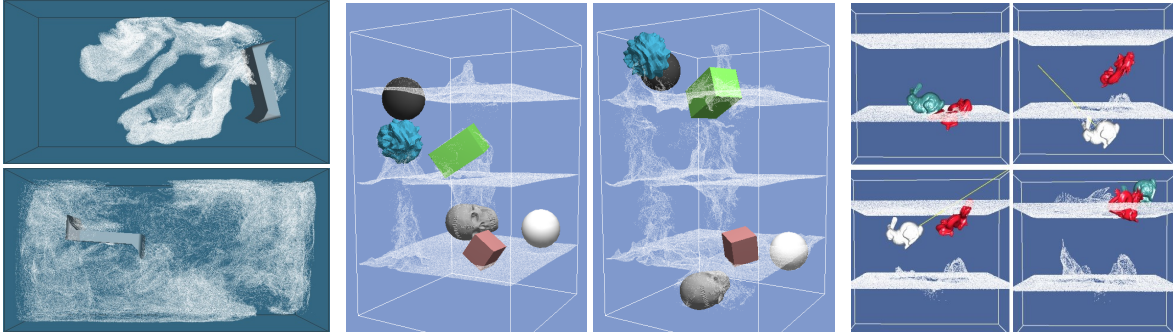# A Fast Variational Framework for Accurate Solid-Fluid Coupling

Christopher Batty   Florence Bertails   Robert Bridson

University of British Columbia[*]

Figure 1: *Left: A solid stirring smoke runs at interactive rates, two orders of magnitude faster than previously. Middle: Fully coupled rigid bodies of widely varying density, with flow visualized by marker particles. Right: Interactive manipulation of immersed rigid bodies.*

Abstract

Physical simulation has emerged as a compelling animation technique, yet current approaches to coupling simulations of fluids and solids with irregular boundary geometry are inefficient or cannot handle some relevant scenarios robustly. We propose a new variational approach which allows robust and accurate solution on relatively coarse Cartesian grids, allowing possibly orders of magnitude faster simulation. By rephrasing the classical pressure projection step as a kinetic energy minimization, broadly similar to modern approaches to rigid body contact, we permit a robust coupling between fluid and arbitrary solid simulations that always gives a well-posed symmetric positive semi-definite linear system. We provide several examples of efficient fluid-solid interaction and rigid body coupling with sub-grid cell flow. In addition, we extend the framework with a new boundary condition for free-surface flow, allowing fluid to separate naturally from solids.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: fluid simulation, physically-based animation, fluid-solid coupling

1 Introduction

Physical simulation is an increasingly popular approach for producing animations of fluid. It holds out the promise of automatically generating detailed and physically-plausible motion for phenomena such as smoke, water and explosions, and their complex interactions with dynamic solid objects. However, current techniques encounter problems with both efficiency and robustness. In this paper, we propose a new variational framework which allows robust and accurate solid-fluid coupling on relatively coarse Cartesian grids, providing potentially orders of magnitude faster simulation.

For the purposes of this paper, we focus on splitting/projection algorithms using fluid velocity and pressure as primary variables to solve the incompressible Euler equations:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} + \frac{1}{\rho}\boldsymbol{\nabla}p = \mathbf{f} \\ \boldsymbol{\nabla} \cdot \mathbf{u} = 0 \end{cases}$$

We use the standard notation ($\mathbf{u}$ for fluid velocity, p for pressure, ρ for density, $\mathbf{f}$ for acceleration due to body forces such as gravity, etc.); Bridson et al.'s course notes [2006] provide further background. The essential steps in such an algorithm are advection, corresponding to updating the positions of fluid elements (analogous to updating positions in a solid simulation), and projection, corresponding to solving for pressure to make the velocity field incompressible (analogous to updating velocities from forces in a solid simulation). We use the near-zero-dissipation FLIP method of Zhu and Bridson [2005] for advection but do not use vorticity confinement [Fedkiw et al. 2001] in our examples. The contributions of this paper are concerned with the pressure projection step.

We do note there are several compelling alternatives to this class of methods, such as vorticity approaches (e.g. [Angelidis et al. 2006]), Smoothed Particle Hydrodynamics (e.g. [Keiser et al. 2005]) and the Lattice Boltzmann Method (e.g. [Thürey et al. 2006]), which are beyond the scope of this paper.

1.1 Previous Work

Solid Wall Boundary Conditions

The dominant approach has been to discretize the fluid equations on a regular Cartesian grid, using the staggered "MAC-grid" arrangement of pressure and velocity unknowns [Harlow and Welch 1965]. Foster and Metaxas [1996] pioneered its use in computer graphics, implementing solid boundary conditions by voxelizing

---

[*]e-mail: {batty,bertails,rbridson}@cs.ubc.ca
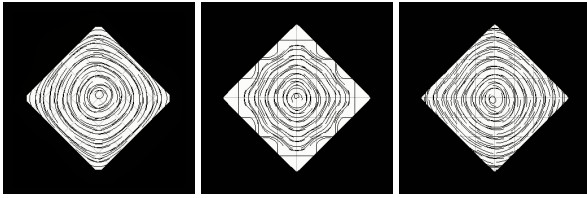
Figure 2: *Streamlines of a vortex-in-a-box. From left to right: grid-aligned geometry ground truth, voxelized pressure solve, variational pressure solve. Note the stair-step artifacts in the voxelized solve, eliminated with the variational method.*

obstacles onto the grid. This method, which we shall refer to as the "voxelized pressure solve", works very well if all object boundaries happen to be aligned with the grid, but otherwise introduces significant stair-step artifacts which unfortunately do not converge to zero as grid resolution is increased. In figure 2, observe that the streamlines of the flow match the voxelized version, not the original geometry; for comparison we show that our method matches the correct streamlines regardless of orientation with respect to the grid. These artifacts are particularly objectionable in water simulations, where water will pool on the steps rather than freely flowing down a slope. However, even in smoke simulations they are noticeable as undue numerical viscosity: all components of velocity (including those tangential to the true surface) are driven to zero. The best that can be done is to excessively increase grid resolution until the affected grid cells are small enough to be visually negligible.

