$p_1/9 = w_{f,i} = v_s^{n+1}/2 + u_{f,i}^{n+1}/2. \tag{23}$$

Then, the volume-weighted pressure gradient force distribution causes the new solid velocity to be $(20 - p_1/2)$ $/2.5$ and the new fluid velocity to be $(-p_1/2)/0.5$. Plugging into Equation 23 yields

$$p_1/9 = (8 - p_1/5)/2 - p_1/2, \tag{24}$$

which yields $p_1 = 45/8$, an upward velocity of $45/8$ for the fluid, and a downward velocity of $55/8$ for the solid. In the limit of an infinitely tall column, the left-hand side of Equation 24 again approaches zero. In this case, we obtain $p = 20/3$, an upward fluid velocity of $20/3$, and a downward solid velocity of $20/3$. Thus, our model allows the solid to fall, avoiding spurious velocity equilibration.

Although our formulation allows for highly improved behavior insofar as the solid and fluid slipping past each other in a single cell as opposed to momentum and mass lumped formulations, it unfortunately does not fully alleviate these issues in a manner that is fully independent of the computational grid. In order to see this, we consider the same problem except with the solid located half way in between the two topmost dual cells, with half its area in each of them. Let p_2 be the pressure just below p_1, dividing the bottom eight cells from the top two. Then $p_2/8$ will be the downward velocity of the bottom eight cells, and since the solid only occupies one-quarter of the top two dual cells, its velocity will be $(20 - p_2/4)/2.5$, whereas the fluid velocity will be $(-3p_2/4)/(3/2)$. Plugging into Equation 23 (with $p_1/8$ on the left) and solving yields $p_2 = 160/17$, which in turn results in different solid and fluid velocities than when the solid overlaps only one dual cell. In the limit where the fluid column extends infinitely downwards, the solution is $p_2 = 40/3$. Note that we have ignored p_1 here since its effect on the solid cancels when the solid is midway between the two top dual cells; however, $p_1 = p_2/2$, which makes the fluid velocities in the top two cells match. For completeness, one can consider any position of the solid overlapping the top two cells via the following equations:

$$\begin{aligned} A_{S,1} v_s^{n+1} + A_{F,1} u_{f,1}^{n+1} &= 0 \\ A_{S,2} v_s^{n+1} + A_{F,2} u_{f,2}^{n+1} &= 0 \\ M_{F,1} u_{f,1}^{n+1} &= -p_1 A_{F,1} \\ M_{F,2} u_{f,2}^{n+1} &= (p_1 - p_2) A_{F,2} \\ M_S v_s^{n+1} &= M_S v_s^* - p_1 (A_{S,1} - A_{S,2}) - p_2 A_{S,2}. \end{aligned} \tag{25}$$

The first two equations follow from our area-weighted incompressibility formulation, and the next three equations are the momentum updates for the solid and fluid components in the two mixed dual cells. Substituting the last three equations into the first two equations to eliminate the velocities results in a 2×2 system for p_1 and p_2. Figure 6 shows the variance in p_1 and p_2 as the solid transitions from the top cell to the next lower cell. Since p_2 always sees the same material above it, one would hope that it would be flat, but it bows downward in the figure. Figure 7 shows the velocities; again, one would hope that v_s (depicted at the bottom) would be flat. Note how the two fluid velocities contribute equally when the solid is equally in both cells, but only contribute fractions of upward velocities otherwise. Figure 8 shows that the neutrally buoyant case correctly gives a flat line (top curve), and as the density is increased, one achieves a flatter line and the correct analytical solution of -10, although there is inaccurate bowing of curves in between.

3.7. Additional examples

It is interesting to consider the limiting behavior of the solid velocity in the column test of the previous section. In particular, the solid momentum update in Equation 25 suggests that as the time t^* solid velocity approaches infinity, so too will the solid velocity after the solve; however, the fluid's response increases for faster solid velocities. Consider the case when the solid lies entirely in the lower of the two top y-direction dual cells. Then, solving Equation 25 yields

$$p_1 = 0, \qquad p_2 = \frac{2M_s v_s^*}{2M_s + 1}, \qquad v_s^{n+1} = \frac{2M_s v_s^*}{2M_s + 1}. \tag{26}$$

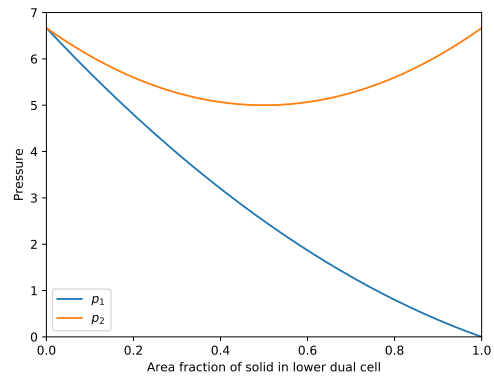


Figure 6: Pressures p_1 and p_2 for the column example at the end of Section 3.6.

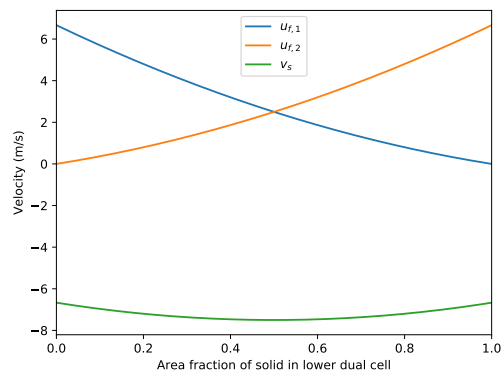


Figure 7: Velocities for the column example at the end of Section 3.6.

For instance, when the density of the solid is 5 (and thus, the solid is 5/6 of the total mass of the dual cell), the solid's velocity after one time step is 5/6 of the solid's post-advection velocity. Additionally, as v_s^n is increased, gravity and the ∇p_0 forces eventually have insignificant effect on v_s^{n+1} as compared to the projective $\nabla p'$ force, as shown in Figure 9.

Next, we study the narrow column example from the previous section as the solid falls from rest over 60s. The density of the solid is modified to be 1.05kg/m^3 unless otherwise stated, and the time step is 1.0s unless otherwise stated. The total height of the column is $1024 + \Delta y$ m, and the solid's initial y-position is fixed at 1024m. We numerically explore the effects of increasing the solid's density, refining the time step, refining the y-resolution of the computational grid, and refining simultaneously in space and time. We ignore fluid advection, only applying ∇p_0, gravity, and the forces from the coupled solve.

Figure 10 shows the velocities and vertical positions of the solid at various densities, using a time step of 0.1s. The dotted lines indicate the trajectory of an infinitely dense solid under only the influence of gravity and ∇p_0, neglecting the upward fluid pressure force from the coupled solve which should become negligible at high density. At lighter densities, the solid's velocity more quickly reaches a terminal velocity (up to periodic oscillations due to under-resolved vortices), while at heaver densities, the solid's velocity more quickly approaches the analytic linear solution.

Next, we refine the time step by powers of two from 2^0s to 2^{-10}s for a fixed spatial resolution of 1024 primal y-direction grid cells. The results are shown in Figure 11. Again, the analytic trajectory of the solid is plotted as a dotted line based on only the gravitational and ∇p_0 forces. Refining the time step results in the solid's velocity following the linear analytic trajectory for a longer amount of time, before leveling off to a terminal velocity.

In Figure 12, we coarsen the grid from 1024 primal y-direction grid cells up to just 1, noting that our

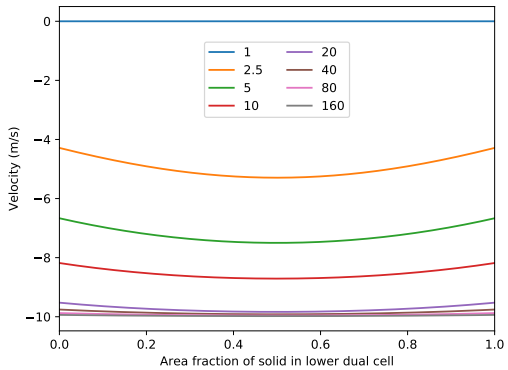


Figure 8: Solid velocities for densities varying from 1 to 160, for the column example at the end of Section 3.6.

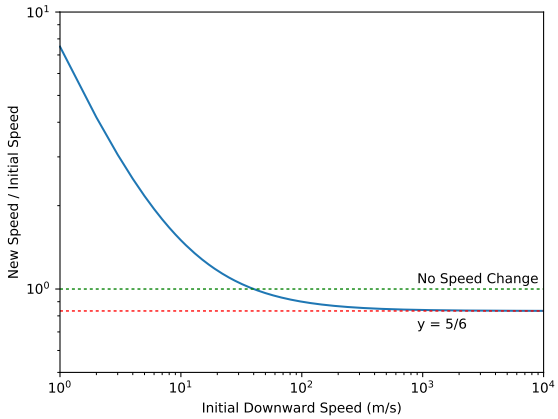


Figure 9: The y-velocity of the solid particle, of density 5, in the column example after one time step with varying initial y-velocities. The independent axis is the initial y-velocity, and the dependent axis is the ratio of the y-velocity after one time step to the initial y-velocity. Both axes are log-scaled.

formulation is equally appropriate for rectangular grid cells. At very coarse resolutions, the rectangular prism construction vertically smears the solid across almost the entire domain, which results in the solid only feeling the effects of gravity and the ∇p_0 force from boundary conditions; thus, the results are in close agreement with the corresponding analytic curves. At finer resolutions, there are more fluid pressure degrees of freedom between the solid and the bottom of the domain, which results in increased influence of the fluid pressure force from the coupled solve and hence more retarded solid motion.

Finally, Figure 13 demonstrates refining the example simultaneously in both space and time. We note that while terminal velocities are noticeably grid-dependent for sub-grid solids, they can be carefully tuned by introducing velocity drag, which is the subject of Section 4. The intended purpose of the examples in this section is not to demonstrate grid convergence of decidedly unresolved phenomena, i.e. sub-grid solids, but rather to provide a thorough analytical and numerical analysis of the behavior of such a formulation.

We repeated these experiments for the falling solid in a wider column, solving the full 2D system of Equation 22. We place the solid at $(255.5, 1024)$ (horizontally centered) in a column of width 511m and height 2044m. The results, shown in Figures 14–17, are largely similar to the pseudo-one-dimensional examples just discussed, though we observe that there is relatively less variation between different temporal resolutions.

As an additional experiment, we simulated the pseudo-one-dimensional column example and repeatedly made the column narrower, while also shrinking the sidelengths of the square solid. Solid density was kept

constant. The results, in Figure 18, indicate that for a narrower column (higher number of refinements), the terminal velocity of the solid can be considerably higher than for a wider column. The narrower column contains less fluid and hence cannot provide as strong of an upward pressure force on the solid.

4. Velocity Drag

It is well known that viscous drag forces participate in momentum exchanges between fluid and solid components, in addition to pressure drag. This is especially true for small, sub-grid solids, where under-resolved viscous forces may dominate. In order to model this behavior for sub-grid solids, we augment Equation 9 with a drag term:

$$\left(V_{F,i}\rho_{f,i}u_{f,i}\right)_t = -A_{F,i}\left(p_{R,i} - p_{L,i}\right) - \sum_j k_i^j \left(u_{f,i} - J_i^j \vec{v}\right), \tag{27}$$

where k_i^j (assumed to be > 0) is a drag coefficient for body j in dual cell D_i. We may express the drag term in the above as

$$\bar{k}_i \left(u_{f,i} - \sum_j \frac{k_i^j}{\bar{k}_i} J_i^j \vec{v}\right) = \bar{k}_i \left(u_{f,i} - \tilde{Z}_i J \vec{v}\right) = \bar{k}_i \left(u_{f,i} - \tilde{J}_i \vec{v}\right), \tag{28}$$

where $\bar{k}_i = \sum_j k_i^j$. $\tilde{Z}_i$ and $\tilde{J}_i$ are defined similarly to $\hat{Z}_i$ and $\hat{J}_i$ from Section 3.2, except they use partial drag coefficients as weights rather than partial solid face areas.

Similar to Equation 11, we obtain

$$u_{f,i}^{n+1} = \left[1 + \Delta t M_{F,i}^{-1}\bar{k}_i\right]^{-1} \left(u_{f,i}^* - M_{F,i}^{-1} A_{F,i} \Delta x G_i \begin{bmatrix}\tilde{p}_{L,i} & \tilde{p}_{R,i}\end{bmatrix}^T + \Delta t M_{F,i}^{-1} \bar{k}_i \tilde{J}_i \vec{v}^{n+1}\right). \tag{29}$$

We substitute the definition $K_i = 1 + \Delta t M_{F,i}^{-1}\bar{k}_i$ to obtain

$$u_{f,i}^{n+1} = K_i^{-1} \left(u_{f,i}^* - M_{F,i}^{-1} A_{F,i} \Delta x G_i \begin{bmatrix}\tilde{p}_{L,i} & \tilde{p}_{R,i}\end{bmatrix}^T + \Delta t M_{F,i}^{-1} \bar{k}_i \tilde{J}_i \vec{v}^{n+1}\right), \tag{30}$$

which may be stacked along the lines of Equation 12 to obtain

$$\vec{u}_f^{n+1} = K^{-1} \left(\vec{u}_f^* - M_F^{-1} A_F H_F G_F \tilde{p} + \Delta t M_F^{-1} \bar{k} \tilde{J} \vec{v}^{n+1}\right). \tag{31}$$

Here, $\tilde{J}$ is a matrix of non-zero rows $\tilde{J}_i$ that correspond to dual cells that contain one or more solid components and overlap a primal grid cell where $\tilde{p}$ is defined. $\bar{k}$ has a row for each dual cell that contains some fluid and a column for each dual cell that contains some solid and overlaps a primal cell where $\tilde{p}$ is defined. $\bar{k}$ contains the entry $\bar{k}_i$ at the intersection of the row and column corresponding to dual cell D_i when such a row and column are both present in $\bar{k}$. K is a diagonal matrix of the K_i.

The drag forces on the solids should balance the drag forces applied to the fluid. For the portion of solid body j in dual cell D_i, the drag force acting on it is taken as

$$k_i^j \left(u_{f,i} - J_i^j \vec{v}\right) \tag{32}$$

in order to conserve momentum. Alternatively, this can be written as

$$\tilde{Z}_i^j \bar{k}_i u_{f,i} - k_i^j J_i^j \vec{v}, \tag{33}$$

where $\tilde{Z}_i^j$ is the entry of $\tilde{Z}_i$ corresponding to solid j. (Recall from Equation 28 that $\tilde{Z}_i$ is a row vector with entries $k_i^j/\bar{k}_i$ that multiplies $J\vec{v}$ in the middle of that equation to obtain the sum on the left of that equation via dot product.)

Stacking Equation 33 for all dual cells where $\vec{u}_f$ is defined yields

$$\tilde{Z}^T \bar{k}^T \vec{u}_f - \bar{k}_S J \vec{v}, \tag{34}$$

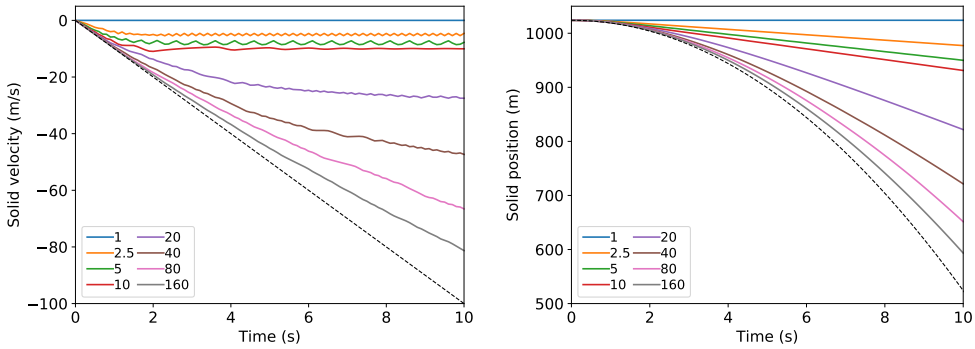


Figure 10: Increasing the density of the falling sub-grid solid for the narrow column example of Section 3.6.

