Fluidic Topology Optimization with an Anisotropic Mixture Model

YIFEI LI, MIT CSAIL, USA
TAO DU, MIT CSAIL, USA
SANGEETHA GRAMA SRINIVASAN, University of Wisconsin-Madison, USA
KUI WU, LightSpeed Studios, Tencent, USA
BO ZHU, Dartmouth College, USA
EFTYCHIOS SIFAKIS, University of Wisconsin-Madison, USA
WOJCIECH MATUSIK, MIT CSAIL, USA

Fig. 1. We present a topology optimization pipeline for designing Stokes-flow fluidic systems with flexible and accurate boundary conditions. Our method automatically creates the structure of this fluidic twister on a 100 × 100 × 100 grid after optimizing nearly four million decision variables. The goal of this device is to generate a swirl flow at its outlet given a constant inflow. Left: our final design is made of spatially-varying anisotropic materials, which we visualize as a small disk in each voxel colored based on its anisotropic direction (bottom-left inset). Our method automatically synthesizes a propeller-like structure (top-right inset) to facilitate the vortex generation near the outlet. Middle: the flow simulated from the final design, visualized as streamlines. A vortex emerges near the outlet (top-right inset). Right: visualization of flows from the fluidic device after 1, 10, 20, and 50 iterations of topology optimization.

Fluidic devices are crucial components in many industrial applications involving fluid mechanics. Computational design of a high-performance fluidic system faces multifaceted challenges regarding its geometric representation and physical accuracy. We present a novel topology optimization method to design fluidic devices in a Stokes flow context. Our approach is featured by its capability in accommodating a broad spectrum of boundary conditions at the solid-fluid interface. Our key contribution is an anisotropic and differentiable constitutive model that unifies the representation of different phases and boundary conditions in a Stokes model, enabling a topology optimization method that can synthesize novel structures with accurate boundary conditions from a background grid discretization. We demonstrate the efficacy of our approach by conducting several fluidic system design tasks with over four million design parameters.

CCS Concepts: • Computing methodologies → Physical simulation.

Additional Key Words and Phrases: Topology optimization, Stokes flow, computational design, fluidic system design

ACM Reference Format:
Yifei Li, Tao Du, Sangeetha Grama Srinivasan, Kui Wu, Bo Zhu, Eftychios Sifakis, and Wojciech Matusik. 2022. Fluidic Topology Optimization with an Anisotropic Mixture Model. ACM Trans. Graph. 41, 6, Article 239 (December 2022), 16 pages. https://doi.org/10.1145/3550454.3555429

1 INTRODUCTION

Fluidic systems play a vital role in today’s industry and engineering, supporting applications from jet engines, hydraulic actuators, to heart valves and bioreactors. Computational design of a fluidic system that manifests precise functionality and complex geometry remains as a substantial challenge. Despite the rapid advent of additive manufacturing, which enabled the fabrication of intricate flow-driven systems on an unprecedented level of printing resolution, the computational exploration of the design space of even a simple flow “twister” (one that converts a laminar input flow to a swirling pattern at its outlet) remains a difficult task due to the...
interleaving complexities of the flow physics simulation, topological structure exploration, and accurate boundary representations.

We identify two underlying challenges for building a fluidic system design framework. First, an accurate flow simulator needs to enforce both the incompressibility constraint and accurate boundary conditions (i.e., slip and non-slip) within a topologically complicated domain. Especially, when a fluid structure gets narrow, an accurate model to characterize the solid boundary’s impacts on the near-boundary flow behavior is essential for evaluating the system’s performance and design sensitivities. Conventional approaches, as a strict adherence to a non-slip boundary condition (which is a modeling hypothesis rather than physical principles), could effectively “clog up” such narrow fluid pathways, and hinder the optimization process to generate new design features. Second, a capable optimizer is expected to effectively explore the high-dimensional design space of both shape and topology without being constrained by any parameterization priors. Traditional shape optimization frameworks, despite their ability in featuring the local geometry of the solid-fluid boundary accurately, could not generate new topological features that differ drastically from the current shapes. Considering these two aspects, we need to carefully choose the design’s discrete representation that can balance its geometric expressiveness (i.e., accurately featuring the local shape) and topological complexities (i.e., freely evolving the global topology).

Traditional fluidic design frameworks, categorized into field-based methods and shape-based methods according to their design representations, suffer from a number of limitations. In particular, current field-based approaches lack an accurate boundary representation, and shape-based approaches are limited in topological flexibility. A field-based approach (e.g., see [Borrvall and Petersson 2003]) represents the fluid domain using a density field discretized on a background grid. Akin to the volume-of-fluid (VOF) method [Hirt and Nichols 1981], the density of each cell specifies the fraction between fluid and solid phases occupying the cell’s volume. Such fraction-based representation, like in VOF, suffers from its inherent ambiguities in reconstructing the accurate geometry of a sharp interface and therefore lacks its ability in enforcing accurate boundary conditions.

Shape-based approaches, exemplified by the implicit level sets [Fedkiw and Osher 2002] and explicit parametric shapes [Du et al. 2020], lack their flexibility in exploring complex topological changes. In particular, the lack of ability in tackling topological changes such as merging, splitting (for parametric shapes), and adding/removing holes (for level sets), will constrain the algorithm’s exploration of the design space within a limited scope. A method that can combine the merits of both field-based and shape-based approaches, despite their successes in solving forward simulation problems in computational physics, such as the Coupled Level-Set and Volume of Fluid (CLSVOF) [Sussman and Puckett 2000] and particle level-set method [Enright et al. 2002], remains largely unexplored in the field of computational design of fluidic systems.

To address these two challenges, we propose a non-parametric topology optimization framework enhanced by accurate boundary treatments to enable large-scale fluidic system designs. The critical challenge we addressed in this work is to devise a geometric representation that can express the phase (solid or fluid), sharp interface (with boundary normals), and anisotropy (with local flow directions) in a unified geometric representation. Our method was inspired by the anisotropic material model for elastic simulation [Li and Barbic 2015] and the diffusive imaging model in Magnetic Resonance Imaging (MRI) [Basser et al. 1994], which use tensor fields to encode the local, anisotropic geometry. We propose to represent the solid, fluid, and their boundary as an anisotropic tensor field discretized on a background grid, which enables us to accurately handle different boundary types and maintain flexibility in evolving topology. Based upon this novel representation, we further formulated the differentiable simulation and optimization models in conjunction with our novel block-based incompressibility constraint to explore designs in a high-dimensional parameter space. Compared with the flow optimization literature, our design system tackles topologically complicated flow design problems by expressing its spatially filling, multi-phase, and heterogeneous material features in a continuous and unified fashion. We validate the efficacy of our approach in multiple fluidic device design problems with as many as two million design parameters, many of which showed for the first time designs with intricate solid structures and free-slip flow fields in complex fluid domains, which were impractical for previous methods.

We summarize the main contributions of our paper as follows:

- We propose an anisotropic Stokes flow model, as well as its discretization scheme, numerical solver, and gradient computation, which jointly enable flexible modeling of both free-slip and no-slip boundary conditions.
- We propose an approach to incorporate volume-preserving constraints aggregating in rectangular regions that improve the conditioning of our system.
- We propose a field-based topology optimization framework for computational optimization of fluidic systems with accurate, flexible solid-fluid boundaries.
- We demonstrate the capacity of our framework for obtaining a variety of complex fluidic devices design.

