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

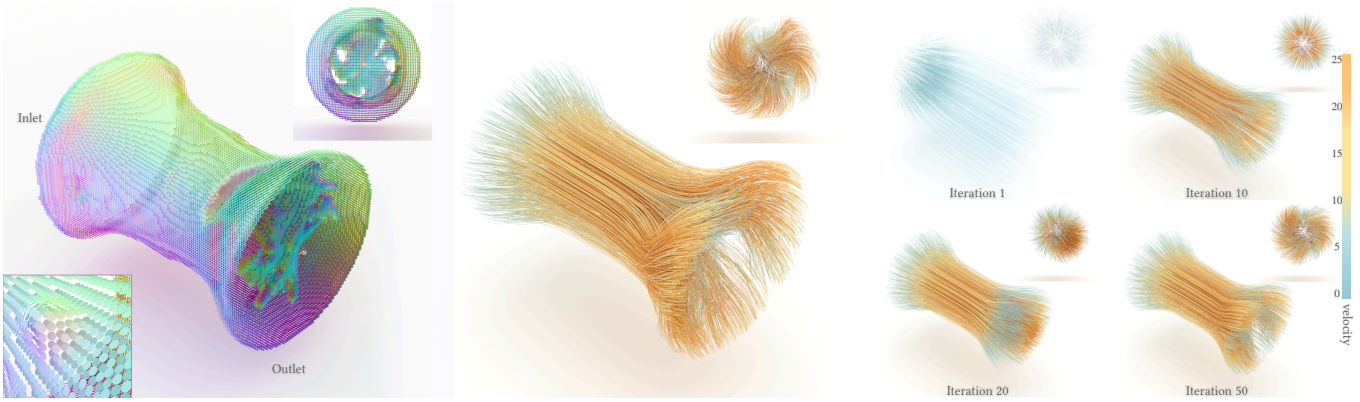

Fig. 1. We present a topology optimization pipeline for designing Stokes-flow fluidic systems with flexible and accurate boundary conditions. Our method automatically creates the structure of this fluidic twister on an $100 \times 100 \times 100$ grid after optimizing nearly four million decision variables. 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

Authors' addresses: Yifei Li, MIT CSAIL, Cambridge, 02139, MA, USA, liyifei@csail.mit.edu; Tao Du, MIT CSAIL, Cambridge, 02139, MA, USA, taodu@csail.mit.edu; Sangeetha Grama Srinivasan, University of Wisconsin-Madison, Madison, 53706, WI, USA, sgsrinivasa2@wisc.edu; Kui Wu, LightSpeed Studios, Tencent, USA, Los Angeles, CA, kwwu@tencent.com; Bo Zhu, Dartmouth College, Hanover, 03755, NH, USA, sifakis@cs.wisc.edu; Eftychios Sifakis, University of Wisconsin-Madison, Madison, 53706, WI, USA, sifakis@cs.wisc.edu; Wojciech Matusik, MIT CSAIL, Cambridge, MA, USA, wojciech@csail.mit.edu.

0730-0301/2022/12-ART239
https://doi.org/10.1145/3550454.3555429

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 anisotropicity (with local flow directions) in a unified geometric representation. Our method was inspired by the anisotropic material model for elastic simulation [Li and Barbič 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 Andreasen 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 2008], 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) [Andreasen and Sigmund 2013; Casas and Pavanello 2017; Yoon 2010], fluid-thermal interaction [Matsumori et al. 2013; Yaji et al. 2015], microfluidics

[Andreasen et al. 2009], and aerodynamics [Jameson 1995; Maute and Allen 2004], to name a few. The development of topology optimization algorithms to explore functional flow systems remains largely unexplored due to the complexities regarding both the simulation and optimization. In computer graphics, Du et al. [2020] developed a differentiable framework to simulate and optimize Stokes flow systems governed by design specifications with different types of boundary conditions. Although this work realized a multitude of design examples, such as the flow averager, gates, mixer, and twister, the generated designs were all limited within the design space spanned by the predefined shape parameters.

Topology optimization. Topology optimization has demonstrated its efficacy in creating mechanical designs with complex structures and extreme properties in many engineering problems (see [Deaton and Grandhi 2014; Rozvany 2009; Sigmund and Maute 2013] for surveys). Starting from a volumetric domain with uniform material distribution, a topology optimization algorithm iteratively redistributes material to develop a structure that minimizes a design objective (e.g., structural compliance), given the prescribed target volume and boundary conditions. In computer graphics, a wide range of topology optimization algorithms have been developed to accommodate computational fabrication applications and 3D printing designs, including examples of elastic structures [Liu et al. 2018], shells [Skouras et al. 2014], porous materials [Wu et al. 2018], and microstructures [Panetta et al. 2015; Schumacher et al. 2015; Zhu et al. 2017]. Despite their successes in structural optimization, research on topology optimization algorithms for solving flow systems with accurate boundary conditions is scarce, limiting their applications in designing thin and delicate structures in fluidic devices. Finally, we share inspiration from various works from the graphics community on differentiable simulation [Du et al. 2021; Hu et al. 2018; Li et al. 2022] and shape optimization [Bächer et al. 2017; Wang and Whiting 2016] for computational design and control applications.

Anisotropic methods. Anisotropic methods have been explored in all aspects of computer graphics. Examples include meshing [Narain et al. 2012], texturing [McCormack et al. 1999], rendering [Wang et al. 2008], surface reconstruction [Yu and Turk 2013], and various physics simulations such as cloth [Narain et al. 2012], solid [Li and Barbič 2015; Schreck and Wojtan 2020], and fluid [Pfaff et al. 2010; Xiao et al. 2020]. Typically, an anisotropic method encode the local orientation information either in discrete spatial discretizations, such as anisotropic mesh elements or grid cells, or through continuous tensor representations, such as anisotropic elastic material models or fluid turbulent models. In this paper, we chose to explore anisotropic tensor representations discretized on a uniform grid, which mimics the aniostropic continuum mechanics models and uses it in a new context for representing solid-fluid boundaries in topology optimization.

