Boundary control for optimal mixing via Stokes flows and numerical implementation

Weiwei Hu * Xiaoming Zheng†

Abstract

This work develops scientific computing techniques to further the exploration of using boundary control alone to optimize mixing in Stokes flows. The theoretical foundation including mathematical model and the optimality conditions for solving the optimal control has been established by Hu and Wu in a series of work. The scalar being mixed is purely advected by the flow and the control is exerted tangentially on the domain boundary through the Navier slip conditions. The control design is motivated by the physical observations that the moving or rotating walls accelerate mixing.

A gradient descent-based optimization algorithm is designed. A critical problem is the computation of the Gâteaux derivative or the gradient of the cost functional. Two methods are proposed: one is based on the Variational Formula (VF) and one utilizes Algorithmic Differentiation (AD). The convergence of the algorithm is studied and various designs of boundary control using cosine and sine functions with time segmentation are computed.

The algorithm has a first order convergence rate and the VF method is more efficient by taking only one third of the time as the AD method when the dimension of control basis is large. The numerical implementations show that the boundary control produces similar mixing results as internal mixings in the existing literature. The mixing effect becomes better when more diverse basis control functions and more time segmentation are utilized. It is shown that the mixing decay rate in time follows power rules, approximately. The numerical study in this work suggests that boundary control alone could be an effective strategy for mixing in incompressible fluid flows.

Keyword. optimal mixing, boundary control, pure advection, Stokes flow, optimality conditions, gradient descent algorithm

1 Introduction

Transport and mixing in fluids is a topic of great interest and fundamental importance in engineering and natural sciences, with broad applications ranging from industrial and chemical mixing on various scales, to preventing the spreading of environmental pollutants. A long-lasting and central problem is to design the velocity field that can enhance or prevent transport and mixing, or steer a scalar field to a desired distribution, which has drawn great attention to researchers in many fields. The current work is concerned with the control design for optimal mixing via flow advection, and the corresponding numerical schemes to implement the design. The mathematical model has three features: the flow velocity is governed by the Stokes equations, the scalar or density being mixed has

*Department of Mathematics, University of Georgia, Athens, GA 30602, USA (e-mail: Weiwei.Hu@uga.edu)
†Department of Mathematics, Central Michigan University, Mount Pleasant, MI 48859, USA (e-mail: zheng1x@cmich.edu)
no molecular diffusion and thus is purely advected by the flow, and the control is exerted through the domain boundary for steering mixing. Although it is widely accepted that many mixing processes occur in turbulent flows, effective mixing can also occur in laminar flows under certain conditions. In fact, many mixing experiments are conducted in Stokes flows with Reynolds number one or less (cf. [35, 3]). The neglect of diffusion can be justified in the case where the Péclet number (the ratio of the rate of advection to the rate of diffusion) is sufficiently large. In this case, the transport and mixing are dominated by the flow advection.

Optimal mixing and stirring of passive scalars via pure advection has been extensively discussed by means of theoretical analysis and numerical simulations in recent years (cf. [3, 18, 27, 31, 32, 30, 36]), where the majority focuses on the mixing subject to prescribed velocity fields or a set of flows with constraints on toroidal domains. The optimal control approach for fluid mixing in stationary Stokes flows has been studied, for example, by Vikhansky in [39] and Matthew et al. in [27]. These works assume that the flow is induced by a finite set of force fields that can be modulated arbitrarily in time. However, real-time control of the flow dynamics was not taken into account in the aforementioned literature. Recently, Hu and Wu in [19, 22, 18, 23] have conducted a theoretical study on optimal mixing via boundary control of the flow dynamics governed by the incompressible Stokes and the Navier-Stokes equations.

The boundary control design is motivated by the study that rotating or moving walls accelerate mixing compared with fixed walls (cf. [3, 10, 11, 12, 16, 38]). Chakravarthy and Ottino in [3] showed that moving walls with time-periodic forcing play a significant role in the rate of stretching the interface of two initially segregated fluids at low Reynolds number, where the length of the interface is used as a good indicator of the degree of mixing. The results in [11, 10, 38] have also shown that fixed walls with no-slip boundary condition can turn an exponential decay in time into a power decay due to the presence of separatrices on the walls, which slow down the whole mixing region. However, this can be overcome by moving the walls to create closed orbits near them, which effectively insulate the central mixing region from the walls [12, 38]. In these studies, the moving/rotating walls were not physically implemented and thus their effects were investigated numerically. The objective of the current work is to investigate the numerical schemes for implementing the control design established in [19, 18], i.e., optimal boundary control for mixing via Stokes flows. Specifically, the control inputs are applied tangentially on the domain boundary through the Navier slip boundary conditions.

Although the first order necessary conditions of optimality have been established for characterizing the optimal controller in [19, 18], they have not been exemplified numerically. In fact, there are barely any numerical algorithms developed for solving the optimal control design for mixing governed by the coupled flow-transport system in a general open bounded domain. This work, to the authors’ best knowledge, is the first numerical study of optimal mixing via boundary control of the unstationary Stokes flow. However, implementing the boundary control for fluid mixing encounters new and significant challenges in computation. First, to solve the resulting optimality system, one has to solve the governing system forward in time, coupled with the adjoint system backward in time together with a nonlinear optimality condition. It presents a formidable computational challenge even for relatively simple geometries with relatively coarse meshes. The second major obstacle is due to mass conservation of scalar transport in incompressible flows. Specifically, the approximate velocity being divergence free is critical to preserving the accuracy, stability, and global conservation properties of the scalar. Moreover, small-scale structures and large gradients of the scalar field will develop in the mixing process. The mesh size must be small enough to capture the smallest spatial scales of the thin filaments that arise in evolution.

The rest of this paper is outlined as follows. In section 2, we present the mathematical model of boundary control design for optimal mixing in Stokes flows. In section 3, we introduce a gradient
descent numerical method for optimization and the mixed finite elements and discontinuous Galerkin (DG) methods for discretizing the governing system. In section 4, we first present two relatively long-time (the time is from 0 to 100) forward simulations to demonstrate the effectiveness of mixing in Stokes flows. In section 5, we conduct the convergence study of the optimization algorithm. Finally, we apply the numerical algorithm to investigate various boundary control designs for mixing problems in sections 6-7. The conclusions are presented in section 8.

2 Boundary control design for optimal mixing

2.1 Mathematical model

Consider a passive scalar field advected by an incompressible Stokes flow on an open bounded and connected domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, with a sufficiently smooth boundary $\Gamma$. The scalar field is governed by the transport equation. The mathematical model considered in this work is governed by

$$\frac{\partial \theta}{\partial t} + v \cdot \nabla \theta = 0,$$

(2.1)

$$\frac{\partial v}{\partial t} - \Delta v + \nabla p = 0,$$

(2.2)

$$\nabla \cdot v = 0, \quad x \in \Omega,$$

(2.3)

with the Navier slip boundary conditions (cf. [25, 34]),

$$v \cdot n|_{\Gamma} = 0 \quad \text{and} \quad (2n \cdot D(v) \cdot \tau + kv \cdot \tau)|_{\Gamma} = g \cdot \tau,$$

(2.4)

and the initial condition is given by

$$(\theta(0), v(0)) = (\theta_0, v_0),$$

(2.5)

where $\theta$ is the density, $v$ is the velocity, $p$ is the pressure, $D(v) = (1/2)(\nabla v + (\nabla v)^T)$ is the strain tensor, and $n$ and $\tau$ denote the outward unit normal and tangential vectors with respect to the domain boundary $\Gamma$. Navier slip boundary conditions admit the fluid to slip with resistance on the boundary and the friction between the fluid and the wall is proportional to $-v$ with the positive coefficient of proportionality $k$. Here $g$ satisfying $g \cdot n|_{\Gamma} = 0$ is the tangential boundary control input employed to steer the velocity field for mixing.

2.2 Mix-norm and optimal control formulation

Due to the divergence-free and no-penetration boundary conditions imposed on the velocity field, it can be shown that any $L^p$-norm of $\theta$ is conserved (cf. [20, 2]), i.e.,

$$\|\theta(t)\|_{L^p} = \|\theta_0\|_{L^p}, \quad t \geq 0, \quad p \in [1, \infty].$$

(2.6)

To quantify mixing, a classical measure is the spatial variance of the scalar concentration [6], which is related to the $L^2$-norm of the scalar field. However, this measurement fails in the case of zero diffusivity since it is unable to quantify pure stirring effects, as any $L^p$-norm of $\theta$ is invariant with respect to time indicated by (2.6). Recently, the mix-norm and negative Sobolev norms have been adopted to quantify this process based on ergodic theory, which are sensitive to both stirring and
In [33], Mathew et al. first showed the equivalence of the mix-norm to the $H^{-1/2}$-norm on a $d$-dimensional torus $\mathbb{T}^d$. In fact, any negative Sobolev norm $H^{-s}$, for $s > 0$, can be used as a mix-norm thanks to the property of weak convergence [37]. Since a general open and bounded domain will be considered in this paper, the negative Sobolev norm will be replaced by the norm of the dual space $(H^s(\Omega))'$ of $H^s(\Omega)$ with $s > 0$, as in [19, 18]. Without loss of generality, we shall continue to use $(H^1(\Omega))'$ following [19, 22, 18, 23], to quantify mixing in this work.

Throughout this paper, we use $(\cdot, \cdot)$ and $\langle \cdot, \cdot \rangle$ for the $L^2$-inner products in the interior of the domain $\Omega$ and on the boundary $\Gamma$, respectively. To set up the abstract formulation for the velocity field, we define

$$V^s_n(\Omega) = \{ v \in H^s(\Omega) : \text{div } v = 0, \ v \cdot n|_\Gamma = 0 \}, \quad s \geq 0,$$

$$V^s_n(\Gamma) = \{ g \in H^s(\Gamma) : g \cdot n|_\Gamma = 0 \}, \quad s \geq 0,$$

where $H^0(\Omega) = L^2(\Omega)$ and $H^0(\Gamma) = L^2(\Gamma)$.

The optimal control problem is formulated as follows. For a given $T > 0$, find a control $g$ minimizing the cost functional

$$J(g) = \frac{1}{2} \| \theta(T) \|_{(H^1(\Omega))'}^2 + \frac{\gamma}{2} \| g \|_{U_{\text{ad}}}^2,$$

subject to (2.1)–(2.5), where $\gamma > 0$ is the control weight parameter and $U_{\text{ad}}$ is the set of admissible controls. The choice of $U_{\text{ad}}$ is often determined based on the physical properties as well as the need to guarantee the existence of an optimal solution. Here we set $U_{\text{ad}} = L^2(0, T; V^0_n(\Gamma))$ as used in [18]. The details of the choice of $U_{\text{ad}}$ are explained in Appendix 9.1.

Note that the boundary control of the velocity field gives rise to a bilinear control problem of the scalar equation, due to the one-way coupling through the advective term $v \cdot \nabla \theta$, and therefore, the problem (P) is non-convex. To solve problem (P), we apply a variational inequality [28], that is, if $g$ is an optimal solution of problem (P), then

$$J'(g) \cdot (\varphi - g) \geq 0, \quad \forall \varphi \in U_{\text{ad}},$$

(2.7)

where $J'(g) \cdot \varphi$ stands for the Gâteaux derivative of $J$ with respect to $g$ in every direction $\varphi \in U_{\text{ad}}$. A rigorous definition is given by

$$J'(g) \cdot \varphi = \lim_{\delta \to 0} \frac{J(g + \delta \varphi) - J(g)}{\delta} = \frac{dJ(g + \delta \varphi)}{d\delta} \bigg|_{\delta = 0}, \quad \forall \varphi \in U_{\text{ad}}.$$

If the limit exists for all $\varphi \in U_{\text{ad}}$, then $J$ is called Gâteaux differentiable at $g$. The process of solving (2.7) involves finding a suitable adjoint system associated with the governing model (2.1)–(2.5) and using it to explicitly compute $J'(g)$. The resulting first order optimality system provides the characterization of the optimal control and will be used to construct a gradient based algorithm to compute the optimal solution in our work.

**Remark 2.1.** Note a concept ‘action’ is used in [32] as a constraint for mixing, which is defined as the time integral of the kinetic energy. As pointed out in Appendix 9.4, this concept lacks physical meanings and is not used here. But the kinetic energy is employed to check the flow properties in some examples in this work.
2.3 First order optimality conditions

To better interpret the dual norm \((H^1(\Omega))'\), we introduce a higher regularity counterpart of \(\theta\), denoted as \(\eta\), and define it by

\[
(-\Delta + I)\eta = \theta \quad \text{in } \Omega, \quad \frac{\partial \eta}{\partial n} = 0 \quad \text{on } \Gamma.
\]

(2.8)

Letting \(\Lambda = (-\Delta + I)^{1/2}\), then \(\|\theta\|_{(H^1(\Omega))'} = \|\Lambda^{-1}\theta\|_{L^2(\Omega)}\) \cite{18} and the cost functional \(J\) becomes

\[
J(g) = \frac{1}{2}(\Lambda^{-2}\theta(T),\theta(T)) + \frac{\gamma}{2} \int_0^T \langle g,g \rangle_{\Gamma} \, dt.
\]

(2.9)

We define an operator \(L : U_{ad} \to L^2(0,T;V_n^0(\Omega))\) such that for \(g \in U_{ad}\), \(Lg\) is the velocity field of the Stokes system (2.2)–(2.5) when \(v_0 = 0\) and the boundary control input is \(g\). The rigorous definition of \(L\) is given in Appendix 9.2 and it is a linear operator according to \cite{18}. Let \(L^*\) denote the \(L^2(0,T;\cdot)\)- adjoint operator of \(L\). The following results establish the first order optimality conditions for charactering the optimal solution to problem \((P)\) by using the variational inequality (2.7), which are proven in \cite{18}, Theorem 5.5 and Theorem 6.1 through an approximating control approach.

**Theorem 2.2.** Let \(\theta_0 \in L^\infty(\Omega) \cap H^1(\Omega)\) and \(v_0 \in V_n^{d/2-1+\epsilon}(\Omega)\), where \(d = 2, 3\) and \(\epsilon > 0\) is sufficiently small. Assume that \(g \in U_{ad} = L^2(0,T;L^2(\Gamma))\) is an optimal solution to problem \((P)\). If \((v,\theta)\) is the corresponding solution to the governing system (2.1)–(2.5) and \(\rho\) is the solution to the adjoint problem

\[
-\frac{\partial}{\partial t}\rho - v \cdot \nabla \rho = 0, \\
\rho(T) = \Lambda^{-2}\theta(T),
\]

(2.10)

(2.11)

then the first order optimality condition is given by

\[
\gamma g + L^*(\mathbb{P}(\theta\nabla \rho)) = 0 \quad \text{or} \quad g = -\frac{1}{\gamma} L^*(\mathbb{P}(\theta\nabla \rho)) \in L^2(0,T;V_n^{3/2}(\Gamma)) \cap H^{3/4}(0,T;V_n^0(\Gamma)).
\]

(2.12)

Furthermore, the optimal solution \(g\) is unique when \(d = 2\) and \(\gamma\) is sufficiently large.

