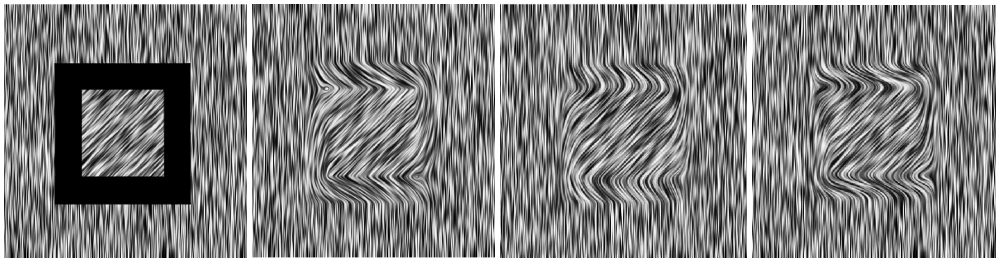
Editing Fluid Flows with Divergence-Free Biharmonic Vector Field Interpolation

Tümay Özdemir
University of Waterloo
Waterloo, ON, Canada
tozdemir@uwaterloo.ca

Jiamin Shi
University of Waterloo
Waterloo, ON, Canada
jiamin.shi@uwaterloo.ca

Nathan King
University of Waterloo
Waterloo, ON, Canada
n5king@uwaterloo.ca

Christopher Batty
University of Waterloo
Waterloo, ON, Canada
christopher.batty@uwaterloo.ca



(a) Input Flows **(b) Potential Flow** **(c) Stokes Flow** **(d) Ours**

Figure 1: Comparison of potential flow, Stokes flow, and our constrained biharmonic interpolation applied to fill in the missing black region shown in (a). In this challenging case, potential flow (b) and Stokes flow (c) interpolation, while divergence-free, only match velocity *values* at the boundaries. Our constrained biharmonic interpolation (d) additionally matches velocity *gradients*, yielding smoother transitions across the boundaries, especially at the top and bottom borders.

Abstract

Achieving a satisfying fluid animation through numerical simulation can be time-consuming and there are few practical post-processing tools for editing completed simulations. To address this challenge, we present a divergence-free biharmonic vector field interpolation method that can be used to perform smooth spatial blending between input incompressible flows. Given flow data on the boundary of a desired interpolation domain at each time step, we fill in the given domain by constructing an optimally smooth, divergence-free, boundary-satisfying vector field. We ensure smoothness using the Laplacian energy and enforce divergence constraints through Lagrange multipliers. Prior methods for this problem suffer from visible artifacts due to non-zero divergence and discontinuous velocity gradients. By then replacing the Laplacian energy with the Hessian energy we further extend our method to extrapolation in the presence of open boundaries. We demonstrate that our approach produces smooth and incompressible flows, which enables a range of natural simulation editing capabilities: copy-pasting, hole-filling, and domain extension.

CCS Concepts

• **Computing methodologies** → **Procedural animation**; **Physical simulation**.

Keywords

vector field interpolation, incompressible fluid, animation, biharmonic equation, Hessian

1 Introduction

Artistic modification of fluid motion is a challenging problem in computer animation. One simultaneously wishes to (mostly) respect the governing physics, which gives the fluid its natural appearance, and at the same time produce an artistically desirable outcome. Repeatedly adjusting scene parameters and rerunning simulations from scratch can be impractical because of the relatively high computational cost. A potentially promising approach is to mix and match pieces of existing simulations, relying on a velocity interpolation procedure to spatially blend them together [Sato et al. 2018]. This approach can enable intuitive operations such as copy-pasting, hole-filling, and domain extension. The central mathematical task is to fill a volumetric region with a smooth incompressible vector field, given only information on the region's boundary.

