ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2018
T. Beeler and N. Thuerey
(Guest Editors)

Volume 37 (2018), Number 8

An Efficient Solver for Two-way Coupling Rigid Bodies with Incompressible Flow

Mridul Aanjaneya
Department of Computer Science, Rutgers University

Abstract

We present an efficient solver for monolithic two-way coupled simulation of rigid bodies with incompressible fluids that is robust to poor conditioning of the coupled system in the presence of large density ratios between the solid and the fluid. Our method leverages ideas from the theory of Domain Decomposition, and uses a hybrid combination of direct and iterative solvers that exploits the low-dimensional nature of the solid equations. We observe that a single Multigrid V-cycle for the fluid equations serves as a very effective preconditioner for solving the Schur-complement system using Conjugate Gradients, which is the main computational bottleneck in our pipeline. We use spectral analysis to give some theoretical insights behind this observation. Our method is simple to implement, is entirely assembly-free besides the solid equations, allows for the use of large time steps because of the monolithic formulation, and remains stable even when the iterative solver is terminated early. We demonstrate the efficacy of our method on several challenging examples of two-way coupled simulation of smoke and water with rigid bodies. To illustrate that our method is applicable to other problems, we also show an example of underwater bubble simulation.

CCS Concepts

•***Computing methodologies*** → *Computer graphics; Physical simulation;*

1. Introduction

Dynamic interactions of solid objects with fluids are ubiquitous in our day-to-day lives, such as a rising air balloon, fishes swimming under water, a ship floating in the sea, a flying airplane, etc. As a result, designing efficient methods for physically simulating such effects is an active area of research in Computer Graphics. The simplest approach to solid-fluid coupling is a *partitioned* scheme, which requires going back-and-forth between the solid and the fluid with appropriate boundary conditions [AGHmD03, GSLF05]. While this allows the straightforward reuse of existing solvers for both the solid and the fluid, such schemes are known to suffer from stability issues in the presence of large density ratios, requiring the use of small time steps [RMSF11, APF13]. Thus, researchers have explored the design of *monolithic* schemes which alleviate these issues and allow for large time steps [CGFO06, BBB07, RMSG*08, RMSF11, TLK16, ZB17]. While several compelling examples have been demonstrated in prior work, very little work has been done on fast solvers for such systems (beyond numerical discretization), despite the continued progress in the design of efficient solvers for both solids [ZSTB10, MZS*11, BML*14, NOB16, MDS16, LBK17] and fluids [MST10, ZB14, SABS14, LMAS16, CZY17]. The monolithic system exhibits poor conditioning when the solid and fluid have widely different densities, resulting in slow convergence rates.

We consider the problem of two-way coupling rigid bodies which are simulated using Lagrangian meshes [BET14] with Eulerian grid-based incompressible fluids [Bri15]. Our approach is based on the theory of *Domain Decomposition*, and in particular, *Schur-complement methods* [QV99], which are a class of divide-and-conquer algorithms. Previously, this method has been used for fluid control [RTWT12], character skinning [GMS14], nonlinear elasticity [MDS16], and the Poisson equation for incompressible fluids [LMAS16, CZY17]. While [BBB07] also used this technique for solid-fluid coupling, they assembled the entire Schur-complement system as a sparse matrix and did not explore preconditioning of this system. From an algebraic perspective, the Schur-complement method can be used to temporarily decouple the solid equations and solve them independently, and then subsequently combine them to obtain the global solution for the fully coupled monolithic system. Since each rigid body has only 6 degrees of freedom (a translational and a rotational velocity), the solid equations comprise a strictly lower dimensional problem, requiring force terms only on the boundary of all rigid bodies. Thus, the solid equations can be solved cheaply and inexpensively, making the divide part of the algorithm computationally very efficient. However, one is confronted with the significant challenge of efficiently solving the Schur-complement system (the "conquer" part), which makes it difficult to simply reuse an existing fast Poisson solver. By using an iterative Krylov solver, we avoid explicitly building this system as a sparse matrix, similar to [LMAS16, CZY17]. Our key observation is that a single Multigrid V-cycle for the fluid equations [MST10] is a very effective preconditioner for this system, facilitating efficient solutions using an assembly-free implementation of Conjugate Gradients. Our method is simple to implement, allows

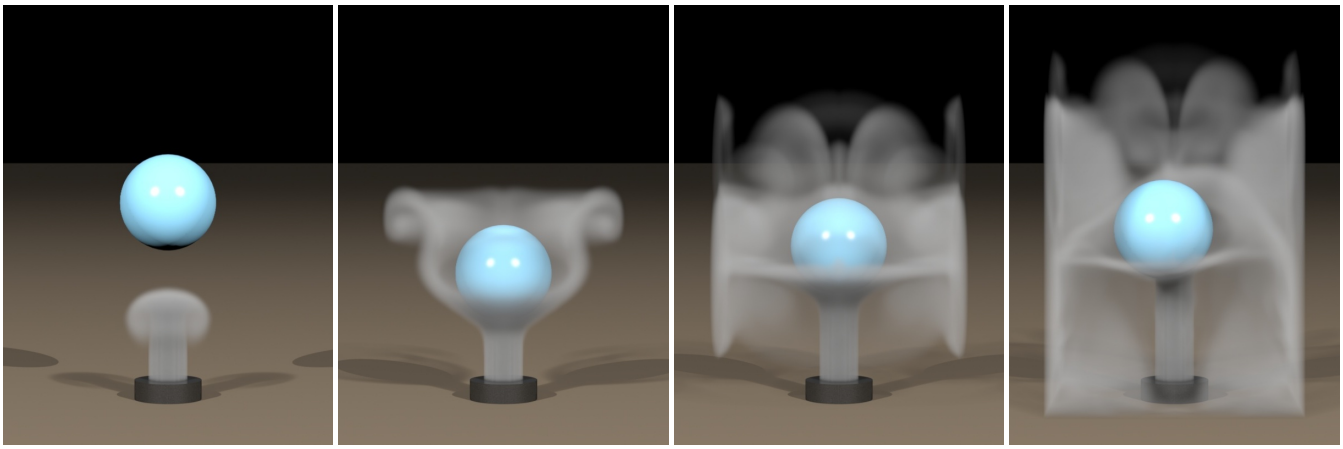


Figure 1: *A light ball is dropped on a smoke plume (*$128 \times 256 \times 128$ *grid). The ball sinks until buoyancy forces balance its weight, after which it begins to rise. Subsequently, instabilities develop in the flow field, breaking the symmetry of the flow and causing the ball to topple.*

for large time steps because of the monolithic formulation, and remains stable even when the iterative solver is terminated early.

In summary, our technical contributions are:

- the observation that a Multigrid V-cycle for the fluid part is a very effective preconditioner for the Schur-complement system;
- a hybrid implementation that combines direct solvers for the solid with iterative fluid solvers, showing robustness to poor conditioning of the coupled system with large density ratios; and
- a further application of our method to bubble simulation.

2. Related Work

Since our focus is on two-way coupled interactions between rigid bodies and incompressible fluids, we intentionally omit discussing prior work on solid-fluid coupling with deformable bodies, although several nice methods have been proposed [CGFO06, RMSG*08, RMSF11, MEB*12, CWSO13, LJF16, TLK16, ZB17].

