Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2010)
M. Otaduy and Z. Popovic (Editors)

Practical Animation of Compressible Flow for Shock Waves and Related Phenomena

Nipun Kwatra† Stanford University
Jón T. Grétarsson † Stanford University
Ronald Fedkiw† Stanford University Industrial Light + Magic

Abstract

We propose a practical approach to integrating shock wave dynamics into traditional smoke simulations. Previous methods either simplify away the compressible component of the flow and are unable to capture shock fronts or use a prohibitively expensive explicit method that limits the time step of the simulation long after the relevant shock waves and rarefactions have left the domain. Instead, we employ a semi-implicit formulation of Euler's equations, which allows us to take time steps on the order of the fluid velocity (ignoring the more stringent acoustic wave-speed restrictions) and avoids the expensive characteristic decomposition typically required of compressible flow solvers. We also propose an extension to Euler's equations to model combustion of fuel in explosions. The flow is two-way coupled with rigid and deformable solid bodies, treating the solid-fluid interface effects implicitly in a projection step by enforcing a velocity boundary condition on the fluid and integrating pressure forces along the solid surface. As we handle the acoustic fluid effects implicitly, we can artificially drive the sound speed c of the fluid to ∞ without going unstable or driving the time step to zero. This permits the fluid to transition from compressible flow to the far more tractable incompressible flow regime once the interesting compressible flow phenomena (such as shocks) have left the domain of interest, and allows the use of state-of-the-art smoke simulation techniques.

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

1. Introduction

Shock waves have had a deep and varied impact across multiple disciplines within the graphics community. The solids community, for example, has put significant effort into capturing the destructive effect that blasts have on rigid bodies, realistically fracturing [MMA99, NF99] and generating interesting small-scale debris and dust [IJN09]. Unfortunately, these methods suffer from an over-simplified model of the blast itself, making them useful only in the very limited venue where the dynamic effect of the fluid is negligible.

In the fluids community focus has been more on modeling the after-effects of an explosion, e.g. the smoke plumes of [FM97,Sta99,FSJ01]. A few papers have simulated phenomena related to the explosion itself. For example [RNGF03, TOT*03, FOA03, IKC04, HSF07, KJI07] modify the incompressible flow equations in various ways, such as by adding a divergence term to approximate the expansion due to chemical reactions. These generate fantastic fireball-style effects, but are held back by the underlying modeling assumptions. In particular, by simulating the fluid as *incompressible* they discard the compression waves and the potentially dramatic effects therein–such as shocks.

In order to capture the physics that drive shock waves, we must instead consider the compressible Euler equations, as done in [YOH00, SMML07, SGTL09]. The equations which drive the fluid flow naturally yield the information necessary to generate physically accurate shock waves, albeit at a significantly increased computational cost. Compressible flows require conservative advection schemes such as ENO-Roe [SO88] in order to capture shocks at the correct speeds and properly account for the highly non-linear, dis-

† e-mail: {kwatra|jontg}@stanford.edu, fedkiw@cs.stanford.edu

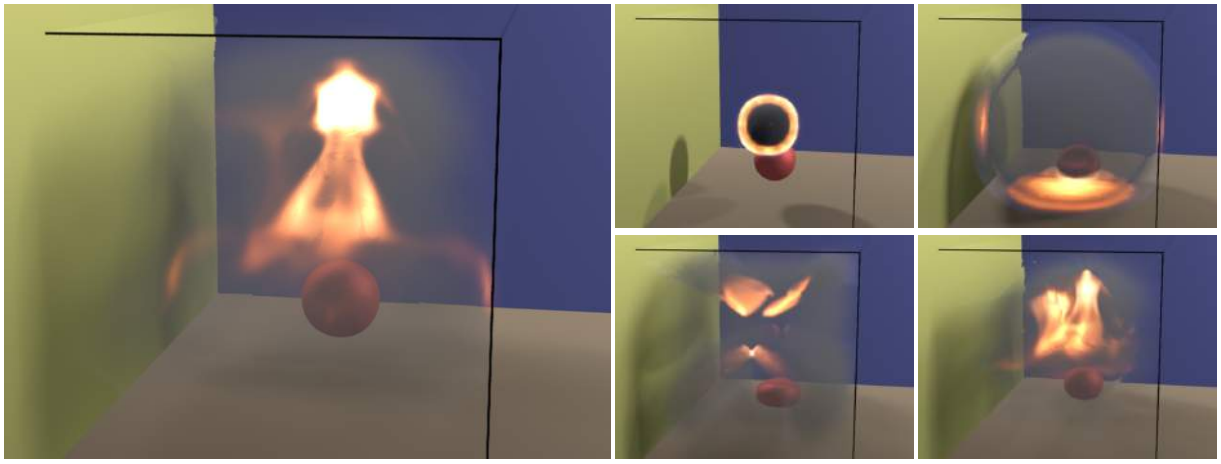

Figure 1: *A charge is detonated near a deformable ball. The ball compresses and bounces off the ground as it interacts with the shock. The soot heats up near the shock front and emits blackbody radiation. This was solved on a* $150 \times 100 \times 100$ *grid.*

continuous nature of compressible flow. This excludes the fast schemes typically used for incompressible simulation, such as semi-Lagrangian, BFECC or MacCormack advection [Sta99, DL03, KLLR05, SFK*08]. Moreover, the time step of a compressible flow simulation is constrained by the sound speed c in addition to bulk velocity; this severe restriction is necessary to properly resolve the shock wave and related phenomena, but is unduly limiting once these effects have left the domain of interest.

