SIAM Journal on Applied Dynamical Systems, accepted on 19 May 2020.

CONTROLLING TRAJECTORIES GLOBALLY VIA SPATIOTEMPORAL FINITE-TIME OPTIMAL CONTROL*

LI ZHANG† AND SANJEEVA BALASURIYA‡

Abstract. The problems of (i) maximizing or minimizing Lagrangian mixing in fluids via the introduction of a spatiotemporally varying control velocity, and (ii) globally controlling the finite-time location of trajectories beginning at all initial conditions in a chaotic system, are considered. A particular form of solution to these is designed, which uses a new methodology for computing a spatiotemporally-dependent optimal control. An L^2-error norm for trajectory locations over a finite-time horizon is combined with a penalty energy norm for the control velocity in defining the global cost function. A computational algorithm for cost minimization is developed, and theoretical results on global error and cost presented. Numerical simulations (using velocities which are specified, and obtained as data from computational fluid dynamics simulations) are used to demonstrate the efficacy and validity of the approach in determining the required spatiotemporally-defined control velocity.

Key words. Global optimal control, Lagrangian trajectory control, chaos control, flow control, ABC flow, Navier-Stokes flow

AMS subject classifications. 49J15, 34H10

1. Introduction.

In fluid mechanical systems, particles move according to a velocity field $\mathbf{v}$ which is typically dependent on both space $\mathbf{x}$ and time t. This field is often known only numerically, through observational data or computational fluid dynamics simulations. This has the inevitable consequence that the data is *finite-time*, which has resulted in a preponderance of studies on understanding the flow characteristics and important moving flow regions ('coherent structures') in finite-time nonautonomous flows [28, 8, 54]. Often, there is a desire to *control* the flow, usually in order to enhance or suppress mixing (e.g., in optimizing performance in mixing/combustion devices, or reducing the impact of a spreading pollutant). A specific example arises in oil recovery, where one might be interested in driving oil flows to a target region, by using the control strategy of forcing a secondary flow (a chemical slug) in certain locations [35, 62]. Current theoretical developments in the area of mixing optimization/suppression area are varied (e.g., parametric investigation of flow protocols in specific geometries in zero-flow situations [57, 45, 41], maximizing mixing [48, 24], controlling particular trajectories [5, 9] or optimizing fluid mixing across flow barriers [7, 4]). Most do not utilize *optimal control* theory to control particle trajectories, but rely on other aspects of optimization, control, or numerical methods. (Some exceptions: controlling the Navier-Stokes equations [43, 31] and multiobjective mixing control [48].) Here, we specifically examine globally controlling *trajectories* of an existing flow, whose nonautonomous velocities may only know from

*Submitted to the editors May 20, 2020.

Funding: This work was partially funded by Shandong Provincial Government Grants, the Foundation for the National Natural Science Foundation of China-Shandong Joint Fund (No. U1806203), the Australian Research Council through grants FT130100484 and DP200101764, the Australian Department of Education and Training through an Endeavour Research Leadership Award, and the Cooperative Innovation Team of Economic Development and Data Science of Shandong University of Political Science and Law.

†School of Mathematical Sciences, University of Adelaide, Adelaide SA 5005, Australia, and Business School, Shandong University of Political Science and Law, Jinan 250014, Shandong, P. R. China

‡School of Mathematical Sciences, University of Adelaide, Adelaide SA 5005, Australia (sanjeevabalasuriya@yahoo.com, http://maths.adelaide.edu/sanjeeva.balasuriya/).

observational or experimental data. This is 'one step before' the issue of controlling *mixing* [41, 45, 48, 24], in which diffusion also needs to be taken into account. In this case, we are able to specify the targetted locations of *all* initial conditions after a finite-time flow, and seek an added spatio-temporally dependent control velocity which helps achieve this target globally.

Physically, the flow can be controlled by introducing additional velocities which are spatiotemporally-dependent, e.g., by moving a solid or flexible boundary in some specified way [19, 58], introducing fluid inlets/outlets at various locations [46, 60, 29], displacement by chemical slugs [35, 62], or via microtransducers [32]. Thus, a control velocity which is both spatially- and temporally-dependent is achievable physically. In this case, since we specify eventual trajectory locations, we build a cost function which includes both a distance norm (which captures how closely *all* particles reached the targetted location) and a penalty term (which limits the size of the control velocity). Assuming that the original velocity field is given (possibly in terms of data), in this paper we develop a method for determining the spatio-temporally varying control velocity field which minimizes the cost function. This is achieved by modifying and adapting optimal control methods to this setting, while providing both theoretical results and computational strategies for using our technique.

Many methods have been suggested in the fluid mechanics literature for different types of flow control. These include turbulence control in various ways by conditioning velocity gradients, energy or enstrophy [11, 31, 43, 53], drag forces [11], or boundary layers and skin friction [34, 38]. In most of these cases, the issue is to control the (Eulerian) velocity field, which evolves according to the Navier-Stokes equation (or some approximation/modification). This is a challenging infinite-dimensional situation, often requiring geometry-specific methods and projections into finite dimensions (e.g, Fourier modes or orthogonal decompositions [34, 53]). Of course, in highly turbulent situations in which gradients are large over small scales, achieving such a control would require velocity modifications at smaller and smaller scales, which is impractical. Moreover, difficulties in achieving control over long-term time-horizons are well-established [11]. In contrast to controlling Eulerian velocities which are solutions to the Navier-Stokes equations, what we study in this paper is the control of *Lagrangian trajectories* associated with such Eulerian velocity field. Given that Lagrangian trajectories are solutions to an ordinary differential equation associated with the Eulerian velocity, the control problem is now a *low-dimensional* one, with dimensionality given by the spatial dimension of the flow. However, the difficulty here is that we seek spatially *global* trajectory control at a final time, which we are able to achieve in a certain way while taking advantage of the low-dimensionality of the control problem. Given that control velocities can in reality be achieved only at some spatial resolution, our methods are expected to have accuracy if the turbulence is moderate, but not excessive.