2 RELATED WORK

Flow optimization. Beginning with the pioneering work of Borrvall and Petersson [2003], a vast literature has been devoted to the optimization of fluid systems [Alexandersen and Andreassen 2020]. Given a predefined design domain with boundary conditions, a typical optimization objective is to maximize some performance functional of a fluid system (e.g., the power loss of the system) constrained by the physical equations. Similar to a conventional structural optimization problem, the design domain is discretized. The optimization algorithm decides for each element whether it should be fluid or solid to optimize some performance function such as the power loss. Examples of flow optimization applications include Stokes flow [Aage et al. 2008; Borrvall and Petersson 2003; Challis and Guest 2009; Guest and Prévost 2006], steady-state flow [Zhou and Li 2015], weakly compressible flow [Evgrafov 2006], unsteady flow [Deng et al. 2012], channel flow [Gersborg-Hansen et al. 2005], ducted flow [Othmer et al. 2007], viscous flow [Kontoleontos et al. 2013], fluid-structure interaction (FSI) [Andreassen and Sigmund 2013; Casas and Pavanello 2017; Yoon 2010], fluid-thermal interaction [Matsumori et al. 2013; Yaji et al. 2015], microfluidics
We present an overview of our method in Fig. 2. The input to our method is the specification of a fluidic device defined as the target inlet and outlet flow profiles. The fluidic device is represented using a regular grid filled with anisotropic materials in each voxel. The materials are parametrized with scalar fields describing its anisotropy, viscosity, and impedance for flow. The material distribution induces a multi-phase fluidic device design whose solid-fluid boundaries can be extracted from cells with highly anisotropic materials. A numerical differentiable simulator then simulates the design to compute its Stokes flow, which is compared with the target outlet flow profile to evaluate its performance. The pipeline computes the gradients of the performance metric with respect to material parameters and backpropagating them through the numerical differentiable simulator. The pipeline then runs MMA [Svanberg 1987], the standard gradient-based optimizer in topology optimization, to improve the performance of the design by evolving its anisotropic material distribution. After this optimization converges, the resulting design is computed by a post-processing step to extract the design surface.

We organize the remainder of our paper as follows. We first describe the governing equations for our Stokes flow model in Sec. 4. Then, we discuss its discretization and the numerical solver in Sec. 5. We next formulate the fluidic system design problem as a numerical optimization problem and present our optimization algorithm in Sec. 6. Finally, we present applications and evaluation of our method in Sec. 7 and provide conclusions in Sec. 8.

4 GOVERNING PARTIAL DIFFERENTIAL EQUATIONS

This section describes the physical model of the Stokes flow problem used in this paper. While Stokes equations have been extensively studied for decades, we revisit this problem with a focus on developing a novel anisotropic constitutive model that jointly represents different phases (solid and fluid) and boundary conditions (no-slip and free-slip) in a unified manner. Our constitutive model provides a uniform, grid-friendly parametrization of the design space of fluidic devices without sacrificing the flexibility and accuracy in solid-fluid boundary conditions.

4.1 Isotropic Stokes Equations

Quasi-incompressible Stokes flow We briefly review the quasi-incompressible Stokes flow model described in Du et al. [2020]. Consider a problem domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or $3$). The velocity field $v : \Omega \rightarrow \mathbb{R}^d$ of the quasi-incompressible Stokes flow is given by the following energy minimization problem:

$$
\min_{v} \int_{\Omega} \mu \|
abla v\|^2_{L^2} dx + \int_{\Omega} \lambda ( \nabla \cdot v )^2 dx,
$$

(1)

s.t. $v(x) = v_D(x)$, \quad \forall x \in \partial \Omega_D, \quad v(x) \cdot n(x) = 0, \quad \forall x \in \partial \Omega_F.$

(2)

Note that Eqn. (1) excludes the external-force energy defined in Du et al. [2020] because we assume no external forces (e.g., gravity) in our design problem. Here, the notation $\| \cdot \|$ is the Frobenius norm of a matrix, and $\mu \in \mathbb{R}^+$ and $\lambda \in \mathbb{R}^+$ are two scalar parameters denoting the flow’s dynamic viscosity and incompressibility, respectively. In particular, $\lambda \rightarrow +\infty$ implies perfectly incompressible Stokes flow. The problem considers the following boundary conditions defined on a partition of the domain boundary $\partial \Omega = \partial \Omega_D \cup \partial \Omega_F \cup \partial \Omega_G$: The Dirichlet boundary condition specifies a desired velocity profile $v_D$ on the boundary $\partial \Omega_D$, which is either from prescribed inlet
Anisotropic, quasi-incompressible Stokes flow problems, and the free-slip boundary condition defined on $\partial Q_f$ requires the velocity’s projection along the normal direction $n$ to be zero; finally, the open boundary on $\partial Q_O$ imposes no explicit constraints on the velocity and automatically satisfies zero-traction conditions once the energy in Eqn. (1) is minimized, which is suitable for modeling free flows at an outlet of a fluidic system.

Although we do not consider external forces or non-zero-traction boundary conditions in our problem, they can be accommodated in a similar way described in Du et al. [2020]. We refer readers to Du et al. [2020] for a comprehensive discussion on the derivation of this quasi-incompressible Stokes flow model and its numerical benefits in fluidic device design problems. While their paper presented a computational design pipeline for fluidic devices and demonstrated examples with moderately sophisticated solid structures, the method constrained the designs in the space of parametric shapes, which inhibits topologically different designs from emerging.

4.2 Anisotropic Stokes Equations

The challenges in previous methods motivate us to develop a new geometric representation that simultaneously accommodates expressive topology, flexible boundary conditions, and accurate simulation in Stokes-flow fluidic systems. Noting that fluid near solid-fluid boundaries satisfies different physical constraints in the normal and tangent directions, we propose an anisotropic material model that uniformly represents solid, fluid, and solid-fluid boundaries, which we describe in detail below.

**Anisotropic, quasi-incompressible Stokes flow.** We propose the following energy minimization problem that modifies the previous isotropic Stokes flow:

$$
\min_{\mathbf{v}} \ E_{m,\mu}[\mathbf{v}] + E_{m,\lambda}[\mathbf{v}] + E_f[\mathbf{v}],
$$

s.t. \ $\mathbf{v}(x) = \mathbf{v}_D(x), \ \forall x \in \partial Q_D$.  \hspace{1cm} (4)

where each energy component is defined as follows:

$$
E_{m,\mu}[\mathbf{v}] : = \int_B \mu \|
\nabla \mathbf{v} K_m^1(x) \|
^2 dx,
$$

$$
E_{m,\lambda}[\mathbf{v}] : = \int_B \lambda(x) (\nabla \cdot \mathbf{v})^2 dx,
$$

$$
E_f : = \int_B \| K_f^2(x) \mathbf{v} \|^2 dx,
$$

where we use the subscripts $m$ and $f$ in the energy names to indicate they model the material and the frictional effects, respectively. Note that this formulation incorporates the standard, isotropic Stokes model as a special case, namely by setting $K_m = I$ and $K_f = 0$. The new energy minimization problem introduces a few critical modifications to the original problem in Eqns. (1-3): First, we change the problem domain from $\Omega$ to $B \subset \mathbb{R}^d$, which we assume to be an axis-aligned, sufficiently large box that encloses the fluidic region $\Omega$. Second, we introduce two symmetric positive semi-definite matrix fields $K_m, K_f : B \rightarrow S_d^+$ and replace the scalar parameter $\lambda$ with a spatially varying field $\lambda : B \rightarrow \mathbb{R}^+$. These three new fields define a new material model that enables anisotropic responses to velocities at different directions. Finally, with the domain changing from $\Omega$ to a box-shaped $B$, we adjust the boundary conditions as follows: We consider a partition of the boundary $\partial B$ into $\partial B = \partial B_D \cup \partial B_O$ where $\partial B_D$ and $\partial B_O$ represent the locations of the Dirichlet boundary and the open boundary conditions, respectively. The Dirichlet boundary $\partial B_D$ now consists of the inlet of the fluidic system where we enforce a prescribed flow profile and the border of the solid phase, on which we directly assign zero velocities. The open boundary $\partial B_O$ still models a zero-traction, free-flow outlet like before. The new formulation of boundary conditions does not mean we exclude no-slip or free-slip solid-fluid boundary conditions in our problem, however. In fact, solid-fluid boundaries are now absorbed into the interior of $B$ and will be represented by a careful choice of $K_m, K_f$, and $\lambda$. We illustrate the new Stokes flow model in Fig. 3.
we will show that by properly setting them everywhere in Eqns. (1-3) and Eqns. (4-5). Moreover, by choosing $\mathbf{K}_m$ to be its unit normal vector. We augment $\mathbf{n}$ with two unit vectors $t_1, t_2$ orthogonal to $\mathbf{n}$ so that the matrix $\mathbf{R} := (\mathbf{n}, t_1, t_2)$ defines an orthonormal basis in $\mathbb{R}^3$.