Shock and other compressible flow phenomena impose small time steps and therefore require a large amount of computational effort to simulate a mere fraction of a second. Other authors who have carried out these types of simulations show shocks moving around (in slow motion) etc., and stop their simulations/video after a short time. If they would have continued simulating, one would see more of the same, shocks moving, etc. for a few more fractions of a second. In the real world these shocks eventually dissipate as do the effects of compressibility, leading eventually to a plume type structure more representative of smoke and fire - governed more appropriately by incompressible flow. It is computationally infeasible for existing methods to simulate what happens to a flow field over 5-10 seconds when a large amount of computational resources are needed to advance a fraction of a millisecond.

Instead we propose to transition the flow from compressible to fully incompressible by sending $c \to \infty$. Non-physically driving the sound speed to ∞ accelerates the behavior of the fluid in order to obtain incompressible style flow phenomena such as rolling and plumes much quicker than one would otherwise attain. Any explicit method would have its time step driven to zero as the sound speed is driven to ∞, and therefore would not make any progress towards the incompressible flow behavior we are after. Thus a semi-implicit method such as [KSGF09] lends itself well to this approach as their formulation naturally yields the Godunov splitting methods intrinsic to incompressible flow. Once the flow is fully incompressible, there are many mature simulation techniques that can be used to enhance the visual fidelity and speed of traditional smoke simulations. Vorticity confinement [SU94] and vortex particles [SRF05] help to reduce the numerical viscosity introduced by fast, low-order advection schemes. Non-uniform mesh refinement techniques such as Octrees [LGF04] and RLE [HNB*06] grids permit faster simulations by discarding information away from the area of interest.

One of the main contribution of our paper is the ability to show both the initial states of the explosion including shock waves along with the long time behavior of rolling plumes and other incompressible flow effects. To the best of our knowledge this has not been previously addressed and other authors merely stop their simulations after shocks have moved around a little bit.

2. Euler Equations

We briefly describe the semi-implicit evolution of compressible flow which is based on [KSGF09]. Consider the multi-dimensional Euler equations

$$\begin{pmatrix} \rho \\ \rho\vec{u} \\ E \end{pmatrix}_t + \begin{pmatrix} \nabla \cdot \rho\vec{u} \\ \nabla \cdot (\rho\vec{u})\vec{u} \\ \nabla \cdot (E\vec{u}) \end{pmatrix} + \begin{pmatrix} 0 \\ \nabla p \\ \nabla \cdot (p\vec{u}) \end{pmatrix} = \vec{0} \qquad (1)$$

where ρ is the density, $\rho\vec{u}$ is the momentum, E is the total energy per unit volume and p is the pressure. Note that the total energy E is the sum of ρe, where e is the internal energy

Figure 2: *A planar shock enters an enclosed domain and disrupts a stack of rigid bodies. It reflects off of the back wall, hits the stack of objects again, and exits the domain. Times .0011s, .0018s, .003s, .0043s and .0053s are shown. The grid size is* $225 \times 150 \times 150$.

(a function of temperature) and kinetic energy $\rho\|\vec{u}\|^2/2$. The system of equations is closed with an equation of state (EOS) which defines pressure p as a function of ρ and e. The EOS is chosen to model the fluid in question–we use the gamma gas law $p = (\gamma - 1)\rho e$ since we are simulating compression and expansion of air, with an ideal gas constant $\gamma = 1.4$. The flux terms in Equation (1) have been split into advection and non-advection components. In one spatial dimension, the purely advective component has a Jacobian given by

$$\mathbf{J} = \begin{pmatrix} 0 & 1 & 0 \\ -u^2 & 2u & 0 \\ -\frac{Eu}{\rho} & \frac{E}{\rho} & u \end{pmatrix},$$

whose eigenvalues are all $|u|$. Since all the characteristic velocities are identical a component-wise upwinding can be used, avoiding the expensive transformation into characteristic variables typically done in compressible flow. Note that this explicit step imposes a time step restriction based on $|u|$, rather than the more prohibitive $|u| \pm c$ which arises when one uses a fully explicit method. We have used a second order ENO scheme in all of our examples unless otherwise mentioned. We denote the quantities obtained after integrating the advection terms as $\rho^\star$, $\rho\vec{u}^\star$ and $E^\star$. Note that pressure fluxes do not directly affect density, so $\rho^{n+1} = \rho^\star$.