3 METHOD OVERVIEW

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 $\Omega \subset \mathcal{R}^d$ ($d = 2$ or 3). The velocity field $\boldsymbol{v} : \Omega \rightarrow \mathcal{R}^d$ of the quasi-incompressible Stokes flow is given by the following energy minimization problem:

$$\min_{\boldsymbol{v}} \quad \int_\Omega \mu \|\nabla \boldsymbol{v}\|_F^2 d\boldsymbol{x} + \int_\Omega \lambda (\nabla \cdot \boldsymbol{v})^2 d\boldsymbol{x}, \tag{1}$$

$$s.t. \quad \boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v}_D(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial\Omega_D. \tag{2}$$

$$\boldsymbol{v}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in \partial\Omega_F. \tag{3}$$

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 $\|\cdot\|_F$ is the Frobenius norm of a matrix, and $\mu \in \mathcal{R}^+$ and $\lambda \in \mathcal{R}^+$ 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 $\partial\Omega = \partial\Omega_D \cup \partial\Omega_F \cup \partial\Omega_O$: The *Dirichlet boundary* condition specifies a desired velocity profile $\boldsymbol{v}_D$ on the boundary $\partial\Omega_D$, which is either from prescribed inlet

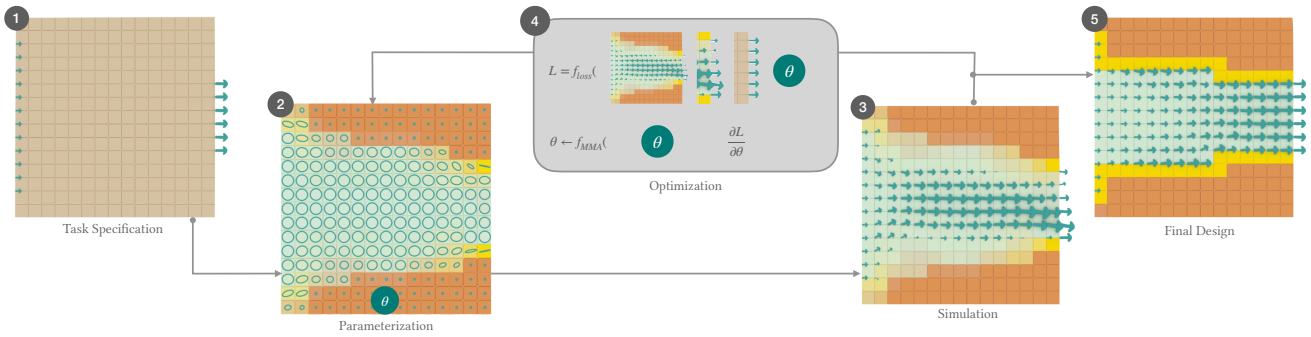

Fig. 2. An overview of our pipeline: (1) Our system takes as input the specification of a fluidic device design task on a regular grid, including the locations and desired profiles of the Stokes flow at the inlet and outlet; (2) The system then represents the fluidic device using anisotropic materials in each voxel, which we further parametrize using its anisotropic direction (the principal axes of the ellipse in the illustration), viscosity (radius of the ellipse), and impedance for fluid (shown as the background color of the voxel); (3) A numerical differentiable simulator receives the design and solves the Stokes flow; (4) The pipeline then compares the simulated flow at the outlet with the target profile given in the specification and computes a loss function characterizing their discrepancy. The loss is then backpropagated through the numerical simulator to compute its gradients with respect to the anisotropic material parameters. The pipeline runs the method of moving asymptotes (MMA, [Svanberg 1987]), a gradient-based optimizer, to improve the design; (5) The pipeline outputs a final design after post-processing the results from a converged optimization process.

flow profiles or from *no-slip boundary* conditions ($\boldsymbol{v}_D = \mathbf{0}$); the *free-slip boundary* condition defined on $\partial\Omega_F$ requires the velocity's projection along the normal direction $\boldsymbol{n}$ be zero; finally, the *open boundary* on $\partial\Omega_O$ 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:

$$\min_{\boldsymbol{v}} \quad E_{m,\mu}[\boldsymbol{v}] + E_{m,\lambda}[\boldsymbol{v}] + E_f[\boldsymbol{v}], \tag{4}$$

$$s.t. \quad \boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v}_D(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial\mathcal{B}_D. \tag{5}$$

where each energy component is defined as follows:

$$E_{m,\mu}[\boldsymbol{v}] := \int_{\mathcal{B}} \mu \|\nabla \boldsymbol{v} \boldsymbol{K}_m^{\frac{1}{2}}(\boldsymbol{x})\|_F^2 d\boldsymbol{x}, \tag{6}$$

$$E_{m,\lambda}[\boldsymbol{v}] := \int_{\mathcal{B}} \lambda(\boldsymbol{x})(\nabla \cdot \boldsymbol{v})^2 d\boldsymbol{x}, \tag{7}$$

$$E_f := \int_{\mathcal{B}} \|\boldsymbol{K}_f^{\frac{1}{2}}(\boldsymbol{x})\boldsymbol{v}\|_2^2 d\boldsymbol{x}, \tag{8}$$

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 $\boldsymbol{K}_m = \mathbf{I}$ and $\boldsymbol{K}_f = \mathbf{0}$. 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 Ω to $\mathcal{B} \subset \mathcal{R}^d$, which we assume to be an axis-aligned, sufficiently large box that encloses the fluidic region Ω. Second, we introduce two symmetric positive semi-definite matrix fields $\boldsymbol{K}_m, \boldsymbol{K}_f : \mathcal{B} \to \mathcal{S}_+^d$ and replace the scalar parameter λ with a spatially varying field $\lambda : \mathcal{B} \to \mathcal{R}^+$. These three new fields define a new material model that enables anisotropic responses to velocities at different directions. Finally, with the domain changing from Ω to a box-shaped $\mathcal{B}$, we adjust the boundary conditions as follows: We consider a partition of the boundary $\partial\mathcal{B}$ into $\partial\mathcal{B} = \partial\mathcal{B}_D \cup \partial\mathcal{B}_O$ where $\partial\mathcal{B}_D$ and $\partial\mathcal{B}_O$ represent the locations of the Dirichlet boundary and the open boundary conditions, respectively. The Dirichlet boundary $\mathcal{B}_D$ 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 $\partial\mathcal{B}_O$ 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 $\mathcal{B}$ and will be represented by a careful choice of $\boldsymbol{K}_m$, $\boldsymbol{K}_f$, and λ. We illustrate the new Stokes flow model in Fig. 3.

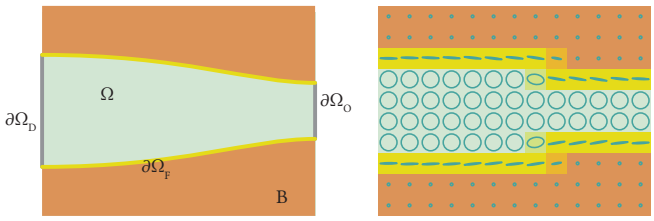

Fig. 3. An illustration of the anisotropic material model in the quasi-incompressible Stokes flow. We selected three representative material points from the solid phase (orange), the fluid phase (green), and the free-slip solid-fluid boundary (yellow). We associate each material point with an ellipse visualization of $\boldsymbol{K}_f$ in a way that the material inhibits flow along its minor principal axis (small radii). For example, a solid material point forces flow along all directions to be zero and is therefore associated with a small circle. Similarly, a material point on the free-slip boundary impedes flow only in the normal direction, so its $\boldsymbol{K}_f$ is a highly eccentric ellipse aligned with the tangent and normal direction of the boundary.