While fluid mechanics is the motivation for this paper, our development is independent of it. Our methodology applies to *general systems* of ordinary differential equations in any dimension, which are moreover either autonomous or nonautonomous and subject to a state equation governed by a vector field $\mathbf{v}$. However—pertinent to the fact that fluid mechanical systems are confined to two or three dimensions—we only claim efficiency at low dimensions. A particular application is to the control of chaos [42, 22, 51, 55]. Generally, chaotic systems have unpredictable trajectories, and classical methods for chaos control include the determination of controls which result in chaotic synchronization [30, 14] or which locally push trajectories towards unstable ones [22, 51, 27, 52]. Since we seek to push trajectories globally over the given time period, our method can be construed as a *global* control framework, in which we simultaneously specify the required fate of *all* trajectories in our phase space. Additionally, we will not confine attention to equilibria (which generically do not exist anyway for nonautonomous systems) or invariant sets such as periodic orbits, and neither will we be concerned about the stability of such sets. Thus, instead of working within this realm of 'classical chaos control,' our method targets the fate of all trajectories after a given finite time.

Some background to our work comes from optimal control theory. Optimal control methods for determining a time-dependent control function for *individual* trajectories is a mature research area [33, 49, 2, 61, 1]. One class of this focuses on obtaining different laws, e.g., feedback control theory as coverage control with diffusive term [44], sliding control with mismatched uncertainties [33], synchronization for non-autonomous chaotic system using integral control [40], global criteria via linear state error equation [17], and via delayed term [15]. Another aspect is that of nonautonomous system, e.g., minimum time control [13]. Classically, optimal control methods focus on a single trajectory of an autonomous system, and often relate to stabilizing unstable equilibria [16, 18, 25, 37]. The theoretical results are usually based on the Pontryagin minimum (or maximum) principle and the associated Hamilton-Jacobi-Bellman partial differential equations. In this work, we extend optimal control to the problem of determining a *spatiotemporally-dependent* control, to *globally* control trajectories over a finite time. By 'globally,' we mean that we can specify the fate of initial conditions as a global function on the initial space, rather than, for example, insisting that all initial conditions go to *one* invariant set [22, 51, 52, e.g.]. Moreover, the method works for general nonautonomous (unsteady) vector fields, and thus is not dependent on the presence of fixed points, periodic orbits or chaotic attractors.

The reminder of this paper is organized as follows. The problem and its theoretical solution is outlined in Section 2. We develop both the computational methodology for determining a spatiotemporal optimal control function, as well as theoretical results indicating the robustness of the procedure and error analyses on the achievement of the finite-horizon target. The algorithm we develop includes novel uses of the Newton-Raphson algorithm and an 'approximant' [20] method to determine the control function spatiotemporally. Section 3 demonstrates the efficacy of the spatiotemporal optimal control in several examples. We demonstrate the ease of implementation of our algorithm, as well as validate the theoretical results concerning the target achievement, and the cost function. The proofs of the theoretical results of Section 2 are separated out for easy readability of the paper, and given in Section A. Finally, in Section 4 we briefly remark on potential extensions of this work.

2. Spatiotemporal optimal control.

Suppose we are given a nonautonomous nonlinear state equation

$$\dot{\mathbf{x}} = \mathbf{v}\left(\mathbf{x}, t\right) \quad ; \quad t \in [0, T]\,, \tag{2.1}$$

where $\mathbf{x} \in \Omega$, and Ω is an open and bounded subset of $\mathbb{R}^n$. We will assume that $\mathbf{v}$ is smooth, and that solutions to (2.1) exist for all $t \in [0, T]$ (thereby precluding issues such as 'blow-up in finite-time' [56]). For $\mathbf{v}$ obtained on a spatio-temporal grid instead, we imagine that $\mathbf{v}$ is smoothly extended to the subgrid level (a strategy that is usually done when computing trajectories in such cases; see the citations in [8, 28]). Since such an extension may give values of $\mathbf{v}$ which are in reality inaccurate, we will (in Theorem 2.4) establish that our method is robust towards these errors.

Our goal is to find an additive spatiotemporal control $\mathbf{c}\left(\mathbf{x}, t\right)$ such that initial conditions $\mathbf{x}_0$ at $t = 0$, in a restricted domain $\Omega_0 \subseteq \Omega$, approach at the final time T specified target locations, which are identified via a globally defined target function $\Theta : \Omega_0 \to \mathbb{R}^n$. This target function must be *achievable* in that it is generated by a flow (i.e., there exists a velocity field $\mathbf{u}(\mathbf{x}, t)$ such that the flow map of $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}, t)$ from time 0 to T is $\Theta(\mathbf{x})$). In particular, Θ cannot demand flow trajectories which must cross each other, or reverse orientation in other ways. (For example, setting $\Theta(x) = -x$ if $x \in \mathbb{R}$ is unachievable, since this requires trajectories to cross each other—which is impossible for a flow.) However, we may specify Θ to have jump discontinuities, enabling for example steering trajectories into three different target locations. We emphasize that there is no restriction to equilibria or other specialized trajectories of (2.1), but we rather seek to steer *all* trajectories globally to *any* achievable specified locations by time T. The controlled nonautonomous state equation will take the form

$$\dot{\mathbf{x}} = \mathbf{v}\left(\mathbf{x}, t\right) + \mathbf{c}\left(\mathbf{x}, t\right) \quad ; \quad t \in [0, T]\,, \tag{2.2}$$

where we use the notation $\mathbf{c}(\mathbf{x}, t)$ for the control. In the standard language of fluid mechanics, this represents the control velocity in the 'natural' Eulerian coordinate $\mathbf{x}$, based on information from the Lagrangian trajectories of (2.1). We will denote by $\mathbf{x}\left(\mathbf{x}_0, t\right)$ solutions of (2.2) at time $t \in [0, T]$ subject to the initial condition $\mathbf{x}_0$ at time 0. The optimal control problem globally on the $t = 0$ spatial domain Ω_0 can then be posed as the determination of the control $\mathbf{c}$ (defined on a spatiotemporal domain $(\mathbf{x}, t)$) which minimizes the cost function