The pressure component of flux is solved implicitly using a modified version of Poisson's equation. The momentum update can have the time derivative of density removed by using row 1 of Equation (1); then dividing by ρ^{n+1} and discretizing with a first order method in time, one obtains

$$\vec{u}^{n+1} - \vec{u}^\star = \Delta t \frac{\nabla p}{\rho^{n+1}}. \tag{2}$$

For incompressible flow one would set $\nabla \cdot \vec{u}^{n+1} = 0$, but for compressible flow one can get an estimate of $\nabla \cdot \vec{u}^{n+1}$ using the pressure evolution equation,

$$p_t + \vec{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \vec{u}. \tag{3}$$

Fixing $\nabla \cdot \vec{u}$ at time t^{n+1} and discretizing $p_t + \vec{u} \cdot \nabla p$ explicitly using a forward Euler time step, (i.e. $\frac{p^{n+1}-p^n}{\Delta t} + \vec{u}^n \cdot \nabla p^n$), and defining the advected pressure as $p^a = p^n - (\vec{u}^n \cdot \nabla p^n)\Delta t$ we obtain

$$\nabla \cdot \vec{u}^{n+1} = \frac{p^a - p^{n+1}}{\Delta t \rho c^2}. \tag{4}$$

Taking the divergence of Equation (2), substituting the value of $\nabla \cdot \vec{u}^{n+1}$ from Equation (4) and rearranging reveals an implicit equation for pressure,

$$\frac{p^{n+1}}{\rho^n (c^2)^n} - \Delta t^2 \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{p^a}{\rho^n (c^2)^n} - \Delta t \nabla \cdot \vec{u}^\star. \tag{5}$$

Discretizing the gradient and divergence operators yields

$$\left[\frac{I}{\rho^n (c^2)^n \Delta t^2} + G^T \frac{1}{\hat{\rho}^{n+1}} G \right] \tilde{p}^{n+1} = \frac{\tilde{p}^a}{\rho^n (c^2)^n \Delta t^2} + G^T \hat{\vec{u}}^\star, \tag{6}$$

where G is a discretized gradient operator and $-G^T$ is a discretized divergence operator, $\hat{\rho}$ and $\hat{u}$ represent variables interpolated to cell faces, and $\tilde{p}$, $\tilde{p}^a$ denote pressure quantities scaled by Δt. Note that the advected pressure p^a can be solved using a standard non-conservative method such as semi-Lagrangian advection. The initial p^n used for this advection is initialized from the EOS in order to avoid numerical drift, since we only solve Equation (6) to a tolerance rather than to floating point precision. One should note that in the limit as $c \to \infty$ this equation reduces to the standard incompressible Poisson equation.

This implicit solve yields pressures at cell centers, but in order to properly conserve momentum and capture correct shock speeds we need to apply the pressure in a flux-based manner and thus need p^{n+1} and $(p\vec{u})^{n+1}$ at cell faces. These are acquired by using a density-weighted averaging of pressure from the cell centers and setting $\hat{u}^{n+1}_{i+1/2} = \hat{u}^\star_{i+1/2} - \Delta t(\nabla p^{n+1}/\hat{\rho}^{n+1}_{i+1/2})$. Finally, we compute

$$(\rho\vec{u})^{n+1} = (\rho\vec{u})^\star - \Delta t \frac{p^{n+1}_{i+1/2} - p^{n+1}_{i-1/2}}{\Delta x} \tag{7}$$

and

$$E^{n+1} = E^\star - \Delta t \frac{(p\hat{u})^{n+1}_{i+1/2} - (p\hat{u})^{n+1}_{i-1/2}}{\Delta x}. \tag{8}$$

This approach to compressible flow simulation yields a

semi-implicit formulation with a time step restriction significantly more forgiving than traditional methods, being based on the bulk velocity of the flow and not the sound speed of the fluid. Furthermore we note that, being strongly diagonally dominant, the implicit solve typically converges after only a few passes of a fast iterative solver such as the conjugate gradient method.

3. Two-way coupling with solid bodies

Two-way coupling of fluids to rigid and deformable solid bodies is most commonly done by applying pressure forces on the solid from neighboring fluid nodes and applying velocity boundary constraints on the fluid velocity field. [KFCO06, CGFO06, BBB07, RMSG*08] proposed handling this coupling implicitly by modifying the pressure solve, albeit only for incompressible flow. [RMSG*08] introduces a matrix operator W which rasterizes solid degrees of freedom to fluid grid-cell faces in a conservative fashion, permitting a stable semi-implicit coupling. This method can be extended into the compressible regime, resulting in the following symmetric matrix equation for fluid pressures and solid velocities:

$$\begin{pmatrix} \frac{VI}{\Delta t^2 \rho c^2} + V G_f^T \frac{1}{\rho} G_f & -A_f B^T W \\ -W^T B A_f & -M_S + \Delta t D \end{pmatrix} \begin{pmatrix} \tilde{p}^{n+1} \\ V_S^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{V}{\Delta t^2 \rho c^2} \tilde{p}^a + V G_f^T u^\star \\ -M_S V_S^\star \end{pmatrix} \quad (9)$$

where V is the volume of a fluid grid cell, I is the identity matrix, A_f is the area of a fluid face, and B extrapolates cell-centered fluid quantities to neighboring coupled faces. M_S is the mass matrix of the solid, V_S are the solid velocity degrees of freedom, and D is the damping matrix which represents linearized implicit damping forces. Solving this symmetric system yields V_S^{n+1} and p^{n+1}, which must then be applied back to the conserved variables $\rho\vec{u}$ and E. At coupled faces we use $p_{i+1/2} = (Bp)_{i+1/2}$ and $\hat{u}_{i+1/2} = (WV_S^{n+1})_{i+1/2}$, then apply equations (7) and (8) as usual to get time t^{n+1} conserved quantities.