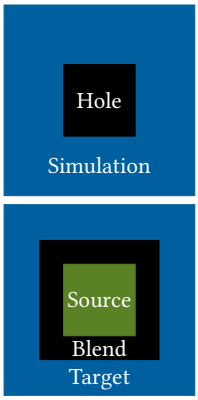
Existing methods possess key limitations. The method of Sato et al. [2018] fails to consistently produce divergence-free velocity fields: if the input flows are not well-aligned, significant expansion or contract occurs within the interpolation region. Earlier authors achieved strict incompressibility by using a potential flow-based interpolation [Nielsen and Bridson 2011] and later, to achieve better preservation of rotational motion, by solving a Stokes flow problem

[Bhattacharya et al. 2012]. Unfortunately, the generated vector field only matches the prescribed boundary velocity *values*, but not their *gradients*, i.e., only C^0 continuity. Consequently, undesirable derivative discontinuities along the border of the interpolation region can emphasize the presence of the numerical boundaries and disrupt the illusion of a natural flow.

To address this challenge, we develop a novel divergence-free biharmonic interpolation technique for fluid animation editing. We pose this task as an optimization problem over the interpolation region, where we minimize an appropriately chosen smoothness energy subject to the incompressibility constraint. The result is an optimally smooth field that simultaneously matches prescribed velocities *and* velocity gradients on the boundary. The use of Lagrange multipliers for the constraints couples the velocity components to ensure a divergence-free field. We demonstrate our new vector field interpolation tool on various flow interpolation scenarios.

2 Problem Statement

We seek to fill a volumetric domain Ω with a divergence-free fluid vector field $\mathbf{u}$, given prescribed vector field data on its boundary, $\partial\Omega = \Gamma$, i.e., *incompressible vector field interpolation*. The incompressibility assumption implies an additional *compatibility* condition on the input boundary velocities: they must integrate to zero so a consistent, incompressible interior field exists. (This stipulation can be straightforwardly upgraded to a prescribed net in/out flux, if desired.) For time-dependent simulations, the process should be performed for every timestep.



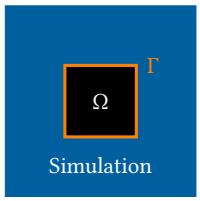
The simplest example is a vector field hole-filling scenario, where Ω is a region of an existing simulation that has been deleted and must now be replaced (see inset top). For example, if a region of a simulation yields a local motion that turns out to be undesirable (e.g., due to an object, external force, etc.) one can simply delete that region and interpolate a new vector field to replace it.

Another possibility is to combine two (or more) previously computed simulations in a copy-and-paste fashion as follows. Select the region to be copied in the source simulation. Place it into the desired location in the target simulation. Add a *blend region* of a user-defined width around the source region (see inset bottom), which will provide space to smoothly interpolate between the simulations. Finally, using the boundary data from the source and target regions, interpolate a new vector field for this blend region. This copy-and-paste methodology generalizes to multiple or even nested flow regions.

By slicing a simulation into a series of layers along one axis, spacing them apart, and filling the voids via interpolation/extrapolation, a *scene-stretching* mechanism can be supported, as shown by Sato et al. [2018]. Similarly, they showed how multiple distinct simulations, computed in parallel, can be patched together as a basic domain decomposition strategy. We additionally develop a smooth, divergence-free *extrapolation* technique by introducing support for open rather than prescribed boundaries, allowing greater flexibility and seamless domain extension.

3 Mathematical Background

Our approach is motivated by a sequence of increasingly expressive mathematical formulations.