To derive a proper $\mathbf{K}_f$ for free-slip boundaries, we recall that the isotropic energy $\|\mathbf{v}\|_F^2$ in the original Stokes flow model is the variational form of $\Delta \mathbf{v}$, the Laplacian of the velocity field component-by-component, in the PDE form of Stokes flow [Borrvall and Petersson 2003; Du et al. 2020]. Intuitively, this states that Stokes flow creates a component-by-component as-harmonic-as-possible velocity field, subject to the (quasi-)incompressibility constraint.

We further point out that both the Frobenius norm and the Laplacian of the velocity field component-by-component are rotationally invariant, allowing us to change the coordinate systems freely when computing $\nabla \mathbf{v}$. Therefore, we can consider computing $\nabla \mathbf{v}$ in a local frame spanned by the normal and tangent directions at $\mathbf{x}$, i.e., the columns of $\mathbf{R}$:

$$\mathbf{v}_R := \mathbf{R}^T \mathbf{v},$$

$$\mathbf{x}_R := \mathbf{R}^T \mathbf{x},$$

$$\nabla\mathbf{x}_R \mathbf{R} = \mathbf{R}^T \nabla \mathbf{v} \mathbf{R}.$$  

Here, the subscript $(\cdot)_R$ means the quantity is defined in the local frame spanned by $\mathbf{R}$. For example, the first row in $\nabla \mathbf{x}_R \mathbf{v}_R$ represents the spatial gradient of the normal flow magnitude.

It is now straightforward to see how the physical intuition behind free-slip boundaries can motivate the definition of $\mathbf{K}_f$: Essentially, free-slip boundaries retain the physical property of Stokes flow along the tangent directions and dismiss any spatial gradients along the normal direction. From the perspective of the Laplacian operator, this means directly dropping the second-order derivative along the normal direction and requiring the flow to be harmonically smooth only along the tangent directions. Mapping it back to the variational form, we see it is equivalent to zeroing out the column in $\nabla \mathbf{x}_R \mathbf{v}_R$ that corresponds to the normal direction, leading to the following energy density to be integrated in $E_{m,\mu}$:

$$\Psi_{m,\mu} := \mu \|\nabla \mathbf{x}_R \mathbf{v}_R\Lambda(0, 1, 1)\|^2_F$$

$$= \mu \|\mathbf{R}^T (\nabla \mathbf{v}) \Lambda(0, 1, 1)\|^2_F$$

$$= \mu \|\nabla (\nabla \mathbf{v}) \mathbf{R}(0, 1, 1)\|^2_F$$

$$= \mu \text{trace}((\nabla \mathbf{v}) (I - \mathbf{n} \mathbf{n}^T) (\nabla \mathbf{v})^T)$$

where $\Lambda$ constructs a diagonal matrix from its input. Comparing Eqn. (16) with Eqn. (6), we see $\mathbf{K}_m$ should be defined as follows:

$$\mathbf{K}_m = I - \mathbf{n} \mathbf{n}^T.$$  

Similarly, we propose the following $\mathbf{K}_f$ to impede the normal flow in $E_f$:

$$\mathbf{K}_f = \mathbf{R} \Lambda(k_f, 0, 0) \mathbf{R}^T = k_f \mathbf{n} \mathbf{n}^T,$$

where $k_f \rightarrow +\infty$. Plugging this definition into $E_f$ in Eqn. (8) it will confirm that the proposed $\mathbf{K}_f$ leads to the expected behavior: It first converts $\mathbf{v}$ to a local frame spanned by $\mathbf{R}$ then forces the normal component of the flow to be zero and keeps the tangent flow intact.

ACM Trans. Graph., Vol. 41, No. 6, Article 239. Publication date: December 2022.
Finally, regarding the choice of $\lambda$, we set its value to 0 so that $E_{\text{m,}\lambda}$ is ignored for points on free-slip boundaries. We justify this decision from two perspectives: First, using a positive $\lambda$ in this case will attempt to preserve fluidic volume at free-slip boundaries, which typically means enforcing zero divergence in a neighborhood near the boundary after discretization. For example, it may suggest that a mixed cell’s solid and fluid velocities be divergence-free even though they come from different phases, implying a physically problematic constraint imposed on the discretized problem. Such constraints typically lead to conditioning issues in the numerical system after discretization that calls for a more careful, subcell-accurate treatment of these mixed cells. Second, and more importantly, the real objective that incompressibility plays near the boundary is to avoid fluid leakage out of the boundary, which is automatically satisfied as long as we disallow normal flows crossing the boundary and ensure the interior of the fluid phase is properly incompressible. Both reasons imply that it is unnecessary to consider a positive $\lambda$ that encourages the (ill-defined) divergence at locations on free-slip boundaries to be zero, hence our decision to set $\lambda = 0$.

Summary. To conclude, we have presented an anisotropic material model which provides a uniform representation of different phases and boundary conditions encountered in the quasi-incompressible Stokes flow problem. We summarize the anisotropic material parameters $K_m$, $K_f$, and $\lambda$ for all cases in Table 1.

Table 1. We summarize the anisotropic material parameters for modeling different phases and boundary conditions in the quasi-incompressible Stokes problem: $\lambda_0 \in \mathbb{R}^n$ represents a predefined scalar parameter controlling the incompressibility in the quasi-incompressible Stokes flow; $k_f \in \mathbb{R}^n$ is a scalar parameter that determines the material’s impedance for fluid flow, with $k_f \to +\infty$ creating the solid phase; $n \in \mathbb{R}^d$ is the unit normal vector on the solid-fluid boundary that decides the material’s anisotropic responses to normal and tangent flows.

| $K_m$ | Fluid | Solid | No-slip boundary | Free-slip boundary |
|------|-------|-------|------------------|-------------------|
| $K_f$ | 0 | $k_f I$ | $I - nn^T$ |
| $\lambda$ | $\lambda_0$ | $k_f I$ | $k_f n n^T$ |

5 NUMERICAL SIMULATION

This section outlines the discretization scheme and the numerical solver for the continuous Stokes flow model described in the previous section. The Stokes flow system, described in Section 4, is discretely modeled over the entire domain and parameterized to represent fluid, solid, and fluid-solid interface regions on the domain. Our discretization enables easy specification of boundary conditions, allows seamless application of additional design constraints over the domain, such as volume fraction, and accommodates a highly flexible parameter space that helps optimize for complex designs for fluid-solid interfaces. We use a uniform Cartesian lattice to discretize domain descriptors and state variables. Fluid velocities $\nu$ are stored at grid nodes and bilinearly (2D) or trilinearly (3D) interpolated, while the design parameters $K_m$, $K_f$, and $\lambda$ are treated as having a constant value on each grid cell (as we will see, they are stored indirectly, through some other parameters that are ultimately stored per-cell). We design a solver that solves for flow for a given parameter set, using a symmetric positive-definite (SPD) stiffness matrix. The solver also includes additional constraints, such as block divergence, to satisfy the flow divergence constraint in an aggregate fashion.

5.1 Anisotropic Material Parametrization

Our anisotropic material is characterized by SPD matrix fields $K_m$ and $K_f$ and a scalar field $\lambda$ in the whole domain. From Table 1, we notice that these fields can be induced from the solid/fluid material distributions and boundary normals, which have lower degrees of freedom than the full SPD matrices. This inspires us to reparametrize $K_m$, $K_f$, and $\lambda$ with three new fields: the fluidity $\rho : \mathcal{B} \to [0, 1]$, which assigns 0 to pure solid and 1 to pure fluid; the isotropy $\epsilon : \mathcal{B} \to [0, 1]$ in which larger $\epsilon$ means more isotropic material; and the anisotropic orientation $\alpha : \mathcal{B} \to \mathbb{R}^{d-1}$, which is a field of rotational angles in 2D and spherical coordinates in 3D. Furthermore, we compute a unit normal field $n : \mathcal{B} \to \mathbb{R}^d$ induced from $\alpha$.