By treating the interactions between fluids and solids implicitly, we avoid introducing new stability concerns such as those which arise from standard two-way coupling methods, like the lumped-mass instability discussed in [CGN05]. This coupling approach is quite general, working for deformable bodies with arbitrary constitutive models and rigid bodies (for which the damping matrix $D = 0$).

4. Flow Regime Transition

One drawback of existing methods is that the small time steps required for simulation, coupled with the complexity of simulating compressible flow, result in simulations that are relatively short. Shock waves travel across the domain and a tiny plume starts to form, just before the simulation ends. Obviously, transitioning from compressible to incompressible flow allows one to take bigger time steps and show more of the interesting incompressible flow-style smoke effects which persist long after the shocks have exited the domain. Simply initializing an incompressible flow from the output of a compressible flow simulation leads to significant velocity discontinuities and unusable results as the compressible flow velocities can be far from divergence-free. Therefore one needs a smooth transition, which can be achieved by sending $c \to \infty$. Unfortunately when using an explicit time step, pushing c to ∞ drives the time step to zero and no progress can be made whatsoever. This is not a concern for a semi-implicit method.

As $c \to \infty$, the EOS decouples entirely from the solve and the pressure evolution equation becomes $\nabla \cdot \vec{u} = 0$, which is exactly incompressibility. This in turn decouples E from the simulation, and sends $\nabla \cdot (\rho\vec{u}) \to \vec{u} \cdot \nabla\rho$, giving the more familiar advection equations that drive incompressible flow. Most of the terms from Equation (5) vanish, leaving us with

$$\begin{cases} \vec{u}_t + \vec{u} \cdot \nabla\vec{u} + \frac{\nabla p}{\rho} = 0 \\ \nabla \cdot \vec{u} = 0 \end{cases} \quad (10)$$

It remains, then, to chose how to send $c \to \infty$. When a flow becomes incompressible it forcibly damps out discontinuities such as shock waves, potentially causing drastic changes in the flow field. Consider the speed of a shock U, given for a gamma-law gas [Lig01] as

$$U = \left(1 + \frac{\gamma+1}{2\gamma} \frac{p_1 - p_0}{p_0}\right)^{\frac{1}{2}} c_{\text{EOS}}, \quad (11)$$

where c_{EOS} is the sound speed as determined by the EOS (as opposed to c, which we artificially accelerate). By artificially driving $c \to \infty$ over an interval of time (t_s, t_f), we force shock waves to travel faster and faster, effectively dispersing them before going fully incompressible. Equation (9) only contains $(1/c)$ terms, so it is more convenient numerically to send this term to 0. A naïve approach might linearly interpolate between $1/c_{\text{EOS}}$ and 0, however this simply does not accelerate the sound speed sufficiently fast, being only a $10\times$ amplification by the time we are 90% through the tran-

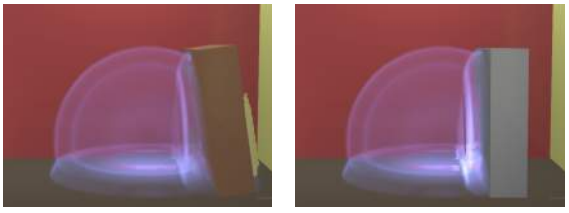

Figure 4: *A shock interacts with a light wall (left) and a heavy wall (right) respectively, at $t = 0.316s$. Note how the shock passes through the light wall, and strongly reflects off of the heavy wall. The grid resolution is $225 \times 150 \times 150$.*

Figure 3: *A charge is detonated within a small four-walled chamber at $t = 0$. Shown on the left is a smoke plume at $t = 0.04s$, and on the right its development, at times .005s, .02s, .03s and .115s. This was solved on a $200 \times 300 \times 200$ grid.*

sition. Experimentally, we found however that

$$(1/c)(t) = \frac{1}{c_{\text{EOS}}} \left(1 - \frac{t - t_s}{t_f - t_s}\right)^3 \tag{12}$$

gives reasonable results. Note that since we are handling the acoustic component of the flow implicitly, this artificial acceleration of shock waves does not affect the stability of our method.

Once the flow has fully transitioned, we can switch to using a traditional incompressible flow solver and take advantage of the rich body of literature which has been invested in making fast, visually stunning smoke simulations.

As we artificially change c during our transition, the EOS pressure becomes increasingly inaccurate estimate of p^n. One could alleviate this by keeping the pressure from the previous time step rather than reinitializing p^n, however this choice does not appear to have any effect on the flow, likely because the contribution from p^n vanishes rapidly, disappearing entirely from the governing equations in the limit. Instead we prefer to reinitialize p^n from the EOS until the flow is fully transitioned.

