

A Positive-Definite Cut-Cell Method for Strong Two-Way Coupling Between Fluids and Deformable Bodies

Omar Zarifi
University of Waterloo
ozarifi@uwaterloo.ca

Christopher Batty
University of Waterloo
christopher.batty@uwaterloo.ca

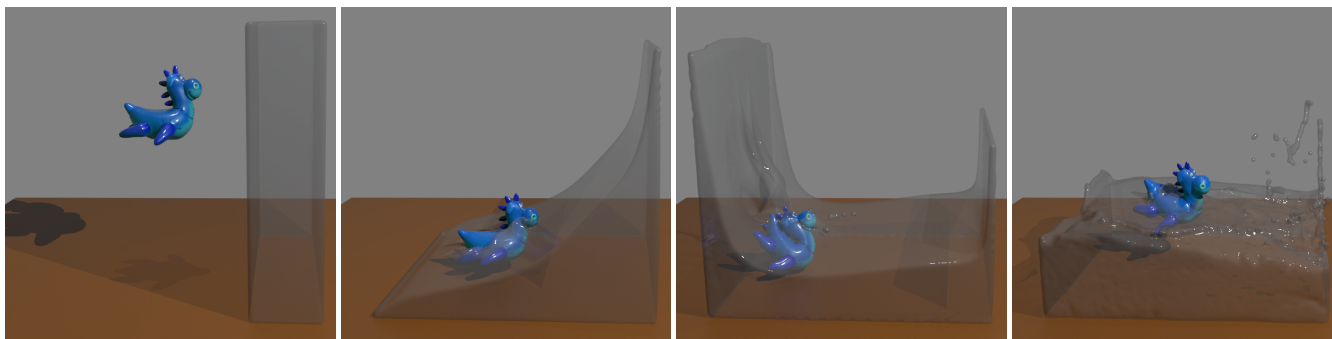


Figure 1: A low density elastic dinosaur toy deforming and interacting with liquid under our cut-cell coupling scheme.

ABSTRACT

We present a new approach to simulation of two-way coupling between inviscid free surface fluids and deformable bodies that exhibits several notable advantages over previous techniques. By fully incorporating the dynamics of the solid into pressure projection, we simultaneously handle fluid incompressibility and solid elasticity and damping. Thanks to this strong coupling, our method does not suffer from instability, even in very taxing scenarios. Furthermore, use of a cut-cell discretization methodology allows us to accurately apply proper free-slip boundary conditions at the exact solid-fluid interface. Consequently, our method is capable of correctly simulating inviscid tangential flow, devoid of grid artefacts or artificial sticking. Lastly, we present an efficient algebraic transformation to convert the indefinite coupled pressure projection system into a positive-definite form. We demonstrate the efficacy of our proposed method by simulating several interesting scenarios, including a light bath toy colliding with a collapsing column of water, liquid being dropped onto a deformable platform, and a partially liquid-filled deformable elastic sphere bouncing.

CCS CONCEPTS

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SCA '17, Los Angeles, CA, USA

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-5091-4/17/07...\$15.00
DOI: 10.1145/3099564.3099572

KEYWORDS

fluid simulation, solid simulation, pressure projection

ACM Reference format:

Omar Zarifi and Christopher Batty. 2017. A Positive-Definite Cut-Cell Method for Strong Two-Way Coupling Between Fluids and Deformable Bodies. In *Proceedings of SCA '17, Los Angeles, CA, USA, July 28-30, 2017*, 11 pages.

DOI: 10.1145/3099564.3099572

1 INTRODUCTION

The interactions between liquids and deformable objects are a familiar element of our daily experience, ranging from children's toys in a swimming pool to flexible kitchen implements in the sink to a rubber duck in the bath. In the present work, we develop a novel method to simulate such interactions between grid-based Eulerian fluids and tetrahedral Lagrangian solids, in a stable, robust, and efficient manner.

While we are not the first to consider this general problem [Chentanez et al. 2006; G enevaux et al. 2003; Guendelman et al. 2005; Robinson-Mosher et al. 2009, 2008], we believe our new method offers several attractive features compared to prior animation techniques. The approach we take is to generalize an increasingly popular family of symmetric positive-definite cut-cell fluid methods [Azevedo et al. 2016; Batty et al. 2007; Bridson 2015; Gibou and Min 2012; Ng et al. 2009; Roble et al. 2005; Weber et al. 2015] to the case of strongly (or monolithically) coupled two-way interactions with deformable solids, in which solid and fluid dynamics are solved *simultaneously*. To reconcile the Eulerian fluid and Lagrangian solid domains, we construct a mutually conforming *cut-cell mesh* at each frame by clipping the cells of the fluid grid against the deformable solid object's geometry. We apply a finite volume discretization over the fluid cells to enforce incompressibility through pressure

projection, and at the same time integrate the velocities of the elastic solid with a standard finite element method. With the cut-cell mesh in hand, we accurately and straightforwardly treat interactions by integrating pressure over the solid's surface to determine the forces from the fluid, while using the solid's surface velocities to dictate velocity boundary conditions at the clipped faces of fluid cells.

This approach avoids grid-induced stairstep artefacts and accurately satisfies the desired tangential free-slip boundary conditions precisely at the solid-fluid boundary, without the need for extra explicit constraints or Lagrange multipliers beyond the existing per-cell fluid pressures. Our strongly-coupled solution of the fluid and solid dynamics, along with our treatment of the solid dynamics with a fully implicit corotational model, endows our method with very good stability properties. We further demonstrate how a simple linear algebra transformation can carry over the symmetric positive-definiteness property of this family of cut-cell methods to our setting. Given the flexibility and increasing prevalence of cut-cell methods, we believe this extension is quite practically relevant (e.g., Bridson's recent book [Bridson 2015] mentioned it as an unsolved challenge), and it further broadens the domain of applicability of these cut-cell methods.

In summary, we present a novel cut-cell method for strongly coupled two-way fluid-deformable interaction that features

- excellent stability;
- boundary condition enforcement at the true interface;
- tangential free-slip without grid artefacts;
- a symmetric positive-definite linear system; and
- no extra constraints or degrees of freedom.

2 RELATED WORK

Coupling Fluids to Rigid Bodies. We begin with a review of techniques for the related problem of rigid body-fluid interaction. Since a typical fluid simulation often involves an enclosing container, the problem of rigid boundaries and objects has been considered from the earliest days of fluid animation. Use of the now-ubiquitous marker-and-cell (MAC) discretization was introduced to the computer graphics community by Foster and Metaxas [1996]. In this seminal paper, grid-aligned solid boundaries were readily handled; however, significant voxelization artefacts unavoidably cropped up in the presence of sloped or curved geometry. Furthermore, only a limited discussion of dynamic rigid bodies was included, with the authors remarking that a more sophisticated method is necessary to properly capture two-way coupling.

Handling of solid boundary conditions, particularly for advection, was subsequently improved by Foster and Fedkiw [2001], Houston et al. [2003], and Rasmussen et al. [2004]. However, issues can still surface due to the voxelized pressure solve. For instance, none of the aforementioned methods can adequately simulate a body of water in hydrostatic equilibrium within a spherical container. Feldman et al. [2005] circumvented this problem by using conforming volumetric meshes in order to accurately capture the fluid region (and, hence, its boundaries), and Klingner et al. [2006] adapted this approach for the purpose of two-way coupling with dynamic rigid bodies. Unfortunately, both of these works relied on the use of conforming unstructured tetrahedral meshes and frequent

remeshing of the fluid domain, adding computational overhead. Another approach to two-way coupling is the "rigid fluid" method proposed by Carlson et al. [2004], which handles interactions by temporarily treating the rigid body as a fluid region, but this approach can lead to severe leakage through objects.

