Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009)
E. Grinspun and J. Hodgins (Editors)

# Accurate Tangential Velocities For Solid Fluid Coupling

Avi Robinson-Mosher† R. Elliot English† Ronald Fedkiw†
Stanford University Stanford University Stanford University
ImageMovers Digital Industrial Light + Magic

**Abstract**

*We propose a novel method for obtaining more accurate tangential velocities for solid fluid coupling. Our method works for both rigid and deformable objects as well as both volumetric objects and thin shells. The fluid can be either one phase such as smoke or two phase such as water with a free surface. The coupling between the solid and the fluid can either be one-way with kinematic solids or fully two-way coupled. The only previous scheme that was general enough to handle both two-way coupling and thin shells required a mass lumping strategy that did not allow for freely flowing tangential velocities. Similar to that previous work, our method prevents leaking of fluid across a thin shell, however unlike that work our method does not couple the tangential velocities in any fashion, allowing for the proper slip independently on each side of the body. Moreover, since it accurately and directly treats the tangential velocity, it does not rely on grid refinement to obtain a reasonable solution. Therefore, it gives a highly improved result on coarse meshes.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling— Physically Based Modeling;

## 1. Introduction

Since solid fluid coupling can model a wide range of visually interesting phenomena, there has been great interest in algorithms to solve this problem in the graphics community. Predominantly, researchers use Lagrangian methods for solids and Eulerian methods for fluids, however there are some notable exceptions. For example, Lagrangian approaches for fluids were discussed in [TPF89, HMT01, MKN*04, MST*04, KAG*05, YHK07, BWHT07, WT08]. Additionally, various authors have used Eulerian methods to treat solids as high viscosity or viscoelastic Eulerian fluids [CMVT02, REN*04, GBO04, LSSF06]. Our focus is on fully two way coupling an Eulerian fluid simulation to that of a Lagrangian solid.

Two important boundary conditions need to be satisfied when coupling the fluid to the solid. First, the normal velocity needs to be continuous unless of course the fluid is separating away from the solid. Second, the net force must be balanced at the interface. Authors have accounted for these conditions in various ways and often separately one way couple each of these two boundary conditions. For example, one could rasterize the solid velocity to the fluid grid and use the solid velocities as a boundary condition for the fluid, meanwhile integrating the pressure force from the fluid on the surface of the solid, see e.g. [YOH00, GHD03, CMT04, GSLF05, LIGF06] for variations on this. Later work has pointed out that the one way coupling of these boundary conditions can lead to both stability and accuracy issues and thus fully implicit stable two way interactions were proposed by various authors [KFCO06, CGFO06, BBB07, RMSG*08].

Whereas the normal velocity and net force should be continuous at the surface between a solid and a fluid, both the mass and tangential velocities should be uncoupled unless the physics dictates otherwise. For example, if the solid is porous then fluid mass can cross the interface into the solid, as in porosity [LAD08]. Similarly, if viscosity is present the tangential velocity of the fluid will match that of the solid (except in hypersonic regimes, where slip can occur). In real flows, this continuity of tangential velocity falls off quickly away from the solid, such that the fluid near the object does tend to flow freely in the tangential direction and only conforms to the solid velocity across what can be a quite thin boundary layer. Unfortunately, it can take a large number of grid cells to resolve this thin boundary layer, meaning that on

† e-mail: {avir|eenglish|fedkiw}@cs.stanford.edu

a coarse grid one instead sees an extremely thick, non physical, visually disturbing layer with a thickness of a number of grid cells. Even if no viscosity is added directly, numerical viscosity will cause this spurious boundary layer to be present. Therefore our aim is to properly treat the tangential velocity near a solid object so that it is free to slip past the object without sticking, under the assumption that the boundary layer is thinner than what can be resolved by the computational mesh, or more importantantly, visually imperceptible.

While some authors have worked to improve the ability of fluid to freely flow near a solid/fluid interface ( [FF01, HBW03, REN*04, RbZF05, BBB07]), none of them has incorporated this into a fully two way coupled solid/fluid simulation framework that is robust enough to handle the full range of rigid and deformable objects that may be either volumetric bodies or thin shells. Our method treats all of the above, and although it is built as an extension of [RMSG*08] we note that these same ideas could be incorporated into the other state-of-the-art methods.

Our main contribution is a method which results in more accurate tangential fluid flow near boundaries by constraining only the normal component of velocity. The key ideas in the method are the addition of normal velocity constraints to the typical projection method for solving the incompressible Navier-Stokes equations, the use of ghost cells to allow tangential flux through the edges of solid boundaries, and the decoupling of the normal and tangential components of fluid velocity in order to add enough degrees of freedom to keep the system from being overconstrained. After discussing our constraints on the normal velocities, Sections 3 and 4 explains our method in the context of one-way coupling to static and moving solids respectively. This illustrates the impact and efficacy of our method as applied to the relatively common case of one-way coupled solids, before describing its generalization and applicability to the more complex, less common but nonetheless important case of full two-way solid fluid coupling.

## 2. Constraining Velocities

In this section we describe how we solve the Navier-Stokes equations for incompressible flow subject to the additional constraint that the velocity field allow flow only in a reduced set of directions.

