Avoiding Particle Dissipation for Adaptive Vortex Methods Through Circulation Conservation

Avoiding Particle Dissipation for Adaptive Vortex Methods Through Circulation Conservation

M. Harper Langston*
New York University, Courant Institute of Mathematical Sciences

1 Introduction

Adaptive fluid solvers have been built in a grid-based fashion [Losasso et al. 2004], for which desired areas of refinement must be known beforehand. This is not always possible, and further, these methods can be slow for turbulent flows where small time-stepping is required. Vortex methods can address this issue such as in [Selle et al. 2005]. Here, vorticity is transported by particles, causing the non-linear term in the Navier-Stokes equations to be handled more easily than in grid-based methods and allowing for better modeling of phenomena such as smoke.

Vortex methods pose additional difficulties, however, in that particles can become clustered, and resolution can become lost in areas of interest. [Selle et al. 2005] address this through vorticity confinement, but this may lead to unnatural or physically inconsistent flows. To address the problem of particle dissipation without introducing an increase in vorticity field strength, we introduce a method which maintains local *circulation* about a closed curve.

2 Our Technique

Free-Space Particle Simulation. In order to solve the Navier-stokes equations in vortex form, we first solve the vortex Euler equations in free space, discretely represented as:

$$\mathbf{u}_\omega(\mathbf{x}_j, t_k) = \sum_{i\neq j}^{N} K\left(\mathbf{x}_j^k - \mathbf{x}_i^k\right)\omega_j^k h^2, \qquad (1)$$

where K is the *Biot-Savart* kernel. Here, h represents the initial distance between each particle, and at the kth timestep, $\omega_j^k = \omega_j^0$ in 2D. Equation 1 is computed quickly using the Barnes-Hut adaptive quadtree algorithm with smoothing, and particle positions are updated through a basic advection step.

Particle Adaptivity through Circulation. To keep particles from dissipating or clustering, we use a non-adaptive quadtree of an appropriate resolution on top of our existing quadtree structure to locate empty or near-empty cells. New particles are introduced, and a new particle's vorticity is interpolated from near neighbors. Simply adding these particles would increase the vorticity field strength, causing undesired velocities. Our solution is to associate an h value with each particle and maintain local circulation, hence maintaining general field strength.

Circulation about a closed curve remains constant over time for isentropic flows absent external forces. We define local circulation for each particle as $\Gamma_i = \omega_i h_i^2$ such that for a new particle, $\mathbf{x}_{new}$, we maintain local circulation of a nearby particle cluster by adapting all nearby h values. We seek $\hat{h_i}$ values such that for M nearby particles $\sum_i^M \omega_i h_i^2 = \omega_{new} h_{new}^2 + \sum_i^M \omega_i \hat{h_i}^2$, and each particle's local velocity (the velocity derived from the influence of close neighbors) remains unchanged:

$$\mathbf{u}_{local}(\mathbf{x}_i) = K(\mathbf{x}_i - \mathbf{x}_{new})h_{new}^2 + \sum_{i\neq j}^{M} k(\mathbf{x}_i - \mathbf{x}_j)\omega_j \hat{h_i}^2 \qquad (2)$$

We require that at least 20 particles can be found nearby and overdetermine the linear system in equation 2 by a factor of 3-4 and solve using an SVD-based solver. If a particle's h value gets too large or there are too many particles in a single location, we delete one or more particles and distribute the influence to nearby particles using a similar solver. Figure 1 shows the general approach.

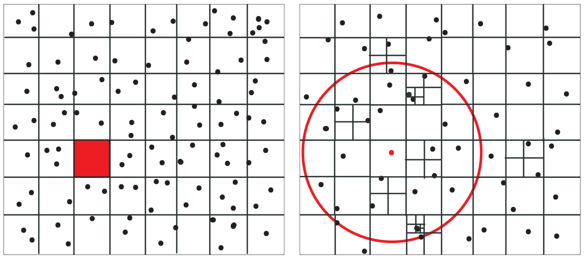

Figure 1: A non-adaptive quadtree locates empty cells for particle insertion while the adaptive tree builds an area of influence, from which particles are used for interpolation, their h values affected by the new particle (seen in red). Deletion works similarly.