A widely adopted solution to the problem of irregular solids is the use of cut-cell methods, which essentially clip the object geometry against a regular background grid. Their use in computer graphics was first suggested by Roble et al. [2005], who proposed a simple modification to regular pressure projection that allows for more accurate enforcement of static solid boundary conditions. The variational formulation of Batty et al. [2007] similarly treated two-way rigid body interactions by casting the coupled pressure solve in an energy minimization form that accounts for partial cell volumes in three dimensions. Subsequently, Ng et al. [2009] used a finite volume cut-cell discretization to derive a more accurate scheme for kinematically scripted solids, involving the same stencils with a different choice of weighting terms. This thread of research culminated in work by Gibou and Min [2012], who presented further accuracy improvements to the method of Ng et al. [2009] and extended it to two-way coupling with dynamic rigid bodies.

Such cut-cell discretizations have become increasingly common in fluid animation, for example in the context of spatial adaptivity [Batty et al. 2010], multigrid methods [Weber et al. 2015], detailed splashes [Edwards and Bridson 2014], and thin solids [Azevedo et al. 2016]. This trend can be attributed to the simplicity of cut-cells (both conceptual and with respect to implementation) as well as their expressive power: fluxes across small geometrical details can be mathematically captured without refining or re-orienting the computational grid. Observing these properties, the present work makes use of cut-cell discretization to accurately enforce coupling conditions at the interface between a fluid and *deformable* bodies.

Coupling Fluids to Lagrangian Deformables. The fluid nature of liquids and gases results in immense shape changes throughout the course of a simulation. Consequently, despite various advantages possessed by Lagrangian mesh-based approaches to fluid simulation [Clausen et al. 2013; Misztal et al. 2014], such as a unified physical model for solid and fluids, better volume preservation, and straightforward support for implicit surface tension, they are less common because they necessitate continuous and expensive remeshing of the fluid region. Many solids, on the other hand, are not ordinarily subject to such extreme distortion and permanent deformation. As a result, Lagrangian mesh methods have predominantly been used to animate elastic deformable objects. A typical example of such a scheme is given by Teran et al. [2003]: the solid domain is partitioned into volumetric elements, allowing for a simple calculation of per-element strains, which can be used to obtain material stresses and internal forces.

This discrepancy in representation of fluids and solids introduces a challenge in their dynamic coupling: because velocity samples for the two entities are not collocated, it becomes harder to accurately enforce boundary conditions at the regions of contact. To bypass the inherent difficulty in simultaneous satisfaction of boundary conditions, early attempts at animating solid-fluid interactions took a weakly coupled approach that alternates solving for solid and fluid

motions. For example, Genevaux et al. [2003] devised a simple simulation scheme that integrates fluid and solid dynamics separately, with interactions facilitated by interfacial forces. Guendelman et al. [2005] take a similar approach, but use solid velocities as prescribed boundary conditions for (voxelized) pressure projection. Effects of the fluid on the solid are then captured through application of forces induced by the computed pressure field. We observe that the simplicity of weakly coupled schemes generally comes at the cost of accuracy, stability, or efficiency: it is often necessary to take restrictively small time steps in order to prevent unphysical behavior or even catastrophic simulation failure.

The problem of instability was largely mitigated by Chentanez et al. [2006], whose implicit, strongly coupled formulation is demonstrably robust, even with relatively large time steps. The tetrahedral mesh variant of this method naturally requires continuous remeshing, while the regular grid variant does not correctly apply the free-slip condition at the exact interface, since it inaccurately treats the solid as voxelized; as a result, visible voxelization artefacts appear at the solid-fluid boundary. Furthermore, the linear system for pressure projection is non-symmetric, rendering it more difficult to solve numerically. The formulation of Robinson-Mosher et al. [2008], on the other hand, instead requires the solution of a symmetric indefinite matrix. While such systems are generally easier to solve numerically than non-symmetric ones, working with positive-definite matrices is still preferred. The effects of the underlying voxelization are present in this method as well; in particular, lumping of fluid and solid momenta within computational cells forces the fluid to inherit the solid's velocity at the boundaries. In essence, despite the use of an inviscid fluid model, this method violates the desired free-slip condition, causing the fluid to stick unnaturally to objects.

In subsequent follow-up work, Robinson-Mosher et al. [2009] incorporated tangential free-slip boundary conditions into their coupled pressure projection scheme via Lagrange multipliers that explicitly constrain only the interpolated *normal* component of relative velocity to be zero. However, these boundary conditions are once again applied at regular grid faces as opposed to the true contact interface, and the resultant linear system remains symmetric indefinite. A positive-definite formulation of the same problem was finally presented by Robinson-Mosher et al. [2011] through linear algebraic manipulations that rely on construction of a symmetric factorization of the solid's damping matrix; our method is more general in avoiding this requirement. In addition to our use of a cut-cell approach that better accounts for solid geometry, the present work is also distinguished from the preceding three approaches by our use of corotational linear elasticity which allows us to incorporate the full implicit solid dynamics into the coupled pressure projection; Robinson-Mosher et al. [2011] include only the solid damping.

An important determinant of computational cost of a deformable simulation is the node count for the mesh. In certain cases, it is not possible to coarsen a mesh without significantly affecting its shape. In these scenarios, one can still improve running time by utilizing a reduced model, whose deformations are restricted to a smaller subset (e.g., [Sheth et al. 2015]). Building on the aforementioned algorithm of Robinson-Mosher et al. [2011], Lu et al. [2016]

developed a method to capture interactions of fluids and reduced deformable bodies. Since our goal is high-fidelity simulation with accurate boundary condition handling, we do not consider reduced models; however, we anticipate that our coupling strategy could similarly be adapted to work with reduced models.

Sotiropoulos and Yang [2014] provide a review of a wide variety of approaches to fluid-structure interaction in computational physics. While various flavors of immersed boundary and cut-cell methodologies are common, we are not aware of any method that offers the specific advantages provided by the present work. As an example, Pasquariello et al. [2016] use a cut-cell finite difference/volume strategy in a manner broadly similar to our work and that of Ng et al. [2009]. In contrast to our work though, they consider compressible flow, handle the boundary interactions via a mortar method based on explicit Lagrange multiplier constraints, and employ a weak/partitioned coupling strategy.

Coupling Fluids to Eulerian Deformables. Although still not as popular as the ubiquitous Lagrangian schemes, Eulerian and hybrid solid simulation approaches have recently gained traction; the Eulerian solids approach [Levin et al. 2011] and the material point method [Stomakhin et al. 2013] are representative examples. Most relevant to our research is the Eulerian two-way solid-fluid coupling method of Teng et al. [2016]. While a fully Eulerian scheme is in many ways very natural, this method exhibits several practical drawbacks. First, the underlying Eulerian solid model suffers from strong numerical damping. More importantly, this unified model necessarily implies a no-slip condition at the boundaries, and the described method is limited to simulating fully immersed, strictly incompressible objects. By contrast, we are interested in coupling of potentially compressible solids to inviscid free surface fluids, for which the free-slip condition yields more realistic results for animation scenarios.

SPH Methods. Smoothed particle hydrodynamics (SPH) methods provide an alternate way to simulate fluids, by distributing mass to particles and using interactions between particles to determine the dynamics of the aggregate fluid. Several algorithms to simulate two-way coupling of deformable solids to SPH fluids have been previously proposed (e.g., [Akinci et al. 2013; Solenthaler et al. 2007]). These methods are outside the scope of the present work, which focuses on Eulerian fluid representations.

3 FLUID EQUATIONS

We simulate the liquid by solving the incompressible Euler equations in an Eulerian fashion, using

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{\nabla p}{\rho} &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \quad (1)$$

where \mathbf{u} is the fluid velocity field, p is pressure, ρ is the fluid density, and \mathbf{f} denotes external forces [Batchelor 2000]. At the interface between air and liquid, the free surface condition, $p = 0$, must be satisfied. For solid objects in contact with the fluid, we enforce the free-slip boundary condition: $\mathbf{u} \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n}$, where \mathbf{v} is the solid's velocity, and \mathbf{n} is the surface normal. For (near-)inviscid fluids at typical resolutions and scales for animation applications,

this condition is preferable, as it avoids the undesirable tangential sticking effects produced by the no-slip condition ($\mathbf{u} = \mathbf{v}$).

