

A Vortex Particle Method for Smoke, Water and Explosions

Andrew Selle*
Stanford University
Intel Corporation

Nick Rasmussen†
Industrial Light + Magic

Ronald Fedkiw‡
Stanford University
Industrial Light + Magic

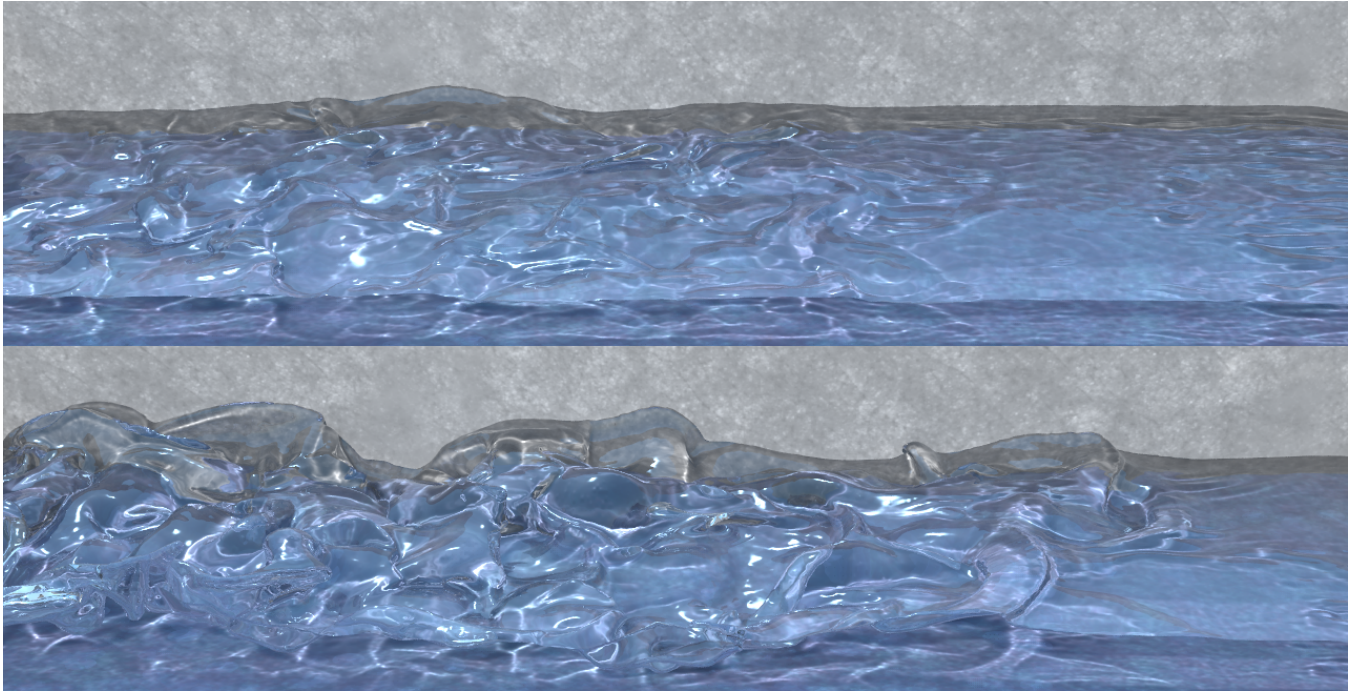


Figure 1: Vortex particles seeded at the inflow (left) create turbulence in water flowing from left to right. The top and bottom show lower and higher amounts of particle induced vorticity. ($320 \times 128 \times 320$ effective resolution octree grid, approximately 600 vortex particles)

Abstract

Vorticity confinement reintroduces the small scale detail lost when using efficient semi-Lagrangian schemes for simulating smoke and fire. However, it only amplifies the existing vorticity, and thus can be insufficient for highly turbulent effects such as explosions or rough water. We introduce a new hybrid technique that makes synergistic use of Lagrangian vortex particle methods and Eulerian grid based methods to overcome the weaknesses of both. Our approach uses vorticity confinement itself to couple these two methods together. We demonstrate that this approach can generate highly turbulent effects unachievable by standard grid based methods, and show applications to smoke, water and explosion simulations.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling;

