Divergence-Free and Boundary-Respecting Velocity Interpolation Using Stream Functions

Jumyung Chang
University of Waterloo
j48chang@uwaterloo.ca

Vinicius C. Azevedo
ETH Zurich
vinicius.azevedo@inf.ethz.ch

Christopher Batty
University of Waterloo
christopher.batty@uwaterloo.ca

ABSTRACT

In grid-based fluid simulation, discrete incompressibility of each cell is enforced by the pressure projection. However, *pointwise* velocities constructed by interpolating the discrete velocity samples from the staggered grid are not truly divergence-free, resulting in unphysical local volume changes that manifests as particle spreading and clustering. We present a new velocity interpolation method that produces analytically divergence-free velocity fields in 2D using a stream function. The resulting fields are guaranteed to be divergence-free by a simple calculus identity: the curl of any vector field yields a divergence-free vector field. Furthermore, our method works on cut cell grids to produce fields that strictly obey solid boundary conditions. Therefore, no artificial gaps are created between fluid particles and solids, and fluid particles do not trespass into solid regions.

ACM Reference Format:
Jumyung Chang, Vinicius C. Azevedo, and Christopher Batty. 2019. Divergence-Free and Boundary-Respecting Velocity Interpolation Using Stream Functions. In *SCA '19:The ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '19), July 26–28, 2019, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3309486.3339890

1 INTRODUCTION

Grid-based and hybrid Eulerian fluid schemes have gained popularity in part due to the ease with which they enforce global incompressibility. Although discrete incompressibility of each grid cell is satisfied by pressure projection, the *pointwise* velocities inside each cell, induced by standard interpolants, turn out not to be divergence-free. Consequently we cannot advect particles or fluid quantities with continuously divergence-free velocities. As an alternative advection scheme, we propose the use of a stream function approach.

If we express the velocity field $\mathbf{u}$ as the curl of the vector potential $\mathbf{\Psi}$, then $\mathbf{\Psi}$ is called a stream function.

$$\mathbf{u} = \nabla \times \mathbf{\Psi} \tag{1}$$

The benefit of this expression is that basic vector calculus ensures the velocity $\mathbf{u}$ is divergence-free by construction:

$$\nabla \cdot \mathbf{u} = \nabla \cdot \nabla \times \mathbf{\Psi} = 0 \tag{2}$$

SCA '19, July 26–28, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6677-9/19/07.
https://doi.org/10.1145/3309486.3339890

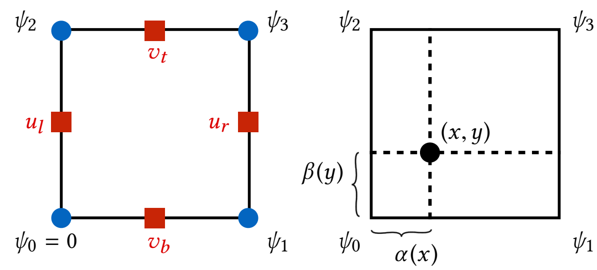


Figure 1: Constructing an interpolated stream function field. Left: Discrete stream function values (blue circles) are constructed from discrete velocity values (red squares). Right: Once discrete nodal stream function values are known, use an interpolation scheme to obtain a pointwise stream function value

The main advantages of our stream function-based interpolation method are:

- the interpolated pointwise velocity field is continuously (analytically) divergence-free;
- the velocity field strictly obeys fluid-solid boundary conditions;
- the distribution of advected particles tends to be well-preserved;
- the interpolant integrate straightforwardly into standard grid-based 2D fluid solvers.

2 OUR APPROACH ON UNIFORM GRIDS

To simulate our fluid, we solve the incompressible Euler equations [Batchelor 1967]

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho} \tag{3}$$

with the usual operator splitting approach [Bridson 2015; Stam 1999]. Our approach only affects the advection steps. We adopted the usual staggered grid structure [Harlow and Welch 1965] where the velocity component samples lie on cell edges (in 2D). After the pressure projection step, we have a discretely divergence-free velocity field, but instead of interpolating the discrete grid velocities directly we apply several steps:

- convert the discrete velocity values to discrete stream function values (Section 2.1),
- determine the (continuous) stream function at a point using an interpolation method (Section 2.2),
- convert the stream function at the point to the final (continuously divergence-free) velocity (Section 2.3).

2.1 Constructing The Discrete Stream Function

The stream function vector in 2D has only one non-zero component: $\mathbf{\Psi} = (0, 0, \psi)$. Therefore (1) can be simplified to

$$\mathbf{u} = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}\right) \tag{4}$$

Using the notation in Figure 1, we further discretize (4) for the four faces of each grid cell as

$$u_l = \frac{\psi_2 - \psi_0}{\Delta y},\ u_r = \frac{\psi_3 - \psi_1}{\Delta y},\ v_b = -\frac{\psi_1 - \psi_0}{\Delta x},\ v_t = -\frac{\psi_3 - \psi_2}{\Delta x}, \tag{5}$$

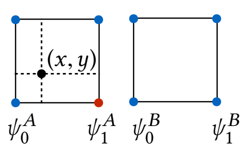
where Δx and Δy represent the width and height of the cell, respectively. The small system of four equations in (5) contains a one-dimensional null space, so it has an infinite number of solutions which differ by a constant offset. We assign an arbitrary value to one variable (i.e., $\psi_0 = 0$) and find a solution to the reduced system of three equations. This can be carried out separately per cell.