5. Combustion

The method described above models compressible flow and the related non-linear phenomena like shocks. However, a lot of visual detail in explosions also comes from chemical reactions due to burning of fuel. We follow an approach similar to [FOA03] for modeling combustion. We track fuel in the domain by using a passively advected scalar fuel field F, defined as the fraction of mass in the cell that is fuel, using

$$F_t + \vec{u} \cdot \nabla F = S_F, \tag{13}$$

where S_F denotes the source terms. If the temperature at a cell i is greater than the ignition point T_I of the fuel, the fuel will burn at a specified rate b, i.e. $S_F(i) = -b/\rho$ for a cell with temperature greater than T_I and 0 otherwise. The burning fuel generates heat at a rate given by $r_h b$, where r_h is the calorific value of the fuel. This generation of heat is easily accounted for in Euler's equations by adding $r_h b$ as an energy source term to the right-hand side of the bottom row of Equation (1). Note that this source term will increase the internal energy/temperature of the compressible fluid, which will in turn increase pressure causing an expansion. This phenomena was modeled in [FOA03] by adding an artificial divergence to their incompressible flow solver yielding impressive results; however, our semi-implicit compressible flow formulation allows us to model this expansion due to burning fuel in a more physical manner.

Another secondary effect of burning fuel is the generation of carbon particles or soot. We model soot by tracking the soot field C, defined as the fraction of mass in the cell that is soot, via

$$C_t + \vec{u} \cdot \nabla C = S_c, \tag{14}$$

where S_c denotes the source terms. The soot generated by burning fuel can be modeled by setting $S_c = r_c b/\rho$, where r_c denotes the mass of soot produced per unit mass of fuel burnt. We initialize both the soot and fuel to be non-zero at the detonation site and zero everywhere else. Note that ρC is the total soot in a control volume, ρF is the total fuel and $\rho(1 - C - F)$ is air.

Figure 5: *A charge is detonated within a small four-walled chamber with a fragile wall at $t = 0$. Shown on the left is a smoke plume at $t = 0.04s$, and on the right its development, at times .005s, .02s, .03s and .115s. This was solved on a $200 \times 300 \times 200$ grid.*

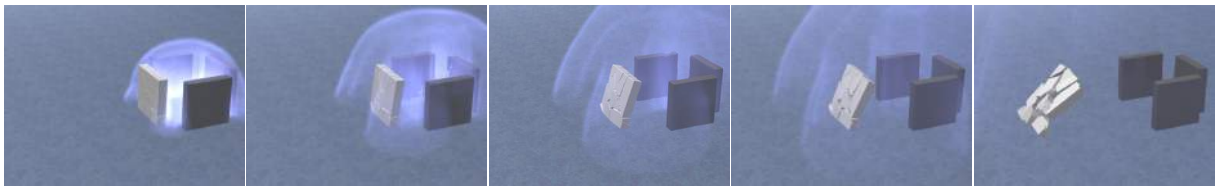

Figure 6: *A charge detonates, fracturing a fragile wall. Shown are $t = .0005s$, $t = .0010s$, $t = .0015s$, $t = .0020s$, $t = .0040s$.*

6. Fracture

Our stable, two-way, strongly coupled methodology lends itself naturally to computing explosive phenomena like fracture. Like [YOH00], we have access to the pressure forces (computed as $W^T B A_f p^{n+1}$), which can then be plugged into existing fracture codes such as the one proposed in [SSF09] to produce visually stunning special effects. We demonstrate this by incorporating said fracture framework, which uses fracture patterns, a threshold and a point of impact to break an object into debris. One could designate the point of impact by searching through the fluid faces coupled to a given solid and choosing the face whose pressure force is maximized. However, as our geometries are simple, we simply cast a ray back along the direction of force and use that intersection with the body surface as the point of impact.

Figure 7: *Smoke plumes which result from a detonation within an enclosed chamber (left), and from a detonation within a chamber whose front left wall is fragile (right).*

7. Rendering

Soot and Heat: We use a standard volumetric smoke rendering [FSJ01] algorithm for the visualization of soot. The soot also emits light with intensity proportional to its density, and color given by blackbody radiation.

Shock Fronts: Shocks are detected in the flow field by examining $|\nabla p|$ and noting any location where the gradient is above some threshold to be the location of a shock front. They are then used to refract light (as in [YOH00]), using $\nabla p/|\nabla p|$ as the surface normal and bending rays according to the difference in refraction index η across the shock front. This effect can have a dramatic visual effect on the simulation, as in Figure 8 where the strength of a nuclear explosion bends light significantly. In a more typical scenario, The effect is subtle but distinct (as seen in Figure 1). We fur-

ther enhance the visual impact of the shock by adding a blue emittance which scales with $|\nabla p|$, demonstrated in Figure 2.

8. Results

We simulate air as an ideal gas with $\gamma = 1.4$, with a rest state temperature of $T_{atm} = 290\ K$, zero initial velocity, and pressure of $p_{atm} = 1.01325 \times 10^5\ Pa$, or atmospheric pressure. This gives a fluid of density $\rho = 1.4\ kg/m^3$, comparable to that of air. Unless otherwise noted we initialize a shock by instantaneously depositing a high internal energy into an initial blast location, corresponding to a temperature of $10 \times T_{atm}$ and pressure $10^3 \times p_{atm}$. Boundary conditions are set to be atmospheric, permitting shocks to smoothly flow out of the domain.

