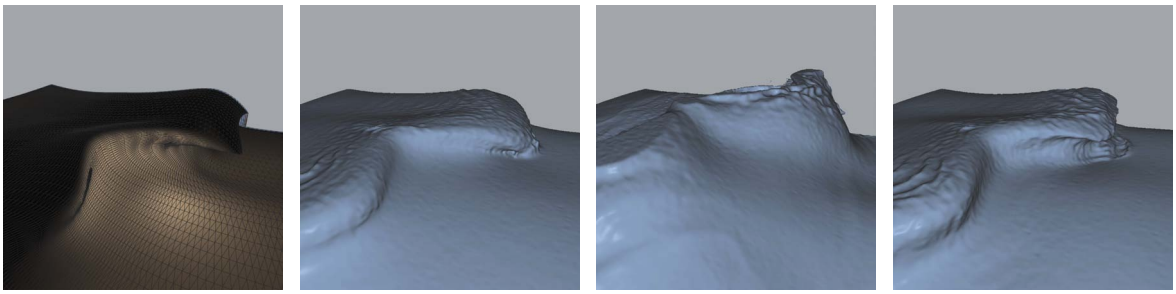
# Steady State Stokes Flow Interpolation for Fluid Control

Haimasree Bhatacharya[†1,2] Michael B. Nielsen[‡1,4] and Robert Bridson[§1,3]

[1]Weta Digital, New Zealand
[2]University of Utah, USA
[3]University of British Columbia, Canada
[4]Aarhus University, Denmark



(a) Input mesh (b) Potential flow (c) Velocity extrapolation (d) Steady state Stokes flow

**Figure 1:** *The surface velocities of an input mesh are interpolated throughout its interior and used as a feedback force in a free surface liquid simulation of a breaking wave. Interpolation based on potential flow and velocity extrapolation – explored by previous work – fail to capture the distinct curl of the wave. Steady state Stokes flow interpolation preserves the curl.*

**Abstract**

*Fluid control methods often require surface velocities interpolated throughout the interior of a shape to use the velocity as a feedback force or as a boundary condition. Prior methods for interpolation in computer graphics — velocity extrapolation in the normal direction and potential flow — suffer from a common problem. They fail to capture the rotational components of the velocity field, although extrapolation in the normal direction does consider the tangential component. We address this problem by casting the interpolation as a steady state Stokes flow. This type of flow captures the rotational components and is suitable for controlling liquid animations where tangential motion is pronounced, such as in a breaking wave.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

## 1. Introduction

To bring artistic control of fluids closer to traditional disciplines such as modelling and painting with which artists are familiar, new fluid control algorithms that support such established interaction metaphors must be investigated [TMPS03, MTPS04]. In this paper we explore steady state Stokes flow for interpolating the surface velocities of a possibly time-dependent shape throughout its interior. The surface velocities can be painted directly onto the shape or be determined implicitly from the shape's movement. Thereby the artist's toolset for controlling complex effects in a production environment is enhanced. In particular the interpo-

† e-mail:hb123@cs.utah.edu
‡ e-mail: nielsenmb@gmail.com
§ e-mail:rbridson@cs.ubc.ca



DOI: 10.2312/conf/EG2012/short/057-060

lated volumetric velocity field can be used to guide free surface liquid simulations or create complex particle flows that inherit the motion and shape specified by the artist by means of modelling, animating and painting. Traditionally Stokes flow has been studied in continuum mechanics [Lau05] where it is also known as creeping flow. For steady, *i.e.* time-independent, flows, Stokes flow becomes a boundary value problem that couples velocity and pressure, and the solution is determined entirely by the values of velocity at the boundary.Recent work [SY05a, NB11] has explored potential flow and velocity extrapolation in the normal direction for fluid control. For velocity extrapolation, velocities are propagated from the surface in the normal direction into the interior of the input shape *e.g.* by a discrete closest point transform where each unknown velocity value on the grid is set equal to the closest known sample. This is fast to compute, but the produced velocity field is discontinuous and usually not divergence-free. Potential flow models incompressible inviscid irrotational flow, solving for a scalar potential field whose gradient is the interpolated velocity field. This produces a divergence-free velocity field, provided the surface velocities integrate to a flux of zero. Both methods suffer from the problem that the rotational motion present in the velocity field is lost. Steady state Stokes flow interpolation on the other hand produces a divergence-free velocity field in addition to capturing rotations, perfectly reproducing rigid motion for example. In this paper we demonstrate several benefits of steady state Stokes flow interpolation over conventional velocity extrapolation and potential flow interpolation approaches.

