Title

Simultaneous coupling of fluids and deformable bodies

Permalink

https://escholarship.org/uc/item/3923x8tp

ISBN

9781595933645

Authors

Chentanez, Nuttapong
Goktekin, Tolga G
Feldman, Bryan E
et al.

Publication Date

2006

DOI

10.1145/1179849.1179930

Supplemental Material

https://escholarship.org/uc/item/3923x8tp#supplemental

Peer reviewed

Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Simultaneous Coupling of Fluids and Deformable Bodies

Nuttapong Chentanez Tolga G. Goktekin Bryan E. Feldman James F. O'Brien

University of California, Berkeley

Abstract

This paper presents a method for simulating the two-way interaction between fluids and deformable solids. The fluids are simulated using an incompressible Eulerian formulation where a linear pressure projection on the fluid velocities enforces mass conservation. Similarly, elastic solids are simulated using a semi-implicit integrator implemented as a linear operator applied to the forces acting on the nodes in Lagrangian formulation. The proposed method enforces coupling constraints between the fluid and the elastic systems by combining both the pressure projection and implicit integration steps into one set of simultaneous equations. Because these equations are solved simultaneously the resulting combined system treats closed regions in a physically correct fashion, and has good stability characteristics allowing for relatively large time steps. This general approach is not tied to any particular volume discretization of fluid or solid, and we present results implemented using both regular-grid and tetrahedral simulations.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling, Physically Based Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, Animation; I.6.8 [Simulation and Modeling]: Types of Simulation, Animation

Keywords: Natural phenomena, physically based animation, computational fluid dynamics, two-way coupling, deformable bodies.

1. Introduction

The interaction between a fluid and deformable body can create complex and interesting motion that would be difficult to convincingly animate by hand. In this work we present a method to solve for the interaction by simultaneously conserving the momentum of the fluid and deformable body while enforcing the conservation of fluid mass.

Previous approaches for coupling fluid and deformable bodies use a time splitting procedure. They alternately fix the fluid pressure when simulating the solid, then fix the solid's velocity when simulating the fluid. While this approach works reasonably well for physical systems with non-stiff coupling, it can lead to instability and visual artifacts for other systems. These problems occur because while solid velocities are fixed they will ignore arbitrarily large fluid pressures, and the converse when the fluid velocities are fixed. For a tightly coupled system like a water balloon where small changes in the state of the solid cause nearly instantaneous changes in the fluid and vice versa, time splitting becomes untenable and difficulties can still arise even for less tightly coupled systems. Time splitting also creates problems for systems that include fluid regions completely surrounded by a deformable body. The volume of these regions can change during the solid simulation. In this case applying Neumann boundary conditions to the fluid simulation

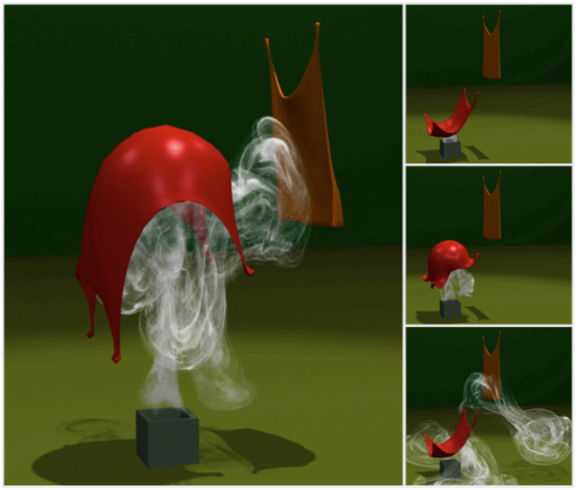

Figure 1: *A sequence of images showing a jet of smoke interacting with multiple thin rubber sheets*

is incompatible with the requirement that the fluid be divergence free. To remedy this problem, time-splitting methods typically use non-physical fixes. For example, [GSLF05] proposes detecting these closed regions using a flood fill algorithm and then performing an area-weighted adjustment of the velocities on the boundary such that there is no net flux across the boundary of the closed region. It is not clear however, how to generalize the fix to correctly account for a deformable object with varying thickness or material properties. By enforcing coupling simultaneously our method avoids these problems and allows for substantially larger time steps.

The method we describe is largely independent of the discretization scheme used. We have implemented the coupling with fluid simulators discretized using dynamic tetrahedral meshes [KFCO06] as well as on a fixed regular Eulerian grid similar to that of [EMF02]. We include several examples, such as the one in Figure 1, that demonstrate the effectiveness of the method.