Vector Harmonic Interpolation. Consider the properties the new field $\mathbf{u}$ should possess. First, as one approaches Γ from the interior of Ω, $\mathbf{u}$ should approach the given boundary data, denoted $\mathbf{u}_b$. Second, $\mathbf{u}$ should be smooth. With just these two stipulations, a good choice is *harmonic interpolation* [Joshi et al. 2007] (adapted to vector data), which can be expressed as minimizing the vector Dirichlet energy (e.g., [Stein et al. 2020]):

$$\begin{aligned} &\arg\min_{\mathbf{u}} \int_\Omega \frac{1}{2}\|\nabla \mathbf{u}\|_F^2, \\ &\text{subject to } \mathbf{u} = \mathbf{u}_b \text{ on } \Gamma. \end{aligned} \tag{1}$$

However, a third desired field characteristic is incompressibility, which makes the Dirichlet energy insufficient: it completely fails to control the field's divergence (Figure 2; refer to the supplemental document for this and subsequent figure references). Subsequently projecting the resulting $\mathbf{u}$ to be incompressible as a post-process sacrifices some of the result's smoothness. More critically, since pressure projection implies only a *free-slip* condition (matching boundary normal component), tangential velocity discontinuities arise at boundaries.

Potential Flow. Incompressibility can be strictly enforced through a potential flow (irrotationality) assumption: let $\mathbf{u} = \nabla\phi$ for a scalar potential ϕ [Nielsen and Bridson 2011; Shi and Yu 2005]. Then simply minimizing the kinetic energy of the flow, subject to matching boundary (normal) velocities, gives

$$\begin{aligned} &\arg\min_{\phi} \int_\Omega \frac{1}{2}\|\nabla \phi\|^2, \\ &\text{subject to } \nabla\phi \cdot n = \mathbf{u}_b \cdot \mathbf{n} \text{ on } \Gamma. \end{aligned} \tag{2}$$

The minimizer of (2) satisfies a scalar Laplacian problem, $\nabla\cdot\nabla\phi = 0$, with Neumann boundary conditions. Since $\nabla\cdot\nabla\cdot\phi = \nabla\cdot\mathbf{u} = 0$, the resulting flow is incompressible. Unfortunately, as discussed for pressure projection above, only normal continuity is ensured and tangential free-slip artifacts arise again. Bhattacharya et al. [2012] further demonstrate that the irrotationality assumption sacrifices important rotational motions, as we show in Figure 3.

Stokes Flow. The strengths of harmonic interpolation (smoothness, no-slip boundaries) and potential flow (incompressibility) can be combined by adding an explicit divergence-free constraint to (1):

$$\begin{aligned} &\arg\min_{\mathbf{u}} \int_\Omega \frac{1}{2}\|\nabla \mathbf{u}\|_F^2, \\ &\text{subject to } \nabla\cdot\mathbf{u} = 0 \text{ on } \Omega, \\ &\qquad\qquad \mathbf{u} = \mathbf{u}_b \text{ on } \Gamma. \end{aligned}$$

Applying constraints with Lagrange multiplier p, the optimality conditions yield the *steady Stokes equations* with unit viscosity,

$$\Delta\mathbf{u} - \nabla p = 0, \qquad \nabla\cdot\mathbf{u} = 0,$$

where the operator $\Delta = \nabla\cdot\nabla$ denotes the Laplacian. Bhattacharya et al. [2012] were the first to suggest using Stokes flow for fluid

interpolation and demonstrated its superior rotational behavior compared to potential flow. Our derivation above provides some additional intuition and exposes its variational underpinnings.

4 Our Proposed Methods

In typical applications where the newly interpolated region is inserted into a surrounding presimulated flow, the no-slip boundary condition $\mathbf{u} = \mathbf{u}_b$ is insufficient. The resulting combined vector field is only C^0 across Γ. The derivative is discontinuous on Γ, where the flow crosses from the presimulated region into the newly interpolated region. Visually, this can produce kinks in the flow that emphasize the borders of the interpolation domain (e.g., Figure 1(c)). We consider new formulations that address this limitation.