Foster and Fedkiw [2001] attempted to mitigate these problems by using an accurate normal to enforce the solid wall velocity boundary condition (allowing the fluid to slip past tangentially), and then constraining the voxelized pressure solve not to touch these velocities. Houston et al. [2003] later introduced the constrained velocity extrapolation method to further improve on this, extrapolating the tangential component of fluid velocity into solids accurately and thus eliminating grid artifacts from fluid advection. Rasmussen et al. [2004] further elaborated the approach for level set advection. Unfortunately, all these methods suffer from artifacts due to the voxelized pressure solve, which is highly visible on coarse grids, necessitating high resolutions. Furthermore, though the approach works well for highly dynamic splashing scenarios, it fails in the simple hydrostatic case of fluid sitting still in a container: unphysical currents develop and unstably blow up, since the pressure cannot cancel the tangential component of force due to gravity on oblique boundaries.

Here we take an aside to discuss grid resolution. To be able to simulate a flow with features such as vortices as small as some length h, grid cells must be no larger than h. Since convergence to a quantitatively accurate solution is generally irrelevant to animation, it would be ideal to have grid spacing in fact equal to h, much coarser than typically used in scientific simulations. Making the most of coarse grids is particularly important since memory scales as $O(n^3)$, for an $n \times n \times n$ grid, and the time required for simulation can scale as badly as $O(n^5)$ if the typical time step restriction $\Delta t \sim 1/n$ is taken and the typical Incomplete Cholesky Level 0 Preconditioned Conjugate Gradient algorithm is used for solving linear systems. Increasing grid resolution just to account for the failure of an algorithm to faithfully reflect the physics at the appropriate resolution is clearly a very steep price to pay.

Octree methods [Losasso et al. 2004] allow one to use high resolution only where needed, partially overcoming the efficiency problem of the voxelized approach. However, the node-averaging used for octrees gives results inferior to the MAC grid at the same resolution [Irving et al. 2006] requiring even finer resolution (and more

stringent restriction on time step); the considerable computational overhead of the pointer structure compared to a regular grid reduces the potential benefit further.

Another promising approach is introduced by Feldman et al. [2005] where unstructured tetrahedral meshes, which can align with arbitrary geometry, are used in lieu of grids. These have the side benefit of easy adaptivity, and recent advances in mesh generation mean only a fraction of the simulation time is spent on making meshes even if rebuilt nearly from scratch each time step [Klingner et al. 2006]. However, matching fine-scale geometry or porous objects requires an impractically large mesh, and the extensive averaging used in interpolation introduces more numerical dissipation, demanding higher resolution meshes than comparable grid simulations. Furthermore, the overhead of unstructured meshes compared to regular grids of similar resolution is considerable.

Roble et al. [2005] take an intermediate approach, using an underlying regular grid, but modifying only boundary cells to align with objects. A growing body of similar work exists within the computational fluid dynamics community [Johansen and Colella 1998; Udaykumar et al. 2001; Kirkpatrick et al. 2003; Marella et al. 2005; Liu et al. 2005; Schwartz et al. 2006]. While both finite volume and finite difference techniques are represented in this sampling, a common implementation difficulty is the complexity of robustly handling the fully three-dimensional geometry and/or stencils required to capture the non-grid-aligned Neumann boundary condition.

Solid-Fluid Coupling

Takahashi et al. [2002] presented a simple method for coupling fluids and rigid bodies, with the same problematic voxelization as before, where the rigid bodies provide velocity boundary conditions to the fluid and the resulting pressure subsequently provides a net force and torque on the rigid bodies. We note that in certain situations, such as a rigid stopper resting on fluid in a closed tube, the alternating nature of the coupling results in failure: the fluid may be constrained to compress by rigid body velocities, giving an inconsistent linear system for pressure.

Génevaux et al. [2003] introduced coupling between a free surface fluid simulation using marker particles, and elastic solid simulation using masses and springs. The coupling is achieved by attaching the solid with *ad hoc* damped springs to nearby fluid marker particles (averaging the force down onto the grid for the fluid simulator), then using the voxelized pressure solve.

Carlson et al. [2004] simulated coupled fluid and rigid bodies with Distributed Lagrange Multipliers, conceptually considering rigid bodies as fluid on a grid, solving for pressure, then projecting velocity in those regions back to rigid motion with careful additions to the body force to account for density differences between solids and fluids. However, the method cannot stably handle light solids (less than ~ 0.45 the fluid density), and fails in some cases, allowing fluid to erroneously leak through a rigid plug supporting it.