2. Background

Physically-based methods have proven to be popular and effective for generating animations of fluids and deformable bodies. The use of the 3D Navier-Stokes equations for animating fluids was introduced to the graphics community by [FM96]. A number of papers have since appeared which improve the stability [Sta99], enhance the resolution of small features [FSJ01], and extend the capabilities to include fire [NFJ02], explosions [FOA03], and visco-elastic fluids [GBO04]. Simulating liquids requires surface tracking methods. The use of level set methods for surface tracking was introduced in [FF01] which was improved by the particle level set method in [EMF02]. More recently, a robust technique based on semi-Lagrangian contouring was demonstrated in [BGOS06].

Deformable objects were first simulated for use in computer graphics in [TPBF87]. A full description of the methods for simulating deformable solids is beyond the scope of this paper. We recommend the following extensive survey on this topic [NMK*05].

An early example of coupling fracturing solids and compressible explosions appears in [YOH00]. They use a pressure accumulation approach that is potentially very accurate, however modeling stiff pressure waves which arise in a compressible simulation necessitates very small time steps for stability. A standard method to avoid the stability problems is to model the fluid as incompressible. Two-way interaction between incompressible fluid and rigid bodies is accomplished in [CMT04] by projecting the fluid velocities within the solids to behave rigidly at the end of the time step. Unfortunately, this method uses a two-step projection approach that can lead to visual artifacts and fluid loss. The problems caused by two-step projection are avoided in [KFCO06] by enforcing the coupling constraints and incompressibility simultaneously. Our method extends their approach to include deformable bodies as well as rigid ones.

Another common method for simulating the interaction between fluid and deformable solids is the time splitting procedure where the fluid and solid are solved for alternately. During the simulation of each process, the constraints imposed by the other system are held fixed. An application of this scheme for simulating fluids interacting with filaments or thin flexible sheets is presented in [Pes02] which has been improved by [LL01] and [GSLF05] to account for discontinuities across thin interfaces. [GAD03] proposes a technique for coupling a liquid and a mass spring solid by modeling repulsion forces between solid's nodal masses and fluid's marker particles. Coupling of a deformable body via time splitting with an SPH-based fluid solver appears in [MST*04].

3. Methods

The interaction between a fluid and a deformable solid occurs at the interface where the fluid applies pressure forces to the solid's boundary while the solid imposes boundary fluxes on the fluid. For real physical systems these forces lead to the formation of pressure and elastic waves. These waves typically propagate very rapidly such that direct simulation would require very small time steps. A common method to relieve the small time step requirement is to model the propagation as occurring instantaneously via a projection step. In the case of the fluid, this projection is the Poisson pressure projection. For elastic solids the implicit integration can be viewed as a projection on the nodal accelerations. In order to properly account for both the fluid and solid systems, the projections should occur simultaneously. Therefore, in order to simulate the complex interactions between a solid and a fluid we need to augment both simulations to account for the other system and create a combined coupled system. Specifically, when performing pressure correction on the fluid and implicitly solving for the solid's node velocities, we need to simultaneously account for how fluid pressure changes will effect the deformable solid and how the solid's motion will effect the fluid.

3.1. Fluid Simulation

The behavior of an inviscid incompressible fluid is governed by conservation of momentum:

$$\frac{\partial v}{\partial t} = -(v \cdot \nabla) v - \frac{\nabla p}{\rho} + \frac{f_e}{\rho} \tag{1}$$

subject to the mass conservation constraint:

$$\nabla \cdot v = 0 \tag{2}$$

where v is velocity, p is pressure, ρ is density, and f_e represents external forces. The standard Eulerian method for simulating incompressible fluids is to first accelerate the fluid by all terms in Equation (1) excluding the pressure term to obtain an intermediate velocity, v^*. Then, to conserve mass, a linear system is solved to find the pressure that accelerates v^* to be divergence free. For more detail on fluid simulation, see [FM96], [FF01], [EMF02] for regular grid fluid simulator, [LGF04] for an octree based simulator, and [FOK05], [FOKG05], [KFCO06] for tetrahedral fluid simulator.