$$G := \int_{\Omega_0} \left[\left\| \mathbf{x}(\mathbf{x}_0, T) - \Theta\left(\mathbf{x}_0\right) \right\|^2 + \eta \int_0^T \left\| \mathbf{c}(\mathbf{x}(\mathbf{x}_0, t), t) \right\|^2 \mathrm{d}t \right] \mathrm{d}\mathbf{x}_0\,, \tag{2.3}$$

in which $\|\mathbf{.}\|$ is the standard Euclidean norm and $\eta > 0$ encapsulates the penalty for the energy contained in $\mathbf{c}$ over the time period $[0, T]$. This regularizes the problem (and hence jump discontinuities are specifiable in Θ; these will be approximately achieved when minimizing G for small but nonzero η).

We will develop a method for solving the minimization problem numerically for any given initial domain Ω_0, final time T, evolution law $\mathbf{v}$ defined on $[0, T]$, target function Θ, and energy parameter η. We will moreover provide theoretical estimates on how the error in achieving the target decays with time and η. We also remark that within this formulation (specifically attempting to find a control velocity in the form $\mathbf{c}(\mathbf{x}, t)$ for minimizing the spatially-integrated cost function G), proceeding through the Hamilton-Jacobi-Bellman approach directly is unfeasible because a numerical minimization is required within the partial differential equation. We instead adopt an approach which uses different established methods (from optimal control, fluid mechanics, differential equations theory, computer visualization) in an unusual way.

For $\mathbf{x}_0 \in \Omega$, we define

$$g(\mathbf{x}_0) := \left\| \mathbf{x}(\mathbf{x}_0, T) - \Theta\left(\mathbf{x}_0\right) \right\|^2 + \eta \int_0^T \left\| \mathbf{c}(\mathbf{x}(\mathbf{x}_0, t), t) \right\|^2 \mathrm{d}t\,, \tag{2.4}$$

and note that $G = \int_{\Omega_0} g(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0$. Since $g \geq 0$ for all $\mathbf{x}_0 \in \Omega$, minimizing G can be accomplished by minimizing g at each $\mathbf{x}_0$—a canonical optimal control problem—and then combining the results. (There is a caveat to this statement, which we will return to in describing the process in more detail subsequently.) Now, once minimizing g has been achieved for a particular initial condition $\mathbf{x}_0$, it will result in a control $\mathbf{c}$ defined along the specific trajectory $(\mathbf{x}(\mathbf{x}_0, t), t)$ of spacetime. Subsequently, we will detail a method for concatenating the results for each such $\mathbf{x}_0$ to be able to define $\mathbf{c}$ across all (relevant) spacetime $(\mathbf{x}, t)$.

THEOREM 2.1 (Single-trajectory optimal control). *For $\mathbf{x}_0$ fixed in Ω, any optimal control $\mathbf{c}$ locally minimizing (2.4) is representible as*

$$\mathbf{c}(\mathbf{x}(\mathbf{x}_0, t), t) = -\frac{1}{2\eta}\mathbf{p}(t) \quad ; \quad t \in [0, T], \tag{2.5}$$

in which the conjugate momentum $\mathbf{p}$ obeys the coupled system

$$\left.\begin{array}{rcl} \dot{\mathbf{x}} & = & \mathbf{v}(\mathbf{x}, t) - \frac{1}{2\eta}\mathbf{p} \\ \dot{\mathbf{p}} & = & -\left[\nabla \mathbf{v}(\mathbf{x}, t)\right]^\top \mathbf{p} \end{array}\right\} \tag{2.6}$$

subject to the implicitly-defined initial and end conditions

$$\left.\begin{array}{rcl} \mathbf{x}(0) & = & \mathbf{x}_0 \\ \mathbf{p}(T) & = & 2\left(\mathbf{x}(T) - \Theta(\mathbf{x}_0)\right) \end{array}\right\}. \tag{2.7}$$

Here, $[\bullet]^\top$ denotes the matrix transpose, and $\nabla \mathbf{v}$ is the $n \times n$ matrix derivative of $\mathbf{v}$ with respect to the spatial variable $\mathbf{x}$.

Proof. See Section A.1; this is an elementary application of optimal control. ∎

The fact that the condition on $\mathbf{p}$ in (2.7) is an *end* condition (while that of $\mathbf{x}$ is an initial condition), and moreover depends on the unknown value $\mathbf{x}(T)$, necessitates some care when solving (2.6)-(2.7) numerically. Methods such as indirect shooting, multiple shooting, collocation approaches, as well as sequential, simultaneous or direct transcription have been suggested for this well-known problem. While indirect methods suffer difficulties in acquiring a good initial guess and in repeated differentiation, the discretization associated with direct methods tends to obtain less accurate solutions. Here, we opt for a Newton-Raphson based (indirect) method which, as we demonstrate, has quick convergence. Having guessed an initial condition $\mathbf{q} := \mathbf{p}(0) \in \mathbb{R}^n$, we implement (2.6) in forward time (in this case, we use the built-in ordinary differential equation solvers in Matlab$^{\text{TM}}$), and consequently, determine $\mathbf{x}(T)$ and $\mathbf{p}(T)$ for that initial choice. Given that these depend on the initial guess $\mathbf{q}$, we use the notation $\mathbf{x}(T, \mathbf{q})$ and $\mathbf{p}(T, \mathbf{q})$ respectively, and define

$$\mathbf{F}(\mathbf{q}) := \mathbf{p}(T, \mathbf{q}) - 2\,\mathbf{x}(T, \mathbf{q}) + 2\,\Theta(\mathbf{x}_0). \tag{2.8}$$