Guendelman et al. [2005] returned to the alternating voxelized approach of Takahashi et al., generalized to include octree grids, thin solids and arbitrary solid dynamics. To improve upon the noisy pressure resulting from voxelization to cell faces, a second pressure solve, doubling the expense of the calculation, is done with solid masses added to the fluid grid density in the style of the Immersed Boundary Method [Peskin 2002]; this smoother pressure is used to calculate force on solids. Inconsistent linear systems arising in enclosed fluid regions are handled by projecting out the null-space components from the right-hand-side of the pressure equation; this has the unfortunate effect of allowing closed regions to change volume as the solid dictates, meaning fluid-filled balloons, for example, cannot be simulated.

Full simultaneous coupling between fluids and solids was achieved by Klingner et al. for rigid bodies [2006] and by Chentanez et al. for deformable objects [2006], with an approach that produces similar discretizations to ours. While focusing mostly on tetrahedral meshes that align with solid boundaries Chentanez et al. do note that this approach can be applied on regular grids, though the accompanying animation of fluid pouring on a bunny model shows apparent grid artifacts.

Of course, many scientific works in computational fluid dynamics address fluid-solid coupling. We highlight Peskin's Immersed Boundary Method [2002], which averages solid properties onto the fluid grid (and thus cannot stop fluid leaking through solids, for example); the ALE approach of Hirt et al. [1974], which requires tetrahedral meshes fully resolving all solid boundaries; and Le et al.'s use of the Immersed Interface Method to couple fluid with rigid and elastic bodies [2006], whose expenses each time step include a solve with an unsymmetric matrix and a Singular Value Decomposition of a matrix with size proportional to the number of points used to discretize solids.

1.2 Contributions

We introduce a new variational interpretation of the pressure equation for coupled fluids in section 2, namely that pressure minimizes the kinetic energy of the system. The pressure update and total kinetic energy may be easily discretized on a regular grid with arbitrary immersed solid geometry. Then the discrete pressure which minimizes this discrete kinetic energy is found; this reduces automatically to a well-posed, sparse, symmetric positive semi-definite linear system. The solution is free of grid artifacts, permitting fast solution on coarse grids. It enforces simultaneous coupling correctly, handling all problem cases mentioned in the previous section, and can be applied with arbitrary solid dynamics. In addition, we automatically account for sub-grid-resolution solid geometry in our discrete estimate of kinetic energy; this gives us approximate sub-grid accuracy in coupling, allowing robust handling of fluid flowing through thin gaps that could not be efficiently resolved with octrees or tetrahedral meshes. We work out the details of our solver for rigid bodies in section 3.

Finally, the variational framework highlights an analogy between fluid pressure and modern treatment of inelastic contact forces between rigid bodies: they both are based on kinetic energy minimization. Inspired by this connection, we introduce a new free surface/solid boundary condition in section 4, expressed as an inequality constraint on our minimization. This allows fluid to freely separate from solid walls, similar to rigid bodies separating from contact, fixing some enduring artifacts seen in previous fluid simulation work where fluid unnaturally crawls along walls and ceilings.

2 A Variational Interpretation of Pressure

Consider a fluid and immersed solids. In continuous space variables the pressure update for fluid velocity from time n to $n+1$ is

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \frac{\Delta t}{\rho}\boldsymbol{\nabla}p \qquad (1)$$

where $\tilde{\mathbf{u}}$ is the intermediate velocity field resulting from advection and integration of body forces such as gravity. The accompanying update to solid velocities is

$$\mathbf{V}^{n+1} = \mathbf{V}^n + \Delta t \mathbf{M}_S^{-1}\mathbf{J}p \qquad (2)$$

where $\mathbf{V}$ is the generalized velocity of the solid (possibly a continuous field for a deformable object, or a six-dimensional vector for a single rigid body, etc.), $\mathbf{M}_S$ is the mass linear operator (convolution

with solid density for a deformable object, the usual 6×6 matrix containing the inertia tensor and the total mass for each rigid body, etc.), and $\mathbf{J}$ is a linear operator converting pressure on the boundary of the solid to generalized forces.

The pressure enforces the incompressibility condition $\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} = 0$ inside the fluid, and the boundary condition $\mathbf{u}^{n+1} \cdot \hat{\mathbf{n}} = \mathbf{v}^{n+1} \cdot \hat{\mathbf{n}}$ on the solid boundary, with $p = 0$ on the free surface. Here $\mathbf{v}$ is the velocity of the solid evaluated at a point on the boundary, which must be $\mathbf{v} = \mathbf{J}^*\mathbf{V}$ from basic physical principles, with $\mathbf{J}^*$ the adjoint or transpose of $\mathbf{J}$. The solid boundary condition is simply stating the fluid may flow neither in nor out of a solid. Substituting the update equations in these conditions gives the PDE form of the coupled pressure equations.

The total kinetic energy of the system is

$$\text{KE} = \iiint_{\text{fluid}} \frac{1}{2}\rho||\mathbf{u}||^2 + \frac{1}{2}\mathbf{V}^*\mathbf{M}_S\mathbf{V} \qquad (3)$$

where the solid term may be a double convolution integral for continua or just a finite quadratic form for rigid bodies. It is a straightforward exercise in variational calculus to show that the pressure PDE is in fact the Euler-Lagrange equation for minimizing kinetic energy with respect to pressure (see Bridson et al.'s course notes [2006] for a proof of the simpler case with motionless solids). Note that the kinetic energy is bounded below by zero and depends only quadratically on pressure, so this minimization is well-posed. Put another way, the pressure solve is computing a projection of velocities onto the space of divergence-free fluid velocities and compatible solid velocities. Projection is equivalent to finding the closest point on that space, and the metric defining "closest" is kinetic energy. This statement of the pressure problem is also in fact exactly the Lagrange Multiplier approach to constrained mechanics (e.g. [Baraff 1996]), with pressure playing the role of Lagrange Multiplier.