Constructing $K_m$. We define the material matrix field $K_m$ as follows:

$$K_m = I - (1 - \epsilon)\rho nn^T.$$  \hfill (19)

Therefore, $K_m \approx I$ whenever $\epsilon \approx 1$ (isotropic material) or $\rho \approx 0$ (solid phase), and $K_m$ becomes highly anisotropic only if $\epsilon \approx 0$ and $\rho \approx 1$, i.e., anisotropic fluid that we use to represent free-slip boundaries.

Constructing $K_f$. We define $k_f$ with the following nonlinear mapping function borrowed from previous work [Borrval and Petersson 2003]:

$$k_f(\rho) = k_{f,\max} + (k_{f,\min} - k_{f,\max})\rho \frac{1 + q}{\rho + q},$$  \hfill (20)

where $k_{f,\min} = 0$ and $k_{f,\max} = 1e5$ indicate the range of $k_f$ and $q = 0.1$ is a hyperparameter that controls the sharpness of the mapping: A smaller $q$ generates a more binary mapping with the output $k_f$ concentrating on the bound values. Intuitively, $k_f(\rho)$ is a monotonically decreasing function that maps small $\rho$ (solid phase) to $k_{f,\max}$ and large $\rho$ (fluid phase) to $k_{f,\min}$. We then define $K_f$ as follows:

$$K_f = k_f(\rho)I + (k_f(\epsilon) - k_f(\rho))nn^T.$$  \hfill (21)

We can see such a definition matches what Table 1 suggests for each type of material: For fluid phase, which satisfies $\epsilon \approx 1 \approx \rho$, we have $k_f(\epsilon) \approx k_f(\rho) \approx 0$, and therefore $K_f \approx 0$; for solid phases and no-slip boundaries, we have $\epsilon \approx 1 \approx \rho \approx 0$, which leads to a large $k_f(\rho)$ and $K_f \approx k_{f,\max}I$; finally, for free-slip boundaries, we model them as anisotropic fluid with $\epsilon \approx 0$ and $\rho \approx 1$, which means $k_f(\epsilon) \gg k_f(\rho) \approx 0$ in the definition, and it follows that $K_f \approx k_{f,\max}nn^T$.

Constructing $\lambda$. Finally, we construct the $\lambda$ field as follows:

$$\lambda = \lambda_{\min} + [1 - (1 - \epsilon)\rho]^2\lambda_{\max},$$  \hfill (22)

where $\lambda_{\min} = 0.1$ and $\lambda_{\max} = 1e3$ sets the range of $\lambda$ and $\rho = 12$ is a power index that pushes $\lambda$ to be a binary choice between $\lambda_{\min}$ and $\lambda_{\max}$. 


and $\lambda_{\text{min}} + \lambda_{\text{max}}$. Similar to $K_m$ defined above, this definition of $\lambda$ requires that $\lambda \approx \lambda_{\text{min}}$ only if $\varepsilon = 0$ and $\rho = 1$, i.e., near free-slip boundaries.

Summary. In this manner, the anisotropic material distribution in the entire domain is parameterized by three fields $\rho$, $\epsilon$, and $\alpha$. Comparing the results above with Table 1, we can see that these new fields implement the intended properties of the anisotropic material in fluid, solid, and flexible solid-fluid boundaries.

5.2 Discretization Scheme

We now describe our discretization scheme for solving the Stokes flow problem in Sec. 4. We solve the energy minimization problem defined by Eqns. (4-5) on a regular grid of $N^d$ cells and store all physical quantities at the center of each cell except for flow velocities, which we store at grid nodes. We first describe how we discretize the material parameters. Next, we discretize the variational form and introduce block divergence, a novel numerical treatment of the (quasi-)incompressibility constraint that improves the condition number of the numerical system to be solved.

Discretizing material parameters. We discretize the continuous fields $\rho$, $\epsilon$, and $\alpha$ on the regular grid by storing their values at each cell center. For brevity, we use $\theta$ from now on to refer to the set of discretized material parameters $\rho$, $\epsilon$, and $\alpha$:

$$\theta := \{\rho, \epsilon, \alpha\}.$$  

(23)

The induced fields $K_m$, $K_f$, and $\lambda$ are also computed and stored at these cell centers based on the mapping described previously. These parameters are treated as constant throughout the spatial extent of each cell; quadrature rules that are discussed next that need to access such parameters at quadrature point locations throughout the cell will just use the same constant value assigned to such cell.

Discretizing fluid velocity. We discretize and store the fluid velocity field $\mathbf{u}$ on the nodes of each cell in the grid. Then, we compute the velocity and its spatial gradients at any point with bi/trilinear interpolation of the nodal values.

Discretizing variational energy terms. With the material parameters and fluid velocity fully discretized, we can now discretize the variational form in Eqns. (4-5). Here, we will strategically use a different numerical integration scheme for each of the energy terms in Eqns. (6-8) as detailed next:

For the term $E_{m,M}$ in Eqn. (6) – we can label this the Anisotropic Laplace term, recognizing that the isotropic case $K_m = I$ would yield the Laplace term in the PDE version of Stokes, as in Du et al. [2020] – we use the Gaussian-Legendre quadrature rule to integrate this energy numerically within each grid cell. Hence, we employ four quadrature points in 2D and eight quadrature points in 3D, in a fashion directly analogous to prior work [Du et al. 2020]. As mentioned, we use the single value of $K_m$ associated with the cell at all quadrature points.

The term $E_{m,D}$ in Eqn. (7) can be labeled the cell-wise divergence term, and essentially seeks to enforce volume preservation at each individual cell where it is applied with $\lambda \neq 0$. For a given grid cell $C$, we approximate this term with the following expression:

$$E_{m,D}^C \approx \lambda C W_C \left[ \frac{1}{W_C} \int_C (\nabla \cdot \mathbf{u}) \, dx \right]^2$$

where $W_C$ is the volume ($h^2$ in 2D, $h^3$ in 3D) of the cell $C$. Intuitively, what this approximation suggests is that instead of penalizing a non-zero divergence at each individual interior point of the cell, we only seek to drive the net flux \(\text{Flux}(C) = \int_C (\nabla \cdot \mathbf{u}) \, dx\) to zero, that treats the entire cell in an aggregate fashion (allowing a non-zero divergence at interior locations, as long as the aggregate net flux is zero). This is an important modification to the numerical discretization that circumvents locking behaviors for highly incompressible materials, and has been shown effective and compatible with mixed FEM formulations for highly incompressible materials [Patterson et al. 2012]. What is more, since the integrand is linear in the nodal velocities (when using bilinear/trilinear interpolation), it is possible to obtain an analytic expression for the flux, namely in 2D (if $\mathbf{u} = (u, v)$ are the individual scalar velocity components):

$$\text{Flux}(C) = h \left( \frac{u_{11} + u_{11}}{2} + \frac{v_{11} + v_{11}}{2} - \frac{u_{01} + u_{01}}{2} - \frac{v_{01} + v_{01}}{2} \right)$$

(24)

where the subscripts denote each of the four cell vertices. The four terms of this expression can be easily and intuitively indentified as, respectively, the (average) signed fluxes through the right, top, left, and bottom faces of the cell. We would exactly arrive at this same analytic expression if we simply used a 4-point Gauss quadrature for the flux integral, since the linear integrand would be integrated exactly. An exactly analogous expression can be derived in 3D; the only difference is that we would create the average velocity of each face by averaging four nodal velocities, and the scalefactor of $h^2$ would replace $h$ in Eqn. (24) to account for the area of each 3D cell.

Finally, for the term $E_f$ in Eqn. (8) we employ a quadrature scheme that uses the cell vertices themselves as quadrature points, namely:

$$E_f^C = \frac{W_C}{2^d} \sum_I \| K_f^C(C) \mathbf{v}_I \|^2_2$$

where the index $I$ traverses all cell vertices (4 in 2D; 8 in 3D). We observe that this term seeks to model a cell as viscous/rigid (in the solid phase) or permeable (in the fluid phase), hence it is perfectly reasonable to apply this viscous penalty on a per-vertex basis. Doing so, in fact, avoids certain hazards of low-order quadrature schemes (for example a single-point quadrature scheme evaluated at the cell center could yield a zero result even if non-zero velocities at the vertices happen to average out to zero at the cell center).