With the help of the condition (2.11), the cost functional \(J\) given by (2.9) can be rewritten as

\[
J(g) = \frac{1}{2} \int_\Omega \rho(x,T)\theta(x,T) \, dx + \frac{\gamma}{2} \int_0^T \langle g,g \rangle_{\Gamma} \, dt.
\]

(2.13)

This is the formula used in the numerical computation of the cost. It involves two steps: first, evolving \(\theta\) from \(t = 0\) to \(t = T\) with the boundary control \(g\); second, computing \(\rho(T)\) from (2.11).

In addition, the following relation between \(\theta\) and \(\rho\) holds, which is proven in Appendix9.3 and is used to validate numerical scheme in Section3.7.

**Corollary 2.3.** For a fixed final time \(T > 0\), let \(\rho\) be the solution to the adjoint system (2.10)–(2.11). Then the quantity \(\int_\Omega \rho(x,t)\theta(x,t) \, dx\) is invariant with respect to \(t \in [0,T]\).
2.4 Weak formulation

Using Green’s formula, one can easily show that the weak formulation for Stokes problem with non-homogeneous Navier slip boundary conditions (2.2)-(2.5) is given by

\[
\frac{\partial v}{\partial t},w + 2(D(v),D(w)) + k\langle v \cdot \tau, w \cdot \tau \rangle_{\Gamma} = \langle g \cdot \tau, w \cdot \tau \rangle_{\Gamma},
\]

where \( w \in V^1_n(\Omega) \) is the test function and \( \langle g \cdot \tau, w \cdot \tau \rangle_{\Gamma} = \langle g, w \rangle_{\Gamma} \). The weak form of the optimality condition (2.12) reads

\[
\langle g, \varphi \rangle_{\Gamma} = -\frac{1}{\gamma}(L^*(P(\theta \nabla \rho)), \varphi)_{\Gamma} = -\frac{1}{\gamma}(P(\theta \nabla \rho), L\varphi),
\]

for every \( \varphi \in V^0_n(\Gamma) \). Because \( L\varphi \) is divergence free and \( (L\varphi) \cdot n|_{\Gamma} = 0 \), (2.15) becomes

\[
\langle g, \varphi \rangle_{\Gamma} = -\frac{1}{\gamma}(\theta \nabla \rho, L(\varphi)).
\]

In summary, if \((g, \hat{v}, \theta)\) is an optimal solution, then it satisfies the following optimality system in weak form

\[
\begin{align*}
(\partial_t \theta, \phi) - (\theta, \hat{v} \cdot \nabla \phi) &= 0, \quad \theta(0) = \theta_0, \\
(-\partial_t \rho, \psi) + (\rho, \hat{v} \cdot \nabla \psi) &= 0, \quad \rho(T) = \Lambda^{-2} \theta(T), \\
(\partial_t \hat{v}, w) + 2(D(\hat{v}),D(w)) + k(\hat{v} \cdot \tau, w \cdot \tau)_{\Gamma} &= \langle g \cdot \tau, w \cdot \tau \rangle_{\Gamma}, \quad \hat{v}(0) = v_0, \\
\langle g \cdot \tau, w \cdot \tau \rangle_{\Gamma} &= \langle g, w \rangle_{\Gamma} = -\frac{1}{\gamma}(\theta \nabla \rho, L(w|_{\Gamma})),
\end{align*}
\]

for every \( \phi \in H^1(\Omega), \psi \in H^1(\Omega) \), and \( w \in V^1_n(\Omega) \).

Directly solving (2.17) encounters several numerical challenges. First of all, the numerical schemes to be constructed are required to capture the hyperbolic-parabolic features of (2.17). Second, the governing system needs to be solved forward in time, coupled with the adjoint system backward in time together with a nonlinear optimality condition. Furthermore, the elliptic problem (2.8) has to be solved in order to compute the mixing norm. The whole process leads to extremely high computational costs and often intractable problems. We shall integrate the mixed discontinuous and continuous Galerkin methods together with the gradient decent based algorithms to tackle these difficulties.

3 Numerical methods for optimization

To solve the optimization problem \((P)\) numerically, we employ a gradient descent method whose fundamental idea is to seek for an iterative sequence \(\{g^0, g^1, \ldots, g^n, g^{n+1}, \ldots\}\) following

\[
g^{n+1} = g^n - \beta^n \nabla J(g^n),
\]

where the scalar \( \beta^n > 0 \) is determined by the backtracking approach. The basic algorithm consists of an outer iteration generating a sequence \(\{g^0, g^1, \ldots, g^n, g^{n+1}, \ldots\}\) and an inner iteration in each step of proceeding from \(g^n\) to \(g^{n+1}\).
3.1 Control basis and cost gradient $\nabla J$

From the viewpoint of real world applications, a finite number of control inputs is a more realistic assumption since it is not practical to create arbitrarily distributed force fields for stirring. In the rest of our work, we restrict our interest to the control input function of the form

$$g(x, t) = \sum_{i=1}^{M} \alpha_i g_i(x, t),$$

(3.2)

where $\{g^1, \ldots, g^M\} \subset L^2(0, T; V_n^0(\Gamma))$ forms a finite basis and $\alpha_i \in \mathbb{R}, i = 1, 2, \ldots, M$, are the control weight parameters. In this case, $U_{ad} = \text{span}\{g_i\}_{i=1}^M$ equipped with the norm $\|\cdot\|_{U_{ad}} = \|\cdot\|_{L^2(0, T; L^2(\Gamma))}$ is a finite dimensional Hilbert space. Moreover, if $g_i$ is sufficiently smooth, then $J$ is Gâteaux differentiable at $g$ and

$$J'(g) \cdot \varphi = (J'(g), \varphi)_{U_{ad}} = \gamma \int_0^T \langle g, \varphi \rangle_{\Gamma} dt + \int_0^T (\theta \nabla \rho, L \varphi) dt = \int_0^T \langle \gamma g + L^*(\theta \nabla \rho), \varphi \rangle_{\Gamma} dt,$$

(3.3)

for any $\varphi \in U_{ad}$, where $\theta$ is the solution of the transport equation in the system (2.1)–(2.5) with boundary input $g$, and $\rho$ is the adjoint state solved from (2.10)–(2.11). Note that for the current choice of $U_{ad}$, if $g$ is an optimal solution, then the variational inequality (2.7) becomes

$$J'(g) \cdot \varphi = 0, \quad \forall \varphi \in U_{ad}.$$  

(3.4)

This implies

$$\gamma g + L^*(\theta \nabla \rho) = 0 \quad \text{or} \quad g = -\frac{1}{\gamma} L^*(\theta \nabla \rho),$$

(3.5)

which is another form of (2.16). A formal derivation of (3.3) is provided in Appendix 9.5 and for the rigorous proof concerning the differentiability of $J$, the readers are referred to [19, 18]. Note that the Gâteaux differential $J'(g)$ is a linear functional on $U_{ad}$. Since $U_{ad}$ is a Hilbert space, we use the gradient $\nabla J(g) \in U_{ad}$ as the Reisz representation of $J'(g)$ (cf. [17, p. 67]) in our numerical implementations. Therefore, $\nabla J(g)$ can be expressed as a linear combination of $g_i$’s. Denote the coefficient of $g_i$ as $(\nabla J(g))_i$. This leads to the following form

$$\nabla J(g) \triangleq \sum_{i=1}^{M} (\nabla J(g))_i g_i.$$  

(3.6)

3.2 Numerical methods to compute the cost gradient $\nabla J$

We shall present two numerical methods to compute $\nabla J(g)$. The first one utilizes the optimality condition (2.16) derived from the Variational Formula (VF). The second one directly computes the Algorithmic Differentiation (AD) of $J(g)$ without involving the adjoint system. Numerical experiments will be conducted to compare these two approaches in Section 5.2.

3.2.1 Method 1: Variational Formula (VF)

To recover the gradient $\nabla J(g)$, we use the representation $\nabla J(g) = \sum_{i=1}^{M} (\nabla J(g))_i g_i$ in (3.6). According to (3.3), if $\varphi = g_j, \ j = 1, 2, \ldots, M$, then we get the following linear system

$$\sum_{i=1}^{M} (\nabla J(g))_i (g_i, g_j) = (\nabla J(g), g_j) = \gamma \int_0^T \langle g, g_j \rangle_{\Gamma} dt + \int_0^T (\theta \nabla \rho, L(g_j)) dt \triangleq \xi_j, \quad j = 1, \ldots, M.$$  

(3.7)
Let $G$ be the matrix $G_{ij} = (g_i, g_j)_{U_{ad}}, i, j = 1, \cdots, M$, and vectors $b = ((\nabla J(g))_1, \cdots, (\nabla J(g))_M)^T$, and $\xi = (\xi_1, \cdots, \xi_M)^T$. Then system (3.7) is reduced to

$$Gb = \xi, \quad (3.8)$$

and thus $b = G^{-1}\xi$. Note that if the basis functions $\{g_i\}_{i=1}^M$ are orthonormal, then $G = I$ and $b = \xi$. Moreover, the squared norm of the gradient $|\nabla J(g)|^2$ used in (3.30) for each iteration is given by

$$|\nabla J(g)|^2 = (\nabla J(g), \nabla J(g))_{U_{ad}} = \left( \sum_{i=1}^M (\nabla J(g)_i)g_i, \sum_{j=1}^M (\nabla J(g)_j)g_j \right)_{U_{ad}} = b^T G b. \quad (3.9)$$

The VF method needs the values of $\theta$ and $\rho$ in the time domain $[0, T]$, which will be computed in the following three steps.

1. Evolving $\theta(t)$ from $\theta_0$ to $\theta(T)$ through (2.1).

2. Solving $\rho(T)$ from (2.11) using the $\theta(T)$ from Step (1).

3. Solving $\rho(t)$ backward from $t = T$ to $t = 0$ with the advection equation (2.10). This equation can be rewritten as a forward problem by letting $s = T - t$ and $\tilde{\rho}(s) = \rho(t)$ and $\tilde{v}(s) = v(t)$. Then $(\tilde{\rho}, \tilde{v})$ satisfies

$$\frac{\partial \tilde{\rho}(s)}{\partial s} - \tilde{v}(s) \cdot \nabla \tilde{\rho}(s) = 0,$$

$$\tilde{\rho}(0) = \rho(T).$$

### 3.2.2 Method 2: Algorithmic Differentiation (AD)

The second method employs an algorithmic differentiation or the finite difference approximation of the directional derivative $(\nabla J(g), \varphi)$, that is,

$$(\nabla J(g), \varphi) \approx \frac{J(g + \delta \varphi) - J(g)}{\delta}, \quad (3.10)$$

where $\delta$ is a small scalar. In our numerical implementations, a typical value of $\delta$ is $1e-4$. With this approach, the linear system (3.7) for $\nabla J(g) = \sum_{i=1}^M (\nabla J(g)_i)g_i$ is replaced by

$$\sum_{i=1}^M ((\nabla J(g)_i)g_i, g_j) = \frac{J(g + \delta g_i) - J(g)}{\delta}, \quad i = 1, \cdots, M. \quad (3.11)$$

Comparing with the first method, this approach does not need to solve the dual problem for $\rho$, but it requires to compute a forward evolution process for each basis function $g_i, i = 1, \cdots, M$, in order to compute $J(g + \delta g_i)$. The computation of $J(g)$ also requires one forward evolution of $\theta$. Therefore, every time of computing $\nabla J$, the AD scheme requires $M + 1$ forward evolutions, while the VF method only takes two evolutions: one forward for $\theta$ and one backward for $\rho$. If $M$ is large, then the AD method of computing the gradient would be very time consuming. The detailed comparison are presented in sections 5 and 6.2.
3.3 Stokes equations: Projection/BDF2/Taylor-Hood Finite element method

The velocity field is required to evolve the scalars $\theta$ and $\rho$ for a given boundary control $g$. Recall when the initial velocity $v_0 = 0$, the velocity field induced by $g$ through the Stokes equations is denoted by $L(g)$ and $L$ is a linear operator. Let $v_i = L(g_i)$. Then the velocity field $v$ induced by $g = \sum_{i=1}^{M} \alpha_i g_i$ is simply given by $v = \sum_{i=1}^{M} \alpha_i v_i$. This enables us to solve the velocity fields $v_i$, $i = 1, \cdots, M$ prior to the optimization process and save them for later use. If $v_0 \neq 0$, we solve velocity by letting $g = 0$ and denote it by $v_{0,g=0}$. Then the full solution $v$ can be written as $v = \sum_{i=1}^{M} \alpha_i v_i + v_{0,g=0}$. However, in our numerical experiments we restrict our discussion to the cases with $v_0 = 0$.

We use the standard Taylor-Hood elements to approximate the velocity and pressure in the Stokes equations (2.2)–(2.4). That is, the velocity is approximated by the continuous $P^2$ functions and the pressure by the continuous $P^1$ functions. Denote the triangulated domain as $\Omega_h$ where all the elements are triangles. The finite element spaces are defined as

$$V_h = \{ w = (w_1, w_2) \in (C^0(\Omega))^2 : w \cdot n|_{\Gamma} = 0, w_i|_{K} \in P^2(K), i = 1, 2, \forall K \subset \Omega_h \},$$

$$Q_h = \{ q \in C^0(\Omega) : q|_{K} \in P^1(K), \forall K \subset \Omega_h \},$$

where $n$ is the unit outward normal on the boundary. Note there are two differences between $V_h$ and $V^n_\tau(\Omega)$. First, the vectors in $V_h$ are not divergence free. Second, the discretized domain $\Omega_h$ is not identical to $\Omega$, a unit disk in this work, because the boundary of $\Omega_h$ is made of straight line segments and $\Omega$ is a circular domain.

The basis functions of $V_h$ are chosen as follow. Denote the inner nodes of the mesh as $x_i$, $i = 1, \cdots, MI$ and the boundary nodes as $x^B_j$, $j = 1, \cdots, BN$. Denote $\phi_i$ as the scalar basis function that is continuous in $\Omega$, piecewise $P^2$ in each element, taking value 1 at node $i$ and zero on all other nodes. Let vectors $e_1 = (1, 0)^T$ and $e_2 = (0, 1)^T$. At an inner node $x_i$, there are two basis functions of velocity, which are $\phi_i e_1$ and $\phi_i e_2$. At a boundary node $x^B_j$, there is only one basis function, $\phi_j e_\tau$, where $e_\tau$ is the unit tangential vector at $x^B_j$. Denote $e_\tau = (e_{\tau_1}, e_{\tau_2})^T$. Then $\phi_j e_\tau = (\phi_j e_{\tau_1}, \phi_j e_{\tau_2})^T$.