4.1 Divergence-Free Biharmonic Interpolation

Our approach is to upgrade our objective function from the vector Dirichlet to the vector Laplacian energy. This higher order energy necessitates providing an additional boundary condition, for which various options exist (see e.g., [Stein et al. 2018]). By specifying the vector field values *and* (first) derivatives across the boundary, we recover C^1 continuity and achieve the desired visual smoothness:

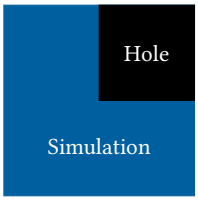
$$\begin{aligned} &\arg\min_{\mathbf{u}} \int_\Omega \frac{1}{2}\|\Delta \mathbf{u}\|^2, \\ &\text{subject to } \nabla \cdot \mathbf{u} = 0 \text{ on } \Omega, \\ &\qquad \mathbf{u} = \mathbf{u}_b \text{ on } \Gamma, \\ &\qquad \nabla \mathbf{u} \cdot \mathbf{n} = \nabla \mathbf{u}_b \cdot \mathbf{n} \text{ on } \Gamma. \end{aligned} \tag{3}$$

Again using a Lagrange multiplier p to enforce incompressibility, the optimality conditions yield the partial differential equations

$$\Delta^2 \mathbf{u} - \nabla p = 0, \qquad \nabla \cdot \mathbf{u} = 0, \tag{4}$$

where Δ^2 represents the biharmonic operator. The solution to these equations is our *divergence-constrained biharmonic interpolant*.

4.2 Divergence-Free Biharmonic Extrapolation



The formulation above is well-suited to interpolation, i.e., where data on the entire boundary Γ is prescribed. However, one might alternatively wish to fill in a region where parts of Γ lack specified values and derivatives. Consider the hole-filling task in a case where the region to be filled overlaps part of the exterior boundary (see inset). Artificial velocities and derivatives should not be specified, but instead an optimally smooth extension of the vector field with unspecified boundary conditions should be found. The natural boundary conditions of the Laplacian energy might seem a reasonable choice, but Stein et al. [2018] showed (in the scalar case) that these conditions do not provide the desired behavior, leading to boundary-dependent bias. Instead, they replace the Laplacian energy with the Hessian energy. The optimality conditions of the latter provably yield the same PDE operator on the interior (i.e., the biharmonic operator) but its natural boundary conditions are unbiased.

We apply this same insight to our problem, extending to the vector case by applying the Hessian energy componentwise. Denoting the Frobenius norm as $\|\cdot\|_F$, our formulation becomes:

$$\begin{aligned} &\arg\min_{\mathbf{u}} \int_\Omega \frac{1}{2}\|\mathbf{H}_\mathbf{u}\|_F^2, \\ &\text{subject to } \nabla \cdot \mathbf{u} = 0 \text{ on } \Omega, \\ &\qquad \mathbf{u} = \mathbf{u}_b \text{ on } \Gamma, \\ &\qquad \nabla \mathbf{u} \cdot \mathbf{n} = \nabla \mathbf{u}_b \cdot \mathbf{n} \text{ on } \Gamma. \end{aligned} \tag{5}$$

4.3 Discretization

Our mathematical formulation is quite general, but our particular discretization assumes that the input boundary data is drawn from standard grid-based fluid simulations, where velocity components are distributed in a staggered fashion on the cell faces. We similarly discretize the interpolation domain with a staggered grid, and place Lagrange multiplier (pressure) degrees of freedom at each grid cell center. The divergence-free constraint on each cell is formulated in the usual finite difference manner. The discrete biharmonic operator is formed by applying the classic Laplacian stencil twice, which leads to a 13-point stencil in 2D and a 25-point stencil in 3D (see e.g., [Altas et al. 2002; Chen et al. 2008] for explicit coefficients). Letting B_x, B_y, B_z represent the scalar biharmonic operator on each velocity component and D_x, D_y, D_z represent the components of the divergence operator for each cell, we have the linear system

$$\begin{bmatrix} B_x & 0 & 0 & D_x^T \\ 0 & B_y & 0 & D_y^T \\ 0 & 0 & B_z & D_z^T \\ D_x & D_y & D_z & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \end{bmatrix} = \mathbf{b}. \tag{6}$$