Coupling Eulerian Fluids to Lagrangian Solids Eulerian fluid simulation commonly uses the marker-and-cell (MAC) discretization [HW65] which was first introduced to Computer Graphics by [FM96], and later improved by [Sta99, FF01]. Since realistic two-way coupling requires accurate treatment of solid boundary conditions, volumetric meshes were used by [FOKG05] to correctly capture the fluid region, and this approach was extended to dynamic coupling with rigid bodies in [KFCO06]. While several compelling examples were demonstrated, frequent remeshing of the entire fluid region presents a substantial bottleneck. The "rigid-fluid" method for two-way coupling was proposed in [CMT04], which momentarily treats the rigid body as fluid, but this can result in visual artifacts such as fluid leaking through the solid. A more accurate "leakproof" treatment was proposed in [GSLF05] that used one-sided interpolation during the advection step. However, they required small time steps for stability because of the partitioned nature of their method. The popular technique of "cut-cell" methods was introduced in [RZF05] for accurately treating irregular geometry in the pressure projection step by clipping the Eulerian grid cells to conform to the object boundaries. A similar treatment for two-way coupling with rigid bodies was proposed in [BBB07] by formulating the problem as an energy minimization that accounts for partial cell volumes. In this work, we follow the constraint-based approach of [RMSF11] which, for the special case of rigid bodies without any damping, reduces to the formulation in [BBB07].

Coupling Lagrangian Fluids to Lagrangian Solids Lagrangian mesh-based liquids offer several attractive features, such as good volume preservation, support for implicit surface tension, and straightforward solid-fluid coupling through a unified formulation, albeit, at the expense of continuous remeshing of the entire fluid region [MEB*12, CWSO13]. Moreover, gases are not readily supported by such methods. Particle-based methods, on the other hand, offer a mesh-free formulation and avoid this issue [SSP07, AIA*12, MMCK14, MHNT15]. However, their unstructured nature forgoes the benefits of regular grid-based data structures and numerical solvers [MST10]. Recently, the material point method (MPM) has been used to simulate many interesting phenomena with two-way coupling [SSC*13, SSJ*14, RGJ*15, JSS*15, KGP*16]. Particles are used as primary material carriers, avoiding the numerical dissipation issues present in Eulerian advection methods, while a background grid is used as a "scratchpad" for force computations, benefiting from cache locality and regular numerical stencils. Since our focus is on coupling mesh-based rigid bodies to grid-based incompressible fluids, such methods are outside the scope of our work.

Comparison to prior work Our approach is inspired by ideas proposed in [LMAS16, CZY17], so we highlight some key differences. First, the focus of [LMAS16, CZY17] was on the Poisson problem for incompressible fluids, while we consider monolithic two-way coupling between rigid bodies and incompressible fluids. Second, the methods proposed by [LMAS16, CZY17] offer computational gains only for extremely large problem sizes (exceeding hundred million variables), while our approach offers performance benefits on modest 2D problems (see Section 6). In part, this is because the Poisson problem for incompressible fluids is well understood, with mature implementations readily available, as opposed to monolithic two-way coupling which has received less attention so far. Third, we do not formulate the Schur-complement system on a lower dimensional interface. As a consequence, our approach does not require approximations (for computational feasibility) that compromise the accuracy of the entire operator, unlike [LMAS16].

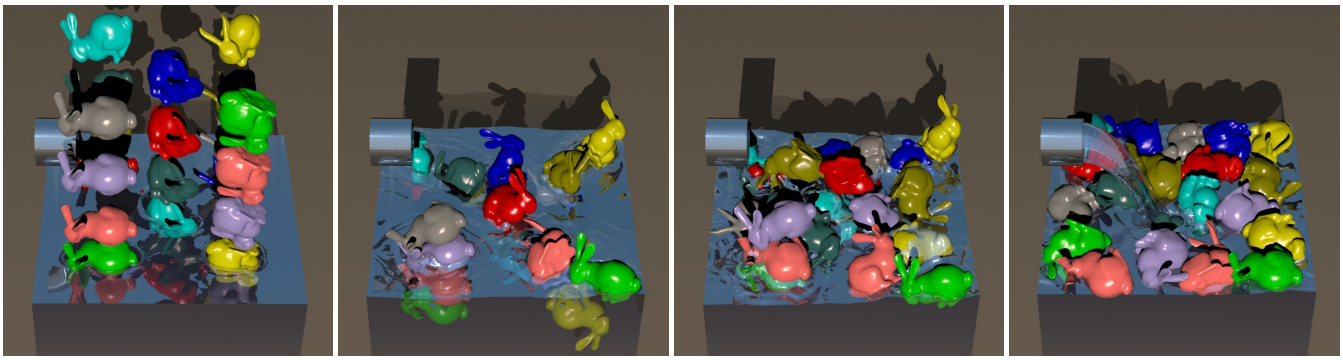


Figure 2: 20 *light bunnies are dropped in a pool of water (*128^3 *grid) demonstrating contact and collisions with two-way coupling. The bunnies initially take a plunge, but eventually rise to the surface. Subsequently, a source starts pouring water, further stirring up the motion.*

Finally, we show that a single Multigrid V-cycle for the fluid equations [MST10] is a very effective preconditioner for solving the Schur-complement system using Conjugate Gradients, greatly simplifying implementation. Our use of a direct solver for the solid equations and an iterative solver for the fluid equations is inspired by [MDS16], who used a similar approach for nonlinear elasticity. However, in the absense of a good preconditioner, they used a fixed number of iterations of Conjugate Gradients for solving the Schur-complement system, relying on subsequent Newton iterations to further reduce the error. Instead, we solve the Schur-complement system *exactly* using Preconditioned Conjugate Gradients.

3. The Schur-complement Method

We first review the algebraic theory behind the Schur-complement method [QV99], and show how the numerical solution of a coupled system can be cast as a divide-and-conquer algorithm. Subsequently, we highlight the mathematical principles that underlie the design of our preconditioner for monolithic two-way coupling.

3.1. Divide-and-conquer

Consider a domain Ω that has been partitioned into two subdomains Ω_1 and Ω_2. Assume we are solving a linear system $Ax = b$ on Ω, and the degrees of freedom have been reordered such that

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{1}$$

where x_i, b_i correspond to values in Ω_i, for $i \in \{1,2\}$. Then the matrix A can be decomposed into the following block form:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2}$$

In our case, A is symmetric, and so $A_{21} = A_{12}^T$. Moreover, the off-diagonal matrix A_{12} is extremely sparse. The Schur-complement method effectively uses this decomposition to solve the system $Ax = b$ in four separate steps, as follows:

1. Solve $A_{22} f_2 = b_2$.
2. Compute $f_1 = b_1 - A_{12} f_2$.
3. Solve $\Sigma x_1 = f_1$.
4. Solve $A_{22} x_2 = b_2 - A_{21} x_1$.

The matrix $\Sigma = A_{11} - A_{12}A_{22}^{-1}A_{21}$ is the Schur-complement matrix. Suppose Ω_1 corresponds to the fluid, and Ω_2 to the solid. Since each rigid body has only 6 degrees of freedom, the solid equations comprise a very small system, permitting a sparse Cholesky factorization for efficiently inverting A_{22}. Similarly, the extreme sparsity of the off-diagonal matrices A_{12}, A_{21} allows for very efficient matrix multiplication, making step 3) the major bottleneck. The Schur-complement matrix Σ is symmetric and positive definite (SPD), provided that A is SPD [QV99]. We use Preconditioned Conjugate Gradients (PCG) for solving this system, noting that such iterative Krylov solvers never require the assembly of the full matrix, but only its action on an input vector [LMAS16, CZY17]. Thus, our method is assembly-free (excluding A_{22}) and memory-efficient.