We take an operator-splitting approach to solve the fluid equations [Stam 1999]. First, the fluid's velocity field is advected. Our method is agnostic with respect to the advection scheme; due to its ease of implementation, efficiency, and excellent stability, we opted for a standard semi-Lagrangian technique [Stam 1999]. Next, the external forces are integrated explicitly to acquire an intermediate velocity field \mathbf{u}^* . The final step is to perform pressure projection, which involves finding a pressure field to apply to the fluid such that the updated velocities

$$\mathbf{u} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p \quad (2)$$

are divergence-free, $\nabla \cdot \mathbf{u} = 0$, and satisfy the appropriate boundary conditions. Fluid-solid interaction will be mediated through our coupled projection scheme that simultaneously integrates fluid pressure forces and both elasticity and damping forces in the deformable solid. We delay discussion of the fluid's spatial discretization and our solid-fluid coupling strategy to section 5.

4 SOLID EQUATIONS

We assume a linear elastic material model for the deformable solids, leading to equations of motion of the form

$$M\ddot{\mathbf{x}} + D\dot{\mathbf{x}} + K(\mathbf{x} - \mathbf{x}_0) = \mathbf{f}, \quad (3)$$

where M , D , and K are solid's mass, damping, and stiffness matrices, respectively, \mathbf{x} is the stacked vector of node coordinates (and overdots represent time-derivatives), \mathbf{x}_0 is the vector of node coordinates in a rest configuration, and \mathbf{f} is the vector of external forces [Terzopoulos et al. 1987]. This system of ordinary differential equations is integrated on a tetrahedral mesh of the desired solid. We assume M is a diagonal lumped mass matrix and D is a Rayleigh damping matrix given by

$$D = \tau_m M + \tau_k K, \quad (4)$$

for non-negative constants τ_m , τ_k . Because it provides rotational invariance and requires only a linear solve per step, we compute the global stiffness matrix using the corotational finite element model described by Müller and Gross [2004]. (Note that this standard corotational formulation assumes fixed element rotations so it requires only a single linear solve per time step; an exact approach can account for changing rotations, but makes the problem nonlinear [Barbič 2012].)

The dynamics equations are discretized in time using backwards Euler and expressed in terms of nodal velocities, \mathbf{v} :

$$M \left(\frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} \right) + D\mathbf{v}^{n+1} + K(\mathbf{x}^n + \Delta t\mathbf{v}^{n+1} - \mathbf{x}_0) = \mathbf{f}, \quad (5)$$

where the superscripts are used to identify the time step. Upon rearranging, we arrive at the linear system that governs evolution of the deformable body's nodal velocities:

$$\left(\frac{1}{\Delta t} M + D + \Delta t K \right) \mathbf{v}^{n+1} = \frac{1}{\Delta t} M \mathbf{v}^n - K(\mathbf{x}^n - \mathbf{x}_0) + \mathbf{f}. \quad (6)$$

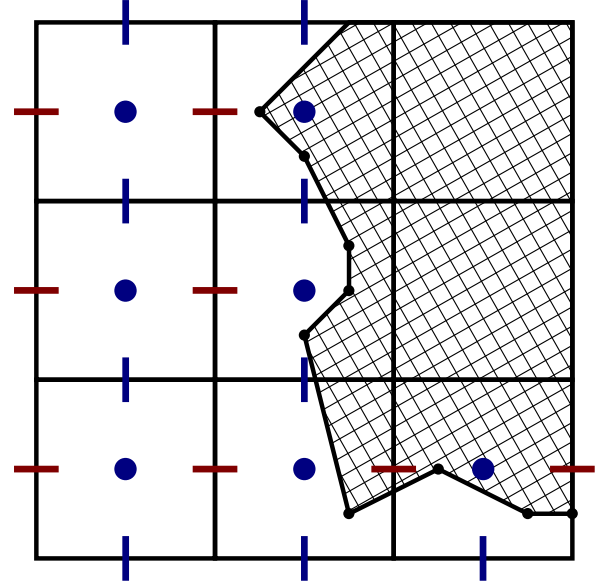


Figure 2: A sample MAC grid configuration. The solid region is shaded, and black dots denote nodes of its mesh. Fluid pressures are at cell centers (blue dots), while fluid velocity components are on cell faces (red dashes for x components, blue dashes for y components). Only locations of valid pressure and velocity samples are marked.

5 PRESSURE PROJECTION

We now proceed to describe our coupled pressure projection that simultaneously enforces fluid incompressibility and integrates the solid's internal forces, allowing for accurate enforcement of boundary conditions at the solid-fluid interface.

5.1 Fluid Incompressibility

Throughout the fluid domain, we place velocity components on cell face midpoints and pressures at cell centers, in accordance with the standard marker-and-cell (MAC) method [Foster and Metaxas 1996] (see Figure 2 for a sample configuration in two dimensions). This staggered sampling allows for straightforward central finite difference approximations of the key differential operators (gradient and divergence), and avoids odd-even pressure decoupling artefacts of collocated schemes.

Following Ng et al. [2009], we employ the finite volume method to enforce incompressibility. Consider the fluid region, denoted Ω , contained within a cubic computational cell (see Figure 3); incompressibility in conjunction with the velocity update (2) imply

$$\begin{aligned} 0 &= \int_{\Omega} \nabla \cdot \mathbf{u} dV = \int_{\partial\Omega} \mathbf{u} d\mathbf{n} \\ &= \int_{\partial\Omega_F} \left(\mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p \right) d\mathbf{n} + \int_{\partial\Omega_S} \mathbf{v} d\mathbf{n}, \end{aligned} \quad (7)$$

where the divergence theorem was used to convert the expression into a surface integral. In the last step, we partitioned the boundary into the region in contact with the solid, $\partial\Omega_S$, and the

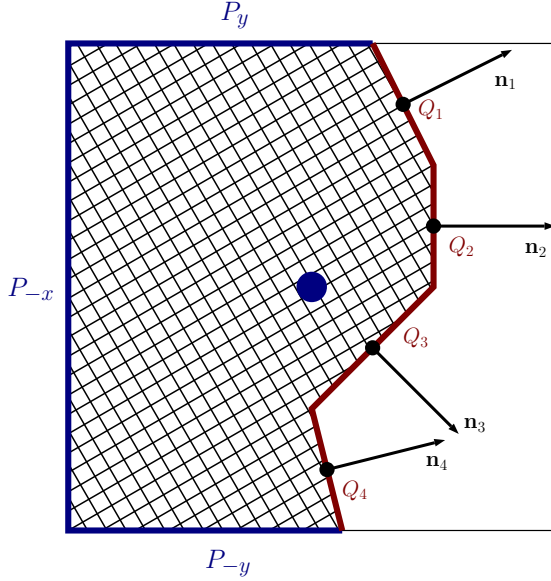


Figure 3: A sample configuration within a cell. The fluid region (Ω) is shaded, its boundary in contact with neighboring fluid cells ($\partial\Omega_F$) is given in blue, and the region in contact with the solid ($\partial\Omega_S$) is given in red. In this case, we have $\partial\Omega_F = P_x \cup P_y \cup P_z$ and $\partial\Omega_S = Q_1 \cup Q_2 \cup Q_3 \cup Q_4$. Black dots represent centroids of regions Q_i , and the inward-oriented normal for each region is also depicted. The pressure sample is located at the central blue dot. This diagram corresponds to the central cell in Figure 2.