Keywords: vortex methods, fluids, smoke, water, explosions

*e-mail: aselle@graphics.stanford.edu

†e-mail: nick@ilm.com

‡e-mail: fedkiw@cs.stanford.edu

1 Introduction

While the numerical simulation of fluids is now common in the special effects industry, highly turbulent phenomena such as explosions remain challenging. It is difficult to resolve these effects even on the highest resolution grids using state of the art techniques. Regardless, directors frequently desire these exciting and compelling effects, and filming them practically is not always possible especially when complex camera motions (such as flying through an explosion) are required.

The recent popularity of computer graphic smoke simulation using the three dimensional Navier-Stokes equations approximately began with [Foster and Metaxas 1997]. The practicality of this was enhanced through the introduction of semi-Lagrangian advection techniques in [Stam 1999] and vorticity confinement in [Fedkiw et al. 2001] (see also [Steinhoff and Underhill 1994]). Despite the usefulness of this approach, some major drawbacks remain. For example, a three dimensional computational grid requires a lot of memory, so it can be difficult to simulate large scale phenomena. Also, vorticity confinement can only amplify *existing* grid vorticity, so if the resolution of the grid is not fine enough to capture object interaction, combusting fuel pockets, upwelling, etc., vorticity confinement cannot recover them.

[Losasso et al. 2004] simulated smoke and water on an octree grid addressing the memory requirements to some degree, however it can be difficult to choose refinement criteria that ensure adequate grid resolution everywhere interesting flow might develop. And

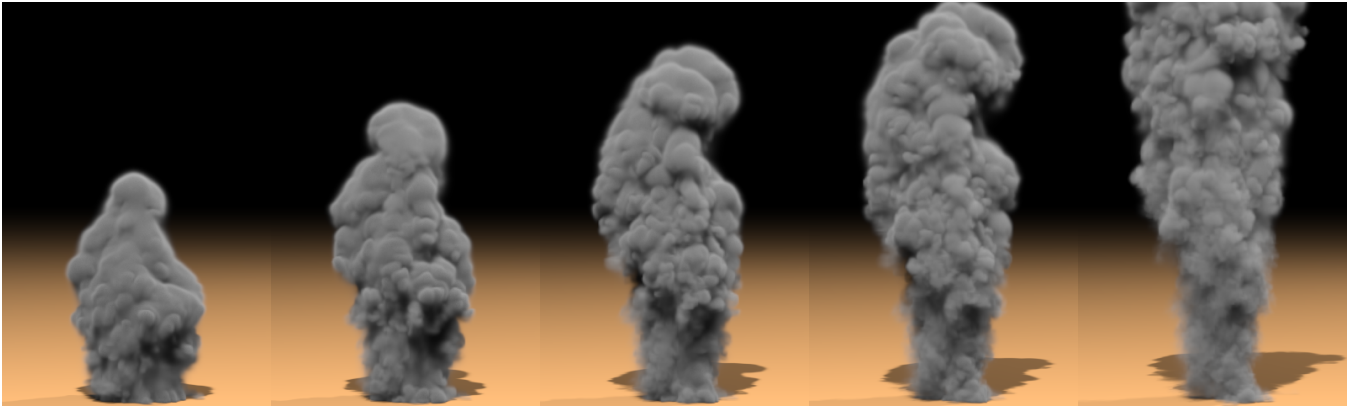


Figure 2: Time evolution of a smoke explosion enhanced with vortex particles seeded as the smoke undergoes expansion at the source. ($180 \times 260 \times 180$ uniform grid, approximately 6000 vortex particles)

if the refinement criteria are poor, small scale detail will never be formed and vorticity confinement cannot amplify it. [Rasmussen et al. 2003] introduced a method for simulating large scale explosions that avoids the high memory requirements of three dimensional grids by simulating a series of two dimensional slices that are placed in three dimensional space and used to define a wind field to advect particles. The technique produced impressive nuclear explosions, but is not as applicable to problems that have less inherent symmetry. Moreover, interesting phenomena such as fuel pocket combustion, etc. cannot be modeled in the free space between slices where interpolation is relied on to generate the velocity field.