In the interior of the fluid, fluid is accelerated by the gradient of the pressure as seen in Equation (1). The fluid at the interface with a solid must have the same normal component of velocity as that of the solid. This constraint prevents fluid from leaking into or coming from the solid. We enforce this condition by constraining the normal component of fluid velocity to be that of the solid (see Section 3.3).

3.2. Solid Simulation

Using a finite element or finite difference method, the dynamics of an elastic solid body deforming under pressure forces can be discretized in the following general form:

$$\mathbf{M}a + \mathbf{C}(u,d) + \mathbf{K}(d) - f_p - f_e = 0 \tag{3}$$

where the solid's node acceleration, velocity and displacement are respectively denoted by a, u and d, $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ and $\mathbf{K}$ are the non-linear damping and stiffness functions respectively. Forces due to fluid's pressure are denoted by f_p while f_e accounts for any additional external forces. For general finite element method, we recommend [CMP89] to the readers. By applying a trapezoidal rule implicit Newmark scheme for time integration and linearizing the stiffness function around the current displacement, Equation (3) can be put into the following form (see Appendix):

$$\mathbf{A}u^{n+1} - \mathbf{J}p^{n+1} = b \tag{4}$$

where

$$\mathbf{A} = \frac{1}{h}\mathbf{M} + \mathbf{C} + \frac{h}{2}\mathbf{K}' \quad b = \frac{1}{h}\mathbf{M}u^n - \frac{h}{2}\mathbf{K}'u^n - \mathbf{K}(d^n) + f_e$$

also,

$$d^{n+1} = d^n + \frac{1}{2}h(u^n + u^{n+1}) \tag{5}$$

$\mathbf{K}'$ is the Jacobian of the elastic force function evaluated at d^n. Forces due to the pressure of the fluid are computed and mapped by the $\mathbf{J}$ matrix. Finally, h and n denote the time step and time index respectively.

For the simplicity we assume Rayleigh damping, so that $\mathbf{C}$ is a linear combination $\mathbf{K}'$ and $\mathbf{M}$, and that the damping is a linear function of only velocity. We use a finite element method with constant element Green strain and linear isotropic stress-strain relationship as in [OH99] to obtain $\mathbf{M}$, $\mathbf{K}$, and $\mathbf{K}'$, however alternative solid constitutive models could be used. Other semi-implicit integration schemes that can be formulated as a linear equations of u, d and p can be used as well.

3.3. Fluid and Solid Coupling

The interaction between a fluid and deformable solid occurs as a result of the following conditions:

- The velocities in the normal direction at the interface are the same for both solid and fluid,
- The fluid velocity is divergence free and the deformable solid velocity behaves according to its constitutive equation,
- The fluid's pressure exerts a force on the solid and the solid's elastic force affects the fluid.

The proposed technique augments a standard method of simulating incompressible fluids to handle interaction with deformable bodies by including the effect of pressure at the interface in addition to the effect of the pressure within the fluid interior. The fluid pressure at the interface exerts force on the solid, thus changing its velocity. This change in solid velocity in turn alters the boundary condition imposed on the fluid. By simultaneously solving for both the solid's velocities that satisfies Equation (3) and fluid pressure that enforces Equation (2), we obtain a velocity field for the fluid and node velocities for the solid that satisfy all the coupling requirements.

To achieve this effect, we modify the pressure projection step to include the divergence due to boundary flux in addition to the divergence of the intermediate internal fluid velocity field, v^*. This concept is similar in spirit to the rigid body and fluid coupling technique presented in [KFCO06].

The Pressure-Poisson equation can be formulated as:

$$-\mathbf{D}_1\mathbf{H}u^{n+1} + \frac{h}{\rho}\mathbf{D_2}\mathbf{G}p^{n+1} = \mathbf{D_2}v^* \tag{6}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are matrices for computing the divergence due to the boundary and internal fluid fluxes respectively, $\mathbf{G}$ is the gradient matrix for internal fluxes and $\mathbf{H}$ is a matrix that maps the solid's boundary node velocities to fluxes on the fluid's boundary. Finally, v^* is the intermediate fluid velocity computed from the fluid simulation that needs to be projected onto its divergence free component, v^{n+1}.

In order to simulate full coupling between the fluid and the solid, we combine Equation (4) and Equation (6) and solve for u^{n+1} and p^{n+1} simultaneously:

$$\begin{bmatrix} \mathbf{A} & -\mathbf{J} \\ -\mathbf{D}_1\mathbf{H} & \frac{h}{\rho}\mathbf{D_2}\mathbf{G} \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} b \\ \mathbf{D}_2 v^* \end{bmatrix} \tag{7}$$

Although the system of equations is not symmetric, it is very sparse and can be solved efficiently using an iterative method with a preconditioner. In our implementation, we used a bi-conjugate gradient stabilized method with an incomplete Cholesky preconditioner.