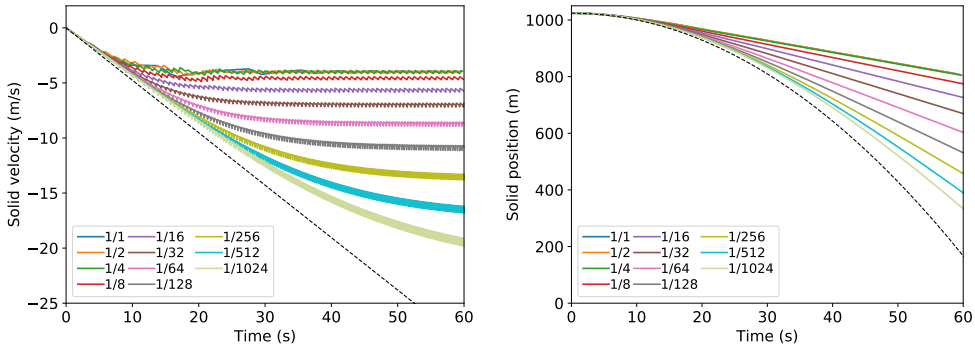


Figure 11: Refining the time step for the narrow column example of Section 3.6.

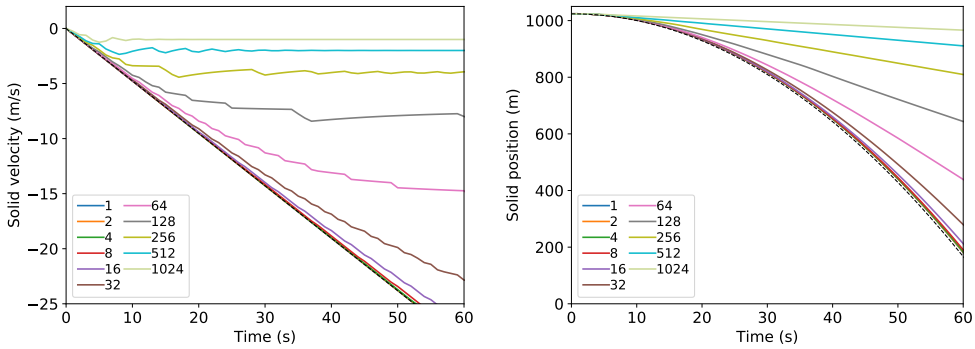


Figure 12: Refining the y-axis of the computational grid for the column example of Section 3.6.

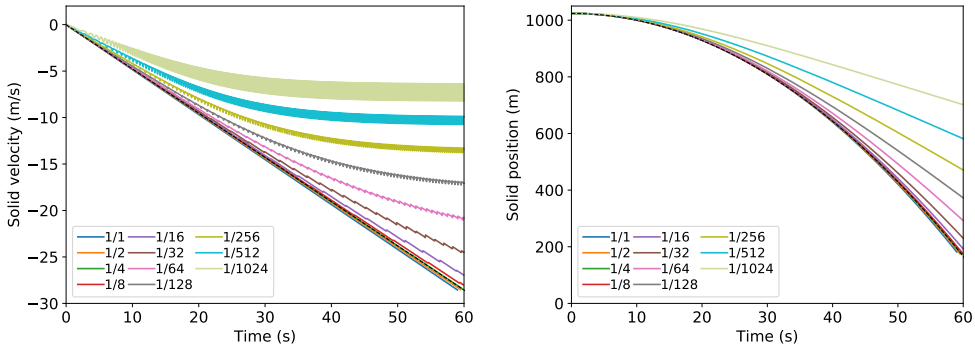


Figure 13: Refining the narrow column example in space and time. Labels indicate the time step, the inverse of y-resolution.

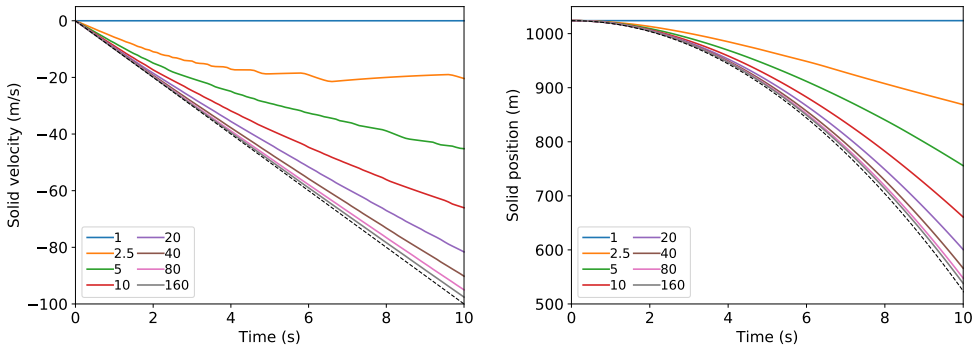


Figure 14: Solving the full 2D coupling system for a sub-grid particle in a wide channel at varying solid densities.

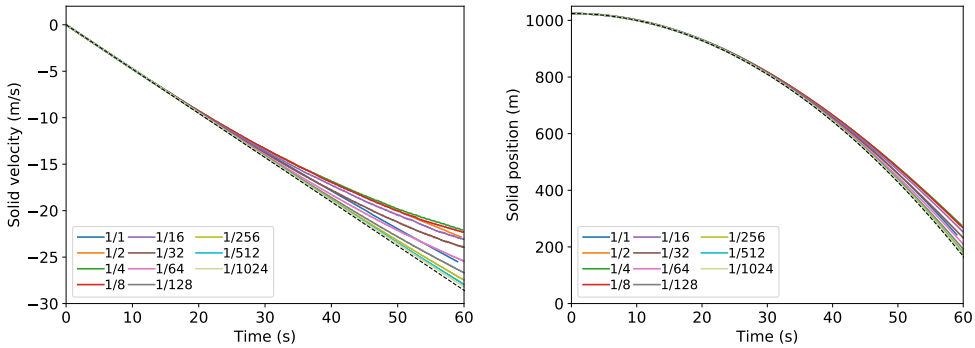


Figure 15: Refining the time step for the wide channel example.

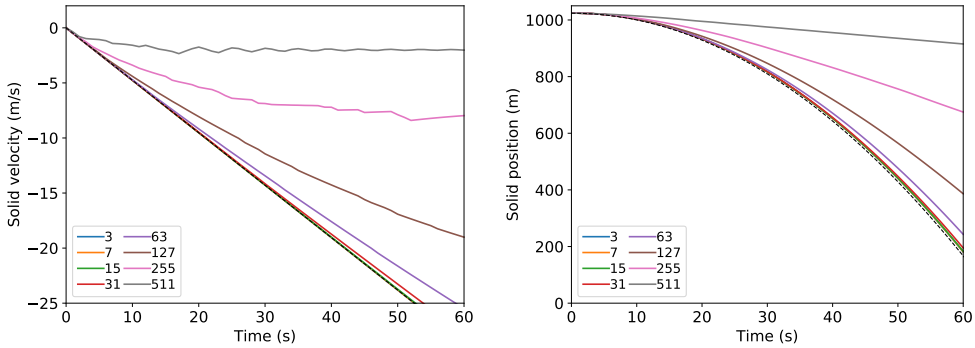


Figure 16: Varying the spatial resolution of the 2D wide channel example. The labels indicate the number of grid cells in the x-direction; the y-resolution is chosen to maintain square grid cells.

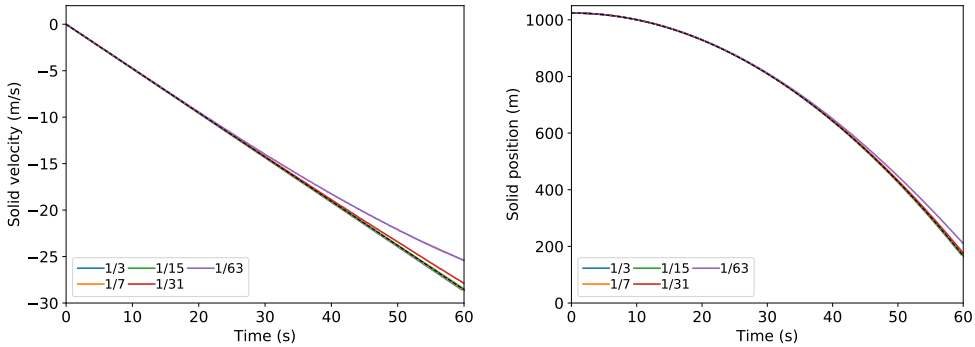


Figure 17: Refining space and time simultaneously for the 2D wide channel example.

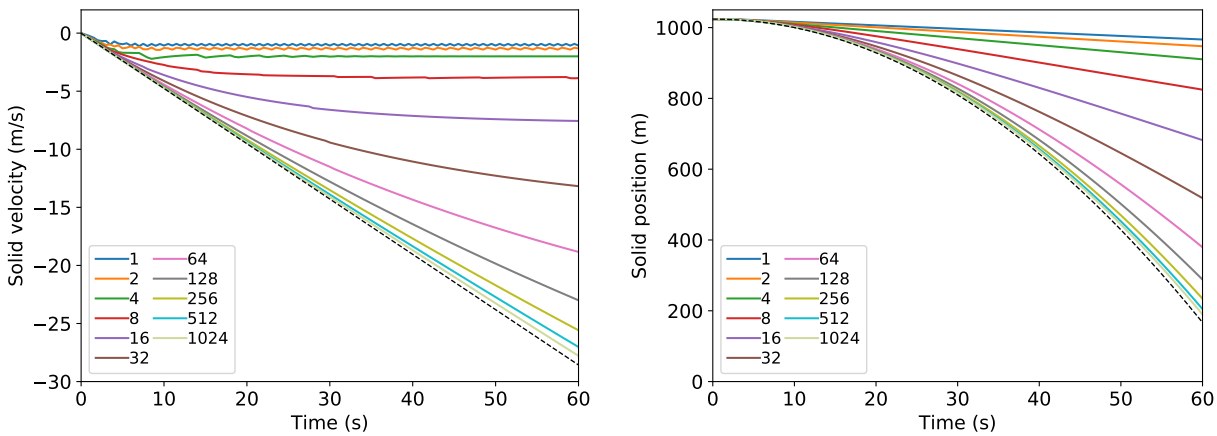


Figure 18: Shrinking the width of the pseudo-one-dimensional column and the sidelengths of the solid, keeping solid density constant. The dashed line indicates the analytic solution for an infinitely narrow column. Legend labels are the number of refinements (the inverse of the column width).

where $\bar{k}_S$ is a diagonal matrix of all defined k_i^j. We note that in the above expression, $\bar{k}^T$ scales the fluid velocities $\vec{u}_f$ by $\bar{k}_i$, as in Equation 33. Then, $\tilde{Z}^T$ distributes those values to each solid component within each dual cell, such that the $u_{f,i}$ are weighted by the k_i^j, as in Equation 32. Finally, as with the coupling term due to fluid pressure, applying J^T to the expression in Equation 34 applies those drag forces to the solids. Incorporating both coupling terms, the solid momentum update is

$$M_S\vec{v}^{n+1} = M_S\vec{v}^* - \hat{J}^T A_S \hat{G}_S \tilde{p} + \left(\Delta t \bar{k} \tilde{J}\right)^T \vec{u}_f^{n+1} - \Delta t J^T \bar{k}_S J \vec{v}^{n+1}, \tag{35}$$

where we have used the fact that $\tilde{J} = \tilde{Z}J$. Substituting Equation 31 into the above, simplifying, and then rearranging terms yields

$$\begin{aligned} &\left[M_S + \Delta t J^T \bar{k}_S J - \left(\Delta t \bar{k} \tilde{J}\right)^T (M_F K)^{-1} \left(\Delta t \bar{k} \tilde{J}\right)\right] \vec{v}^{n+1} \\ &+ \left[\hat{J}^T A_S \hat{G}_S + \left(\Delta t \bar{k} \tilde{J}\right)^T (M_F K)^{-1} A_F \hat{G}_F\right] \tilde{p} = M_S \vec{v}^* + \left(\Delta t \bar{k} \tilde{J}\right)^T K^{-1} \vec{u}_f^*. \end{aligned} \tag{36}$$

Similar to the derivation in Section 3, one may substitute $\vec{u}_f^{n+1}$ into the incompressibility condition $\nabla_V \cdot \vec{w}_f^{n+1} = 0$ to obtain

$$-\left(A_F \hat{G}_F\right)^T (M_F K)^{-1} \left(A_F \hat{G}_F\right) \tilde{p} + \left(\hat{G}_F^T A_F (M_F K)^{-1} \Delta t \bar{k} \tilde{J} + \hat{G}_S^T A_S \hat{J}\right) \vec{v}^{n+1} = -\hat{G}_F^T A_F K^{-1} \vec{u}_f^*. \tag{37}$$

Then, combining Equations 37 and 36 yields the following monolithic system that incorporates velocity drag:

$$\begin{aligned} &\begin{pmatrix} -\left(A_F \hat{G}_F\right)^T (M_F K)^{-1} \left(A_F \hat{G}_F\right) & \left(A_F \hat{G}_F\right)^T (M_F K)^{-1} \Delta t \bar{k} \tilde{J} + \hat{G}_S^T A_S \hat{J} \\ \hat{J}^T A_S \hat{G}_S + \left(\Delta t \bar{k} \tilde{J}\right)^T (M_F K)^{-1} A_F \hat{G}_F & M_S + \Delta t J^T \bar{k}_S J - \left(\Delta t \bar{k} \tilde{J}\right)^T (M_F K)^{-1} \left(\Delta t \bar{k} \tilde{J}\right) \end{pmatrix} \begin{pmatrix} \tilde{p} \\ \vec{v}^{n+1} \end{pmatrix} \\ &\qquad = \begin{pmatrix} -\hat{G}_F^T A_F K^{-1} \vec{u}_f^* \\ M_S \vec{v}^* + \left(\Delta t \bar{k} \tilde{J}\right)^T K^{-1} \vec{u}_f^* \end{pmatrix}. \end{aligned} \tag{38}$$

Following the discussion in Section 3, note that setting $V_F = A_F H_F$ and $V_S = A_S H_S$ (the rectangular prism

construction) simplifies Equation 38 to

$$\begin{pmatrix} -\left(V_F G_F\right)^T \left(M_F K\right)^{-1} \left(V_F G_F\right) & \left(V_F G_F\right)^T \left(M_F K\right)^{-1} \Delta t \bar{k} \tilde{J} + G_S^T V_S \hat{J} \\ \hat{J}^T V_S G_S + \left(\Delta t \bar{k} \tilde{J}\right)^T \left(M_F K\right)^{-1} V_F G_F & M_S + \Delta t J^T \bar{k}_S J - \left(\Delta t \bar{k} \tilde{J}\right)^T \left(M_F K\right)^{-1} \left(\Delta t \bar{k} \tilde{J}\right) \end{pmatrix} \begin{pmatrix} \tilde{p} \\ \vec{v}^{n+1} \end{pmatrix} = \begin{pmatrix} -G_F^T V_F K^{-1} \vec{u}_f^* \\ M_S \vec{v}^* + \left(\Delta t \bar{k} \tilde{J}\right)^T K^{-1} \vec{u}_f^* \end{pmatrix}, \tag{39}$$

and, as expected, we recover Equation 22 if we let $K \to I$ and $\bar{k} \to 0$.

5. Symmetric Positive Definite Formulation

It is interesting to consider how the drag coupling system of Section 4 behaves as the k_i^j tend towards infinity. The form chosen for the drag in Equation 32 implies that the fluid and solid velocities equilibrate at time t^{n+1} in the limit of high drag. However, in the full coupling system of Equation 38, it is not immediately clear how certain terms such as the off-diagonal blocks behave as drag goes to infinity, and thus it is difficult to reason about the overall behavior of the system in this limit. At the very least, there are clear numerical issues with at least some terms in the system tending towards infinity as drag increases.