If we find a root $\mathbf{q}$ of $\mathbf{F}$, this is a correct initial condition $\mathbf{p}(0)$ to use to generate $\mathbf{c}$ from (2.5). To find such a $\mathbf{q}$, we make an initial guess $\mathbf{q}_0$, and choose a small quantity δ. We then take the $2n$ 'nearest neighbors' of $\mathbf{q}_0$, i.e., $\mathbf{q}_0 \pm \delta \mathbf{e}_i$ for $i = 1, 2, \cdots, n$ where the $\mathbf{e}_i$s are the rectangular basis elements on $\mathbb{R}^n$. Given $\mathbf{x}(0) = \mathbf{x}_0$ and each of these initial conditions for $\mathbf{p}$, we then advect (2.6) numerically forward to time T. We can now calculate the value of $\mathbf{F}$ using $\mathbf{q} = \mathbf{q}_0$, and can use the results of all the nearest neighbor advections to numerically evaluate each of the values $\mathbf{F}(\mathbf{q}_0 \pm \delta \mathbf{e}_i)$, and hence estimate the matrix $\nabla \mathbf{F}(\mathbf{q}_0)$ using standard finite-differencing. We then make an improved guess for the root $\mathbf{q}$ (which we call $\mathbf{q}_1$) using the Newton-Raphson method. More concretely, we go from our jth guess to the $(j+1)$st guess by

$$\mathbf{q}_{j+1} = \mathbf{q}_j - \left(\left[\nabla \mathbf{F}(\mathbf{q}_j)\right]^{-1}\right)^\top \mathbf{F}(\mathbf{q}_j), \tag{2.9}$$

and stop the process once $\|\mathbf{F}(\mathbf{q}_j)\|$ is smaller than a specified threshold. The corresponding solution $\mathbf{p}(t)$ then gives us the required (single-trajectory) control $\mathbf{c}$ using (2.5).

Thus, for any $\mathbf{x}_0 \in \Omega_0$, we can determine the solution trajectories $\mathbf{x}(\mathbf{x}_0, t)$. To quantify how we approach the target at time T, we define the global target error

$$E(t) := \left(\int_{\Omega_0} \left\| \mathbf{x}(\mathbf{x}_0, t) - \Theta(\mathbf{x}_0) \right\|^2 \, \mathrm{d}\mathbf{x}_0 \right)^{1/2} \tag{2.10}$$

for times $t \in [0, T]$. We note that $E(0)$ is the $L^2(\Omega_0)$-norm of the function $\mathbf{x}_0 - \Theta(\mathbf{x}_0)$, and can be assumed known from the problem statement. We now characterize, in terms of 'given' quantities (i.e., information about $\mathbf{v}$, Ω, T, η and Θ), the rate at which $E(t)$ approaches its final value $E(T)$. We first require to define some norms for functions $\mathbf{h} : \Omega \times [0, T] \to \Omega$. If $\|\centerdot\|$ is the standard Euclidean norm in $\mathbb{R}^n$, let

$$\|\mathbf{h}\|_a := \sup_{(\mathbf{x},t) \in \Omega \times [0,T]} \|\mathbf{h}(\mathbf{x}, t)\| \ , \ \text{and} \tag{2.11}$$

$$\|\mathbf{h}\|_b := \sup_{(\mathbf{x},t) \in \Omega \times [0,T]} \sup_{\mathbf{y} \in \Omega, \mathbf{y} \neq \mathbf{0}} \frac{\left\| \nabla \mathbf{h}^\top (\mathbf{x}, t)\, \mathbf{y} \right\|}{\|\mathbf{y}\|} \,. \tag{2.12}$$

THEOREM 2.2 (Global error decay). *If there exists constants A and B such that $\|\mathbf{v}\|_a \leq A < \infty$ and $\|\mathbf{v}\|_b \leq B < \infty$, then the rate of decay of $E(t)$ to $E(T)$ obeys*

$$|E(t) - E(T)| \leq \sqrt{2} \left[A \sqrt{\mu(\Omega_0)} (T - t) + \frac{E(T)}{\eta} \frac{\left(e^{B(T-t)} - 1 \right)}{B} \right] , \tag{2.13}$$

where $\mu(\Omega_0)$ is the standard Lebesgue measure on Ω_0.

Proof. See Section A.2. ∎

As t approaches T, $E(t)$ approaches $E(T)$ due to two effects: a linear rate which is characterized by $\|\mathbf{v}\|_a$, and an exponential rate which is characterized by $\|\mathbf{v}\|_b$. We emphasize that these results hold even if Θ is discontinuous (as we will demonstrate in Section 3).

Next, we address η-dependence in the cost and global error. By choosing η smaller and smaller, since the penalization of the control velocity $\mathbf{c}$ is reduced in (2.3), one can achieve a target Θ with infinitesimal accuracy by choosing $\mathbf{c}$ closer and closer to the exact value $\mathbf{u} - \mathbf{v}$, where $\mathbf{u}$ is the velocity field which engenders the flow map Θ when considering the flow from time 0 to T. Thus, $E(T)$ decreases with η. We now establish a relationship with the decay of the total cost G.

THEOREM 2.3 (Comparative η-dependence). *Suppose the hypotheses of Theorem 2.2 are satisfied. If there exists $\alpha > 1/2$ such that*

$$\lim_{\eta \downarrow 0} \frac{E(T)}{\eta^\alpha} < \infty \quad , \quad \text{then} \quad \lim_{\eta \downarrow 0} \frac{G}{\eta^{2\alpha - 1}} < \infty \,.$$

Proof. See Section A.3. ∎

If we know that the global error decays as $\mathcal{O}(\eta^\alpha)$, then the total cost will decay as $\mathcal{O}(\eta^{2\alpha-1})$. We note that if $\alpha = 1$, Theorem 2.3 implies that $E(T)$'s $\mathcal{O}(\eta)$ decay implies that G *also* has $\mathcal{O}(\eta)$ decay. We will demonstrate this particular behavior in our numerical simulations.