3.2. Preconditioning the Schur-complement System

We derive an adequate preconditioner for the Schur-complement system by noting that the inverse of Σ can be written as follows:

$$(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} = (I - A_{11}^{-1}A_{12}A_{22}^{-1}A_{21})^{-1}A_{11}^{-1} \tag{3}$$

For the matrix $B = A_{11}^{-1}A_{12}A_{22}^{-1}A_{21}$, the spectral radius $\rho(B) < 1$. To see this, let $C = A_{12}A_{22}^{-1}A_{21}$. This implies $A_{11} = C + \Sigma$, and thus

$$\begin{aligned} \rho(A_{11}^{-1}C) &= \rho((C+\Sigma)^{-1}C) = \rho(\{C^{-1}(C+\Sigma)\}^{-1}) \\ &= \rho(\{I + C^{-1}\Sigma\}^{-1}) = \frac{1}{\lambda_{\min}(I + C^{-1}\Sigma)} \\ &= \frac{1}{1 + \lambda_{\min}(C^{-1}\Sigma)} = \frac{1}{1 + \lambda_{\min}(C^{-1/2}\Sigma C^{-1/2})} \end{aligned} \tag{4}$$

Let $D = C^{-1/2}\Sigma C^{-1/2}$. The last equality above uses the fact that $C^{-1}\Sigma$ and D are related by a similarity transform via $C^{1/2}$. Since C and Σ are SPD, $C^{1/2}$ is well-defined, and D is SPD as well. Thus, $\rho(D) > 0$ which gives $\rho(A_{11}^{-1}C) = \rho(B) < 1$. Similar ideas were also used in [LMAS16] to prove the convergence of their interface smoother. For such matrices, the following identity holds [Hea02]:

$$(I - B)^{-1} = I + B + B^2 + B^3 + B^4 + \ldots = \sum_{i=0}^{\infty} B^i \tag{5}$$

The above series can be truncated to k terms to obtain a "reasonable" approximation to $(I - B)^{-1}$, with the understanding that the overall accuracy increases as k increases. A naïve evaluation of this truncated series would require computing individual powers of B, leading to an algorithm that scales quadratically with k. Instead, Horner's method [Hea02], an efficient technique for evaluating such expressions, can be used to reduce the complexity to $O(k)$.

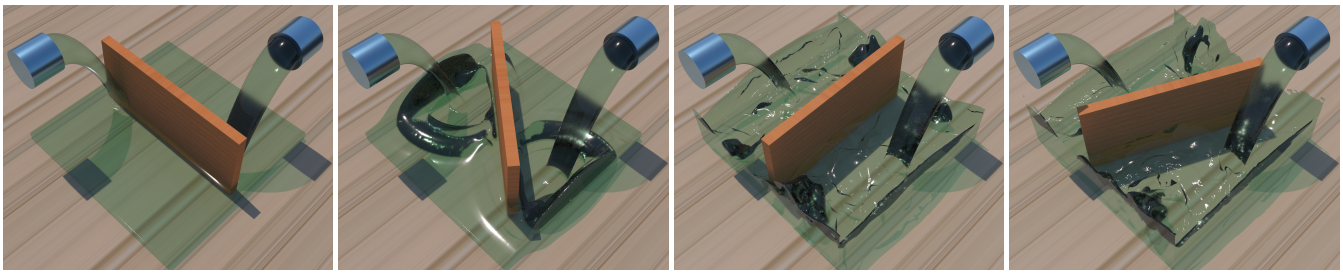


Figure 3: *Two sources pour water in a pool with a plank hinged at its center, but free to rotate (256^3 grid). The plank is only slightly shorter than the width of the container, forcing the water to continuously push it forward, creating rotational motion with interesting flow patterns.*

$$(I-B)^{-1} \approx I + \underbrace{B(I+B(I+B(\ldots)))}_{k \text{ terms}} \tag{6}$$

Equation (6) can be re-written as the following iterative scheme:

$$M^{(s+1)} = I + BM^{(s)} \tag{7}$$

where $M^{(0)} = I$, and $M^{(s)}$ is a parameterized family of approximations to $(I-B)^{-1}$. Multiplying equation (7) throughout by A_{11}^{-1} from the right, and substituting back $B = A_{11}^{-1}A_{12}A_{22}^{-1}A_{21}$ gives

$$M^{(s+1)}A_{11}^{-1} = (I + BM^{(s)})A_{11}^{-1} = A_{11}^{-1}(I + A_{12}A_{22}^{-1}A_{21}M^{(s)}A_{11}^{-1}) \tag{8}$$

From equation (3), note that $Q^{(s)} = M^{(s)}A_{11}^{-1}$ is an approximation to Σ^{-1}. Substituting this expression in equation (8) gives

$$Q^{(s+1)} = A_{11}^{-1}(I + A_{12}A_{22}^{-1}A_{21}Q^{(s)}) \tag{9}$$

where $Q^{(0)} = A_{11}^{-1}$. Thus, the system $\Sigma x = f$ could be solved by first computing $A_{11}^{-1}f$, and then using equation (9) iteratively to compute better approximations to $x = \Sigma^{-1}f$. This would be necessary if equation (9) was used as a *solver* for Σ, but since we intend to design a preconditioner, we can replace A_{11}^{-1} in equation (9) by a Multigrid V-cycle $A_{11}^{\dagger}$, to arrive at the following iterative scheme:

$$P^{(s+1)} = A_{11}^{\dagger}(I + A_{12}A_{22}^{-1}A_{21}P^{(s)}) \tag{10}$$

where $P^{(0)} = A_{11}^{\dagger}$. For incompressible fluids, few iterations of a Multigrid V-cycle forms a very effective preconditioner for Conjugate Gradients [MST10]. As equation (10) shows, with slightly extra work per V-cycle that requires adding a small correction term to account for the solid, this preconditioner can be extended to monolithic two-way coupling with rigid bodies in a straightforward fashion. Algorithm 1 presents pseudocode for applying equation (10).

Algorithm 1 Preconditioner application $P^{(s)}z = r$

```
Compute z = A_11^† r
for i = 1...s do
    Compute f_2 = A_21 z
    Solve A_22 x_2 = f_2
    Compute x_1 = A_12 x_2
    Compute z = A_11^† (r + x_1)
end for
```

As shown in Section 6, fewer iterations of PCG are required for convergence as the iteration level s is increased, validating the theory outlined above. However, the computational cost of each PCG iteration increases linearly with s. In practice, we observe that $s = 0$ works quite effectively, based on the per iteration cost, and the simulations remain stable even when the iterative solver is terminated early. We present examples in Section 6 to illustrate these benefits.

4. Governing Equations and Numerical Discretization

We review the fluid and solid equations, and describe the formulation in [RMSF11] for monolithic two-way coupling.

4.1. Fluid Equations

Consider the incompressible Euler equations

$$\vec{u}_t + (\vec{u}\cdot\nabla)\vec{u} + \frac{\nabla p}{\rho} = \vec{f}, \qquad \nabla\cdot\vec{u} = 0 \tag{11}$$

where ρ is the density, $\vec{u}$ is the velocity, p is the pressure, and $\vec{f}$ comprises of all external accelerations (such as gravity). We discretize these equations on a MAC grid [HW65] and use operator-splitting [Sta99] to first explicitly update the advection terms

$$\frac{\vec{u}^{\star} - \vec{u}^{n}}{\Delta t} + (\vec{u}^{n}\cdot\nabla)\vec{u}^{n} = \vec{f}^{n} \tag{12}$$

using the semi-Lagrangian advection scheme [Sta99]. Near solids, we clip the backward cast rays using the approach of [GSLF05]. Subsequently, we solve the Poisson equation for the pressure and enforce incompressibility by updating the velocity field as follows:

$$\vec{u}^{n+1} = \vec{u}^{\star} - \Delta t\frac{\nabla p}{\rho} \tag{13}$$

For interface tracking, we use the method of [ELF05], the reinitialization scheme of [LFO05], velocity extrapolation of [AS99], and the second order accurate pressure projection of [ENGF03].

4.2. Solid Equations

We evolve rigid bodies using Newton's laws of motion

$$\begin{aligned} x_t &= \vec{v}, & q_t &= \tfrac{1}{2}\omega q \\ \vec{v}_t &= \vec{F}/m, & L_t &= \tau \end{aligned} \tag{14}$$

where x is the position, q the orientation (in quaternions), m the mass, $\vec{v}$ the linear velocity, ω the angular velocity, $\vec{F}$ the net force, τ the net torque, and $L = I\omega$ the angular momentum with inertia tensor $I = RDR^T$ (R is the world space orientation matrix and D the diagonal inertia tensor in object space). We generally follow [GBF03] for collisions and contact, and [WTF06] for articulation.