To better understand this limiting behavior, it is natural to consider the quantity

$$\hat{\lambda} = Z^T \vec{u}_f - J\vec{v}, \tag{40}$$

which matches, for each coupled dual cell, a copy of the fluid velocity in the dual cell separately with each of the interpolated velocities of solids in that dual cell. Here, $Z = \bar{k}\tilde{Z}\bar{k}_S^{-1}$, making Equation 40 the non-drag-weighted velocity differences from Equation 34. Z sums the appropriate entries of $\hat{\lambda}$ for each dual cell where $\vec{u}_f$ is defined, and Z^T makes copies of the appropriate entries of $\vec{u}_f$ according to the number of solids in each solid-fluid dual cell. $\hat{\lambda}$ is precisely the quantity that should be zero at t^{n+1} in the case of infinite drag. Thus, we aim to improve the asymptotic numerical behavior of the system by reformulating it in terms of $\hat{\lambda}$. Then, Equations 31 and 35 become

$$\vec{u}_f^{n+1} = \vec{u}_f^* - M_F^{-1} A_F \hat{G}_F \tilde{p} - \Delta t M_F^{-1} Z \bar{k}_S \hat{\lambda} \tag{41}$$

and

$$M_S \vec{v}^{n+1} = M_S \vec{v}^* - \hat{J}^T A_S \hat{G}_S \tilde{p} + \Delta t J^T \bar{k}_S \hat{\lambda}. \tag{42}$$

Using the incompressibility of $\vec{w}_f$ and Equations 40–42, one may derive (similarly to the previous section) the coupling system

$$\begin{pmatrix} \left(A_F \hat{G}_F\right)^T M_F^{-1} \left(A_F \hat{G}_F\right) & -\hat{G}_S^T A_S \hat{J} & \Delta t \hat{G}_F^T A_F M_F^{-1} Z \bar{k}_S \\ -\hat{J}^T A_S \hat{G}_S & -M_S & \Delta t J^T \bar{k}_S \\ \Delta t \bar{k}_S Z^T M_F^{-1} A_F \hat{G}_F & \Delta t \bar{k}_S J & \Delta t \bar{k}_S + \Delta t^2 \bar{k}_S Z^T M_F^{-1} Z \bar{k}_S \end{pmatrix} \begin{pmatrix} \tilde{p} \\ \vec{v}^{n+1} \\ \hat{\lambda}^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{G}_F^T A_F \vec{u}_f^* \\ -M_S \vec{v}^* \\ \Delta t \bar{k}_S Z^T \vec{u}_f^* \end{pmatrix}, \tag{43}$$

which again has several problematic terms in the limit of large drag. As $\bar{k}_S$ approaches infinity, $\hat{\lambda}$ tends towards zero, making their product potentially finite. Thus we define their product as

$$\lambda = \bar{k}_S \left(Z^T \vec{u}_f - J\vec{v}\right). \tag{44}$$

Using this drag force λ, we may rewrite Equations 41 and 42 as

$$\vec{u}_f^{n+1} = \vec{u}_f^* - M_F^{-1} A_F \hat{G}_F \tilde{p} - \Delta t M_F^{-1} Z \lambda \tag{45}$$

and

$$M_S \vec{v}^{n+1} = M_S \vec{v}^* - \hat{J}^T A_S \hat{G}_S \tilde{p} + \Delta t J^T \lambda, \tag{46}$$

and the coupling system becomes

$$\begin{pmatrix} \left(A_F\hat{G}_F\right)^T M_F^{-1}\left(A_F\hat{G}_F\right) & -\hat{G}_S^T A_S \hat{J} & \Delta t \hat{G}_F^T A_F M_F^{-1} Z \\ -\hat{J}^T A_S \hat{G}_S & -M_S & \Delta t J^T \\ \Delta t Z^T M_F^{-1} A_F \hat{G}_F & \Delta t J & \Delta t \bar{k}_S^{-1} + \Delta t^2 Z^T M_F^{-1} Z \end{pmatrix} \begin{pmatrix} \tilde{p} \\ \vec{v}^{n+1} \\ \lambda^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{G}_F^T A_F \vec{u}_f^* \\ -M_S \vec{v}^* \\ \Delta t Z^T \vec{u}_f^* \end{pmatrix}. \tag{47}$$

In this formulation, there is only one term in the system involving the drag coefficients, and this term clearly tends toward zero as drag approaches infinity. We stress that λ represents a physical drag force between the fluid and the solid and is not simply some Lagrange multiplier.

While this system is symmetric and has certain attractive numerical attributes, it is indefinite. Thus, we eliminate the second row of the system by substituting M_S^{-1} times the second row into both the first and third rows in order to eliminate $\vec{v}^{n+1}$, obtaining

$$\begin{pmatrix} \left(A_F\hat{G}_F\right)^T M_F^{-1}\left(A_F\hat{G}_F\right) + \left(\hat{J}^T A_S\hat{G}_S\right)^T M_S^{-1}\left(\hat{J}^T A_S\hat{G}_S\right) & \Delta t\left(A_F\hat{G}_F\right)^T M_F^{-1} Z - \Delta t\left(\hat{J}^T A_S\hat{G}_S\right)^T M_S^{-1} J^T \\ \Delta t Z^T M_F^{-1}\left(A_F\hat{G}_F\right) - \Delta t J M_S^{-1}\left(\hat{J}^T A_S\hat{G}_S\right) & \Delta t \bar{k}_S^{-1} + \Delta t^2 Z^T M_F^{-1} Z + \Delta t^2 J M_S^{-1} J^T \end{pmatrix} \begin{pmatrix} \tilde{p} \\ \lambda^{n+1} \end{pmatrix} = \begin{pmatrix} \left(A_F\hat{G}_F\right)^T \vec{u}_f^* + \left(\hat{J}^T A_S\hat{G}_S\right)^T \vec{v}^* \\ \Delta t\left(Z^T\vec{u}_f^* - J\vec{v}^*\right) \end{pmatrix} \tag{48}$$

This system is symmetric positive semi-definite since it admits the decomposition

$$\begin{pmatrix} \left(A_F\hat{G}_F\right)^T \\ \Delta t Z^T \end{pmatrix} M_F^{-1} \begin{pmatrix} \left(A_F\hat{G}_F\right)^T \\ \Delta t Z^T \end{pmatrix}^T + \begin{pmatrix} \left(\hat{J}^T A_S\hat{G}_S\right)^T \\ -\Delta t J \end{pmatrix} M_S^{-1} \begin{pmatrix} \left(\hat{J}^T A_S\hat{G}_S\right)^T \\ -\Delta t J \end{pmatrix}^T + \begin{pmatrix} 0 & 0 \\ 0 & \Delta t \bar{k}_S^{-1} \end{pmatrix}, \tag{49}$$

which is the sum of three positive semi-definite matrices. Similar to the discussion in [37], we note that our coupling system is SPD, except in degenerate cases such as regions completely enclosed by Neumann boundary conditions where it is only symmetric positive semi-definite.

5.1. Interpolating and Convolving Velocities

Equation 40 may be extended to use alternative interpolation schemes. Similar to [33], where operators were defined to allow coupling to occur at arbitrary solid sample points, we may rewrite Equation 40 as

$$\hat{\lambda} = H\vec{u}_f - J\vec{v}, \tag{50}$$

where H maps from sample fluid velocity dual cell centers to solid sample points. Note that J may also be made more general than was discussed in Section 3.2. In particular, we consider defining H either as piecewise constant interpolation (i.e., identical to Z^T), higher-order bilinear interpolation, or via a convolutional average. In these instances, the form of Equation 48 remains unchanged except that Z^T is replaced by H and Z is replaced by H^T.

The size of the stencils in both the piecewise constant and bilinear interpolation schemes depends on the resolution of the computational grid; more refined grids sample the fluid more locally to the sub-grid solid. This can be problematic, as discussed in [22, 24], which point out how local interpolation of fluid velocities can cause increasing error in drag force computations as grid resolution increases, especially because the local velocity is influenced by the solid and therefore typically quite different than the free-stream velocities often used to experimentally measure drag coefficients. Hence, some conjecture that it may be more appropriate to consider an interpolation scheme that is not dependent on grid resolution. To demonstrate the ability of our method to handle such desired approaches, we also propose using a convolutional kernel as follows. We define an axisymmetric 2D quartic kernel function in polar coordinates, $f(r) = D\left(r^4 - Br^2 + C\right)$, which is centered at the solid-fluid coupling location and has support restricted to a closed box Ω_I of width s_w centered at the coupling location and excluding the solid geometry, see Figure 19. We choose $B = s_w^2$ in order to make f monotone decreasing from $r = 0$ to $r = \sqrt{2}s_w/2$, and we choose $C = s_w^4/4$ to set the minimum value at $r = \sqrt{2}s_w/2$ identically equal to 0. Finally, D is chosen to enforce the constraint $\int_{\Omega_I} f dV = 1$. Given a satisfactory f, interpolation weights $w_k = \int_{\Omega_I \cap D_k} f dV$ are computed for each dual cell D_k that overlaps Ω_I, noting that the design of f ensures $\sum_k w_k = 1$. We discuss some preliminary results below but emphasize that the point of this discussion is to elucidate the generality of our scheme and of Equation 48; researchers and practitioners may choose their own method of interpolation for H as well as for J. Importantly, we note that since we use a monolithic coupling system, our method is stable for all the interpolation schemes we consider, even for large Δt.

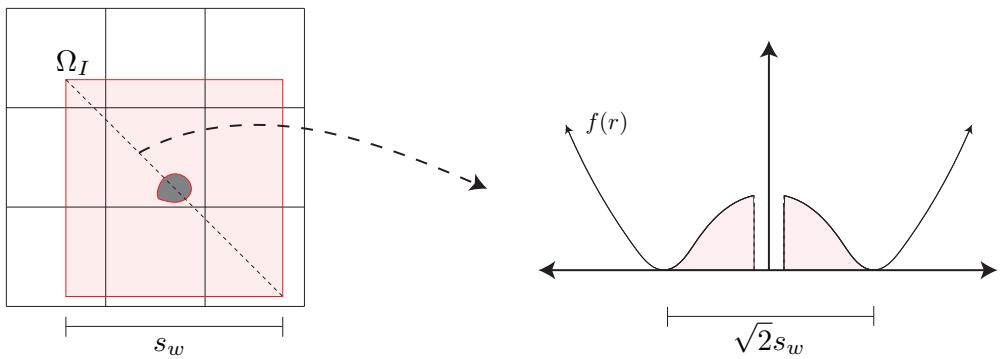


Figure 19: An illustration of our convolutional kernel. *(Left)* the domain of f is Ω_I (shaded red) and excludes the solid (gray). Ω_I is centered at the solid-fluid coupling location, and its outer domain is a square of width s_w. *(Right)* Plotting f for a diagonal cross-section of the interpolation domain. In practice we choose coefficients so that f has minima of 0 on the corners of Ω_I and is thus strictly positive on the interior.

5.2. Examples

Since Equation 48 is SPD, we solve it using a preconditioned conjugate gradient scheme with a simple diagonal preconditioner. Note that we choose a drag coefficient k^j for each solid body j and set $k_i^j = \left(V_i^j / V^j\right) k^j$, where V_i^j is the volume of solid j in dual cell D_i and V^j is the total volume of solid j. This ensures that the total drag seen on each solid is independent of the number of dual cells it overlaps. Also, as in the previous examples, fluid advection is ignored unless otherwise stated. We re-use the two-dimensional wider column setup from Section 3.7 (Figures 14–17) to parametrically explore the effects of grid refinement, temporal refinement, and drag coefficient. For the convolutional scheme, we consider two stencil widths, $s_w = 2$ and $s_w = 24$, in order to demonstrate the effect of using a small or large stencil. Unless otherwise stated, we set $k^j = 0.05$.

We first consider only the piecewise constant interpolation scheme in order to isolate the effect of the introduction of velocity drag. Figures 20–22 use the same parameters as Figures 15–17, respectively, varying Δt, the grid resolution, and both spatial and temporal resolution simultaneously. The overall behavior of each of these examples is similar to the earlier figures, except the presence of velocity drag makes the solid particle more rapidly reach a terminal velocity in all cases. Also note that because of the slower solid velocities caused by fluid drag, the solid spends more time intersecting multiple dual cells and hence there are higher-frequency sub-grid oscillations. Figure 23 illustrates the effect of varying the drag coefficient k^j for a fixed spatial resolution of 63×252 primal grid cells and a fixed time step of $\Delta t = 0.1$s. As the drag coefficient increases, the solid's freedom to move relative to the fluid is diminished, approaching the hydrostatic solution as expected.

We repeat the experiments of Figures 20–22 for each of the aforementioned interpolation schemes and plot the superimposed results. Figure 24 shows the effect of refining the temporal resolution, as in Figure 20. Figures 25 and 26 show the effects of refining the spatial resolution of the computational grid and the spatial and temporal resolutions simultaneously, respectively. In each of the plots, the apparent clustered curves are the different interpolation schemes for each resolution. For example, in Figure 25, the resolutions are the same used in Figure 21; the highest resolution of 511×2044 primal grid cells is the topmost cluster, 255×1020 is the cluster below that, 127×508 is the next lower cluster, and the other five coarser resolutions are all in the bottom cluster. That is, all the interpolation schemes seem to behave similarly, and on coarser grids, they seem to be independent of grid resolution. Of course, this does not mean that there are not examples where one might prefer one interpolation scheme over the other.

5.3. Handling More Particles

One can add collision processing to our time integration loop in order to treat particle-boundary and particle-particle collisions. We propose an iterative scheme for collision resolution that is performed after

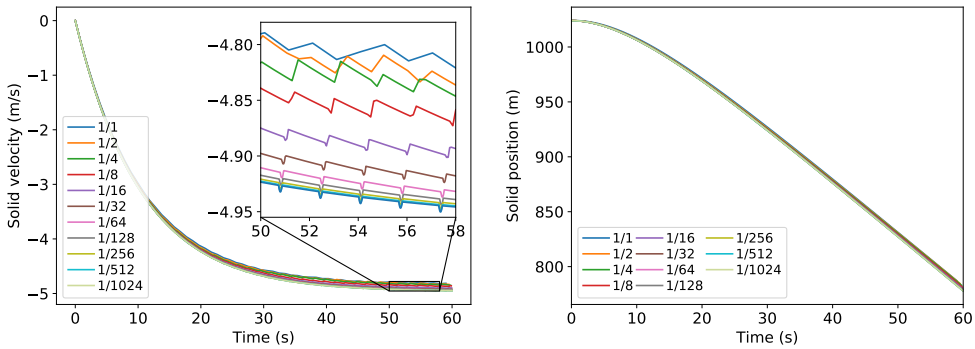


Figure 20: Refining the 2D wide channel example in time, using the velocity drag system Equation 48; labels indicate Δt.

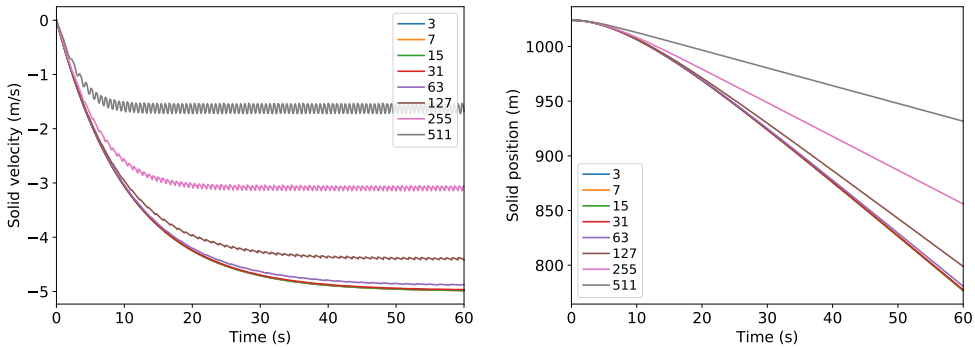


Figure 21: Varying the spatial resolution of the 2D wide channel example, with velocity drag. The labels indicate the number of grid cells in the x-direction; the y-resolution is chosen to maintain square grid cells.