remainder, $\partial\Omega_F$, and used the fact that $\mathbf{u} \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n}$ on $\partial\Omega_S$. To simplify our derivation, we ignore the presence of a free surface (this boundary condition is handled separately with the ghost fluid method [Enright et al. 2003]). It follows that the non-solid portion of the boundary is a union of axis-aligned planar regions, which are parts of the cell walls neighboring other fluid cells: $\partial\Omega_F = P_x \cup P_{-x} \cup P_y \cup P_{-y} \cup P_z \cup P_{-z}$, where subscripts are used to denote outward-pointing normals. Consequently, we can approximate the flux across $\partial\Omega_F$ as follows:

$$\int_{\partial\Omega_F} \mathbf{u}^* \cdot d\mathbf{n} \approx \sum_{\alpha \in \{x, y, z\}} (A(P_\alpha) u^*(P_\alpha) - A(P_{-\alpha}) u^*(P_{-\alpha})). \quad (8)$$

In the above equation, $A(P)$ is the area of region P , and $u^*(P)$ is the normal component of the intermediate fluid velocity at the center of the face that contains P ; by construction, this velocity value can be directly read from the MAC grid. Likewise, we have

$$\int_{\partial\Omega_F} \nabla p \cdot d\mathbf{n} \approx \sum_{\alpha \in \{x, y, z\}} \left(A(P_\alpha) \frac{\partial p}{\partial \alpha}(P_\alpha) + A(P_{-\alpha}) \frac{\partial p}{\partial \alpha}(P_{-\alpha}) \right), \quad (9)$$

where the partial derivatives at cell face centers can be estimated with finite differences between the cell-centered pressures.

In a similar vein, we assume the solid's surface is represented by a triangle mesh and partition $\partial\Omega_S$ into planar polygonal regions

Q_1, \dots, Q_k . (These regions are not necessarily triangular, since they result from clipping mesh triangles against the cubic fluid cells). Letting a_i and \mathbf{n}_i be the area and unit normal of region Q_i (pointing out of the fluid), and \mathbf{c}_i be the centroid of Q_i , we can use the following approximation:

$$\int_{\partial\Omega_S} \mathbf{v} \cdot d\mathbf{n} \approx \sum_{i=1}^k a_i \mathbf{v}(\mathbf{c}_i) \cdot \mathbf{n}_i. \quad (10)$$

To estimate the velocity at the centroid, $\mathbf{v}(\mathbf{c}_i)$, linear interpolation is used. Specifically, note that Q_i is contained within a surface triangle of the solid, and let the nodes of this triangle be numbered $i(1), i(2), i(3)$. If \mathbf{x}_j is the location of node j , then we can express \mathbf{c}_i in terms of its barycentric coordinates:

$$\mathbf{c}_i = \alpha \mathbf{x}_{i(1)} + \beta \mathbf{x}_{i(2)} + (1 - \alpha - \beta) \mathbf{x}_{i(3)}. \quad (11)$$

Finally, this allows us to approximate the velocity at \mathbf{c}_i using

$$\mathbf{v}(\mathbf{c}_i) = \alpha \mathbf{v}_{i(1)} + \beta \mathbf{v}_{i(2)} + (1 - \alpha - \beta) \mathbf{v}_{i(3)}, \quad (12)$$

where \mathbf{v}_j is the velocity of node j .

We now use \mathbf{u}^* to denote the stacked vector of all intermediate velocities. Then the stacked vector of boundary integrals (8) can be written as $-G^T \mathbf{A} \mathbf{u}^*$, where A is the diagonal matrix of face areas, and G is the discrete gradient matrix. Likewise, the stacked vector of integrals (9) can be expressed as $-G^T A G \mathbf{p}$, where \mathbf{p} is the vector of pressure samples at cell centers. Lastly, the integral (10) can be combined in vector form as $N^T C \mathbf{v}$, where N is the matrix containing area-weighted normals (oriented into the solid), C is the matrix of barycentric coordinates, and \mathbf{v} is the stacked vector of nodal velocities. Substituting these expressions into (7) yields the discrete equations for the fluid volume,

$$\mathbf{0} = -G^T \mathbf{A} \mathbf{u}^* + \frac{\Delta t}{\rho} G^T A G \mathbf{p} + N^T C \mathbf{v}. \quad (13)$$

5.2 Solid Dynamics

On the solid side, (6) already captures the internal dynamics; we need only add the effects of fluid pressure, by incorporating them into the vector of external forces.

Consider a planar patch P acquired by intersecting one of the solid's surface triangles T with a fluid cell; we will use p to denote fluid pressure at this cell's center. We make the simplifying assumption that pressure is constant within the computational cell; though this may incur some minor additional error, it is a common and effective choice in graphics applications, e.g., [Batty et al. 2007; Bridson 2015; Yngve et al. 2000]). The force applied by the liquid pressure on P is then given by $\mathbf{f} = p A(P) \mathbf{n}$, where $A(P)$ and \mathbf{n} are the area and unit normal of P (oriented into the solid). We further assume that \mathbf{f} is applied at \mathbf{c} , the centroid of the patch P , and distribute this force among the nodes of T . Specifically, if the nodes of T are numbered 1, 2, 3, and the barycentric representation of \mathbf{c} within this triangle is

$$\mathbf{c} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \alpha_3 \mathbf{x}_3, \quad (14)$$

then the force distributed to node i is

$$\mathbf{f}_i = \alpha_i \mathbf{f} = \alpha_i p A(P) \mathbf{n}. \quad (15)$$

Stacking all nodal pressure forces into a vector, we find that it can be expressed as $C^T N \mathbf{p}$. As such, (6) with pressure forces factored

in reads

$$\left(\frac{1}{\Delta t}M + D + \Delta tK\right)\mathbf{v} = \frac{1}{\Delta t}M\mathbf{v}^* - K(\mathbf{x} - \mathbf{x}_0) + C^T N\mathbf{p}, \quad (16)$$

where \mathbf{v}^* are solid nodal velocities prior to pressure projection.

5.3 Coupled System

Given the solid and fluid discretizations, we can proceed to assemble our combined linear system. We note that (13) can be rearranged into

$$\frac{\Delta t}{\rho}G^T AG\mathbf{p} + N^T C\mathbf{v} = G^T Au^*. \quad (17)$$

Likewise, (16) can be equivalently written as

$$\left(\frac{1}{\Delta t}M + D + \Delta tK\right)\mathbf{v} - C^T N\mathbf{p} = \frac{1}{\Delta t}M\mathbf{v}^* - K(\mathbf{x} - \mathbf{x}_0). \quad (18)$$

Stacking these last two equations gives the linear system for our novel coupled pressure projection scheme

$$\begin{pmatrix} \frac{\Delta t}{\rho}G^T AG & N^T C \\ C^T N & -\left(\frac{1}{\Delta t}M + D + \Delta tK\right) \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} G^T Au^* \\ -\frac{1}{\Delta t}M\mathbf{v}^* + K(\mathbf{x} - \mathbf{x}_0) \end{pmatrix} \quad (19)$$

which we solve for fluid pressures and solid velocities. This allows us to simultaneously account for fluid incompressibility, solid internal forces, and the two-way interactions between these two entities. (Note that the solid equations were negated to achieve symmetry.)

Following solution of the above system, the acquired pressure vector is used to update the fluid velocities to be divergence-free as

$$\mathbf{u} = \mathbf{u}^* - \frac{\Delta t}{\rho}G\mathbf{p}. \quad (20)$$

6 POSITIVE-DEFINITE FORMULATION

The matrix associated with the linear system (16) for the corotational solid dynamics is positive-definite, a fact that has been exploited in prior work (e.g. [Hecht et al. 2012]). While the negation of the solid equations above allows us to obtain a symmetric system, it also causes the principal sub-block corresponding to solid velocities to be negative-definite. Since $(\Delta t/\rho)G^T AG$ is positive-definite, the overall coefficient matrix for our pressure projection solve (19) is only symmetric indefinite. Although various methods tailored towards such equations exist (e.g., MINRES or QMR, as discussed by Robinson-Mosher et al. [2009; 2008]), it is generally more desirable to work with symmetric positive-definite (SPD) systems. Fortunately, the symmetric coefficient matrix in (19) is endowed with a special structure: the principal submatrix corresponding to pressures is positive-definite, and the principal submatrix for solid velocities is negative-definite and constructed as a weighted sum of a stiffness matrix and a *diagonal* mass matrix. We exploit these facts in order to efficiently convert the system into SPD form; our method to do so is described below.