The new energy minimization problem uses a single material model characterized by spatially-varying $\boldsymbol{K}_m$, $\boldsymbol{K}_f$, and λ in the new problem domain $\mathcal{B}$. We stress that this material model design is not arbitrary but inspired by strong physics intuition. Below, we will discuss the subtleties in these three parameters by demonstrating their capability of expressing different phases (solid and fluid) and various boundary types (no-slip and free-slip). Concretely speaking, we will show that by properly setting them everywhere in $\mathcal{B}$, we can draw an analogy between the two physical models described in Eqns. (1-3) and Eqns. (4-5).

Fluid-phase material. For any point $\boldsymbol{x}$ in the interior of the fluid phase in Eqns. (1-3), i.e., $\boldsymbol{x}$ belongs to the interior of Ω, we choose $\boldsymbol{K}_m = \boldsymbol{I}$, $\boldsymbol{K}_f = \boldsymbol{0}$, and $\lambda = \lambda_0$ where λ_0 indicates the scalar parameter used in the original quasi-incompressible Stokes problem (Eqn. (1)). This way, the energy in Eqns. (6-7) becomes identical to Eqn. (1), confirming that it preserves the physics model of quasi-incompressible Stokes flow in the fluid phase.

Solid-phase material. Similarly, for any point $\boldsymbol{x}$ in the interior of the solid phase, we set $\boldsymbol{K}_f = k_f \boldsymbol{I}$ where $k_f \to +\infty$. According to the energy in Eqn. (8), this will force the velocity $\boldsymbol{v}$ at $\boldsymbol{x}$ to be $\boldsymbol{0}$ just as expected. As a result, $\boldsymbol{v}$ will be an all-zero field in the interior of the solid phase, leading to $E_{m,\mu} = E_{m,\lambda} = 0$ regardless of the choice of $\boldsymbol{K}_m$ and λ. We suggest $\boldsymbol{K}_m = \boldsymbol{I}$ and $\lambda = \lambda_0$ in this case, modeling the solid phase as an isotropic, quasi-incompressible material that impedes fluid.

No-slip-boundary material. It remains to show how a proper combination of $\boldsymbol{K}_m$, $\boldsymbol{K}_f$, and λ can represent solid-fluid boundary conditions inside the domain $\mathcal{B}$. We consider two types of solid-fluid boundaries in this work: no-slip and free-slip. Note that a no-slip solid-fluid boundary simply states the flow velocity near the boundary should be zero, so we can treat it the same way as we define the solid material above, i.e., $\boldsymbol{K}_f = k_f \boldsymbol{I}$ with $k_f \to +\infty$, $\boldsymbol{K}_m = \boldsymbol{I}$, and $\lambda = \lambda_0$.

Free-slip-boundary material. The last and most challenging case is to model free-slip boundary conditions (Eqn. (3)) with a proper choice of $(\boldsymbol{K}_m, \boldsymbol{K}_f, \lambda)$. For brevity, we will present our results in 3D only. Consider a point $\boldsymbol{x}$ on a free-slip solid-fluid boundary, and let $\boldsymbol{n}$ be its unit normal vector. We augment $\boldsymbol{n}$ with two unit vectors $\boldsymbol{t}_1, \boldsymbol{t}_2$ orthogonal to $\boldsymbol{n}$ so that the matrix $\boldsymbol{R} := (\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2)$ defines an orthonormal basis in $\mathcal{R}^3$.

To derive a proper $\boldsymbol{K}_m$ for free-slip boundaries, we recall that the isotropic energy $\|\nabla \boldsymbol{v}\|_F^2$ in the original Stokes flow model is the variational form of $\Delta \boldsymbol{v}$, 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 are rotationally invariant, allowing us to change the coordinate systems freely when computing $\nabla \boldsymbol{v}$. Therefore, we can consider computing $\nabla \boldsymbol{v}$ in a local frame spanned by the normal and tangent directions at $\boldsymbol{x}$, i.e., the columns of $\boldsymbol{R}$:

$$\boldsymbol{v}_{\boldsymbol{R}} := \boldsymbol{R}^\top \boldsymbol{v}, \tag{9}$$

$$\boldsymbol{x}_{\boldsymbol{R}} := \boldsymbol{R}^\top \boldsymbol{x}, \tag{10}$$

$$\nabla_{\boldsymbol{x}_{\boldsymbol{R}}} \boldsymbol{v}_{\boldsymbol{R}} = \boldsymbol{R}^\top \nabla \boldsymbol{v} \boldsymbol{R}. \tag{11}$$

Here, the subscript $(\cdot)_{\boldsymbol{R}}$ means the quantity is defined in the local frame spanned by $\boldsymbol{R}$. For example, the first row in $\nabla_{\boldsymbol{x}_{\boldsymbol{R}}} \boldsymbol{v}_{\boldsymbol{R}}$ 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 $\boldsymbol{K}_m$: 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 $\nabla_{\boldsymbol{x}_{\boldsymbol{R}}} \boldsymbol{v}_{\boldsymbol{R}}$ that corresponds to the normal direction, leading to the following energy density to be integrated in $E_{m,\mu}$:

$$\Psi_{m,\mu} := \mu \|\nabla_{\boldsymbol{x}_{\boldsymbol{R}}} \boldsymbol{v}_{\boldsymbol{R}} \boldsymbol{\Lambda}(0,1,1)\|_F^2 \tag{12}$$

$$= \mu \|\boldsymbol{R}^\top (\nabla \boldsymbol{v}) \boldsymbol{R} \boldsymbol{\Lambda}(0,1,1)\|_F^2 \tag{13}$$

$$= \mu \|(\nabla \boldsymbol{v})(\boldsymbol{0}, \boldsymbol{t}_1, \boldsymbol{t}_2)\|_F^2 \tag{14}$$

$$= \mu \mathrm{trace}((\nabla \boldsymbol{v})(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^\top)(\nabla \boldsymbol{v})^\top) \tag{15}$$

$$= \mu \|\nabla \boldsymbol{v} (\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^\top)^{\frac{1}{2}}\|_F^2 \tag{16}$$

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

$$\boldsymbol{K}_m = \boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^\top. \tag{17}$$

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

$$\boldsymbol{K}_f = \boldsymbol{R} \boldsymbol{\Lambda}(k_f, 0, 0) \boldsymbol{R}^\top = k_f \boldsymbol{n}\boldsymbol{n}^\top, \tag{18}$$

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

Finally, regarding the choice of λ, we set its value to 0 so that $E_{m,\lambda}$ is ignored for points on free-slip boundaries. We justify this decision from two perspectives: First, using a positive λ 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 λ 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 $\boldsymbol{K}_m$, $\boldsymbol{K}_f$, and λ for all cases in Table 1.