The weak form of equations of (2.2)–(2.5) is finding $(v, p) \in V_h \times Q_h$ such that for all $w \in V_h$ and $q \in Q_h$,

$$\int_{\Omega} \frac{\partial v}{\partial t} \cdot w + 2 \int_{\Omega} \nabla(v) : \nabla(w) + \int_{\Gamma} k(v \cdot \tau)(w \cdot \tau) - \int_{\Omega} p \nabla \cdot w = \int_{\Gamma} (g \cdot \tau)(w \cdot \tau),$$

$$\int_{\Omega} q \nabla \cdot v = 0,$$

where $v = (v_1, v_2)^T$, $w = (w_1, w_2)^T$, $\nabla(v) : \nabla(w) = \frac{1}{4} \sum_{i,j=1,2}(\partial_i v_j + \partial_j v_i)(\partial_i w_j + \partial_j w_i)$. The only difference with (2.14) is that here the solution $v$ and the test function $w$ are not divergence free.

A classic projection method with BDF2 time discretization is used to solve the velocity and pressure. Denote the numerical solution at time $t^k$ as $(v^k, p^k)$. To obtain $(v^{k+1}, p^{k+1})$, the so-called “rotational incremental pressure-correction scheme” in [15] is adopted, that is,

$$\frac{1.5\tilde{v}^{k+1} - 2v^k + 0.5v^{k-1}}{\Delta t} + 2\nabla : \nabla(\tilde{v}^{k+1}) + \nabla(2p^k - p^{k-1}) = 0,$$

$$\tilde{v}^{k+1} = v^{k+1} + \Delta t \nabla \phi, \quad \frac{\partial \phi}{\partial n}|_{\Gamma} = 0,$$

$$p^{k+1} = (2p^k - p^{k-1}) + 1.5\phi - \nabla \cdot \tilde{v}^{k+1}.$$

In addition, the Uzawa iterations are added to pursue the divergence free property [9], whose implementation is given below. Denote $l$ as the iteration index at time step $t^k$. For $l = 0, 1, 2, \cdots$,

9
When convergent, we have the estimate
\[ \|p^{s+1} - p^{s-1}\| = (2p^s - p^{s-1}), \]
and
\[
\int_{\Omega} \frac{1}{\Delta t} \left( \frac{1}{2} \vec{v}^{s+1,l+1} - 2 \vec{e}^{s} + 0.5 \vec{v}^{s-1} \right) \cdot w + 2 \int_{\Omega} \mathbb{D}(\vec{v}^{s+1,l+1}) \cdot \mathbb{D}(w) + \int_{\Gamma} k(\vec{v}^{s+1,l+1} \cdot \tau)(w \cdot \tau) = \int_{\Omega} p^{s+1,l+1} \nabla \cdot w + \int_{\Gamma} (g \cdot \tau)(w \cdot \tau), \quad \forall w \in V_h, \tag{3.19}
\]
\[
\int_{\Omega} \nabla \phi^{l+1} \cdot \nabla q = -\frac{1}{\Delta t} \int_{\Omega} q(\nabla \cdot \vec{v}^{s+1,l+1}), \quad \forall q \in Q_h, \tag{3.20}
\]
\[
\int_{\Omega} p^{s+1,l+1} q = \int_{\Omega} (p^{s+1,l+1} + 1.5 \phi^{l+1} - \nabla \cdot \vec{v}^{s+1,l+1}) q, \quad \forall q \in Q_h, \tag{3.21}
\]
\[
\int_{\Omega} v^{s+1,l+1} w = \int_{\Omega} \hat{\theta}^{s+1,l+1} w - \Delta t \phi^{l+1} \nabla \cdot (\nabla \cdot w), \quad \forall w \in V_h. \tag{3.22}
\]

The stopping criterion for the Uzawa iteration can be chosen as when \( \|p^{s+1,l+1} - p^{s+1,l}\|_{L^2(\Omega)} < \varepsilon \). When convergent, we have the estimate \( \|\vec{v}^{s+1,l+1} - \vec{v}^{s+1,l}\| < \varepsilon \), \( \forall q \in Q_h \). The threshold \( \varepsilon \) is set as 1e-10 in this work. Therefore, although the divergence of the numerical velocity is not pointwise zero, it is almost zero in the weak sense.

The convergence test of the projection scheme is presented in Appendix 9.6, where the computation of the end-of-step velocity is found to be indispensable to the numerical stability, which is contrary to some conventional understanding as in [15].

### 3.4 Transport equations: Discontinuous Galerkin method

A standard Runge-Kutta Discontinuous Galerkin (RKDG) scheme [4] is used to solve the scalar \( \theta \) governed by the transport equation (2.1), and the adjoint quantity \( \rho \) from (2.10). Define the discontinuous finite element space
\[
W_{h,M}^{DG} = \{ w_h \in P^M(K), \forall K \subset \Omega_h \}, \tag{3.23}
\]
where \( P^M(K) \) denotes the set of M-th degree polynomials in each triangle \( K \) of the discrete domain \( \Omega_h \). To ensure stability, a Courant-Fredrichs-Lewy condition in [4] is used to determine the time step size \( \Delta t \),
\[
\frac{||v||_{max}}{h} \Delta t \leq \text{CFL}_{L^2} \tag{3.24}
\]
where the constant \( \text{CFL}_{L^2} \) for degree \( M \) of polynomials is given in Table 2.2 of [4].

To show the idea, a first order temporarily discretized numerical scheme is as follows. Given the numerical solution \( \theta^s \in W_{h,M}^{DG} \) at time step \( t^s \), we obtain \( \theta^{s+1} \in W_{h,M}^{DG} \) from
\[
\int_{K} \frac{\theta^{s+1} - \theta^s}{\Delta t} \phi + \int_{e \in \partial K} (v \cdot \hat{n}) \theta^s \phi - \int_{K} \theta^s (v \cdot \nabla \phi) = \int_{K} \hat{\theta}^s \nabla \cdot \phi, \quad \forall \phi \in P^M(K), \tag{3.25}
\]
where \( \hat{n} \) is the unit outward normal on edge \( e \) of \( K \) and \( \hat{\theta}^s \) is the numerical flux. The Godunov flux (see [4] page 206) is used, i.e.,
\[
\hat{\theta}^s_{|\partial K} = \left\{ \begin{array}{ll}
\theta^s_{|K_+} & v \cdot \hat{n} > 0, \\
\theta^s_{|K} & v \cdot \hat{n} < 0,
\end{array} \right. \tag{3.26}
\]
where \( K_+ \) is the neighbour triangle that \( K \) bounds across the edge \( e \). In practice, we use a third order Runge-Kutta scheme to evolve in time, of which the details can be found in [4] p. 90.
3.4.1 Choices of basis functions of $P^M(K)$ and quadrature rules

The basis functions of $P^M(K)$, $M \geq 0$, $K \subset \Omega_h$ are chosen as follows. Denote the center point of $K$ as $(x_0, y_0)$ and a generic basis function as $\phi_{i,j} = (x - x_0)^i (y - y_0)^j$, $i \geq 0, j \geq 0, i + j \leq M$. There are $M_t = (M + 1)(M + 2)/2$ such basis functions, or $\dim(P^M(K)) = M_t$. For any smooth function $\theta(x,y)$, its representation $\theta_h \in P^M(k)$ has the expression $\theta_h = \sum_{i,j \geq 0, i+j \leq M} \theta_{i,j} \phi_{i,j}$, where $
abla^s \theta(x_0,y_0) = \frac{1}{s!} \frac{\partial^{s+i} \theta(x_0,y_0)}{\partial x^i \partial y^j}$. Re-order these bases as $\psi_i = \phi_{i,j}$ where $s = (i + j)(i + j + 1)/2 + (j + 1)$, which is a one-to-one correspondence from the double-index set $\{(i, j) : i \geq 0, j \geq 0, i + j \leq M\}$ to the single-index set $\{1, \cdots, M_t\}$.

The mass matrix $A_{M_t \times M_t}$ on each triangle $K$ is

$$A_{i,j} = \int_K \psi_i(x,y) \psi_j(x,y) \, dx \, dy, \quad i, j = 1, \cdots, M_t. \tag{3.27}$$

Suppose the above integral is approximated by the following quadrature rule,

$$\int_K f(x,y) \approx \sum_{l=1}^G w_l f(x_l,y_l), \tag{3.28}$$

where all the weight $w_l > 0$. Denote the resulting matrix generated from the above quadrature rule as $A^G$. The next lemma provides a necessary condition to ensure the invertibility of $A^G$.

**Lemma 3.1.** For the matrix $A^G$ to be invertible, the number of quadrature points in the triangular integral (3.28), which approximates (3.27), must be greater than or equal to the number of basis functions of $P^M(K)$, that is, $G \geq M_t$.

**Proof.** For any $c \in \mathbb{R}^{M_t}$, $c^T A^G c = \sum_{l=1}^G \sum_{i,j=1}^{M_t} w_l c_i \psi_i(x_l,y_l) \psi_j(x_l,y_l) c_j$. Let $f_l = \sum_{i=1}^{M_t} c_i \psi_i(x_l,y_l)$. Then $c^T A^G c = \sum_{l=1}^G w_l f_l^2 \geq 0$ since $w_l > 0$. It is clear that the matrix $A^G$ is symmetric and positive semi-definite. To be invertible, it requires that $A^G$ is positive definite or $c^T A^G c = 0$ has only the zero solution $c = 0$. Because $w_l > 0$ for $l = 1, \cdots, G$, it leads to $f_l = 0$ for all $l$, i.e., $\sum_{i=1}^{M_t} \psi_i(x_l,y_l) c_i = 0$. This system has $G$ linear equations and $M_t$ variables ($c_i$). If $G < M_t$, then this system must have free variables and thus nonzero solutions. \qed

Some choices of basis functions and quadrature rules are given in Table 1. In the implementations with $M = 3$ or 4, a 16-point Gaussian quadrature rule on a triangle from [11] is used, which is exact for 8-th degree polynomials. As for the line integral, a 16-point quadrature rule in [7] is used, which is exact for polynomials of degree $\leq 31$. In the implementations with $M = 0, 1,$ or $2$, a 7-point Gaussian quadrature rule on a triangle is used, which is exact for 5-th degree polynomials, and a 3-point quadrature rule is used for the line integral, which is exact for polynomials of degree $\leq 5$. A convergence test of DG scheme is conducted in Appendix 9.7.

| $M$, order of polynomial | 0 | 1 | 2 | 3 | 4 | 5 |
|-------------------------|---|---|---|---|---|---|
| $M_t = (M + 1)(M + 2)/2$, dimension of $P^M(K)$ | 1 | 3 | 6 | 10 | 15 | 21 |
| $G$, minimum number of quadrature points | 1 | 3 | 6 | 10 | 15 | 21 |
3.5 Numerical solution of $\rho(T) = \Lambda^{-2}\theta(T)$

The quantity $\rho(T)$ satisfies $(I - \Delta)\rho(T) = \theta(T)$ and $\frac{\partial \rho(T)}{\partial n} |_{\Gamma} = 0$, whose weak form is

$$\int_{\Omega} (\rho(T)\psi + \nabla \rho(T) \cdot \nabla \psi) dx = \int_{\Omega} \theta(T)\psi dx$$

(3.29)

for each $\psi \in H^1(\Omega)$. We use the standard continuous piecewise linear finite elements on $\Omega_h$ to approximate $\rho(T)$ and $\psi$, i.e., $\rho(T), \psi \in Q_h$. When $\rho(T)$ is used in the backward transport process for $\rho(t)$ which is in the space $W_{h,M}^{DG}$, $\rho(T)$ is projected to this space by $L^2$-projection.

3.6 Gradient descent method with backtracking

Suppose that before the start of the gradient descent step $n$, a control input function $g^n$ and a descent direction $d$ are given. The aim of the step $n$ is to find a new guess $g^{n+1} = g^n + \beta^n d$, $\beta^n > 0$, such that the following sufficient descent condition is satisfied,

$$J(g^n + \beta^n d) \leq J(g^n) + \beta^n \mu(d, \nabla J(g^n)),$$

where $\mu$ is a positive constant. In the spirit of gradient descent method, $d = -\nabla J(g^n)$. Therefore, the above formula becomes

$$J(g^n - \beta^n \nabla J(g^n)) \leq J(g^n) - \beta^n \mu |\nabla J(g^n)|^2,$$

(3.30)

The backtracking technique (e.g., see [13]) is an iteration process to find the value of $\beta^n$ such that it is the first value in the sequence

$$\left\{ \beta^n_i = \frac{\epsilon^n_b}{2^i}, i = 0, 1, \ldots \right\}$$

(3.31)

that satisfies the condition [3.30], where $\epsilon^n_b$ is a positive parameter. This inner iteration produces a sequence $\{\beta^n_i, i = 0, 1, \ldots\}$ and the stopping criterion is given by

1. $J(g^n - \beta^n_i \nabla J(g^n)) \leq J(g^n) - \beta^n_i \mu |\nabla J(g^n)|^2$ or $i > \text{back\_MAX},$

or

3. $\frac{||\beta^n_i d||_{L^2(0,T;L^2(\Gamma))}}{||g^n||_{L^2(0,T;L^2(\Gamma))}} \leq \epsilon_0$ and $\frac{|J(g^n + \beta^n_i d) - J(g^n)|}{J(g^n)} \leq \epsilon_0,$

(3.32)

where back\_MAX is the maximum iterations for the backtracking and the logic operation in the second line checks whether the relative change in the values of $g$ and that of the corresponding costs are sufficiently small.

For the outer iterations, the stopping criterion is set as

1. $\frac{||\nabla J(g^{n+1})||}{1 + J(g^{n+1})} \leq \epsilon_0$ or $n > \text{grad\_MAX},$

or

3. $\frac{||g^{n+1} - g^n||_{L^2(0,T;L^2(\Gamma))}}{||g^n||_{L^2(0,T;L^2(\Gamma))}} \leq \epsilon_0$ and $\frac{|J(g^{n+1}) - J(g^n)|}{J(g^n)} \leq \epsilon_0.$

(3.33)

where $\epsilon_0$ is the same as in [3.32] and grad\_MAX is the maximum outer iterations.