Scripted or stationary solids can be incorporated by taking the limit as their mass goes to infinity, and instead of total kinetic energy using the difference in energy of the open system (excluding infinite masses) from one time step to the next. The coupling with the scripted objects reduces to calculating the work done by pressure on their boundaries, i.e. the exchange of energy between the open system and the scripted objects. Once discretized and reduced to a linear system, it is in fact easier to take the limit as mass goes to infinity, i.e. as $\mathbf{M}_S^{-1}$ goes to zero, which has the effect of just eliminating terms from the matrix.

2.1 Fluid Discretization

Instead of discretizing the local pressure PDE with boundary conditions, we discretize the global variational principle. This avoids directly discretizing the tricky velocity boundary condition at non-grid-aligned solid boundaries, instead relying on the easier task of estimating the kinetic energy. Moreover, it reduces to the standard PDE discretizations for grid-aligned geometry, and leads to simulation code largely the same or simpler than previous techniques.

We discretize the fluid variables on a standard MAC grid, and use the regular finite difference approximation to the gradient for the pressure update. For example, the $x-$ component update is

$$u_{i+1/2,j,k}^{n+1} = \tilde{u}_{i+1/2,j,k} - \frac{\Delta t}{\rho}\left(\frac{p_{i+1,j,k} - p_{i,j,k}}{\Delta x}\right)$$

We use the standard second order accurate ghost fluid boundary condition for pressures lying on the other side of a free surface [Gibou et al. 2002].
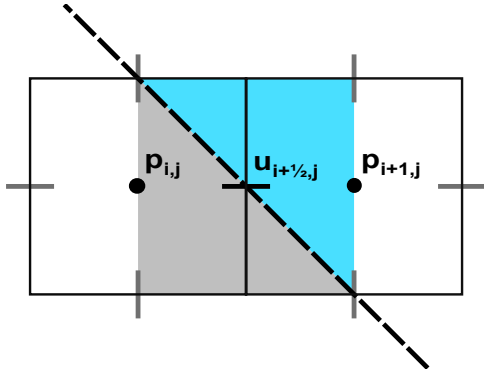
Figure 3: *Area weights in 2D: The shaded region, in blue (fluid) and grey (solid), indicates the staggered cell surrounding velocity sample $u_{i+\frac{1}{2},j}$ on a standard MAC grid. The area used to compute the corresponding mass, $m_{i+\frac{1}{2},j}$, is that of the fluid region.*

The kinetic energy integral of the fluid decouples into the sum of the kinetic energies from the $x-$, $y-$, and $z-$ components of fluid velocity, which we approximate individually:

$$
\mathrm{KE}_F \approx \sum_{i,j,k} \frac{1}{2} m_{i+1/2,j,k} u_{i+1/2,j,k}^2 + \frac{1}{2} \sum_{i,j,k} m_{i,j+1/2,k} v_{i,j+1/2,k}^2
$$
$$
+ \frac{1}{2} \sum_{i,j,k} m_{i,j,k+1/2} w_{i,j,k+1/2}^2 \tag{4}
$$

Here the m's are estimates of the mass of the fluid in the Δx^3 cube surrounding the appropriate MAC velocity sample point: e.g. $m_{i+1/2,j,k}$ is ρ times the volume of fluid inside $[x_i, x_{i+1}] \times [y_{j-1/2}, y_{j+1/2}] \times [z_{k-1/2}, z_{k+1/2}]$. See figure 3 for a 2D example. These volumes can easily and efficiently be calculated exactly from a polygonal representation of the geometry, or they may be approximated by Δx times the area of the associated cell face (giving rise to a method related to Finite Volumes), or even just Δx^2 times the extent of the fluid on the line segment between pressure samples (calculated trivially from a level set representation). The problematic voxelized pressure solve corresponds to setting masses equal to $\rho \Delta x^3$ or 0, losing all sub-grid information about the boundary.

Expressing the vector of all fluid velocity components as **u** and pressures as **p**, the gradient finite difference operator as matrix **G** and the diagonal matrix of all the fluid cell masses as matrix $\mathbf{M}_F$, the discrete pressure update is

$$
\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \frac{\Delta t}{\rho} \mathbf{G}\mathbf{p} \tag{5}
$$

and the discrete fluid's kinetic energy is

$$
\begin{aligned}
\mathrm{KE}_F^{n+1} &= \frac{1}{2} \mathbf{u}_{n+1}^T \mathbf{M}_F \mathbf{u}_{n+1} \\
&= \frac{1}{2} \left(\tilde{\mathbf{u}} - \frac{\Delta t}{\rho} \mathbf{G}\mathbf{p} \right)^T \mathbf{M}_F \left(\tilde{\mathbf{u}} - \frac{\Delta t}{\rho} \mathbf{G}\mathbf{p} \right)
\end{aligned}
$$

We will add the terms corresponding to solids in later sections. For now notice that minimizing KE with respect to pressure is a weighted linear least-squares problem. Since the weights in $\mathbf{M}_F$ are non-negative masses, it is automatically well-posed (up to addition of pressure differences in the null-space of **G**, which of course have no influence on velocities). The normal equations are automatically a consistent, symmetric positive semi-definite linear system:

$$
\frac{\Delta t}{\rho^2} \mathbf{G}^T \mathbf{M}_F \mathbf{G}\mathbf{p} = \frac{1}{\rho} \mathbf{G}^T \mathbf{M}_F \tilde{\mathbf{u}} \tag{6}
$$