Finally, we can compute the divergence free fluid velocities v^{n+1} as follows:

$$v_i^{n+1} = \begin{cases} (v^* - \frac{h}{\rho}(\mathbf{G}p))_i & \text{if } i \text{ is not a boundary face} \\ (\mathbf{H}u^{n+1})_i & \text{if } i \text{ is a boundary face} \end{cases} \tag{8}$$

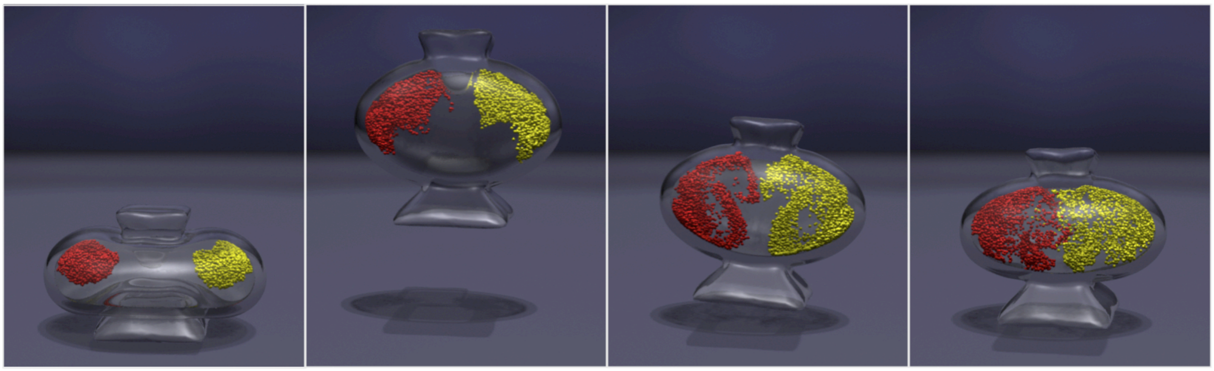

Figure 2: *A closed system where a liquid-filled deformable toy bounces of the floor and sets the fluid inside in motion.*

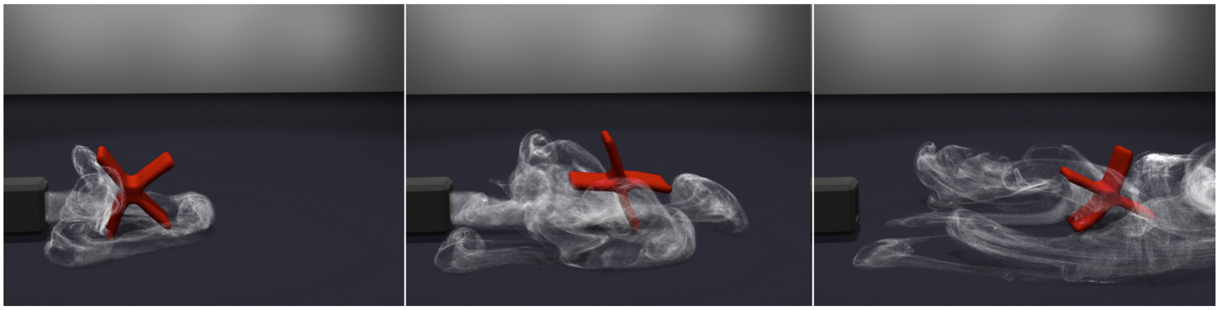

Figure 3: *A deformable paddle blown by a jet of smoke rotates and bounces against the floor.*

3.4. Conversion Matrices

The above described procedure requires the **J** and **H** matrices which respectively map solid node velocities to fluid face velocities and fluid pressure to solid node forces.

For the unstructured tetrahedral dynamic mesh discretization described in [KFCO06], the matrix **J** encodes a computation of looping through all the fluid's boundary tetrahedrons and adding up the contribution of force on the solid's boundary faces due to the pressure. Matrix **H** represents a computation of looping through all the solid's boundary faces, computing the flux, and adding the contribution to the corresponding fluid boundary faces. Other interpolation techniques that can be represented as a linear map can also be used.