Adding Viscosity. To add viscous diffusion for the Navier-Stokes equations, we use random walks based on independent and random Gaussianly distributed values to update particle positions.

Enforcing Boundary Conditions. At the boundary, $\mathbf{u}\cdot\mathbf{n} = 0$ and $\mathbf{u}\cdot\mathbf{t} = 0$ for normal and tangential vectors, respectively. For the former (no-flow), a *potential flow*, $\mathbf{u}_p$ is constructed using a boundary integral formulation as in [Chorin 1973] such that $\mathbf{u} = \mathbf{u}_\omega + \mathbf{u}_p$. The latter (no-slip) is enforced by introducing particles on each segment of the boundary of length l with vorticity $(\mathbf{u}_\omega + \mathbf{u}_p)\cdot\mathbf{t}l$ at appropriate timesteps.

3 Conclusion

We have introduced a method in 2D for *a priori* adaptivity in a particle-based fluid solver using conservation of circulation. Further, our approach preserves the computational advantages vortex methods provide over traditional grid-based solvers. Moving towards 3D, several issues are being addressed including vorticity stretching and particle generation on tangential boundary planes.

Figures 2 to 5 refer to the Perlman test case $\omega(r) = \left(\max\left(0, 1 - r^2\right)\right)^{10}$ on the domain $[-0.5, 0.5]^2$, for which the solution is known. All of the following simulations are also available in a supplemental video. Further images below show results for this method.

*e-mail: harper@cims.nyu.edu

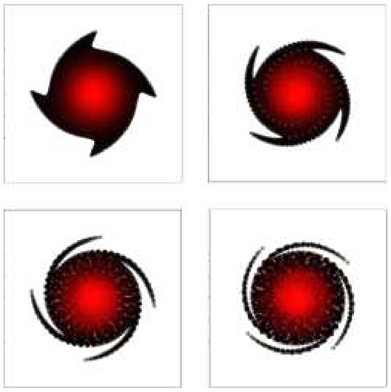

Figure 2: 16,000 particles are advected using the initial Perlman vorticity values through 100 time steps with no particle addition, blended together using their h values as their radii. Color intensity is based on vorticity strength. By the end, “holes” can be seen between particles.

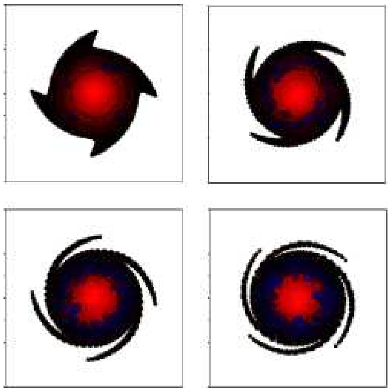

Figure 3: Again for the Perlman test case, we begin adding particles, colored blue to show contrast, maintaining local circulation among particles nearby. These particles fill the holes from figure 2 as desired.

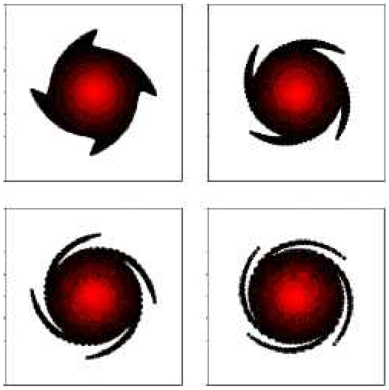

Figure 4: Instead of using blue particles to fill in holes, we interpolate the vorticity and colors of the new particles. By the end of this simulation, the total field strength and circulation has not increased significantly while the particles have increased from 16,000 to 17,500.