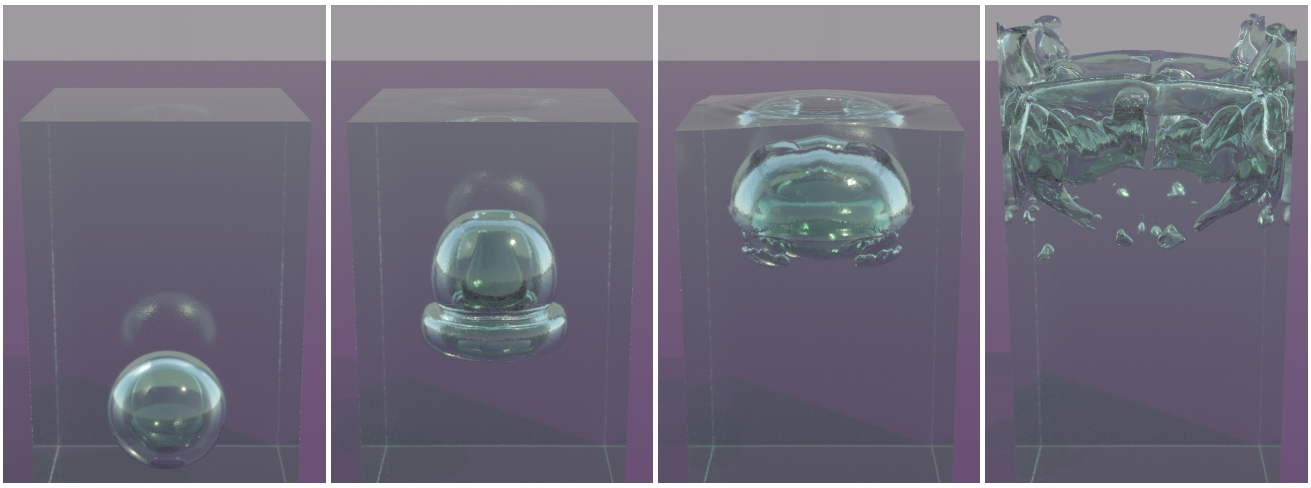
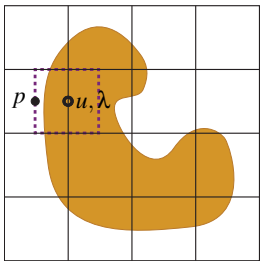


Figure 4: *An air bubble rises under buoyancy, undergoing complex topological deformations, before exiting the water* ($256^2 \times 512$ *grid).*

4.3. Coupling Fluids with Rigid Bodies



For two-way coupling, the governing equations require some modifications to capture the dynamic behavior of the solid-fluid interface. We follow the impulse-based approach of [RMSF11] that introduces a constraint whenever a ray cast between two grid cell centers, one of which is fluid, intersects a solid. The constraint is placed at the face between the two cells and enforces the projected solid and fluid velocities at that point to be equal via an impulse λ (see inset figure). To account for this impulse, equation (13) is modified as follows:

$$\beta \vec{u}^{n+1} = \beta \vec{u}^{\star} - \hat{G}\hat{p} + W^T \lambda \tag{15}$$

where V is the diagonal matrix of dual cell volumes, $\beta = \rho V$ is the dual cell mass matrix, $-\hat{G}^T = -V\nabla^T$ is the volume weighted divergence which has been modified to drop rows corresponding to cells without fluid, its negative transpose $\hat{G}$ is the volume weighted gradient, $\hat{p} = p\Delta t$ is the scaled pressure, and W^T is the matrix that maps the impulses λ to the appropriate fluid faces. Similarly, the solid equations are modified to account for the impulses as follows:

$$M\vec{v}^{n+1} = M\vec{v}^{\star} - J^T W^T \lambda \tag{16}$$

where M is the generalized mass matrix, $\vec{v}^{\star}$ is the generalized velocity that accounts for external forces, collisions, and contact, and J is the matrix that interpolates from solid degrees of freedom to fluid velocity locations. Combining equations (15) and (16) with the closure condition $W(\vec{u} - J\vec{v}) = 0$, that enforces the fluid and solid to move with the same velocity at the solid-fluid interface, gives the following SPD system after some rearranging

$$\begin{bmatrix} \hat{G}^T\beta^{-1}\hat{G} & -\hat{G}^T\beta^{-1}W^T \\ -W\beta^{-1}\hat{G} & A_{22} \end{bmatrix} \begin{bmatrix} \hat{p} \\ \lambda \end{bmatrix} = \begin{bmatrix} \hat{G}^T\vec{u}^{\star} \\ WJ\vec{v}^{\star} - W\vec{u}^{\star} \end{bmatrix} \tag{17}$$

where $A_{22} = W\beta^{-1}W^T + WJM^{-1}J^TW^T$, and $\vec{v}^{n+1}$ was eliminated using equation (16). As described in Section 3, we compute a sparse Cholesky factorization of A_{22} exploiting its low dimensionality,

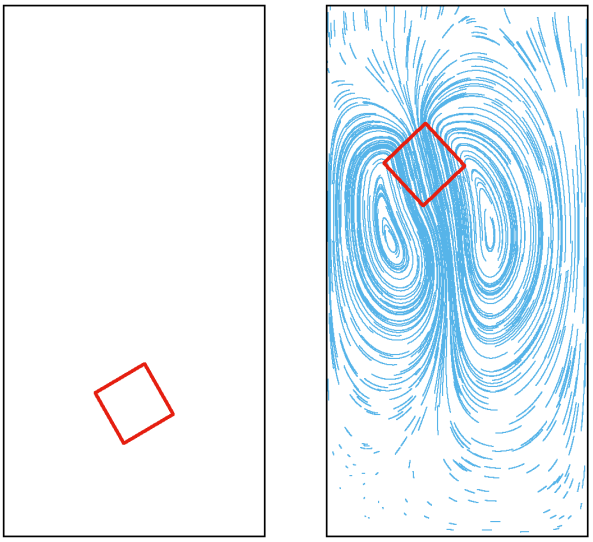


Figure 5: *A test problem where a light block with a slight initial rotation (left) rises under buoyancy forces (right). The streamlines show a continuous velocity field across the solid-fluid interface.*

and use PCG for efficiently solving the Schur-complement system. A slight technicality is our use of a high order defect correction [TOS01] to boost the accuracy of our first order accurate Multigrid V-cycle near the liquid surface, similar to [LMAS16, AGL*17].

5. Bubble Simulation

We were inspired by [APF13] for underwater bubble simulation, but opted for the approach in [GB17] because of its simplicity and memory efficiency. However, our formulation is equally applicable to [APF13]. The basic idea in [GB17] is to augment equation (13) with a volume preservation constraint for each bubble as follows

$$B_i(\vec{u}) = \int \int_{\partial\Omega_{A_i}} \vec{u} \cdot \vec{n} dA = 0 \tag{18}$$

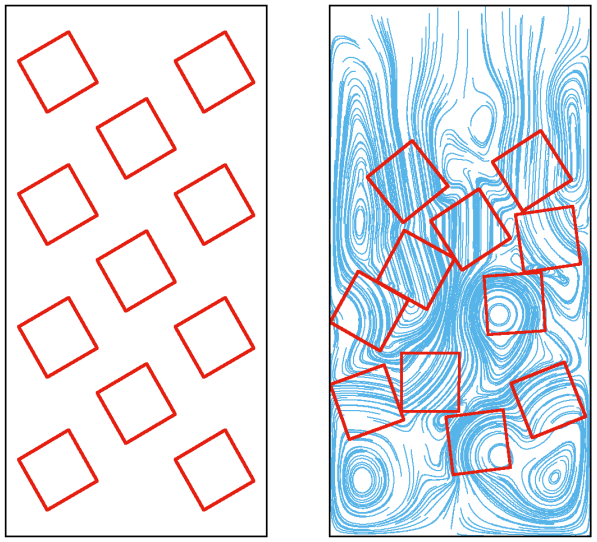


Figure 6: *A collection of blocks (left) interact with each other under buoyancy forces (right). Velocity streamlines are shown in blue.*

where Ω_{A_i} refers to the volume of the ith bubble. Enforcement of this constraint via an impulse λ requires knowledge of the velocity field *only* on the bubble surface, producing a similar low-dimensionality as the rigid body case. Assembling all bubble constraints B_i into a constraint matrix B, the velocity update becomes

$$\beta\vec{u}^{n+1} = \beta\vec{u}^{\star} - \hat{G}\hat{p} + B^T\lambda \tag{19}$$