For the regular grid case, we compute the forces due to pressure by integrating interpolated pressure values over each face of the solid. Therefore, **J** is constructed by looping through all solid faces and storing the interpolation stencils needed for sample points on the face used for integration. In order to construct **H** we employ a method similar to that used in [GSLF05] where we first check if the ray between two cell centers intersects a solid face, and if so, we constrain the flux of the corresponding cell face to the flux at the point of intersection of the solid triangle.

	Figure	Avg time per step	Time step
Bar simultaneous	6	45.2 sec	$\frac{1}{30}$ sec
Bar time splitting (unstable)	6	33.0 sec	$\frac{1}{30}$ sec
Bar time splitting	6	32.8 sec	$\frac{1}{120}$ sec
Paddle	3	72.6 sec	$\frac{1}{30}$ sec
Toy	2	54.3 sec	$\frac{1}{60}$ sec
Sheet	1	172.3 sec	$\frac{1}{60}$ sec
Bunny	4	150 sec	$\frac{1}{300}$ sec
Bowl	5	120 sec	$\frac{1}{300}$ sec

Table 1: *Timing results of the examples used in this paper. Per step running time of the bunny and bowl examples include surface tracking as well.*

4. Results and Discussion

We have implemented this method for simultaneous coupling of fluids and deformable solids for both grid based and tetrahedral simulations, and used it to generate several example animations. All of the examples shown in this paper also appear on the accompanying video. The implementations are done in MATLAB[†] and C++. We rendered all examples with an open source renderer PIXIE[‡].

Figure 1 shows frames from an animation of a fluid interacting with multiple deformable solids where a jet of smoke

[†] http://www.mathworks.com

[‡] http://sourceforge.net/projects/pixie

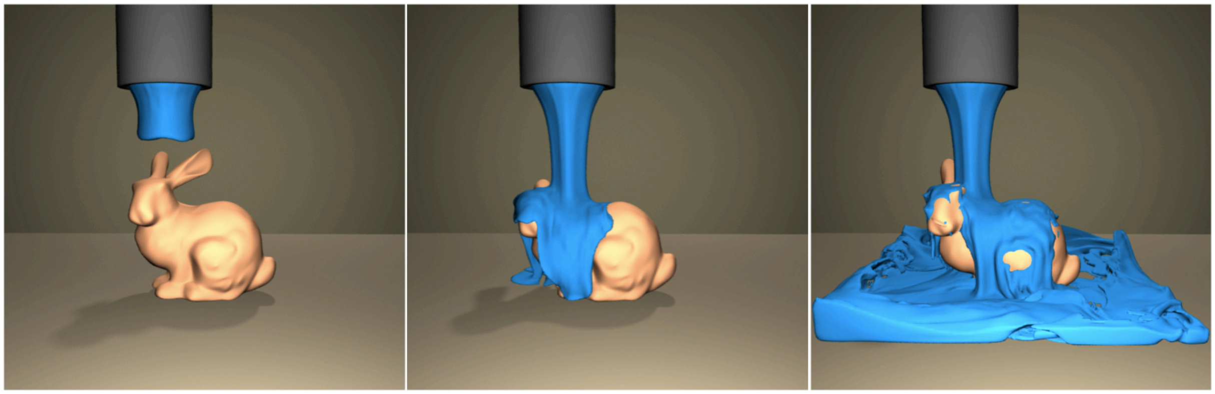

Figure 4: *Paint is poured over a deformable bunny.*

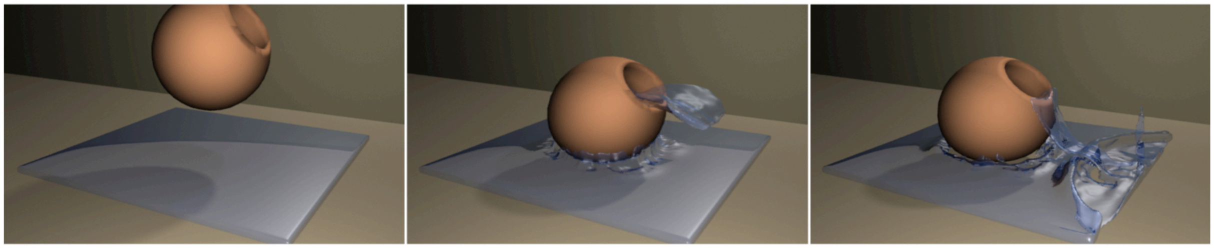

Figure 5: *A deformable bowl filled with liquid falls on shallow water contained in an invisible water tank, hits the bottom of the tank and squirts the liquid out.*