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

	Fluid	Solid	No-slip boundary	Free-slip boundary
$\boldsymbol{K}_m$	$\boldsymbol{I}$	$\boldsymbol{I}$	$\boldsymbol{I}$	$\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^\top$
$\boldsymbol{K}_f$	$\boldsymbol{0}$	$k_f\boldsymbol{I}$	$k_f\boldsymbol{I}$	$k_f\boldsymbol{n}\boldsymbol{n}^\top$
λ	λ_0	λ_0	λ_0	0

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 $\boldsymbol{v}$ are stored at grid nodes and bilinearly (2D) or trilinearly (3D) interpolated, while the design parameters $\boldsymbol{K}_m$, $\boldsymbol{K}_f$ and λ 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 $\boldsymbol{K}_m$ and $\boldsymbol{K}_f$ and a scalar field λ 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 $\boldsymbol{K}_m$, $\boldsymbol{K}_f$, and λ 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 ϵ means more isotropic material; the anisotropic orientation $\boldsymbol{\alpha} : \mathcal{B} \to \mathcal{R}^{d-1}$, which is a field of rotational angles in 2D and spherical coordinates in 3D. Furthermore, we compute a unit normal field $\boldsymbol{n} : \mathcal{B} \to \mathcal{R}^d$ induced from $\boldsymbol{\alpha}$.

Constructing $\boldsymbol{K}_m$. We define the material matrix field $\boldsymbol{K}_m$ as follows:

$$\boldsymbol{K}_m = \boldsymbol{I} - (1 - \epsilon)\rho\boldsymbol{n}\boldsymbol{n}^\top. \tag{19}$$

Therefore, $\boldsymbol{K}_m \approx \boldsymbol{I}$ whenever $\epsilon \approx 1$ (isotropic material) or $\rho \approx 0$ (solid phase), and $\boldsymbol{K}_m$ 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 $\boldsymbol{K}_f$. We define k_f with the following nonlinear mapping function borrowed from previous work [Borrvall and Petersson 2003]:

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

where $k_{f_{\min}} \approx 0$ 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 ρ (solid phase) to $k_{f_{\max}}$ and large ρ (fluid phase) to $k_{f_{\min}}$. We then define $\boldsymbol{K}_f$ as follows:

$$\boldsymbol{K}_f = k_f(\rho)\boldsymbol{I} + (k_f(\epsilon\rho) - k_f(\rho))\boldsymbol{n}\boldsymbol{n}^\top. \tag{21}$$

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

Constructing λ. Finally, we construct the λ field as follows:

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

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

and $\lambda_{\min} + \lambda_{\max}$. Similar to $\boldsymbol{K}_m$ defined above, this definition of λ requires that $\lambda \approx \lambda_{\min}$ only if $\epsilon \approx 0$ and $\rho \approx 1$, i.e., near free-slip boundaries.

Summary. In this manner, the anisotropic material distribution in the entire domain is parameterized by three fields ρ, ϵ, and $\boldsymbol{\alpha}$. 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 ρ, ϵ, and $\boldsymbol{\alpha}$ on the regular grid by storing their values at each cell center. For brevity, we use $\boldsymbol{\theta}$ from now on to refer to the set of discretized material parameters ρ, ϵ, and $\boldsymbol{\alpha}$:

$$\boldsymbol{\theta} := \{\rho, \epsilon, \boldsymbol{\alpha}\}. \tag{23}$$

The induced fields $\boldsymbol{K}_m$, $\boldsymbol{K}_f$, and λ 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 $\boldsymbol{v}$ 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 $E_{m,\mu}$ in Eqn. (6) – we can label this the *Anisotropic Laplace* term, recognizing that the isotropic case $\boldsymbol{K}_m = \mathbf{I}$ 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 $\boldsymbol{K}_m$ associated with the cell at all quadrature points.

The term $E_{m,\lambda}$ 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:

$$E^C_{m,\lambda} \approx \lambda_C W_C \left[\frac{1}{W_C}\int_C (\nabla \cdot \boldsymbol{v}) dx\right]^2$$

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* $\text{Flux}(C) = \int_C (\nabla \cdot \boldsymbol{v}) dx$ 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 posssible to obtain an analytic expression for the flux, namely in 2D (if $\boldsymbol{v} = (u, v)$ are the individual scalar velocity components):

$$\text{Flux}(C) = h\left(\frac{u_{10}+u_{11}}{2} + \frac{v_{01}+v_{11}}{2} - \frac{u_{00}+u_{01}}{2} - \frac{v_{00}+v_{10}}{2}\right) \tag{24}$$

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 velocies, 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:

$$E^C_f \approx \frac{W_C}{2^d}\sum_I \|\boldsymbol{K}_f^{\frac{1}{2}}(C)\boldsymbol{v}_I\|_2^2$$

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 $E = \boldsymbol{v}^\top \boldsymbol{K}\boldsymbol{v} - \boldsymbol{b}^\top \boldsymbol{v}$, where $\boldsymbol{v}$ stacks all velocity degrees of freedom and $\boldsymbol{K}$ and $\boldsymbol{b}$ 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:

$$\min_{\boldsymbol{v}} \quad \boldsymbol{v}^\top \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{v} - \boldsymbol{b}(\boldsymbol{\theta})^\top \boldsymbol{v} \tag{25}$$

$$s.t. \quad \boldsymbol{v}_i = (\boldsymbol{v}_D)_i, \forall (i, (\boldsymbol{v}_D)_i) \in \mathcal{D}, \tag{26}$$

where $\mathcal{D}$ 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 $E_{m,\lambda}$ penalizing the divergence with a coefficient $\lambda_{\max} \to +\infty$, and free-slip boundaries are replaced with E_f scaled by $(k_f)_{\max} \to +\infty$ 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 λ 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 λ 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 in. 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:

$$0 = \mathrm{Flux}(\mathcal{B}_b) = \sum_{C \in \mathcal{B}_b} \mathrm{Flux}(C)$$

As one would intuitively expect, the net flux over the block $\mathcal{B}_b$ is a linear expression of the averaged (signed) fluxes through the faces of the aggregate block; the contribution of faces interior to $\mathcal{B}_b$ 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 $C\boldsymbol{v} = \mathbf{0}$. Ultimately, we reformulate our governing anisotropic Stokes equations as a linearly-constrained, quadratic optimization problem, where we minimize the functional $E(\boldsymbol{v}) = \boldsymbol{v}^T\mathbf{K}(\boldsymbol{\theta})\boldsymbol{v} - \boldsymbol{b}(\boldsymbol{\theta})^T\boldsymbol{v}$ (resulting from the quadrature schemes in the previous paragraph), subject to the linear constraint $C\boldsymbol{v} = \mathbf{0}$; $\boldsymbol{\theta}$ represents the design parameters. We ultimately solve this constrained optimization problem via the Karush-Kuhn-Tucker condition, in the system:

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

(where $\mathbf{q}$ are the Langrange 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 λ and k_f produced results that were practically indistinguishable from using a very high λ, 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 $\boldsymbol{\theta}$, contrasted to $\mathbf{K}$ and $\boldsymbol{b}$ that do. Ultimately, we view the solution of this constraint optimization problem as the *simulation* function $\boldsymbol{v} = F(\boldsymbol{\theta})$ 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:

$$\min_{\boldsymbol{\theta}} \quad L_f(\boldsymbol{v}) + w_c L_c(\boldsymbol{\theta}) + w_d L_d(\boldsymbol{\theta}) + w_a L_a(\boldsymbol{\theta}), \tag{27}$$

$$s.t. \quad \boldsymbol{v} = F(\boldsymbol{\theta}), \tag{28}$$

$$\boldsymbol{\theta}_{\min} \le \boldsymbol{\theta} \le \boldsymbol{\theta}_{\max}, \tag{29}$$

$$0 \le V_{\text{iso-fluid}}(\boldsymbol{\theta}) \le V_{\max}, \tag{30}$$

$$0 \le V_{\text{all-fluid}}(\boldsymbol{\theta}) \le V_{\text{b}} + V_{\max}. \tag{31}$$

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 $\boldsymbol{\theta}$: Following the standard practice in the previous field-based fluidic topology optimization method [Borrvall 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:

$$L_c(\boldsymbol{\theta}) := \boldsymbol{v}_c^\top \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{v}_c - \boldsymbol{b}(\boldsymbol{\theta})^\top \boldsymbol{v}_c, \tag{32}$$

$$\boldsymbol{v}_c = F(\boldsymbol{\theta}; \mathcal{D} \cup \mathcal{D}_O), \tag{33}$$

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 normals from its neighborhood. We first filter out cells that contain anisotropic materials with two thresholds ϵ_0 and ρ_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:

$$L_d := \sum_{c \in \mathcal{A}} 1 - \psi_{\cos}(\boldsymbol{\alpha}_c, \boldsymbol{\alpha}_{\text{nbr}}) \tag{34}$$

where $\boldsymbol{\alpha}_c$ is the anisotropic direction (a unit vector) of cell c, $\boldsymbol{\alpha}_{\text{nbr}}$ 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:

$$L_a(\epsilon) := \sum_c \epsilon_c \rho_c (\rho^{local}_{\max} - \rho^{local}_{\min}), \tag{35}$$

where the sum loops over each cell c and $\rho^{local}_{\max}$ and $\rho^{local}_{\min}$ 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 $\rho^{local}_{\max} - \rho^{local}_{\min}$) to use anisotropic materials (small ϵ). To avoid chasing a moving target, we freeze $\rho^{local}_{\max}$ and $\rho^{local}_{\min}$ in L_d when optimizing ρ in each iteration.

Constraints. Apart from the objective function, the optimization problem also contains a number of constraints: Eqn. (28) ensures the flow field $\boldsymbol{v}$ 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:

$$V_{\text{iso-fluid}} := \sum_c \epsilon_c \rho_c, \tag{36}$$

$$V_{\text{all-fluid}} := \sum_c \rho_c. \tag{37}$$

The difference between them implies the volume of anisotropic cells, and $V_{\max}$ and V_b 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 $\boldsymbol{\theta}$ 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 exception. 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 topology 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 $\boldsymbol{\theta}$, we extended 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 process, we dynamically update the upper bound on the isotropy ϵ_c for each cell c based on the following heuristic:

$$(\epsilon_c)_{\max} := 1 - (\rho^{local}_{\max} - \rho^{local}_{\min}). \tag{38}$$

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 ϵ). 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 $k_{f_{\min}}$. We chose a near-zero value for $\lambda_{\min}$. The choices of these two hyperparameters followed the convention in topology optimization. We did not observe noticeable differences when their values were perturbed. For $k_{f_{\max}}$ and $\lambda_{\max}$ 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 divergence 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 λ, 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 simulator 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 demonstration 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 $\rho = V_{\max}$. We run all optimizations with 300 iterations and use the design that achieves minimum final

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.

	Resolution	Volume Limit ($V_{\max}$)	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 $(v_{in}, 0, 0)$. The objective is to amplify the flow by a factor of $\frac{5}{3}$. In *Mixer*, the design objective is to mix a high-pressure and low-pressure flow from two inlets to produce equal middle-pressure flows at the two target outlets. Specifically, the two inlets have inflow velocities $(v, 0, 0)$ and $(0, 2v, 0)$ and the two outlets have outflow velocities $(1.5v, 0, 0)$ and $(0, 1.5v, 0)$, respectively. The volume fraction limit is 0.3 for *Amplifier* and 0.4 for *Mixer*. We visualize the optimized designs in Fig. 4.

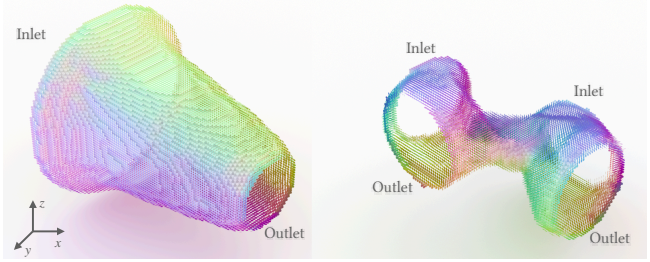

Fig. 4. Designing an *Amplifier* (left) and a *Mixer* (right) under a grid resolution of $80 \times 80 \times 80$. The anisotropic boundaries of the optimized designs are visualized using small colored disks.

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 progress of the branching in the optimized design. This example highlights the ability of our method to synthesize an intricate structure with 16-outlets without any prior on its shape or topology.

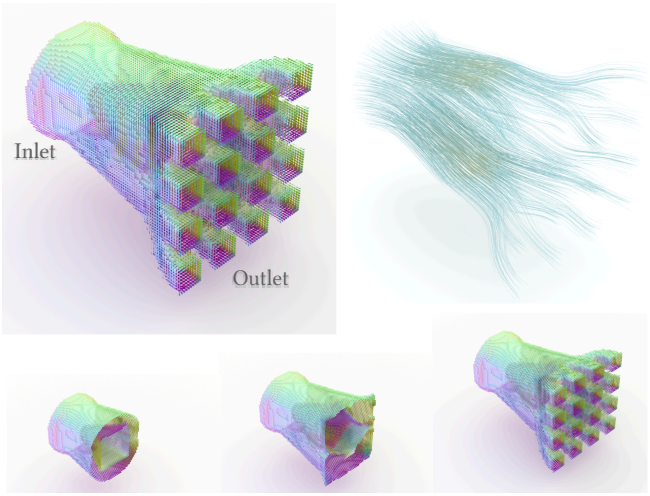

Fig. 5. Our pipeline generates a tree diffuser on a $80 \times 80 \times 80$ grid. Top left: The anisotropic boundary of the optimized design is visualized using small colored disks. Bottom: The 3 images in the middle visualize the perpendicular cross-sections of the optimized design at different depths from the inlet. These cross-sections 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 $(v_{in}, 0, 0)$. 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

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.