For binary voxel weights it is straightforward to see this is exactly the same as the traditional discrete pressure equation; in general we have the same 7-point-stencil sparsity structure, but with coefficients based on cell masses. For the sake of space we explicitly write only the 5-point two-dimensional version for cell i, j:

$$
\frac{\Delta t}{\rho^2 \Delta x^2} \begin{pmatrix} (m_{i+1/2j} + m_{i-1/2j} + m_{ij+1/2} + m_{ij-1/2}) p_{ij} \\ -m_{i+1/2j} p_{i+1j} - m_{i-1/2j} p_{i-1j} \\ -m_{ij+1/2} p_{ij+1} - m_{ij-1/2} p_{ij-1} \end{pmatrix} =
$$
$$
\frac{1}{\rho \Delta x} \begin{pmatrix} -m_{i+1/2j} u_{i+1/2j} + m_{i-1/2j} u_{i-1/2j} \\ -m_{ij+1/2} v_{ij+1/2} + m_{ij-1/2} v_{ij-1/2} \end{pmatrix} \tag{7}
$$

Note that we have multiplied both sides by -1 to make the system positive semi-definite. This is still a symmetric M-matrix, and thus may be solved efficiently with Modified Incomplete Cholesky Preconditioned Conjugate Gradient using exactly the same code as a traditional voxelized solver.

Figure 2 shows a comparison of a 2D vortex-in-a-box, simulated with the box grid-aligned (our ground truth), the box rotated and classic voxelized weights used, and the box rotated with our new scheme. The bumpy stair-step grid artifacts of the voxelized scheme are essentially eliminated with the variational approach. We also note that unlike previous partial fixes we are guaranteed to be stable, and for the hydrostatic case the exact hydrostatic pressure $p = -\rho gy$ is the solution to our minimization, perfectly canceling out gravitational acceleration (giving $\mathbf{u} = \mathbf{0}$).

3 Coupling Fluids and Rigid Bodies

3.1 Pressure Discretization

To couple a rigid body, we just approximate the **J** operator that maps pressure to net force and torque on the body for the pressure update, and add the kinetic energy of the solid to our minimization.

In continuous variables, the translational part of the J operator is defined by the net force equation:

$$
\mathbf{F} = \mathbf{J}_{trans}\mathbf{p} = -\iint_S p\hat{\mathbf{n}} \tag{8}
$$

where S is the full surface of the solid, $\hat{\mathbf{n}}$ is the outward pointing normal (hence the negative sign), and recalling $p = 0$ on dry parts of the solid surface. From the fundamental theorem of calculus this is equivalent to a volume integral:

$$
\mathbf{J}_{trans}\mathbf{p} = -\iiint_{\text{solid}} \nabla p \tag{9}
$$

where p is conceptually smoothly extended into the volume (all interior values cancel, thus we never actually refer to pressures beyond a grid cell into the interior). This can be discretized in a manner consistent with the fluid pressure solve. For example, for the horizontal component we have

$$
J_x\mathbf{p} = -\sum_{i,j,k} vol_{i+1/2,j,k} \frac{p_{i+1,j,k} - p_{i,j,k}}{\Delta x} \tag{10}
$$

The volume weights are the volume of the rigid body occupying the cell centered on each particular MAC grid velocity sample position, exactly analogous to the mass weights used to define kinetic energy of the fluid. For fully submerged cells, we can in fact compute the fluid weights by subtracting off the solid volumes (however they are approximated) from the volume of a full cell. We note that in the interior of the solid, where all the volumes are full, the sum telescopes and cancels out pressure unknowns in the interior.

The torque part of the J operator is likewise defined:

$$\mathbf{T} = \mathbf{J}_{rot}\mathbf{p} = -\iint_S (\mathbf{x} - \mathbf{X}_{com}) \times p\hat{\mathbf{n}} \qquad (11)$$

where $\mathbf{X}_{com}$ is the center of mass of the object. Again we transform this into a volume integral:

$$\mathbf{J}_{rot}\mathbf{p} = \iiint_{\text{solid}} \boldsymbol{\nabla} \times p(\mathbf{x} - \mathbf{X}_{com}) \qquad (12)$$

For each component of torque we approximate this with a sum, using volume weights, as for translation.

Note that if approximations to the volume weights are used in defining $\mathbf{J}$, the 6×6 mass matrix $\mathbf{M}_S$ used to compute the rigid body's kinetic energy from translation and rotation should ideally be consistent with those volumes, multiplied by rigid body density, rather than the exact mass matrix. However, this only becomes an issue for achieving perfect hydrostatic rest with neutrally buoyant rigid bodies, and may be ignored in more dynamic scenes.

Once $\mathbf{J}$ and $\mathbf{M}_S$ have been computed, we add the rigid body's terms to the kinetic energy minimization:

$$\text{KE}_S^{n+1} = \frac{1}{2}(\mathbf{V}^n + \Delta t \mathbf{M}_S^{-1}\mathbf{J}\mathbf{p})^T \mathbf{M}_S(\mathbf{V}^n + \Delta t \mathbf{M}_S^{-1}\mathbf{J}\mathbf{p}) \qquad (13)$$

This modifies the linear system for pressure, equation (6), by adding $\Delta t \mathbf{J}^T \mathbf{M}_S^{-1}\mathbf{J}$ to the matrix and $-\mathbf{J}^T \mathbf{V}^n$ to the right-hand side. The sparsity of this addition depends on how many grid cells the solid boundary overlaps; we currently naïvely use a general sparse matrix data structure to handle it, but note that the addition is low rank (rank 6) which could be gainfully exploited by more sophisticated numerical linear algebra.

