

Efficient Large-Scale Hybrid Fluid Simulation

Abhinav Golas¹ Rahul Narain² Jason Sewall³ Pavel Krajcevski¹ Ming Lin¹
¹University of North Carolina at Chapel Hill, ² University of California, Berkeley, ³ Intel Corporation

1 Introduction

State-of-the-art methods for fluid simulation, including velocity-based grid methods and smoothed particle hydrodynamics [Bridson and Müller-Fischer 2007], require a detail vs. domain size tradeoff. As a result, scenes with large spatial scales are restricted to coarse detail under the restriction of limited computational resources. The elliptic problems solved for incompressibility projection in these simulations are *bandwidth-bound*, since domains of interest are not cache resident on current generation hardware. As a result, limited optimization is possible, and interactive performance is not possible for medium-large scenes.

An alternate approach is to model fluid *vorticity*, *i.e.* the curl of the velocity field using vortex singularities, which can be compactly represented using particle systems [Cottet and Koumoutsakos 1998]. These methods are (1) free of numerical dissipation, which can be significant with grid-based methods, (2) offer computational savings for scenes with unbounded fluid, and (3) have better compute to I/O ratios as well as trivially parallel algorithms. However, robust modeling of free-surfaces or efficient modeling of deformable obstacles with two-way coupling using vorticity methods is challenging. We present an efficient vortex particle implementation capable of interactive smoke simulations, and a novel hybrid algorithm coupling these methods with existing grid methods for robust and efficient modeling of obstacles and free-surfaces.

2 Efficient Vortex Particle Simulation

A vortex particle simulation is defined by a particle system where each particle carries a vorticity ω , and induces a velocity computed using the Biot-Savart law, the velocity at any point in space being the sum of all induced velocities. However, the Biot-Savart kernel is singular, and therefore unstable, hence a smoothed version of this kernel is used in practice. We use the Rosenhead-Moore kernel, which uses polynomial smoothing controlled by a smoothing radius. More details about vortex particle methods can be obtained from [Cottet and Koumoutsakos 1998].

Interactive vortex particle simulation can be obtained using a careful combination of *data layout*, *optimal algorithms*, and *efficient utilization of computational resources*. The particle system underlying our simulation is laid out as a struct-of-arrays, making it suitable for streaming operations. There are 3 major computational kernels, convection, strength exchange, and obstacle avoidance. For convection, naive summation is faster for small-medium scenes, while for large scenes (> 40000 particles), a hierarchical method is used. Strength exchange models viscous and shear components of the Navier Stokes equations by pair-wise exchange of vorticity. This is accelerated by the underlying spatial hierarchy used for convection. To model obstacles, we use the *panel method*, where vortex elements are placed on the obstacle mesh, and their strengths are determined as a solution to a linear system enforcing zero normal flux through the mesh. For rigid obstacles, this linear system can be precomputed, resulting in computational savings.

3 Hybrid Fluid Simulation

2-way coupled free-surfaces or deformable objects are not possible in this formulation since pressure boundary conditions are non-trivial. To remedy this, we couple vortex methods with existing grid-based fluid simulations with domain decomposition, where such surfaces

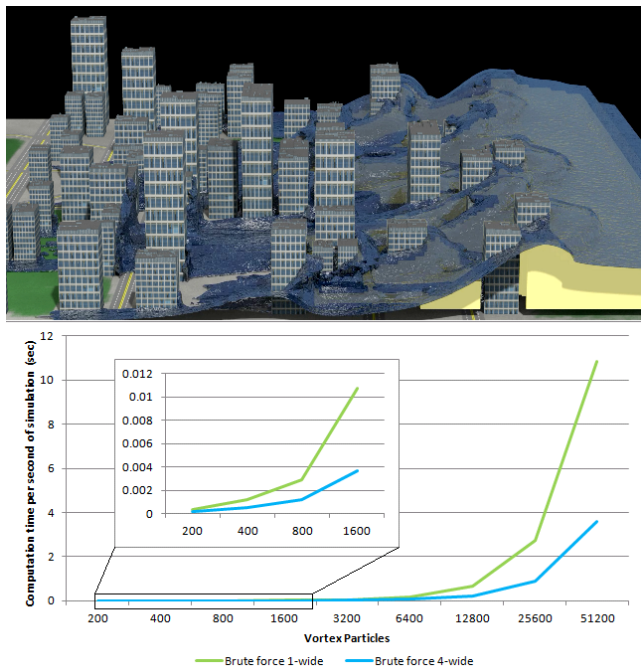


Figure 1: (top) Flooding a city, yellow surface denotes the vortex domain, remaining fluid simulated using grid methods (bottom) 4-core computation time, with inset. Note: > 2000 fps performance for < 1600 particles

are modeled using grids, while the rest of the scene uses vortex methods, as illustrated in 1(top). Consistency in velocity is maintained by a novel iterative coupling algorithm, where velocity from each domain is used as a boundary condition for other overlapping domains, as part of an incompressibility projection step for grids, or as a flux matching panel method for vortex methods. Vorticity is transferred across domains, in a grid using advection, and by creation of vortex particles in the vortex domain. Vortex particles are also allowed to go through grid domains where they can be used for vorticity confinement. This is an improvement over existing confinement approaches which use noise to generate vorticity, since fluid vorticity is used to maintain detail.

4 Results

Using our hybrid algorithm, free surfaces can be modeled with vortex methods, and deformable obstacles can be modeled with an $O(n^2)$ algorithm, as opposed to the naive $O(n^6)$ algorithm, n being the number of mesh faces. Our approach maintains fluid detail, and gives speedups ranging from 2X (water) to 100X and beyond (smoke) as compared to comparable grid simulations. Our vortex particle implementation can simulate smoke in game-like scenarios at real-time rates, using 500-1000 vortex particles instead of much larger grids.

References

- BRIDSON, R., AND MÜLLER-FISCHER, M. 2007. Fluid simulation: Siggraph 2007 course notes. In *ACM SIGGRAPH 2007 courses*, ACM, SIGGRAPH '07, 1–81.
- COTTET, G. H., AND KOUMOUTSAKOS, P. D. 1998. *Vortex Methods: Theory and Practice*. Cambridge University Press.