We begin by expressing the (negative) principal velocity submatrix as a sum of arbitrary symmetric positive-definite matrices Z_1 and Z_2 :

$$\frac{1}{\Delta t}M + D + \Delta tK = Z_1 + Z_2. \quad (21)$$

Then the second set of equations in (19) reads

$$C^T N\mathbf{p} - Z_1\mathbf{v} - Z_2\mathbf{v} = \mathbf{w}, \quad (22)$$

where we let \mathbf{w} denote the right-hand side for convenience. This equation is equivalent to

$$\mathbf{v} = Z_1^{-1}(C^T N\mathbf{p} - Z_2\mathbf{v} - \mathbf{w}). \quad (23)$$

Substituting this expression for \mathbf{v} into the first set of equations from (19) yields

$$\frac{\Delta t}{\rho}G^T AG\mathbf{p} + N^T CZ_1^{-1}(C^T N\mathbf{p} - Z_2\mathbf{v} - \mathbf{w}) = G^T Au^*, \quad (24)$$

which can be rearranged into

$$\begin{aligned} \left(\frac{\Delta t}{\rho}G^T AG + N^T CZ_1^{-1}C^T N\right)\mathbf{p} - N^T CZ_1^{-1}Z_2\mathbf{v} \\ = G^T Au^* + N^T CZ_1^{-1}\mathbf{w}. \end{aligned} \quad (25)$$

Also, note that we can use (23) to derive

$$\mathbf{0} = Z_2\mathbf{v} - Z_2\mathbf{v} = Z_2\mathbf{v} - Z_2Z_1^{-1}(C^T N\mathbf{p} - Z_2\mathbf{v} - \mathbf{w}), \quad (26)$$

which is equivalent to

$$-Z_2Z_1^{-1}C^T N\mathbf{p} + (Z_2 + Z_2Z_1^{-1}Z_2)\mathbf{v} = -Z_2Z_1^{-1}\mathbf{w}. \quad (27)$$

Finally, combining (25) and (27) gives the following SPD system:

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} G^T Au^* + N^T CZ_1^{-1}\mathbf{w} \\ -Z_2Z_1^{-1}\mathbf{w} \end{pmatrix}, \quad (28)$$

where $\mathbf{w} = -1/\Delta tM\mathbf{v}^* + K(\mathbf{x} - \mathbf{x}_0)$ and

$$\begin{aligned} A_{11} &= \frac{\Delta t}{\rho}G^T AG + N^T CZ_1^{-1}C^T N, \\ A_{12} &= -N^T CZ_1^{-1}Z_2, \\ A_{21} &= -Z_2Z_1^{-1}C^T N = A_{12}^T, \\ A_{22} &= Z_2 + Z_2Z_1^{-1}Z_2. \end{aligned} \quad (29)$$

Note that Z_1^{-1} appears extensively in (28); as such, it is ideal to choose Z_1 to be a diagonal matrix. Upon experimenting, we found that letting Z_1 be a multiple of the solid mass matrix provides good results. Considering our use of Rayleigh damping and the constraint that Z_2 be SPD, it is reasonable to set $Z_1 = \sigma(1/\Delta t + \tau_m)M$ (where τ_m is the mass damping coefficient from (4)), for $\sigma \in (0, 1)$; we exclusively used $\sigma = 0.9$ to generate our simulations without encountering numerical difficulties. This construction implies

$$\begin{aligned} Z_2 &= \frac{1}{\Delta t}M + D + \Delta tK - Z_1 \\ &= \left(\frac{1}{\Delta t} + \tau_m\right)M + (\Delta t + \tau_k)K - \sigma\left(\frac{1}{\Delta t} + \tau_m\right)M \\ &= (1 - \sigma)\left(\frac{1}{\Delta t} + \tau_m\right)M + (\Delta t + \tau_k)K. \end{aligned} \quad (30)$$

It is fairly straightforward to establish that the transformed system (28) is indeed SPD. To this end, take vectors \mathbf{p}, \mathbf{v} (at least one of which is non-zero). Because $(\Delta t/\rho)G^T AG$ and Z_2 are both SPD, it can be concluded that

$$\mathbf{p}^T \left(\frac{\Delta t}{\rho}G^T AG\right)\mathbf{p} + \mathbf{v}^T Z_2\mathbf{v} > 0. \quad (31)$$

Also, Z_1 is SPD, so its inverse (which is necessarily SPD as well) can be written as $Z_1^{-1} = Y^T Y$ for some matrix Y . Use of these facts allows us to derive

$$\begin{aligned} & \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \end{pmatrix} \\ &= \mathbf{p}^T \left(\frac{\Delta t}{\rho} G^T A G \right) \mathbf{p} + \mathbf{p}^T N^T C Z_1^{-1} C^T N \mathbf{p} \\ & - 2\mathbf{p} N^T C Z_1^{-1} Z_2 \mathbf{v} + \mathbf{v}^T Z_2 \mathbf{v} + \mathbf{v}^T Z_2 Z_1^{-1} Z_2 \mathbf{v} \\ & > \mathbf{p}^T N^T C Y^T Y C^T N \mathbf{p} - 2\mathbf{p} N^T C Y^T Y Z_2 \mathbf{v} \\ & + \mathbf{v}^T Z_2 Y^T Y Z_2 \mathbf{v} = \|Y C^T N \mathbf{p} - Y Z_2 \mathbf{v}\|^2 \geq 0, \end{aligned} \quad (32)$$

as required.

Lastly, we note that the system (28) may suffer from poor conditioning due to different element magnitudes in the solid and fluid blocks. We resolved this issue through rescaling; that is, we instead solve the modified system

$$\begin{pmatrix} A_{11} & \alpha A_{12} \\ \alpha A_{21} & \alpha^2 A_{22} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \frac{1}{\alpha} \mathbf{v} \end{pmatrix} = \begin{pmatrix} G^T A \mathbf{u}^* + N^T C Z_1^{-1} \mathbf{w} \\ -\alpha Z_2 Z_1^{-1} \mathbf{w} \end{pmatrix}. \quad (33)$$

Theoretically, any non-zero scaling factor α can be plugged into the above system; however, certain values of this parameter may in fact be detrimental. Using the termination condition for iterative solvers as a guide, we propose a simple heuristic for choosing α automatically. In particular, for a linear system $A\mathbf{x} = \mathbf{b}$, an iterative solver refines the current solution until the relative residual, $\|A\mathbf{x} - \mathbf{b}\|/\|\mathbf{b}\|$, falls below some specified threshold. Writing the two parts of (28) as $A_1\mathbf{x} = \mathbf{b}_1$ and $A_2\mathbf{x} = \mathbf{b}_2$, we can prevent either \mathbf{b}_1 or \mathbf{b}_2 from dominating the denominator of residual error by scaling them to be of roughly equal norm; this can be achieved by setting $\alpha = \|\mathbf{b}_1\|/\|\mathbf{b}_2\|$. Choosing α in this manner ensures that both fluid and solid dynamics equations are solved to comparable accuracy by an iterative scheme. Alternatively, it is possible to use a static value of α throughout a simulation. For instance, we found that $\alpha = 0.001$ worked quite well and used it exclusively for all our experiments. However, this setting may need modification if quite different material parameters are used.

While this transformation allows us to acquire an SPD system, we have not yet considered whether doing so might significantly deteriorate sparsity and conditioning as a side-effect. Section 8 describes the results of several numerical experiments designed to assess these properties. To summarize our findings, the transformation provides us with an SPD system at a permissibly modest cost to sparsity and conditioning.

7 ALGORITHM OVERVIEW AND IMPLEMENTATION DETAILS

This section describes the stages of a single simulation step, in the order they occur, as outlined in Algorithm 1.