### 2.1. Navier-Stokes Equations

The Navier-Stokes equations for incompressible flow are

$$\frac{\partial v}{\partial t} + v \cdot \nabla v = -\frac{1}{\rho}\nabla p + \mu \nabla^2 v + f \tag{1}$$

where $v$ is the fluid velocity, $\rho$ is the fluid density, $p$ is pressure, $\mu$ is the kinematic viscosity, and $f$ is the body force per unit density, along with the constraint that the velocity be divergence free,

$$-\nabla^T v = 0 \tag{2}$$

where the divergence is defined as minus the gradient transposed. We do not add any viscosity directly, but there is some effective viscosity due to grid resolution. We treat these equations with a typical projection method, first using explicit semi-Lagrangian advection to advect the velocity at time $t^n$ to an intermediate $v^\star$ velocity (using Equation 3) and then applying a pressure projection to obtain final divergence free velocities (using Equations 2 and 4).

$$\frac{v^\star - v^n}{\Delta t} + v^n \cdot \nabla v^n = f \tag{3}$$

$$\frac{v^{n+1} - v^\star}{\Delta t} + \frac{1}{\rho}\nabla p = 0 \tag{4}$$

### 2.2. Slip Boundary Conditions

No-slip boundary conditions constrain all components of the fluid velocity to match a target velocity. In contrast, slip boundary conditions decompose the fluid velocity vector $\vec{V}$ into the normal and tangential components,

$$\vec{V}_N = (\vec{V} \cdot \vec{N})\vec{N} \tag{5}$$

and

$$\vec{V}_T = \vec{V} - (\vec{V} \cdot \vec{N})\vec{N} = \vec{V} - \vec{V}_N = (I - \vec{N}\vec{N}^T)\vec{V}, \tag{6}$$

and subsequently constrain only the normal component to match a prescribed target.

For the sake of exposition, we assume our goal is to project

**Figure 1:** *A jet of smoke follows the curve of a helix as it rises (60 × 180 × 60 fluid grid). The fluid velocities obey slip constraints at the boundaries of the helix, and the smoke is still free to diffuse.*

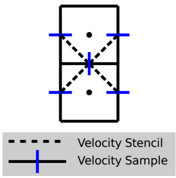


**Figure 2:** *Stencil for interpolating velocity scalars to get a velocity vector at a face.*

out the normal component of the velocity constraining it to be 0. We address the more general case of a moving solid in Section 4 (note that this constraint allows the tangential component of the velocity to flow freely). Thus for each point that needs to be constrained we add an equation of the form

$$\vec{N}^T \vec{V} = 0 \tag{7}$$

and use these to augment Equation 2. For the case of solid fluid coupling Equation 7 will be applied throughout the surface of the solid, but this slip constraint can also be used to perform a simple form of flow control by constraining velocities not to flow in certain directions (for example, see Figure 1; the slip constraints prevent fluid flow in the normal direction to the surface of the helix, which allows swirling motion but constrains the smoke to follow the shape).

We use a uniform Marker and Cell (MAC) grid discretization to represent fluid velocities where instead of storing a velocity vector at each grid cell center, the components of velocity are stored in a staggered fashion on grid cell faces. Since the normal velocity constraint requires a full velocity vector we use multilinear interpolation at each constraint location to form a full velocity vector from scalar velocity samples. We represent this interpolation as a linear relation $\vec{V} = Wv$ where $W$ is the interpolation matrix mapping from a vector of scalar velocity samples $v$ to a full velocity vector at each constraint location (see Figure 2 for illustration of the velocity stencil we use). Therefore we can write the equation corresponding to the $C$th constraint as $N_C^T W v$, where $N_C$ is a column vector of the form $(0, 0, ..., \vec{N}_C^T, ..., 0)^T$. Then we combine the slip constraint equations into a linear system, $N^T W v = 0$, where $N$ is the matrix whose columns are the $N_C$.

We project the velocities produced by the explicit part of the integration scheme, $v^\star$, using the velocity update formula

$$\frac{v^{n+1} - v^\star}{\Delta t} - \frac{1}{V\rho} W^T N \lambda = 0, \tag{8}$$

where $\lambda$ is the force applied along each normal to the fluid and $V$ is the volume of the control volume surrounding the velocity sample. In summary, we would like to constrain the normal component of the velocity field at a set of points in space, and each new equation has a corresponding new unknown $\lambda$ which is the force required to constrain the normal velocity in this fashion. For now this force is fictitious, but when considering solid fluid coupling this is the force the solid applies to the fluid to constrain its normal velocity.

In order to satisfy both the divergence-free constraint in Equation 2 and our new constraints on the normal component of the velocity at the same time we combine the forces on $v^\star$ from Equation 4 and Equation 8 to obtain

$$v^{n+1} = v^\star + \frac{\Delta t}{\rho}(-\nabla p + \frac{1}{V} W^T N \lambda). \tag{9}$$

Then instead of taking the divergence of Equation 4 and setting the divergence of $v^{n+1}$ to 0, we would use Equation 9. That is, as a standard $v^{n+1}$ from Equation 9 should satisfy Equation 2, but in addition we want it to satisfy Equation 7. Substituting Equation 9 into Equation 2 (scaled by $V$) and Equation 7 yields the following coupled system for $p$ and $\lambda$.

$$\Delta t \begin{pmatrix} \nabla^T \frac{V}{\rho} \nabla & -\nabla^T \frac{1}{\rho} W^T N \\ -N^T W \frac{1}{\rho} \nabla & N^T W \frac{1}{V\rho} W^T N \end{pmatrix} \begin{pmatrix} p \\ \lambda \end{pmatrix} = - \begin{pmatrix} V \nabla^T v^\star \\ N^T W v^\star \end{pmatrix} \tag{10}$$

This system is both symmetric and positive definite, so it is amenable to fast solution with the conjugate gradient method. As is typical, the $\Delta t$ can be absorbed into $p$ and $\lambda$, instead solving for $\hat{p}$ and $\hat{\lambda}$.

## 3. One-way Coupling

In this section we focus on the case of coupling the fluid to a stationary solid body. In particular, we demonstrate our discretization by examining the case of a plane splitting the grid as shown in Figure 3a. We discuss our treatment for only one side of the plane noting that the treatment is symmetric on the other side of object.