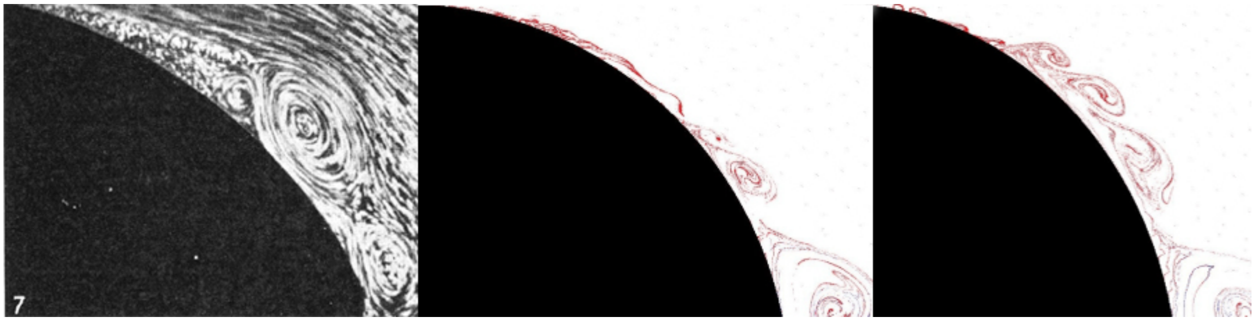

Figure 8: In order to satisfy the no-slip boundary condition, $\mathbf{u} \cdot \mathbf{t} = 0$ on ∂D, we generate new particles as described. The image on the left is from a real fluid simulation from G.K. Batchelor’s *An Introduction to Fluid Dynamics*, Cambridge 1967, used for visual comparison to our simulation. In both cases, we see the vortex “roll-up” effect.

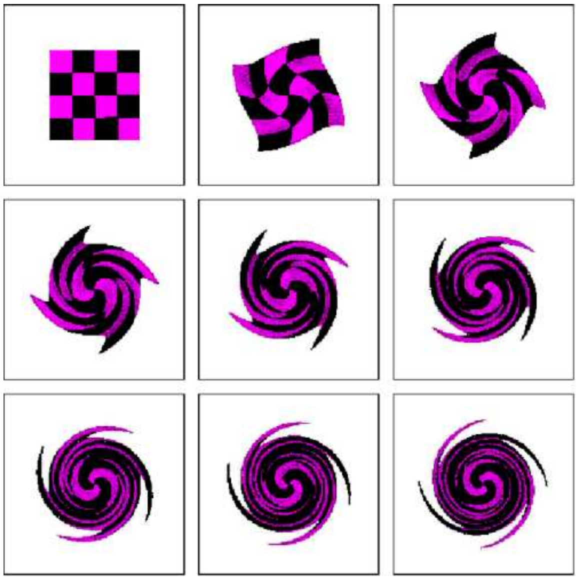

Figure 5: Again with the Perlman test case, we look at how a texture is deformed by the vorticity. Particles are added *a priori* as deemed appropriate by the system. Texture values are interpolated along with vorticity to show consistency of the interpolation scheme. This simulation begins with 64,000 particles and ends with 68,000.

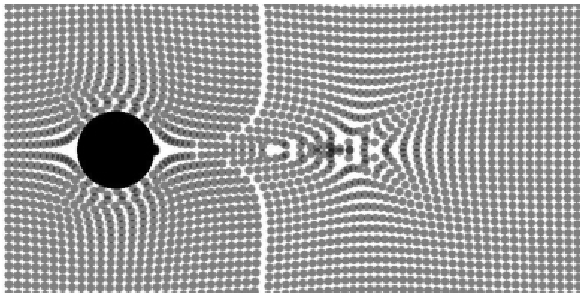

Figure 6: For particles with an induced velocity $U_\infty = 1.0$ and no vorticity, we use our boundary integral solver to show that we can induce a potential flow $\mathbf{u}_p$ to enforce no-flow boundary conditions $\mathbf{u} \cdot \mathbf{n} = 0$ on ∂D as described.

