Fast Fluid Simulation Using Residual Distribution Schemes

Department of Computer Science University of North Carolina at Chapel Hill August 2007

The Challenge

Fluid phenomena play important roles in everyday life --- as blowing winds, jet streams, chemical dispersion, granular flows, ocean waves and currents, et cetera. Although these phenomena are commonplace, they are fascinating, visually and physically, for the effects they produce. Mathematical models that describe them properly are nonlinear and lead to computational simulation processes that are very complex and challenging to perform efficiently; the intricate interplay of essential processes such as convection, diffusion, turbulence, surface tension, and their interaction with rigid and deformable solids demands careful attention to stability, temporal and spatial scales, and domain representations.

This work explores the phenomena of high-energy fluid elastic solid interaction. The two simulations, elastic and fluid, are computed separately and coupled in time. This splitting greatly simplifies the formulation and allows for a more efficient process. Between steps, boundary conditions are propagated to provide bidirectional interaction.

Existing simulation methods for this type of interaction are better suited for lower energy scales; this work aims to capture shocks and other phenomena requiring compressible flows in a high-energy state. To achieve this, we employ a method known as *Residual Distribution Scheme* (RDS) for the fluid simulation [Roe 1987]. Our approach has the following benefits:

- It is easily adapted to take advantage of recent parallel hardware.
- The simulation offers a nice trade-off between performance and accuracy.
- RDS is capable of multi-physics - different physical laws can be defined on a per-cell basis.

The most commonly used *Residual Distribution Scheme* formulations assume linear solutions and therefore operate on a simplicial meshes; in our work we use an unstructured, adaptive tetrahedral mesh to represent the computational domain. To refine areas of the mesh, we subdivide the simplicies at the edge midpoints.

RDS is applicable to systems of the form:

$$q_t + f(q)_x + g(q)_y + h(q)_z = 0$$

also known as *conservation laws*. The Euler equations of compressible, inviscid fluid dynamics are naturally expressed in this way. RDS operate as iterations of a narrow stencil on the unstructured grid - each iteration, per-simplex residuals are computed and distributed to adjacent vertices. The basic procedure for the fluid solve is described with the following pseudo-code:

```
for each node n ∈ LeafFluidNodes
    do ClearAccumulator(n)
▷ implicit barrier
for each cell c ∈ LeafFluidCells
    do NodeUpdates ← EulerRDS(c)
        for each node n ∈ IndicentNodes(c)
            do AtomicInc(n, NodeUpdates[n])
▷ implicit barrier
for each node n ∈ LeafFluidNodes
    do TimeIntegrate(n)
```

Our current elastic simulation employs the well known finite element method (FEM). A Galerkin formulation of the equations of linear elasticity [Hughes 2000] is used with implicit time integration for stability. The resulting system is efficiently handled using an iterative solver; we use the method of conjugate gradients [Shewchuk 1994].

To mitigate the problem of inverted elements and slow convergence due to differences in the formulations of elasticity and fluid dynamics, we also propose a simple moving mesh system to "push" the fluid tetrahedra out of the way of the moving elastic boundary.

Results

We have tested and applied our method to a number of challenging problems with applications in computer animation: (a) foggy air current speeding past an iconic bridge, rocking it back and forth, (b) a flow of solar particles passing over a space station suspended high above the Earth, (c) wind buffeting a skyscraper, causing it to bend and twist. These scenarios are shown in Fig. 1. The numbers of tetrahedra listed in Table 1 are for the input meshes given to the solver.

The timing table shows the runtime performance achieved by our prototype implementation on the three benchmarks. The timings were collected on a Pentium D 3.4GHz processor with 2 GB of RAM. Our fluid simulation using RDS runs in real time. The dominating computational cost in our simulator is due to modeling and simulation of deformable soids using FEM.

To demonstrate the scalability of RDS, we have implemented our algorithm with the parallization facilities provided by OpenMP. This model of parallel computing is well-suited to the multi-core, shared-memory architectures commonly available on desktop workstations and laptops. It will also be directly applicable to many-core architectures. We achieve near-linear scaling for up to eight processors on the skyscraper model, as shown in Graph 1.