Particle methods such as SPH (e.g. [Desbrun and Cani 1996; Hadap and Magnenat-Thalmann 2001; Premoze et al. 2003; Muller et al. 2003]) avoid the memory requirements of a three dimensional grid, but exhibit other difficulties such as the cost of finding the nearest neighbors, complications involved with enforcing incompressibility, particle redistribution, etc. Another class of particle methods are the vortex methods which are based on the curl of the Navier-Stokes equations, i.e. the vorticity. This form of the equations was solved with a fully grid based method in [Yaeger and Upson 1986], and a more typical particle based approach was considered in [Gamito et al. 1995]. Neither of these approaches treated obstacles and both were limited to two spatial dimensions. Particle based vortex methods suffer from some of the same issues as SPH methods, see e.g. [Lindsay and Krasny 2001; Ploumhans et al. 2002], and authors such as [Walther and Koumoutsakos 2001; Cottet and Poncet 2003] have worked to alleviate a number of these with the use of a background grid. Typically, the vorticity is mapped from the particles to a background grid where a *vector* valued Poisson equation is solved (this is only a scalar equation in two spatial dimensions as in [Gamito et al. 1995]), and the results are used to calculate the velocity and incorporate the effects of vortex stretching before transferring the velocity field back to the particles for advection.

The advantage of vortex methods is that the particles carry the vorticity, and its values can be preserved to compute inviscid, high Reynolds number turbulent flows, i.e. one avoids the grid based damping artifacts that vorticity confinement works to reduce. In addition, particle methods are optimal from a memory storage standpoint for adaptively resolving a flow field. The major disadvantages are in finding boundary conditions for the *vector* valued Poisson equation especially for moving and deforming solid objects, dealing with particle redistribution techniques to adequately represent and resolve the flow, and difficulties associated with the vortex stretching term (that happens to be identically zero in two spatial dimensions as in [Gamito et al. 1995]). Solving the standard veloc-

ity and pressure form of the Navier-Stokes equations alleviates all of these difficulties at the cost of increased numerical dissipation. [Drela and Murman 1987] and [Felici and Drela 1990] proposed coupling these techniques together in two and three dimensions, respectively, with what they referred to as “ad hoc” techniques. Some of the problems with their coupling procedures were discussed in [Felici and Drela 1993a; Felici and Drela 1993b].

We also propose solving both sets of equations, but in a more fully hybridized manner. The grid based velocity obtained from the Navier-Stokes equations is used to both advect the particles and to spin them as dictated by the vortex stretching term, while vorticity confinement is used to convey vorticity information to the particles to the grid. In particular, vorticity confinement acts in a manner that conserves the vorticity of the flow, thus providing visually appealing results. We demonstrate the ability to generate very turbulent effects that cannot be achieved with vorticity confinement alone.

2 Previous Work

Besides those works mentioned above, a number of authors simulated the equations of fluid dynamics (and variants) before [Foster and Metaxas 1997], see e.g. [Kajiya and von Herzen 1984; Kass and Miller 1990; Chen and Lobo 1994]. There have also been many works since then, including the proposed hybridization of particle and grid based methods to simulate water [Foster and Fedkiw 2001; Enright et al. 2002], and augmentations of the equations to model fire [Lamorlette and Foster 2002; Nguyen et al. 2002], clouds [Miyazaki et al. 2002], particle explosions [Feldman et al. 2003], chemically reacting gases [Ihm et al. 2004], and viscoelastic fluids [Goktekin et al. 2004]. Other interesting work includes control methodologies [Treuille et al. 2003; McNamara et al. 2004; Fattal and Lischinski 2004], flow on surfaces [Stam 2003], the use of advected radial basis functions for editing simulation results [Pighin et al. 2004], and the use of the compressible version of the equations to simulate explosions [Neff and Fiume 1999; Yngve et al. 2000].