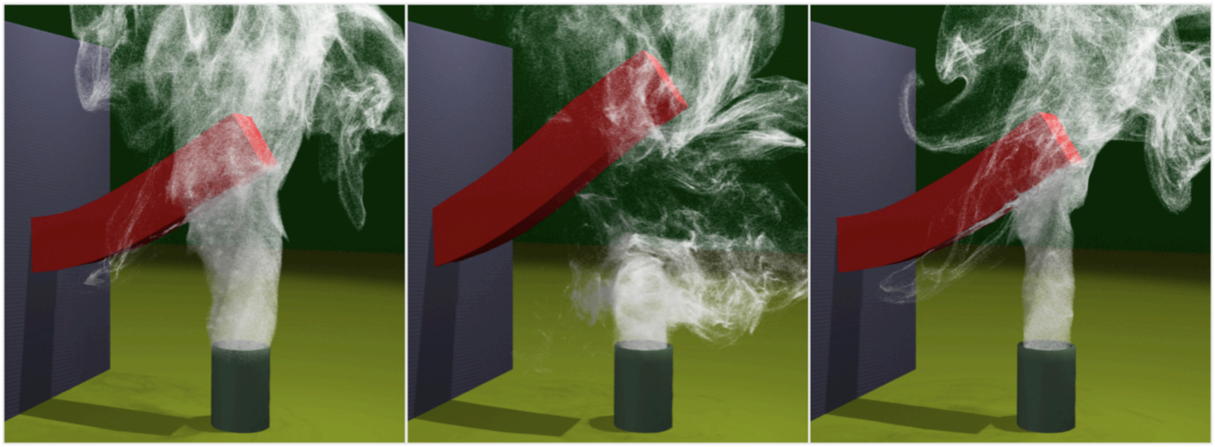

Figure 6: *Comparisons of our simultaneous coupling method with time step of $\frac{1}{30}$sec (left), a time-splitting method with time steps of $\frac{1}{30}$sec (middle) and a time-splitting method with time steps of $\frac{1}{120}$sec (right). Our method is stable at time step $\frac{1}{30}$sec while the time splitting method requires time step of $\frac{1}{120}$sec to be stable.*

is blown into a rubber sheet which deflects the smoke towards another sheet. The rubber sheets are simulated with tetrahedral elements and have thickness as opposed to the infinitesimally thin cloth sheets used in [GSLF05].

An example of a fluid fully surrounded by a deformable solid creating a closed system is shown in Figure 2. Here a liquid-filled deformable toy bounces of the floor causing the liquid inside to swirl. The liquid conserves its volume inside the cavity of the toy and prevents the toy from collapsing on itself upon impact with the ground.

Another interesting interaction between a fluid and a solid is shown in Figure 3 where a jet of smoke is blown on a deformable paddle. The paddle rotates and bounces off the floor, and in turn causes the smoke to swirl around the paddle.

We also generated examples of liquids with free surfaces interacting with deformable solids. In Figure 4 paint is poured over a bunny causing it to deform under it. Frames from an animation of a bowl filled with liquid falling onto a shallow water contained in an invisible tank are shown in Figure 5. The impact causes the bowl to deform and squirt the liquid out. Both of these examples are generated using a regular grid fluid simulator.

In Figure 6 we compare results obtained using a time-splitting method with our method of simultaneous coupling. The time-splitting method alternates between simulating the solid, with forces exerted by the fluid pressure and simulating the fluid with the boundary condition provided by the solid. Using a time step of $\frac{1}{30}$ the time-splitting method quickly goes unstable while our simultaneous method remains stable. Only after reducing the time step to $\frac{1}{120}$ (4× smaller) are we able to obtain a stable time-splitting solution. On a per-time step basis our method takes longer because we solve the larger non-symmetric linear system from Equation (7), but this difference is more than compensated by the fact that we can take larger time steps. Additionally, we had originally wanted to simulate a bar with less damping in order to get a more interesting motion of the bar, however obtaining a stable solution for the time-splitting case required a small enough time step to render our attempts impractical.