After each outer iteration, the step length $\beta^n$ is doubled to pass to the next iteration as the starting step length in [3.31], i.e., $\epsilon_{b}^{n+1} = 2\beta^n$. There are two benefits compared with a uniform $\epsilon_b$ value for all the outer iterations. First, it saves time when the step lengths remain close between
two successive outer iterations. Second, the doubling of the step length accounts for the possible larger step length allowed in the next outer iteration, thus a larger descent of the cost. For example, if the step length at step \( n \) turns out \( \beta^n = 1e-3 \), then the starting step length at step \( n + 1 \) will be \( \epsilon_b^{n+1} = 2e-3 \). In contrast, if the uniform value \( \epsilon_b = 1 \) for all outer iterations is used and the maximum step length in the next step is 1.5e-3, then it would take many more inner iterations to get down to it.

Moreover, the update (3.1) with (3.6) yields

\[
\alpha_i^{n+1} = \alpha_i^n - \beta^n (\nabla J(g^n))_i, \quad i = 1, \cdots, M.
\]

The basic algorithm with a fixed mesh size \( h \) is summarized as follows.

**[Basic Algorithm with a fixed mesh size \( h \)]**

1. Initialization Step

   (a) Set the final time \( T \), time step size \( \Delta t \), initial value of \( \theta \), gradient descent maximum iteration number \( \text{grad}_\text{MAX} \), backtracking maximum iteration number \( \text{back}_\text{MAX} \), stopping tolerance \( \epsilon_0 \), backtracking parameter \( \mu \) and \( \epsilon_b^0 \), basis functions \( g_i, i = 1, \cdots, M \), and solve for velocity basis \( v_i \) from \( g_i \) and store them for future use. Set the initial control as

   \[
g^0 = \sum_{i=1}^{M} \alpha_i^0 g_i.
\]

   (b) Compose the velocity field

   \[
v^0 = \sum_{i=1}^{M} \alpha_i^0 v_i.
\]

   (c) Evolve \( \theta^0 \) by solving (2.1) with velocity \( v^0 \) from time \( t = 0 \) to \( T \).

   (d) Compute \( \rho^0(T) = \Lambda^{-2} \theta^0(T) \).

   (e) Compute the cost \( J(g^0) \) from (2.13).

   (f) Compute the gradient \( \nabla J(g^0) \) with VF or AD method.

   (g) \( \text{err}_3 = \frac{||\nabla J(g^0)||}{1 + J(g^0)} \).

2. Outer iterations (Gradient Descent).

   Set \( \text{err}_1 = \text{err}_2 = 10^{10} \) and \( n = 0 \).

   While \( (n < \text{grad}_\text{MAX} \text{ and } (\text{err}_1 > \epsilon_0 \text{ or } \text{err}_2 > \epsilon_0) \text{ and } (\text{err}_3 > \epsilon_0) ) \) do

   (a) Set \( \text{Jcost} = 10^{10} \) and \( \beta^n = 2\epsilon_b^n \).

   (b) Inner iterations (Backtracking).

   Set \( \text{iback} = 0 \).

   While ( \( \text{iback} < \text{back}_\text{MAX} \text{ and } (\text{Jcost} \geq J(g^n) - \beta^n \mu ||\nabla J(g^n)||^2) \text{ and } (\text{err}_1 > \epsilon_0 \text{ or } \text{err}_2 > \epsilon_0) \) ) do

   i. \( \beta^n = \beta^n / 2 \).

   ii. \( \alpha_i = \alpha_i^n - \beta^n (\nabla J(g^n))_i, \quad i = 1, \cdots, M \).

   iii. \( g = \sum_{i=1}^{M} \alpha_i g_i \).

   iv. \( v = \sum_{i=1}^{M} \alpha_i v_i \).

   v. Evolve \( \theta \) by solving (2.1) with velocity \( v \) from time \( t = 0 \) to \( T \).

   vi. Compute \( \rho(T) = \Lambda^{-2} \theta(T) \).

   vii. Compute the cost \( J(g) \) from (2.13).

   viii. \( \text{Jcost} = J(g) \).

   ix. \( \text{err}_1 = \frac{||g - g^n||_{L^2(0,T;L^2(\Gamma))}}{||g^n||_{L^2(0,T;L^2(\Gamma))}} \).
\[
\text{x. } err_2 = \frac{|J(g) - J(g^n)|}{J(g^n)}.
\]

\text{xi. back = back + 1.}

Enddo

(c) \alpha_i^{n+1} = \alpha_i, \text{ } i = 1, \ldots, M, \text{ } g^{n+1} = g, \text{ } J(g^{n+1}) = J_{\text{cost}}.

(d) Compute the gradient \(\nabla J(g^{n+1})\) with VF or AD method.

(e) \(err_3 = \|\nabla J(g^{n+1})\|/J(g^{n+1})\).

(f) \(\epsilon^{n+1} = 2\beta^n\).

(g) \(n = n + 1\).

Enddo

Numerical practices show that if the above basic algorithm with a fine mesh uses the solution from a coarse mesh as the initial guess, then efficiency would be increased. Therefore, the following relay scheme emerges.

[Relay Algorithm with refined meshes]

1. Run basic algorithm with initial guess \(g^0\) with mesh size \(h_1\). Denote the solution as \(g^{h_1}\).

2. Run basic algorithm with initial guess \(g^{h_1}\) with mesh size \(h_2 < h_1\).

3.7 A simple check of the numerical code for the solution of \(v, \theta\) and \(\rho\)

We make use of Corollary 2.3 to check the code that solves the velocity \(v\) from the Stokes equations from given controls, evolves \(\theta\) with \(v\) from \(t = 0\) to \(t = T\), computes \(\rho(T) = \Lambda^{-2}\theta(T)\), and tranports \(\rho\) backward with \(v\) from \(t = T\) to \(t = 0\). We set \(\theta_0 = \sin(2\pi y)\) and choose control \(g = 10 \cos(2\omega)\tau\) when \(t \in [0, 0.5]\) and \(g = 20 \sin(2\omega)\) when \(t \in [0.5, 1]\). The velocity \(v\) is computed using the projection scheme and \(\theta\) and \(\rho\) are solved by DGP2 (M=2) method. The test results are shown in Figure 1, where

\[
\text{Mean}_T = \frac{1}{T} \int_0^T \int_\Omega \rho(x,t)\theta(x,t)dxdt. \tag{3.34}
\]

In this test, \(T = 1\). The maximum error of \((\int_0^T \int_\Omega \rho^T(x,t)\theta(x,t) dx - \text{Mean}_T)\) over \(t \in [0, 1]\) is 1.05e-4 when \(h = 0.1, 3.15e-5\) when \(h = 0.05\), and 8.70e-6 when \(h = 0.025\), which shows roughly second order convergence to zero when the mesh is refined. This partially verifies the code.

4 Two forward mixing simulations

To examine the effectiveness of mixing in the Stokes flow with boundary inputs, we first conduct two simulations of the governing system (2.1)–(2.5) based on the numerical schemes described above without optimization strategy. The domain is chosen as the two dimensional unit disk, i.e., \(\Omega = \{(x,y): x^2 + y^2 < 1\}\), and the friction coefficient \(k\) in the Navier slip boundary condition (2.4) is chosen to be \(k = 0.5\). These two setups are the same for all the simulations in this work.

In the first simulation, the flow is induced through the Navier slip boundary conditions (2.4) with the tangential input \(g = \cos(\omega)\tau\). Here \(\omega\) is the polar angle of the point \((x,y)\), that is,
Figure 1: A test for Corollary 2.3 \( \int_{\Omega} \rho T(x,t) \theta T(x,t) dx - \text{Mean}_T \) over time. Initial value \( \theta_0 = \sin(2\pi y) \) if \( t \in [0, 0.5] \) and \( g = 10 \cos(2\omega) \) if \( t \in [0.5, 1] \). \( \text{Mean}_T \) is the mean value in time defined in (3.34).

\[ \omega = \arccos(x/r) \text{ when } y \geq 0 \text{ and } 2\pi - \arccos(x/r) \text{ when } y < 0. \]

Numerical results for \( v_0 = 0 \) and \( \theta_0(x,y) = 1 \) for \( y \geq 0 \) and \( \theta_0(x,y) = -1 \) for \( y < 0 \) over the time interval \([0, 100]\) are shown in Figure 2 where all the figures of \( \theta \) are on the mesh \( h = 0.025 \). It is observed that there are two vortices generated in the steady state velocity field, which are vertically across the density interface. The mix-norm decays approximately with a power rule \( O(t^{-1}) \) as indicated in Figure 6[f]. Since \( g(x,0) \neq 0 \), the compatibility condition for the initial and boundary data does not hold. Thus the velocity field satisfies \( v \in H^{5/4-\varepsilon/2}/(0, 100; H^{5/2-\varepsilon}(\Omega)) \subset L^\infty(0, 100; H^{5/2-\varepsilon}(\Omega)) \) for \( \varepsilon > 0 \) sufficiently small. There is an extensive literature on the relation between the mixing decay rate in time and the regularity of the flow velocity. However, it remains open whether the exponential mixing decay rate can be achieved for any initial \( \theta_0 \in L^\infty(\Omega) \) in such a smooth flow (for \( v \in W^{s,p}(\Omega) \) with \( s > 2 \) and \( 1 \leq p \leq \infty \), uniformly in time). A seemingly exponential decay was observed in the numerical experiments conducted in [32], where the velocity is assumed to be the form of the product of sine and cosine functions in space and time. More detailed mathematical analysis regarding this issue can be found in [1, 31, 5, 8, 40, 24, 36] and the references cited therein. On the other hand, the gradient of the scalar field increases as the mixing scale decreases. Since thinner filaments of the scalar distribution will develop as time evolves, the mesh size must be refined sufficiently to capture the smallest spatial scales for simulations spanning over longer time intervals.

The kinetic energy, \( \frac{1}{2} \int_{\Omega} |v(x,t)|^2 dx \), is plotted for the velocity field \( v \) generated by \( g = \cos(\omega) \tau \) in Figure 3. Clearly it is not a constant in time but it approaches a constant value after \( t = 0.4 \), because the velocity field converges to a steady state. This is in contrast to the work in [32] where the kinetic energy is enforced to be invariant in time.

However, if the boundary input is not appropriately chosen, it may be not effective for mixing at all. The second simulation uses \( g = \sin(\omega) \tau \). As shown in Figure 4, this force generates two vortices parallel to the density interface and the two colors are barely mixed. As studied in [3, 39], where two viscous immiscible fluids in Stokes flows are considered and the stretching and deformation of the interface is used to qualify mixing, the velocity must be directed across the material interface in order to attain the maximum interface length.
Figure 2: [a]: steady state velocity field induced by the boundary input \( g = \cos(\omega) \tau \). The maximum magnitude is about 0.4. [b, c, d, e]: distribution of \( \theta \) at \( t = 0, 20, 50, 100 \). [f]: \((H^1(\Omega))'\) norm of \( \theta \) over time.

Figure 3: Kinetic energy evolution for \( g = \cos(\omega) \tau \).
5 Convergence study of the optimization algorithm when basis $g_1 = \cos(\omega)\tau$ and $\theta_0 = \tanh(y/0.1)$

In all the rest numerical implementations in this paper, the final time is set as $T = 1$. In this section, we use several examples with only one control basis function to demonstrate the convergence of our numerical schemes. At the same time, we point out some important features of numerical methods for mixing problems. In these experiments, we set the initial distribution of the scalar field as $\theta_0(x, y) = \tanh(y/0.1)$, a smoothed step function as plotted in Figure 6[a], which satisfies the regularity required in Theorem 2.2. The control input is simply spanned by one time-independent basis function $g_1 = \cos(\omega)\tau$, where $\omega$ is the polar angle of the point $(x, y)$. In most implementations in this work, three meshes are used to discretize this domain with mesh sizes $h = 0.1, 0.05,$ and $0.025$. Moreover, the discontinuous Galerkin method uses the P2 elements and RK3 time discretization.

5.1 Cost when $g = \alpha_1 \cos(\omega)\tau$, $\alpha_1 \in [0, 90]\]

Before applying the optimization scheme, we solve $\theta$ from $t = 0$ to $t = 1$ using the control functions $g = \alpha_1 \cos(\omega)\tau$, $\alpha_1 \in [0, 90]$ on the three meshes mentioned above and compute the costs for two different values of $\gamma = 1e-3$ and $1e-6$. The costs are only computed at the integer values of $\alpha_1$ to save time and the results are shown in Figure 5. When $\gamma$ is as large as $1e-3$, the costs computed with the three meshes are almost identical for all these control values and have a unique global minimizer in the interval $\alpha_1 \in [0, 10]$. The minimized scalar $\theta$ obtained with the numerical scheme when $\gamma = 1e-3$ and $h = 0.025$ is shown in Figure 6[b].

When $\gamma$ is reduced to $1e-6$, the costs reported from different meshes are indistinguishable when $\alpha_1 \in [0, 20]$, but start to depart from each other when $\alpha_1 > 40$, and become drastically different when $\alpha > 70$. Also, multiple local minimizers of the cost functional are observed even with the most refined mesh size when $\gamma = 1e-6$. The different costs of different meshes for large $\alpha$ values is due to the fact that the coarse meshes are unable to capture the thin filaments of the highly stretched scalar field.
Figure 5: [a]: The costs at time $t = 1$ over coefficient $\alpha_1$ of $g = \alpha_1 \cos(\omega) \tau$ for different value of $\gamma$ and meshes. [b]: zoom-in of [a] for $\gamma = 1e-6$ and $\alpha_1 \in [40, 90]$.

Contours for large $\alpha_1$ values, thus producing inaccurate costs.

### 5.2 Convergence of optimization algorithm when $\gamma = 1e-3$

When $\gamma = 1e-3$, this optimization problem has a unique global minimizer for $\alpha_1 \in [0, 10]$ based on the observation of the cost values in Figure 5[a], which also indicates that the numerical solutions under different mesh sizes $h = 0.1, 0.05, 0.025$ should be similar. We run the basic optimization algorithm with the following parameter values: the stopping tolerance $\epsilon_0 = 10^{-5}$, $\mu = 0.3$ and back MAX=10, and $\epsilon^b_0 = 1e3$. The initial guess of the control function is chosen as $g^0 = \cos(\phi) \tau$, that is, $\alpha^0_1 = 1$.

Figure 6: [a]: initial value $\theta_0$. [b]: optimal numerical solution $\theta$ at $t = 1$ when $\gamma = 1e-3$. The variational formula (VF) is used to compute the cost gradient. Both [a] and [b] use the mesh size $h = 0.025$.