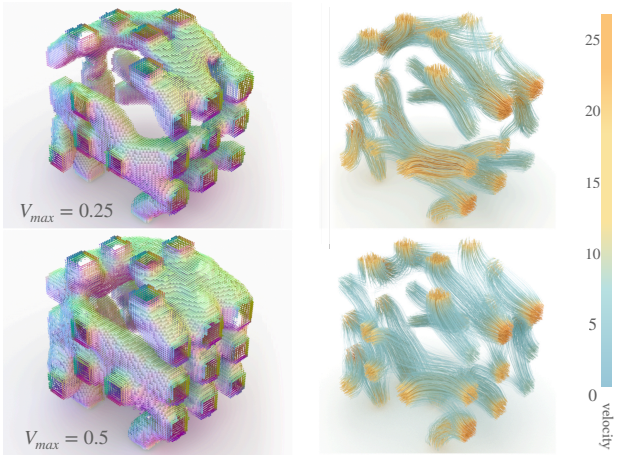

Fig. 6. We design the *Circuit* with maximum volume fraction $V_{\max} = 0.25$ (top) and $V_{\max} = 0.5$ (bottom). Inset: We visualize the domain setup. Different faces of the domain are marked as inlet and outlet with different flow velocity. Left: The anisotropic boundary of the generated design is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

Fluid Circuit. In this example, we mimic a fluid circuit that connects multiple inlets, located at two faces of the cubic domain, to multiple outlets located at the remaining four faces of the cubic domain. 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 $V_{\max} = 0.25$ and $V_{\max} = 0.5$ 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.

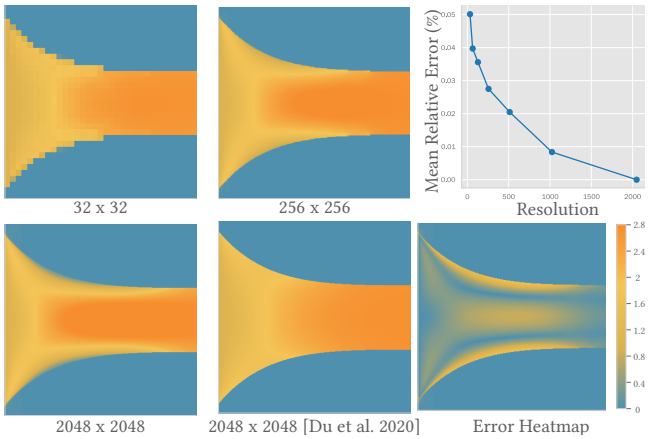

Fig. 7. Evaluation of our solver on a 2D fluid amplifier design. We visualize the norm of the nodal velocity field simulated in square domains of dimension 32, 256 and 2048 (top left, top middle, bottom left). The relative error (to the solution at dimension 2048) monotonically decreases as the resolution increases. We additionally simulate the same design at 2048x2048 using a traditional Stokes flow solver from Du et al. [2020] (bottom middle), and visualize the difference field (bottom right, normalized to inflow value).

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 100×100 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

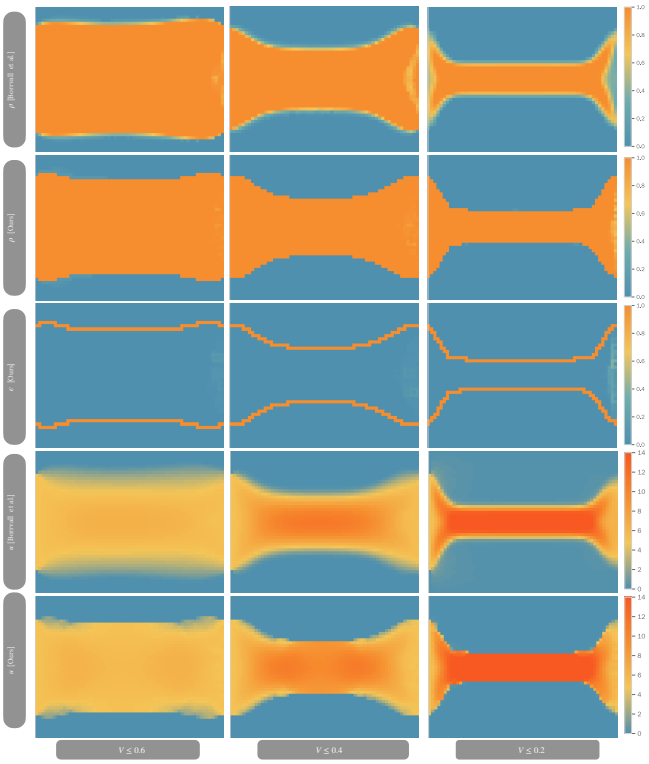

Fig. 8. Comparisons between our method and [Borrvall and Petersson 2003] using a 2D design problem (60×60 cells). The domain has one inlet and outlet at its left and right, respectively. The goal is to synthesize a structure that transports the inlet flow with velocity $(v_i, 0)$ to the outlet. From left to right: 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 isotropicity field ϵ 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.

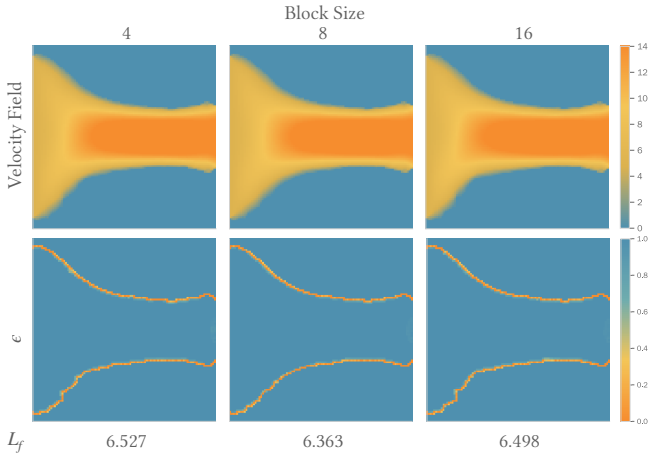

Fig. 9. Sensitivity of optimization results to block size. We optimize the 2D amplifier design problem with block sizes 4, 8, and 16 where initial L_f = 141.938. We visualize the norm of the velocity field (top) and the optimized isotropy field (bottom) and report the optimized L_f value (bottom).

presented. We used the open source implementation for comparison with [Du et al. 2020].