The collision detection and response module for our finite element based tetrahedral deformable solid simulator is in its early stages. Even though we respond to the collisions with surrounding obstacles our current implementation does not account for self-collisions of a deformable body. This is apparent in our bunny example where the right ear of the bunny goes through the head and pokes through the nose of the bunny in our video. We also do not prevent inversion of solid tetrahedra which could lead into problems with accuracy. Currently we are investigating methods that are robust against element inversions such as the one proposed in [ITF04].

The main drawback of the method presented in this paper is that we need to solve a large non-symmetric linear system that is composed of all fluid and solid degrees of freedom in the domain instead of solving smaller symmetric linear systems for each separately. The main structure of the system resembles a block diagonal matrix where each block represents the individual solid body and fluid system. The only entries outside the block diagonal region are those used in the coupling of the boundary values. Therefore the system is still very sparse and efficient to solve with preconditioned iterative methods, and by solving it our method does not incur an overhead large enough to offset the advantage of using large time steps.

Acknowledgments

We thank the other members of the Berkeley Graphics Group for their helpful criticism and comments. This work was supported in part by California MICRO 04-066 and 05-044, and by generous support from Apple Computer, Pixar Animation Studios, Autodesk, Intel Corporation, Sony Computer Entertainment America, and the Alfred P. Sloan Foundation. Feldman were supported by NSF Graduate Fellowships.

References

[BGOS06] Bargteil A. W., Goktekin T. G., O'Brien J. F., Strain J. A.: A semi-Lagrangian contouring method for fluid simulation. *ACM Transactions on Graphics 25*, 1 (2006).

[CMP89] Cook R. D., Malkus D. S., Plesha M. E.: *Concepts and Applications of Finite Element Analysis*, third ed. John Wiley & Sons Inc., New York, 1989.

[CMT04] Carlson M., Mucha P. J., Turk G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. In *the Proceedings of ACM SIGGRAPH 2004* (Aug. 2004), pp. 377–384.

[EMF02] Enright D., Marschner S., Fedkiw R.: Animation and rendering of complex water surfaces. In *the Proceedings of ACM SIGGRAPH 2002* (Aug. 2002), pp. 736–744.

[FF01] Foster N., Fedkiw R.: Practical animation of liquids. In *the Proceedings of ACM SIGGRAPH 2001* (Aug. 2001), pp. 23–30.

[FM96] Foster N., Metaxas D.: Realistic animation of liquids. In *Graphics Interface 1996* (May 1996), pp. 204–212.

[FOA03] Feldman B. E., O'Brien J. F., Arikan O.: Animating suspended particle explosions. In *Proceedings of ACM SIGGRAPH 2003* (Aug. 2003), pp. 708–715.

[FOK05] Feldman B. E., O'Brien J. F., Klingner B. M.: Animating gases with hybrid meshes. In *Proceedings of ACM SIGGRAPH 2005* (Aug. 2005).

[FOKG05] Feldman B. E., O'Brien J. F., Klingner B. M., Goktekin T. G.: Fluids in deforming meshes. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2005* (July 2005).

[FSJ01] Fedkiw R., Stam J., Jensen H. W.: Visual simulation of smoke. In *the Proceedings of ACM SIGGRAPH 2001* (Aug. 2001), pp. 15–22.

[GAD03] Genevaux O., A H., Dischler J. M.: Simulating fluid-solid interaction. In *Graphics Interface* (2003), pp. 31–38.

[GBO04] Goktekin T. G., Bargteil A. W., O'Brien J. F.: A method for animating viscoelastic fluids. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004) 23*, 3 (2004), 463–468.

[GSLF05] Guendelman E., Selle A., Losasso F., Fedkiw R.: Coupling water and smoke to thin deformable and rigid shells. In *Proceedings of ACM SIGGRAPH 2005* (Aug. 2005), pp. 457–462.

[ITF04] Irving G., Teran J., Fedkiw R.: Invertible finite elements for robust simulation of large deformation. In *2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (July 2004), pp. 131–140.

[KFCO06] Klingner B. M., Feldman B. E., Chentanez N., O'Brien J. F.: Fluid animation with dynamic meshes. In *Proceedings of ACM SIGGRAPH 2006* (Aug. 2006).

[LGF04] Losasso F., Gibou F., Fedkiw R.: Simulating water and smoke with an octree data structure. In *Proceedings of ACM SIGGRAPH 2004* (Aug. 2004), pp. 457–462.