All of our simulation are run with second order ENO [SO88] and third order Runge-Kutta. The examples took between 30 minutes to several hours on our unoptimzied research code with a lot of I/O. For the purposes of comparison we set up a simulation similar to Figure 11 in [SGTL09] and Figure 2 in [YOH00]. The explicit version of our code ran in 4 minutes and 24 seconds and the semi-implicit one in 3 minutes 11 seconds, which is comparable to the numbers reported in [SGTL09]. Even though the semi-implicit method was faster, if one only cares about the short time simulations with no rolling smoke, etc, explicit methods are just fine.

Trinity Test: Figure 8 shows a simulation of the trinity test of 1945, which we model by depositing an initial temperature of $2.62497 \times 10^8\ K$ and pressure of $9.41831 \times 10^{10}\ Pa$ into an initial blast of radius $6.5\ m$. This corresponds to approximately $90\ KJ$ worth of internal energy being introduced to the simulation.

Enclosed Detonation: We show in Figure 3 a detonation that goes off in an enclosed blast chamber composed of four massive walls, as suggested by [SGTL09]. The detonation drives air out the top of the chamber and through the small openings at the four corners. After the initial shock waves exit the domain, we transition the flow from compressible to incompressible over the time interval $t \in (.15, .16)$, and simulate the resulting smoke plume using a traditional incompressible flow solver, incorporating vorticity confinement.

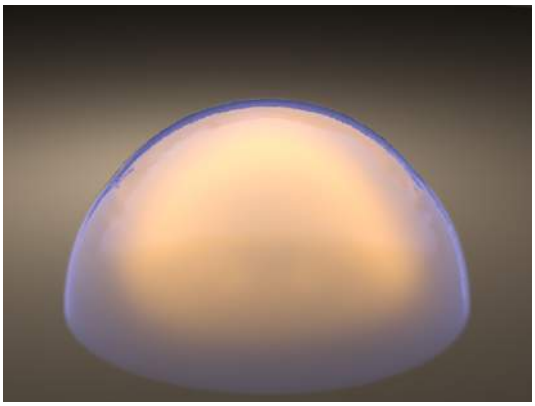

Figure 8: *The 1945 Trinity Test, simulated on a* $200 \times 100 \times 200$ *grid.*

Shock Hitting Smoke: Transitioning from incompressible flow to compressible flow is a relatively easy task, and can be done by setting ρ, T and p to their atmospheric values at the time of transition. To show this we create a smoke plume and then hit it with a shock wave. The results are shown in Figure 10. The smoke plume is driven by buoyancy dynamics, but as the effects of buoyancy are vanishingly small in the time scale of the shock wave we neglect them while the flow is compressible.

Shock Affecting a Light/Heavy Solid: In Figure 4 we show a shock interacting with a heavy object, and a shock interacting with a light object. The shock mostly reflects off of the heavy object, generating a strong secondary shock that reflects off the wall. The light object instead absorbs most of the shock wave, rather than reflecting it. Once the light object collides with the static right wall, it creates a secondary shock due to the sudden change in velocity.

Shock Driving a Stack of Rigid Bodies: Figure 2 shows a planar shock wave interacting with a stack of rigid bodies, reflecting off of a wall, and hitting them again before exiting the domain.

Shock Interacting with a Deformable Body: The two-way coupling technique we use is quite general, and works with deformable bodies with arbitrary constitutive models in addition to the rigid bodies shown above. Figure 1 shows a shock interacting with a deformable ball which is modeled as a mass-spring system. It has 21528 elements, edge springs with $k = 10^4\ N/m$, and we use altitude springs with $k = 10^4\ N/m$.

Enclosed Detonation with a Fragile Wall: The two-way coupled effects of interacting solids and fluids are demonstrated in Figure 5 and Figure 6, where one of the walls from Figure 3 is replaced with a light wall that fractures as a result of pressure forces. The resulting smoke plumes are compared in Figure 7. After the initial shock waves exit the domain, we transition the flow from compressible to incompressible over the time interval $t \in (.15, .16)$, and simulate the resulting smoke plume using a traditional incompressible flow solver, incorporating vorticity confinement.

Cannon Fired at a Bunny: In Figure 9 we use an explosive charge to fire a cannonball at an unsuspecting bunny. The cannonball is initially at rest in the chamber of the cannon, creating a seal separating the high-energy blast charge from the outside air. This charge is detonated at $t = 0$ and accelerates the cannonball to a velocity over $1.5 km/s$. As the cannonball exits the barrel it is followed by a dynamic, automatically-generated cloud of soot, fuel and fire.