Combining equations (18) and (19) with the constraint $-\hat{G}^T\vec{u}^{n+1} = 0$, and eliminating $\vec{u}^{n+1}$, we obtain the following coupled system:

$$\begin{bmatrix} \hat{G}^T\beta^{-1}\hat{G} & -\hat{G}^T\beta^{-1}B^T \\ -B\beta^{-1}\hat{G} & B\beta^{-1}B^T \end{bmatrix} \begin{bmatrix} \hat{p} \\ \lambda \end{bmatrix} = \begin{bmatrix} \hat{G}^T\vec{u}^{\star} \\ -B\vec{u}^{\star} \end{bmatrix} \tag{20}$$

Note the similarity of equations (17) and (20), making our formulation from Section 3 applicable for efficient numerical solution.

6. Results

To validate our solver against existing methods, we consider two test problems. In the first problem, a light block immersed in an incompressible fluid, with a slight initial rotation, rises under the effects of buoyancy forces (see Figure 5). For a 128×256 grid in 2D (resp. $128^2 \times 256$ grid in 3D), we list the average iteration count and timing information for each method in Table 1, and show the corresponding numerical convergence profiles in Figure 7. *Unpreconditioned CG* refers to the application of Conjugate Gradients with no preconditioning for the full monolithic system of equation (17). (The convergence profile for this method is not shown in Figure 7(top) because it is rather erratic, and difficult to scale to illustrate the profiles for all other methods.) The second method uses Conjugate Gradients without preconditioning to solve the Schur-complement system $\Sigma = A_{11} - A_{12}A_{22}^{-1}A_{21}$ (shown in Figure 7(top) under the name *CG*). Both these methods use an extremely large number of iterations (> 5000) for the 3D case, and so we do not show them. The third method uses a block diagonal preconditioner for the full system in equation (17) with an Incomplete Cholesky factorization for the fluid part, and a diagonal scaling matrix for the solid part [EQYF13]. For brevity, we refer to this method as *ICPCG*. Finally, we list the performance of the preconditioner from equation (10) for the Schur-complement system using a Multigrid V-cycle with 4 levels. Our preferred method corresponds to the case $s = 0$, which has the lowest computational cost per iteration. For larger values of s, the iteration count becomes smaller since the preconditioner more accurately approximates Σ^{-1}, albeit, at higher cost per iteration. In all cases, we used 3 boundary smoothing iterations and 1 interior smoothing iteration in our application of a V-cycle [MST10]. The second test problem considers multiple boxes of various densities immersed in an incompressible fluid, as shown in Figure 6. This problem is more challenging because of the additional solid wall boundary conditions. Figure 8 and Table 2 summarize the convergence behavior and performance of all solvers.

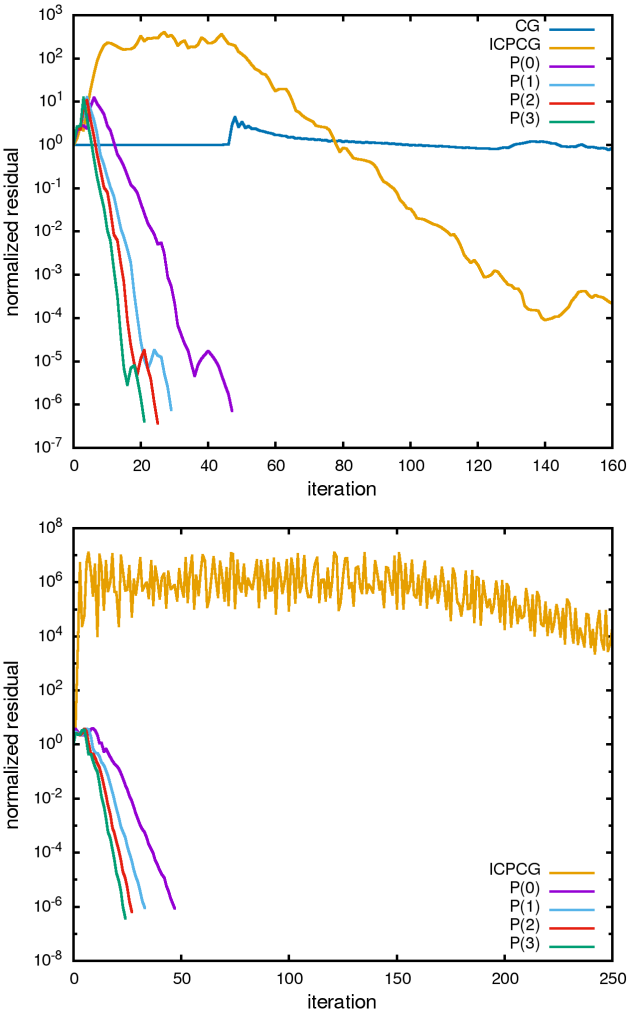


Figure 7: *Convergence profiles for unpreconditioned CG, ICPCG, and our preconditioner with various iteration levels in (top) two dimensions and (bottom) three dimensions, for the box lift problem.*

Solver	2D (128×256 grid)	3D ($128^2 \times 256$ grid)
Unpreconditioned CG	2603 iterations 0.3405 seconds	N/A
Unpreconditioned CG for Schur-complement	799 iterations 0.3694 seconds	N/A
ICPCG with diagonal scaling for solid part	204 iterations 0.1134 seconds	879 iterations 78.5363 seconds
$P^{(0)}$ (MGPCG)	47 iterations 0.1444 seconds	47 iterations 14.6466 seconds
$P^{(1)}$	29 iterations 0.1656 seconds	33 iterations 19.6749 seconds
$P^{(2)}$	25 iterations 0.2161 seconds	27 iterations 23.7428 seconds
$P^{(3)}$	21 iterations 0.2401 seconds	24 iterations 28.3989 seconds

Table 1: *Solver comparisons for the box lift problem from Figure 5. Timings computed on an Intel Core i7-7700 (4 Core, 3.6GHz) CPU.*

To highlight the robustness of our method against poor conditioning of the monolithic system in equation (17), we varied the density of the block from being lighter than the fluid, to progressively denser. For reference, we consider the fluid density $\rho = 1$. Figure 10 shows the average iteration count with $P^{(0)}$ as the preconditioner for Σ.