### 3.1. Slip Constraints at Occluded Grid Faces

We enforce our slip constraints at *occluded* grid faces, by which we mean that a visibility ray cast from the cell center on one side of the face to the opposite cell center intersects object geometry. In Figure 3(a), the faces marked in red are occluded because the green dotted lines intersect the solid. Figure 3(b) shows which cells we wish to include in our discretization. Slip boundary conditions need to be enforced at each face marked by a green segment. In order to apply our slip constraint at a face we need to define a velocity vector. We do so by forming an interpolation stencil $W$ from the nearby visible velocity samples using one-sided interpolation as shown in Figure 3(c) (see [GSLF05] for a thorough review of one-sided discretizations). Whereas the interpolation stencil described in Section 2.2 would use four horizontal velocity samples to define the horizontal component of velocity on the constrained vertical face, marking a face as occluded eliminates two of those samples. $W$ still uses a single scalar sample for the vertical component of velocity, as shown. However we stress that since the grid is split by

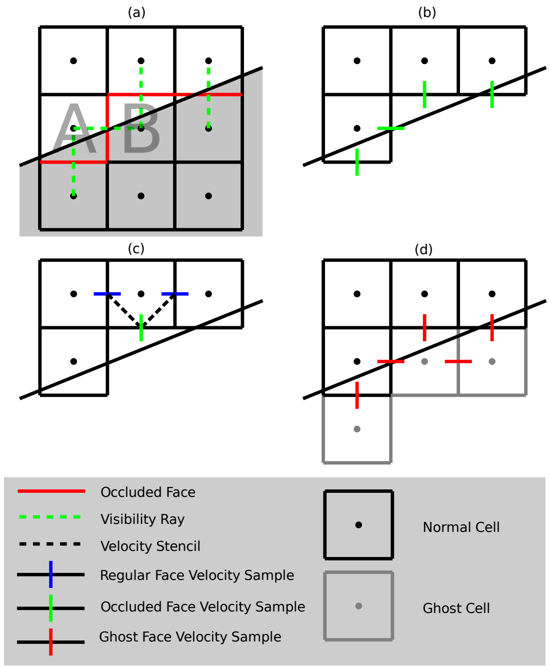


**Figure 3:** *(a) shows the voxelization of the solid to the grid using visibility checks between cell centers. (b) Shows occluded face velocity samples. (c) a one sided velocity interpolation stencil. (d) illustrates placement of ghost velocities and cells.*

a thin plane the treatment described here must also be applied to the grey shaded area in Figure 3(a) and in doing so the vertical velocity at the same geometric location will be a different unknown, since these values are decoupled across the solid. For instance, if fluid is flowing in opposite directions on either side of the separating plane as in Figure 5, the two scalar velocity samples stored at an occluded face must have opposite signs. Once we have a velocity vector $\vec{V}$ at a face, we apply the slip constraint to it by taking the dot product with the solid normal in that dual cell (Equation 7). We define the solid normal in a dual cell as the area weighted average of the normals of all surface simplices within that dual cell.

Here we stress the major departure of our method from the standard discretization. The standard one-sided discretization in Figure 3(a) would discard the five cells whose cell centers are in the grey shaded region, but it would also discard the notion of a degree of freedom at the green marked face velocities in Figure 3(b). This is because those face velocities would be set equal to the normal velocity of this "voxelixed" solid. Unfortunately deleting these degrees of freedom removes the ability to specify tangential velocity on these faces, which is the primary goal of our method. Therefore we retain these degrees of freedom allowing us to apply the constraint equations from Section 2.2 to the vector-valued velocities on these faces. If these face scalar velocities were discarded, then there would be no degree of freedom for the vertical flow shown at the face marked by the green hash in Figure 3(c), and we would only have unknowns for the horizontal component of velocity.

### 3.2. Ghost Cells For Capturing Tangential Flow

Voxelization of the solid onto the fluid grid results in "stair-step" formations on the grid, as shown in Figure 3 (described as "plateaus" by [RbZF05]). Consider the cell "A", where the right and bottom faces are occluded, forming a corner. As fluid flows from left to right, it has to flow up and over the step created by the voxelized solid. Moreover, the incompressibility condition dictates that what flows into "A" from the left must equal what flows out of "A" to the top. Unfortunately, this creates flow oriented at 45 degrees to the horizontal, and combined with our aim of calculating a correct tangential velocity overconstrains the system. We propose removing this constraint by allowing some amount of fluid to flow into and out of the solid regions as a model for the partially filled and unfilled cells which our discretization does not properly represent. In particular the lower triangular portion of cell "A" should actually be solid and not fluid. Also, the upper triangular region of empty space in cell "B" should be occupied by a fluid cell.

We now consider the decomposition of the velocity vector into the components normal and tangential to the solid, $\vec{V}_N$ and $\vec{V}_T$. Since we are dealing with the case of a static solid, $\vec{V}_N$ is constrained to be zero. Therefore, any flux through the face is due to $\vec{V}_T$. In order to represent the flux through occluded faces in the tangential direction, it is necessary to create ghost cells inside the solid, as shown in Figure 3(d). This allows tangential flow to traverse "stair-step" formations on the grid (similar in spirit to [RbZF05]). To represent this model of tangential flow into and out of the ghost cells, we add a separate degree of freedom on the faces marked by red hashes in Figure 3(d). For the sake of exposition we denote these degrees of freedom by $v_{ghost}$.

Using $v_{ghost}$ requires a modification to our divergence operator $-\nabla^T$ for cells next to occluded faces. Consider the grid face highlighted with the green hash in Figure 3(c). The flux across this face would typically be the size of the face multiplied by the scalar velocity sample stored at this location. However, this scalar velocity sample should be the appropriate component of our vector velocity at this location. So when we compute the flux used in Equation 2 at this face, we use the vertical component of $\vec{V}_N$ plus the scalar $v_{ghost}$ (since $v_{ghost}$ here represents the vertical component of $\vec{V}_T$). For the case of a static solid $\vec{V}_N$ will be zero, so the flux is just $v_{ghost}$. We will discuss the case for a dynamic solid in Section 3.

