Sparse Smoke Simulations in Houdini

Omar Zarifi
SideFX Software Inc.
omarz@sidefx.com

ABSTRACT

High resolution fluid simulations are commonly used in the visual effects industry to convincingly animate smoke, steam, and explosions. Traditional volumetric fluid solvers operate on dense grids and often spend a lot of time working on empty regions with no visible smoke. We present an efficient sparse fluid solver that effectively skips the inactive space. At the core of this solver is our sparse pressure projection method based on unsmooth aggreggation multigrid that treats the internal boundaries as open, allowing the smoke to freely move into previously-inactive regions. We model small-scale motion in the air around the smoke with a noise field, justifying the absence of reliable velocities in the inactive areas.

CCS CONCEPTS

• **Computing methodologies → Physical simulation**.

KEYWORDS

fluid simulation, smoke simulation

ACM Reference Format:

Omar Zarifi. 2020. Sparse Smoke Simulations in Houdini. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '20 Talks), August 17, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388767.3407380

1 INTRODUCTION

From stylized cigarette smoke and realistic condensation droplets to massive explosions and ocean waves: fluid simulators are a ubiquitous tool in the visual effects indistry. Based on the type of fluid they aim to animate, fluid simulators in computer graphics can be classified into two broad categories. The primary concern for a *liquid* simulation is resolution of the surface. Due to the high relative density of typical liquids to air, the free surface approximation can be applied to treat the air as a passive vacuum. On the other hand, *smoke* simulations, aim to capture the evolution of varying quantities embedded in the air, such as soot density and temperature. The goal is thus to simulate motion of the fluid surrounding the visible area of interest.

In this talk, we focus on fast and efficient smoke simulations. Traditional simulators operate on dense grids, performing the solve everywhere inside of a cubic domain. This approach fullfills the need that fluid surrounding the smoke be simulated, but can spend a significant portion of time computing velocities in inconsequentially far areas. This drawback can be somewhat mitigated by using adaptive grids with coarser resolution outside the region of interest [Nielsen et al. 2018]. However, the solution domain still forms a bounding box, and grading restrictions can inhibit savings offered by this strategy.

Our approach is instead to completely ignore air that is far from the smoke, thereby restricting the solve to a band around the visible region. This is aided by our pressure projection scheme that enjoys a very good convergence rate and naturally accommodates sparse domains. By leveraging a simple algebraic multigrid cycle as a preconditioner, we can tackle the pressure equation without getting bogged down by topology of the domain. Our use of unsmooth aggreggation allows for trivial data transfer between levels and preserves sparsity structure of the coefficient matrix (unlike classical algebraic multigrid).

We note that our *sparse* simulator is not conceptually different from one that operates on the full bounding box, since both approaches truncate the domain to a finite region. However, the solution domain can be an arbitrarily small subset of its bounding box with our method. Furthermore, a common way to break up the smoke is by introducing a *disturbance* to velocities in the air around it (modelling small-scale chaotic motion). This addition of noise suggests that accurate far-field velocities are not of critical importance and justifies their absence from our method.

2 PRESSURE PROJECTION

Fluid simulators for computer graphics use operator splitting to solve the Navier-Stokes equations [Stam 1999]. In particular, each simulation step involves advecting the velocity field with the flow, applying external forces, followed by pressure projection. Since the first two steps can readily be performed in a sparse manner, we focus on solving the pressure equation:

$$-\nabla^2 p = d - \nabla \cdot \mathbf{u}^*. \quad (1)$$

Here, p is the unknown pressure field, d denotes the prescribed divergence field, and $\mathbf{u}^*$ are the intermediate velocities prior to projection. The common finite-difference discretization of this differential equation on a 3D grid yields a linear system $L\mathbf{p} = \mathbf{b}$. Our aim is to solve this system in a manner that is suitable for sparse simulations.

Geometric multigrid is a very attractive and widely-used approach to solving the pressure equation [Molemaker et al. 2008]. It works by discretizing the underlying problem at different resolutions and transferring data between these levels in a careful manner, allowing it to simultaneously home in on the high- and low-frequency components of the solution. However, sparse domains are increasingly difficult to capture at lower resolutions, rendering geometric multigrid unsuitable for sparse scenarios.

SIGGRAPH '20 Talks, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7971-7/20/08.
https://doi.org/10.1145/3388767.3407380

By interpreting source terms in the Poisson equation as electric charges, the pressure field can be determined by integrating the electrostatic field [Lait 2018]. This approach naturally fits in the sparse framework. Although this method enjoys excellent asymptotic running times, it is not practical at typical simulation resolutions.