After combining all terms in this discretization scheme, we arrive at a quadratic energy form, given by $E = \mathbf{v}^T K \mathbf{v} - \mathbf{b}^T \mathbf{v}$, where $\mathbf{v}$ stacks all velocity degrees of freedom and $K$ and $\mathbf{b}$ the SPD matrix and vector composed of the material parameters, respectively. We enforce the Dirichlet boundary condition by forcing the corresponding nodal velocities to be the desired value. Putting them together, we state the discrete variational form as the following quadratic programming problem:

$$\min_{\mathbf{v}} \quad \mathbf{v}^T K(\theta) \mathbf{v} - \mathbf{b}(\theta)^T \mathbf{v}$$

s.t. $\mathbf{v}_I = (v_D)_I, \forall (i, (v_D)_I) \in D,$

(25)

(26)

where $D$ states the Dirichlet boundary conditions.
5.3 Block divergence
The variational form and the anisotropic material parametrization frequently use large hyperparameters to satisfy constraints that are supposed to be strict, e.g., perfect incompressibility is approximated with $E_{\text{m,lambda}}$ penalizing the divergence with a coefficient $\lambda_{\text{max}} \to +\infty$, and free-slip boundaries are replaced with $E_f$ scaled by $(k_f)_{\text{max}} \to +\infty$ that penalizes nonzero normal flows. Unfortunately, while setting such parameters to near infinity tightens these constraints from an optimization perspective, it also leads undesirable behaviors. Specifically, using an exceptionally high $\lambda$ value is known to compromise the conditioning of the system. Using a high $k_f$ value has an even more obscure side effect; since boundary cells use this penalty to prevent flows from having a component in the normal direction to the boundary, if there are any two neighboring boundary cells that have even slightly different directions of anisotropy, a high $k_f$ value would effectively cause the velocities at any shared vertices between the two cells to be driven to zero, as their projection to two non-parallel directions (the normals at the two cells) will jointly and strongly be required to be zero. Hence, this might inadvertently once again drive us to a situation where we are unintentionally forcing a no-slip condition.

Using moderately-high values for $\lambda$ and $k_f$ certainly helps alleviate these issues. The risk of doing so, however, is that we open up the possibility for volume loss that, even not egregious at the local level, could add up to a substantial flow loss at the global scale. This is aggravated by the fact that (with sound motivation) we do not enforce incompressibility (we use $\lambda = 0$) at boundary cells paired with a free-slip condition, depending on the zero-normal-flow condition (which is not strictly enforced if $k_f$ is not very high) to prevent leakage at free-slip boundaries.

We introduce an original solution to this issue, by pairing the moderately-high parameters at the per-cell level with a hard-constraint of absolute volume preservation at a more aggregate scale, namely large blocks (typically rectangular boxes of 4 or 8 cells across) that we partition our domain into. This is illustrated in Figure 11 (right); we refer to the associated ablation study for an illustration of the effect of this technique (or the consequences of its omission).

We partition the domain $\mathcal{B}$ into large blocks (indexed by $b$) of uniform sizes and enforce the (aggregate) net flux over each block to be zero. Since the per-cell flux is a linear expression on the nodal velocities, the aggregate constraint is simply the sum:

$$0 = \text{Flux}(B_b) = \sum_{C \in B_b} \text{Flux}(C)$$

As one would intuitively expect, the net flux over the block $B_b$ is a linear expression of the averaged (signed) fluxes through the faces of the aggregate block; the contribution of faces interior to $B_b$ cancels out when fluxes of neighboring cells are summed. Ultimately, the net flux constraint on each block yields a single linear equation (with a zero right-hand side), and these block-divergence constraints for the entire domain are ultimately distilled into a linear constraint system $C\nu = 0$. Ultimately, we reformulate our governing anisotropic Stokes equations as a linearly-constrained, quadratic optimization problem, where we minimize the functional $E(\nu) = \nu^T [K(\theta) \nu - b(\theta)]^T \nu$ (resulting from the quadrature schemes in the previous paragraph), subject to the linear constraint $C\nu = 0$; $\theta$ represents the design parameters.

We ultimately solve this constrained optimization problem via the Karush-Kuhn-Tucker condition, in the system:

$$\begin{pmatrix} 2K(\theta) & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \nu \\ q \end{pmatrix} = \begin{pmatrix} b(\theta) \\ 0 \end{pmatrix}$$

(where $q$ are the Lagrange multipliers associated with the block-divergence constraint). We use direct factorization methods (we employ PARDISO [Alappat et al. 2020]) to solve these symmetric/ indefinite problems both for forward simulation as well as for the inverse problems associated with optimization; we may omit, for brevity, an explicit reference to the constraint in our upcoming discussion of the optimization pipeline, with the understanding that it is always present in our pipeline. From a practical perspective, we found this technique to be highly effective; in our 3D examples, enforcing a hard block-divergence constraint on 4x4x4 blocks, paired with moderate to moderately-high parameters $\lambda$ and $k_f$ produced results that were practically indistinguishable from using a very high $\lambda$, and much more resilient to artifacts associated with using high $k_f$ values. Note that the block-divergence constraint does not depend on the design parameters $\theta$, contrasted to $K$ and $b$ that do. Ultimately, we view the solution of this constraint optimization problem as the simulation function $\nu = F(\theta)$ that maps the design parameters to the corresponding simulation result.

6 OPTIMIZATION
We now describe our optimization problem which is built upon the numerical simulator described before. We first formulate the fluidic device design task as a numerical optimization problem and state its formal definition. Next, we describe the algorithm for solving this optimization problem numerically.

6.1 Problem Statement
We formally define the task of designing a fluidic device as the following numerical optimization problem:

$$\min_{\theta} L_f(\nu) + w_{L_c}(\theta) + w_{d}(\theta) + w_{L_d}(\theta),$$

s.t. $\nu = F(\theta)$,

$$\theta_{\text{min}} \leq \theta \leq \theta_{\text{max}}.$$ (27)

$$0 \leq V_{\text{min-fluid}}(\theta) \leq V_{\text{max}}.$$ (29)

$$0 \leq V_{\text{all-fluid}}(\theta) \leq V_b + V_{\text{max}}.$$ (30)

Here, $L_f$ states the functional loss given by the task specification, which is typically defined as the $L_2$ difference between the outlet flow in simulation and the desired outlet flow profile.

Regularizers. The next three terms in the objective are regularizers on the material parameter $\theta$. Following the standard practice in the previous field-based fluidic topology optimization method [Borrval and Petersson 2003], the compliance regularizer $L_c$ computes the elastic energy accumulated from enforcing the outlet flow profile to be the same as the target as extra Dirichlet constraints:

$$L_c(\theta) := \nu_{e}^T K(\theta) \nu_{e} - b(\theta)\nu_{e},$$ (32)

$$\nu_{e} = F(\theta; \mathcal{D}) \cup \mathcal{D}_O,$$ (33)

where $\mathcal{D}_O$ summarizes the extra Dirichlet conditions from the target outlet flow profile. The motivation behind $L_c$ is that a lower elastic
energy means the outlet flow profile will be more similar to the
target if we release the extra Dirichlet conditions on the outlet.
Next, the directional regularizer $L_d$ is defined as the differences
between the normal direction of an anisotropic cell and the nor-
mals from its neighborhood. We first filter out cells that contain
anisotropic materials with two thresholds $\epsilon_0$ and $\rho_0$: a cell is
considered to be anisotropic if its $\epsilon < \epsilon_0$ and $\rho > \rho_0$, i.e., it is modeled
as anisotropic fluid. Let $\mathcal{A}$ be the set of anisotropic cells, we define
$L_d$ as follows:

$$L_d := \sum_{c \in \mathcal{A}} 1 - \psi\cos(\alpha_c, \alpha_{nbr})$$

(34)

where $\alpha_c$ is the anisotropic direction (a unit vector) of cell $c$, $\alpha_{nbr}$
an average direction fitted from its small neighbors ($3 \times 3 \times 3$ in
our implementation), and $\psi\cos$ the cosine similarity between them.
Therefore, minimizing $L_d$ encourages smooth free-slip solid-fluid
boundaries.