We use the following implementation for defining the ghost

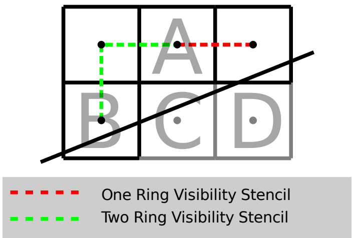


**Figure 4:** *Neighboring ghost cells are connected when their corresponding real cells are visible in each other's one rings. Ghost cells in the same physical location are merged if their real cells are visible in each other's two rings.*

cells, their connectivity, and their new degrees of freedom. Each occluded face falls between one ghost cell and one real cell. Multiple ghost cells can share a geometric location (ghost cells can also share a geometric location with non-ghost cells), so it is necessary to determine whether two occluded faces whose ghost cell would lie in the same geometric location are adjacent to the same or to different ghost cells. This is determined by tracing the shortest path through unoccluded faces from one occluded face to the other. If the occluded faces are in each other's two ring, where a ring is defined as a path through unoccluded cell faces, they are adjacent to the same ghost cell. Otherwise, they are adjacent to different ghost cells. In Figure 4, cell "A" and cell "B" are in each other's two ring and thus their adjacent occluded faces are both adjacent to the same ghost cell at location "C".

Connectivity among ghost cells is modeled by adding scalar velocity degrees of freedom between adjacent ghost cells

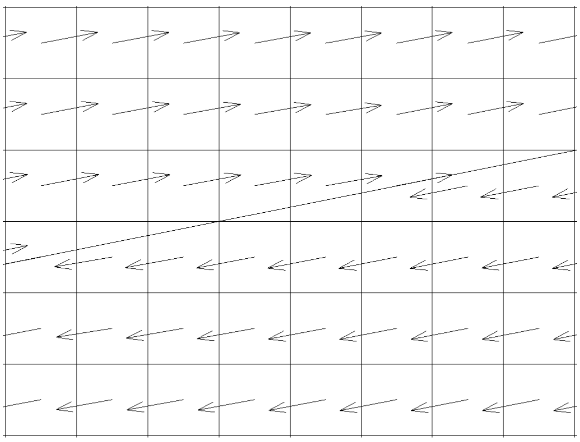

**Figure 5:** *Velocities on either side of the plane are allowed to flow tangentially in opposite directions. The plane is oriented at $\frac{pi}{16}$ radians from the horizontal, and the method works for arbitrary orientations.*

(see for example the lower right red hashed face in Figure 3(d)). Two ghost cells are adjacent if any of their adjacent occluded faces are in each other's one ring. Ghost cells "C" and "D" in Figure 4 are adjacent. In three spatial dimensions we use the same two ring and one ring definitions except that the rings are defined across the true three dimensional faces of the cell (as opposed to across the edges in two dimensions).

In spite of our modeling of flow into and out of the solid, we emphasize that our method conserves mass. This is due to the fact that any flux across a ghost cell flows either into another ghost cell or into a real cell. This means that each connected set of ghost cells has a net incoming flux equal to the net outgoing flux so what flows into the solid must flow out (exactly!). Our ghost cells merely allow bits of fluid to flow into them in one region and out of them in another region slightly downstream in a manner that relaxes the nonphysical step-like constraints on the tangential velocity. In particular consider the thin solid plane in Figure 3. Our ghost cells allow some flow to be transported tangentially along the plane but the flow on top can only be transported to other locations on top of the plane, and the flow on the bottom can likewise only be transported to other locations on the bottom, and no flow is allowed to cross from the top to the bottom or vice versa (see for example Figure 5). Note that most of the flow into and out of the object is actually prevented with our constraint on the velocity in the solid normal direction, but this ghost cell model allows us to alleviate the overconstrained configurations which would actually contradict this constraint.

### 3.3. Velocity Advection And Revalidation

We use semi-Lagrangian advection in order to advect our velocity quantities in Equation 3. In order to update a velocity sample on a grid face, we must interpolate our fluid velocity scalars to the center of the face, then cast a ray back along the velocity vector direction to find our new velocity. This back-cast ray is collided with solids, so that it always remains on the correct side of the object. However, some of our occluded faces may be inside the solid (see e.g. Figure 3(b)), and thus we cannot construct a valid back-cast ray. We set these faces as invalid, and after we update all of the valid faces we use constant extrapolation from close valid velocity samples to set the velocities at invalidated faces. This follows the approach used in [GSLF05].

Velocity interpolation at an arbitrary point in space is performed by using multilinear interpolation in a collidable fashion (see Figure 6). As described above, every velocity sample, including velocities on occluded faces which may fall inside the solid, is associated with at least one of its two neighbor cells (non-occluded faces are associated with both of their neighbor cells). We determine whether a velocity sample which ordinarily should appear in the non-collidable multilinear interpolation stencil (Figure 6, right) may be used by casting a visibility ray from the interpolation point to the associated cell center (Figure 6, left). Note that even if one of the faces associated with the visible cell center is inside the object, it is used in the interpolation stencil for advection. This differs from the usual practice of casting rays to the velocity sample itself. The samples associated with non-visible cell centers are labeled invalid and their values are replaced in the stencil by an average of the values at all of the valid samples. If there are no valid velocity samples, the object velocity (in this case zero) is used. Visibility is based purely on rays cast to cell center, so there can never be cases where some components of the interpolated velocity have valid samples and others do not. Note that since the stencil used for velocity interpolation exactly at the center of a grid face is the same as the $W$ used to get a vector velocity for use in the normal constraint, the solve properly enforces slip on the velocities as they are used for advection, as can be seen by the passively advected particles in Figure 7.