Table 2 shows the computational results of $\nabla J$ using Variational Formula (VF) and Algorithmic Differentiation (AD), with the same control input function. Under different meshes, the values of $\nabla J$ computed by the AD method are about the same, which could be regarded as the reliable approximations of the gradient. When the meshes are refined from $h = 0.1$ to 0.05 and then to 0.025, the errors of gradients between the two methods are roughly reduced by one half, indicating the first order accuracy of the VF method compared with the AD method in the computation of the gradient.

As for the optimization simulations, the comparison of the sequence of the control parameter $\alpha_1$ and the corresponding costs is plotted in Figure 7, which shows that the solutions using the VF method converge to the same limit as the AD method. The optimal numerical state of $\theta$ is shown in Figure 6[b].
Table 2: Computations of $\nabla J$ with Variational Formula (VF) and Algorithmic Differentiation (AD) when $g = \alpha \cos(\phi)\tau$, $\gamma=1e-3$, and initial $\theta = \tanh(y/0.1)$. The error is computed as $|VF - AD|/AD$.

| Method | $h=0.1$ | $h=0.05$ | $h=0.025$ |
|--------|---------|----------|-----------|
| VF     | $\nabla J = -8.80205e-03$ | $\nabla J = -8.42116e-03$ | $\nabla J = -8.17538e-03$ |
| AD     | $\nabla J = -7.82339e-03$ | $\nabla J = -7.84103e-03$ | $\nabla J = -7.84398e-03$ |
| Error  | 13.2%   | 7.4%     | 4.2%      |

Figure 7: Numerical results when $\gamma = 1e-3$. [a]: sequence of $\alpha^i_1$ in $g = \alpha^i_1 \cos(\phi)\tau$ vs iteration. The curves of $\{\alpha^i_1\}$ solved by using the VF method under three meshes are indistinguishable. When $h = 0.025$, the end-of-iteration values of $\alpha_1$ are 5.46045 for the AD method and 5.47040 for the (VF) method. [b]: cost vs iteration.

Table 3 provides the overall backtracking numbers and the CPU runtime of both methods under three meshes. For each method, when the mesh size is halved, the CPU runtime is roughly increased eightfold. This is because the mesh points are quadrupled and the explicit evolution further halves the time step size, which gives eight folds. This test indicates that the computational cost of the AD method would be lower than the VF method when the dimension of the control inputs is very small. However, this is not the case when the number of control basis functions increases as in section 6.

Table 3: Iterations and CPU runtime using Variational Formula (VF) and Algorithmic Differentiation (AD). Here, RK3 and 7 Gaussian quadrature points are used.

| Method                | $h=0.1$       | $h=0.05$      | $h=0.025$     |
|-----------------------|---------------|---------------|---------------|
| VA: backtracings      | 20            | 20            | 22            |
| VA: CPU runtime (h:m:s)| 00:04:18     | 00:30:23      | 04:55:56      |
| AD: backtracings      | 12            | 10            | 10            |
| AD: CPU runtime (h:m:s)| 00:03:57     | 00:19:12      | 02:51:25      |

5.3 Convergence of optimization algorithm when $\gamma = 1e-6$

When $\gamma = 1e-6$, the cost curves in Figure 5 show a big difference among the local minimizers under different mesh resolutions when the control weight $\alpha_1 > 50$. We run the simulations using variational formula and the initial value $\alpha_1^0 = 70$. All of other setups are the same as in the case of $\gamma=1e-3$ in section 5.2. The values of $\alpha_1$ reported from the numerical optimization are about 62.56, 70.56, and 70.51, respectively, when $h = 0.1$, 0.05 and 0.025, coinciding with the critical points of the cost curves in Figure 5[b]. This demonstrates the convergence of this numerical method. The CPU time is 53:31 (minute:second) for $h = 0.1$, 1:36:33 (hour:minute:second) for $h = 0.05$, and 19:12:18 for $h = 0.025$. The velocity field reaches a steady state at $t = 0.5$ in each of three simulations, and shows
a strong rotation pattern as illustrated in Figure 8[a]. The corresponding scalar fields are rendered in Figure 8[bcd]. Clearly, when the mesh size $h = 0.1$ and $0.05$, the regions of $\theta = 1$ (yellow color) are broken and even fragmented in some places compared with that when $h = 0.025$. The scalar field is colored on top of meshes in Figure 9, which shows that the thin yellow filaments cannot be seized by the coarse mesh with $h = 0.05$ and thus diffuse away, but can be well-captured by the fined mesh with $h = 0.025$. Moreover, Figure 9[f] suggests that the mixing decay rate in time obeys a power law as $O(t^{-1})$, when $t$ approaches to the final time $T = 1$ window.

Figure 8: [a]: Optimal numerical solution of $v$ at $t = 1$ under mesh $h = 0.025$. [b,c,d]: optimal numerical solutions of $\theta$ at $t = 1$ under meshes $h = 0.1$, $h = 0.05$, and $h = 0.025$, respectively. [f]: $(H^1(\Omega))'$-norm of $\theta$ over time.

Figure 9: Mesh and scalar field plotted together at $t = 1$ under meshes $h = 0.05$[e] and $h = 0.025$[f].

Overall, the proposed numerical method using the variational formula for computing $\nabla J$ is convergent when the mesh size is refined. From this convergence study we realize an appalling challenge of the numerical methods for solving mixing problem: the mesh size must be refined sufficiently to capture the fine spatial scales of the thin filaments of the scalar distribution. When the filament thickness tends to infinitely small, it would be impossible to take hold of it with any finite meshes.
6 Optimization with control basis $\cos(2\omega)$ and $\sin(2\omega)$ for $\theta_0 = \sin(2\pi y)$

In this section, we utilize the piecewise time-segmented functions based on two boundary inputs, $\cos(2\omega)$ and $\sin(2\omega)$, to construct the control functions. The mixing with these two basis functions is called ‘cos-sin’ case. These two functions are chosen to generate the velocity fields resembling those used in [32]. More precisely, consider

$$g = \alpha_1(t) \cos(2\omega)\tau + \alpha_2(t) \sin(2\omega)\tau,$$

$$\alpha_i(t) = \sum_{j=1}^{N} \alpha_{ij} \chi((i-1)\Delta s, i\Delta s)(t),$$

for $\alpha_{ij} \in \mathbb{R}, i = 1, 2; j = 1, 2, \ldots, N$. Here $N$ is number of time segments, $\Delta s = \frac{1}{N}$ is the segment size, and

$$\chi((i-1)\Delta s, i\Delta s)(t) = \begin{cases} 1, & \text{if } t \in ((i-1)\Delta s, i\Delta s), \\ 0, & \text{otherwise}, \end{cases}$$

is the characteristic function. For convenience, we let $\vec{\alpha} = (\alpha_{11}, \ldots, \alpha_{1N}, \alpha_{21}, \ldots, \alpha_{2N})^T \in \mathbb{R}^{2N}$. The velocity fields induced by the tangential boundary inputs $\cos(2\omega)\tau$ and $\sin(2\omega)\tau$ are shown in Figure 10 [a] and [b], respectively. The initial distribution of the scalar field $\theta_0 = \sin(2\pi y)$ is shown in Figure 10 [c], which is the same initial density considered in [32]. The initial backtracking parameter is $\epsilon_0^b = 1e3$.

Set $\gamma = 1e-6$. We run the optimization algorithm when $N = 1, 2, 5, 10$ and for each value of $N$ we test several initial values of the control parameters. Due to the possible existence of many local minimizers, there is no guarantee that it is the global minimizer.

When $N = 1$, the basis control input functions are simply $g_1 = \cos(2\omega)\tau$ and $g_2 = \sin(2\omega)\tau$, each one applied to the entire time interval $[0, 1]$. The following initial values are adopted: $\vec{\alpha}^0 = (\alpha_1^0, \alpha_2^0) = (40, 40), (50, 40), (0, 10), (40, 50)$, and their corresponding solutions are presented in Table 4. In these tests, the control weight of the sine function $\sin(2\omega)$ outweighs that of the cosine function $\cos(2\omega)$. The scalar $\theta$ related to the smallest cost is shown in Figure 12 [a].

Second, set $N = 2$. The scalar field of $\theta$ is plotted in Figure 12 [b]. In particular, we compared the performance between the VA and AD methods for the calculation of gradient when the initial control weight is $\vec{\alpha}^0 = (40, 40, 40, 40)$ and it turns out that the CPU time of VA is 8-20:48:48 (day:hour:minute:second) and that of AD is 29-11:44:09 (day:hour:minute:second). Although the
Table 4: cos-sin case: N=1 in (6.2)

| Initial value of \( \vec{\alpha} \) | Ending value of \( \vec{\alpha} \) | mix-norm | g-norm | cost   |
|--------------------------------------|-----------------------------------|----------|--------|--------|
| (40,40)                              | (1.86e+01, 4.34e+01)             | 1.16e-01 | 8.36e+01 | 1.02e-02 |
| (50,40)                              | (4.97e+00, 5.18e+01)             | 1.04e-01 | 9.22e+01 | 9.69e-03 |
| (0,10)                               | (6.06e-08, 7.15e+00)             | 2.07e-01 | 1.27e+01 | 2.14e-02 |
| (40,50)                              | (6.88e+00, 5.12e+01)             | 1.05e-01 | 9.17e+01 | 9.71e-03 |

Table 5: cos-sin case: N=2 in (6.2). VF: Variational Formula. AD: Algorithmic Differentiation.

| Initial value of \( \vec{\alpha} \) | Ending value of \( \vec{\alpha} \) | CPU time | mix-norm | g-norm | cost   |
|--------------------------------------|-----------------------------------|----------|----------|--------|--------|
| (40,40,40,40) \( \text{VF} \)       | (1.26e+01, 1.20e+01, 2.98e+01, 4.16e+01) | 8-20:48:48 | 1.19e-01 | 6.88e+01 | 9.44e-03 |
| (40,40,40,40) \( \text{AD} \)       | (6.08e+00, 2.64e+00, 2.97e+01, 3.92e+01) | 29-11:44:09 | 1.19e-01 | 6.31e+01 | 9.10e-03 |
| (50,30,50,30) \( \text{VF} \)       | (9.52e+00, 6.86e+00, 4.74e+01, 5.37e+01) |          | 1.02e-01 | 9.22e+01 | 9.47e-03 |
| (30,60,30,60) \( \text{VF} \)       | (8.17e+00, 6.29e+01, 2.22e+01, 5.33e+01) |          | 8.29e-02 | 1.09e+02 | 9.33e-03 |

For \( N=10 \), we used the relay method, that is, first we ran the simulations with \( h = 0.05 \), then set their solution of \( \vec{\alpha} \) as the initial values for simulations with \( h = 0.025 \). In the four simulations presented in Table 7, the first one took about 13 days to finish and all others took 4 days or less.

Table 6: cos-sin case: N=5 in (6.2). VF: Variational Formula method. AD: Algorithmic Differentiation method. CPU time: day-hour:minute:second. ‘N/A’ means the data is lost due to storage issues.

| Initial value of \( \vec{\alpha} \) | Ending value of \( \vec{\alpha} \) | CPU time | mix-norm | g-norm | cost   |
|--------------------------------------|-----------------------------------|----------|----------|--------|--------|
| (40,40,40,40,40,40,40,40,40,40) \( \text{VF} \) | (1.69e+01, 2.90e+01, 2.45e+01, 2.52e+01, 1.99e+01, 1.71e+00, 4.70e+01, 3.04e+01, 3.70e+01, 2.27e+01) | 10-23:32:54 | 7.97e-02 | 7.39e+01 | 5.91e-03 |
| (40,40,40,40,40,40,40,40,40,40) \( \text{AD} \) | (2.24e+01, 2.89e+01, 2.90e+01, 3.08e+01, 3.32e+01, 3.00e+00, 4.86e+01, 4.27e+01, 3.98e+01, 2.81e+01) | 29-11:10:20 | 7.03e-02 | 8.72e+01 | 6.28e-03 |
| (50,0,50,0,50,0,50,0,50,0,50) \( \text{VF} \) | (5.41e+01, 2.44e+01, 2.89e+01, 3.05e+01, 1.99e+01, -5.52e+01, 7.66e+01, -1.23e+01, -7.74e+00, -1.48e+00) |          | 4.92e-02 | 9.78e+01 | 6.00e-03 |
| (50,50,50,-50,50,-50,-50,-50,50,-50) \( \text{VF} \) | (0.67e+01, 5.15e+01, 3.38e+01, 4.42e+01, 3.18e+01, -3.74e+01, -4.50e+01, -3.21e+01, -3.03e+01, -2.71e+01) | N/A      | N/A      | N/A    | 7.80e-03 |

Note from the above tables, the solutions obtained are very different when the initial values of \( \alpha \) are different, which suggest the existence of multiple local minimizers. This is true for all the examples in this and the next sections, where the dimensions of the control bases are more than one. The optimal control coefficients \( \vec{\alpha} \) of these four cases of \( N \) (the starred data in Tables 4, 5, 6, 7) are plotted in Figure 11. To compare the effects of cosine and sine functions in these cases, we compute the scaled \( L^2 \)-norm as follows.

\[
\beta_{\text{cos}} = \sqrt{\sum_{j=1}^{N} \alpha_{1j}^2 \Delta s}, \quad \beta_{\text{sin}} = \sqrt{\sum_{j=1}^{N} \alpha_{2j}^2 \Delta s}.
\]
Table 7: cos-sin case: N=10 in (6.2). CPU time: day-hour:minute:second. Different from Table 4, 5, 6, these simulations listed below use relay method and VF for gradient.

| Initial value of $\vec{\alpha}$ | Ending value of $\vec{\alpha}$ | CPU time | mix-norm | g-norm | cost |
|----------------------------------|---------------------------------|----------|-----------|--------|------|
| $\alpha^0_{ij} = 40, i = 1, 2; j =$ | $3.58e+01, 2.76e+01, 2.65e+01, 3.28e+01,$ | 13-04:31:02 | 6.79e-02 | 8.17e+01 | 5.65e-03 |
| $\alpha^0_{ij} = 50, i = 1, 2; j =$ | $3.83e+01, 3.50e+01, 1.45e+01, 3.21e+01,$ | 4-00:01:47 | 6.46e-02 | 8.55e+01 | 5.75e-03 |
| $\alpha^0_{ij} = 40,$ | $2.72e+01, 2.44e+01, 2.29e+01, 3.19e+01,$ | 1-11:10:55 | 6.93e-02 | 8.13e+01 | 5.75e-03 |
| $\alpha^0_{ij} = 30,$ | $3.91e+01, 2.44e+01, 1.67e+01, 3.93e+01,$ | 3-15:59:37 | 6.71e-02 | 8.30e+01 | 5.70e-03 |

These norms are computed for these optimal controls and the results are shown in Table 8. Observe that the weight of sine function is always larger than that of the cosine function. However, their weights get closer when the time intervals are more divided.