We do highlight an assumption used in this derivation: rigid bodies are thick enough to have an interior sampled on the grid. For thin rigid bodies, shells in particular, this is violated and the above approach does not work as described. There we need some method for encoding the unknown discontinuous pressure jump from one side of the rigid body to the other. We expect, in future work, to define ghost pressures on either side of such bodies, where we use the ghost pressure rather than the real pressure on the other side. A similar approach was successfully adopted by Tam et al., who simulated fluid interaction with thin rods, albeit for high-speed compressible flows [2005].

To handle scripted rigid bodies (objects with prescribed motion unaffected by the fluid) we let $\mathbf{M}_S^{-1}$ drop to zero, removing that term from the matrix but keeping the contribution to the right-hand side
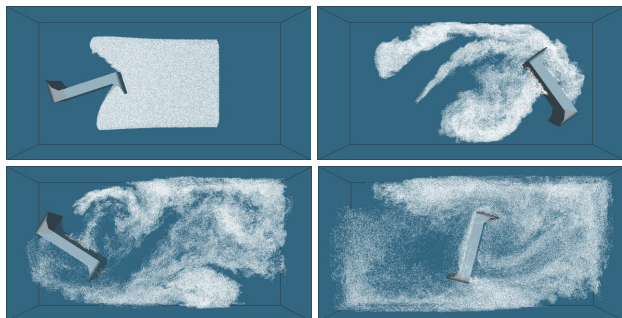
Figure 4: *Simulation of a paddle rotating through smoke on a* $80 \times 40 \times 60$ *grid, running at 3 seconds per frame. Note the fine-scale turbulent vortices captured by our approach.*

of the linear system. Note that if a scripted motion constrains the fluid to compress (e.g. in a piston), the linear system becomes inconsistent. If the user insists on this scenario, we remove the null-space component of the right-hand side as in Guendelman et al.'s work and allow the fluid to change volume [2005].

3.2 Time Integration

For time integration, we use the following scheme at each time step:

- Advance fluid positions (advection) and rigid body positions/orientations independently with current velocities.
- Process collisions.
- Add Δt times body forces to all velocities.
- Solve the energy minimization problem for pressure.
- Update fluid and solid velocities with pressure.

Note that for advection the tangential fluid velocity should be extrapolated into sample points with zero fluid mass, similar to Houston et al. [2003]. In future work we plan to add frictional contact forces to the energy minimization problem, which extends it to a Quadratic Program (QP) with constraints; currently our simulations use the simpler rigid body algorithm of Guendelman et al. [2003].

3.3 Results

We ran our simulator on several examples comparable to previous papers, on an older 2.8 GHz Pentium 4 desktop. We begin with the paddle wheel by Klingner et al. [2006]: they report simulation times of approximately one minute per frame. On a strictly larger $80 \times 40 \times 60$ grid that approximately matches their smallest tetrahedra, our code runs at 3 seconds per frame, a factor of 20 speed-up (presumably due to the overhead of the unstructured mesh). However, this grid contains more velocity samples than the tetrahedral mesh, and due to the sharper interpolation possible on a regular grid gives significantly more detailed results. If we instead find a grid size, $40 \times 20 \times 30$, that better matches the look of the original (albeit still ending with a much higher degree of turbulent mixing from more finely resolved vortices than the original—even though we do not use vorticity confinement in our simulation) our simulator runs interactively at 5 frames per second, a factor of 300 speed-up. We observe here the critical importance of methods which can accurately capture details on coarse grids. Figure 1, left, shows frames from the coarse grid, and figure 4 from the higher resolution grid for comparison.

Figure 1, middle, shows a simulation of a variety of rigid bodies of differing densities in a fluid-filled container. The asteroid object (in cyan), however, has a density 0.1 times that of the fluid which Carlson et al.'s algorithm cannot stably handle [2004]. Our simulation on a $60 \times 90 \times 60$ grid, ran at 25 seconds per frame. Figure 1, right, shows an interactive simulation of 2 complex solids immersed in a fluid, running at 2 frames per second on a $20 \times 20 \times 20$ grid. The user interactively selects the heavy blue bunny (selection in white), drags it up, and launches it so that it collides with the much lighter red dragon.

We believe that by exploiting the low rank of the rigid body additions to the matrix in the future, we will achieve further significant improvements, particularly since the most time is spent on easily optimized matrix-vector multiplies.