3 Grid Based Method

The incompressible Navier-Stokes equations can be written as

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p / \rho = \mu \nabla^2 \mathbf{u} + \mathbf{f} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

with velocity $\mathbf{u} = (u, v, w)$, pressure p , density ρ , and \mathbf{f} representing buoyancy, vorticity confinement, etc. In particular, the vorticity confinement force is computed by taking the curl of the velocity field to obtain the vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, computing normalized vorticity location vectors $\mathbf{N} = \nabla|\boldsymbol{\omega}|/|\nabla|\boldsymbol{\omega}||$, and then applying a force $\mathbf{f} = \epsilon h(\mathbf{N} \times \boldsymbol{\omega})$ scaled by the size of the grid h and a strength parameter ϵ . We solve the inviscid ($\mu = 0$) form of these equations on either a uniform or octree grid as in [Fedkiw et al. 2001; Losasso et al. 2004].

4 Vortex Particle Method

The Navier-Stokes equations can be put into vorticity form by taking the curl of equation 1 to obtain

$$\boldsymbol{\omega}_t + (\mathbf{u} \cdot \nabla) \boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} = \mu \nabla^2 \boldsymbol{\omega} + \nabla \times \mathbf{f} \quad (3)$$

where the velocity advection term has been split into a vorticity advection term $(\mathbf{u} \cdot \nabla) \boldsymbol{\omega}$ and a vortex stretching term $(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}$. Note that the pressure term vanishes for constant density fluids.

Although these equations can be solved on a grid, particle based methods have the distinct advantage of avoiding grid based numerical dissipation that smears out the flow making it appear more viscous. In our implementation, each vortex particle stores a vorticity value $\boldsymbol{\omega}$ which includes both a magnitude and direction. A kernel (we use a clamped Gaussian with compact support or a tent function) is used to define the vorticity in a region of space nearby the particle. Given a collection of particles, the vorticity at a point is defined by summing the contributions from all nearby particles. The flow evolves as the particles move around and their vorticity values change. For example, viscous flow strongly dissipates large velocity gradients according to the $\mu \nabla^2 \boldsymbol{\omega}$ term. This is typically implemented with some sort of particle exchange method or with the aid of a background grid. However, our goal in using particle based methods is to eliminate dissipation, so we ignore this term solving the inviscid form of the equations similar to our approach to solving equation 1.

The solution of equation 3 requires a velocity field, which can be determined from the vorticity values stored on the individual particles. This is typically a rather complex process. Even with the aid of a background grid, one has to solve a *vector* valued Poisson equation and deal with complicated boundary conditions. One of the major benefits of our approach is that this step can be avoided entirely, as we instead use the velocity field determined by solving equations 1 and 2 which only requires the solution of a simple *scalar* Poisson equation with straightforward boundary conditions. Moreover, a standard vortex method needs to carefully place particles to resolve the flow. However, our technique does not require perfect distribution (and redistribution) of particles, because the grid based method adequately resolves the flow at least as well as in [Fedkiw et al. 2001]. Our vortex particles just provide increased details where they happen to exist. Thus we did not need to redistribute or reseed particles for any of our examples. This is a major contribution of using the grid based solver to determine the velocity field.

Given the velocity field, \mathbf{u} , determined via the grid based method, trilinear interpolation is used to define a velocity for advecting each particle. This accounts for the $(\mathbf{u} \cdot \nabla) \boldsymbol{\omega}$ term in equation 3. We typically inject particles with random initial vorticity at a uniform rate at a source, and let them passively advect through the flow. However, particles could also be created on the fly either near objects or near concentrations of high vorticity, and given the initial vorticity of the surrounding flow. Another nice feature of our approach is that