[LL01] LI Z., LAI M.-C.: The immersed interface method for the Navier−Stokes equations with singular forces. *Journal of Computational Physics 171* (2001), 822–842.

[MST*04] MUELLER M., SCHIRM S., TESCHNER M., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids. In *Proceedings of Computer Animation & Social Agents CASA'04* (July 2004), pp. 159–171.

[NFJ02] NGUYEN D., FEDKIW R., JENSEN H.: Physically based modeling and animation of fire. In *Proceedings of ACM SIGGRAPH 2002* (2002).

[NMK*05] NEALAN A., MÜLLER M., KEISER R., BOXERMANN E., CARLSON M.: Physically based deformable models in computer graphics. In *STAR Proceedings of Eurographics 2005* (Sept. 2005), pp. 71–94.

[OH99] O'BRIEN J. F., HODGINS J. K.: Graphical modeling and animation of brittle fracture. In *Proceedings of ACM SIGGRAPH 1999* (Aug. 1999), pp. 137–146.

[Pes02] PESKIN C.: The immersed boundary method. *Acta Numerica 11* (2002), 479–517.

[Sta99] STAM J.: Stable fluids. In *the Proceedings of ACM SIGGRAPH 99* (Aug. 1999), pp. 121–128.

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *International Conference on Computer Graphics and Interactive Techniques* (1987), 205–214.

[YOH00] YNGVE G. D., O'BRIEN J. F., HODGINS J. K.: Animating explosions. In *the Proceedings of ACM SIGGRAPH 2000* (July 2000), pp. 29–36.

Appendix

The time integration of Equation (3) using the general Newmark scheme results in the following set of equations:

$$\mathbf{M}a^{n+1} + \mathbf{C}(u^{n+1}, d^{n+1}) + \mathbf{K}(d^{n+1}) - f_p^{n+1} - f_e^{n+1} = 0 \tag{9}$$

$$d^{n+1} = d^n + hu^n + (\frac{1}{2} - \beta)h^2 a^n + \beta h^2 a^{n+1} \tag{10}$$

$$u^{n+1} = u^n + (1-\gamma)ha^n + \gamma h a^{n+1} \tag{11}$$

Choosing the integration parameters $\beta = \frac{1}{2}$ and $\gamma = 1$, and substituting Equation (11) into Equation (10) we obtain the trapezoidal rule:

$$d^{n+1} = d^n + \frac{h}{2}(u^n + u^{n+1}) \tag{12}$$

$$u^{n+1} = u^n + ha^{n+1} \tag{13}$$

According to Equation (13) the acceleration a^{n+1} can be estimated as:

$$a^{n+1} = \frac{1}{h}(u^{n+1} - u^n) \tag{14}$$

Substituting Equation (14) and Equation (12) into Equation (9), and assuming Rayleigh damping we obtain:

$$\frac{1}{h}\mathbf{M}(u^{n+1} - u^n) + \mathbf{C}u^{n+1} + \mathbf{K}(d^n + \frac{h}{2}(u^n + u^{n+1})) - f_p^{n+1} - f_e^{n+1} = 0 \tag{15}$$

The first order approximation to the non-linear stiffness above can be written as:

$$\mathbf{K}(d^n + \frac{h}{2}(u^n + u^{n+1})) = \mathbf{K}(d^n) + \frac{h}{2}\mathbf{K}' \; (u^n + u^{n+1}) \tag{16}$$

where $\mathbf{K}' = \frac{\partial K}{\partial d}|_{d=d^n}$. Substituting this into Equation (15) we obtain:

$$\frac{1}{h}\mathbf{M}(u^{n+1} - u^n) + \mathbf{C}u^{n+1} + \mathbf{K}(d^n) + \frac{h}{2}\mathbf{K}'(u^n + u^{n+1}) - f_p^{n+1} - f_e^{n+1} = 0 \tag{17}$$

Collecting the terms involving u^{n+1} and u^n, and using the fact that $f_p^{n+1} = \mathbf{J}p^{n+1}$, we finally obtain:

$$(\frac{1}{h}\mathbf{M} + \mathbf{C} + \frac{h}{2}\mathbf{K}')\,u^{n+1} - \mathbf{J}p^{n+1} = (\frac{1}{h}\mathbf{M} - \frac{h}{2}\mathbf{K}')\,u^n - \mathbf{K}(d^n) + f_e^{n+1} \tag{18}$$

which is same as Equation (4).