Finally, the anisotropic regularizer $L_a$ concentrates anisotropic
cells near solid-fluid boundaries:

$$L_a(\epsilon) := \sum_c \epsilon_c p_c (\rho_{\text{max}} - \rho_{\text{local}})$$

(35)

where the sum loops over each cell $c$ and $\rho_{\text{max}}$ and $\rho_{\text{local}}$ are
the maximum and minimum fluidity from its small neighborhood
($3 \times 3 \times 3$ in our implementation). The proposed loss encourages
cells near the solid-fluid boundaries (large $\rho_{\text{max}} - \rho_{\text{local}}$) to use
anisotropic materials (small $\epsilon$). To avoid chasing a moving target, we
freeze $\rho_{\text{local}}$ and $\rho_{\text{max}}$ in $L_d$ when optimizing $\rho$ in each iteration.

Constraints. Apart from the objective function, the optimization
problem also contains a number of constraints: Eqn. (28) ensures the flow field $\nu$ is computed from the numerical simulator, and Eqn.
(29) states the bound constraints on the material parameters. The
last two equations (30) and (31) define volume constraints on the
fluidic region and the free-slip fluid-solid boundaries:

$$V_{\text{iso-fluid}} := \sum_c \epsilon_c p_c$$

(36)

$$V_{\text{all-fluid}} := \sum_c p_c$$

(37)

The difference between them implies the volume of anisotropic cells,
and $V_{\text{max}}$ and $V_{\text{lb}}$ are two task-dependent thresholds defining the
maximum fluidic volume and the maximum volume of anisotropic
cells (free-slip boundaries), respectively.

Summary. In summary, the optimization problem aims to find an
optimal material distribution $\theta$ that minimizes the functional loss
under bound constraints and volume constraints. The optimization
process is facilitated by three regularizers that encourage spatially
smooth yet clear solid-fluid boundaries.

6.2 Numerical Optimizer

Standard topology optimization typically consists of hundreds of
thousands of decision variables even for moderate-size problems
(e.g., on a $64^3$ grid), and our numerical optimization is no excep-
tion. In fact, due to the inclusion of additional anisotropic material
parameters, the numerical optimization problem we formulated
has a larger number of decision variables than its isotropic topol-
ogy optimization counterparts. Following the standard practice in
topology optimization, we use the method of moving asymptotes
(MMA) [Svanberg 1987], a widely used gradient-based optimization
algorithm, to solve this large-scale optimization problem. As MMA
requires gradients with respect to the material parameters $\theta$, we ex-
tended the numerical simulation method in Sec. 5 to a differentiable
simulator in a way similar to Du et al. [2020], through which the
gradients can be computed from backpropagating the loss function.
To encourage more structured designs during the optimization pro-
cess, we dynamically update the upper bound on the isotropy $\epsilon_c$ for
each cell $c$ based on the following heuristic:

$$(\epsilon_c)_{\text{max}} := 1 - (\rho_{\text{local}} - \rho_{\text{local}}_{\text{max}}).$$

(38)

In other words, when a cell is surrounded by both solid and fluid
cells, we force it to choose anisotropic materials (small upper bound
on $\epsilon$). Our empirical experience suggests that this dynamic update
scheme biased the optimization process towards generating more
structured fluidic device designs.

6.3 Choice of Parameters and Interpolation Functions

For the hyperparameter settings, we reused the value from [Borrvall
and Petersson 2003] for $k_{\text{min}}$. We chose a near-zero value for $\lambda_{\text{min}}$.
The choices of these two hyperparameters followed the convention
in topology optimization. We did not observe noticeable differences
when their values were perturbed. For $k_{\text{max}}$ and $\lambda_{\text{max}}$ values, we
chose values that can balance the solver performance and block
divergence (See our discussion in Sec. 5.3). We chose the current
block size by experimenting with a minimum block size that gives
satisfactory free-slip boundaries without introducing large diver-
gence errors in small neighborhood. We include an experiment on the
sensitivity of our method to block size in the supplementary
materials. For the choices of interpolation functions, we used the
interpolation function (Eq. 20) in [Borrvall and Petersson 2003] with
the same $q$ value for $k_f$, and we used a power-indexed interpolation
function for $\lambda$, which is conventional in density-based topology
optimization, to encourage the design’s binary convergence. We did
not observe noticeable differences between these two functions.

7 RESULTS

In this section, we present various 3D design problems to evaluate
the performance of our differentiable anisotropic Stokes flow simu-
lator as well as the optimization pipeline. Next, we compare our
method with two previous state-of-the-art baseline methods and
validate our method with ablation studies. A complete demonstra-
tion of our design problems with the evolution of our optimization
process can be found in our supplemental video.

7.1 Applications

We demonstrate a variety of complex fluidic device designs obtained
using our optimization pipeline. We use a grid resolution of $100 \times
100 \times 100$ for the Fluid Twister example and $80 \times 80 \times 80$ for all other
optimization examples and initialize the material parameters to be
isotropic ($\epsilon = 1$) with fluidity $\rho = V_{\text{max}}$. We run all optimizations
with 300 iterations and use the design that achieves minimum final

ACM Trans. Graph., Vol. 41, No. 6, Article 239. Publication date: December 2022.
We report the statistics from optimizing the design problems in Sec. 7.1, including the maximum fluidic volume fraction, the functional loss before and after optimization, and the time decomposition in each optimization iteration: “Forward” represents the time spent on computing the loss function, which is dominated by the numerical simulator; “Backprop.” stands for the time spent on computing the gradients; “MMA optimizer” reports the time cost from running one iteration of MMA after obtaining the loss and gradients; “Total” is the sum of all the time above.

| Design Problem | Resolution | Volume Limit ($V_{\text{max}}$) | Functional Loss ($L_F$) | Time per Optimization Iteration (s) |
|----------------|------------|---------------------------------|------------------------|------------------------------------|
|                |            |                                 | Initial | Final | Forward | Backprop. | MMA optimizer | Total |
| Twister        | 100x100x100| 0.30                            | 22.575 | 0.519 | 1374.1  | 152.6     | 153.2       | 1679.9 |
| Tree Diffuser  | 80x80x80   | 0.25                            | 3.966  | 0.133 | 721.9   | 85.0      | 61.1        | 868.0  |
| Circuit-1      | 80x80x80   | 0.25                            | 86.281 | 2.125 | 737.2   | 86.5      | 83.1        | 896.7  |
| Circuit-2      | 80x80x80   | 0.50                            | 86.153 | 1.490 | 721.3   | 86.4      | 82.7        | 889.7  |

loss as our optimized design. We report the task-specific volume fraction limit, initial and optimized functional loss, and the execution time for each task in Table 2. We additionally include design domain illustration and task specifications in supplementary materials.

We implement our optimization pipeline in C++ and use the implementation of PARDISO [Alappat et al. 2020] for solving our linear systems and MMA [Svanberg 1987] as our optimizer. Since the sparsity pattern of the system matrix remains the same over optimization iterations, we also optimize the matrix factorization time by performing symbolic factorization only once. We perform our experiments on an Intel Ice Lake 128-core server and Ubuntu 20.04 operating system.

**Motivating examples.** As motivating examples, we present the 3D design problems of an amplifier and a mixer under a 80 $\times$ 80 $\times$ 80 grid. The 2D versions of both examples are presented in prior works [Borrvall and Petersson 2003; Du et al. 2020]. In Amplifier, we enforce a constant circular input with inflow velocity ($v_{\text{in}}$, 0, 0). The objective is to amplify the flow by a factor of $\frac{5}{2}$. In Mixer, the design objective is to mix a high-pressure and low-pressure at two inlets to produce equal middle-pressure flows at the two target outlets. Specifically, the two inlets have inflow velocities (z, 0, 0) and (0, 2z, 0) and the two outlets have outflow velocities (1.5z, 0, 0) and (0, 1.5z, 0), respectively. The volume fraction limit is 0.3 for Amplifier and 0.4 for Mixer. We visualize the optimized designs in Fig. 4.