Solid Position Update. First, the solid nodes are moved with their current velocities via forward Euler:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^n. \quad (34)$$

Once the nodal positions are updated, the invalidated corotational stiffness matrix must be reassembled by updating all elemental rotations (see [Müller and Gross 2004] for details).

Algorithm 1 Algorithm Overview

- 1: **while** Simulating **do**
 - 2: Update solid positions
 - 3: Advect fluid particles and velocities
 - 4: Apply external forces
 - 5: Compute distance fields and cut-cells
 - 6: Perform coupled pressure projection
 - 7: Extrapolate fluid velocities
 - 8: **end while**
-

Fluid Advection. Fluid advection is comprised of two parts: particle advection and velocity advection. We use simple marker particles to track the fluid volume, so the former evolves the geometry of the fluid, whereas the latter is called for by the governing dynamics equations. Particle advection is performed according to the grid velocities using a second-order explicit Runge-Kutta method with trilinear interpolation. A basic semi-Lagrangian method is used to advect the velocity field, with the same explicit integration scheme to trace a sample's trajectory backwards in time.

Application of Forces. Next, external forces are applied to both the solid and the fluid. We considered only a constant gravitational force, integrated using forward Euler.

Distance Field Computations. The fourth stage of a simulation step involves various geometric operations that aid the subsequent pressure projection solve. First, the liquid distance field is sampled at cell centers of the MAC grid, using a simple union-of-spheres approximation of the surface based on the particles. This step is followed by calculation of the solid distance field, sampled at nodes of the MAC grid. Lastly, the cut-cells are formed by intersecting every solid surface triangle with the computational grid. For each planar surface region intersecting a cell, we save the area, unit normal (oriented into the solid), and barycentric representation of its centroid.

Pressure Projection. The signed distance fields are used to first identify active pressure cells (a cell is marked active if it contains some fluid and its center is inside either the fluid or solid regions). For simplicity, solid MAC grid face areas are calculated with the help of the solid's signed distance field (a more geometrically faithful approach would compute these based on the exact cut-cell geometry, although we encountered no issues from this mild inconsistency). To avoid conditioning problems, face area fractions below 0.01 are clamped down to 0, and fractions above 0.99 are rounded up to 1, as is common practice [Bridson 2015]. At this point, the SPD linear system (33) is assembled and solved for fluid pressures and solid velocities. Finally, the pressure values are used to update the fluid velocities, in accordance with (20).

Our CPU-based C++ implementation uses Eigen [Guennebaud et al. 2010] to handle linear algebra routines and OpenMP to facilitate simple loop parallelization throughout the code. The SPD system is iteratively solved with the conjugate gradient method aided by an incomplete Cholesky preconditioner, with convergence tolerance on the relative residual set to 10^{-10} .

Velocity Post-Processing. Successive advection steps will inevitably query the fluid velocity field outside of the valid region computed

Table 1: Table of simulation resolutions.

Simulation	Grid Size	Solid	
		Nodes	Elements
Dam Break	$100 \times 100 \times 100$	725	2688
Buoyancy	$100 \times 100 \times 100$	481	2000
Compressibility	$100 \times 100 \times 100$	343	1296
Thin Solid 1	$100 \times 100 \times 100$	882	2400
Thin Solid 2	$64 \times 64 \times 64$	2450	6936
Fluid in Sphere	$100 \times 100 \times 100$	992	2948
Light Dino	$100 \times 100 \times 100$	1424	5920
Stability	$35 \times 35 \times 35$	189	480

during pressure projection; we use a simple constrained velocity extrapolation approach to assign fluid grid velocities to these regions [Houston et al. 2003; Rasmussen et al. 2004]. Due to truncation error, fluid particles may still occasionally enter the solid and must be projected back to the solid surface, but the aforementioned modification reduces such occurrences.

8 RESULTS

We conducted several experiments to test and demonstrate our scheme’s efficacy; the interested reader is referred to our supplemental video for a rendered collection of our test scenarios. All visualizations were obtained by rendering our simulation data in Houdini [Side Effects Software 2017]. Table 1 contains simulation resolutions for the described tests. Excluding the stability or conditioning assessments discussed later, the average time per experiment to carry out a single step of simulation ranged between 18 and 59 seconds; of that time, between 3.5 and 39 seconds were spent performing pressure projection.¹ In order to minimize the likelihood of tunneling of liquid particles through solids from overly large timesteps relative to the fluid velocity, we made use of substepping within frames, leading to approximately 4.42 simulation steps per animation frame on average.

Dam Break. Our first scenario consists of an initially vertical column of liquid interacting with a dangling deformable beam pinned at its top. Upon collapsing under gravity, the liquid collides with the beam at which point two-way coupling effects can be witnessed: the fluid’s motion is deflected by the solid, and the beam deforms as it is subjected to the accelerating liquid’s momentum. After the initial violent interactions, the solid and the liquid settle into a calmer rhythm, gently swaying in tandem.

Buoyancy Example. To demonstrate that our coupling method correctly applies buoyancy forces, we conducted drop experiments with solid balls of different densities. The lightest solid is realistically propelled out of the liquid body, while the heaviest one sinks rapidly to the bottom of the pool. The neutrally buoyant ball (i.e., with rest density equal to the fluid) can be observed to initially maintain its path of motion, then more passively flow with the fluid, and finally sink to the bottom as compression induced by liquid pressure gives it a slightly higher density.

¹Simulations were performed on a Linux machine (running Ubuntu 16.04) equipped with 16 GB of RAM and AMD FX-8370E clocked at 3.3 GHz.

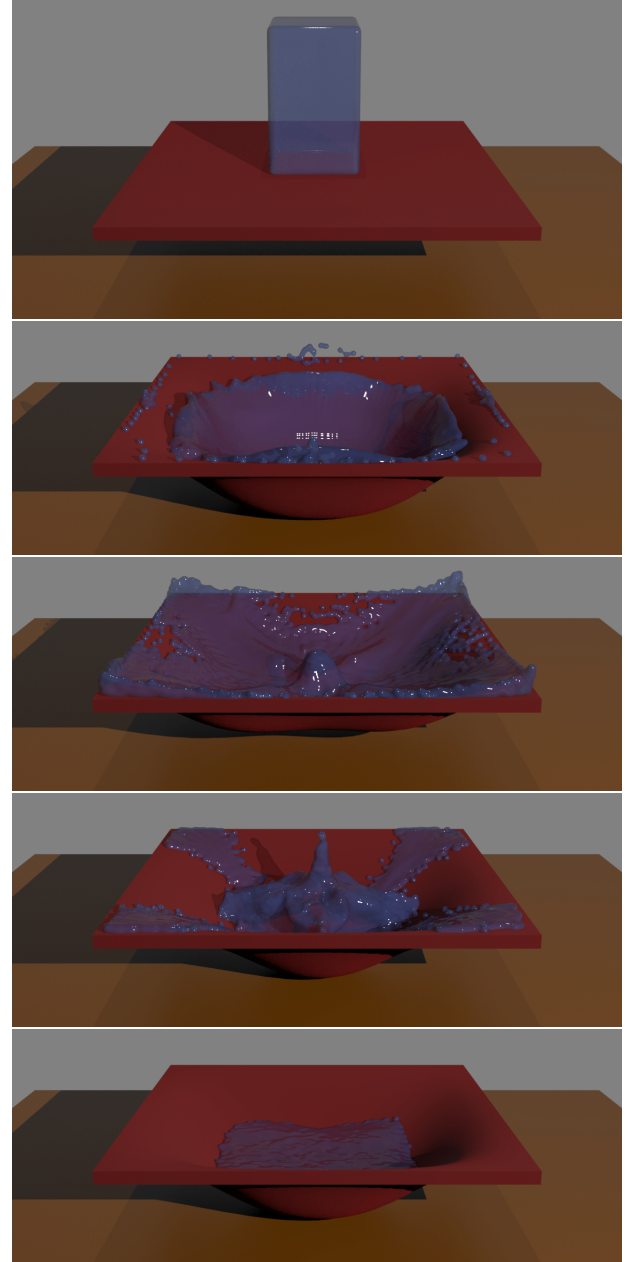


Figure 4: A volume of liquid falls onto and comes to rest atop a deformable platform.