The end-of-the-time density fields of optimal controls are illustrated in Figure 12. It is noted that when $N$ increases, the optimal cost is lowered (Figure 11[e]), which suggests the advantage of using more time segments. Furthermore, the kinetic energy of optimal controls is more oscillatory in time when $N$ is larger (Figure 11[f]). This indicates that the velocity field is more irregular. A sequence of snapshots of density field of the optimal control when $N = 10$ is shown in Figure 13. It seems that two separatrices are present and hence, the mixing is only confined into four small regions. Indeed, the similar separatrices exist for all the simulations in this section, as observed in Figure 12[abcd].

The evolutions of the mix-norms of the optimal numerical solutions associated with the cos-sin cases are shown in Figure 12[e]. These indicate again that the mixing decay rate in time follows a power law as $O(t^{-1})$, when $t$ goes to the final time $T = 1$ window and the dimension of $U_{ad}$ increases.

6.1 Impact of different values of $\gamma$

In section 5, we have seen the effects of $\gamma$ on the optimal solution, where the smaller $\gamma$ leads to better mixing effect. Here, one more comparison is presented in the case of $N = 1$ in (6.1–6.2) where the initial value is $\vec{\alpha} = (50, 40)$, with which the optimal solution when $\gamma = 1e-6$ is obtained as shown in Table 4. Table 9 illustrates that when $\gamma$ is decreased from 1e-6 to 1e-8, the solution $\vec{\alpha}$ becomes larger in both entries but more significant in the second entry. This leads to the reduced mix-norm
Figure 11: Optimal control weight $\vec{\alpha}$ in (6.2). [a]: $N = 1$, cost=$9.69\times10^{-3}$. [b]: $N = 2$, cost=$9.10\times10^{-3}$. [c]: $N = 5$, cost=$5.91\times10^{-3}$. [d]: $N = 10$, cost=$5.65\times10^{-3}$. [e]: optimal cost versus $N$. [f]: kinetic energy evolutions of optimal controls.

Figure 12: Optimal numerical solutions of $\theta$ using the cos-sin based control design in (6.1)–(6.2). [a]: $N = 1$, cost=$9.69\times10^{-3}$. [b]: $N = 2$, cost=$9.10\times10^{-3}$. [c]: $N = 5$, cost=$5.91\times10^{-3}$. [d]: $N = 10$, cost=$5.65\times10^{-3}$. [e]: mix-norm of $\theta$ over time.
Figure 13: Sequence of density evolution at $t = 0, 0.1, \cdots, 1$ when $N=10$. Initial control parameter is $\alpha_{ij}^0 = 40$, $i = 1, 2; j = 1, 2, \cdots, 10$. This case produces the smallest cost among all simulations in Section 6.

and enlarged $g$-norm, but the overall cost $J$ is decreased. The renderings of the density in Figure 14 confirm the better mixing result when $\gamma = 1e-8$, which is characterized by the thinner and longer foldings.

Table 9: Comparison of the solutions under different $\gamma$ values when $N = 1$ in (6.1)–(6.2) and initial $\bar{\alpha} = (50, 40)$

| $\gamma$  | ending value of $\bar{\alpha}$ | mix-norm | g-norm | cost $J$ |
|-----------|---------------------------------|----------|--------|----------|
| 1e-6      | (4.97, 51.76)                  | 1.04e-01 | 9.22e+01 | 9.69e-03 |
| 1e-8      | (6.47, 76.59)                  | 9.96e-02 | 1.36e+02 | 4.97e-03 |

Figure 14: Numerical optimal solutions of $\theta$ using the cos-sin based control design in (6.1)–(6.2) with $N = 1$. Initial value $\bar{\alpha} = (50, 40)$. [a]: $\gamma = 1e-6$. [b]: $\gamma = 1e-8$.

6.2 Performance study: VF v.s. AD, relay v.s. single resolution

First, we compare the performance of VF and AD methods in the case of $N = 5$ in (6.2) with the initial value $\bar{\alpha}_{ij} = 40$, $i = 1, 2, j = 1, \cdots, 5$ and mesh size $h = 0.025$. The information of simulation performance is illustrated in Figure 15 and the numerical solutions or ending values of $\bar{\alpha}_{ij}$ are shown in Table 6 (Row 1 and Row 2). The VF scheme uses 133 outer iterations and 11 days of CPU time, in contrast to AD scheme with 72 outer iterations and 30 days of CPU time. The inner backtracking iterations range from 1 to 6 with an average 1.93 for VF and 1.72 for AD. Although the AD scheme
Let \( J_k \) be the cost at the outer iteration \( k \) and \( J^* \) denote the final cost of the numerical algorithm. Define the error \( e_k = J_k - J^* \). The convergence rate at an iteration step \( k \) is defined as 
\[
  r_k = \frac{\ln |e_{k+2}/e_{k+1}|}{\ln |e_{k+1}/e_k|}.
\]
The cost \( J_k \) is a strictly decreasing function of iteration number \( k \) (Figure 15[ac]), resulting from the sufficient descent condition (3.30). The convergence rates are plotted in Figure 15[bd] and the average rates are a little larger than one for both VF and AD methods. This observation is consistent with the classic theory of gradient descent method, that is, linear convergence. Note that the convergence rates in the first few iterations typically reach 2 or 3, which are much higher than the later and average convergence rates, as shown in Figure 15[bd]. This is consistent with the observation in Figure 16[b] that the initial decrease of the cost is much steeper than later. An investigation of the cost decay split to the mix-norm and g-norm is given in Section 7.

![Figure 15: Performance of the single mesh optimization algorithm with comparisons between VF and AD methods.](image)

The step length \( \epsilon^n \) used in the iteration (3.1) (shown in Figure 15[ac]) is of the magnitude 1e4. The three errors in the algorithms are 
\[
  err_1 = \frac{|g^{k+1} - g^k|_{L^2(0,T;L^2(\Omega))}}{|g^0|_{L^2(0,T;L^2(\Omega))}},
  err_2 = \frac{|J(g^{k+1}) - J(g^k)|}{J(g^k)},
  err_3 = \frac{||\nabla J(g^{k+1})||_1+J(g^k)}{|1+J(g^k)|}
\]
at each outer iteration \( k \). These errors oscillate all the time but decrease in the long run. The numerical algorithm stops when \( err_3 \) reaches the threshold value 1e-5, which implies the norm of cost gradient is sufficiently small. This fulfills the necessary condition of being a local minimizer.

The results correspond to Row 1 and Row 2 of Table 6. The computation service at Institute for Cyber-Enabled Research at Michigan State University requires the simulation to stop after every seven days of running. Thus a long simulation has to be re-submitted several times.

Next, we examine the performance of the relay algorithm on the optimization process through the case of \( N = 5 \) with initial value \( \bar{\alpha}_{ij} = 40, i = 1, 2, j = 1, \cdots, 5 \) under two meshes: \( h = 0.05 \) and 0.025. The whole relay process includes two stages: the first stage runs the simulation with initial...
$\vec{\alpha}_{ij} = 40$ and mesh $h = 0.05$ to the desired accuracy. In the second stage, the ending value of $\vec{\alpha}_{ij}$ of the first stage is set as the initial value for the simulation with $h = 0.025$. Figure [16a] shows the information of both stages. When $h = 0.05$, it costs 109 outer iterations and 1 day and 5 hours. The simulation $h = 0.025$ only takes 9 outer iterations and 11 hours. So this entire relay operation only uses 118 iterations and 1 day and 16 hours of CPU time, far shorter than the single-resolution simulation starting from the initial value $\vec{\alpha}_{ij} = 40$ with $h = 0.025$ (11 days of CPU time in VF scheme and 30 days of CPU time in AD scheme as mentioned above). Figure [16b] shows the cost in the whole relay process with a jump at the transition from $h = 0.05$ to $h = 0.025$, which is produced by the more accurate computation of the cost in the refined mesh for the same control. The average convergence rate when $h = 0.05$ is 1.15 and 1.47 when $h = 0.025$, which shows the acceleration in the finer mesh. The renderings of the density at the end of two stages are given in Figure [16cd], which indicates that the coarse mesh already provides a pretty decent but a little rough approximation to the final solution and the finer mesh polishes it to the final solution.

Figure 16: Performance of relay algorithm in the case of $N = 5$ with initial value $\vec{\alpha}_{ij} = 40$, $i = 1, 2, j = 1, \cdots, 5$ under two meshes: $h = 0.05$ and 0.025. The rendering of the density at the end of $h = 0.05$ simulation is plotted in [c] and that at the end of $h = 0.025$ simulation in [d].

7 Optimization with more boundary control basis functions for $\theta_0 = \sin(2\pi y)$

Note that the basis control functions $\sin(2\omega)\tau$ and $\cos(2\omega)\tau$ generate an array of spatial periodic vortices as shown in Figure [10]. Consequently, the fixed points (domain centers) and regions of regular motion divided by separatrices appear in the density evolution, which limit the physical points from reaching everywhere of the flow domain. To enhance the mixing efficiency, we add more control options to possibly eliminate regions of regular motion. Applying aperiodic stirring protocols to avoid the flow periodicity was also discussed in [14, 29].

Consider two more combined cases: the first case uses three functions $\cos(2\omega)$, $\sin(2\omega)$, and 1, and the second case uses five functions $\cos(2\omega)$, $\sin(2\omega)$, 1, $\cos(\omega)$, $\sin(\omega)$. For convenience, we name
the first case “COMB1” and the second one “COMB2”. More precisely, the control functions are

\[
\begin{align*}
g_{\text{COMB1}} &= \alpha_1(t) \cos(2\omega \tau) + \alpha_2(t) \sin(2\omega \tau) + \alpha_3(t) \tau, \\
g_{\text{COMB2}} &= \alpha_1(t) \cos(2\omega \tau) + \alpha_2(t) \sin(2\omega \tau) + \alpha_3(t) \tau + \alpha_4(t) \cos(\omega \tau) + \alpha_5(t) \sin(\omega \tau).
\end{align*}
\] (7.1)

(7.2)

where \(\alpha_i(t), i = 1, \ldots, 5\), are defined in (6.2). The time domain \([0, 1]\) is divided into \(N = 5\) and 10 subintervals.

The cost parameter \(\gamma = 1e-6\) and all the simulations use the relay algorithm with VF method for gradient. The initial value of control coefficient \(\alpha^0\) in each simulation is taken as an alternating sequence of \(\pm 80\) (or \(\pm 60\), or \(\pm 40\)). That is, \(\alpha^0 = (-80, 80, -80, \cdots), (-60, 60, -60, \cdots), (\pm 40, 40, -40, \cdots)\). All the simulations have taken two to three months CPU time to achieve the results reported in this work.

Table 10 shows the terminal mix-norms, g-norms, and costs of all the simulations. The COMB2 simulations achieve smaller costs than the COMB1 simulations with the same \(N\) values and initial control paramters (with the exception in the case with \(N = 5\) and \(\alpha^0 = \pm 60\) in COMB2). This demonstrates the advantage of applying more diverse control functions. Note also in all the combinations of COMB1, the minimum cost when \(N = 10\) is less than that when \(N = 5\). The same observation carries to COMB2. Thus, it implies the merit of using more time segments.

Table 10: Terminal values of mix-norms, g-norm, and cost in COMB1 and COMB2. The red color highlights the minimum mix-norm, g-norm, and cost.

| \(N\) | \(\alpha^0\) | COMB1 mix-norm | COMB1 g-norm | COMB1 cost | COMB2 mix-norm | COMB2 g-norm | COMB2 cost |
|------|-------------|----------------|---------------|------------|----------------|---------------|------------|
| 10 | \(\pm 80\) | 7.68e-02 | 2.16e+02 | 2.63e-02 | 6.69e-02 | 1.70e+02 | 1.68e-02 |
| 10 | \(\pm 60\) | 9.08e-02 | 1.78e+02 | 2.00e-02 | 5.54e-02 | 1.31e+02 | 1.02e-02 |
| 10 | \(\pm 40\) | 6.63e-02 | 7.88e+01 | 5.31e-03 | 5.47e-02 | 1.98e+02 | 1.14e-03 |
| 5 | \(\pm 80\) | 7.28e-02 | 2.01e+02 | 2.30e-02 | 5.00e-02* | 1.98e+02 | 2.09e-02 |
| 5 | \(\pm 60\) | 6.06e-02 | 1.43e+02 | 1.20e-02 | 5.18e-02 | 1.81e+02 | 1.78e-02 |
| 5 | \(\pm 40\) | 6.96e-02 | 1.24e+02 | 1.01e-02 | 5.77e-02 | 1.19e+02 | 8.78e-03 |

Figure 17 shows the changes of costs, mix-norms, and g-norms in the optimization processes in both COMB1 and COMB2. Identical to the cost decay pattern seen in Section 6.2, there exist two decay stages: the first is the initial fast decay period that coincides with the rapid drop of mix-norms at the first a few iterations. The second stage is the later long and slow decay period whose cost decay profiles are similar to that of the g-norm. In addition, the mix-norm curves flatten in the second stage in the most optimization processes. Therefore, the cost minimization in the second stage is mainly caused by the reduction of the g-norm. The convergence rate behavior of the cost is similar to that presented in section 6, thus not shown here.

According to Table 10, the COMB2 case with \(N = 10\) and \(\alpha^0 = \pm 40\) leads to the optimal solution from all these simulations, which not only achieves the minimum cost, but also the minimum g-norm. The density and terminal control parameter of this case are shown in Figure 18 and a sequence of snapshot of density evolution is illustrated in Figure 19. The COMB2 case with \(N = 5\) and \(\alpha^0 = \pm 80\)

\(^1\)In this case, the sudden jump of cost may be due to the mistake of copying the data from the previous running result to the next job submission, because HPCC (High Performance Computing Center at Michigan State University) requires each job to drop in seven days and then to be re-submitted if it is not finished.
produces the smallest mix-norm in all the COMB1 and COMB2 cases and the largest g-norm among all the COMB2 cases. The density, terminal control parameter, and kinetic energy evolution are shown in Figure 20 and some images of the density evolution are shown in Figure 21.

Figure 22 shows the mix-norms change over time when the terminal control parameters are used to evolve the density in both COMB1 and COMB2 cases. We observe that the mixing decay rate in time follows a power law as $O(t^{-2})$ when $t$ approaches the final time $T = 1$ and the dimension of $U_{ad}$ increases. Based on the numerical results, it is expected that increasing the frequency of the basis functions may further enhance mixing.
Figure 19: Sequence of density evolution at $t = 0, 0.1, \cdots, 1$ in the case of COMB2 when $N=10$ and $a^0 = \pm 40$. This simulation has the smallest cost among all the cases in COMB1 and COMB2.