## 2. Related Work

Many authors have worked in the general area of fluid control, matching input animations and low resolution simulations to fluid simulations of smoke and liquid [TMPS03, REN*04, FL04, HK04, MTPS04, SY05a, SY05b, TKPR06, NCZ*09, NC10, HMK11, NB11]. Additionally, Mihalef and colleagues [MMS04] control and simulate breaking waves by constructing an initial condition for the wave geometry from 2D height fields which is used as input to a free surface liquid simulation that simulates the plunge of the wave.

## 3. Method

Consider the Navier Stokes equations [Lau05]

$$\frac{D\mathbf{u}}{dt} = \nu\nabla^2\mathbf{u} - \nabla p/\rho + f \quad (1)$$

$$\nabla\cdot\mathbf{u} = 0 \quad (2)$$

where $p$ is pressure, $f$ is external body force-density, $\nu$ is kinematic viscosity, $\rho$ is density and $\mathbf{u}$ is velocity. Assuming inertial and time effects are negligible and letting $\nu = 1$, $\rho = 1$ and $f = 0$, the momentum equation (Eq.(1)) simplifies to

$$\nabla^2\mathbf{u} - \nabla p = 0 \quad (3)$$

Equations (3) and (2) combined with the boundary condition $\mathbf{u} = \mathbf{u}_{\text{input}}$ on $\partial\Omega$, the boundary of the shape into which we are interpolating, represent the steady state Stokes flow equations. These equations are well-posed provided that the compatibility condition $\int_{\partial\Omega}\mathbf{u}_{\text{input}}\cdot\mathbf{n} = 0$ is satisfied, where $\mathbf{n}$ is the surface normal. If this is not the case, *i.e.* a non-zero total flux across the surface is present, we modify the right-hand side of Eq.(3) to be $\int_{\partial\Omega}\mathbf{u}_{\text{input}}\cdot\mathbf{n}/|\Omega|$, where $|\Omega|$ is the volume of the input shape. This adjustment leaves the flux unchanged but introduces a uniform nonzero divergence in each grid cell. By discretizing on a MAC grid [Lau05], we get the following system of equations:

$$\begin{pmatrix} A_x & 0 & 0 & G_x \\ 0 & A_y & 0 & G_y \\ 0 & 0 & A_z & G_z \\ G_x^T & G_y^T & G_z^T & 0 \end{pmatrix}\begin{pmatrix} u \\ v \\ w \\ p \end{pmatrix} = \mathbf{b} \quad (4)$$

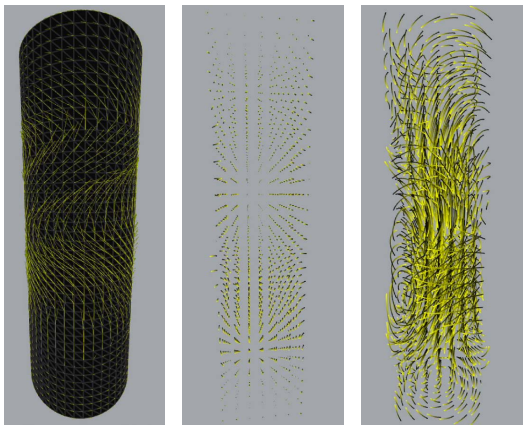
where $A$ is the second order accurate central difference approximation to the Laplacian centered on faces normal to the $x$, $y$ and $z$ directions respectively,$G_{\{x,y,z\}}$ are the central difference approximations of the $x$, $y$ and $z$ components of the gradient operator which acts on cell-centered values of pressure and produces face-velocities, and $G^T_{\{x,y,z\}}$ are the central difference approximations of the $x$, $y$ and $z$ contributions to the divergence operator which acts on face-velocities and produces a cell-centered divergence. The right-hand side $\mathbf{b}$ is constructed by substituting in known values for the boundary-velocities as well as adjustments for a non-zero flux. No boundary conditions on pressure are required. To avoid a singular matrix (due to the lower-right block of zeros), a small perturbation ($10^{-8}$ in our implementation) is added to the diagonal elements of the lower-right block of the matrix.

$$\begin{pmatrix} -A_x^{-1} & 0 & 0 & 0 \\ 0 & -A_y^{-1} & 0 & 0 \\ 0 & 0 & -A_z^{-1} & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \quad (5)$$

Evaluating this preconditioner amounts to solving three independent Poisson problems with pure Dirichlet boundary conditions for face-velocities in the $x$, $y$ and $z$ directions respectively. Elman et al. provide a good discussion of the optimality of the $LDL^T$ approach for further reference [ESW05].