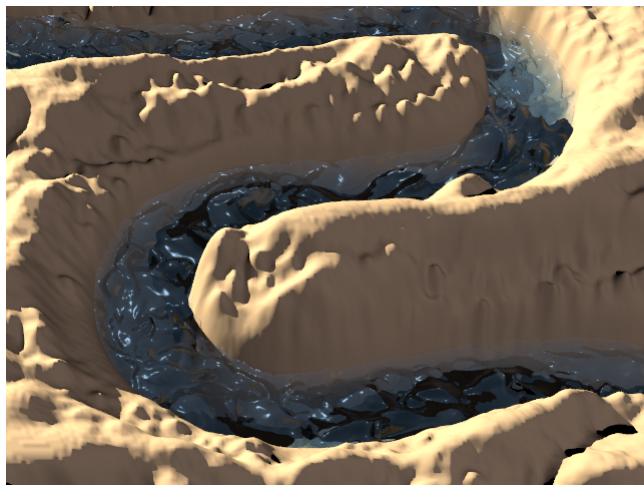


Figure 3: Vortex particles interact with complex geometry creating a turbulent water stream. ($272 \times 112 \times 272$ effective resolution octree grid, approximately 800 vortex particles)

the grid based solver creates a velocity field with proper boundary conditions. And since the particles are advected with that velocity field, they tend to avoid interpenetration with obstacles. However, if particles do enter solid geometry, we could delete them or project them back out of the object using an object level set. Since we use a high density of particles (typically thousands), either option suffices.

Besides advecting the particles, we need to consider the effects of the vortex stretching term in equation 3. This is done by computing the derivatives of the velocity field on the grid with central differences, trilinearly interpolating them to the particle location, and then augmenting the vorticity on the particle with $\boldsymbol{\omega} += \Delta t (\boldsymbol{\omega} \cdot \nabla) \mathbf{u}$. In isolation, this term can be thought of as an ordinary differential equation (ODE) that changes both the magnitude and direction of the particle’s vorticity. Unfortunately, the vorticity magnitude can exponentially increase when the ODE has a positive eigenvalue based on the fluid velocity gradient. To ensure stability one could clamp the magnitude, only allow it to decrease, etc. However, since the goal of our particle based method is to preserve vorticity concentration, we rescale the final vorticity to preserve its magnitude in all of our simulations. In that case, the effect of this term is to *spin* the particle’s vorticity vector without affecting its magnitude. This limits the numerical accuracy of the vortex particle method, but is consistent with our reliance on the the grid based method to provide most of the bulk flow features with the vortex particles providing an extra level of detail via vorticity concentration preservation. Along the same lines, we completely ignore the $\nabla \times \mathbf{f}$ term noting that forces (such as buoyancy) still have influence as they affect the velocity field via equation 1.

5 Vorticity Forcing

Equation 3 can be rewritten in conservation form

$$\boldsymbol{\omega}_t^T + \nabla \cdot (\mathbf{u} \boldsymbol{\omega}^T - \boldsymbol{\omega} \mathbf{u}^T - \mu (\nabla \boldsymbol{\omega})^T - \mathbf{f}^*) = 0 \quad (4)$$

where we have written the equations in row instead of column form, and \mathbf{f}^* is the skew symmetric cross product matrix based on \mathbf{f} . This equation demonstrates that vorticity should be conserved (neither created nor destroyed), highlighting one of the major problems with the work of [Felici and Drela 1990]. They used an “ad hoc” method to transmit the vorticity from the particles to the grid based

velocity field that does not conserve the total vorticity of that velocity field, i.e. they change the values of the grid based velocity without regard for vorticity conservation. We believe that vorticity conservation is what leads to better quality, especially *visual* quality. Without this, fluid swirling, etc., seems to appear magically. Our key innovation is to use the force \mathbf{f} in equation 1 to drive the grid based velocity field towards the desired vorticity. Although equation 4 dictates that *all* body forces conserve vorticity, the vorticity confinement force is the only one we know of that can introduce vorticity in the fashion required.

The simplest approach is to use the particles' vorticity magnitude *only* (ignoring direction) to define a spatially varying confinement strength ϵ , transferring the particles values of this parameter to the grid with the distribution kernel mentioned above. This allows vorticity confinement to be activated independent of the existing grid based vorticity, but ignores the directional component of the particle's vorticity. Even this simple approach readily creates visually rich phenomena difficult to obtain with vorticity confinement alone, and we used it early on in a production pipeline to create many explosion effects for a feature film, see Figure 4.