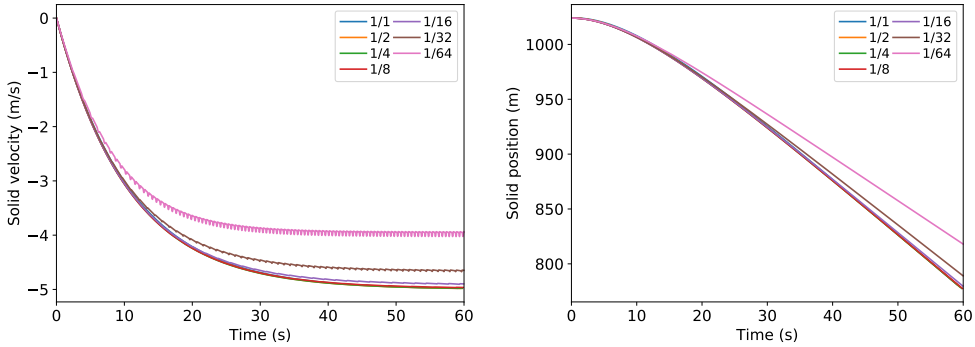


Figure 22: Refining the 2D wide column example simultaneously in space and time. Labels indicate Δt.

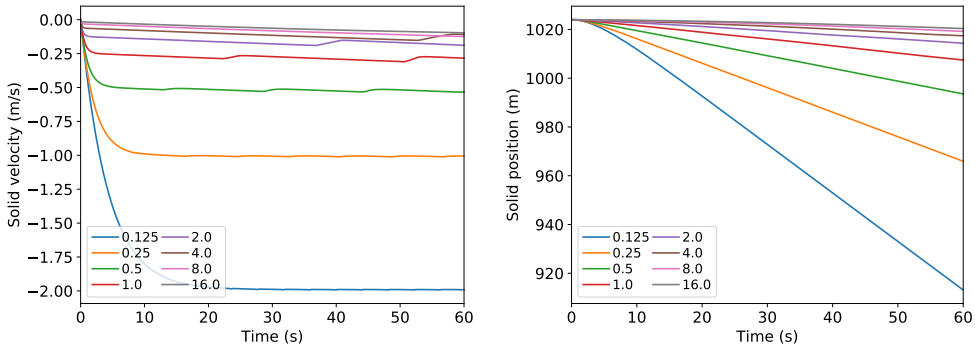


Figure 23: Varying the drag coefficient for the 2D wide column example. Labels indicate values of k^j.

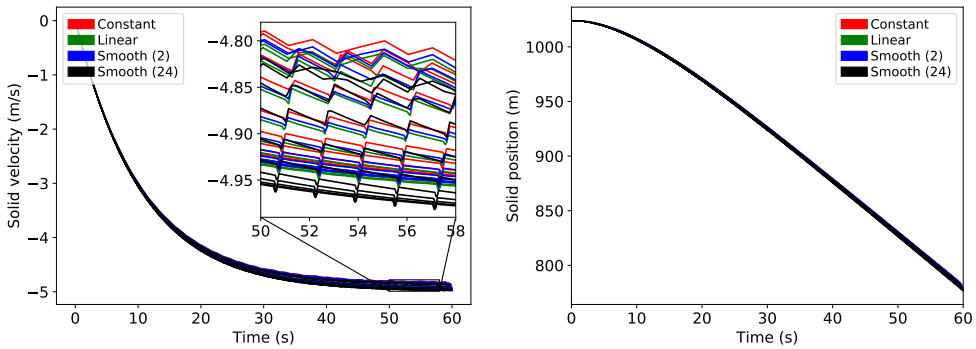


Figure 24: Refining the time step for the 2D wide column example for each interpolation scheme. All interpolation schemes behave in a similar fashion.

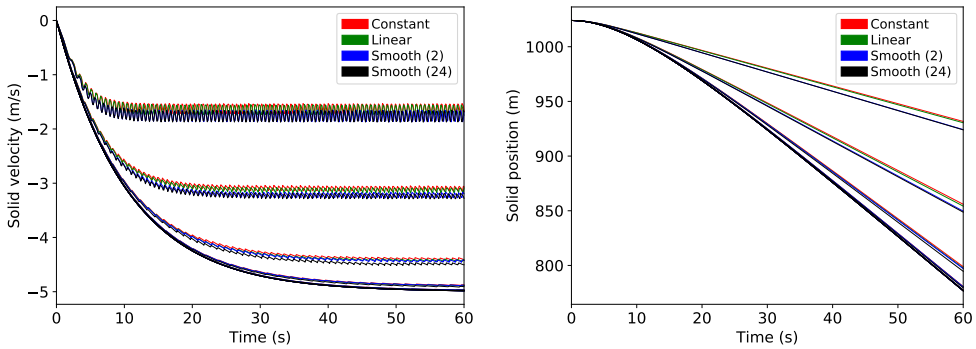


Figure 25: Solving the drag coupling system for a sub-grid particle in a wide 2D channel at varying spatial resolutions for each interpolation scheme. The different interpolation schemes seem to cluster; the clusters represent decreasing grid resolution from top to bottom. The lowest resolutions appear to overlap in the bottommost cluster.

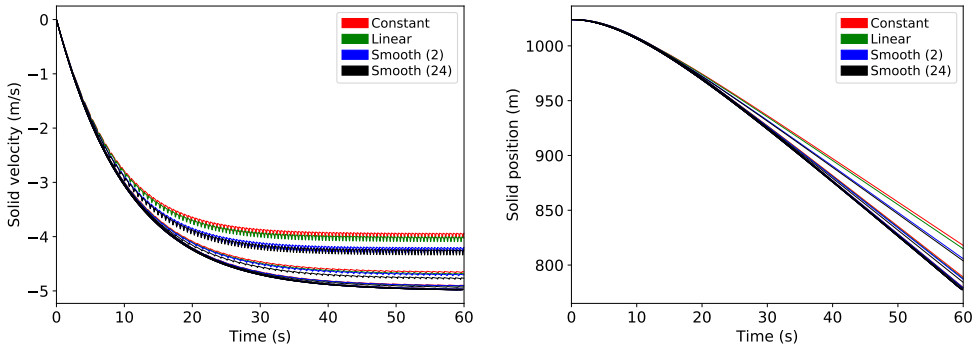


Figure 26: Refining both the spatial resolution and time step for the 2D wide column example. Clusters of curves represent increasing spatiotemporal resolution from bottom to top.

advancing solid positions but before applying explicit forces like gravity. For each of the n iterations, the following steps are performed:

1. Handle particle-boundary collisions. If a solid particle has moved outside a side of the fluid domain with open-container Dirichlet boundary conditions, such as the open top of a domain, we delete it from the simulation. If a particle has penetrated a dual cell with a solid wall boundary condition, we move the particle along the opposite direction of its velocity vector until it exactly touches the wall but does not penetrate it, and project out the component of the particle's velocity in the direction normal to the wall.

2. Handle particle-particle collisions. We build a map at the start of each time step that lists nearby particles for a given solid particle. For each of these neighbor particles, we perform a standard analytic

geometric collision check, draw a line between the two particle centers, and move the two particles apart along that line until they touch but do not interpenetrate. The collision is treated in a perfectly inelastic fashion, updating both particles' velocities so that they have no components pointing towards each other.

A final particle-boundary collision check is performed after n iterations in order to enforce as a hard constraint that all solids should be within the fluid domain. As noted, this scheme is iterative and does not necessarily resolve all particle-particle collisions. However, for sufficiently large numbers of sufficiently small particles, this algorithm will capture the bulk of the collision dynamics; and importantly, it is cheap to compute even with many particles since it only scales linearly with the number of solids. Finally, we note that in order to achieve improved accuracy, we repeat the steps of advancing solid positions and processing collisions m times for each time step, using a time step of $\Delta t/m$ for each "substep" ($m = 10$ in all of our experiments).

Having added collision processing to our scheme, we briefly consider an additional single-particle example, which was proposed in [17]. A tank of fluid with an open (homogeneous Dirichlet) top boundary and solid wall boundary conditions on the other boundaries is created with dimensions 2cm $\times$ 6cm (excluding the boundaries). A circular particle of radius 0.125cm is initially placed at $(1\text{cm}, 4\text{cm})$, with the origin taken to be the bottom left of the fluid domain (i.e. again excluding the boundaries). The particle is allowed to fall under the influence of gravity and drag, and it eventually collides with the bottom wall. Two different solid densities are considered: 1.25g/cm^2 (Figure 27) and 1.5g/cm^2 (Figure 28). We use a spatial resolution of 4×12 primal grid cells in both cases, so that the solid is sub-grid, and a time step of 0.025s. We compare our method with the results of [17, 42, 23]. In both test cases, we are able to produce results that are in line with previous studies of this example in the literature. We note that the drag coefficients we used, $k^j = 2.25$ for Figure 27 and $k^j = 1.75$ for Figure 28, were hand-chosen in a few minutes of manual tuning. We stress that the emphasis of our method is to enable simulation of sub-grid particles with flexibility for modeling dynamics such as drag, rather than to prescribe any particular empirical relation for drag forces and coefficients. Moreover, the comparison works also treated viscous fluid rather than inviscid incompressible flow; as such, our drag coefficients must attempt to account for viscous effects. With an extremely coarse grid (48 grid cells versus e.g. 120k in [23]), a very coarse time step (2.5×10^{-2}s instead of e.g. 1×10^{-4}s in [23]), and without separate treatment of viscosity, our method is still able to (quite efficiently) produce results that closely align with literature values. Moreover, since our particle-boundary collision method is exact to numerical precision, we suffer no spurious numerical overshoots or oscillations when the particle hits the bottom wall; its y-position and velocity reach and remain constant at the correct values. This is in contrast to [17, 42, 23], which all use repulsive force models to attempt to avoid interpenetration; one could argue that such models may be more appropriate when attempting to resolve lubrication forces, viscous boundary layer effects, etc., but these are likely unable to be resolved anyway for very small particles and introduce numerical issues not present with our scheme.

To demonstrate the effect of the collision processing algorithm, we take an open tank of 2m$\times$6m and place 1,000 solid circular particles approximately uniformly randomly distributed within a circle of radius 0.5m. The solid particles are given diameter 1.0cm and density 5.0kg/m^2. The particles are allowed to fall under gravity. A time step of 0.1s is used, and each particle has a drag coefficient of $k^j = 0.1$. Figures 29 and 30 show the results of this simulation using zero and three iterations of our particle-particle collision processing scheme, respectively. In both results, we observe cavitation of the circular arrangement of particles due to the vortices that form in the fluid flow field; this causes some particles to reach the bottom of the tank more quickly, while it causes other particles to swirl upwards with the fluid. Most particles reach the bottom of the tank after about 300 time steps. We also observe that preventing interpenetration of particles has visually significant effects on the results, particularly evident as the particles settle, e.g. Figure 30f versus Figure 29f. The total kinetic energy of all the solids, as well as the total kinetic energy of the fluid, is shown for the case of Figure 30 in Figure 31.

In order to show that our system can scale to simulate many coupled solids, we repeat the previous experiment but instead fill a rectangular region with 100,000 randomly-placed circular particles of uniform density 1.05kg/m^2. We use a computational grid of 32×96 primal grid cells and a time step of $\Delta t = 0.1$s. We use 10 solid substeps per iteration and a uniform solid diameter of 0.5cm. The results of this simulation are shown in Figure 32. By $t = 400$, the majority of the particles settles in the tank, though the light density of the solids means that some particles near the top of the domain are still settling at that time. At this point, we pause to stress how well the particles settle and pack without the use of a material point method,

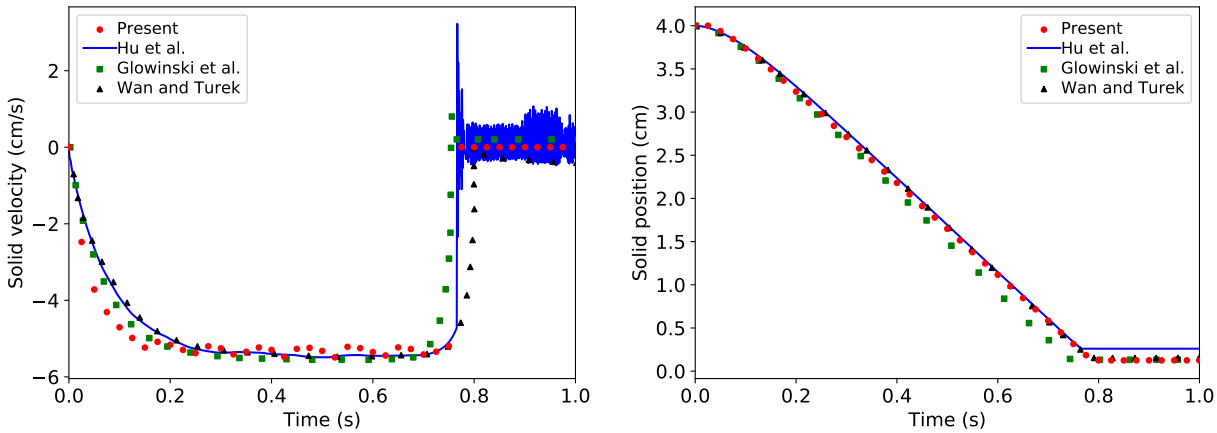


Figure 27: The single-particle sedimentation test of Section 5.3 with solid density 1.25g/cm^2. We carefully manually digitized the data from [17, 42] and plotted our estimates as points in the figures. The authors of [23] provided their data at very small time steps, and thus it is shown as a line in the figures[2]. Our simulation used only 40 time steps, all of which are shown in the figures.