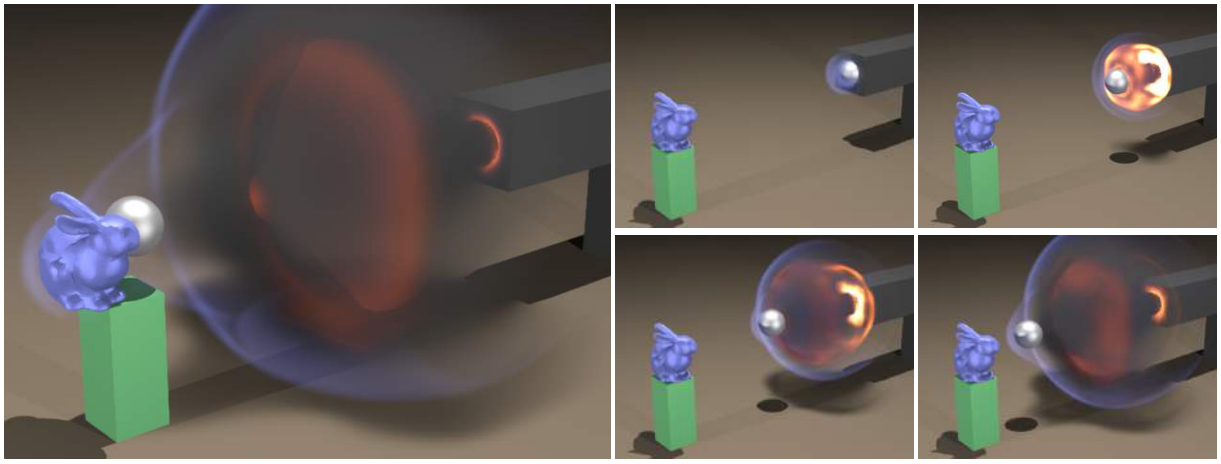

Figure 9: *A cannonball is accelerated by an explosion in the barrel of the cannon. It reaches a super-sonic speed of* $1500\ m/s$*, and generates a secondary shock wave as it compresses the air in front of it. The grid resolution is* $300 \times 120 \times 120$*.*

Figure 10: *After a smoke plume develops, it is hit by a planar shock. Times shown are* $.83s$*,* $1.67s$*,* $2.0835s$*,* $2.0836s$*,* $2.0838s$*. The grid resolution is* $512 \times 256 \times 256$*.*

9. Conclusions

We present a novel approach to incorporate the ability to handle both the initial states of an explosion (including shock waves) along with the long time behavior of rolling plumes and other incompressible flow effects. Our method handles compressible flow in a semi-implicit manner, permitting the fast and stable simulation of complex dynamical phenomena, including shock waves and combustion. It supports two-way coupled interactions in a way that permits the integration of complex solid-fluid interactions such as fracture.

Acknowledgements

Research supported in part by ONR N0014-06-1-0393, ONR N00014-06-1-0505, ONR N00014-05-1-0479 for a computing cluster, NIH U54-GM072970, NSF ACI-0323866, and King Abdullah University of Science and Technology (KAUST) 42959.

References

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

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

[CGN05] CAUSIN P., GERBEAU J.-F., NOBILE F.: Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comp. Meth. Appl. Mech. Engng. 194*, 42-44 (2005).

[DL03] DUPONT T., LIU Y.: Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *J. Comput. Phys. 190/1* (2003), 311–324.

[FM97] FOSTER N., METAXAS D.: Controlling fluid animation. In *Comput. Graph. Int.* (1997), pp. 178–188.

[FOA03] FELDMAN B., O'BRIEN J., ARIKAN O.: Animating suspended particle explosions. *ACM Trans. Graph. (SIGGRAPH Proc.) 22*, 3 (2003), 708–715.

[FSJ01] FEDKIW R., STAM J., JENSEN H.: Visual simulation of smoke. In *Proc. of ACM SIGGRAPH 2001* (2001), pp. 15–22.

[HNB*06] HOUSTON B., NIELSEN M., BATTY C., NILSSON O., MUSETH K.: Hierarchical RLE level set: A compact and versatile deformable surface representation. *ACM Trans. Graph. 25*, 1 (2006), 1–24.

[HSF07] HONG J., SHINAR T., FEDKIW R.: Wrinkled flames

and cellular patterns. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 47.

[IJN09] IMAGIRE T., JOHAN H., NISHITA T.: A fast method for simulating destruction and the generated dust and debris. *The Visual Computer 25*, 5 (2009), 719–727.

[IKC04] IHM I., KANG B., CHA D.: Animation of reactive gaseous fluids through chemical kinetics. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2004), pp. 203–212.

[KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. *ACM Trans. Graph. 25*, 3 (2006), 820–825.

[KJI07] KANG B., JANG Y., IHM I.: Animation of chemically reactive fluids using a hybrid simulation method. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), Eurographics Association, p. 208.

[KLLR05] KIM B.-M., LIU Y., LLAMAS I., ROSSIGNAC J.: Using BFECC for fluid simulation. In *Eurographics Workshop on Natural Phenomena 2005* (2005).

[KSGF09] KWATRA N., SU J., GRÉTARSSON J. T., FEDKIW R.: A method for avoiding the acoustic time step restriction in compressible flow. *J. Comput. Phys. 228*, 11 (2009), 4146–4161.

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.) 23* (2004), 457–462.

[Lig01] LIGHTHILL J.: *Waves in fluids*. Cambridge Univ Pr, 2001.

[MMA99] MAZARAK O., MARTINS C., AMANATIDES J.: Animating exploding objects. In *Proc. of Graph. Interface 1999* (1999), pp. 211–218.

[NF99] NEFF M., FIUME E.: A visual model for blast waves and fracture. In *Proc. of Graph. Interface 1999* (1999), pp. 193–202.

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

[RNGF03] RASMUSSEN N., NGUYEN D., GEIGER W., FEDKIW R.: Smoke simulation for large scale phenomena. *ACM Trans. Graph. (SIGGRAPH Proc.) 22* (2003), 703–707.