## 4. Moving Solid

Enhancing our description to include the effects of a moving solid requires only a simple addition to the description for a static solid. Rather than setting the normal component of the fluid velocity equal to zero, it should be set equal to the normal component of the solid velocity. We obtain a single solid velocity vector at the occluded face by taking a weighted average of the nearby velocity degrees of freedom on the solid surface as in [RMSG*08]. We call this mapping operator $J$ (see Appendix A). Then the interpolated solid velocity vector at the occluded face is equal to $J\vec{V}_S$, where $\vec{V}_S$ is the vector of solid velocity degrees of freedom. Equation 7 becomes

$$\vec{N}^T\vec{V} = \vec{N}^T J\vec{V}_S, \qquad (11)$$

which constrains the normal component of the fluid velocity to match the normal component of the solid velocity. Plugging Equation 9 into Equation 11 instead of Equation 7

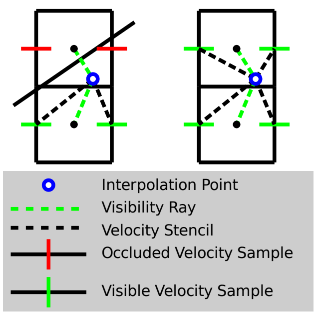


**Figure 6:** *Velocities are interpolated using one sided stencils depending upon visibility to cell centers.*

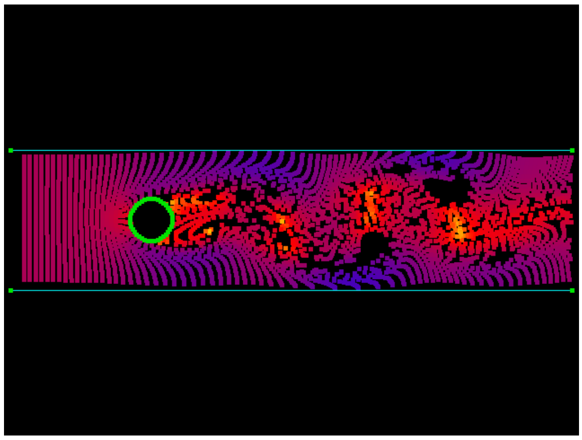

**Figure 7:** *Passively advected particles illustrate the flow past a cylinder with fast particles shown in blue and slow particles shown in red (50 × 200 fluid grid). Vortex shedding is visible behind the cylinder.*

yields the following coupled system

$$\begin{pmatrix} \nabla^T \frac{V}{\rho}\nabla & -\nabla^T \frac{1}{\rho} W^T N \\ -N^T W \frac{1}{\rho}\nabla & N^T W \frac{1}{V\rho} W^T N \end{pmatrix} \begin{pmatrix}\hat{p} \\ \hat{\lambda}\end{pmatrix} = \begin{pmatrix} -V\nabla^T v^\star \\ -N^T W v^\star + N^T J\vec{V}_S\end{pmatrix} \qquad (12)$$

where the only change is the addition of $N^T J\vec{V}_S$ on the right hand side. This system remains symmetric positive definite, and thus can be solved with the conjugate gradient method.

As mentioned in Section 3, the divergence operator $-\nabla^T$ must be modified in cells next to occluded faces. The normal component of velocity at occluded faces can now be

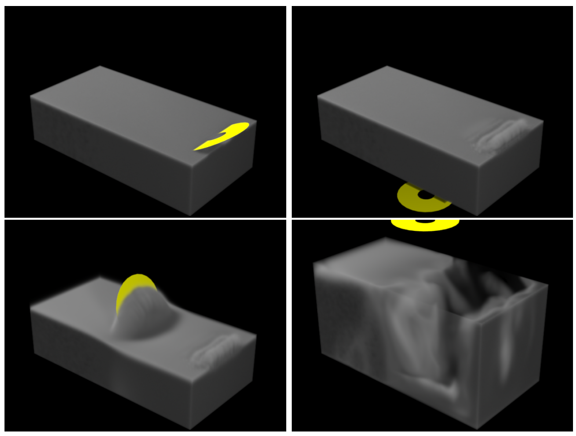

**Figure 8:** *A thin rigid disk passes through a smoke volume (120 × 60 × 60 fluid grid). Clockwise from the top left: (a) as the disk is entering the volume, (b) after passing through edge on, (c) passing through the volume while spinning, (d) after exiting the volume.*

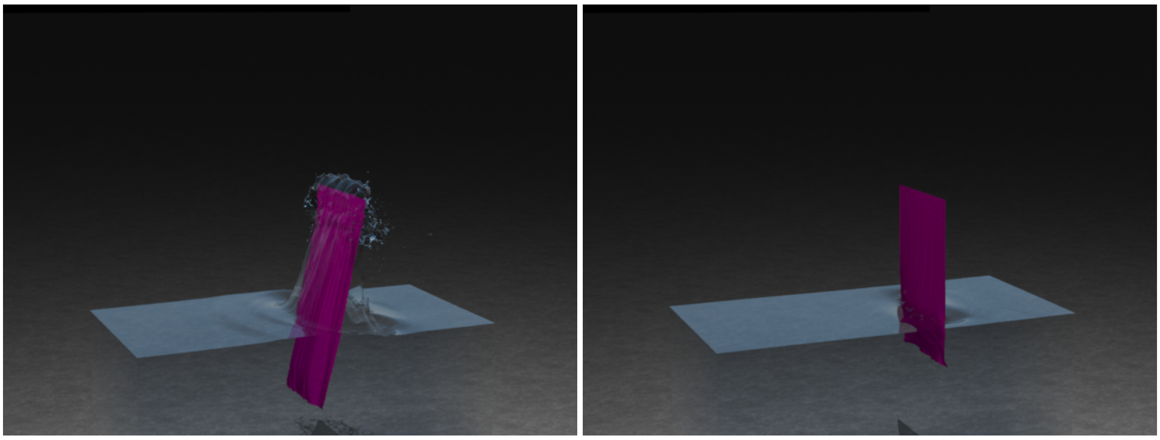

**Figure 9:** *A sheet of cloth pulled out of a pool of water with no-slip boundary conditions (left) and slip boundary conditions (right).*