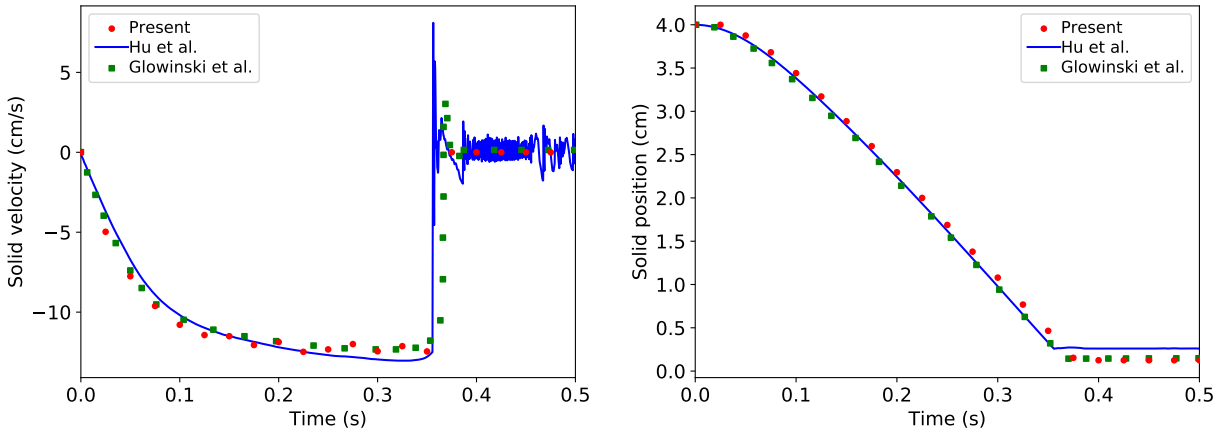
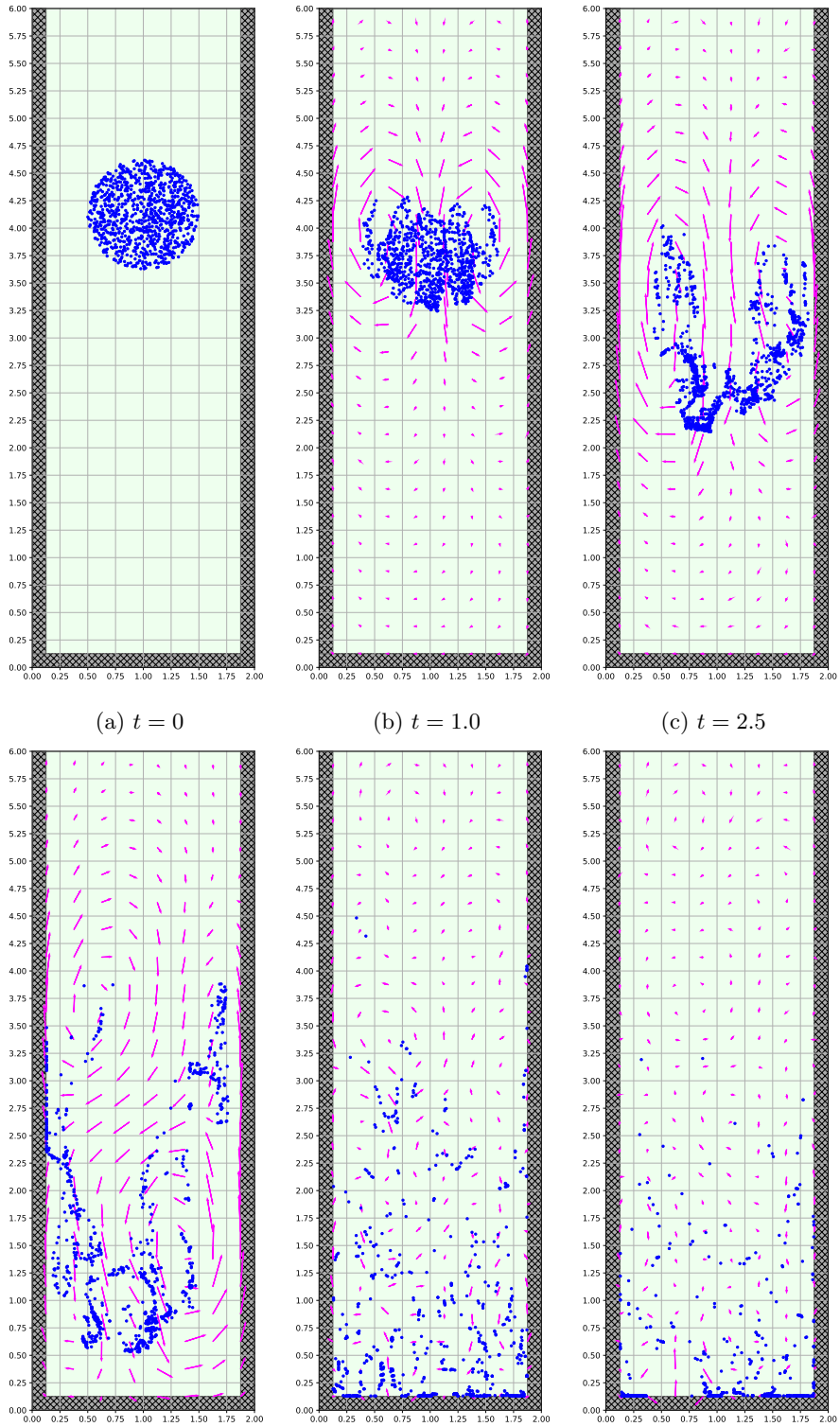


Figure 28: Same as Figure 27 except with solid density 1.5g/cm^2.

plasticity model, etc.; simply treating solid-fluid coupling with accurate buoyancy captures this interesting behavior.

As another large-scale example, we again place 100,000 particles uniformly randomly in a rectangular region of the tank, similar to the previous test. However, each particle is now given one of three distinct densities: the upper group of heavier particles has density 2.0kg/m^2, the middle group of particles has density 1.0kg/m^2, and the lower group of lighter particles has density 0.01kg/m^2; all solids use a drag coefficient of $k^j = 0.1$. A time step of $\Delta t = 0.1$s is used. The results are shown in Figure 33. The heavy particles quickly flow downwards, while the light particles tend to flow upwards. After they swirl past each other, the majority of the light particles leaves the domain, while the majority of heavy particles settles at the bottom of the tank; a mixture of neutrally-buoyant particles with straggling light and heavy particles is still present in Figure 33h. Finally, we observe that some columnar clustering occurs near the centerline of the domain early on in both Figures 32 and 33, and note that clustering of particles into columns has been observed in other sedimentation simulations [41, 27]. This makes sense in the context of Rayleigh-Taylor instabilities where plumes of lighter fluid bubble upward and spikes of heavier fluid jet downward.

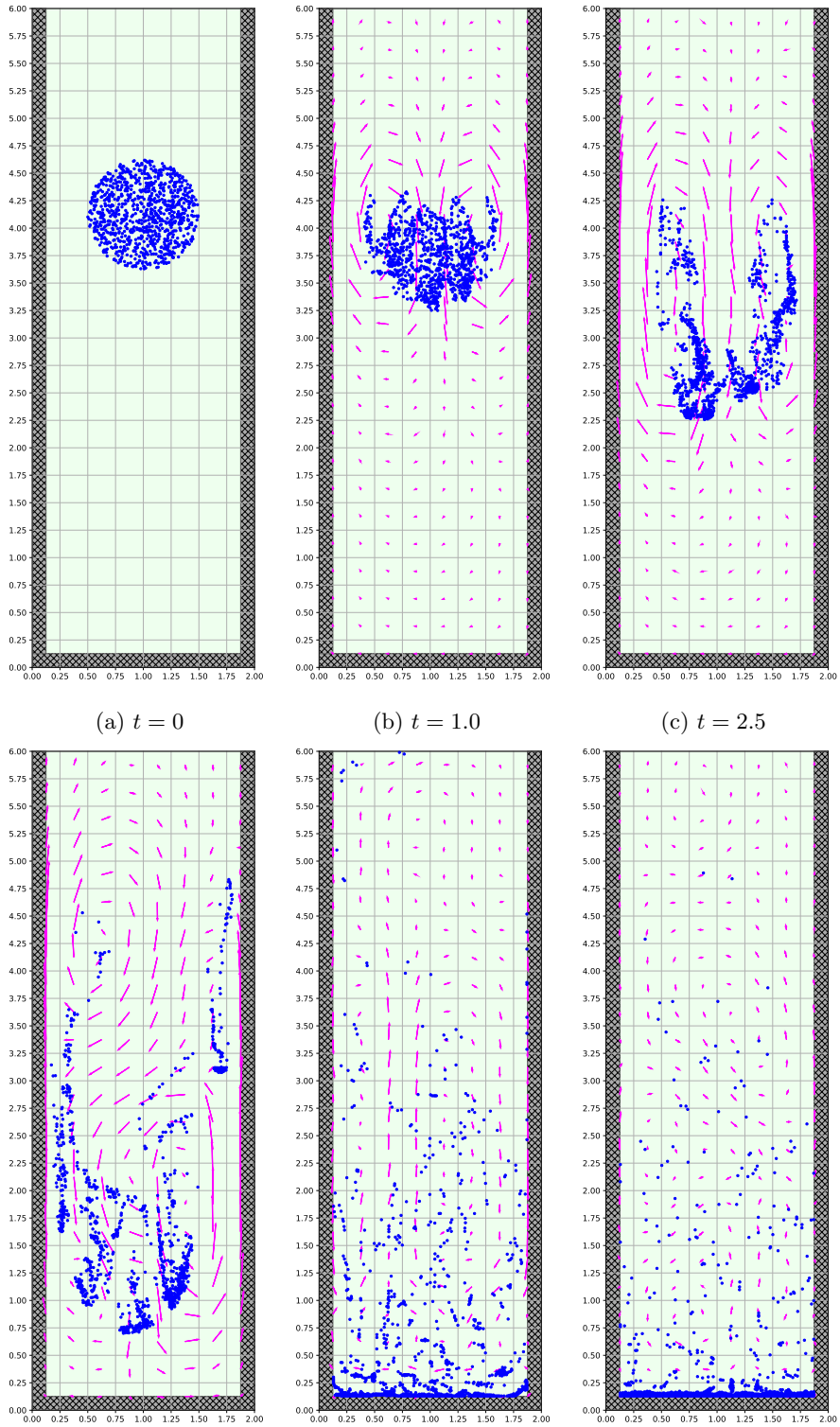
[2]Note that their data is more oscillatory than what appears in the paper; the authors report using a smoothing for the paper (personal communication).



(a) $t = 0$ (b) $t = 1.0$ (c) $t = 2.5$

(d) $t = 5.0$ (e) $t = 15.0$ (f) $t = 30.0$

Figure 29: 1,000 particles fall in a tank, using zero iterations of particle-particle collision processing. There appear to be many fewer solids towards the end, compared to Figure 30, due to the very large number of incorrectly overlapping particles. Cell-averaged fluid velocities are shown as pink vectors. Velocity and solid geometries are uniformly scaled for ease of visibility.



(a) $t = 0$ (b) $t = 1.0$ (c) $t = 2.5$

(d) $t = 5.0$ (e) $t = 15.0$ (f) $t = 30.0$

Figure 30: 1,000 particles fall in a tank, using three iterations of particle-particle collision processing. Cell-averaged fluid velocities are shown as pink vectors. Velocity and solid geometries are uniformly scaled for ease of visibility.

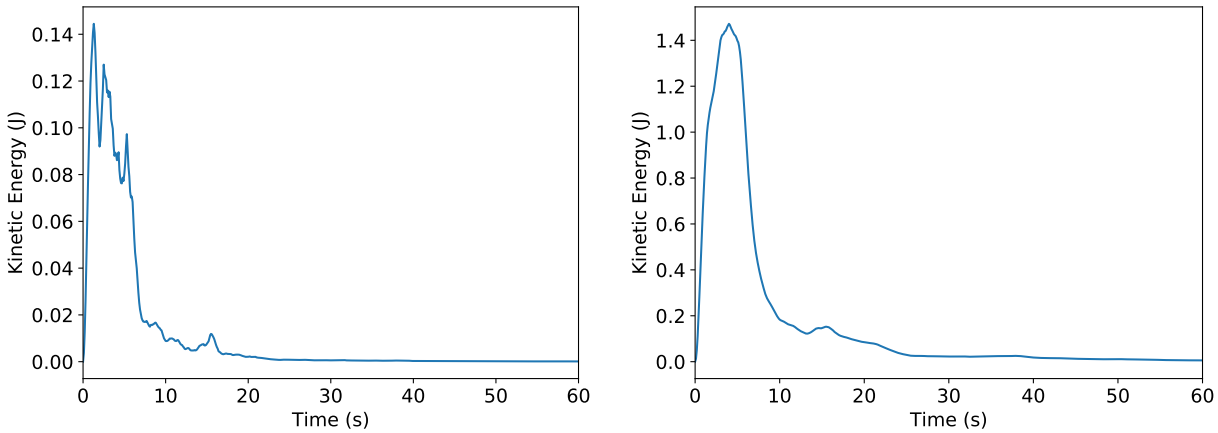


Figure 31: The total kinetic energy of all the solid particles (left) and all of the fluid dual cells (right) over time for the example of Figure 30. Kinetic energy peaks as particles fall in the fluid, then quickly falls off as particles settle. The total solid kinetic energy contains a few sharp dips corresponding to collisions with walls, which reduces kinetic energy of the particles.

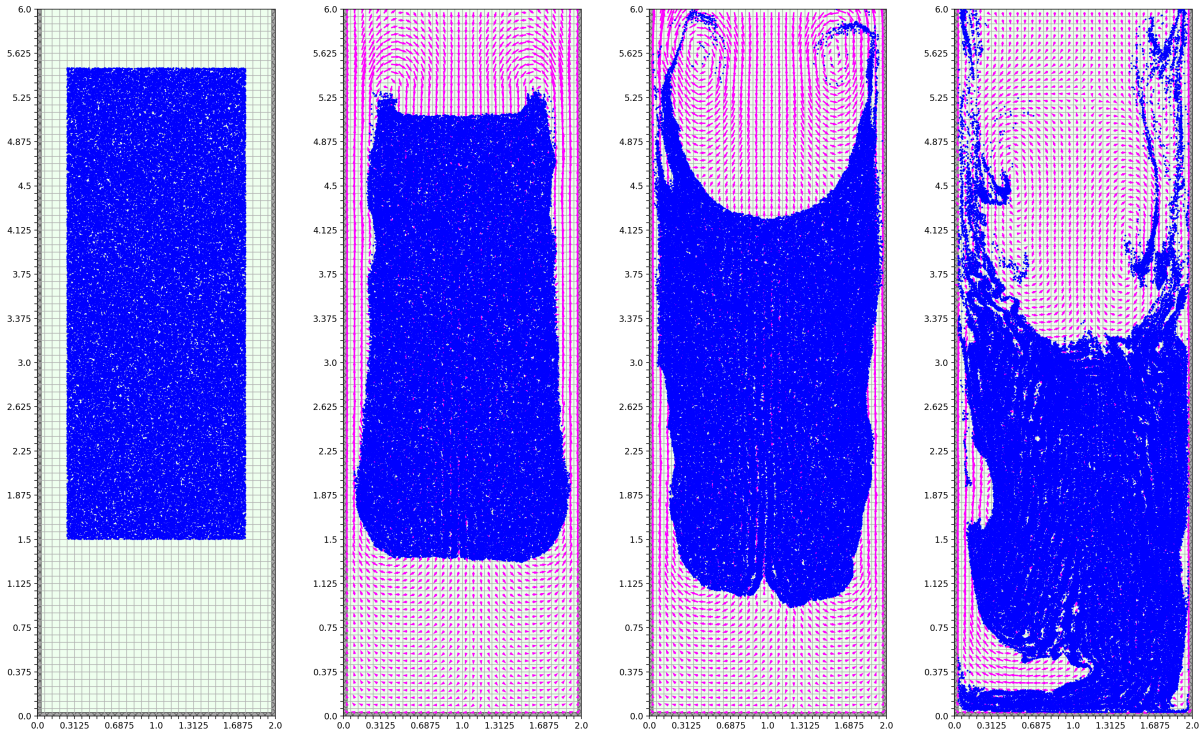
6. More Resolved Solids

While we have thus far concentrated on small particles and coarse computational grids, it is natural to consider what happens to our scheme when a solid becomes large enough relative to the spatial resolution of the grid so that it entirely contains several or more grid cells. Our composite velocity formulation, as discussed in Section 3.3, naturally accounts for this case—fully solid dual cells simply have $w_{f,i}$ as the appropriate interpolated solid velocity, and pressure degrees of freedom are not needed between fully solid dual cells. Thus, the structure of the coupling equations, e.g. Equations 22 and 48, automatically remains valid without any changes or specialized boundary conditions for more resolved solids. Moreover, since we assume the rectangular prism construction throughout, behavior such as neutral buoyancy is still handled correctly for more resolved solids.