Next, we address the issue of how to determine $\mathbf{c}$ as a spatiotemporal function. By the process associated with using Theorem 2.1 and the Newton-Raphson method (2.9), given any initial condition $\mathbf{x}_0 \in \Omega$, we can find the optimal control function $\mathbf{c}$'s values along the spacetime curve $(\mathbf{x}(\mathbf{x}_0, t), t)$. By doing this for a grid of initial conditions $\mathrm{x}_0 \in \Omega_0$, we generate a collection of such spacetime curves along which we know the value of $\mathbf{c}$. We now seek $\mathbf{c}$ as a spatiotemporal function (i.e., as a function on $(\mathbf{x}, t)$). In doing so, we make the assumption that the generated spacetime curves are *consistent*, that is, should any two curves intersect in spacetime, the determined values of $\mathbf{c}$ at the point of intersection by following along either of the curves should give the identical value. (Since numerical approximation is being used, exactly identical is not necessary, but rather these should agree to within the resolution sought.) This is the caveat necessary to ensure that minimizing g and then extending to all spacetime is equivalent to minimizing the global cost G.

Now, performing numerical interpolation proves to be ineffective and difficult because the spacetime curves do not uniformly traverse spacetime. Moreover, given the possibility that the function Θ is nonsmooth (e.g., if different collections of initial conditions are steered towards, say, two different points—this example shall be shown in our simulations in Section 3), the consequent roughness of $\mathbf{c}$ results in wild oscillations in the interpolants. Thus, we instead use an *approximant* for $\mathbf{c}$ based on knowledge of the values of $\mathbf{c}$ along the collection of nonuniformly distributed spacetime curves. This is achieved easily in two-dimensions in Matlab$^{\text{TM}}$ by using the package `gridfit` [20], which regularizes the interpolation problem in seeking a smoother surface fitting for $\mathbf{c} = \mathbf{c}(\mathbf{x}, t)$. (The basic idea, described in detail in [20], is to fit an elastic plate approximately through the given points, with a stiffness parameter which penalizes deviation from the points.) Throughout this work we use the Laplacian as the regularizer (the Laplacian integrated over the fitted surface is to be kept small [20]). An N-dimensional version of this, `regularizeNd`, has also been developed [47], and is what we use when we consider higher-dimensional situations in our examples.

Using our theoretical results, we can comment on the robustness of the process in the following sense. Suppose that instead of $\mathbf{v}$, the true governing vector field is $\tilde{\mathbf{v}}$, where while $\tilde{\mathbf{v}}$ is unknown, we know that it is 'close' to $\mathbf{v}$ (for estimates on Lagrangian trajectory uncertainty resulting from this, see [6]). This is inevitable if $\mathbf{v}$ were known from data; even if know *exactly* at the gridpoints, $\mathbf{v}$ would need to be interpolated in some way at the subgrid level when performing trajectory calculations. Moreover, the values at the gridpoints, if obtained from experimental or observational data, will carry their own measurement errors. Thus, there will always be an error in $\mathbf{v}$ when considered over the domain $\Omega \times [0, T]$.

THEOREM 2.4 (Robustness to uncertainties in $\mathbf{v}$). *Suppose there exists $\epsilon > 0$ such that $\|\mathbf{v} - \tilde{\mathbf{v}}\|_a < \epsilon$ and $\|\mathbf{v} - \tilde{\mathbf{v}}\|_b < \epsilon$. If the 'tilde' variables are the quantities associated with using $\tilde{\mathbf{v}}$ rather than $\mathbf{v}$ in calculations of the control velocity, global error, and cost, then*

$$
\left.\begin{array}{lcl}
\mathbf{c}(\mathbf{x}, t) & = & \tilde{\mathbf{c}}(\mathbf{x}, t) + \mathcal{O}(\epsilon) \quad \text{in } \Omega \times [0, T], \\
E(T) & = & \tilde{E}(T) + \mathcal{O}(\epsilon) \ \text{ and} \\
G & = & \tilde{G} + \mathcal{O}(\epsilon).
\end{array}\right\} \tag{2.14}
$$

Proof. See Section A.4. ∎

Theorem 2.4 ensures that, since we follow our procedure by using $\mathbf{v}$ rather than the unknown (but $\mathcal{O}(\epsilon)$-close) $\tilde{\mathbf{v}}$, all relevant computed quantities are similarly $\mathcal{O}(\epsilon)$-close to the 'true' values. This suggests a (specific type) of robustness of the procedure:

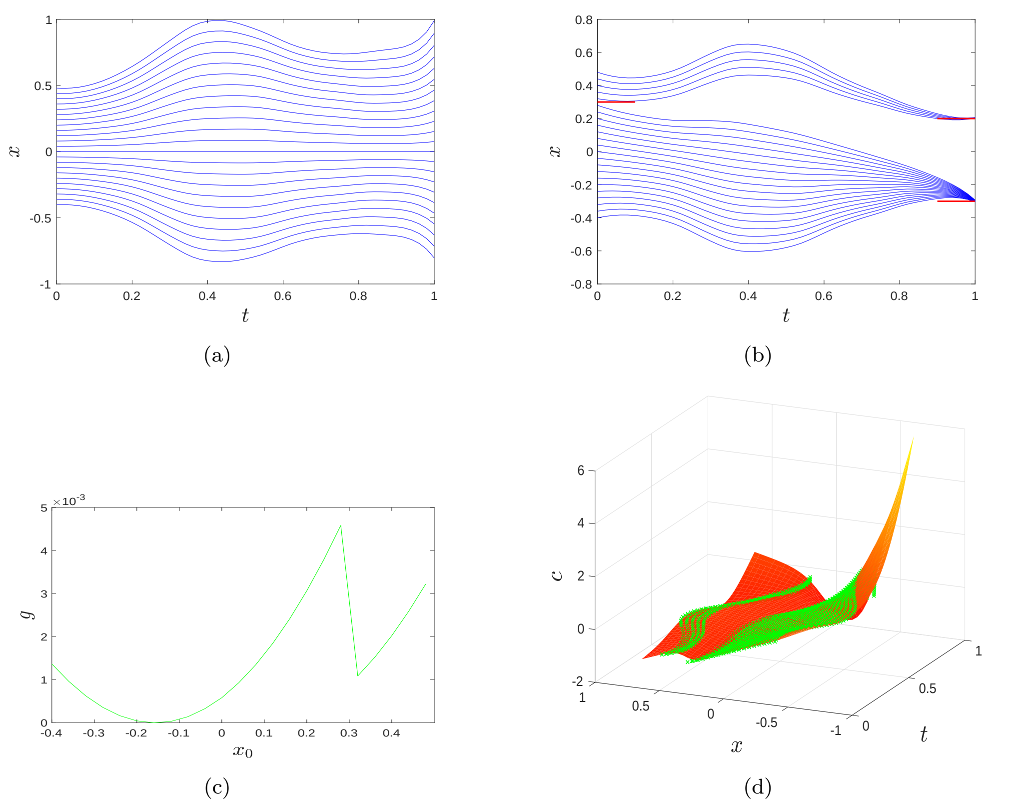


FIG. 1. *Spatiotemporal control for (3.1). (a) uncontrolled, (b) controlled to approach two points, (c) cost distribution, and (d) spatiotemporal control function.*

results will be correct to the same order of uncertainty as in $\mathbf{v}$.

We have thus developed a methodology for determining a spatiotemporal control in finite-time, in relation to globally defined targets, by a process of utilizing an unusual viewpoint and methodology to the optimal control discipline. Our algorithm is summarized below.

1. Reduce the spatiotemporal minimization problem (2.3) to individual single-trajectory optimal control problems related to minimization of (2.4);
2. Solve the resulting initial/end-condition problem (as identified in Theorem 2.1) by using the Newton-Raphson algorithm detailed in (2.9) for each initial condition, thereby determining the trajectory $\mathbf{x}(\mathbf{x}_0, t)$ and the control $\mathbf{c}\left(\mathbf{x}(\mathbf{x}_0, t), t\right)$;
3. Amalgamate the results for each initial condition by applying the `gridfit` [20] or `regularizeNd` [47] method to approximate the spatiotemporal control $\mathbf{c}$ as a function of $(\mathbf{x}, t)$;
4. This algorithm is supported by the theoretical conditions on the decay of $|E(t) - E(T)|$, and the η-dependence of $E(T)$ and G, and robustness towards deviations in $\mathbf{v}$.

3. Simulations. In this Section, we present several simulations which demonstrate the ease at which the spatiotemporal control can be computed, and moreover validate the theoretical results on η-dependence and decay rates.

3.1. A one-dimensional example. For $x \in \mathbb{R}$, and $t \in [0,1]$, let

$$v(x,t) = x \sin(7t+0.3)\, w_1(t) - 4x^3 \cos(5t)\, w_2(t)\,, \tag{3.1}$$

in which some roughness to the velocity is obtained by implementing on a time-scale $\Delta t = 0.02$ a specific realization of stochasticity via $w_1(t) = 3U_1(t) + 0.5$ and $w_2(t) = 2U_2(t) - 2$, where the $U_i(t)$ are independently chosen from the uniform distribution on $[0,1]$. We show in Fig. 1(a) the result of implementing (2.1) for $x_0 \in \Omega_0 = [-0.4, 0.5]$. We first define $\Theta(x_0) = 0.2$ for $x_0 \geq 0.3$ and $\Theta(x_0) = -0.3$ for $x_0 < 0.3$, which separates Ω_0 at 0.3, and aims to send each segment of initial conditions towards a different target point. By using $\eta = 0.01$ and $\delta = 10^{-5}$ and implementing part (2) of our algorithm, we obtain excellent approach to our targets (red lines near $t = 1$), as shown in Fig. 1(b). The desired separation point at $x_0 = 0.3$ is shown by the red line near $t = 0$. The distribution of the required costs for each initial condition x_0 (i.e., (2.4) is shown in Fig. 1(c); the trajectory beginning near -0.15 requires hardly any adjustment, and there is a sharp transition in the cost near $x_0 = 0.3$ because it is necessary to split the trajectories in different directions. The computed control $\mathbf{c}$ for each trajectory is shown in spacetime as the green curves in Fig. 1(d). The red surface—which approximates $c(x,t)$ across the spacetime domain based on the information at the green values—is obtained by applying `gridfit` with its default parameters. We note from Figs. 1(b) and (d) that although this process allows us to determine c on a connected spatiotemporal domain (i.e., the domain associated with the red surface in Fig. 1(d)), in reality its values on the wedge into which trajectories do not enter (because of the separation achieved by the process) are irrelevant.

In Fig. 2(a), we show by the red circles the cost G as η is varied. Performing linear regression on the 10 smallest values of η yields the green line, whose slope indicates that $G \sim \eta^{1.0782}$. We similarly analyze the final target error $E(T)$'s decay with η in Fig. 2(b), and regression reveals that $E \sim \eta^{0.9447}$. Thus, these are consistent with choosing $\alpha \approx 1$ in Theorem 2.3. We next demonstrate the error decay with time in Fig. 2(c). Rapid decay as $t \to T$ is displayed, and in all other implementations (not shown), as predicted by (2.13).

We finally briefly illustrate the impact of choosing different target functions Θ in Fig. 3, noting that Θ must be a monotonic function to avoid trajectories having to cross each other. Excellent results are achieved with these parameters, with costs $G \sim 10^{-3}$ in both instances.