[SFK*08] SELLE A., FEDKIW R., KIM B., LIU Y., ROSSIGNAC J.: An unconditionally stable MacCormack method. *Journal of Scientific Computing 35*, 2 (2008), 350–371.

[SGTL09] SEWALL J., GALOPPO N., TSANKOV G., LIN M.: Visual simulation of shockwaves. *Graphical Models* (2009).

[SMML07] SEWALL J., MECKLENBURG P., MITRAN S., LIN M.: Fast fluid simulation using residual distribution schemes. In *Eurographics Workshop on Natural Phenomena* (2007), pp. 47–54.

[SO88] SHU C.-W., OSHER S.: Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys. 77* (1988), 439–471.

[SRF05] SELLE A., RASMUSSEN N., FEDKIW R.: A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. (SIGGRAPH Proc.) 24*, 3 (2005), 910–914.

[SSF09] SU J., SCHROEDER C., FEDKIW R.: Energy stability and fracture for frame rate rigid body simulations. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2009), pp. 155–164.

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

[SU94] STEINHOFF J., UNDERHILL D.: Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings. *Phys. of Fluids 6*, 8 (1994), 2738–2744.

[TOT*03] TAKESHITA D., OTA S., TAMURA M., FUJIMOTO T., MURAOKA K., CHIBA N.: Particle-based visual simulation of explosive flames. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on* (2003), pp. 482–486.

[YOH00] YNGVE G., O'BRIEN J., HODGINS J.: Animating explosions. In *Proc. SIGGRAPH 2000* (2000), vol. 19, pp. 29–36.

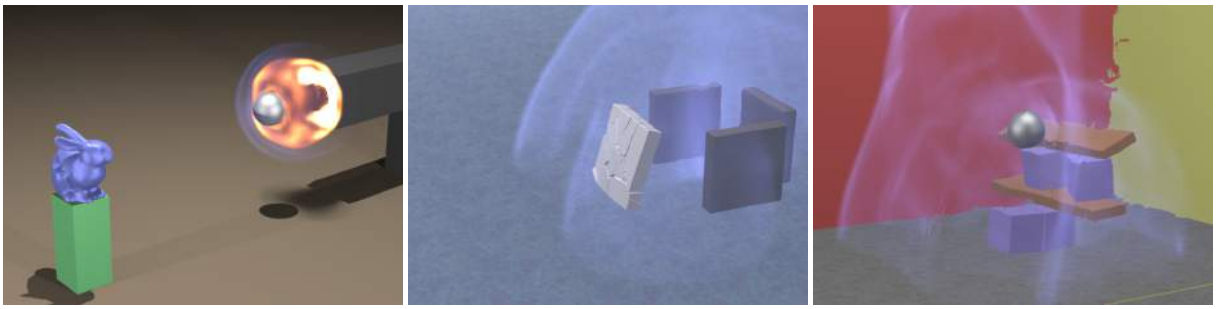

Figure 11: *Compressible flow interacting with a variety of objects. (left) A bunny faces down a super-sonic cannonball, (middle) a detonation goes off in a four-walled chamber with a fragile wall, and (right) a shock tosses a stack of rigid bodies around.*

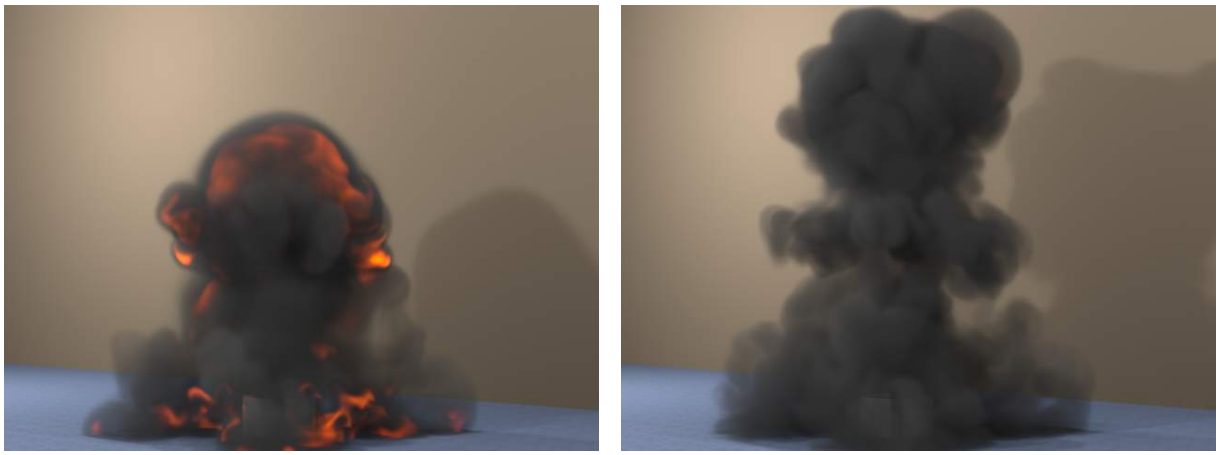

Figure 12: *A charge is detonated within a small four-walled chamber at* $t = 0$*. Shown are the smoke plumes at* $t = 0.04$*s, and* $t = .115$*s. This was solved on a* $200 \times 300 \times 200$ *grid.*