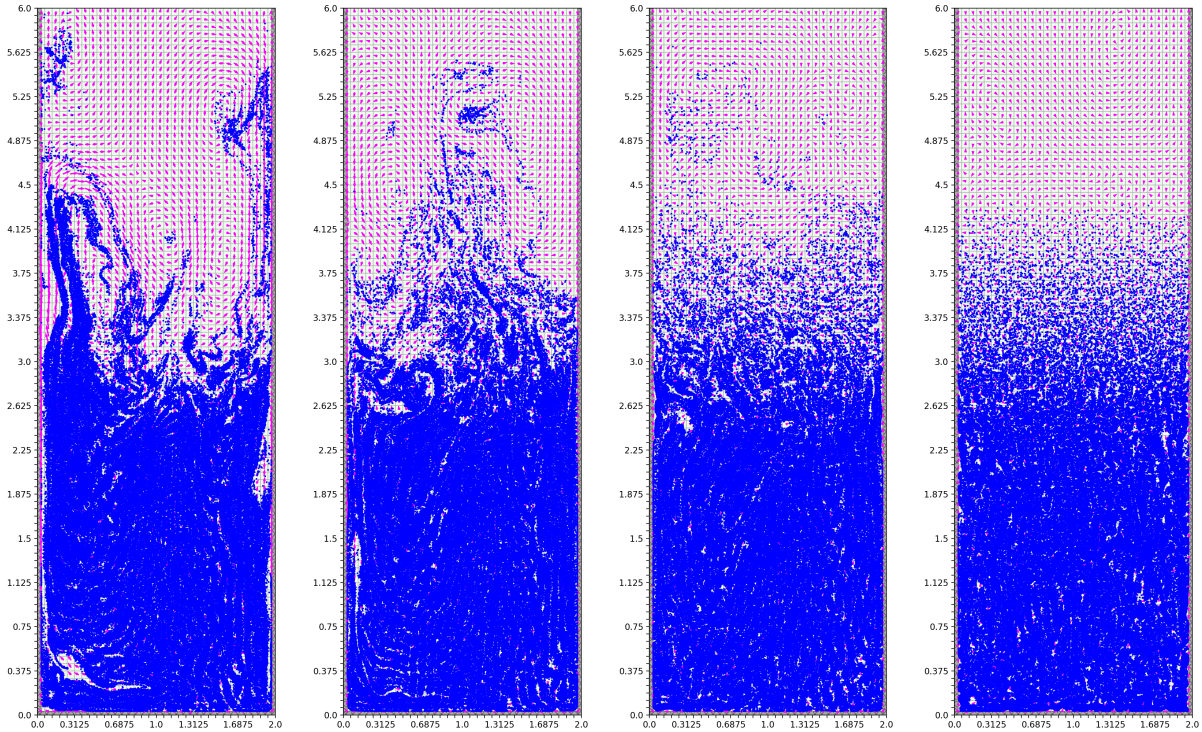
We consider the behavior of more resolved solids under our formulation in Figures 34–35, using the same solid and computational domain as Figures 27–28. Velocity drag is ignored for these basic tests. As such, the velocities attained by the falling sphere in Figures 34–35 are significantly more negative than those in Figures 27–28, though the overall behavior is similar. First-order self-convergence (average order 0.93 for velocity and 1.00 for position) is observed in Figure 34 as the time step is refined, using a fixed computational grid resolution of 64×192 primal grid cells. First-order self-convergence (average order 1.08 for velocity and 1.04 for position) is also seen as the example is spatially refined using a fixed time step of $\Delta t = 0.002$s, see Figure 35. Note the oscillations that sub-grid solids admit as they cross dual cell boundaries become increasingly insignificant in the case of more resolved bodies. We also consider refining simultaneously in both space and time, as shown in Figure 36. In this test, first-order self-convergence emerges as the computational grid and time step are refined, although convergence is degraded at very coarse resolutions.

6.1. Comparison with Klein et al.

Similar to our method, [12, 43, 28, 4] and others use schemes where solid and fluid fluxes are decomposed based on volume fractions. For the sake of comparison with [28, 4], consider a flux in the positive x-direction through a computational cell with the geometry shown in Figure 37a. We note that both the method of [28, 4] and our method use the same fluid flux through the "unshielded" region of the cell unoccluded by any solid in the direction of the flux. For the "shielded" regions of the cell, our rectangular prism construction yields the effective geometries shown in Figure 37b, while [28, 4] obtain the geometry shown in Figure 37c using the average distance from the cell face to the solid-fluid interface. Nonetheless, both approaches result in an identical volume-weighting of the solid and fluid fluxes. The efficacy of our different visual interpretation can be better seen in Figures 37d, 37e, and 37f, where the same geometry is shown vertically instead of horizontally. In the case of incompressible flow and buoyancy, Figure 37e depicts a solid with its natural width for the upward-pushing pressure, whereas Figure 37f depicts the solid as wider and requires devising a

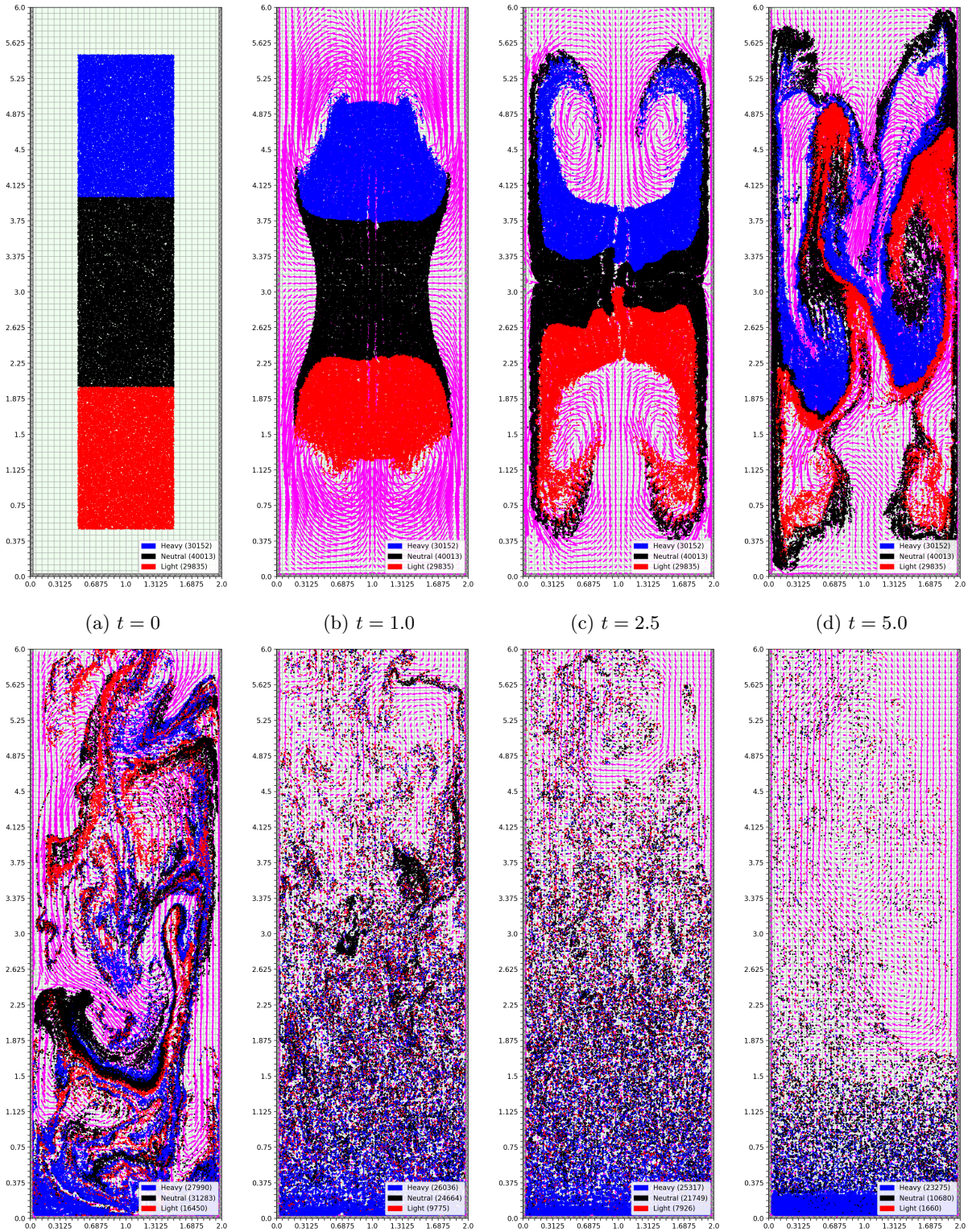


(a) $t = 0$ (b) $t = 5.0$ (c) $t = 10.0$ (d) $t = 30.0$



(e) $t = 40.0$ (f) $t = 60.0$ (g) $t = 100.0$ (h) $t = 400.0$

Figure 32: 100,000 particles fall in a tank, demonstrating sedimentation behavior. Cell-averaged fluid velocities are shown as pink vectors. Velocity vectors and solid geometries are uniformly scaled for ease of visibility.



(a) $t = 0$ (b) $t = 1.0$ (c) $t = 2.5$ (d) $t = 5.0$

(e) $t = 10.0$ (f) $t = 50.0$ (g) $t = 100.0$ (h) $t = 500.0$

Figure 33: 100,000 particles of varying density flow past each other. Velocity and solid geometries are uniformly scaled for ease of visibility. Numbers in the legends indicate how many particles with a given density are still in the domain.

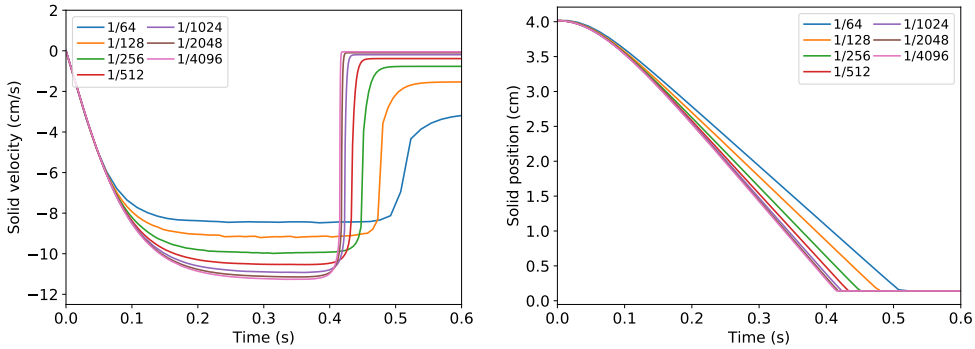


Figure 34: Refining the more-resolved sphere drop example in time. Labels indicate the time step used. We measure self-convergence at $t = 0.25$, obtaining rates 0.44, 0.70, 0.89, 1.09, and 1.56 for velocity (average 0.93) and 0.57, 0.81, 0.95, 1.12, and 1.55 for position (average 1.00).

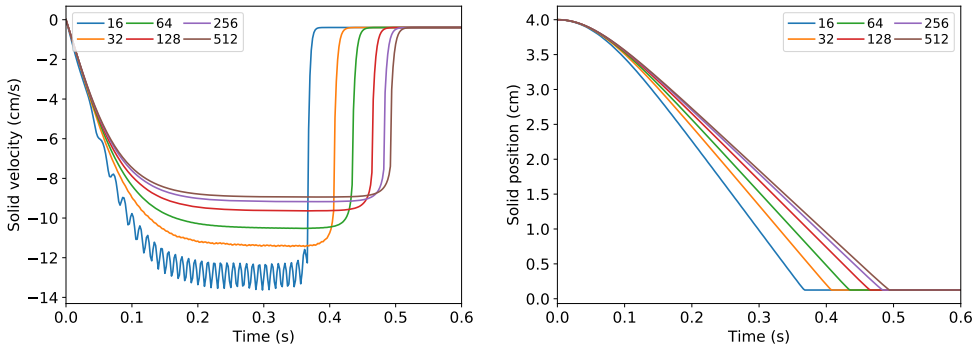


Figure 35: Refining the more-resolved sphere drop example in space. Labels indicate the number of x-direction primal grid cells used; the y-direction resolution is chosen to maintain square grid cells. We measure self-convergence at $t = 0.25$, obtaining rates 0.93, 0.64, 1.18, and 1.57 for velocity (average 1.08) and 0.80, 0.69, 1.13, and 1.56 for position (average 1.04).

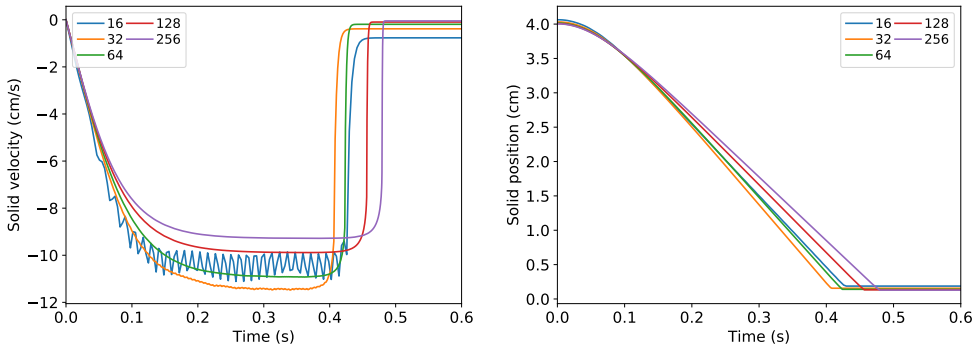


Figure 36: Refining the more-resolved sphere drop example simultaneously in space and time. Labels indicate the number of x-direction primal grid cells used; the y-direction resolution is chosen to maintain square grid cells. The time step varies by powers of two from 2^{-8} at the coarsest grid resolution to 2^{-12} at the finest resolution. We measure self-convergence at $t = 0.25$, obtaining rates -1.49, 0.40, and 1.43 for velocity and -0.45, 0.45, and 1.39 for position.

sub-cell pressure. We note that this distinction matters when thinking of applying a fluid pressure p across a face area A, and that both models give the same answer when considering the volumetric pressure gradient $V\nabla p$. We caveat that there may be situations, especially in compressible flow, where the conceptualizations in Figures 37c and 37f are more useful.

6.2. Mixing Sub-Grid and More Resolved Solids

Finally, we demonstrate the behavior of our method when simulating sub-grid and more resolved bodies simultaneously. We place a more resolved circle in the large tank of Figures 29–33 and make it stationary