Figure 20: The case with the minimum mix-norm among all studies in COMB1 and COMB2 simulations: COMB2 with $N = 5$ and $a^0 = \pm 80$. [a]: density at $t = 1$ with the terminal control parameter. [b]: terminal control parameter. [c]: kinetic energy.

Figure 21: Sequence of density evolution at $t = 0, 0.1, \cdots, 1$ in the case of COMB2 when $N=5$ and $a^0 = \pm 80$. This simulation has the smallest mix-norm at $t = 1$ among all the COMB1 and COMB2 cases.
8 Conclusions

This work is the first numerical study of optimal boundary control for purely advective mixing under unsteady Stokes flows. Built upon the theoretical foundation laid by Hu and Wu [19, 22, 18, 23], a new gradient descent based optimization algorithm is proposed. The numerical results show that the mixing decay rate based on \( (H^1(\Omega))^f \)-norm with the optimal boundary control resembles a power rate as at least \( O(t^{-1}) \) in Stokes flows.

The numerical challenges lie in the following aspects. First, the spatial scale of thin filaments generated in mixing tends to approach infinitely small over time, which poses a significant difficulty to numerical methods. Another numerical challenge is the existence of numerous local minimizers, which can be seen from the example in section 5. As shown in the examples in section 6 and section 7, the optimal numerical solutions are always different if the initial values are different. The above two belong to the inherent physics, while the next challenge is in computation: the storage expense for a long time computation. In the current passive mixing problem, the velocity has no feedback from the density and thus it is generated and stored before the optimization process. This saves the CPU time for the simulations, but increases the storage requirement of hard disk and memory especially for a large time windows or large velocity magnitude. This is because if the velocity magnitude is large with fast change of directions, more time frames of velocity field are needed in order to accurately track the density evolution.

In this work, the numerical method is basically a gradient descent approach, therefore its convergence rate is roughly first order. The accuracy is verified in section 5 through the comparison between numerical solutions and analytic solutions. To increase the efficiency, a relay strategy is used, i.e., a sequence of meshes of different resolutions are employed where the numerical solution from a coarse mesh is relayed to the next finer mesh as the initial guess. This strategy proves quite efficient in reducing the CPU time. The example in section 6.2 shows that the relay process saves roughly 80% CPU time of a single-resolution simulation. However, in the more complicated cases as in section 7, this method is still very slow in convergence. Therefore, more efficient methods with higher order convergence are needed for such problems.

One critical problem is the computation of the gradient or the variation of the cost. Two methods are presented in this work: one is the Variational Formula (VF) method by using the direct formula of the variation, and one is Algorithmic Differentiation (AD) method, through a finite difference approximation of the rate of change of the cost in the direction of each basis function. The VF
method requires the computation of the adjoint problem but only two evolutions, while the AD approach saves the adjoint system but needs a number of evolutions equal to the number of control basis functions. Therefore, the VF method saves CPU time when the number of basis functions is large. According to the performance analysis in this work, the optimization process based on the VF method only takes one third of the time of the AD method when all the other factors are identical.

The numerical results indicate the following features of mixing induced by the boundary control. First, the boundary control can be as efficient as internal body control, which can be seen from the comparison of the results in section 7 with those in [32], where the velocity is from the internal stirring. In section 6, it is observed that a stagnant point appears in the center of the domain because the velocity fields generated by the boundary input functions \( \cos(2\omega) \) and \( \sin(2\omega) \) are zero there. However, by incorporating additional control basis functions \( \cos(\omega) \), \( \sin(\omega) \), and 1, the stagnant point is eliminated and it seems that a physical point can be advected to everywhere in the domain. Second, using more diverse control basis functions and more time segments could help generate more irregular flow motions as well as reduce the cost functional values.

In summary, the gradient descent optimization algorithm with VF method and relay strategy is accurate and effective for this challenging optimization problem. More importantly, the numerical experiments show that boundary control is a plausible option for the optimal mixing control design. How to improve the computational efficiency (possibly by utilizing model reduction type of techniques), construct mixing-efficient boundary control basis functions, and establish the rigorous analysis for identifying the mixing decay rate under such optimal control strategies, merit further investigation in the future work.

Acknowledgments W. Hu was partially supported by the NSF grant DMS-1813570. This work was supported in part by computational resources and services provided by HPCC of the Institute for Cyber-Enabled Research at Michigan State University through a collaboration program of Central Michigan University, USA.

9 Appendix

9.1 Challenge in deriving the first order necessary conditions of optimality

Given \( \theta_0 \in L^\infty(\Omega) \), the existence of an optimal solution to the problem (P) has been proven for \( v_0 \in V_n^0(\Omega) \) and \( g \in L^2(0,T;V_n^0(\Gamma)) \) in [18 Theorem 3.2]. However, the challenge arises in deriving the first order necessary conditions of optimality for solving the optimal controller. As a first step to carry out the variational formula (3.4), one needs the map \( g \mapsto (\theta,v) \) to be Gâteaux differentiable, which gives rise to the major obstruction. To establish the well-posedness of the Gâteaux derivative of \( \theta \), i.e., \( \theta'(h) \cdot h \), for \( h \in U_{ad} \), one needs sup \( t \in [0,T] \| \nabla \theta \|_{L^2} < \infty \), which in turn demands \( \theta_0 \in H^1(\Omega) \) and the flow velocity to satisfy

\[
\int_0^T \| \nabla v \|_{L^\infty(\Omega)} \, dt < \infty. \tag{9.1}
\]

Therefore, the initial condition \( v_0 \) and \( U_{ad} \) have to be chosen such that this a priori estimate is satisfied. In fact, according to [18 Lemma 4.2] we know that if \( v_0 \in V_n^{d/2-1+\epsilon}(\Omega) \) and

\[
g \in \mathcal{S} = L^2(0,T;V_n^{d/2-1/2+\epsilon}(\Gamma)) \cap H^{d/4-1/4+\epsilon/2}(0,T;V_n^0(\Gamma)), \quad d = 2, 3, \tag{9.2}
\]

for \( \epsilon > 0 \) sufficiently small, then (9.1) holds. As a result, in [19] the time derivative of the boundary input \( g \) was penalized in the cost functional as to meet such a high regularity requirement on \( v \).
For computational convenience, the first derivative $\partial g/\partial t$ was adopted. Consequently, the resulting optimality system became intractable to solve, as shown in [19, Theorem 4.1], let alone to analyze the uniqueness of the optimal solution. However, an approximating control design was used in [18] to lower the regularity requirement on velocity and finally, it was possible to find an optimal control in $U_{ad} = L^2(0, T; L^2(\Gamma))$. This was achieved by first introducing a small diffusivity $\epsilon \Delta \theta$, for $\epsilon > 0$, to the transport equation associated with an appropriate boundary condition, and then establishing a rigorous analysis of convergence of the approximating control problem to the original one as the diffusivity approaches zero. Uniqueness of the optimal solution is also obtained for $d = 2$, given $\gamma$ sufficiently large.

In summary, the first order conditions of optimality can be established by setting $\theta_0 \in L^\infty(\Omega) \cap H^1(\Omega)$, $v_0 \in V^{d/2-1+\epsilon}_d(\Omega)$, and $g \in U_{ad} = L^2(0, T; L^2(\Gamma))$ through an approximating control approach.

9.2 Operator $L$ in the non-homogeneous Navier slip boundary value problem

To define the Stokes operator associated with Navier slip boundary conditions, we introduce the bilinear form

$$a_0(v, \psi) = 2(\nabla(v), \nabla(\psi)) + k \langle v, \psi \rangle_\Gamma, \quad k > 0, \quad v, \psi \in V^1_n(\Omega).$$

By Korn’s inequality and trace theorem, it is easy to check that $c_1 \|v\|^2_{H^1} \leq a_0(v, v) \leq c_2 \|v\|^2_{H^1}$, for some constants $c_1, c_2 > 0$. Thus $a_0(\cdot, \cdot)$ is $H^1$-coercive. Let $(V^1_n(\Omega))^\prime$ be the dual space of $V^1_n(\Omega)$. Define the operator $A: V^1_n(\Omega) \to (V^1_n(\Omega))^\prime$ by

$$(Av, w) = a_0(v, w). \quad (9.3)$$

The Lax-Milgram Theorem implies that $A \in L(V^1_n(\Omega), (V^1_n(\Omega))^\prime)$. This also allows us to identify $A$ as an operator acting on $V_0^n(\Omega)$ with the domain $D(A) = \{v \in V^1_n(\Omega): w \mapsto a_0(v, w) \text{ is } L^2\text{-continuous}\}$.

In fact, as shown in [21, (2.9)] and [25, (5.1)], for $v \in V^2_n(\Omega)$ satisfying the homogenous Navier slip boundary conditions in (2.4) and $w \in V^1_n(\Omega)$, we have

$$\int_\Omega \Delta v \cdot w \, dx = -2 \int_\Omega \nabla(v) \cdot \nabla(w) \, dx - \int_\Gamma k(v \cdot \tau)(w \cdot \tau) \, dx. \quad (9.4)$$

Thus (9.3)–(9.4) define the Stokes operator $A = -\mathbb{P}\Delta$ with domain

$$\mathcal{D}(A) = \{v \in V^2_n(\Omega): (2n \cdot \nabla(v) \cdot \tau + kv \cdot \tau)|_\Gamma = g \cdot \tau\},$$

where $\mathbb{P}$ is the Leray projector in $L^2(\Omega)$ on the space $V^0_n(\Omega)$.

By making a change of variable, we may rewrite the nonhomogenous boundary problem (2.2)–(2.4) as a variation of parameters formula

$$v(t) = e^{-At} v_0 + (Lg)(t), \quad (9.5)$$

where $e^{-At}$ is an analytic semigroup generated by $-A$ on $V^0_n(\Omega)$ and $L$ is given by

$$(Lg)(t) = \int_0^t A e^{-A(t-\tau)} Ng(\tau) \, d\tau. \quad (9.6)$$
Here \( N : L^2(\Gamma) \to V^0_n(\Omega) \) is so called the Navier slip boundary operator defined by
\[
Ng = v \iff a_0(v, w) = \langle g, w \rangle_\Gamma, \quad w \in V^1_n(\Omega).
\]
Moreover,
\[
N^* Aw = w|_\Gamma, \quad w \in \mathcal{D}(A), \tag{9.7}
\]
where \( N^*: V^0_n(\Omega) \to L^2(\Gamma) \) is the \( L^2 \)-adjoint operator of \( N \) (cf. [19, 26]). If we let \( L^* \) be the \( L^2(0, T; \cdot) \)-adjoint operator of \( L \), then \( L^* \) is given by
\[
(L^* w)(t) = \int_t^T N^* A e^{-A(t-\tau)} w(\tau) d\tau \big|_\Gamma, \quad w \in L^2(0, T; V^0_n(\Omega)). \tag{9.8}
\]

9.3 Proof of Corollary 2.3

**Proof.** For any \( s \in [0, T] \), taking the inner product of (2.1) with \( \rho \) over \( \Omega \) and integrating \( t \) from \( s \) to \( T \) gives
\[
\int_s^T (\partial \theta/\partial t, \rho) dt + \int_s^T (v \cdot \nabla \theta, \rho) dt = 0, \tag{9.9}
\]
where
\[
(v \cdot \nabla \theta, \rho) = \int_\Omega v \cdot \nabla (\theta \rho) dx - \int_\Omega \theta v \cdot \nabla \rho dx
\]
\[
= \int_\Gamma v \cdot n(\theta \rho) ds - \int_\Omega \nabla \cdot v(\theta \rho) dx - \int_\Omega \theta v \cdot \nabla \rho dx = - \int_\Omega \theta v \cdot \nabla \rho dx.
\]
In the last step, the conditions that \( \nabla \cdot v = 0 \) and \( v \cdot n|_\Gamma = 0 \) are used. Therefore, from (9.9) we have
\[
\int_\Omega (\int_s^T (\rho \theta)_t - \theta \rho_t) dt dx - \int_\Omega (\int_s^T \theta v \cdot \nabla \rho dx) dt = 0,
\]
and thus
\[
\int_\Omega \rho(x, T) \theta(x, T) dx - \int_\Omega \rho(x, s) \theta(x, s) dx = \int_\Omega \int_s^T \theta(\rho_t + v \cdot \nabla \rho) dt dx.
\]
Finally using (2.10) leads to
\[
\int_\Omega \rho(x, T) \theta(x, T) dx = \int_\Omega \rho(x, s) \theta(x, s) dx, \tag{9.10}
\]
which completes the proof.

9.4 Energy, work, and nonphysical concept ‘action’

Taking \( w = v \) in (2.14) and integrating it from \( t = 0 \) to \( t = T \) along with the initial velocity being zero gives
\[
\int_0^T \int_\Gamma (g \cdot \tau)(v \cdot \tau) ds dt = \frac{1}{2} \int_\Omega |v(x, T)|^2 dx + 2 \int_0^T \int_\Omega |D(v)|^2 dx dt + k \int_0^T \int_\Gamma |(v \cdot \tau)|^2 ds dt. \tag{9.11}
\]
The left side is the total work done by the tangential force \( g \cdot \tau \) on the system in the time window \([0,T]\). The first term on the right is the kinetic energy at the final time \( T \), the second term is the total viscous dissipation, and last term is the total dissipation due to the friction between the fluid and the wall.

Since the force \( g \) has only the tangential component, the g-norm square,
\[
\|g\|_{U_{ad}}^2 = \int_0^T \int_\Gamma |g|^2 ds dt = \int_0^T \int_\Gamma (g \cdot \tau)(g \cdot \tau) ds dt,
\]
which is the total work of the force on the flow whose tangential velocity component has the same magnitude and direction as the force. It tightly relates to the left side of (9.11) through the following inequality,
\[
\int_0^T \int_\Gamma (g \cdot \tau)(v \cdot \tau) ds dt \leq \frac{1}{2} \int_0^T \int_\Gamma |g \cdot \tau|^2 ds dt + \frac{1}{2} \int_0^T \int_\Gamma |v \cdot \tau|^2 ds dt = \frac{1}{2} \|g\|_{U_{ad}}^2 + \frac{1}{2} \int_0^T \int_\Gamma |v \cdot \tau|^2 ds dt
\]
(9.13)

The word ‘action’ is defined in [32] as the time integral of the kinetic energy. However, the kinetic energy is the work needed to accelerate a body of a given mass from rest to its stated velocity. Thus, the time integration of kinetic energy lacks physical meaning. In the literature, the concept of ‘action’ used as the time integral of kinetic energy is only found in [32].

9.5 Derivation of the Gâteaux derivative (3.3)

The rigorous derivation of the first order optimality system for \( U_{ad} = L^2(0,T;V_n^0(\Gamma)) \) has been addressed in [18], using an approximating control approach. Here we provide a formal and direct derivation given that all the involved functions are sufficiently smooth and all the operations are valid.