A promising technique is to form an analytic confinement force independently for each particle. The distribution kernel, $\xi_p(\mathbf{x} - \mathbf{x}_p)$, for a particle together with the particle vorticity, ω_p , defines an analytic vorticity $\tilde{\omega}_p(\mathbf{x}) = \xi_p(\mathbf{x} - \mathbf{x}_p)\omega_p$. Choosing a kernel that is rotationally symmetric and strictly decreasing with distance from the particle center implies that $\mathbf{N}_p(\mathbf{x}) = (\mathbf{x}_p - \mathbf{x})/\|\mathbf{x}_p - \mathbf{x}\|$, and the confinement force is then $\mathbf{F}_p(\mathbf{x}) = \epsilon_p(\mathbf{N}_p \times \tilde{\omega}_p)$. We can sum the contributions from all the particles to obtain a grid based force field for use in equation 1. This technique was used to generate Figures 1, 2 and 3. In addition, one can interpolate the grid based vorticity to the particle location and reduce the strength of the particle based force as the grid based vorticity approaches the particle's vorticity. Of course, in practice the grid is typically too coarse for the grid vorticity to match the vorticity of all the particles. Alternatively, one could transfer the magnitude *and* direction of the particle's vorticity to the grid, and compare this to the existing grid based vorticity. The difference between these can be used to calculate a vorticity confinement force (replacing vorticity with this difference in the formulas). However, we have not found these last two options to be necessary.

Finally, we note that vorticity confinement is rather robust for reasonably well chosen parameter values, but can destroy a simulation or cause instabilities if ϵ is set too high as shown in Figure 5. Since we use a vorticity confinement style force to drive the grid based vorticity towards the particle based vorticity, similar issues arise in our method. However, as in standard vorticity confinement, a large range of parameter values seem to perform quite nicely. Although one could limit our vorticity confinement forces as the grid based vorticity approaches the particle based vorticity (as mentioned above), we have not found this necessary.

6 Examples

We implemented our method on both uniform and octree grids and generated a variety of examples demonstrating its versatility. The extra computational cost incurred by using vortex particles was negligible (less than 5%). Most of our examples used a clamped Gaussian kernel

$$\xi_p(\mathbf{x} - \mathbf{x}_p) = e^{-\|\mathbf{x} - \mathbf{x}_p\|^2/2r^2} / (r^3(2\pi)^{3/2})$$

when $\|\mathbf{x} - \mathbf{x}_p\| \leq r$, and 0 otherwise. In Figure 2, we seeded about 6000 particles during an initial divergence driven expansion lasting

Figure 4: Our method has been used in a production environment to create large rolling explosions. ($100 \times 100 \times 100$ uniform grid, approximately 400 vortex particles). Images © Lucasfilm Ltd. & TM. All rights reserved.

.5 seconds. Particles are seeded with random position while directions are placed tangent to the cylinder centered at the source region's midpoint oriented upward. We use a radius extending about 4 grid cells (for octrees we compute the radius using the smallest cells) and a particle vorticity of 2×10^{-3} . Figure 1 demonstrates that our technique also works well for liquids. Particles are seeded randomly at the inflow with vorticity pointing up or down to create toroidal eddies characteristic of rivers. To create larger vortices the kernel radius is increased to cover 40 grid cells and the particle vorticity magnitude is 1×10^{-2} for the top figure and 5×10^{-2} for the bottom figure. Figure 3 depicts a stream illustrating that we can handle complex geometries. The parameters are similar, except that particles that enter geometry are deleted. Also, we used a 4 grid cell particle radius in order to model a larger scale scene. The two images in Figure 4 show explosions generated for a recent feature film. These examples used only particle vorticity magnitude to affect ϵ in standard vorticity confinement. About 200 particles were used with a radius of about 3 grid cells in a $100 \times 100 \times 100$ simulation, and we used a tent kernel.

7 Conclusion

In summary, our method could be viewed as a traditional grid based Navier-Stokes solver with special forces added to obtain interesting fluid flows. These forces are obtained via a particle based approach to the vorticity formulation of the Navier-Stokes equations. Specifically, the requirements of our method are to (1) use vorticity carrying particles to preserve vorticity concentrations, and (2) target the grid based vorticity towards the particle based vorticity using a *vorticity conserving* body force, based on the successful vorticity confinement approach.