**Tree Diffuser.** The goal of this example is to generate a fluidic diffuser that directs fluids from one constant circular-shaped inlet to 16 small square-shaped outlets while bypassing a small obstacle at the center of the domain, which we enforce as zero-velocity Dirichlet constraints. In the optimized design (Fig. 5), an interesting tree-like topology automatically emerged from our pipeline, where the fluid first branches into four chambers and then into the 16 outlets. The resulting shape produced by our pipeline exhibits an intuitive design. The branching is gradual as one moves from the inlet to the interior of the domain and the branching factor increases gradually in this direction. The cross-sections in Fig. 5 highlight the progressive branching from a single inlet to multiple outlets. Top right: The fluid flow, simulated from the optimized design, is visualized as streamlines.

**Fluid Twister.** In this example, we enforce a circular-shaped constant inlet with inflow velocity ($v_{\text{in}}$, 0, 0). The objective of the task is to generate a swirl flow in the yz-plane at the outlet of the domain. This example is solved on a 100 $\times$ 100 $\times$ 100 grid with nearly 4 million decision variables, and the final design is shown in Fig. 1. We show the streamline visualization of the optimized design in the

---

**Table 2.** We report the statistics from optimizing the design problems in Sec. 7.1, including the maximum fluidic volume fraction, the functional loss before and after optimization, and the time decomposition in each optimization iteration: “Forward” represents the time spent on computing the loss function, which is dominated by the numerical simulator; “Backprop.” stands for the time spent on computing the gradients; “MMA optimizer” reports the time cost from running one iteration of MMA after obtaining the loss and gradients; “Total” is the sum of all the time above.
middle of 1, which successfully generates a swirl flow at the outlet.
Using our topology optimization approach, a propeller-like structure automatically emerged from a constant fluidity parameter field, highlighting the ability of our pipeline to create a new topology.

\[ V_{\text{max}} = 0.25 \]
\[ V_{\text{max}} = 0.5 \]

Fig. 6. We design the Circuit with maximum volume fraction \( V_{\text{max}} = 0.25 \) (top) and \( V_{\text{max}} = 0.5 \) (bottom). Inset: We visualize the domain setup. Different faces of the domain are marked as inlet and outlet with different flow velocity. Left: The anisotropic boundary of the generated design is visualized using small colored discs. Right: The resulting fluid flow from the optimized design is visualized using color-coded streamlines.

Fluid Circuit. In this example, we mimic a fluidic circuit that connects multiple inlets, located at two faces of the cubic design, to multiple outlets located at the remaining four faces of the cubic design. The inlets have three types of inlet velocities, and the goal of the circuit is to connect the inlets to produce equal flows at the outlets. The result of our optimization is shown in Fig. 6. The optimized design that emerges from our pipeline is intuitive, as it seems to connect the nearest pairs of inlets and outlets that can produce equal flows in order to meet the small volume fraction constraint (0.25 in this example). For instance, the top-right inlet on the left face of the domain is connected to the nearest outlet on the top face of the domain. We present two results using different volume constraints \( V_{\text{max}} = 0.25 \) and \( V_{\text{max}} = 0.5 \) and observe different topological structures, which exhibit different routing plans between the inlets and outlets.

7.2 Evaluation

Below we show experiments evaluating our method and comparing our optimization methods to previous works [Borrvall and Petersson 2003; Du et al. 2020]. We include an additional evaluation of the sensitivity of our results to initialization, an experiment validating our optimized designs, and an ablation study on our new anisotropic material model in the supplementary material.

Solver evaluation. To verify that our simulation results converge under refinement, we evaluate our solver on a 2D fluid amplifier design represented by two symmetric Bézier curves. The left of the domain has horizontal inflow. We simulate the design in square domains of dimensions 32, 64, 128, 256, 512, 1024, and 2048, and observe that the velocity fields converge to a limit (Fig. 7). To compare our solver with more traditional Stokes flow solvers, we simulate the same amplifier design using the solver from Du et al. [2020], which is an exact interface solver that supports free-slip boundaries. As shown in Fig. 7, the main body of the velocity fields are similar, and the discrepancy mainly exhibits near the solid boundary due to the different boundary treatments. Specifically, Du et al. [2020] assumes an exact interface for the solid-fluid interface and simulates the cells of the interface with subcell precision, while our method only assumes the direction of the interface within the cell.

Sensitivity of results to block size. To evaluate the sensitivity of the optimization results to block size, we optimize the 2D amplifier design problem under 100 x 100 with volume fraction limit 0.5. The domain has an inlet with parallel horizontal inflow located at the left of the domain and an outlet at the right of the domain. We initialize the optimization with \( \epsilon = 1 \) and \( \rho = 0.5 \) and repeat the optimization with block sizes of 4, 8 and 16 (Fig. 9). We observe that the optimized designs are similar.

Comparison with baselines. We compare our method with two representative baseline algorithms in the design and optimization of fluidic devices: One is a field-based fluidic topology optimization method described by Borrvall and Petersson [2003], which uses isotropic materials with varying stiffness to model porous materials that allow or impede water passing. The other is a parametric-shape optimization algorithm [Du et al. 2020], which uses parametric shapes to represent the device boundary and optimize the design by evolving shape parameters. We implemented the method of Borrvall and Petersson [2003] exactly as described in the paper with the same set of hyperparameters and verified that we can obtain the results.
we run both methods to solve this design problem with varying volume fraction limits from 0.6 to 0.2. For both methods, we visualize the optimized fluidity field (row 1, 2) and velocity field (row 4, 5). We additionally visualize the optimized isotropy field $\epsilon$ for our method (row 3). As we decrease the volume fraction from left to right, the method of Borrvall and Petersson [2003] starts to synthesize non-physical designs and flow, i.e., the solid cells near the inlet and outlet as well as nonzero fluid velocity on them.

Fig. 8. Comparisons between our method and [Borrvall and Petersson 2003] using a 2D design problem ($60 \times 60$ cells). The domain has one inlet and outlet at its left and right, respectively. The goal is to synthesize a structure that transports the inlet flow with velocity $(\nu_i, 0)$ to the outlet. From left to right, we run both methods to solve this design problem with varying volume fraction from 0.6 to 0.2. For both methods, we visualize the optimized fluidity field (row 1, 2) and velocity field (row 4, 5). We additionally visualize the optimized isotropy field $\epsilon$ for our method (row 3). As we decrease the volume fraction from left to right, the method of Borrvall and Petersson [2003] starts to synthesize non-physical designs and flow, i.e., the solid cells near the inlet and outlet as well as nonzero fluid velocity on them.

The main difference between our approach and [Borrvall and Petersson 2003] is the introduction of an anisotropic material model in fluidic topology optimization problems. To validate the merits of anisotropic materials, we consider a 2D design problem that aims to synthesize the internal structure between an inlet and an outlet on the opposite sides of a square domain ($60 \times 60$ cells). The inlet flow is given by $(\nu_i, 0)$ and enforced as the Dirichlet boundaries, and the goal is to create a design so that the outlet flow at each node is $(\nu_o, 0)$ too. While this design problem has a trivial solution of a straight pipe connecting the inlet and the outlet, we stress test the problem by imposing various volume fraction $V_{\text{max}}$ ranging from 60% to 20% of the domain (Fig. 8, left to right) and run both methods. As we decrease $V_{\text{max}}$, the method of Borrvall and Petersson [2003] starts to generate physically implausible flow velocities that co-exist with several solid cells near the inlet and the outlet (Fig. 8 top). The deeper reason behind them is that traditional isotropic field-based methods typically lead to blurred solid-fluid boundaries that occupy more fluid volumes than a physical boundary should. In exchange for that, such methods have to trade fluid volumes that should have stayed in the interior of the fluid phase when the volume fraction becomes tight. In contrast, our method maintains the sharp boundary of the final design and the physically plausible flow velocity even if we push $V_{\text{max}}$ to its minimum (Fig. 8 bottom), which we attribute to the anisotropic material model.

Fig. 9. Sensitivity of optimization results to block size. We optimize the 2D amplifier design problem with block sizes 4, 8, and 16 where initial $L_F = 141.938$. We visualize the norm of the velocity field (top) and the optimized isotropy field (bottom) and report the optimized $L_F$ value (bottom).