3.2. A two-dimensional example. For $\mathbf{x} = (x,y) \in \mathbb{R}^2$, suppose that

$$\mathbf{v}(\mathbf{x},t) = \begin{pmatrix} v_1(x,y,t) \\ v_2(x,y,t) \end{pmatrix} = \begin{pmatrix} 2x + ty \\ \sin y - t \end{pmatrix}, \tag{3.2}$$

and let the initial time set be $\Omega_0 = [-1,1] \times [0,1]$, to be advected to time $T = 1$. We specify as our target function

$$\Theta(\mathbf{x}_0) = \begin{pmatrix} \frac{x_0^2}{4} + y_0 \\ \cos x_0 - 2x_0 y_0 \end{pmatrix}. \tag{3.3}$$

There is no difficulty in implementing our methodology in this two-dimensional situation (once again, our default values are $\eta = 0.01$ and $\delta = 10^{-5}$). We show in Fig. 4(a) and (b) the uncontrolled and controlled trajectory evolution respectively; this implementation incurs a cost of $G = 0.0317$, and the target error $E(T) = 0.0069$. As a

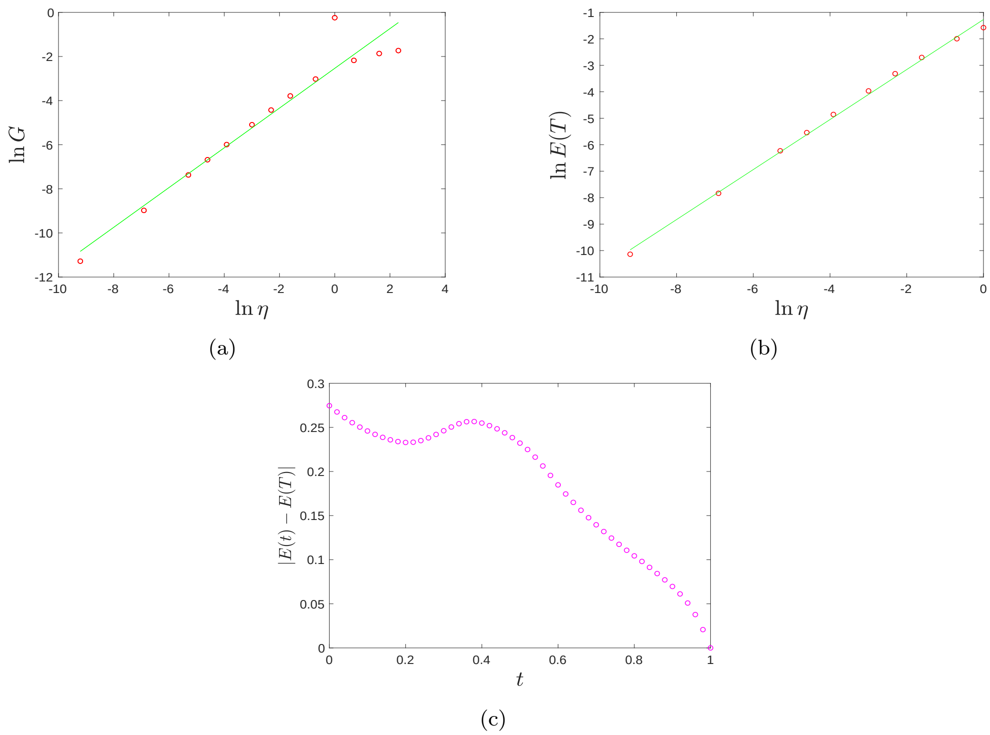


Fig. 2. *Analysis for the optimal control associated with Fig. 1: (a) dependence of cost on* η*, (b) decay of* $E(T)$ *as* η *is reduced, and (c) error decay as per Theorem 2.2.*

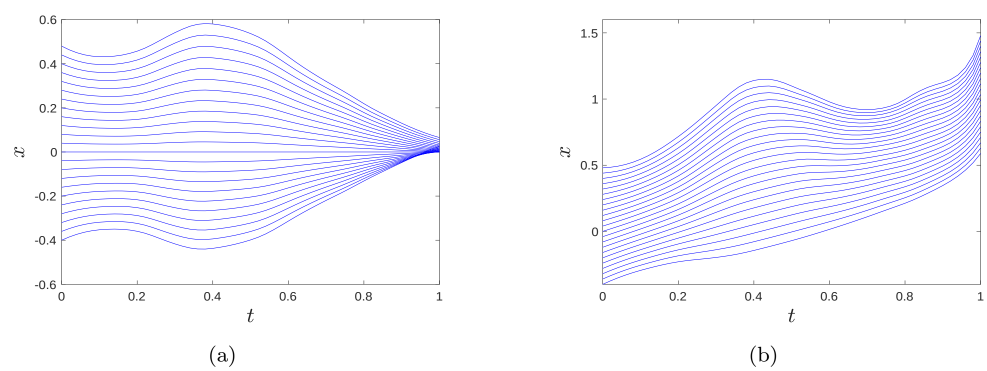


Fig. 3. *Different target functions applied to (3.1). (a)* $\Theta(x_0) = x_0^2/4$ *(*$G = 0.00134$*), and (b)* $\Theta(x_0) = x_0 + 1$ *(*$G = 0.00477$*).*