8 Acknowledgements

Research supported in part by an ONR YIP award and a PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, a Sloan Research Fellowship, ONR N00014-03-1-0071, ONR

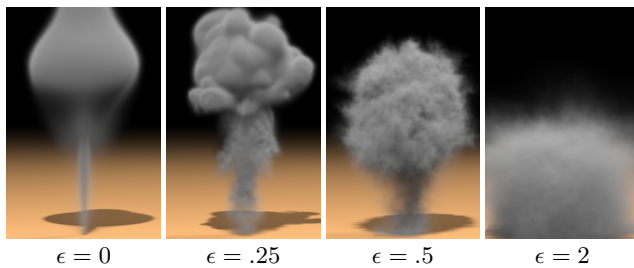


Figure 5: Simulations with varying vorticity confinement illustrate that too much confinement causes artifacts and instabilities. In fact, a large value of ϵ actually prevents the smoke from properly rising.

N00014-02-1-0720, NSF ITR-0121288, NSF DMS-0106694, NSF ACI-0323866 and NSF IIS-0326388. Computing resources were provided in part by Mike Houston, Christos Kozyrakis, Mark Horowitz, Bill Dally and Vijay Pande. We would also like to thank Cliff Plumer, Steve Sullivan, Willi Geiger and Industrial Light + Magic for their support and enthusiasm.