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 × 60 cells). The inlet flow is given by $(v_i, 0)$ and enforced as the Dirichlet boundaries, and the goal is to create a design so that the outlet flow at each node is $(v_i, 0)$ 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 $V_{\max}$ ranging from 60% to 20% of the domain (Fig. 8, left to right) and run both methods. As we decrease $V_{\max}$, 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 $V_{\max}$ to its minimum (Fig. 8 bottom), which we attribute to the anisotropic material model.

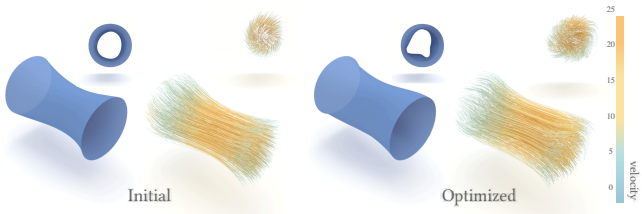

Fig. 10. Solving the Fluidic Twister example using the open-source implementation from Du et al. [2020]. Left: The parametric shape and fluid flow visualization before optimization, with the top-right insets showing the designs viewed from their outlets. Right: The corresponding visualizations after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

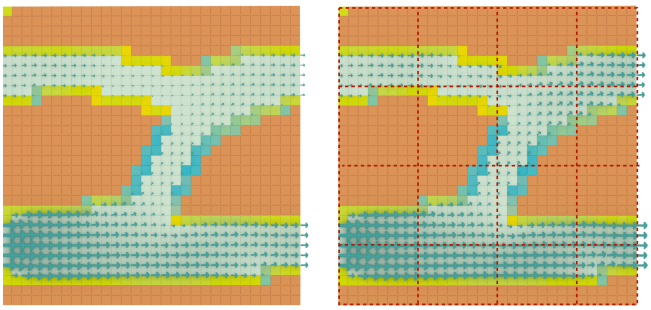

Fig. 11. We study the effects of block divergence for simulating divergence-free Stokes flow using this two-inlet, two-outlet design problem in 2D (30×30). The inlets and outlets are defined on the left and right sides of the domain, respectively. Left: We model the design with moderate k_f and λ_0 without imposing the block divergence constraints in simulation. The outflux on the two outlets is 67% of the influx from the two inlets, indicating the flow velocities dissipate into the solid phase inside the domain. Right: We rerun the same simulation with block divergence constraints with block sizes of 8×8 (dashed red lines). The resulting outflux is 100% of the influx.

(3.31), indicating that we explored a much larger design space thanks to the expressiveness of the field-based representation.

7.3 Ablation Study

Block divergence. To understand the effects of the block divergence constraints in our numerical simulation, we consider a 2D example with two inlets and two outlets on the left and right sides of a square domain (Fig. 11). We use moderately large material parameters k_f and λ_0 defined in Table 1 to model the design and simulate the Stokes flow with and without block divergence. To quantify the divergence in the whole domain, we compute the ratio between the outflux at the two outlets and the influx from the two inlets, which is about 67% without block divergence and 100% with block divergence. These numbers indicate that moderate material parameters, while friendly to a numerical solver, create leaky flows disappearing into the solid phase in the domain. With block divergence, however, the whole domain remains divergence-free in an aggregated sense without messing with the conditioning of the numerical system.

8 CONCLUSIONS

In this paper, we provided a density-based fluidic topology optimization pipeline that handles flexible boundary conditions in the fluidic phase. Our core contribution is to present an anisotropic material model that uniformly represents different phases and flexible boundaries in the Stokes flow model. Building on top of this physical model, we develop numerical solutions to its geometric representation, simulation, and optimization. We ran ablation studies that checked the validity of our approach, and comparisons with existing methods confirmed the superiority of our approach in designing fluidic devices with delicate structures and flexible boundary types.

9 LIMITATION AND FUTURE WORK

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 adressed. 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. Eftychios 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.

Christie Alappat, 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

Joe Alexandersen and Casper Schousboe Andreasen. 2020. A Review of Topology Optimisation for Fluid-Based Problems. *Fluids* 5, 1 (2020). https://doi.org/10.3390/fluids5010029

Casper Schousboe Andreasen, Allan Roulund Gersborg, and Ole Sigmund. 2009. Topology optimization of microfluidic mixers. *International Journal for Numerical Methods in Fluids* 61, 5 (2009), 498–513.

Casper Schousboe Andreasen and Ole Sigmund. 2013. Topology optimization of fluid–structure-interaction problems in poroelasticity. *Computer Methods in Applied Mechanics and Engineering* 258 (2013), 55–62.

Moritz Bächer, Bernd Bickel, Emily Whiting, and Olga Sorkine-Hornung. 2017. Spin-It: Optimizing Moment of Inertia for Spinnable Objects. *Commun. ACM* 60, 8 (jul 2017), 92–99. https://doi.org/10.1145/3068766

P.J. Basser, J. Mattiello, and D. Lebihan. 1994. Estimation of the Effective Self-Diffusion Tensor from the NMR Spin Echo. *Journal of Magnetic Resonance, Series B* 103, 3 (1994), 247–254.

Thomas Borrvall and Joakim Petersson. 2003. Topology optimization of fluids in Stokes flow. *International journal for numerical methods in fluids* 41, 1 (2003), 77–107.

Walter Jesus Paucar Casas and Renato Pavanello. 2017. Optimization of fluid-structure systems by eigenvalues gap separation with sensitivity analysis. *Applied Mathematical Modelling* 42 (2017), 269–289.

Vivien J. Challis and James K. Guest. 2009. Level Set Topology Optimization of Fluids in Stokes Flow. *Internat. J. Numer. Methods Engrg.* 79, 10 (2009), 1284–1308.

Joshua D Deaton and Ramana V Grandhi. 2014. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization* 49, 1 (2014), 1–38.

Yongbo Deng, Zhenyu Liu, and Yihui wu. 2012. Topology optimization of steady and unsteady incompressible Navier–Stokes flows driven by body forces. *Structural and Multidisciplinary Optimization* 47 (11 2012). https://doi.org/10.1007/s00158-012-0847-8

Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2021. DiffPD: Differentiable Projective Dynamics. *ACM Trans. Graph.* 41, 2, Article 13 (nov 2021), 21 pages. https://doi.org/10.1145/3490168

Tao Du, Kui Wu, Andrew Spielberg, Wojciech Matusik, Bo Zhu, and Eftychios Sifakis. 2020. Functional Optimization of Fluidic Devices with Differentiable Stokes Flow. *ACM Trans. Graph.* 39, 6, Article 197 (Dec. 2020), 15 pages. https://doi.org/10.1145/3414685.3417795

Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. 2002. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics* 183, 1 (2002), 83–116.

Anton Evgrafov. 2006. Topology Optimization of Slightly Compressible Fluids. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics* 86, 1 (2006), 46–62.