2.2 Interpolating The Stream Function

To compute a pointwise stream function value we interpolate the nodal discrete stream function values obtained in Section 2.1. For bilinear interpolation we can use the four stream function values (ψ_0, ψ_1, ψ_2, and ψ_3) defined in each cell. The interpolated value at the point (x, y) becomes

$$\begin{aligned}\psi(x, y) = &(1-\alpha(x))(1-\beta(y))\psi_0 + \alpha(x)(1-\beta(y))\psi_1 \\ &+ (1-\alpha(x))\beta(y)\psi_2 + \alpha(x)\beta(y)\psi_3,\end{aligned} \tag{6}$$

where $\alpha(x)$ and $\beta(y)$ are the length fractions shown in Figure 1.



For bicubic interpolation we use the four nodal stream function values *and* the (approximate) partial derivative of ψ. The partial derivatives are estimated by averaging the slopes of ψ in the two adjacent cells in the derivative direction, where the slopes are found by simple finite differencing. For example, $\frac{\partial \psi}{\partial x}$ at the red vertex is

$$\frac{\partial \psi}{\partial x} \approx \frac{1}{2}\left(\frac{\psi_1^A - \psi_0^A}{\Delta x} + \frac{\psi_1^B - \psi_0^B}{\Delta x}\right).$$

Conveniently, this does not require a globally consistent ψ field since the finite differences do not cross between cells.

2.3 Recovering Divergence-Free Velocities

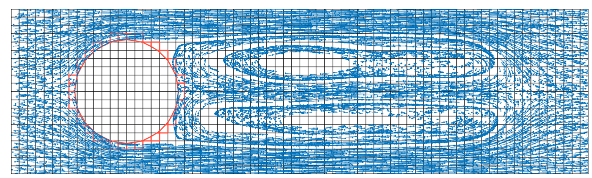
To recover a pointwise incompressible velocity field we apply the curl operator to the analytical (interpolated) stream function field. In the bilinear case we can simply apply (4) to (6), and likewise for the bicubic case.

3 SOLID BOUNDARIES

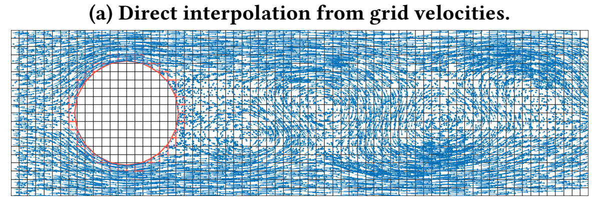
If a fluid cell contains a (static) solid object our new advection method can be applied analogously. We adopted a level set representation for solids and a cut-cell scheme for pressure projection [Batty et al. 2007; Ng et al. 2009]. The fluid part of the cell has a (convex) polygonal shape with net zero flux across its faces. Using Stokes theorem we again convert the discrete velocities on the edges to discrete stream function values on the vertices (similar to (5)). As the stream function samples are located at the polygon vertices, we use generalized barycentric coordinates [Meyer et al. 2002; Warren et al. 2007] for interpolation (similar to (6)). Finally, we analytically calculate the curl of the interpolated stream function value to obtain a pointwise velocity.

4 RESULTS

We compare our stream function method against the corresponding component-wise velocity interpolation scheme under a free-slip condition. We use bicubic interpolation (whether for velocity or stream function) on uniform cells and generalized barycentric interpolation on cut cells. We set the underlying grid velocities to be identical for both using a semi-Lagrangian method [Stam 1999], and observe how the blue secondary particles move differently depending on the interpolation method.



(a) Direct interpolation from grid velocities.



(b) Our stream function interpolation approach better respects solids and preserves particle distributions.

Figure 2: Particle trajectories in a horizontal flow.

Notice that our approach does not generate artificial gaps at the back side of the solid object, and the particle distribution remains more uniform with less (erroneously) empty space.

REFERENCES

George Keith Batchelor. 1967. *An Introduction to Fluid Dynamics*. Cambridge University Press.

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.* 26, 3 (2007), 100.

Robert Bridson. 2015. *Fluid simulation for computer graphics, 2nd edition*. CRC Press.

Francis H. Harlow and Eddie J. Welch. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluid* 8, 12 (1965), 2182–2189.

Mark Meyer, Alan Barr, Haeyoung Lee, and Mathieu Desbrun. 2002. Generalized Barycentric Coordinates on Irregular Polygons. *J. Graph. Tools* 7, 1 (Nov. 2002), 13–22.

Yen Ting Ng, Chohong Min, and Frédéric Gibou. 2009. An Efficient Fluid-solid Coupling Algorithm for Single-phase Flows. *J. Comput. Phys.* 228, 23 (Dec. 2009), 8807–8829.

Jos Stam. 1999. Stable Fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. 121–128.

Joe Warren, Scott Schaefer, Anil N. Hirani, and Mathieu Desbrun. 2007. Barycentric coordinates for convex sets. *Advances in Computational Mathematics* 27, 3 (01 Oct 2007), 319–338.