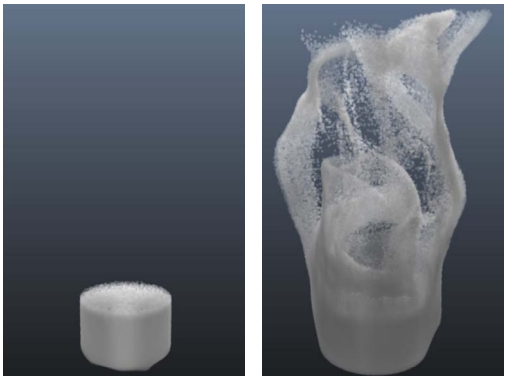
## 4. Results and Discussion

We have evaluated our steady state Stokes flow implementation on two examples. Figure 2.a shows a velocity field painted onto the surface of a cylinder. The velocity field is interpolated throughout the interior by potential flow and steady state Stokes flow in Figures 2.b and 2.c respectively. Since the surface velocity field consists mainly of a tangential component, the potential flow solution does not capture the flow and advecting particles in the interpolated velocity (Figure 3.a) reflects this. Interpolation based on steady state



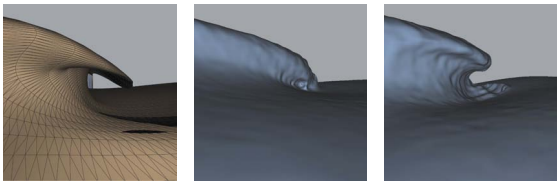
(a) Input (b) Potential flow (c) Stokes flow

**Figure 2:** *Velocities dominated by tangential components painted onto the surface of a cylinder and interpolated throughout the interior. Potential flow fails to capture the tangential components.*



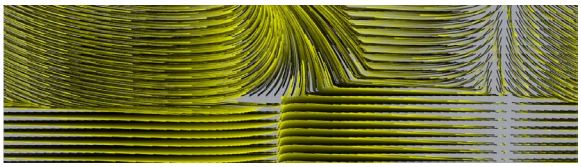
(a) Potential flow (b) Stokes flow

**Figure 3:** *Particles advected through the interpolated velocity fields in Figure 2.*

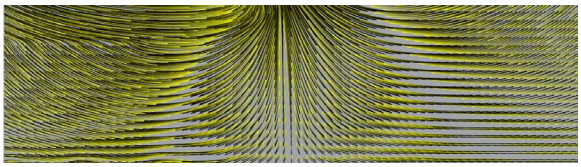


(a) Input (b) Potential flow (c) Stokes flow

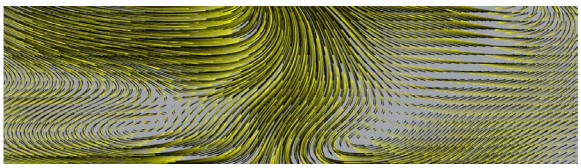
**Figure 4:** *A closeup of the breaking wave free surface liquid simulation in Figure 1. Steady state Stokes flow more distinctively preserves the curl of the wave.*



(a) Velocity extrapolation



(b) Potential flow



(c) Stokes flow

**Figure 5:** *A slice – parallel to wave's direction of travel – of the velocity field interpolated from the input mesh in Figures 1.a and 4.a. The rotation is only captured by steady state Stokes flow.*

Stokes flow on the other hand captures the tangential components of the surface velocities and creates internal structures visible when advecting particles in the interpolated velocity field (Figure 3.b). In a production environment this gives artists the ability to define complex particle flows for effects by means of painting.

In Figures 1, 4 and 5, we evaluate steady state Stokes flow in the context of a breaking wave modelled as a triangle mesh by an effects artist. Surface velocities are derived from finite difference calculations on the vertices of the mesh. In Figure 5, the interpolated velocity in a slice through the wave geometry – parallel to the direction of travel of the wave – is shown. Although velocity extrapolation preserves tangential components it does not capture the rotation present in the wave and introduces discontinuities near the medial axes. Potential flow results in a smooth but irrotational solution whereas steady state Stokes flow interpolation introduces local rotations in addition to producing a smooth solution. Figure 1 shows the wave and controlled free surface liquid simulations from a distance. Figure 4 shows a closeup as the first wave breaks. It is important to note that the steady state Stokes flow velocities are used only as a force in a dynamic full simulation. In particular the free surface liquid simulation is controlled by applying a feedback force-density $f = (\mathbf{v}_{\text{target}} - \mathbf{v}_{\text{liquid}})/\Delta t$ at grid cells further than a distance of $4\Delta x$ away from the surface of the input shape, where $\mathbf{v}_{\text{target}}$ is the interpolated target velocity, $\mathbf{v}_{\text{liquid}}$ is the