Stanley Osher Ronald Fedkiw and Stanley Osher. 2002. Level set methods and dynamic implicit surfaces. *Surfaces* 44, 77 (2002), 685.

F. J. Gaspar, J. L. Gracia, F. J. Lisbona, and C. W. Oosterlee. 2008. Distributive smoothers in multigrid for problems with dominating grad–div operators. *Numerical Linear Algebra with Applications* 15, 8 (2008), 661–683. https://doi.org/10.1002/nla.587 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nla.587

Allan Gersborg-Hansen, Ole Sigmund, and Robert B. Haber. 2005. Topology Optimization of Channel Flow Problems. *Structural and Multidisciplinary Optimization* 30, 3 (2005), 181–192.

James K. Guest and Jean H. Prévost. 2006. Topology Optimization of Creeping Fluid Flows Using a Darcy–Stokes Finite Element. *Internat. J. Numer. Methods Engrg.* 66, 3 (2006), 461–484.

Cyril W Hirt and Billy D Nichols. 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics* 39, 1 (1981), 201–225.

Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B. Tenenbaum, William T. Freeman, Jiajun Wu, Daniela Rus, and Wojciech Matusik. 2018. ChainQueen: A Real-Time Differentiable Physical Simulator for Soft Robotics. https://doi.org/10.48550/ARXIV.1810.01054

Antony Jameson. 1995. Optimum Aerodynamic Design Using CFD And Control Theory. *CFD Review* 3 (06 1995). https://doi.org/10.2514/6.1995-1729

EA Kontoleontos, EM Papoutsis-Kiachagias, AS Zymaris, DI Papadimitriou, and KC Giannakoglou. 2013. Adjoint-based constrained topology optimization for viscous flows, including heat transfer. *Engineering Optimization* 45, 8 (2013), 941–961.

Yijing Li and Jernej Barbič. 2015. Stable anisotropic materials. *IEEE transactions on visualization and computer graphics* 21, 10 (2015), 1129–1137.

Yijing Li and Jernej Barbič. 2015. Stable Anisotropic Materials. *IEEE Transactions on Visualization and Computer Graphics* 21, 10 (2015), 1129–1137.

Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. 2022. DiffCloth: Differentiable Cloth Simulation with Dry Frictional Contact. *ACM Trans. Graph.* (mar 2022). https://doi.org/10.1145/3527660 Just Accepted.

Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. 2018. Narrow-Band Topology Optimization on a Sparsely Populated Grid. *ACM Trans. Graph.* 37, 6, Article 251 (dec 2018), 14 pages.

Tadayoshi Matsumori, Tsuguo Kondoh, Atsushi Kawamoto, and Tsuyoshi Nomura. 2013. Topology optimization for fluid–thermal interaction problems under constant input power. *Structural and Multidisciplinary Optimization* 47, 4 (2013), 571–581.

K Maute and M Allen. 2004. Conceptual design of aeroelastic structures by topology optimization. *Structural and Multidisciplinary Optimization* 27, 1-2 (2004), 27–42.

Joel McCormack, Ronald Perry, Keith I. Farkas, and Norman P. Jouppi. 1999. Feline: Fast Elliptical Lines for Anisotropic Texture Mapping. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 243–250.

Rahul Narain, Armin Samii, and James F O'brien. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)* 31, 6 (2012), 1–10.

Carsten Othmer, Eugene de Villiers, and Henry Weller. 2007. Implementation of a continuous adjoint for topology optimization of ducted flows. In *18th AIAA Computational Fluid Dynamics Conference*. the American Institute of Aeronautics and Astronautics, Reston, VA, USA, 3947.

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Trans. Graph.* 34, 4, Article 135 (July 2015), 12 pages.

Taylor Patterson, Nathan Mitchell, and Eftychios Sifakis. 2012. Simulation of complex nonlinear elastic bodies using lattice deformers. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–10.

Tobias Pfaff, Nils Thuerey, Jonathan Cohen, Sarah Tariq, and Markus Gross. 2010. Scalable Fluid Simulation Using Anisotropic Turbulence Particles. *ACM Trans. Graph.* 29, 6, Article 174 (dec 2010), 8 pages.

George IN Rozvany. 2009. A critical review of established methods of structural topology optimization. *Structural and multidisciplinary optimization* 37, 3 (2009), 217–237.

Camille Schreck and Chris Wojtan. 2020. A Practical Method for Animating Anisotropic Elastoplastic Materials. *Computer Graphics Forum* 39, 2 (2020), 89–99.

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to Control Elasticity in 3D Printing. *ACM Trans. Graph.* 34, 4, Article 136 (jul 2015), 13 pages.

Ole Sigmund and Kurt Maute. 2013. Topology optimization approaches. *Structural and Multidisciplinary Optimization* 48, 6 (2013), 1031–1055.

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing inflatable structures. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 63.

Mark Sussman and Elbridge Gerry Puckett. 2000. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of computational physics* 162, 2 (2000), 301–337.

Krister Svanberg. 1987. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering* 24, 2 (1987), 359–373.

Jiaping Wang, Shuang Zhao, Xin Tong, John Snyder, and Baining Guo. 2008. Modeling Anisotropic Surface Reflectance with Example-Based Microfacet Synthesis. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–9.

Lingfeng Wang and Emily Whiting. 2016. Buoyancy Optimization for Computational Fabrication. *Computer Graphics Forum (Proceedings of Eurographics)* 35, 2 (2016).

Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. 2018. Infill Optimization for Additive Manufacturing—Approaching Bone-Like Porous Structures. *IEEE Transactions on Visualization and Computer Graphics* 24, 2 (2018), 1127–1140.

Yuwei Xiao, Szeyu Chan, Siqi Wang, Bo Zhu, and Xubo Yang. 2020. An adaptive staggered-tilted grid for incompressible flow simulation. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Kentaro Yaji, Takayuki Yamada, Seiji Kubo, Kazuhiro Izui, and Shinji Nishiwaki. 2015. A topology optimization method for a coupled thermal–fluid problem using level set boundary expressions. *International Journal of Heat and Mass Transfer* 81 (2015), 878–888.

Gil Ho Yoon. 2010. Topology Optimization for Stationary Fluid-Structure Interaction Problems Using a New Monolithic Formulation. *Internat. J. Numer. Methods Engrg.* 82, 5 (2010), 591–616.

Jihun Yu and Greg Turk. 2013. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Transactions on Graphics (TOG)* 32, 1 (2013), 1–12.

Shiwei Zhou and Qing Li. 2008. A Variational Level Set Method for the Topology Optimization of Steady-State Navier-Stokes Flow. *J. Comput. Phys.* 227, 24 (2008), 10178–10195.

Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Trans. Graph.* 36, 4, Article 120b (jul 2017), 16 pages.