Divergence Projection with Electrostatics

Jeff Lait
Side Effects Software Inc.

ABSTRACT

The pressure component of the Navier-Stokes equation can be solved by projecting out the divergent component of the velocity field. The Poisson equation used matches the electrostatic field equation, allowing a re-interpretation of the projection operation as a solution of electrostatic potential. Using a hierarchical dipole approximation, we achieve an efficient evaluation of the projection operator across sparse domains in a single pass. The update of each voxel is fully decoupled; allowing full parallelism and distribution.

CCS CONCEPTS

• **Applied computing** → **Physics**; • **Computing methodologies** → Massively parallel algorithms;

KEYWORDS

fluid simulation, parallel computing

ACM Reference Format:
Jeff Lait. 2018. Divergence Projection with Electrostatics. In *Proceedings of SIGGRAPH '18 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3214745.3214752

1 ELECTROSTATIC EQUIVALENCE

In timesteps relevant to computer graphics, pressure waves are often irrelevant. Given a disturbance we seek a steady state velocity field that equalizes pressures. We can find this by replacing the pressure component of the Navier-Stokes equation with a projection to the closest divergence-free velocity field [Stam 1999].

As shown in the supplementary material, if the divergence of a velocity field is replaced by electric charge; the resulting electrostatic field is precisely the correction field required to make the original velocity field divergence-free.

2 SOLVING FOR VELOCITY

2.1 Divergence as Point Charge

While our divergence field is smooth in theory, in practice we compute on a grid and evaluate divergence at cell centers. Ignoring grid-discretization effects, we replace these divergence values with point charges of the same signed magnitude. The electrostatic field of a point charge can be solved analytically. In three dimensions, it has the form $1/r^2$ with r the distance to the charge. With the proper scaling the velocity change for each voxel can be computed as the sum of displacements induced by a collection of point charges. This process is $O(n^2)$ in voxel count, however.

SIGGRAPH '18 Talks, August 12-16, 2018, Vancouver, BC, Canada
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5820-0/18/08.
https://doi.org/10.1145/3214745.3214752

2.2 Local Field

The approximation of a voxel of divergence with a single point charge is tolerable at a distance; but in the near field it is incredibly inaccurate. Consider the velocity update of a single voxel. Setting the voxel radius (face-to-face) at 1 and divergence to 1, a $1/r^2$ field adjusts the face centered velocities by ± 1. Since the unit-radius cube has surface area 24, the change in divergence will be 24. But the expected change in divergence from the field $1/r^2$ is 4π, almost a factor of 2 different.

Rather than derive the required discretized coefficients for the near field, we instead used an existing multigrid solver to compute the velocity change induced by a unit of divergence. Since the velocity change is proportional to the divergence, we can use a single look-up table to accurately compute the effect of divergence from nearby cells.

We used this hybrid approach to validate the point charge approximation. We replaced Houdini's pressure projection with our own, which used the lookup table for the 5^3 grid of nearest points and point charges for all other points. The $O(n^2)$ update required the simulation size to be limited, but we verified empirically that divergence was removed stably.

2.3 Dipole Approximation

To be used beyond toy examples, we had to reduce the computational complexity. The obvious approach is to group distant charges. By coalescing charges into a level-of-detail hierarchy, we can sample from the coarser hierarchy for distant charges, reducing our workload to $O(nlg(n))$

Our desired level of detail system must be easy to compute, evaluate, and stable. The simplest system is similar to mass clumping, and stores total charge on each reduced level. However, consider a cubic volume assigned a constant vertical force. The divergence of the velocity grid shows a surface of negative divergence on the top of the region and a positive divergence on the bottom. If we reduce this entire region to a single grid cell, the total divergence will sum to zero, but this shape clearly should have global effect.

The next simplest is to coalesce positive and negative charges independently. For each grid cell we store two wx, wy, wz, w tuples, one for the positive charges and one for the negative. The w is the magnitude of the charge and x, y, z the location. We store the weighted locations wx, wy, wz in the grid. Building the hierarchy is a simple sum reduction. With 8 floats per voxel, memory requirements of the first reduced grid matches that of single float full-resolution grid. Each additional level is 1/8 the storage so has minimal memory cost.

As shown in the supplementary material, it is important that the coarse levels of the dipole approximation are not used to evaluate nearby velocity grid corrections. Our two sampling patterns are 6^3 and 4^3 based patterns, where we can trade accuracy for speed.

3 BOUNDARY CONDITIONS

Boundary conditions are an important consideration in PCG (Preconditioned Conjugate Gradient), multigrid, and FFT (Fast Fourier Transform) based solvers. Our electrostatic equivalence is instead defined over an infinite unbounded domain.