The solution is the set of interior cell face velocities u, v, w and interior pressures p. The right-hand-side vector $\mathbf{b}$ is initially set to zero. Then, when setting up the stencils for each unknown, wherever a stencil entry "touches" a sample that lies outside the interpolation region, we transfer its contribution to the right-hand-side using the known value at that location (drawn from the input simulations). This simple boundary condition treatment straightforwardly enforces continuous value and derivative conditions on velocity, as well as Neumann conditions on pressure. There is a one-dimensional pressure null space, which we eliminate by simply pinning one pressure degree of freedom to zero. (If expanding or compressing input boundary fluxes are provided, entries of $\mathbf{b}$ corresponding to the pressure DOFs would need to include compensating interior divergence values per cell to ensure compatibility.) For the Hessian-based extrapolation case, we replace the three B terms with the stencil derived by Stein et al. [2018]. Binary per-face weights are used to distinguish active/inactive cells while approximating the integral in (5) to enforce the natural boundary conditions.

As in the work of Sato et al. [2018], the flow of any material to be visualized/rendered (e.g., smoke densities and/or tracer particles) must be recomputed from scratch, because the new combined vector field will differ significantly from its inputs. That is, the data must be advected through the new velocity field to yield a consistent final result. Fortunately, this can often be done efficiently and in parallel, since each (passive) particle's motion affects no other particle.

4.4 Solution Procedure

The linear system is symmetric indefinite for which a wide array of techniques have been proposed [Benzi et al. 2005]. We adopt a

simple conjugate-gradient (CG) scheme as follows. Our system has the form

$$\begin{bmatrix} A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{7}$$

where $A = \begin{bmatrix} B_x & 0 & 0 \\ 0 & B_y & 0 \\ 0 & 0 & B_z \end{bmatrix}$, $C = \begin{bmatrix} D_x & D_y & D_z \end{bmatrix}$, and $\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$. Applying a Schur complement transformation to our system to eliminate the velocity variables, we arrive at an SPD system of the form $CA^{-1}C^T p = CA^{-1}f - g$. This allows us to solve for p with a standard CG solver. Using the solution for p we can recover u.

We take advantage of the property of Krylov solvers like CG not requiring the explicit matrix A^{-1}, but only the *action* of A^{-1} on a vector. That is, given x, only $y = CA^{-1}C^T x$ needs to determined.

To evaluate y we first use Cholesky factorization to decompose A into LL^T. The necessary matrix-vector product y is computed in a three-step process. First multiply $C^T x$, then solve $LL^T z = C^T x$ for z, and finally compute $y = Cz$. In the common case where the shape and location of the interpolation region does not change over time, the Cholesky factorization is computed once and reused.

5 Results

We now consider some illustrative scenarios to demonstrate our method. Figures that use passive marker particles alternate colours in initially horizontal rows to highlight the flow structure.

Our first scenario in Figure 4 consists of a static solid disk in a vertical wind-tunnel, with inflow at the top and outflow at the bottom. We wish to paste the disk from the source simulation into an empty vertically translating wind-tunnel target simulation. This yields a smooth divergence-free combination of the two flows, where the flow outside the blend region is completely undisturbed. Our result necessarily differs from the source animation, since the presence of the disk *globally* disturbed the flow in the source simulation; our interpolation approach must therefore deform the flow more strongly in the blend region to compensate, yet we still achieve a visually plausible flow (Figure 5).

To stress-test our method, we consider some challenging scenarios analogous to those by Sato et al. [2018]. We combine flows where the source and target differ in *direction*. In Sato's approach, more severe failures of the divergence-free condition are seen with larger direction deviations (c.f. the second supplemental video of [Sato et al. 2018]). Figure 6 has the same setup as Figure 4 except we change the ambient flow direction of the source simulation to have steadily increasing angles. While this leads to an increasingly unnatural look, the resulting flow field is still smooth on the blend region interior and divergence-free independent of this artistic decision.