Fig. 10. Solving the Fluidic Twister example using the open-source implementation from Du et al. [2020]. Left: The parametric shape and fluid flow visualization before optimization, with the top-right insets showing the designs viewed from their outlets. Right: The corresponding visualizations after optimization.

The difference between our method and [Du et al. 2020] is that our representation of a fluidic system is field-based while their representation is based on parametric shapes. We run both methods in the Fluidic Twister example, which was also one of the design problems studied in their paper. We report the optimization results in Figs. 1 and 10, respectively. It is evident that the design space in [Du et al. 2020] is limited to parametric shapes that only evolve a simple surface (Fig. 10) without changing its topology. On the contrary, our method automatically synthesizes a propeller-like structure without geometrical or topological priors. Furthermore, our final design achieves a much lower functional loss (0.52) than their approach.
We identify certain current limitations of our approach and discuss possible future directions that could address and overcome them.

First, our model has consciously engaged in certain modeling simplifications of the governing physics of a fluidic system. For example, our approach models the fluid phase as steady-state Stokes flow and ignores — as has been the case with almost any prior work that compares to our feature set — the effects of time-dependent flow behaviors on the system’s performance. Developing optimization frameworks for dynamic fluid systems becomes a natural next step based on our Stokes flow optimizer. On the other hand, our method models the solid phase as rigid. Extending the current model to a compliant solid phase to develop optimization tools for the interaction between incompressible flow and compliant structures rises as an important problem to solve for fluid-driven soft robot design.

Second, our model is currently limited by its scalability. Even though our demonstrated resolutions are quite competitive in the context of fluidic topology optimization, it lacks by orders of magnitude the resolution complexity of topology optimization pipelines that focus on purely solid/elastic (not fluidic) devices. Our current framework employs a direct solver for the algebraic problems/systems arising from our discretization. To achieve a significant next leap in scalability, we will need to devise iterative and possibly multi-resolution (e.g., multigrid, or multigrid-preconditioned) solvers. Although there is precedent for scalable multigrid solvers performing well for Stokes flows [Gaspar et al. 2008], there is a number of complications related to our specific needs that will need to be addressed. For example, most prior Stokes multigrid solvers rely on staggered discretizations and mixed variational formulations for proper performance. We would likely need to adopt a mixed formulation as well [Patterson et al. 2012] but preserving the convergence qualities that are established in staggered discretizations in the contest of a collocated discretization as the one we employ will require attention to stability issues and careful adaptation of relaxation techniques. At the same time, we will need to address the complications that our anisotropic terms might impart on the convergence of techniques for pure Stokes problems.

In addition to addressing the issues of model accuracy and solver scalability, we anticipate to devise new optimization objectives and constraints to consider the system’s manufacturability in the framework. The methods for fabrication and performance benchmark to evaluate optimized fluidic devices also remain as an unexplored yet essential field to bridge the gap between simulation and fabrication.

ACKNOWLEDGMENTS

Yifei Li acknowledges the emotional support from Yihui Li. Wojciech Matusik acknowledges the funding support from NSF IIS-2106962 and the Defense Advanced Research Projects Agency (DARPA) under grant No. FA8750-20-C-0075. Bo Zhu acknowledges the funding supports from NSF IIS-2106733. Efthychios Sifakis acknowledges the funding supports from NSF IIS-2106768, IIS-2008584, IIS-1763638.

REFERENCES

Niels Aage, Thomas H. Poulsen, Allan Gersborg-Hansen, and Ole Sigmund. 2008. Topology optimization of large scale stokes flow problems. Structural and Multidisciplinary Optimization 35, 2 (2008), 175–180.

Christine Alapatt, Achim Basermann, Alan R. Bishop, Holger Fehske, Georg Hager, Olaf Schenk, Jonas Thies, and Gerhard Wellein. 2020. A Recursive Algebraic Coloring Technique for Hardware-Efficient Symmetric Sparse Matrix-Vector Multiplication. ACM Trans. Parallel Comput. 7, 3, Article 19 (June 2020), 37 pages. https://doi.org/10.1145/3399732
Supplementary Materials

A ADDITIONAL EVALUATION

Sensitivity of results to initialization. To evaluate the sensitivity of optimization results to initialization, we optimize the same 2D amplifier design problem with different initial values. Specifically, we perturb the initial fluidity field $\rho$ and anisotropic orientation field $\alpha$ by a noise sampled from $\mathcal{U}(-k,k)$ where $k$ is 0.001, 0.01, and 0.09, respectively (Fig. 12).

![Fig. 12. Sensitivity of optimization results to initial values. We optimize the 2D amplifier design problem with initial values perturbed. Specifically, the fluidity field $\rho$ and anisotropic orientation field $\alpha$ by a noise vector sampled from $\mathcal{U}(-k,k)$ where $k$ is 0.001, 0.01 and 0.09 (left, middle, right), respectively.](image)

Validation of optimized result. To validate the design optimized by our anisotropic mixture model, we extract an exact interface from our optimized design of a 2D amplifier design problem under 100 × 100 and simulate the design using our solver and the conventional FEM solver from [Du et al. 2020] (Fig. 13). To extract the optimized interface, we first binarize the optimized $\epsilon$ field, then fit four cubic Bézier curves to the binarized field. We compare the simulation of the extracted design using the solver from our method and from [Du et al. 2020]. The initial, optimized, and the $L_f$ value achieved with the reparameterized and binarized design (for validation purposes) is 142.70 and 6.42 and 26.51 respectively. The increased $L_f$ value after the two post-processing steps is as expected because to the boundary geometry change introduced during binarization and fitting, and different approaches for handling the boundary between the two methods, which introduce simulation difference near the boundary of the domain.

Anisotropic material model. We study the effects of the two critical anisotropic material parameters $K_m$ and $K_f$ on modeling sharp, free-slip solid-fluid boundary conditions. Specifically, we analyze the Stokes flow problem in a 2D slanted pipe (Fig. 14) with straight and parallel free-slip boundaries and a constant inlet flow parallel to them. Since these boundaries are frictionless, we expect a physically plausible simulator to generate a constant flow inside the pipeline, and in particular, result in a constant outlet flow velocity parallel to the boundaries. We consider using four possible combinations of isotropic/anisotropic $K_m$ and $K_f$ to model and simulate this slanted pipe example discretized on a grid of 20 × 20 cells and report their resultant flow fields in Fig. 14. Comparing these results, we can see that the simulation generates a near-constant flow field only when $K_m$ and $K_f$ are both anisotropic (Fig. 14 bottom right) using the values suggested in Table 1. In particular, whenever $K_m$ or $K_f$ is isotropic, it significantly damps the flow near the solid-fluid boundaries, leading to a non-constant outlet flow profile. This comparison shows that isotropic material models suffer from the inherent difficulty in modeling free-slip boundary conditions and necessitates the need for anisotropic $K_m$ and $K_f$.

B OPTIMIZATION PROBLEMS ILLUSTRATION

We illustrate the design specifications for all of the examples from Sec. 7.1 in Fig. 15 and Fig. 16. In each example, the nodes on the faces not labeled with “outlet” have zero-velocity Dirichlet conditions specified on the non-fluid domain. On faces labeled with “outlet”, the non-fluid nodes have zero velocity Dirichlet conditions specified when computing $L_e$ but the conditions are released when computing $L_f$, as explained in Sec 6.
Fig. 14. We study the effects of anisotropic material parameters $K_m$ and $K_f$ on simulating our Stokes flow model in a 2D slanted pipe ($20 \times 20$ cells). We repeat the simulation with all four combinations of anisotropic ($K_m, K_f$) and their isotropic counterparts ($K_m = I$ and $K_f = k_f I$). The slanted straight pipe has parallel free-slip solid-fluid boundaries (dashed blue lines) and a constant inlet flow parallel to them from left of the domain. We visualize the flow velocity at each node in the pipe as blue arrows and expect the velocity at each outlet node to be identical to the inlet velocity. We additionally highlight the flow norm profile passing through a cross-section in the insets.

Fig. 15. We visualize the domain specification for the Amplifier, Mixer, Tree Diffuser and Twister design problems.

Fig. 16. We visualize the domain specification for the Circuit design problem.