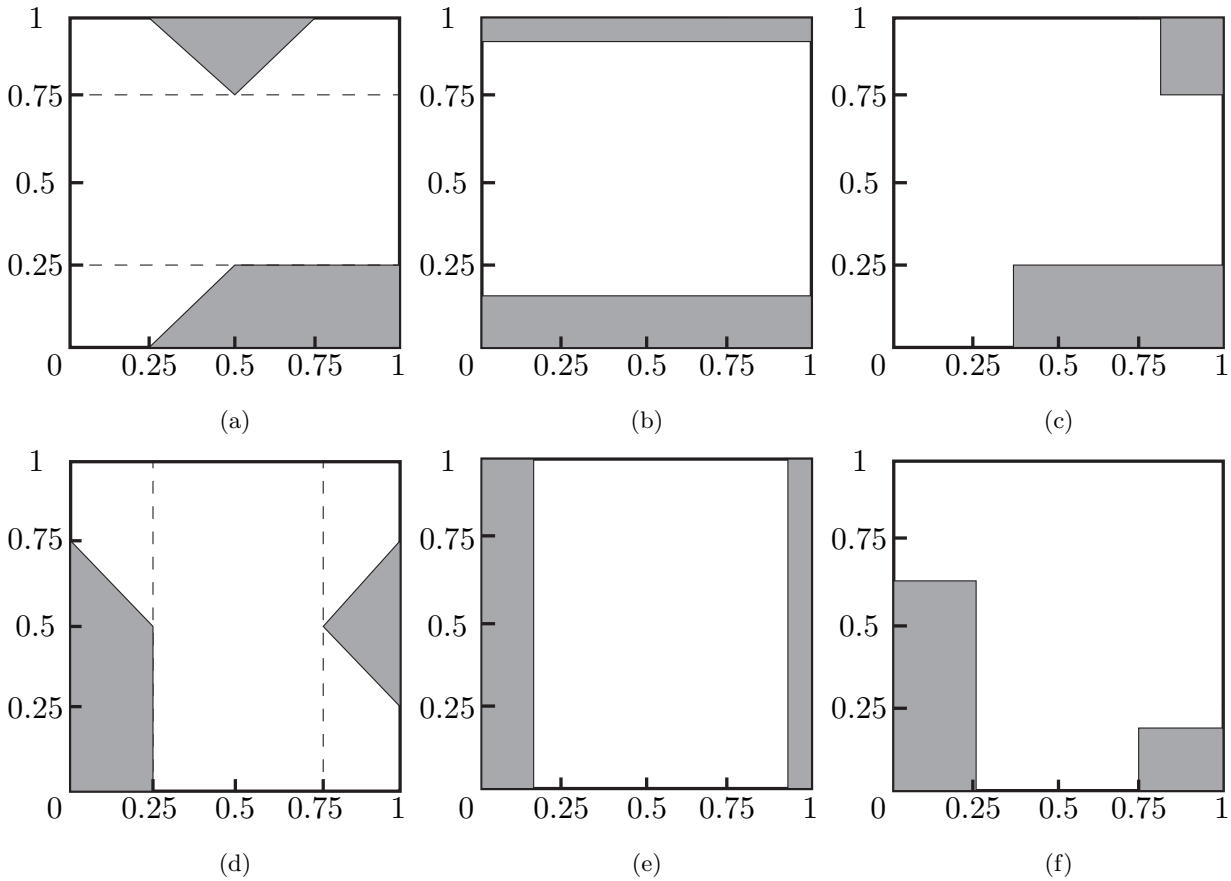
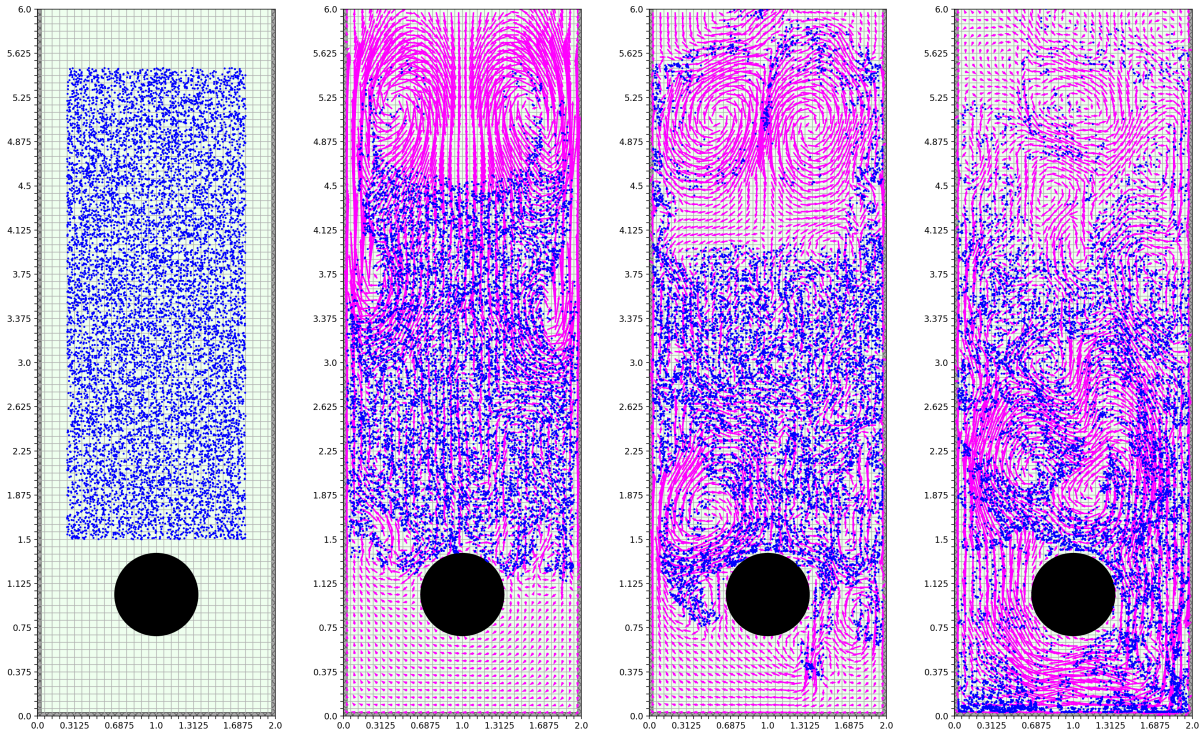


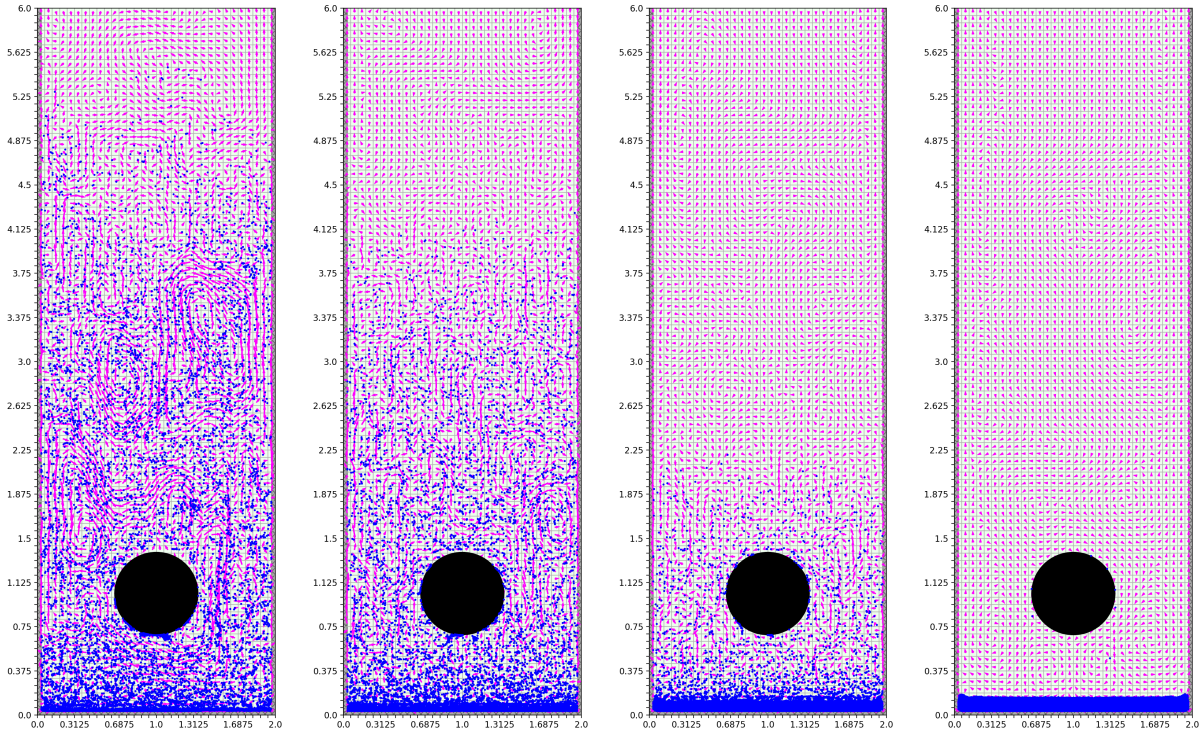
Figure 37: Actual ((a), (d)) and effective solid geometries for a trapezoid and a triangle under our rectangular prism construction ((b), (e)) and the interface shielding of Klein et al. [28, 4] ((c), (f)). (a–c) consider a flux in the positive x-direction, while (d–f) consider a y-direction flux. Both methods compute fluxes using effective solid geometries based on volumetric information about the portion of a solid in a computational cell.

by giving it neutral density and resetting its velocity to zero after each coupled solve. We then add 10,000 sub-grid particles in the tank and observe the results. As before, we use a time step of $\Delta t = 0.1$s, a spatial resolution of 32×96 primal grid cells, and a drag coefficient of $k^j = 0.1$ for all of the solids. Figure 38 shows the case when the sub-grid particles are heavy (density $\rho = 5.0\text{kg/m}^2$) and settle around the more resolved circle, and Figure 39 shows the case when the small particles are light (density $\rho = 0.01\text{kg/m}^2$) and flow around the more resolved circle and out of the tank. In both cases, we notice that our simple collision algorithm causes some small particles to temporarily stick to the more resolved solid (e.g. in Figures 38h and 39h), though these small particles do eventually move around the larger circle.

We combine sub-grid and more resolved moving solids in the simulation shown in Figure 40, which uses 3,536 randomly-placed solids of random radius and density. Sedimentation behavior is observed to roughly correspond to density: the heaviest solids tend to settle at the bottom of the tank, while lighter solids tend to flow upward out of the tank. We note that when running this simulation, we encountered nonphysical situations where so many solids overlapped in a dual cell that the computed fluid volume fraction was negative, despite repeated iterations of solid-solid and solid-boundary collision processing. To alleviate this, we modified our collision processing algorithm by adding a "shock propagation" step as in [19]. We sort all the solids in ascending order according to their minimum y-coordinate; then, for each solid in the sorted list, we resolve any collisions with objects below the current solid by moving the current solid vertically upwards. This sweep resolves all collisions in the domain; however, we note that this step is an overconservative mechanism applied only in the rare case when too many solids are overlapping. While not the focus of our work, more sophisticated collision processing algorithms are discussed in [19] and the references therein.

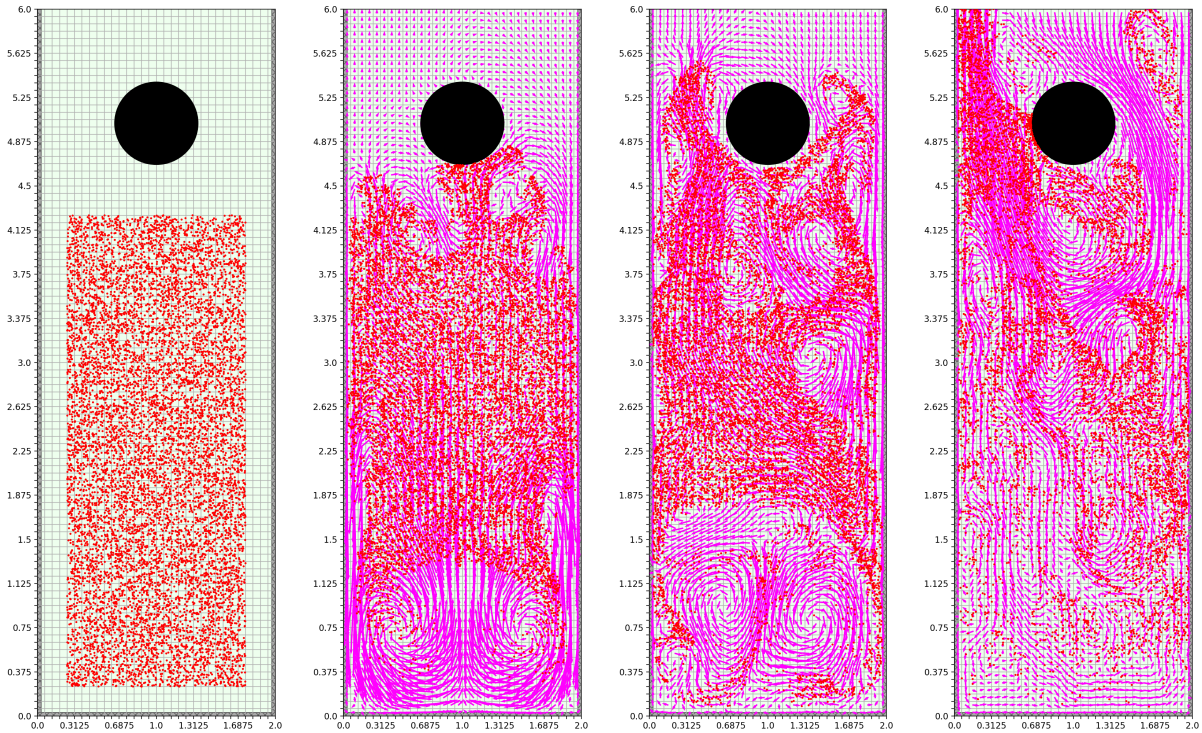


(a) $t = 0$ (b) $t = 2.5$ (c) $t = 5.0$ (d) $t = 10.0$

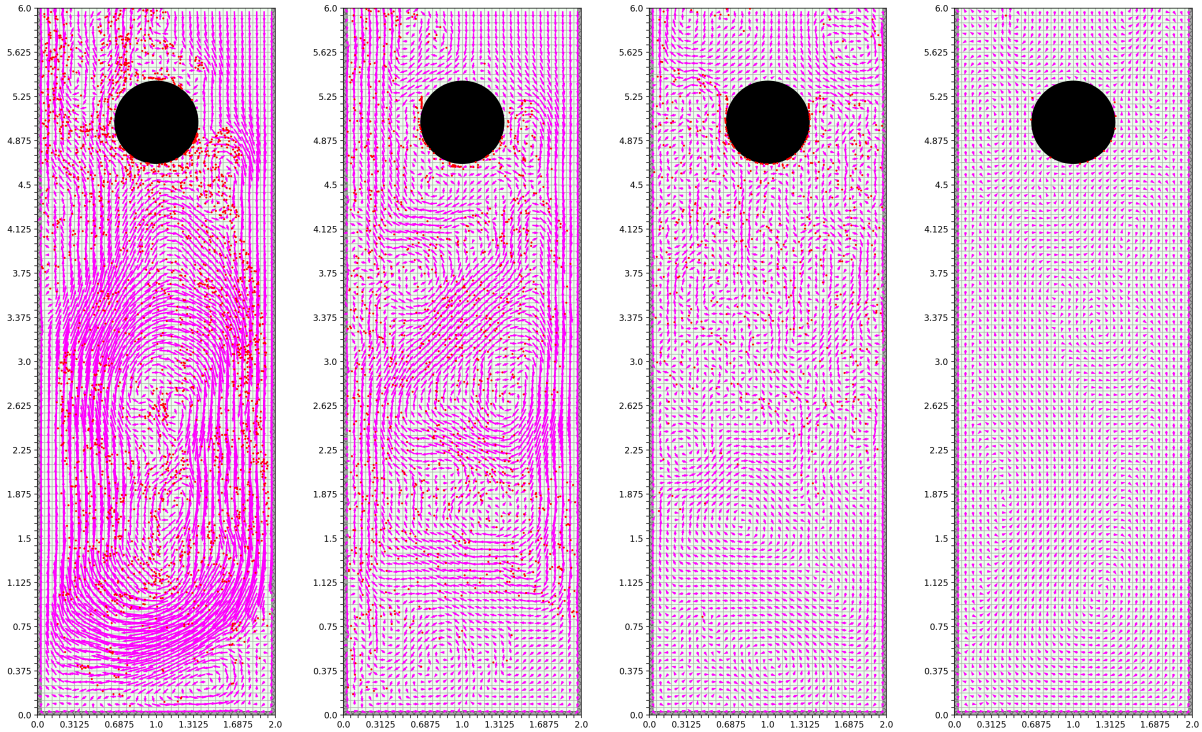


(e) $t = 25.0$ (f) $t = 50.0$ (g) $t = 100.0$ (h) $t = 200.0$

Figure 38: 10,000 heavy sub-grid particles (density $\rho = 5.0\text{kg/m}^2$) settle around a stationary more resolved sphere. Velocity and solid geometries are uniformly scaled for ease of visibility.