References

- CHEN, J., AND LOBO, N. 1994. Toward interactive-rate simulation of fluids with moving obstacles using the navier-stokes equations. *Computer Graphics and Image Processing* 57, 107–116.
- COTTET, G.-H., AND PONCET, P. 2003. Advances in direct numerical simulations of 3d wall-bounded flows by vortex-in-cell methods. *J. Comput. Phys.* 193, 136–158.
- DESBRUN, M., AND CANI, M.-P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Comput. Anim. and Sim. '96 (Proc. of EG Workshop on Anim. and Sim.)*, Springer-Verlag, R. Boulic and G. Hegron, Eds., 61–76. Published under the name Marie-Paule Gascuel.
- DRELA, M., AND MURMAN, E. M. 1987. Prospects for eulerian CFD analysis of helicopter vortex flows. In *American Helicopter Society Specialist Meeting, Arlington Texas*.
- ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21, 3, 736–744.
- FATTAL, R., AND LISCHINSKI, D. 2004. Target-driven smoke animation. *ACM Trans. Graph. (SIGGRAPH Proc.)* 23, 441–448.
- FEDKIW, R., STAM, J., AND JENSEN, H. 2001. Visual simulation of smoke. In *Proc. of ACM SIGGRAPH 2001*, 15–22.
- FELDMAN, B. E., O'BRIEN, J. F., AND ARIKAN, O. 2003. Animating suspended particle explosions. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22, 3, 708–715.
- FELICI, H. M., AND DRELA, M. 1990. Eulerian/lagrangian solution of 3-d rotational flows. In *AIAA 21st Fluid Dynamics, Plasma Dynamics and Lasers Conference*.
- FELICI, H. M., AND DRELA, M. 1993. An eulerian/lagrangian coupling procedure for three-dimensional vortical flows. *AIAA Journal*, 1993-3370.
- FELICI, H. M., AND DRELA, M. 1993. Reduction of numerical diffusion in three-dimensional vortical flows using a coupled eulerian/lagrangian solution procedure. In *AIAA 24th Fluid Dynamics Conference*.
- FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proc. of ACM SIGGRAPH 2001*, 23–30.
- FOSTER, N., AND METAXAS, D. 1997. Modeling the motion of a hot, turbulent gas. In *Proc. of SIGGRAPH 97*, 181–188.
- GAMITO, M. N., LOPES, P. F., AND GOMES, M. R. 1995. Two dimensional Simulation of Gaseous Phenomena Using Vortex Particles. In *Proc. of the 6th Eurographics Workshop on Comput. Anim. and Sim.*, Springer-Verlag, 3–15.
- GOKTEKIN, T. G., BARGTEIL, A. W., AND O'BRIEN, J. F. 2004. A method for animating viscoelastic fluids. *ACM Trans. Graph. (SIGGRAPH Proc.)* 23.
- HADAP, S., AND MAGNENAT-THALMANN, N. 2001. Modeling Dynamic Hair as a Continuum. *Comput. Graph. Forum* 20, 3.
- IHM, I., KANG, B., AND CHA, D. 2004. Animation of reactive gaseous fluids through chemical kinetics. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 203–212.
- KAJIYA, J. T., AND VON HERZEN, B. P. 1984. Ray Tracing Volume Densities. In *Proc. of SIGGRAPH 1984*, 165–174.
- KASS, M., AND MILLER, G. 1990. Rapid, stable fluid dynamics for computer graphics. In *Computer Graphics (Proc. of SIGGRAPH 90)*, vol. 24, 49–57.
- LAMORLETTE, A., AND FOSTER, N. 2002. Structural modeling of natural flames. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21, 3, 729–735.
- LINDSAY, K., AND KRASNY, R. 2001. A particle method and adaptive treecode for vortex sheet motion in three-dimensional. *J. Comput. Phys.* 172, 879–907.
- LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.)* 23, 457–462.
- MCMANARA, A., TREUILLE, A., POPOVIĆ, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *ACM Trans. Graph. (SIGGRAPH Proc.)*.
- MIYAZAKI, R., DOBASHI, Y., AND NISHITA, T. 2002. Simulation of cumuliiform clouds based on computational fluid dynamics. *Proc. Eurographics 2002 Short Presentation*, 405–410.
- MULLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 154–159.
- NEFF, M., AND FIUME, E. 1999. A visual model for blast waves and fracture. In *Proc. of Graph. Interface 1999*, 193–202.
- NGUYEN, D., FEDKIW, R., AND JENSEN, H. 2002. Physically based modeling and animation of fire. In *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 29, 721–728.
- PIGHIN, F., COHEN, J. M., AND SHAH, M. 2004. Modeling and editing flows using advected radial basis functions. In *Proc. of 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*
- PLOUMHANS, P., WINCKELMANS, G. S., SALMON, J. K., LEONARD, A., AND WARRE, M. S. 2002. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the sphere at $re=300, 500$, and 1000 . *J. Comput. Phys.* 178, 427–463.
- PREMOZE, S., TASDIZEN, T., BIGLER, J., LEFOHN, A., AND WHITAKER, R. 2003. Particle-based simulation of fluids. In *Comp. Graph. Forum (Eurographics Proc.)*, vol. 22, 401–410.
- RASMUSSEN, N., NGUYEN, D., GEIGER, W., AND FEDKIW, R. 2003. Smoke simulation for large scale phenomena. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22, 703–707.
- STAM, J. 1999. Stable fluids. In *Proc. of SIGGRAPH 99*, 121–128.
- STAM, J. 2003. Flows on surfaces of arbitrary topology. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22, 724–731.
- STEINHOFF, J., AND UNDERHILL, D. 1994. Modification of the Euler Equations for “Vorticity Confinement”: Application to the Computation of Interacting Vortex Rings. *Phys. of Fluids* 6, 8, 2738–2744.
- TREUILLE, A., MCMANARA, A., POPOVIĆ, Z., AND STAM, J. 2003. Keyframe control of smoke simulations. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22, 3, 716–723.
- WALTHER, J. H., AND KOUMOUTSAKOS, P. 2001. Three-dimensional vortex methods for particle-laden flows with two-way coupling. *J. Comput. Phys.* 167, 39–71.
- YAEGER, L., AND UPSON, C. 1986. Combining physical and visual simulation - creation of the planet jupiter for the film 2010. In *Proc. of SIGGRAPH 1986*, 85–93.
- YNGVE, G., O'BRIEN, J., AND HODGINS, J. 2000. Animating explosions. In *Proc. SIGGRAPH 2000*, vol. 19, 29–36.