nonzero, so it must be accounted for in the divergence. In a real fluid cell the flux across an occluded face becomes the axial component of the normal velocity plus $v_{ghost}$, which is written as $\vec{P}^T\vec{V}_N + v_{ghost}$, where $\vec{P}$ is the unit vector across the face. The modified divergence $-\hat{\nabla}^T$ is written $-\nabla^T - \nabla^T PNN^T$, where $P$ is $\vec{P}$ for every occluded face and $\vec{0}$ elsewhere. Equation 2 then becomes

$$-\hat{\nabla}^T v = 0 \tag{13}$$

and Equation 9 becomes

$$v^{n+1} = v^{\star} + \frac{\Delta t}{\rho}(-\hat{\nabla} p + \frac{1}{V} W^T N\lambda). \tag{14}$$

The expression for divergence in the ghost cell across the occluded face remains the same, using only $v_{ghost}$, because the ghost cells account for only the tangential part of the flow. We replace all occurrences of $\nabla$ with $\hat{\nabla}$ in our coupled system.

## 5. Two-way Coupling

Extending the slip constraints to two-way coupling requires moving the solid contribution to the slip constraint into the left-hand side of the equation (and thus into the system to be solved) and adding the solid momentum equations to the system. We follow the treatment of [RMSG*08] in this and in their coupled Newmark integration scheme, but note that our method could equally be used with other schemes for fully implicit coupling. Conservation of momentum on the solid nodes is written as

$$M_S\vec{V}_S^{n+1} = M_S\vec{V}_S^{\star} + \Delta t D\vec{V}_S^{n+1} - \Delta t J^T N\lambda \tag{15}$$

where $M_S$ is the solid mass matrix (which is block diagonal and symmetric), $\vec{V}_S$ is the solid velocity, $\vec{V}_S^{\star}$ includes all of the explicitly integrated solid forces, $J^T N\lambda$ is the equal and opposite force on the solid that $\lambda$ applies to the fluid, and $D$ is the coefficient matrix for the implicitly integrated solid forces (such as damping). We require only that $D$ be symmetric, as in, for instance, the semi-implicit approach of [BMF03] or the fully implicit approach of [BW98], where it is a linearization of the elastic and damping forces.

Note in particular that since $J$ is an interpolation operator, $J^T$ will conservatively redistribute this force to the nodes of the solid. Tangential flow through ghost cells as described in Section 3 does not couple directly to the object. The pressures in these ghost cells are not felt by the object, because they affect fluid motion only in the (unconstrained) tangential direction. All interaction between the fluid and the solid happens in the normal direction, and is captured by the $\lambda$ forces corresponding to the slip constraints.