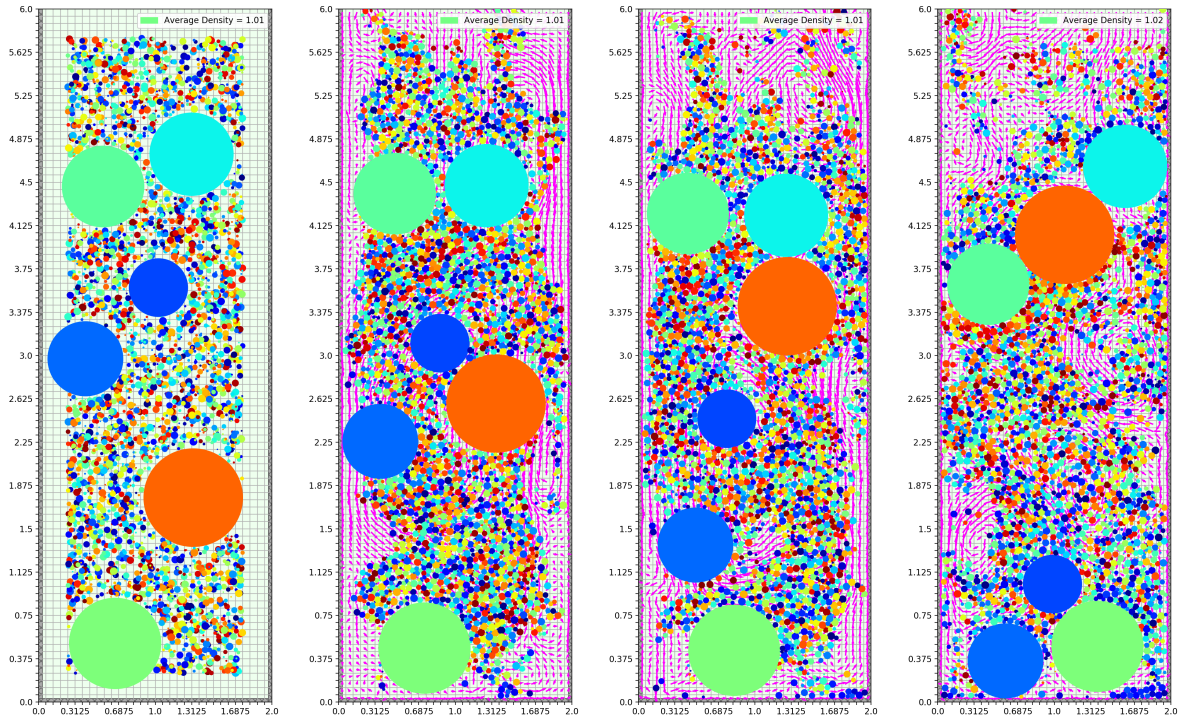


(a) $t = 0$ (b) $t = 2.5$ (c) $t = 5.0$ (d) $t = 10.0$

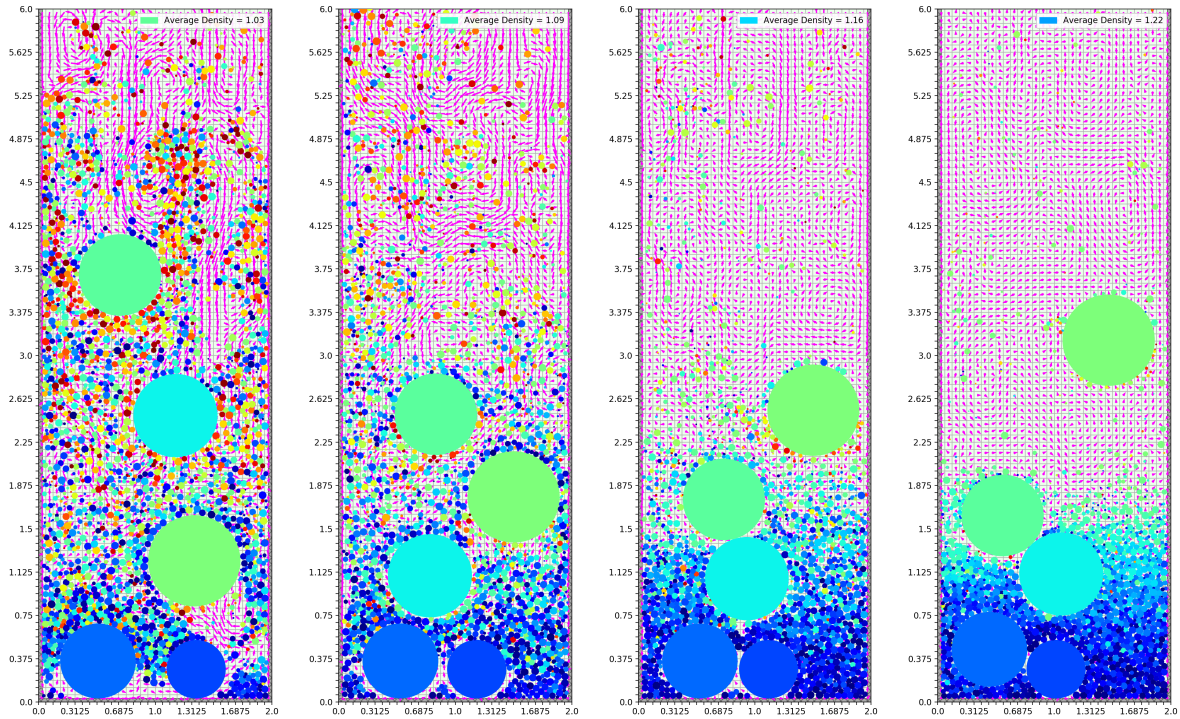


(e) $t = 25.0$ (f) $t = 50.0$ (g) $t = 100.0$ (h) $t = 200.0$

Figure 39: 10,000 light sub-grid particles (density $\rho = 0.01\text{kg/m}^2$) flow upward around a stationary more resolved sphere. Velocity and solid geometries are uniformly scaled for ease of visibility.



(a) $t = 0$ (b) $t = 2.5$ (c) $t = 5.0$ (d) $t = 10.0$



(e) $t = 25.0$ (f) $t = 50.0$ (g) $t = 100.0$ (h) $t = 200.0$

Figure 40: 3,536 more resolved and sub-grid solids of random density and radius flow in an open tank. Solids are colored according to density: red is light, green is neutral, and blue is heavy. Densities range from 0.5kg/m^2 to 1.5kg/m^2; legends indicate the average density of the extant solids at each time.

7. Conclusions and Future Work

We have presented a monolithic, symmetric positive-definite, two-way coupling system for incompressible flow and rigid bodies of various scales. In particular, our scheme naturally handles the coupling of solids that are much smaller than the size of a computational grid cell up through solids that are large enough to be resolved by many grid cells. We incorporate velocity drag into our fully-implicit coupling system, enabling more appropriate physical behavior especially for small solids. Additional features of our method include correctly handling the case of neutral buoyancy for sub-grid and more resolved solids, and the ability to derive an efficient SPD formulation allowing us to run examples with large numbers of solids. Moreover, we remarked upon the generality of our approach to support other models for face area fractions, drag coefficients, fluid and solid velocity interpolations, etc.

There are a number of possible avenues for extending the present work. For instance, one may consider the treatment of dynamic thin shells such as parachutes in surrounding flow. For a grid cell containing a segment of the parachute, one may wish to define multiple $A_{F,i}$ for the distinct fluid components of the cell, which are separated by the shell and thus should be allowed to travel with different velocities. On a similar note, one may consider multimaterial flows, where it should be possible for fluid components within a grid cell to travel at differing speeds due to their distinct material properties. In addition to extensions of the fluid area fractions, one may consider different models for the $A^j_{S,i}$. For instance, one may seek models for $A^j_{S,i}$ that produce more accurate motion in situations such as those depicted in Figure 1d.

We are also interested in the application of our method to the simulation of particle-laden flow and sedimentation, as studied in e.g. [12, 43]. Our coupling scheme does not suffer from stability issues associated with explicit drag forces as discussed in those works, and we are not limited to small time steps. However, for specifically targeting engineering applications of particle-laden flow, one would likely need to more carefully model inter-solid forces, such as Drucker-Prager elastoplasticity, for more dense packing, etc. More broadly, we note that while papers like [12] come from the computer graphics community, many ideas in graphics are equally applicable in computational physics; for instance, the affine particle-in-cell method for fluid simulation [25, 26], a material point method for snow simulation [40, 13], simulation of cutting and crack propagation in triangulated domains [39, 36], and the particle level set method [9, 10].

Finally, we are interested in using computer vision and machine learning techniques to enhance our simulation methods, inspired by the discussion in [15]. In particular, we remark that drag coefficients greatly affect the motion of small solid particles in fluid, yet they are usually based on empirical models which might have errors on the order of 10% [7]. Thus, we are interested in the possibility of creating a "database" of drag coefficients and building a statistical model on top of this collection which could be queried in order to automatically assign accurate drag coefficients to solids during a simulation. An ongoing project includes using high-speed cameras and computer vision techniques to automatically collect samples for such a database from real-world experiments.

Acknowledgments

This line of research was motivated by a question posed by Sanjiva Lele during a Ph.D. student's oral defense. Research was supported in part by ONR N00014-13-1-0346, ONR N00014-17-1-2174, ARL AHPCRC W911NF-07-0027, and a grant from BHP Billiton. D.H. was also supported by the Gerald J. Lieberman Fellowship and by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. Computing resources were provided in part by ONR N00014-05-1-0479.

References

[1] Vinicius C. Azevedo, Christopher Batty, and Manuel M. Oliveira. Preserving geometry and topology for fluid flows with thin obstacles and narrow gaps. *ACM Trans. Graph.*, 35(4):97:1–97:12, July 2016.

[2] J.W. Banks, W.D. Henshaw, D.W. Schwendeman, and Qi Tang. A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part I: Model problem analysis. *Journal of Computational Physics*, 343:432–468, 2017.

[3] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid-fluid coupling. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[4] W.P. Bennett, N. Nikiforakis, and R. Klein. A moving boundary flux stabilization method for Cartesian cut-cell grids using directional operator splitting. *Journal of Computational Physics*, 368:333–358, 2018.

[5] Mark Carlson, Peter J. Mucha, and Greg Turk. Rigid fluid: Animating the interplay between rigid bodies and fluid. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 377–384, New York, NY, USA, 2004. ACM.

[6] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967.

[7] R. Clift, J.R. Grace, and M.E. Weber. *Bubbles, Drops, and Particles.* Dover Civil and Mechanical Engineering. Dover Publications, 2005.

[8] Doug Enright, Duc Nguyen, Frederic Gibou, and Ron Fedkiw. Using the particle level set method and a second order accurate pressure boundary condition for free-surface flows. In *Proceedings of FEDSM'03*, 2003 4th ASME JSME Joint Fluids Engineering Conference, pages 337–342. ASME, 2003.

[9] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[10] Douglas Enright, Stephen Marschner, and Ronald Fedkiw. Animation and rendering of complex water surfaces. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 736–744, New York, NY, USA, 2002. ACM.

[11] Christiane Förster, Wolfgang A. Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278–1293, 2007.

[12] Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. Animating fluid sediment mixture in particle-laden flows. *ACM Trans. Graph.*, 37(4):149:1–149:11, July 2018.

[13] J. Gaume, T. Gast, J. Teran, A. van Herwijnen, and C. Jiang. Dynamic anticrack propagation in snow. *Nature Communications*, 9, 2018.

[14] O. Génevaux, A. Habiri, and J.-M. Dischler. Simulating fluid-solid interaction. In *Graph. Interface*, pages 31–38, June 2003.

[15] Frederic Gibou, David Hyde, and Ron Fedkiw. Sharp interface approaches and deep learning techniques for multiphase flows. *Journal of Computational Physics*, 2018.

[16] Frédéric Gibou and Chohong Min. Efficient symmetric positive definite second-order accurate monolithic solver for fluid/solid interactions. *Journal of Computational Physics*, 231(8):3246–3263, 2012.

[17] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, and J. Périaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *Journal of Computational Physics*, 169(2):363–426, 2001.

[18] Jón Tómas Grétarsson, Nipun Kwatra, and Ronald Fedkiw. Numerically stable fluid-structure interactions between compressible flow and solid structures. *Journal of Computational Physics*, 230(8):3062–3084, 2011.

[19] Eran Guendelman, Robert Bridson, and Ronald Fedkiw. Nonconvex rigid bodies with stacking. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 871–878, New York, NY, USA, 2003. ACM.

[20] Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph.*, 24(3):973–981, July 2005.

[21] Francis H. Harlow and J. Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189, 1965.

[22] J.A.K. Horwitz and Ali Mani. Accurate calculation of Stokes drag for point-particle tracking in two-way coupled flows. *Journal of Computational Physics*, 318:85–109, 2016.

[23] Yang Hu, Decai Li, Shi Shu, and Xiaodong Niu. Modified momentum exchange method for fluid-particle interactions in the lattice Boltzmann method. *Phys. Rev. E*, 91:033301, Mar 2015.

[24] Peter J. Ireland and Olivier Desjardins. Improving particle drag predictions in Euler–Lagrange simulations with two-way coupling. *Journal of Computational Physics*, 338:405–430, 2017.

[25] Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. The affine particle-in-cell method. *ACM Trans. Graph.*, 34(4):51:1–51:10, July 2015.

[26] Chenfanfu Jiang, Craig Schroeder, and Joseph Teran. An angular momentum conserving affine-particle-in-cell method. *Journal of Computational Physics*, 338:137–164, 2017.

[27] Takeo Kajishima. Influence of particle rotation on the interaction between particle clusters and particle-induced turbulence. *International Journal of Heat and Fluid Flow*, 25(5):721–728, 2004. Selected papers from the 4th International Symposium on Turbulence Heat and Mass Transfer.

[28] R. Klein, K. R. Bates, and N. Nikiforakis. Well-balanced compressible cut-cell simulation of atmospheric flow. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1907):4559–4575, 2009.

[29] F. Losasso, G. Irving, E. Guendelman, and R. Fedkiw. Melting and burning solids into liquids and gases. *IEEE Transactions on Visualization and Computer Graphics*, 12(3):343–352, May 2006.

[30] Wenlong Lu, Ning Jin, and Ronald Fedkiw. Two-way coupling of fluids to reduced deformable bodies. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '16, pages 67–76, Aire-la-Ville, Switzerland, Switzerland, 2016. Eurographics Association.

[31] Mark K. McVay. Spall damage of concrete structures. Technical Report SL-88-22, Army Corps of Engineers Waterways Experiment Station, Vicksburg, MS, USA, June 1988.

[32] Christopher C. Paige and Michael A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[33] Saket Patkar, Mridul Aanjaneya, Wenlong Lu, Michael Lentine, and Ronald Fedkiw. Towards positivity preservation for monolithic two-way solid-fluid coupling. *Journal of Computational Physics*, 312:82–114, 2016.

[34] Linhai Qiu, Yue Yu, and Ronald Fedkiw. On thin gaps between rigid bodies two-way coupled to incompressible flow. *Journal of Computational Physics*, 292:1–29, 2015.

[35] Courant Richard, Isaacson Eugene, and Rees Mina. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, 5(3):243–255, 1952.

[36] Casey L. Richardson, Jan Hegemann, Eftychios Sifakis, Jeffrey Hellrung, and Joseph M. Teran. An XFEM method for modeling geometrically elaborate crack propagation in brittle materials. *International Journal for Numerical Methods in Engineering*, 88(10):1042–1065, 2011.

[37] Avi Robinson-Mosher, Craig Schroeder, and Ronald Fedkiw. A symmetric positive definite formulation for monolithic fluid structure interaction. *Journal of Computational Physics*, 230(4):1547–1566, 2011.

[38] Avi Robinson-Mosher, Tamar Shinar, Jon Grétarsson, Jonathan Su, and Ronald Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph.*, 27(3):46:1–46:9, August 2008.

[39] Eftychios Sifakis, Kevin G. Der, and Ronald Fedkiw. Arbitrary cutting of deformable tetrahedralized objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pages 73–80, Aire-la-Ville, Switzerland, 2007. Eurographics Association.

[40] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM Trans. Graph.*, 32(4):102:1–102:10, July 2013.

[41] Markus Uhlmann and Todor Doychev. Sedimentation of a dilute suspension of rigid spheres at intermediate Galileo numbers: the effect of clustering upon the particle motion. *Journal of Fluid Mechanics*, 752:310–348, 2014.

[42] Decheng Wan and Stefan Turek. Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method. *International Journal for Numerical Methods in Fluids*, 51(5):531–566, 2005.

[43] Heng Xiao and Jin Sun. Algorithms in a robust hybrid CFD-DEM solver for particle-laden flows. *Communications in Computational Physics*, 9(2):297–323, 2011.

[44] Gary D. Yngve, James F. O'Brien, and Jessica K. Hodgins. Animating explosions. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 29–36, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[45] Omar Zarifi and Christopher Batty. A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '17, pages 7:1–7:11, New York, NY, USA, 2017. ACM.

[46] F. Zhang, D.L. Frost, P.A. Thibault, and S.B. Murray. Explosive dispersal of solid particles. *Shock Waves*, 10(6):431–443, Jan 2001.