Solver	2D (128×256 grid)	3D ($128^2 \times 256$ grid)
Unpreconditioned CG	3649 iterations 0.4552 seconds	N/A
Unpreconditioned CG for Schur-complement	944 iterations 0.6121 seconds	N/A
ICPCG with diagonal scaling for solid part	350 iterations 0.1534 seconds	3023 iterations 225.5956 seconds
$P^{(0)}$ (MGPCG)	111 iterations 0.1618 seconds	121 iterations 149.1149 seconds
$P^{(1)}$	78 iterations 0.2153 seconds	85 iterations 212.226 seconds
$P^{(2)}$	71 iterations 0.2962 seconds	70 iterations 263.9951 seconds
$P^{(3)}$	55 iterations 0.2928 seconds	61 iterations 306.6343 seconds

Table 2: *Solver comparisons for the box lift problem from Figure 6. Timings computed on an Intel Core i7-7700 (4 Core, 3.6GHz) CPU.*

For reducing the residual by 6 orders of magnitude, our method is $1.5-5\times$ faster than ICPCG. In practice, however, the coupled system may only be solved for a limited time budget. To test this, we placed a hard limit of 2 seconds for the box lift problem from Figure 5. Figure 9(top) shows the box location after 0.6 seconds for our method (far left) and ICPCG (second to far left). Since ICPCG exhibits severe instability, we progressively increased the time budget to 4 seconds, 8 seconds, and 16 seconds. The corresponding results are also shown in Figure 9(top). For reference, the fully converged solution is also shown on the far right. From a visual perspective, our method looks almost identical to the reference solution. ICPCG becomes stable after $4\times$ more compute time, but does not achieve

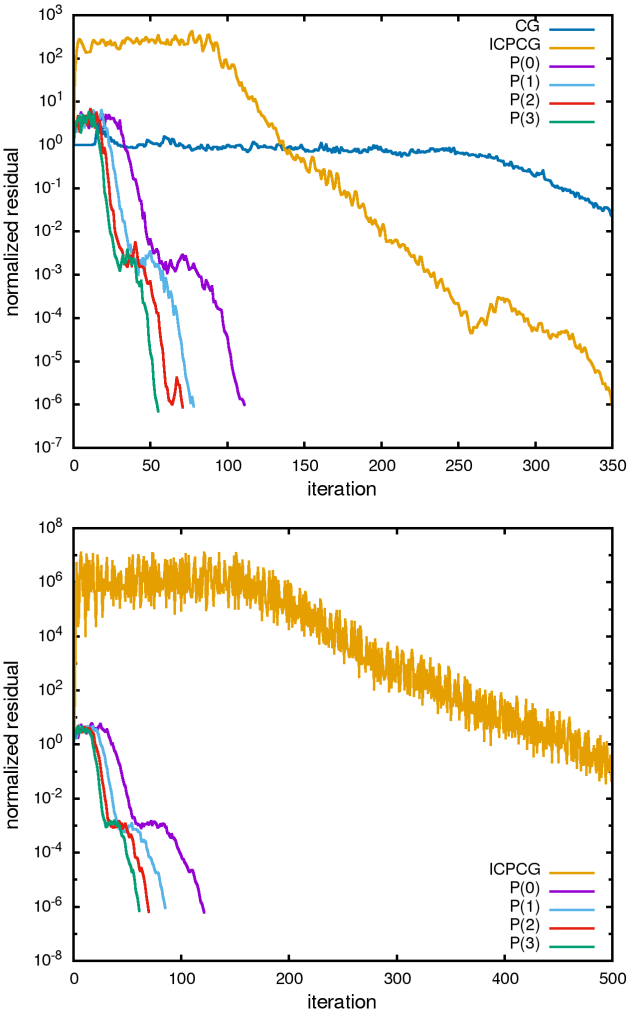


Figure 8: *Convergence profiles for unpreconditioned CG, ICPCG, and our preconditioner with various iteration levels in (top) 2D and (bottom) 3D, for the multiple box interaction problem in Figure 6.*

visual closeness to the reference solution (as our method does) even with $8\times$ more compute time. Note that 2 seconds is an insufficient time budget for our method to converge, so the buoyancy forces are slightly off and the box does not rise to the same height as the reference solution. We also simulated 15 bunnies that are fully immersed in an incompressible fluid with a hard time limit of 4 seconds for our solver (see Figure 9(bottom)). The bottom 5 bunnies are lighter than the fluid, and the top 10 bunnies are heavier. Our method remains stable even with complex contacts and collisions.

Simulation Examples To illustrate the benefits of our method, we simulated end-to-end examples of both smoke and water that are two-way coupled with rigid bodies. Figure 1 shows a light ball that is dropped on a smoke plume. The ball sinks under its own weight until buoyancy forces nullify its initial momentum, after which it begins to rise. Subsequently, the axial symmetry is broken because of instabilities that develop in the flow field, causing the ball to topple. Figure 2 shows 20 light bunnies that are dropped in a pool of

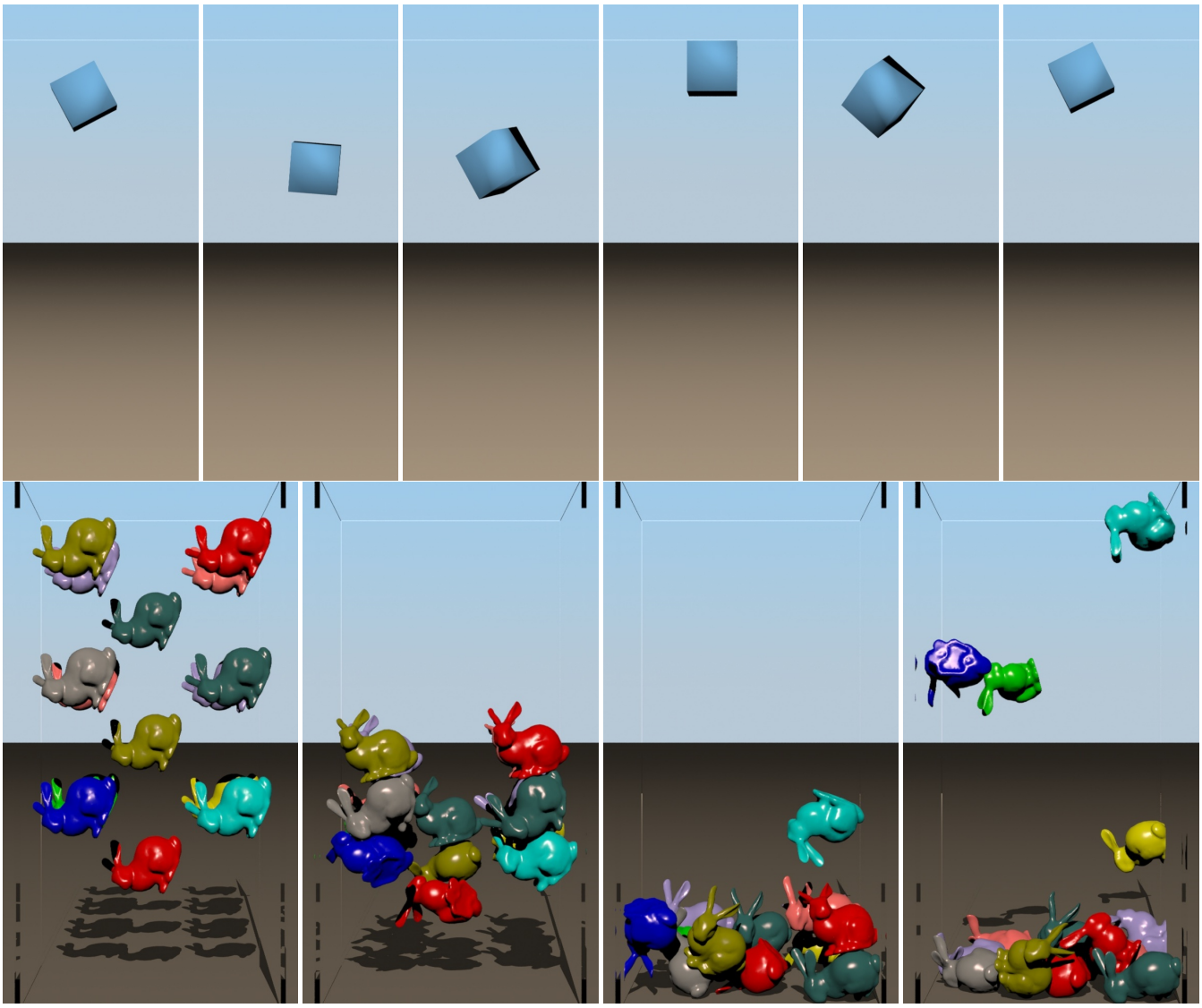


Figure 9: *(Top) The box lift problem on a* $128^2 \times 256$ *grid, simulated using our solver with a hard time limit of* 2 *seconds, ICPCG with a hard time limit of* 2 *seconds,* 4 *seconds,* 8 *seconds,* 16 *seconds, and with sufficient time for full convergence at each time step (for reference). (Bottom)* 15 *bunnies with various densities, fully immersed in an incompressible fluid, simulated on a* $128^2 \times 256$ *grid using our solver with a hard time limit of* 4 *seconds. Our method produces stable results even in the presence of complex contacts and collisions.*

water, which initially sink because of their momentum, but eventually float to the surface. After some time, a source starts pouring water, pushing some bunnies under water and stirring up the motion. Figure 3 shows two sources pouring water in opposite directions, hitting a wooden plank that is hinged at its center. The rotational motion in the water subsequently drives the plank forward, until it becomes perpendicular to the sources again. Finally, Figure 4 shows an air bubble rising under buoyancy, undergoing complex topological deformations before exiting the water surface. This example highlights that our method is also applicable to other problems beyond two-way coupling fluids with rigid bodies, as long as they admit a structure that fits within our formulation. Table 3 gives the average solve times per time step for all our examples.