Finally, to illustrate more clearly the ability of our model to capture sub-grid details, figure 5 shows frames from a 2D animation of a heavy rigid box nearly blocking a fluid channel. The gaps on either side of the box are only half a grid cell wide, yet fluid convincingly flows past, jostling the box from side to side.
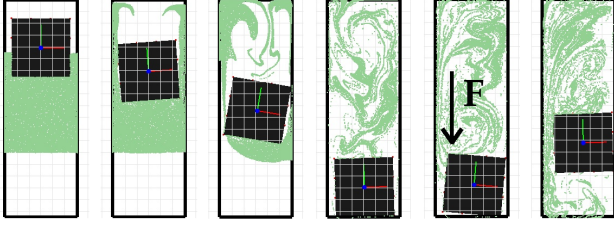
Figure 5: *Our variational framework gives sub-grid resolution in this rigid body flow example, allowing efficient and plausible solution on a coarse grid. The box sinks in a slightly larger tube, jostled from side to side by the fluid; later an applied force **F** drives fluid in sub-grid gaps to push the box upwards. Fluid flow is visualized with marker particles.*

Figure 6: *A ball of water splashes against the left wall. In the top row, the standard solid wall boundary condition is used, resulting in fluid unnaturally sticking to walls. In the bottom row, our new wall separation condition lets the fluid peel off plausibly.*

4   Wall Separation

A common numerical artifact seen in free-surface water simulations is fluid "crawling" up walls and even along ceilings, eventually dripping down or crossing over to another wall to descend. The source of this problem is the $\mathbf{u} \cdot \hat{\mathbf{n}} = \mathbf{v}_{solid} \cdot \hat{\mathbf{n}}$ boundary condition, which states that fluid cannot flow into or out of a solid. While this is well established physically for many flow situations, it has the unfortunate side-effect of not allowing fluid to separate from a wall, a phenomena which is readily observed in everyday life. Without getting into the physical chemistry of molecular-scale interactions that actually govern surface wetting/drying (and that are not captured by continuum mechanics) we argue heuristically that in reality only a thin film of fluid is left on the wall in these situations. This film, at most a wet patch, is far too small to be resolved on an animation grid. Simulations using this boundary condition instead enforce a layer of thickness unrealistically proportional to the grid cell size that sticks to the wall, and rely on numerical error in advection to eventually separate it.

Foster and Fedkiw [2001] (with extensions by Houston et al. [2003] and Rasmussen et al. [2004]) observed this problem and offered a fix which works well in certain highly dynamic splashing situations. After advecting and applying forces, if fluid velocities are found to be separating from solids ($\tilde{\mathbf{u}} \cdot \hat{\mathbf{n}} > \mathbf{v}_{solid} \cdot \hat{\mathbf{n}}$), that separation velocity is enforced in the pressure solve. However, it becomes unstable and physically implausible in more static conditions with oblique boundaries (as mentioned in section 1.1), and completely fails for closed or nearly closed fluid-filled containers. We instead exploit our variational framework to arrive at a robust, physically consistent solution.
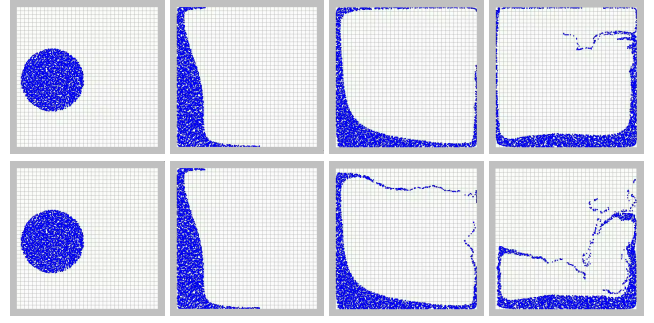
Essentially we want to enforce the boundary condition:

$$\mathbf{u} \cdot \hat{\mathbf{n}} \geq \mathbf{v}_{solid} \cdot \hat{\mathbf{n}} \qquad (14)$$

allowing the fluid to separate from the wall but not flow into it. If it separates from the wall, it becomes a free surface, $p = 0$, but if not we argue one appropriate condition on pressure is $p > 0$: we rule out suction from keeping the fluid stuck. This is then a complementarity condition:

$$0 \leq p \perp \mathbf{u} \cdot \hat{\mathbf{n}} - \mathbf{v}_{solid} \cdot \hat{\mathbf{n}} \geq 0 \qquad (15)$$

This is equivalent to turning our kinetic energy minimization problem into an inequality-constrained QP, with just the linear constraint $p \geq 0$ on solid boundaries: the complementarity is automatically enforced for us by the KKT conditions. Thus we can again avoid discretizing the boundary condition, relying on the discretization of the variational principle to automatically capture it.

Parenthetically, this makes the analogy between solving for pressure and solving for rigid body contact even closer. In rigid body contact, contact forces or impulses are constrained to be nonnegative with a complementarity condition on relative velocity, allowing bodies to separate but not interpenetrate.

Figure 6 shows a 2D comparison of the standard boundary condition and our wall separation condition. We used the PATH solver [Ferris and Munson 1998] to solve the equivalent KKT Linear Complementarity Problem, whose performance limits us to relatively small problem sizes; in future work we plan to investigate more scalable QP solvers.

5   Conclusion

We introduced a variational framework for pressure in fluid flow, allowing easy coupling to solids with arbitrary geometry not aligned with the grid. By exploiting the demonstrated sub-grid accuracy of this approach, the desirable properties of the resulting linear system and the efficiency of Cartesian grid-based simulation, we achieve a performance gain of one or two orders of magnitude over existing techniques, and overcome many limitations associated with previous methods. In addition we introduced a novel wall separation boundary condition, which fits naturally in the variational framework and robustly eliminates unwanted sticky artifacts which have plagued free surface simulations in the past.

Due to the conceptual simplicity of our framework, we believe that extending the coupling mechanism to arbitrary deformable bodies should be straightforward, and preliminary results indicate this to be the case. In future work we also plan to properly account for thin solids with ghost pressure values, exploit the low rank of rigid body matrix additions to improve performance, and use a more scalable QP solver to better handle frictional rigid body contacts and wall separation.