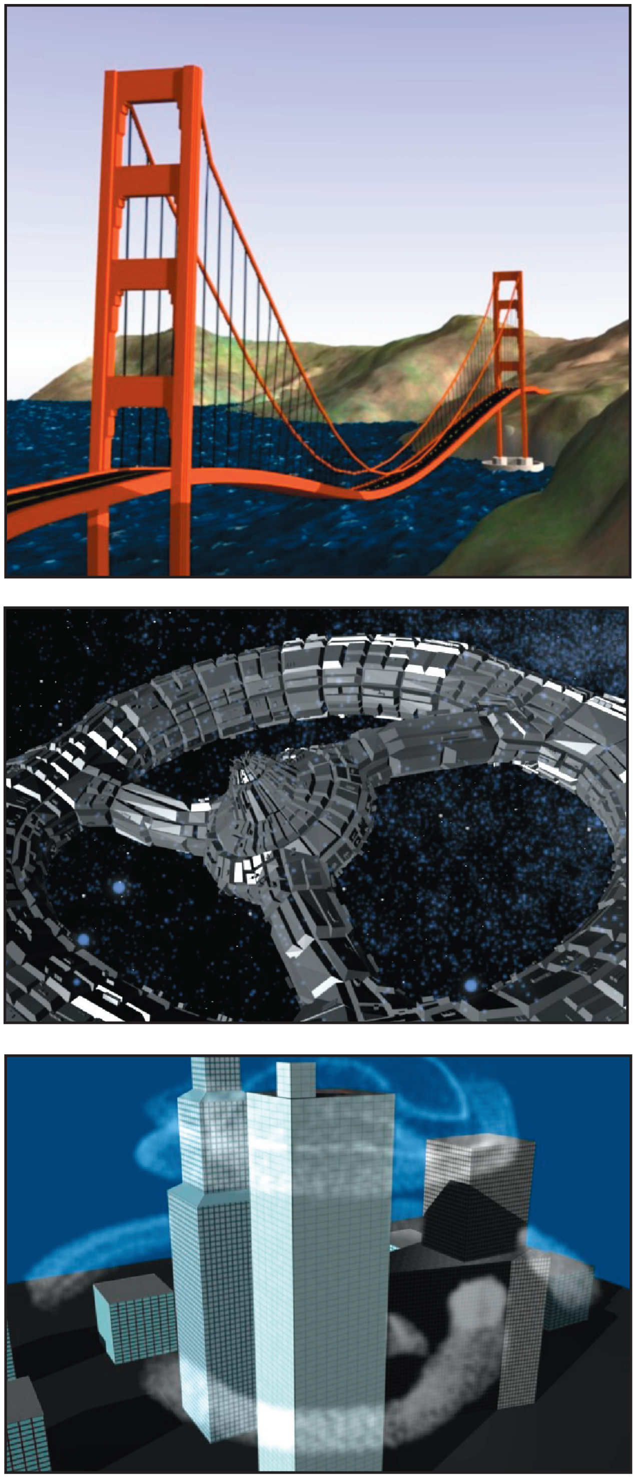
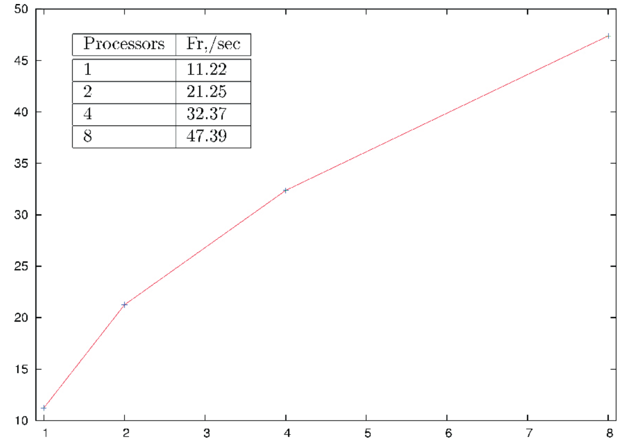


Figure 1: Benchmark Models

Scene	# cells	secs/frame		
		Fluid	Solid	Total
Bridge	31k	0.6	5.73	6.36
Skyscrapers	9k	0.15	4.77	4.92
Space station	25k	0.46	14.53	14.99

Table 1: Performance



Graph 1: Linear performance scaling of residual distribution schemes for the Euler equations over the Skyscraper scene on an SGI Altix cluster.

References

Roe, P. L. "Linear advection schemes on triangular meshes." Cranfield Institute of Technology, 1987.

Shewchuk, J. "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain." Carnegie Mellon University, 1994.

Hughes, T. J. R. *The Finite Element Method---Linear Static and Dynamic Finite Element Analysis*. Dover Publishers, New York, NY, 2000.

Project Leaders

Ming Lin, Beverly W. Long Distinguished Professor

Team Members

Jason Sewall, graduate research assistant
Paul Mecklenburg, graduate research assistant (now at Google)

Collaborators

Sorin Mitran, assistant professor (Math)

Research Sponsors

Intel Corporation
National Science Foundation
Army Research Office
RDECOM

Selected Publications

J. Sewall, P. Mecklenburg, S. Mitran, and M.C. Lin. "Fast Fluid Simulations Using Residual Distribution Schemes," Proc. of Eurographics Workshop on Natural Phenomena 2007.

http://gamma.cs.unc.edu/FFRDS