In Figure 7, we paste a disk obstacle into a scene containing three rectangles. The flow structure is complex due to the additional obstacles. We tightened the blend region to fit more closely around the paste region, but a plausible flow is still constructed. Finally, in Figure 8, we use Hessian extrapolation to extend a velocity field into a region with unspecified boundary conditions. Even though the extrapolated flow (c) differs from the original flow on the full domain (a), the extrapolation is still plausible.

6 Limitations and Future Work

We have presented a new vector field interpolant for simulation editing based on a vector biharmonic equation with added incompressibility constraints. We present 2D examples, but the method will naturally apply in 3D also.

Our work suggests several directions to explore in future work. First, we assumed that the interpolation region borders are voxelized, and saw no artifacts from doing so. However, it would be interesting to explore a cut-cell approach [Batty et al. 2007] for regions with irregular shapes, based on a more precise discretization of the relevant energies. This would add greater flexibility and potentially make region borders less apparent.

The biharmonic operator does not offer a maximum principle similar to the Laplacian, and thus it is possible for interpolated velocities to overshoot the boundary data. However, since we use the resulting velocity only for advecting passive data, rather than advecting velocity, such overshoots cannot feed back into the simulation and cause instabilities. If this were deemed undesirable, enforcing bounds using inequality constraints is an option [Jacobson et al. 2011], though this may harm smoothness of the boundaries.

References

Irfan Altas, Jocelyne Erhel, and Murli M Gupta. 2002. High accuracy solution of three-dimensional biharmonic equations. *Numerical Algorithms* 29, 1 (2002), 1–19.

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 100–es.

Michele Benzi, Gene H Golub, and Jörg Liesen. 2005. Numerical solution of saddle point problems. *Acta numerica* 14 (2005), 1–137.

Haimasree Bhattacharya, Michael Bang Nielsen, and Robert Bridson. 2012. Steady State Stokes Flow Interpolation for Fluid Control.. In *Eurographics (Short Papers)*. Citeseer, 57–60.

Guo Chen, Zhilin Li, and Ping Lin. 2008. A fast finite difference method for biharmonic equations on irregular domains and its application to an incompressible Stokes flow. *Advances in Computational Mathematics* 29, 2 (2008), 113–133.

Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (2011), 78.

Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 71–es.

Michael B. Nielsen and Robert Bridson. 2011. Guide shapes for high resolution naturalistic liquid simulation. *ACM Trans. Graph. (SIGGRAPH)* 30, 4 (2011), 1.

Syuhei Sato, Yoshinori Dobashi, and Tomoyuki Nishita. 2018. Editing fluid animation using flow interpolation. *ACM Trans. Graph.* 37, 5 (sep 2018), 1–12.

Lin Shi and Yizhou Yu. 2005. Taming liquids for rapidly changing targets. In *Symposium on Computer Animation*. 229–236.

Oded Stein, Eitan Grinspun, Max Wardetzky, and Alec Jacobson. 2018. Natural boundary conditions for smoothing in geometry processing. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 1–13.

Oded Stein, Max Wardetzky, Alec Jacobson, and Eitan Grinspun. 2020. A Simple Discretization of the Vector Dirichlet Energy. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 81–92.

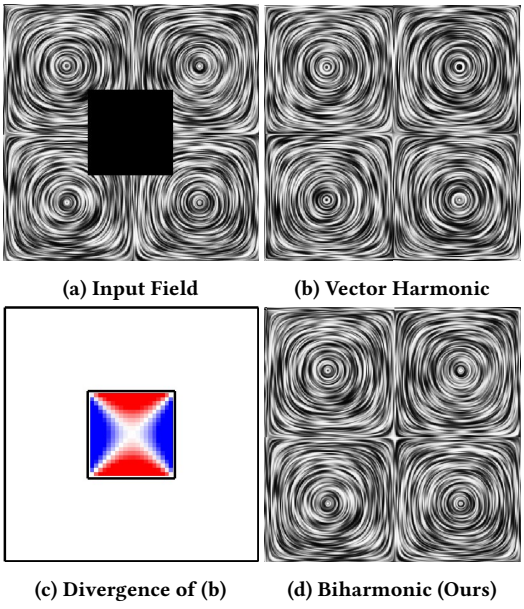
Editing Fluid Flows with Divergence-Free Biharmonic Vector Field Interpolation - Supplemental