Solid Compressibility Example. In contrast to the Eulerian coupling scheme of Teng et al. [2016], our formulation allows for simulation of compressible solids without special treatment. To showcase this feature, we simulated the expansion of an initially squished cube submerged in a body of water. At the start, this solid is compressed to an eighth of its rest volume; however, elastic forces cause it to rapidly recover its original size. Furthermore, the initially compact solid begins to sink slightly under the influence of gravity, only to be pushed to the surface as expansion lowers its density.

Liquid Supported by Deformable Solid. Two different scenes were set up to illustrate that our method can capture interactions between fluids and deformable solids of modest thickness. For the first scenario, a mass of fluid is dropped on an angled rectangular elastic solid panel that is constrained along two of its sloped sides. Upon contact, the fluid deforms the object as expected. Because the solid is angled, the liquid then starts freely flowing down its slope, showing that the free-slip condition is correctly applied at the solid-fluid interface and no grid artefacts are present. As the fluid flows off the object's surface, the solid elastically rebounds and gradually returns to equilibrium.

The second scenario consists of a mass of fluid being dropped on a highly deformable horizontal platform. Rich two-way interactions between the liquid and the solid can be observed: fluid flow stretches and distorts the platform, and, conversely, internal elastic forces of the solid work to undo the induced deformations, further agitating the fluid. As the system settles into equilibrium, the liquid collects into a pool at the center of its deformable enclosure (see Figure 4).

Fluid in a Sphere. In this experiment, a ball of liquid with a high initial velocity is placed inside a hollow deformable sphere. It can be clearly observed that fluid interactions not only locally distort the solid's shape, but also affect the overall trajectory of the combined system. In particular, the sphere is propelled to one side and bounces off a side wall as a result of the liquid's motion within it (see Figure 5).

Light Dino. For our last animation example, we demonstrate a somewhat more complex solid geometry in the form of a light dinosaur-shaped toy (a standard Houdini model) being dropped into a breaking column of water. Two-way interactions can again be observed, with the water being deflected by contact with the solid and the motion of the water distinctly impacting the toy's trajectory. Figure 1 contains several frames from this animation.

SPD System Conditioning. To assess the effect of our SPD transformation technique on conditioning and sparsity of the system, we ran a few instances of our dam break experiment at different grid and mesh resolutions. The simulations were allowed to proceed for 100 frames, and the relevant properties of coefficient matrices were computed and aggregated (condition numbers were estimated using MATLAB's `cond` function [Mathworks 2017]). Additionally, time taken to solve the pressure projection system in Eigen was recorded for each formulation (BiCGSTAB in conjunction with an incomplete LU preconditioner [Saad 2003] was used to solve the indefinite system²). Our findings are summarized in Table 2. The data reveals that the transformation results in a modest rise in matrix density for our test case; in addition, the condition number sees a noticeable increase. (As another point of comparison, a naïve approach to achieve a positive-definite system would be to simply form the normal equations, but this has the significantly worse effect of *squaring* the condition number.) More importantly, average convergence time for the pressure projection illustrates the advantage of our SPD transformation: the indefinite system

generally takes longer to solve. Note also that despite being slightly slower on the coarsest grid, the SPD system can be solved almost twice as fast on the medium grid, and almost four times as fast on the finest grid. That is, the SPD formulation exhibits appreciably better scaling with respect to simulation resolution.

Stability Assessment. The dam break experiment was also used to gauge the effect of strong coupling on stability. To this end, we simulated a low-resolution dam break using several pressure projection schemes: weak (alternating) coupling with the solid, strong coupling with the solid's damping only, and strong coupling of solid damping and elasticity. In order to strain these methods, a stiff but light deformable body was used along with high gravity and relatively large time steps. As expected, the first method was unstable, with the solid undergoing extreme distortion, and the simulation eventually failing. While the simulation was able to complete when only solid damping was coupled with pressure projection, results obtained via this scheme were noticeably unrealistic; specifically, since pressure projection is sequenced after elastic forces, pressure-induced buoyancy inaccurately dominates and incorrectly pushes the stiff solid to the surface. Our proposed fully coupled approach, on the other hand, encountered no difficulties with this scenario. Additionally, the fully coupled simulation was repeated with zero damping force; once again, our formulation had no problems producing the animation. This result is significant, since the current state-of-the-art method of Robinson-Mosher et al. [2009] simultaneously accounts for fluid pressure and solid damping (but not elasticity) and, thus, requires the deformable object to be damped.

9 CONCLUSIONS

We have presented a novel approach to simulating two-way coupling between fluids and deformable bodies. Compared to previous solutions, our cut-cell method facilitates accurate free-slip boundary condition enforcement at the true solid-fluid interface without adding new degrees of freedom. Due to a fully implicit treatment of solid dynamics as part of the pressure projection, our technique also enjoys better stability properties. Additionally, an efficient SPD transformation scheme was introduced that allows us to convert the symmetric indefinite projection system into a form that is appreciably faster to solve numerically.

We conclude by listing possible avenues for future research. With respect to simulation of thin solids, it was established that our method is capable of producing animations if the object's thickness is greater than a grid cell width or two. However, prevention of fluid flow through the solid is much more difficult when it has a very small or zero thickness (e.g., cloth); furthermore, differences in pressure and velocity on opposite sides of the object must be taken into account. We expect that our method could be combined with the techniques proposed by Azevedo et al. [2016] to allow for efficient simulation of scenarios involving such thin solids or shells. We assumed a simple isotropic linear elastic corotational model that yields realistic results for many deformable objects, but the dynamics of more complex materials cannot be adequately captured with this approach. To properly treat nonlinear constitutive models within our framework, a linearization of the solid forces or a full nonlinear solve would likely be required. Improved numerical linear algebra solvers tailored to our SPD system, such

²We also attempted to solve the indefinite system using MINRES with a block-based preconditioner in the manner suggested by Robinson-Mosher et al. [2008], but in our experiments this performed appreciably worse than ILU-preconditioned BiCGSTAB. We therefore present the BiCGSTAB data as our baseline in Table 2.

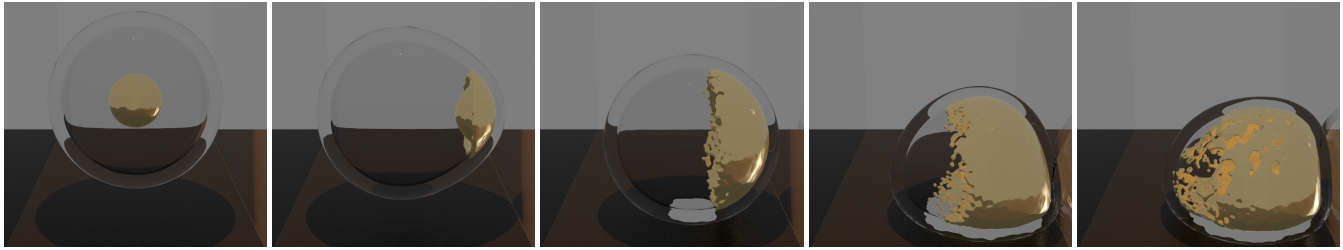


Figure 5: A volume of liquid moving within a thick-walled spherical shell affects its overall motion, forcing to collide with the right wall and bounce.

Table 2: Comparison of coefficient matrices for the indefinite and SPD formulations (all entries in this table are averaged over 100 simulation steps).