2.1 Unsmooth Aggregation

Algebraic multigrid directly analyzes the linear operator to construct coarser approximations [Stüben 2001]. Writing the solution vector as $\mathbf{p} = P\mathbf{q} + \mathbf{e}$, we can deduce that

$$P^T \mathbf{b} = P^T (L\mathbf{p}) = P^T LP\mathbf{q} + P^T L\mathbf{e}. \tag{2}$$

The solution strategy is then to solve the smaller system $P^T LP\tilde{\mathbf{q}} = P^T \mathbf{b}$, then approximate the error by applying relaxation to $L\tilde{\mathbf{e}} = \mathbf{b} - LP\tilde{\mathbf{q}}$.

Classical algebraic multigrid aims to find a restriction operator P whose column space contains the eigenvectors of L that correspond to small eigenvalues (the near null-space). This condition endows P with an irregular structure, making restriction and prolongation steps somewhat nontrivial. More importantly, the variationally-coarsened operator, $P^T LP$, loses the sparsity structure of the original matrix, and each subsequent coarsening results in a yet denser matrix.

Unsmooth aggregation is a variant of algebraic multigrid that uses a very basic restriction rule: it simply aggregates fine elements into super-cells. As a demonstration, let the elements of $\mathbf{p}$ have labels in $\{1, \ldots, n\}$. This set is partitioned into disjoint subsets $S_1, \ldots, S_k$ (denoting the aggregates), yielding the following prolongation matrix:

$$P_{ij} = \begin{cases} 1 & \text{if } i \in S_j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The coarsened operator $P^T LP$ also ends up with a very simple form:

$$(P^T LP)_{ij} = \sum_{a \in S_i} \sum_{b \in S_j} L_{ab}. \tag{4}$$

In particular, we see that the ij entry of the coarsened matrix is non-zero only if there are elements $a \in S_i$ and $b \in S_j$ such that $L_{ab} \neq 0$. Thus, an appropriate choice of aggregates allows us to avoid densifying the coarsened matrix.

We use geometric intuition about the problem to select the aggregates. Since pressure is continuous, adjacent voxels of the discrete field should have similar values. We thus aggregate $2 \times 2 \times 2$ regions into super-cells (though note that not all 8 constituent voxels are necessarily in the sparse domain). As a consequence, with the 6-neighbour Laplacian discretization, all coarsened matrices also end up with no more than 7 non-zeros per row.

Unsmooth aggreggation by itself has been shown to perform well for simulation of deformable solids [Xian et al. 2019]. Our implementation uses unsmooth aggregation multigrid as a preconditioner for conjugate gradient to solve the pressure equation; we provide the details of our algorithm and its evaluation in the supplementary document.

3 DISTURBANCE

Volumetric fluid simulations often exhibit laminar flow, giving rise to smooth smoke caps; although mathematically correct, such results are unrealistic and undesirable. To add more interesting details to the smoke, we disturb the air surrounding it, thereby introducing small-scale chaotic motion. In particular, a zero-mean random force is applied to the air around the smoke. To keep the effect consistent at different resolutions, we integrate the force field in a reference region and control distribution of this random vector. In other words, a packet of air of the reference size is expected to experience similar forces at different simulation resolutions. Figure 1 shows the effect of enabling disturbance.

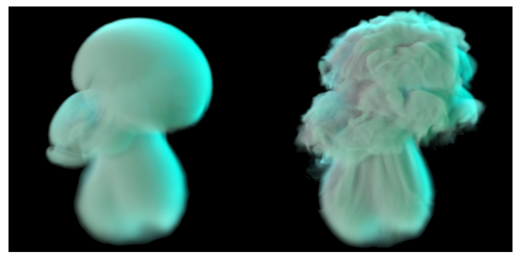

Figure 1: Smoke without and with disturbance.

4 FUTURE WORK

As noted in the supplementary document, our active chunks are quantized in multiples of 16 voxels. This results in significant wasted space for low resolution simulations. Ideally the system could transition to a dense box below a certain resolution to work across scales better. For very sparse simulations (such as scattered explosions) there is no need to keep them in a single grid. Automatically detecting disconnected components would allow creation of separate asynchronous simulations.

REFERENCES

Jeff Lait. 2018. Divergence projection with electrostatics. In *ACM SIGGRAPH 2018 Talks*. 1–2.

Jeroen Molemaker, Jonathan M Cohen, Sanjit Patel, Jonyong Noh, et al. 2008. Low Viscosity Flow Simulations for Animation.. In *Symposium on Computer Animation*, Vol. 2008.

Michael B Nielsen, Konstantinos Stamatelos, Morten Bojsen-Hansen, Duncan Brinsmead, Yannick Pomerleau, Marcus Nordenstam, and Robert Bridson. 2018. A collocated spatially adaptive approach to smoke simulation in bifrost. In *ACM SIGGRAPH 2018 Talks*. 1–2.

Jos Stam. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 121–128.

Klaus Stüben. 2001. A review of algebraic multigrid. In *Numerical Analysis: Historical Developments in the 20th Century*. Elsevier, 331–359.

Zangyueyang Xian, Xin Tong, and Tiantian Liu. 2019. A scalable galerkin multigrid method for real-time simulation of deformable objects. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–13.