current liquid velocity, $\Delta t$ is the time-step and $\Delta x$ is the grid spacing. This control technique can be combined with the method of guide shapes [NB11] to obtain faster simulations. As can be seen, the steady state Stokes flow interpolation does not make the liquid flow follow the input geometry exactly. There are several reasons for this. Firstly, we apply the feedback force a certain distance away from the surface, meaning that the liquid near the surface is evolving freely. Secondly, the thin lip of the wave in the input mesh is not accurately captured on the voxelized simulation grid. Our current implementation is just single-threaded, and might benefit from a more sophisticated preconditioned. We timed steady state Stokes flow and potential flow computations on an Intel Xeon 2.66GHz CPU. For the breaking wave, the steady state Stokes flow interpolation takes on average of 40.4 seconds per frame to converge to a relative residual of $10^{-2}$ with 1.6M unknowns. 13.3 seconds are spent assembling the matrix and 27.1 seconds required for the actual solve. Our multigrid based Potential flow interpolation takes 4.4 seconds. For the cylinder steady state Stokes flow takes 1.73 seconds (0.73 seconds for the matrix assembly and 1.0 seconds for the solve) to converge to a relative residual of $10^{-3}$ with 68K unknowns. Potential flow takes 0.35 seconds. In conclusion, our potential flow implementation is currently 5-10 times faster. We leave as future work to investigate how a more sophisticated solver, preconditioner and parallelization will affect the relative performance. However, we emphasize that the interpolation only has to be performed once for a given input shape, and for all our examples, the velocity interpolation in one frame is independent of all other frames. Hence all frames can be computed in parallel and the overhead of using steady state Stokes flow does not depend on the number of frames, provided enough processing power is available. Although potential flow outperforms Stokes flow in running time, the purpose Stokes flow serves is quite unique. For certain complicated geometry potential flow will never be able to achieve as detailed rotational surface motion as Stokes flow.

## 5. Conclusion and Future Work

We explored steady state Stokes flow in the context of velocity interpolation for fluid control in computer graphics. Contrary to previous work on velocity interpolation for fluid control, steady state Stokes flow captures the rotation present in the input surface velocity. This improves simulation results where the input shape exhibits strong rotational motions such as breaking waves. Scenes involving water bodies with rotational components of velocity are quite common in movies. Artists can use this technique to create rotational motion automatically, which otherwise could be quite hassling.

## Acknowledgments

We wish to thank the following individuals for their help and support of our work: Joe Letteri, Sebastian Sylwan, Kevin Romond, Christoph Sprenger, Dave Gouge, Natasha Turner as well as Brian Goodwin and Diego Trazzi.

## References

[ESW05] ELMAN H., SILVESTER D., WATHEN A.: *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical mathematics and scientific computation. Oxford University Press, 2005. 2

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. In *ACM SIGGRAPH 2004 Papers* (2004), pp. 441–448. 2

[HK04] HONG J.-M., KIM C.-H.: Controlling fluid animation with geometric potential: Research articles. *Comput. Animat. Virtual Worlds 15*, 3-4 (2004), 147–157. 2

[HMK11] HUANG R., MELEK Z., KEYSER J.: Preview-based sampling for controlling gaseous simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 177–186. 2

[Lau05] LAUTRUP B.: *Physics of Continuous Matter*. IOP Publishing Ltd, 2005. 2

[MMS04] MIHALEF V., METAXAS D., SUSSMAN M.: Animation and control of breaking waves. In *Proc. ACM/Eurographics Symp. Comp. Anim.* (2004), pp. 315–324. 2

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 449–456. 1, 2

[NB11] NIELSEN M. B., BRIDSON R.: Guide shapes for high resolution naturalistic liquid simulation. In *ACM SIGGRAPH 2011 papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 83:1–83:8. 2, 4

[NC10] NIELSEN M. B., CHRISTENSEN B. B.: Improved variational guiding of smoke animations. *Comput. Graph. Forum 29*, 2 (2010), 705–712. 2

[NCZ*09] NIELSEN M. B., CHRISTENSEN B. B., ZAFAR N. B., ROBLE D., MUSETH K.: Guiding of smoke animations through variational coupling of simulations at different resolution. In *Proc. ACM/Eurographics Symp. Comp. Anim.* (Aug. 2009), pp. 206–215. 2

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D. Q., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R. P.: Directable photorealistic liquids. In *Proc. ACM/Eurographics Symp. Comp. Anim.* (2004), pp. 193–202. 2

[SY05a] SHI L., YU Y.: Controllable smoke animation with guiding objects. *ACM Trans. Graph. 24*, 1 (2005), 140–164. 2

[SY05b] SHI L., YU Y.: Taming liquids for rapidly changing targets. In *Proc. ACM/Eurographics Symp. Comp. Anim.* (2005), pp. 229–236. 2

[TKPR06] THÜREY N., KEISER R., PAULY M., RÜDE U.: Detail-preserving fluid control. In *Proc. ACM/Eurographics Symp. Comp. Anim.* (2006), pp. 7–12. 2

[TMPS03] TREUILLE A., MCNAMARA A., POPOVIĆ Z., STAM J.: Keyframe control of smoke simulations. In *ACM SIGGRAPH 2003 Papers* (2003), pp. 716–723. 1, 2