Tümay Özdemir
University of Waterloo
Canada
tozdemir@uwaterloo.ca

Jiamin Shi
University of Waterloo
Canada
jiamin.shi@uwaterloo.ca

Nathan King
University of Waterloo
Canada
n5king@uwaterloo.ca

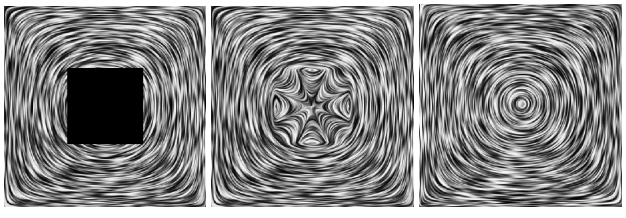
Christopher Batty
University of Waterloo
Canada
christopher.batty@uwaterloo.ca



(a) Input Field (b) Vector Harmonic

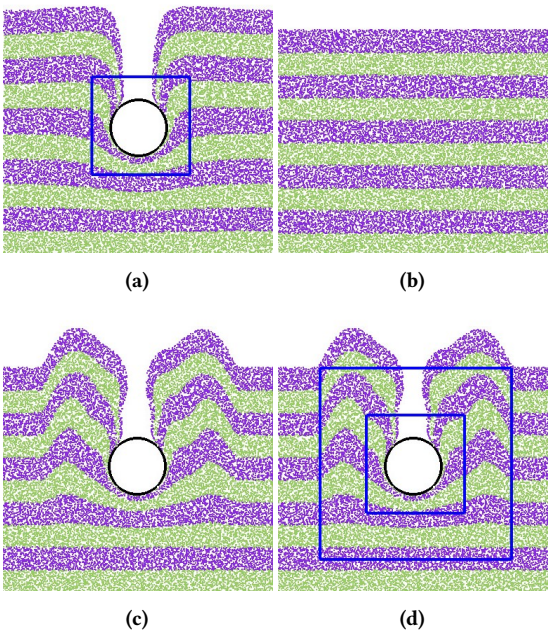
(c) Divergence of (b) (d) Biharmonic (Ours)

Figure 2: Vector harmonic interpolation vs. our constrained biharmonic interpolation on a four-vortex hole-filling test. (a) Input simulation with black indicating the hole to be filled. (b) Result of vector harmonic interpolation. (c) Divergence of the field in (a), where positive and negative values are red and blue, respectively. (d) Our method looks similar to (b) but has uniformly zero divergence. (The line integral convolution visualization above does not indicate velocity magnitudes.)



(a) Input Field (b) Potential Flow (c) Biharmonic (Ours)

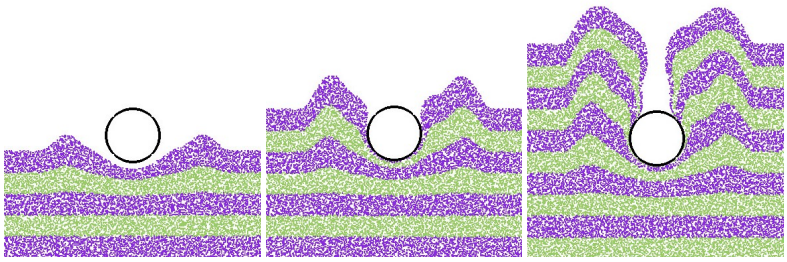
Figure 3: Potential flow interpolation vs. our constrained biharmonic interpolation on a rotational hole-filling test. A square hole in the middle of the image (a) is filled in using (b) potential flow interpolation and (c) our method. Although potential flow is divergence-free, it cannot recover the expected rotation and yields an unnatural result.