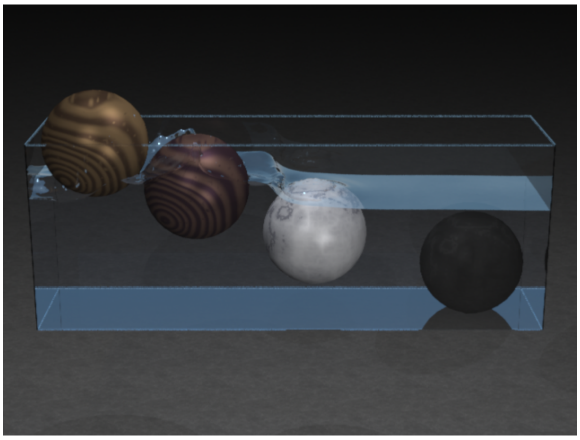

**Figure 10:** *Objects of varying density demonstrate correct buoyancy (200 × 100 × 50 fluid grid).*

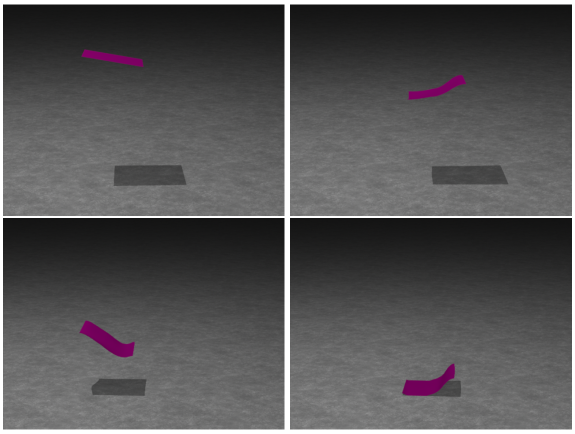

**Figure 11:** *A piece of cloth falling in air (100 × 50 × 50 fluid grid).*

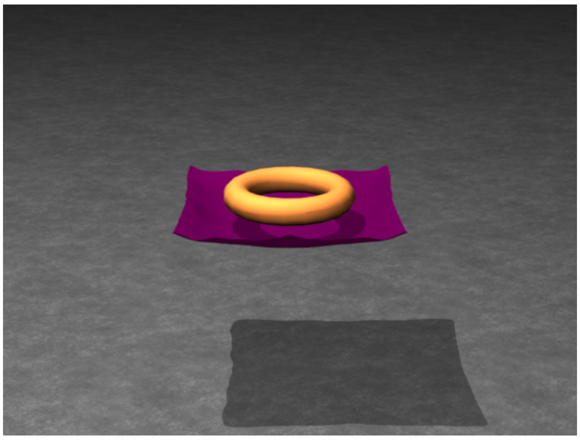

**Figure 12:** *A rigid torus collides with a light piece of cloth falling in air (20 × 20 × 20 fluid grid).*

The resulting fully coupled system is

$$\begin{pmatrix} \hat{\nabla}^T \frac{V}{\rho} \hat{\nabla} & -\hat{\nabla}^T \frac{1}{\rho} W^T N & 0 \\ -N^T W \frac{1}{\rho} \hat{\nabla} & N^T W \frac{1}{V\rho} W^T N & -N^T J \\ 0 & -J^T N & \Delta t D - M_S \end{pmatrix} \begin{pmatrix} \hat{p} \\ \hat{\lambda} \\ \vec{V}_S^{n+1} \end{pmatrix} = \begin{pmatrix} -V \hat{\nabla}^T v^\star \\ -N^T W v^\star \\ -M_S \vec{V}_S^\star \end{pmatrix}. \tag{16}$$

## 6. Results

### 6.1. Solving the linear system

Our two-way coupled system results in an indefinite matrix, due to the positive definiteness of the top left 2 × 2 block and the negative definiteness of the bottom right 1 × 1 block. We use symmetric QMR (SYMMQMR, [FN94]) in order to solve the system. We chose SYMMQMR over MINRES ( [Cho06]) because it allows us to use an indefinite preconditioner. We perform an incomplete Cholesky factorization of the top left block of the matrix and use the negative identity matrix for the bottom right block. We ran most of our examples on a single processor, though we ran some including the buoyancy test on a single eight-processor machine to demonstrate feasibility. While parallelizing the linear solve is somewhat more complicated than that of the usual fluid projection due to the ghost cells and $\lambda$ forces, it remains amenable to efficient parallelization. In all cases, the slip examples ran comparably to or faster than the no-slip versions.

### 6.2. Examples

Slip boundary conditions allow us to avoid the non-physically thick boundary layer which sticks to the object when no-slip boundary conditions are used. In Figure 9, a sheet of cloth is pulled out of a pool of water. In the no-slip case the elements of the cloth are badly stretched by the water clinging to it, and it also drags a too-thick sheet of fluid out of the pool along with it. In the slip case, the cloth slides easily out of the pool, exhibiting nice slipping through the water. The no-slip sticking effect does not go away without computationally inefficient grid refinement. The use of slip boundary conditions allows visually pleasing results with much lower grid resolutions (Figure 12 is an example of this). We note that our method of adding additional constraints to the usual divergence constraints for the fluid can also be used to achieve no-slip coupling without the need for lumping fluid mass onto the solid by simply enforcing a no-slip constraint at any dual cell which contains solid.

Figure 11 shows a sheet of paper or light cloth slipping and tumbling in the air. No-slip boundary conditions damp out the slipping motion of the sheet, resulting in nonphysical and uninteresting behavior; however, slip boundary conditions result in dynamic and physically plausible motion. Experimenting with parameters, we observed both stiff sheets properly gliding through the air and soft sheets bending and rippling.

To demonstrate that our implementation gives the correct two way coupled dynamics, we show our method results in proper buoyancy (Figure 10). A one-way coupled disk slides through smoke in Figure 8 without disturbing it, then sweeps through it throwing up a wake. We demonstrate multiple bodies interacting in Figure 12, where a rigid torus falls onto a piece of light cloth and drags it down.

## 7. Discussion/Future Work

All of our connectivity and decisions as to whether two ghost cells are the same are based on first-order visibility information gathered from ray casts between neighboring cell centers. We chose this over a more complicated method for determining connectivity because it is simple to implement and its failure cases are easy to analyze. Consider the case where cloth folds over itself and traps fluid. If there is at least one cell center inside, incompressibility will be enforced as normal. If there are no cell centers inside, it might collapse, as there could be fluid pressures on the outside and not the inside. However, if the solid is between two cell centers, we do not discriminate between layers of cloth - both sides feel the same forces, so the layers will not in fact get crushed together as they might if a vacuum were present.

The reason that the [BBB07] method handles flow through subgrid regions is the same reason it cannot handle non-volumetric (or even thin volumetric) objects - it effectively shrinks the objects by one grid cell and solves for fluid pressures inside the revealed layer. Our method could be extended with a more complex determination of ghost cells and allow for this behavior - however, we note that there is no easy way to differentiate between a solid object and two objects pressed near to each other. We chose to resolve this by letting the fluid grid resolution determine whether or not objects are separated enough to allow fluid flow. A possible extension of our work would be to take the ghost cells more seriously as fluid cells and use a more complex method for determining where they should be created. However, unless care is taken this can result in an exponentially large number of ghost cells, which can never occur in our method as it stands. If we want the ability to resolve fine features of this sort (e.g. tiny pockets and thin channels) we will be paying for it one way or another - whether in arbitrary numbers of ghost cells or higher grid resolutions.

We have presented a method for obtaining more accurate tangential velocities in a fluid system which may be one or two-way coupled to the solid. Since this results in good quality visual fidelity even at low fluid resolutions, this is a promising direction for investigating the simulation of real time or near-real time fluids. On the flip side, solving the indefinite system - which occurs only in the case of two way coupling - is still slower than solving a positive definite system and work on accelerating this solve would be very useful. While we currently implement the method on a Cartesian grid, it is straightforward to extend to octrees and probably does not require that much more work to extend the concepts to tetrahedral meshes or in general any method which uses control volumes around its fluid velocity degrees of freedom. Finally, it would be interesting to investigate boundary conditions which fall between the extremes of slip and no-slip.