6   Acknowledgements

References

ANGELIDIS, A., NEYRET, F., SINGH, K., AND NOWROUZEZAHRAI, D. 2006. A controllable, fast and stable basis for vortex based smoke simulation. In *ACM-EG Proc. Symposium on Computer Animation*, 25–32.

BARAFF, D. 1996. Linear-time dynamics using Lagrange multipliers. In *Computer Graphics Proc. (Proc. SIGGRAPH)*, 137–146.

BRIDSON, R., MÜLLER-FISCHER, M., AND GUENDELMAN, E. 2006. Fluid simulation. In *ACM SIGGRAPH Course Notes*.

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

CHENTANEZ, N., GOKTEKIN, T. G., FELDMAN, B. E., AND O'BRIEN, J. F. 2006. Simultaneous coupling of fluids and deformable bodies. In *ACM-EG Proc. Symposium on Computer Animation*, 83–89.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Computer Graphics Proc. (Proc. SIGGRAPH)*, 15–22.

FELDMAN, B. E., O'BRIEN, J. F., AND KLINGNER, B. M. 2005. Animating gases with hybrid meshes. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3, 904–909.

FERRIS, M. C., AND MUNSON, T. S. 1998. Complementarity problems in GAMS and the PATH solver. Tech. Rep. Mathematical Programming 98-12, Computer Sciences Dept., University of Madison.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Computer Graphics Proc. (Proc. SIGGRAPH)*, 23–30.

FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical models and image processing: GMIP 58*, 5, 471–483.

GÉNEVAUX, O., HABIBI, A., AND DISCHLER, J.-M. 2003. Simulating fluid–solid interaction. In *Graphics Interface*, 31–38.

GIBOU, F., FEDKIW, R., CHENG, L.-T., AND KANG, M. 2002. A second order accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys. 176*, 205–227.

GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. 2003. Nonconvex rigid bodies with stacking. *ACM Trans. Graph. (Proc. SIGGRAPH) 22*, 871–878.

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

HARLOW, F. H., AND WELCH, J. E. 1965. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Phys. Fluids 8*, 2182–2189.

HIRT, C. W., AMSDEN, A. A., AND COOK, J. L. 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys. 14*, 3, 227–253.

HOUSTON, B., BOND, C., AND WIEBE, M. 2003. A unified approach for modeling complex occlusions in fluid simulations. In *Proc. SIGGRAPH Sketches & applications*.

IRVING, G., GUENDELMAN, E., LOSASSO, F., AND FEDKIW, R. 2006. Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 805–811.

JOHANSEN, H., AND COLELLA, P. 1998. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys. 147*, 1, 60–85.

KEISER, R., ADAMS, B., GASSER, D., BAZZI, P., DUTRÉ, P., AND GROSS, M. 2005. A unified Lagrangian approach to solid-fluid animation. In *Eurographics Symposium on Point-Based Graphics*, 125–148.

KIRKPATRICK, M. P., ARMFIELD, S. W., AND KENT, J. H. 2003. A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid. *J. Comput. Phys. 184*, 1–36.

KLINGNER, B. M., FELDMAN, B. E., CHENTANEZ, N., AND O'BRIEN, J. F. 2006. Fluid animation with dynamic meshes. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, 820–825.

LE, D. V., KHOO, B. C., AND PERAIRE, J. 2006. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. *J. Comp. Phys. 220*, 109–138.

LIU, H., KRISHNAN, S., MARELLA, S., AND UDAYKUMAR, H. 2005. Sharp interface Cartesian grid method II: A technique for simulating droplet interactions with surfaces of arbitrary shape. *J. Comput. Phys. 210*, 1, 32–54.

LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3, 457–462.

MARELLA, S., KRISHNAN, S., LIU, H., , AND UDAYKUMAR, H. 2005. Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations. *J. Comput. Phys. 210*, 1, 1–31.

PESKIN, C. S. 2002. The immersed boundary method. *Acta Numerica 11*, 479–517.

RASMUSSEN, N., ENRIGHT, D., NGUYEN, D., MARINO, S., SUMNER, N., GEIGER, W., HOON, S., AND FEDKIW, R. 2004. Directable photorealistic liquids. In *ACM-EG Proc. Symposium on Computer Animation*, 193–202.

ROBLE, D., BIN ZAFAR, N., AND FALT, H. 2005. Cartesian grid fluid simulation with irregular boundary voxels. In *Proc. SIGGRAPH Sketches & applications*.

SCHWARTZ, P., BARAD, M., COLELLA, P., AND LIGOCKI, T. 2006. A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions. *J. Comput. Phys. 211*, 2, 531–550.

TAKAHASHI, T., HEIHACHI, U., AND KUNIMATSU, A. 2002. The simulation of fluid-rigid body interaction. In *Proc. SIGGRAPH Sketches & applications*.

TAM, D., RADOVITZKY, R., AND SAMTANEY, R. 2005. An algorithm for modelling the interaction of a flexible rod with a two-dimensional high-speed flow. *Int. J. Numer. Meth. Engng 64*, 1057–1077.

THÜREY, N., IGLBERGER, K., AND RÜDE, U. 2006. Free surface flows with moving and deforming objects with LBM. In *Vision, Modeling, and Visualization*.

UDAYKUMAR, H. S., MITTAL, R., RAMPUNGGOON, P., AND KHANNA, A. 2001. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *J. Comput. Phys. 174*, 345–380.

ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3, 965–972.