(a) (b)

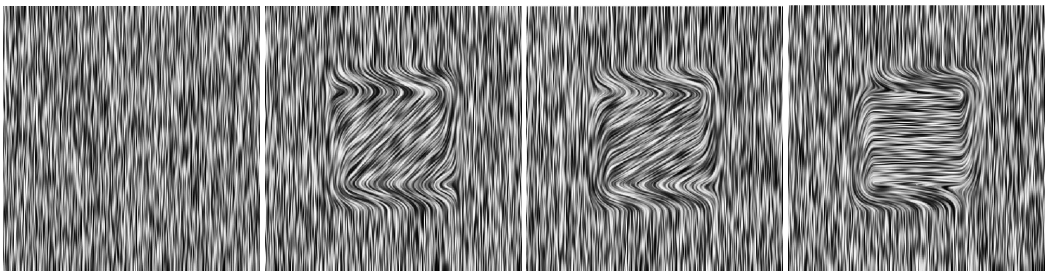
(c) (d)

Figure 4: Copying the flow around a disk from its source simulation within the blue square (a) into an obstacle-free target simulation (b). The result of our method is a new smooth, divergence-free combined flow (c). In (d) our combined flow is overlaid with blue lines denoting the inner and outer borders of the blend region where our interpolation is applied.



(a) (b) (c)

Figure 5: Three frames of the edited animation result using our approach for the scenario described in Figure 4.



(a) (b) (c) (d)

Figure 6: Combining two velocity fields with different angles. (a) A vertical target velocity field. (b) The target field combined with a 45-degree angled source flow pasted inside. (c) The target field combined with a 60-degree angled source flow. (d) The target field combined with a 90 degree-angled source flow. A smooth, divergence-free velocity field is always obtained despite extreme angle differences, contrasting with the work of Sato et al.

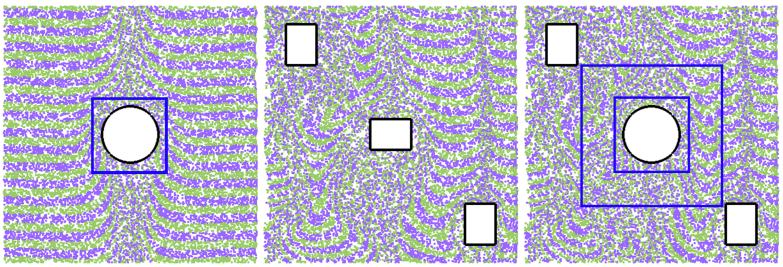
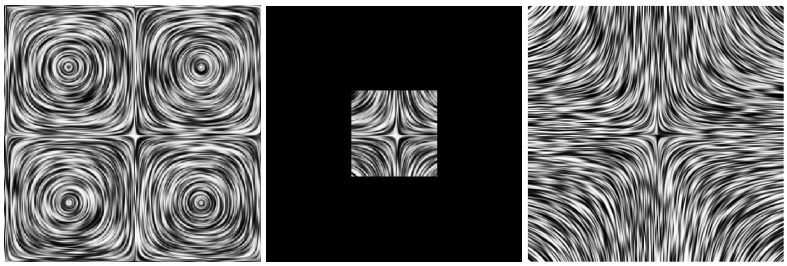


Figure 7: Replacing a rectangle with a disk in a flow with additional obstacles using a narrow blend region. From left to right: source scene, target scene, result.



(a) Original Flow (b) Input Flow (c) Hessian-based flow

Figure 8: Hessian extrapolation (c) is used to extend the input flow (b). The extrapolation (c) differs from the original flow over the full domain (a) but is still plausible, smooth, and incompressible.