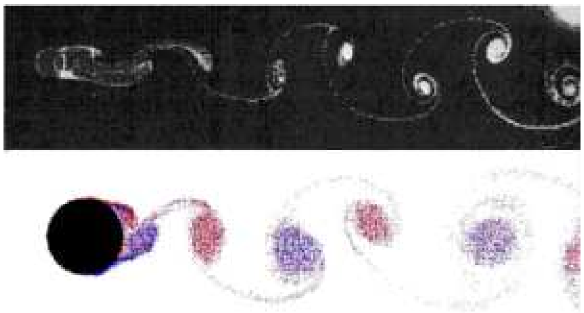

Figure 7: A comparison of adaptive vortex simulations, enforcing no-slip and no-flow boundary conditions to a fluid simulation from G.K. Batchelor's *An Introduction to Fluid Dynamics*, Cambridge 1967.

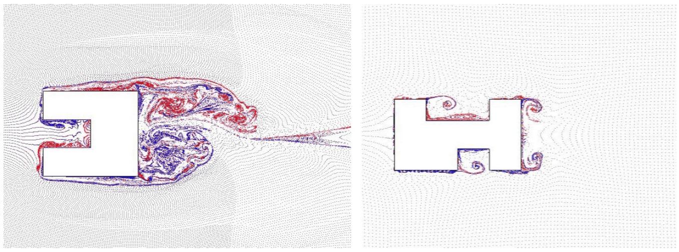

Figure 9: Simulations for odd-shaped geometries can be accomplished for properly discretized boundaries. We use gray ghost particles to look for areas of desired particle insertion and deletion even though these particles carry no vorticity. We note that as particles come in from the left on the sideways 'U' shape in the first image, ghost particles do not penetrate the boundary, and new boundary particles successfully push their way out. The simulation on the left is shown in greater detail in the supplemental video.

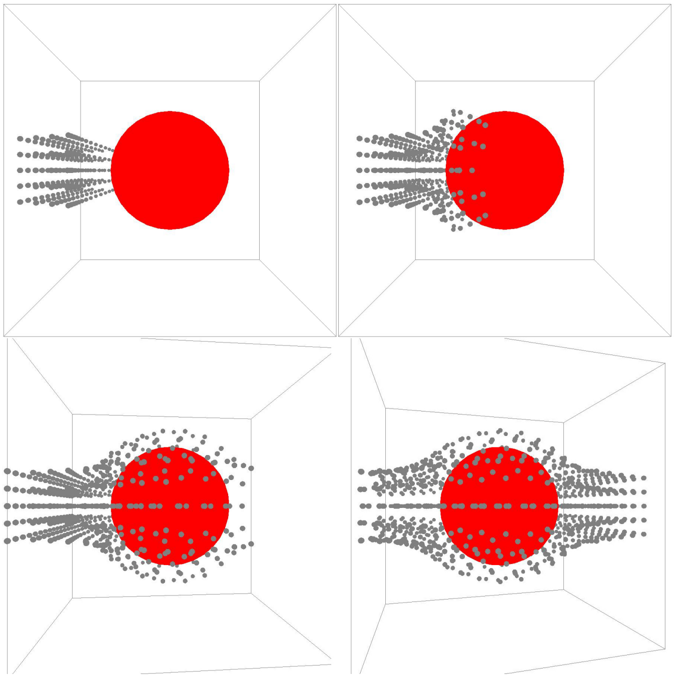

Figure 10: Initial results in three dimensions for simulating the potential flow. In three dimensions, we are working to accurately model vorticity stretching and no-slip boundary conditions.

References

CHORIN, A. 1973. Numerical study of slightly viscous flows. *Journal of Fluid Mechanics 57*, 785–796.

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

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