3.1 Free Surface

To minimize memory use and computational time, fluid simulations operate on a small region of space. For multigrid and FFT solvers these are rectangular in shape, but PCG-based solvers can work with arbitrarily shaped regions. Free surface boundary conditions are used to avoid the look of smoke-in-a-box. This is not physically correct, however. It neglects the inertial effect of changing the border velocities - adjusting them implies also adjusting the external ghost velocities, but this cost is not present in the equations. Similarly, if two disjoint regions are simulated simultaneously, no pressure effects are transferred from one to the other. By contrast, the electrostatic approach induces the same velocity update regardless of the shape of the simulation domain.

While the free surface approximation suffices for visually plausible simulations, this true domain-invariance is one reason that the electrostatic approach can scale well to sparse simulations. The choice of sparsity structure has no effect on the projection.

While the absence of true free surfaces is an advantage for smoke simulations, for liquid simulations, where the inertia of the velocities on the other side are intentionally discounted, the electrostatic method is not useful.

3.2 Collisions

While closed boundary conditions are only easily expressed in PCG solvers, as shown with IOP (Iterated Orthogonal Projection) [Molemaker et al. 2008] they are not required for plausible simulations. The electrostatic method can use the IOP technique to add colliders.

Interestingly, due to domain-invariance, the colliders only need their surfaces immersed in a thin layer of active voxels. The effect of IOP is to introduce divergence at the collision boundary layer, which will then apply to distant voxels even if they are not topologically connected.

3.3 Ground Planes

Ground planes are a common case in smoke effects. IOP produces soft collisions, which are less ideal for a ground plane. Multigrid approaches can use a closed boundary condition on one wall of the domain to provide a solid ground plane. The electrostatic method provides an interesting alternative. By mirroring all electrostatic charges in the ground plane, we create a reflective surface at the cost of an extra charge computation.

4 RESULTS

We generate a white-noise velocity field in a dense 200x200x40 grid and apply the electrostatic solve repeatedly, timing the fifth pass. The 4^3 electrostatic method is competitive with PCG, but both are outclassed by multigrid, as expected for dense grids.

While PCG scales in theory, each iteration requires synchronization and a full memory pass. In practice, we have found PCG scales poorly, which we attribute to modern CPUs being often memory bound. The electrostatic method, by contrast, updates velocity in a single asynchronous pass and scales well with processing power.

Table 1: Fifth Pass on 1.6 MVoxel White Noise

Test	i7-4930K	2x Xeon Gold 5210
6^3 Electrostatic	12.170s	2.856s
4^3 Electrostatic	4.654s	1.083
PCG	1.239s	0.763s
Multigrid	0.075s	0.045s

The electrostatic method's advantage over multigrid is support for sparse grids. We source smoke diagonally in a 62 MVoxel grid, a worse case for dense multigrid approaches. Our sparse version defines a 6-voxel band around non-zero density as active and zeros the rest of the velocity volume. Timings are for the full simulation step. Occupancy is at 1% on the fifth frame.

Table 2: Fifth Frame of Sparse 62 MVoxel Simulation

Test	i7-4930K	2x Xeon Gold 5210
4^3 Electrostatic	4.8s	1.95s
Multigrid	11.06s	3.99s

5 FUTURE WORK

5.1 Distribution

Common approaches for distributing fluid simulations rely on using a PCG projection technique [Bailey et al. 2015], [Lait 2016]. Each PCG iteration must synchronize across the boundary layer. The electrostatic method only requires the border values of the dipole grid. In slice-based distributions, as in [Lait 2016], each slice can determine needed dipoles for its neighbours and provide them in a single transfer per pressure solve.

5.2 Local Block Solves

Splitting local and far computations can be extended to take advantage of fast multigrid and FFT grid techniques. Rather than using a lookup table to solve 6^3 or 4^3 neighbourhoods, dense regions could be isolated, as in [Chu et al. 2017], and solved with more efficient techniques.

REFERENCES

Dan Bailey, Harry Biddle, Nick Avramoussis, and Matthew Warner. 2015. Distributing Liquids Using OpenVDB. In *ACM SIGGRAPH 2015 Talks (SIGGRAPH '15)*. ACM, New York, NY, USA, Article 44, 1 pages. https://doi.org/10.1145/2775280.2792544

Jieyu Chu, Nafees Bin Zafar, and Xubo Yang. 2017. A Schur Complement Preconditioner for Scalable Parallel Fluid Simulation. *ACM Trans. Graph.* 36, 5, Article 163 (July 2017), 11 pages. https://doi.org/10.1145/3092818

Jeff Lait. 2016. Inside Houdini's Distributed Solver System. In *ACM SIGGRAPH 2016 Talks (SIGGRAPH '16)*. ACM, New York, NY, USA, Article 42, 2 pages. https://doi.org/10.1145/2897839.2927421

Jeroen Molemaker, Jonathan M. Cohen, Sanjit Patel, and Jonyong Noh. 2008. Low Viscosity Flow Simulations for Animation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 9–18. http://dl.acm.org/citation.cfm?id=1632592.1632595

Jos Stam. 1999. Stable Fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 121–128. https://doi.org/10.1145/311535.311548