## 8. Acknowledgments

Research supported in part by a Packard Foundation Fellowship, an Okawa Foundation Research Grant, ONR N0014-06-1-0393, ONR N00014-06-1-0505, ONR N00014-05-1-0479 for a computing cluster, NIH U54-GM072970, NSF ACI-0323866, and NSF IIS-0326388. A.R. was supported in part by a Stanford Graduate Fellowship. We would like to thank Bill Dally and Christos Kozyrakis as well as Charbel Farhat for computing resources and Jacob Leverich for helping us use those resources.

## Appendix A: Constructing the mapping operator $J$

The mapping operator $J$ is used to obtain a single solid velocity vector at an occluded face by averaging the solid velocity throughout the dual cell. For each solid surface triangle intersecting the dual cell we find the clipped polygon which lies entirely within the dual cell. Each polygon is then triangulated. For each of the resulting sub-triangles we compute the barycentric coordinates of its centroid in the space of the original triangle. The velocity of each node of the original triangle is then weighted by the corresponding barycentric coordinate scaled by the area of the sub-triangle. These weights are then used to compute a weighted average of the velocities of the solid nodes belonging to triangles intersecting the dual cell. As a result, nodes which are closer to the area of the triangle which falls within the dual cell are weighted more heavily than nodes which are further away.

## References

[BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. (SIGGRAPH Proc.) 26*, 3 (2007), 100.

[BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2003), pp. 28–36.

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *ACM SIGGRAPH 98* (1998), ACM Press/ACM SIGGRAPH, pp. 43–54.

[BWHT07] BARGTEIL A. W., WOJTAN C., HODGINS J. K., TURK G.: A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. (SIGGRAPH Proc.) 26*, 3 (2007).

[CGFO06] CHENTANEZ N., GOKTEKIN T. G., FELDMAN B., O'BRIEN J.: Simultaneous coupling of fluids and deformable bodies. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), pp. 325–333.

[Cho06] CHOI S.-C.: *Iterative Methods for Singular Linear Equations and Least-Squares Problems*. PhD thesis, Stanford University, 2006.

[CMT04] CARLSON M., MUCHA P. J., TURK G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. Graph. (SIGGRAPH Proc.) 23* (2004), 377–384.

[CMVT02] CARLSON M., MUCHA P., VAN HORN R., TURK G.: Melting and flowing. In *ACM Trans. Graph. (SIGGRAPH Proc.)* (2002), vol. 21, pp. 167–174.

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *Proc. of ACM SIGGRAPH 2001* (2001), pp. 23–30.

[FN94] FREUND F., NACHTIGAL N.: A new krylov-subspace method for symmetric indefinite linear systems. In *14th IMACS World Congress on Computational and Applied Mathematics (W. F. Ames, ed.)* (1994).

[GBO04] GOKTEKIN T. G., BARGTEIL A. W., O'BRIEN J. F.: A method for animating viscoelastic fluids. *ACM Trans. Graph. (SIGGRAPH Proc.) 23* (2004), 463–468.

[GHD03] GÉNEVAUX O., HABIBI A., DISCHLER J.-M.: Simulating fluid-solid interaction. In *Graph. Interface* (June 2003), pp. 31–38.

[GSLF05] GUENDELMAN E., SELLE A., LOSASSO F., FEDKIW R.: Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph. (SIGGRAPH Proc.) 24*, 3 (2005), 973–981.

[HBW03] HOUSTON B., BOND C., WIEBE M.: A unified approach for modeling complex occlusions in fluid simulations. In *SIGGRAPH 2003 Sketches & Applications* (2003), ACM Press.

[HMT01] HADAP S., MAGNENAT-THALMANN N.: Modeling dynamic hair as a continuum. *Comput. Graph. Forum 20*, 3 (2001).

[KAG*05] KEISER R., ADAMS B., GASSER D., BAZZI P., DUTRÉ P., GROSS M.: A unified Lagrangian approach to solid-fluid animation. In *Eurographics Symp. on Point-Based Graph.* (2005).

[KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. In *ACM Trans. Graph. (SIGGRAPH Proc.)* (Aug. 2006), vol. 25, pp. 820–825.

[LAD08] LENAERTS T., ADAMS B., DUTRÉ P.: Porous flow in particle-based fluid simulations. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–8.

[LIGF06] LOSASSO F., IRVING G., GUENDELMAN E., FEDKIW R.: Melting and burning solids into liquids and gases. *IEEE Trans. on Vis. and Comput. Graph. 12*, 3 (2006), 343–352.

[LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple interacting liquids. *ACM Trans. Graph. (SIGGRAPH Proc.) 25*, 3 (2006), 812–819.

[MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2004), pp. 141–151.

[MST*04] MÜLLER M., SCHIRM S., TESCHNER M., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids. *J. Comput. Anim. and Virt. Worlds 15*, 3–4 (July 2004), 159–171.

[RbZF05] ROBLE D., BIN ZAFAR N., FALT H.: Cartesian grid fluid simulation with irregular boundary voxels. In *SIGGRAPH 2005 Sketches & Applications* (2005), ACM Press.

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2004), pp. 193–202.

[RMSG*08] ROBINSON-MOSHER A., SHINAR T., GRETARSSON J., SU J., FEDKIW R.: Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 46:1–46:9.

[TPF89] TERZOPOULOS D., PLATT J., FLEISCHER K.: Heating and melting deformable models (from goop to glop). In *Graph. Interface* (1989), pp. 219–226.

[WT08] WOJTAN C., TURK G.: Fast viscoelastic behavior with thin features. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–8.

[YHK07] YUKSEL C., HOUSE D. H., KEYSER J.: Wave particles. *ACM Trans. Graph. (SIGGRAPH Proc.) 26*, 3 (2007), 99.

[YOH00] YNGVE G., O'BRIEN J., HODGINS J.: Animating explosions. In *Proc. of ACM SIGGRAPH 2000* (2000), pp. 29–36.