**Theorem 9.1.** Let \( U_{ad} = \text{span}\{g_1, \ldots, g_M\} \), where \( g_i \in \mathbb{S}, i = 1, 2, \ldots, M \). Then
\[
J'(g) \cdot \varphi = \int_0^T (\theta \nabla \rho, L\varphi) dt + \gamma \int_0^T \langle g, \varphi \rangle_\Gamma dt, \quad \forall \varphi \in U_{ad},
\]
(9.14)

where \( \rho \) is the adjoint state solved from (2.10)–(2.11) and \( L\varphi \) is the velocity field governed by the Stokes system (2.2)–(2.5) with \( v_0 = 0 \) and the tangential boundary control input \( g = \varphi \).

**Proof.** Recall that
\[
J(g) = \frac{1}{2}(\Lambda^{-2}\theta(T), \theta(T)) + \frac{1}{2} \int_0^T \langle g, g \rangle_\Gamma dt,
\]
where \( \Lambda^{-2} = (-\Delta + I)^{-1} \) is a positive and self-adjoint operator. Taking the Gâteaux derivative of \( J \) at \( g \) in the direction \( \varphi \) follows
\[
J'(g) \cdot \varphi = (\Lambda^{-2}\theta(T), (\theta'(g) \cdot \varphi)(T)) + \int_0^T \langle g, \varphi \rangle_\Gamma dt,
\]
(9.15)

where \( (\theta'(g) \cdot \varphi)(T) \) is the Gâteaux derivative of \( \theta \) at \( g \) in the direction \( \varphi \) at time \( T \). Let \( z = \theta'(g) \cdot \varphi \) and \( w = v'(g) \cdot \varphi \). Then it is easy to verify that \( z \) satisfies
\[
\frac{\partial z}{\partial t} + v \cdot \nabla z + w \cdot \nabla \theta = 0,
\]
\[
z(0) = 0.
\]
(9.16)
With the help of the variation of parameters formula (9.5) and the linearity of \( L \), we have \( w = (Lg)' \cdot \varphi = L\varphi \), which is also divergence free and \( (L\varphi) \cdot n|_\Gamma = 0 \).

Next taking the inner produce of (9.16) with some \( \rho \) and integrating with respect to \( t \) over \([0, T]\) give
\[
\int_0^T \left( \frac{\partial z}{\partial t}, \rho \right) dt + \int_0^T (v \cdot \nabla z, \rho) dt + \int_0^T ((L\varphi) \cdot \nabla \theta, \rho) dt = 0,
\]
where \( (v \cdot \nabla z, \rho) = -(v \cdot \nabla \rho, z) \). Using (9.16) we have
\[
(\rho(T), z(T)) - \int_0^T \left( z, \frac{\partial \rho}{\partial t} \right) dt - \int_0^T (v \cdot \nabla \rho, z) dt + \int_0^T ((L\varphi) \cdot \nabla \theta, \rho) dt = 0. \tag{9.17}
\]

Now let \( \rho \) satisfy
\[
-\frac{\partial \rho}{\partial t} - v \cdot \nabla \rho = 0, \tag{9.18}
\]
\[
\rho(T) = \Lambda^{-2}\theta(T). \tag{9.19}
\]

Then from (9.17)–(9.19) it follows
\[
(\Lambda^{-2}\theta(T), z(T)) = (\rho(T), z(T)) = -\int_0^T ((L\varphi) \cdot \nabla \theta, \rho) dt. \tag{9.20}
\]

In fact, \( \rho \) is so called the adjoint state corresponding to \( \theta \) and is solved from the adjoint system (9.18)–(9.19).

Lastly, plugging (9.20) into (9.15), we get
\[
J'(g) \cdot \varphi = -\int_0^T ((L\varphi) \cdot \nabla \theta, \rho) dt + \gamma \int_0^T (g, \varphi)_\Gamma dt = \int_0^T (\theta \nabla \rho, L\varphi) dt + \gamma \int_0^T (g, \varphi)_\Gamma dt,
\]
which completes the proof. \(\square\)

### 9.6 Convergence test of projection scheme and computation of the end-of-step velocity

In this example, the Uzawa iteration number is set as one (then the scheme is reduced to the original one in [15]) and the exact solution is given by \( v_1 = y \cos(t) \sin(3x) \cos(4y), v_2 = -x \cos(t) \sin(3x) \cos(4y), \)
\( p = \cos(t) \cos(5x) \sin(6y) \). Note that the divergence of \( v = (v_1, v_2) \) is not zero and thus the steps (3.20) and (3.21) of the projection scheme introduced in section 3.19 are modified to
\[
\int_\Omega \nabla \phi^{l+1} \cdot \nabla q = -\frac{1}{\Delta t} \int_\Omega q(\nabla \cdot \tilde{v}^{s+1,l+1} - \nabla \cdot v_{\text{exact}}^{s+1}), \quad \forall q \in Q_h, \tag{9.21}
\]
\[
\int_\Omega p^{s+1,l+1} q = \int_\Omega \left[ 2p^s - p^{s-1} + 1.5\phi^{l+1} - (\nabla \cdot \tilde{v}^{s+1,l+1} - \nabla \cdot v_{\text{exact}}^{s+1}) \right] q, \quad \forall q \in Q_h. \tag{9.22}
\]

Consider the unit disk \( \Omega = \{(x, y) : x^2 + y^2 < 1\} \). Table 11 shows the computational results, which demonstrate the classic second order accuracy of velocity in \( H^1 \) norm and of pressure in \( L^2 \) norm.
Table 11: Convergence test of Projection scheme on a unit disk. The errors are measured at $t = 0.5$ and the time step is $\Delta t = 0.01$ for all the cases.

We find that the step (3.22), the computation of the end-of-step velocity, is critical to obtain the accurate and stable results. There is an argument in [15, section 3.5, p. 6016] that the end-of-the-step velocity is not needed. However, according to our computation results shown in Table 12, this is not true. If the end-of-step velocity is not computed, the accuracy cannot be guaranteed and the numerical solutions would blow up. When the time step $\Delta t$ reduces, the blowups become worse (data not shown). One reason that [15] ignored is that their proof is only valid for the divergence free space, but not for the approximate spaces such as Taylor-Hood finite element spaces.

Table 12: Blowups when the end-of-step velocity is skipped. The errors are measured at $t = 0.5$ and the time step is $\Delta t = 0.01$ for both cases.

9.7 A convergence test of DG scheme

Set the velocity to be $(u, v) = (y, -x)$, which is a rotation in the clockwise direction with angular speed 1 radian per unit time. The polar coordinate is denoted as $(r, \omega)$, where the radius $r = \sqrt{x^2 + y^2}$ and the polar angle $\omega = \{ \arccos(x/r), \quad y \geq 0 \\ 2\pi - \arccos(x/r), \quad y < 0 \} \in [0, 2\pi)$. Thus, the scalar advected by this velocity satisfies $\theta(r, \omega, t) = \theta(r, \omega + t, 0)$. The choice of the time step $\Delta t$ follows the CFL condition given in [4, p. 190-191], which states that $\Delta t \leq \frac{h}{|D|_{\max}}$ where $C = 1.256 (M = 0), 0.409 (M = 1), 0.209 (M = 2), 0.130 (M = 3), \text{ and } 0.089 (M = 4)$. In this test, we choose $\theta_0 = \cos(3x + 4y)$. Table 13 shows the errors of RK3 and DGP2 (M=2) method in both maximum and $L^2$ norms, which converge in about the first and the 1.5th orders, respectively. These low orders of convergence are mainly due to the fact that the mesh is a second order approximation of the circular domain.

Table 13: Convergence test of RK3-DGP2 (M=2) scheme on unit disk
References

[1] Alberti, G., Crippa, G., and Mazzucato, A. Exponential self-similar mixing by incompressible flows. Journal of the American Mathematical Society, 32(2):445–490, 2019.

[2] Biswas, A., Foias, C., and Larios, A. On the attractor for the semi-dissipative Boussinesq equations. Annales de l’Institut Henri Poincaré C, Analyse non linéaire, 34(2):381–405, 2017.

[3] Chakravarthy, V. S. and Ottino, J. M. Mixing of two viscous fluids in a rectangular cavity. Chemical Engineering Science, 51(14):3613–3622, 1996.

[4] Cockburn, B. and Shu, C.-W. Runge-kutta discontinuous galerkin methods for convection-dominated problems. Journal of Scientific Computing, 16(3):173–261, 2001.

[5] Crippa, G., Lucà, R., and Schulze, C. Polynomial mixing under a certain stationary euler flow. Physica D: Nonlinear Phenomena, 394:44–55, 2019.

[6] Danckwerts, P. V. The definition and measurement of some characteristics of mixtures. Applied Scientific Research, Section A, 3(4):279–296, 1952.

[7] Davis, P. and Rabinowitz, P. Abscissas and weights for gaussian quadratures of high order. Journal of Research of the National Bureau of Standards, 56:35–37, 1956.

[8] Elgindi, T.M. and Zlatoš, A. Universal mixers in all dimensions. Advances in Mathematics, 356:106807, 2019.

[9] Fortin, M. and Glowinski, R. Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems. North-Holland, 1983.

[10] Gouillart, E., Dauchot, O., Dubrulle, B., Roux, S., and Thiffeault, J.-L. Slow decay of concentration variance due to no-slip walls in chaotic mixing. Physical Review E, 78(2):026211, 2008.

[11] Gouillart, E., Kuncio, N., Dauchot, O., Dubrulle, B., Roux, S., and Thiffeault, J.-L. Walls inhibit chaotic mixing. Physical review letters, 99(11):114501, 2007.

[12] Gouillart, E., Thiffeault, J.-L., and Dauchot, O. Rotation shields chaotic mixing regions from no-slip walls. Physical review letters, 104(20):204502, 2010.

[13] Griva, I., Nash, S.G., and Sofer, A. Linear and nonlinear Optimization. SIAM, 2009.

[14] Gubanov, O. and Cortelezzi, L. Towards the design of an optimal mixer. Journal of Fluid Mechanics, 651:27–53, 2010.

[15] Guermond, J.L., Minev, P., and Shen, J. An overview of projection methods for incompressible flows. Computer Methods in Applied Mechanics and Engineering, 195(44):6011 – 6045, 2006.

[16] Hassanzadeh, P., Chini, G. P., and Doering, C. R. Wall to wall optimal transport. Journal of Fluid Mechanics, 751:627–662, 2014.

[17] Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S. Optimization with PDE constraints, volume 23. Springer Science & Business Media, 2008.

[18] Hu, W. An approximating control design for optimal mixing by Stokes flows. Applied Mathematics & Optimization, pages 1–28, 2018.
[19] Hu, W. Boundary control for optimal mixing by Stokes flows. Applied Mathematics & Optimization, 78(1):201–217, 2018.
[20] Hu, W., Kukavica, I., and Ziane, M. On the regularity for the Boussinesq equations in a bounded domain. Journal of Mathematical Physics, 54(8):081507, 2013.
[21] Hu, W., Wang, Y., Wu, J., Xiao, B., and Yuan, J. Partially dissipative 2d Boussinesq equations with Navier type boundary conditions. Physica D: Nonlinear Phenomena, 376:39–48, 2018.
[22] Hu, W. and Wu, J. Boundary control for optimal mixing via Navier–Stokes flows. SIAM Journal on Control and Optimization, 56(4):2768–2801, 2018.
[23] Hu, W. and Wu, J. An approximating approach for boundary control of optimal mixing via Navier–Stokes flows. Journal of Differential Equations, 267(10):5809–5850, 2019.
[24] Iyer, G., Kiselev, A., and Xu, X. Lower bounds on the mix norm of passive scalars advected by incompressible enstrophy-constrained flows. Nonlinearity, 27(5):973, 2014.
[25] Kelliher, J. P. Navier–Stokes equations with Navier boundary conditions for a bounded domain in the plane. SIAM journal on mathematical analysis, 38(1):210–232, 2006.
[26] Lasiecka, I. and Triggiani, R. Control theory for partial differential equations: Volume 1, Abstract parabolic systems: Continuous and approximation theories, volume 1. Cambridge University Press, 2000.
[27] Lin, Z., Thiffeault, J.-L., and Doering, C. R. Optimal stirring strategies for passive scalar mixing. Journal of Fluid Mechanics, 675:465–476, 2011.
[28] Lions, J.L. Optimal control of systems governed by partial differential equations. 1971.
[29] Liu, M., Muzzio, F.J., and Peskin, R.L. Quantification of mixing in aperiodic chaotic flows. Chaos, Solitons & Fractals, 4(6):869–893, 1994.
[30] Liu, W. Mixing enhancement by optimal flow advection. SIAM Journal on Control and Optimization, 47(2):624–638, 2008.
[31] Lunasin, E., Lin, Z., Novikov, A., Mazzucato, A., and Doering, C. R. Optimal mixing and optimal stirring for fixed energy, fixed power, or fixed palenstrophy flows. Journal of Mathematical Physics, 53(11):115611, 2012.
[32] Mathew, G., Mezić, I., Grivopoulos, S., Vaidya, U., and Petzold, L. Optimal control of mixing in Stokes fluid flows. Journal of Fluid Mechanics, 580:261–281, 2007.
[33] Mathew, G., Mezić, I., and Petzold, L. A multiscale measure for mixing. Physica D: Nonlinear Phenomena, 211(1):23–46, 2005.
[34] Navier, C.-L. Mémoire sur les lois du mouvement des fluides. Mémoires de l'Académie Royale des Sciences de l'Institut de France, 6:389–440, 1823.
[35] Ottino, J. M. The kinematics of mixing: stretching, chaos, and transport, volume 3. Cambridge University Press, 1989.
[36] Seis, C. Maximal mixing by incompressible fluid flows. Nonlinearity, 26(12):3279, 2013.
[37] Thiffeault, J.-L. Using multiscale norms to quantify mixing and transport. Nonlinearity, 25(2):R1, 2012.
[38] Thiffeault, J.-L., Gouillart, E., and Dauchot, O. Moving walls accelerate mixing. Physical Review E, 84(3):036313, 2011.

[39] Vikhansky, A. Enhancement of laminar mixing by optimal control methods. Chemical Engineering Science, 57(14):2719–2725, 2002.

[40] Yao, Y. and Zlatos, A. Mixing and un-mixing by incompressible flows. arXiv preprint arXiv:1407.4163, 2014.

[41] Zhang, L., Cui, T., and Liu, H. A set of symmetric quadrature rules on triangles and tetrahedra. Journal of Computational Mathematics, 27:89–96, 2009.