	Figure 1	Figure 2	Figure 3	Figure 4
Projection	15.0856	19.9432	95.698	65.2402
MG levels	4	4	5	5
PCG iterations	42	20	24	29

Table 3: *Average solve times (in seconds) for all examples. Timings computed on an Intel Core i7-7700 (Quad Core, 3.6GHz) CPU.*

Implementation Notes Similar to prior work, we use the pressure from the previous time step as initial guess when solving for the pressure in the current time step, obtaining good convergence in fewer PCG iterations. As mentioned in [MST10], $P^{(0)}$ should use a zero initial guess for the preconditioner to be symmetric. However, when using multiple V-cycles, the result from the previous

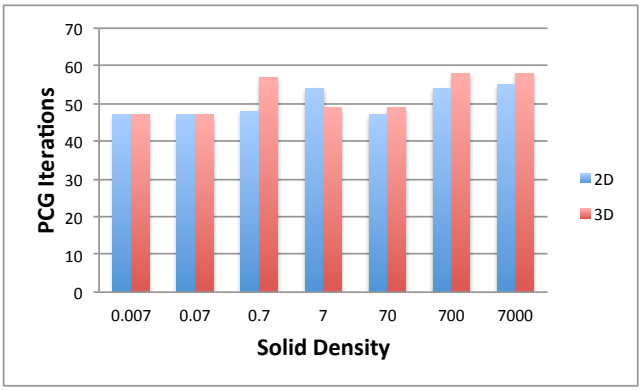


Figure 10: *Comparison of the number of PCG iterations as the solid density is varied for the box lift problem from Figure 5, highlighting the robustness of our method against large density ratios.*

V-cycle *can* be used as the initial guess for the next V-cycle without violating the symmetry of the preconditioner [LMAS16]. Thus, we used the result of the previous V-cycle as initial guess when adding higher order correction terms, obtaining better results. For computing the sparse Cholesky factorization of the solid matrix A_{22} in equation (17), we used the PARDISO library [DCDBK*16]. This factorization can be completely avoided when solving equation (20) for bubbles since $B\beta^{-1}B^T$ is a diagonal matrix, and its entries can be computed per bubble in an assembly-free fashion.

7. Conclusions and Future Work

We designed an efficient solver for two-way coupling rigid bodies with incompressible fluids based on the Schur-complement method, and presented a further application to underwater bubble simulation. Our method allows for the straightforward reuse of fast Multigrid solvers for fluids, is simple to implement and assembly-free (besides the solid equations), allows for large time steps because of the monolithic formulation, and remains stable even when the iterative solver is terminated early. While our implementation was confined to a single CPU workstation, it is conceivable that on *heterogeneous* (multi CPU-GPU) servers where the simulation loop runs on the CPU and the Schur-complement solver is hosted on the GPUs, our preconditioner from equation (10) would become computationally feasible for values of $s > 0$, as the accelerated convergence would hide the cost of offloading the data to the GPUs.

In principle, our formulation can also be used for two-way coupling fluids with volumetric elastic solids. However, this would require factorizing the stiffness matrix K at each time step, making the method less favorable for meshes with a large number of elements. Recently, the *Projective Dynamics* framework has been proposed that circumvents this issue, but requires the use of fixed time steps [BML*14, NOB16, LBK17]. It would be interesting to combine Projective Dynamics with our formulation, while allowing for variable time steps. On a different note, the equations of nonlinear elasticity are also *elliptic*, permitting efficient Multigrid solvers [ZSTB10, MZS*11]. Thus, one could design a generalized Multigrid solver for two-way coupling elastic bodies with incompressible fluids. Finally, it would be interesting to extend our method to thin shells using cut-cell formulations [ABO16].

8. Acknowledgements

The author would like to thank Saket Patkar for helpful discussions, and the anonymous reviewers for their valuable feedback.

References

[ABO16] AZEVEDO V. C., BATTY C., OLIVEIRA M. M.: Preserving geometry and topology for fluid flows with thin obstacles and narrow gaps. *ACM Trans. Graph. 35*, 4 (2016), 1–12. 9

[AGHmD03] ARASH O. E., GÃL'NEVAUX O., HABIBI A., MICHEL DISCHLER J.: Simulating fluid-solid interaction. In *in Graphics Interface* (2003), pp. 31–38. 1

[AGL*17] AANJANEYA M., GAO M., LIU H., BATTY C., SIFAKIS E.: Power diagrams and sparse paged grids for high resolution adaptive liquids. *ACM Trans. Graph. 36*, 4 (2017), 1–12. 5

[AIA*12] AKINCI N., IHMSEN M., AKINCI G., SOLENTHALER B., TESCHNER M.: Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans. Graph. 31*, 4 (2012), 62:1–62:8. 2

[APF13] AANJANEYA M., PATKAR S., FEDKIW R.: A monolithic mass tracking formulation for bubbles in incompressible flow. *Journal of Computational Physics 247* (2013), 17 – 61. 1, 5

[AS99] ADALSTEINSSON D., SETHIAN J.: The fast construction of extension velocities in level set methods. *Journal of Computational Physics 148*, 1 (1999), 2 – 22. 4

[BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. 26*, 3 (2007). 1, 2

[BET14] BENDER J., ERLEBEN K., TRINKLE J.: Interactive simulation of rigid body dynamics in computer graphics. *Computer Graphics Forum 33*, 1 (2014), 246–270. 1

[BML*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.: Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph. 33*, 4 (2014), 1–11. 1, 9

[Bri15] BRIDSON R.: *Fluid simulation for computer graphics, 2nd edition*. A. K. Peters, Ltd., 2015. 1

[CGFO06] CHENTANEZ N., GOKTEKIN T. G., FELDMAN B. E., O'BRIEN J. F.: Simultaneous coupling of fluids and deformable bodies. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), SCA '06, pp. 83–89. 1, 2

[CMT04] CARLSON M., MUCHA P. J., TURK G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. Graph. 23*, 3 (2004), 377–384. 2

[CWSO13] CLAUSEN P., WICKE M., SHEWCHUK J. R., O'BRIEN J. F.: Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Trans. Graph. 32*, 2 (2013), 1–15. 2

[CZY17] CHU J., ZAFAR N. B., YANG X.: A schur complement preconditioner for scalable parallel fluid simulation. *ACM Trans. Graph. 36*, 5 (2017), 1–11. 1, 2, 3

[DCDBK*16] DE CONINCK A., DE BAETS B., KOUROUNIS D., VERBOSIO F., SCHENK O., MAENHOUT S., FOSTIER J.: Needles: Toward large-scale genomic prediction with marker-by-environment interaction. 543–555. 9

[ELF05] ENRIGHT D., LOSASSO F., FEDKIW R.: A fast and accurate semi-Lagrangian particle level set method. *Comput. Struct. 83*, 6-7 (2005), 479–490. 4