Fluid Grid Size	Solid		Nonzeros		log ₁₀ of Condition Number		Solve Time (s)	
	Nodes	Elements	Indefinite	SPD	Indefinite	SPD	Indefinite	SPD
40 × 40 × 40	135	336	157498	185503	8.347	10.497	1.635	1.813
60 × 60 × 60	352	1134	499787	610697	8.579	11.311	16.121	8.292
80 × 80 × 80	725	2688	1177419	1439173	8.819	11.497	100.875	28.020

as multigrid schemes, could benefit our timing results. To allow the liquid to freely separate from solids, as originally proposed by Batty et al. [2007], one could explore extending our method with inequality boundary conditions leading to an LCP system. Finally, we treated contact between the deformable solid and the domain boundaries with a simple explicit correction; it could also be valuable to incorporate implicit contact response into our system. For configurations involving multiple deformable objects and complex contact, we expect that this extension would further improve stability and accuracy.

ACKNOWLEDGEMENTS

This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada (RGPIN-04360-2014, CRDPJ-499952-2016) and Side Effects Software. The first author was supported by funding from NSERC and the Ontario Graduate Scholarship program.

We thank Ryan Goldade for his assistance with rendering our simulations in Houdini.

REFERENCES

Nadir Akinci, Jens Cornelis, Gizem Akinci, and Matthias Teschner. 2013. Coupling elastic solids with smoothed particle hydrodynamics fluids. *Computer Animation and Virtual Worlds* 24, 3-4 (2013), 195–203.

Vinicius C Azevedo, Christopher Batty, and Manuel M Oliveira. 2016. Preserving geometry and topology for fluid flows with thin obstacles and narrow gaps. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 97.

Jernej Barbič. 2012. *Exact corotational linear FEM stiffness matrix*. Technical Report. University of Southern California.

George Keith Batchelor. 2000. *An introduction to fluid dynamics*. Cambridge university press.

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 100.

Christopher Batty, Stefan Xenos, and Ben Houston. 2010. Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. *Computer Graphics Forum* 29, 2 (2010), 695–704.

Robert Bridson. 2015. *Fluid simulation for computer graphics*. CRC Press.

Mark Carlson, Peter J Mucha, and Greg Turk. 2004. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 377–384.

Nuttapong Chentanez, Tolga G Goktekin, Bryan E Feldman, and James F O'Brien. 2006. Simultaneous coupling of fluids and deformable bodies. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 83–89.

Pascal Clausen, Martin Wicke, Jonathan R Shewchuk, and James F O'Brien. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Transactions on Graphics (TOG)* 32, 2 (2013), 17.

Essex Edwards and Robert Bridson. 2014. Detailed water with coarse grids: combining surface meshes and adaptive discontinuous Galerkin. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 136.

Doug Enright, Duc Nguyen, Frederic Gibou, and Ron Fedkiw. 2003. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference*. American Society of Mechanical Engineers, 337–342.

Bryan E Feldman, James F O'Brien, and Bryan M Klingner. 2005. Animating gases with hybrid meshes. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 904–909.

Nick Foster and Ronald Fedkiw. 2001. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 23–30.

Nick Foster and Dimitri Metaxas. 1996. Realistic animation of liquids. *Graphical models and image processing* 58, 5 (1996), 471–483.

Olivier Gènevaux, Arash Habibi, and Jean-Michel Dischler. 2003. Simulating Fluid-Solid Interaction. In *Graphics Interface*, Vol. 2003. 31–38.

Frédéric Gibou and Chohong Min. 2012. Efficient symmetric positive definite second-order accurate monolithic solver for fluid/solid interactions. *J. Comput. Phys.* 231, 8 (2012), 3246–3263.

Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. 2005. Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 973–981.

Gaël Guennebaud, Benoît Jacob, and others. 2010. Eigen v3. <http://eigen.tuxfamily.org>. (2010).

Florian Hecht, Yeon Jin Lee, Jonathan R Shewchuk, and James F O'Brien. 2012. Updated sparse cholesky factors for corotational elastodynamics. *ACM Transactions on Graphics (TOG)* 31, 5 (2012), 123.

Ben Houston, Chris Bond, and Mark Wiebe. 2003. A unified approach for modeling complex occlusions in fluid simulations. In *ACM SIGGRAPH 2003 Sketches & Applications*. ACM, 1–1.

Bryan M Klingner, Bryan E Feldman, Nuttapong Chentanez, and James F O'Brien. 2006. Fluid animation with dynamic meshes. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 820–825.

David IW Levin, Joshua Litven, Garrett L Jones, Shinjiro Sueda, and Dinesh K Pai. 2011. Eulerian solid simulation with contact. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 36.

- Wenlong Lu, Ning Jin, and Ronald Fedkiw. 2016. Two-way coupling of fluids to reduced deformable bodies. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 67–76.
- Mathworks. 2017. MATLAB. <https://www.mathworks.com/>. (2017).
- Marek Krzysztow Misztal, Kenny Erleben, Adam Bargteil, Jens Fursund, Brian Bunch Christensen, Jakob Andreas Bærentzen, and Robert Bridson. 2014. Multiphase flow of immiscible fluids on unstructured moving meshes. *IEEE transactions on visualization and computer graphics* 20, 1 (2014), 4–16.
- Matthias Müller and Markus Gross. 2004. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society, 239–246.
- Yen Ting Ng, Chohong Min, and Frédéric Gibou. 2009. An efficient fluid–solid coupling algorithm for single-phase flows. *J. Comput. Phys.* 228, 23 (2009), 8807–8829.
- Vito Pasquariello, Georg Hammerl, Felix Örley, Stefan HICKEL, Caroline Danowski, Alexander Popp, Wolfgang A Wall, and Nikolaus A Adams. 2016. A cut-cell finite volume–finite element coupling approach for fluid–structure interaction in compressible flow. *J. Comput. Phys.* 307 (2016), 670–695.
- Nick Rasmussen, Douglas Enright, Duc Nguyen, Sebastian Marino, Nigel Sumner, Willi Geiger, Samir Hoon, and Ronald Fedkiw. 2004. Directable photorealistic liquids. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 193–202.
- Avi Robinson-Mosher, R Elliot English, and Ronald Fedkiw. 2009. Accurate tangential velocities for solid fluid coupling. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 227–236.
- Avi Robinson-Mosher, Craig Schroeder, and Ronald Fedkiw. 2011. A symmetric positive definite formulation for monolithic fluid structure interaction. *J. Comput. Phys.* 230, 4 (2011), 1547–1566.
- Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 46.
- Doug Roble, Nafees bin Zafar, and Henrik Falt. 2005. Cartesian grid fluid simulation with irregular boundary voxels. In *ACM SIGGRAPH 2005 Sketches*. ACM, 138.
- Yousef Saad. 2003. *Iterative methods for sparse linear systems*. SIAM.
- Rahul Sheth, Wenlong Lu, Yue Yu, and Ronald Fedkiw. 2015. Fully momentum-conserving reduced deformable bodies with collision, contact, articulation, and skinning. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 45–54.
- Side Effects Software. 2017. Houdini. <https://www.sidefx.com/>. (2017).
- Barbara Solenthaler, Jürg Schläfli, and Renato Pajarola. 2007. A unified particle model for fluid–solid interactions. *Computer Animation and Virtual Worlds* 18, 1 (2007), 69–82.
- Fotis Sotiropoulos and Xiaolei Yang. 2014. Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences* 65 (2014), 1–21.
- Jos Stam. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 121–128.
- Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 102.
- Yun Teng, David IW Levin, and Theodore Kim. 2016. Eulerian solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 200.
- Joseph Teran, Sylvia Blemker, V Hing, and Ronald Fedkiw. 2003. Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 68–74.
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. *ACM Siggraph Computer Graphics* 21, 4 (1987), 205–214.
- Daniel Weber, Johannes Mueller-Roemer, André Stork, and Dieter Fellner. 2015. A Cut-Cell Geometric Multigrid Poisson Solver for Fluid Simulation. *Computer Graphics Forum* 34, 2 (2015), 481–491.
- Gary D Yngve, James F O’Brien, and Jessica K Hodgins. 2000. Animating explosions. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 29–36.