[ENGF03] ENRIGHT D., NGUYEN D., GIBOU F., FEDKIW R.: Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *Proc. 4th ASME-JSME Joint Fluids Eng. Conf.* (2003). 4

[EQYF13] ENGLISH R. E., QIU L., YU Y., FEDKIW R.: Chimera grids for water simulation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2013), SCA '13, pp. 85–94. 6

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *Proc. of ACM SIGGRAPH 2001* (2001), pp. 23–30. 2

[FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. *Graph. Models Image Process. 58*, 5 (1996), 471–483. 2

[FOKG05] FELDMAN B., O'BRIEN J., KLINGNER B., GOKTEKIN T.: Fluids in deforming meshes. In *Symposium on Computer Animation* (2005), SCA '05, pp. 255–259. 2

[GB17] GOLDADE R., BATTY C.: Constraint bubbles: Adding efficient zero-density bubbles to incompressible free surface flow. *CoRR abs/1711.11470* (2017). URL: http://arxiv.org/abs/1711.11470. 5

[GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *ACM TOG 22*, 3 (2003), 871–878. 4

[GMS14] GAO M., MITCHELL N., SIFAKIS E.: Steklov-poincaré skinning. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2014), SCA '14, pp. 139–148. 1

[GSLF05] GUENDELMAN E., SELLE A., LOSASSO F., FEDKIW R.: Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph. 24*, 3 (2005), 973–981. 1, 2, 4

[Hea02] HEATH M. T.: *Scientific Computing*, 2 ed. McGraw-Hill, Inc., New York, NY, USA, 2002. 3

[HW65] HARLOW F. H., WELCH J. E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids 8*, 12 (1965), 2182–2189. 2, 4

[JSS*15] JIANG C., SCHROEDER C., SELLE A., TERAN J., STOMAKHIN A.: The affine particle-in-cell method. *ACM Trans. Graph. 34*, 4 (2015), 1–10. 2

[KFCO06] KLINGNER B., FELDMAN B., CHENTANEZ N., O'BRIEN J.: Fluid animation with dynamic meshes. In *Proc. of ACM SIGGRAPH* (2006), SIGGRAPH '06, pp. 820–825. 2

[KGP*16] KLÁR G., GAST T., PRADHANA A., FU C., SCHROEDER C., JIANG C., TERAN J.: Drucker-prager elastoplasticity for sand animation. *ACM Trans. Graph. 35*, 4 (2016), 1–12. 2

[LBK17] LIU T., BOUAZIZ S., KAVAN L.: Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Trans. Graph. 36*, 3 (2017), 1–16. 1, 9

[LFO05] LOSASSO F., FEDKIW R., OSHER S.: Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids 35* (2005), 2006. 4

[LJF16] LU W., JIN N., FEDKIW R.: Two-way coupling of fluids to reduced deformable bodies. In *Symposium on Computer Animation* (2016), SCA '16, pp. 67–76. 2

[LMAS16] LIU H., MITCHELL N., AANJANEYA M., SIFAKIS E.: A scalable schur-complement fluids solver for heterogeneous compute platforms. *ACM Trans. Graph. 35*, 6 (2016), 1–12. 1, 2, 3, 5, 9

[MDS16] MITCHELL N., DOESCHER M., SIFAKIS E.: A macroblock optimization for grid-based nonlinear elasticity. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), SCA '16, pp. 11–19. 1, 3

[MEB*12] MISZTAL M. K., ERLEBEN K., BARGTEIL A., FURSUND J., CHRISTENSEN B. B., BÆRENTZEN J. A., BRIDSON R.: Multiphase flow of immiscible fluids on unstructured moving meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), SCA '12, pp. 97–106. 2

[MHNT15] MAZHAR H., HEYN T., NEGRUT D., TASORA A.: Using nesterov's method to accelerate multibody dynamics with friction and contact. *ACM Trans. Graph. 34*, 3 (2015), 1–14. 2

[MMCK14] MACKLIN M., MÜLLER M., CHENTANEZ N., KIM T.-Y.: Unified particle physics for real-time applications. *ACM Trans. Graph. 33*, 4 (2014), 1–12. 2

[MST10] MCADAMS A., SIFAKIS E., TERAN J.: A parallel multigrid poisson solver for fluids simulation on large grids. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010), SCA '10, pp. 65–74. 1, 2, 3, 4, 6, 8

[MZS*11] MCADAMS A., ZHU Y., SELLE A., EMPEY M., TAMSTORF R., TERAN J., SIFAKIS E.: Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph. 30*, 4 (2011), 1–12. 1, 9

[NOB16] NARAIN R., OVERBY M., BROWN G. E.: ADMM ⊇ Projective Dynamics: Fast simulation of general constitutive models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), SCA '16, pp. 21–28. 1, 9

[QV99] QUARTERONI A., VALLI A.: *Domain decomposition methods for partial differential equations*, vol. 10. Clarendon Press, 1999. 1, 3

[RGJ*15] RAM D., GAST T., JIANG C., SCHROEDER C., STOMAKHIN A., TERAN J., KAVEHPOUR P.: A material point method for viscoelastic fluids, foams and sponges. In *Symposium on Computer Animation* (2015), SCA '15, pp. 157–163. 2

[RMSF11] ROBINSON-MOSHER A., SCHROEDER C., FEDKIW R.: A symmetric positive definite formulation for monolithic fluid structure interaction. *J. Comput. Phys. 230*, 4 (2011), 1547–1566. 1, 2, 4, 5

[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 Trans. Graph. 27*, 3 (2008), 1–9. 1, 2

[RTWT12] RAVEENDRAN K., THUEREY N., WOJTAN C., TURK G.: Controlling liquids using meshes. In *Symposium on Computer Animation* (2012), SCA '12, pp. 255–264. 1

[RZF05] ROBLE D., ZAFAR N. B., FALT H.: Cartesian grid fluid simulation with irregular boundary voxels. In *ACM SIGGRAPH 2005 Sketches* (2005), SIGGRAPH '05. 2

[SABS14] SETALURI R., AANJANEYA M., BAUER S., SIFAKIS E.: Spgrid: A sparse paged grid structure applied to adaptive smoke simulation. *ACM Trans. Graph. 33*, 6 (2014), 1–12. 1

[SSC*13] STOMAKHIN A., SCHROEDER C., CHAI L., TERAN J., SELLE A.: A material point method for snow simulation. *ACM Trans. Graph. 32*, 4 (2013), 1–10. 2

[SSJ*14] STOMAKHIN A., SCHROEDER C., JIANG C., CHAI L., TERAN J., SELLE A.: Augmented mpm for phase-change and varied materials. *ACM Trans. Graph. 33*, 4 (2014), 1–11. 2

[SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid–solid interactions: Research articles. *Comput. Animat. Virtual Worlds 18*, 1 (2007), 69–82. 2

[Sta99] STAM J.: Stable fluids. In *Proc. of ACM SIGGRAPH* (1999), SIGGRAPH '99, pp. 121–128. 2, 4

[TLK16] TENG Y., LEVIN D. I. W., KIM T.: Eulerian solid-fluid coupling. *ACM Trans. Graph. 35*, 6 (2016), 1–8. 1, 2

[TOS01] TROTTENBERG U., OOSTERLEE C. W., SCHULLER A.: *Multigrid*. Academic Press, 2001. 5

[WTF06] WEINSTEIN R., TERAN J., FEDKIW R.: Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Transactions on Visualization and Computer Graphics 12*, 3 (2006), 365–374. 4

[ZB14] ZHANG X., BRIDSON R.: A pppm fast summation method for fluids and beyond. *ACM Trans. Graph. 33*, 6 (2014), 1–11. 1

[ZB17] ZARIFI O., BATTY C.: A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In *Symposium on Computer Animation* (2017), pp. 7:1–7:11. 1, 2

[ZSTB10] ZHU Y., SIFAKIS E., TERAN J., BRANDT A.: An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Trans. Graph. 29*, 2 (2010), 1–18. 1, 9