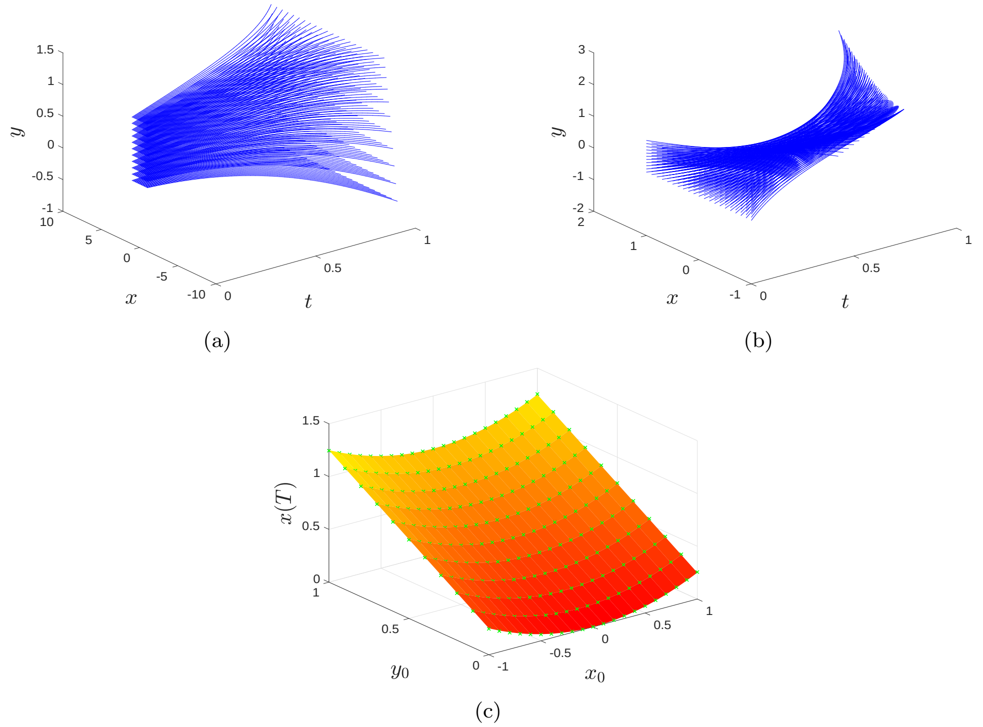


FIG. 4. *(a) Flow associated with (3.2); (b) the controlled flow from our algorithm subject to the target function (3.3); (c) target x-value surface [orange], and achieved values [green crosses].*

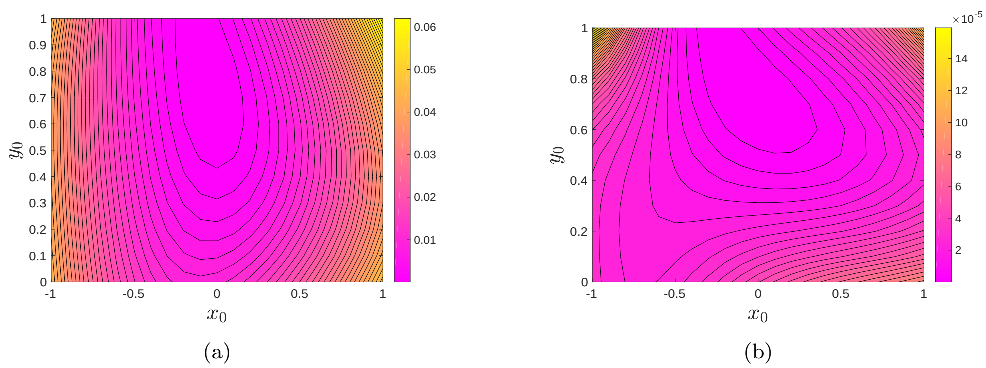
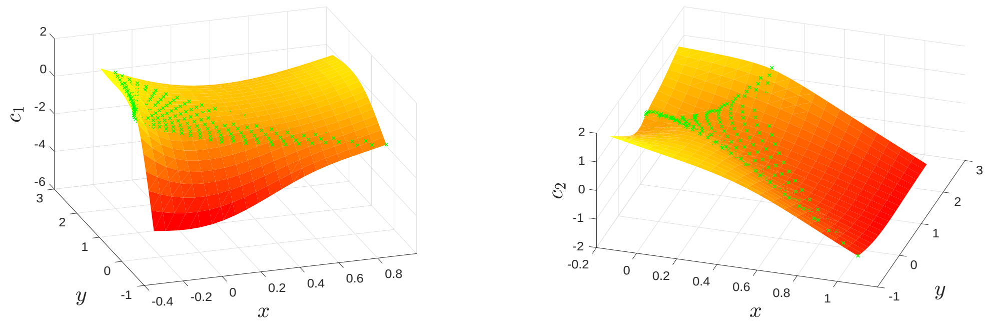


FIG. 5. *The distribution of contributions to (a) the cost G, and (b) the total error $E(T)$, over points $(x_0, y_0) \in \Omega$ for the two-dimensional example.*

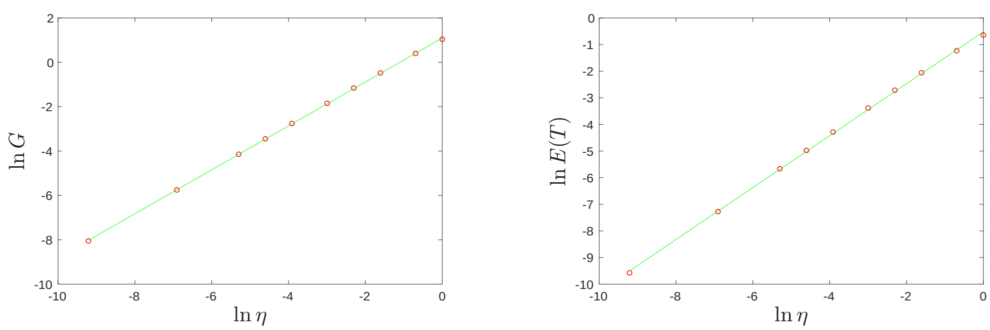
sample to demonstrate the achievement of the target locations, we shown in Fig. 4(c) how the controlled final location (green crosses) is close to the target x-coordinate (orange surface) for $(x_0, y_0) \in \Omega_0$.

The total cost G and final target error $E(T)$ are composed by integrating over Ω_0. The distribution of the contributions to each of these integrals with $(x_0, y_0) \in \Omega_0$ are shown in Fig. 5. The largest contributions to each of these occurs along the sides of Ω_0; this is since the middle regions require the least effort to control for this chosen



(a) (b)

FIG. 6. *Computed control function* $\mathbf{c} = (c_1, c_2)$ *illustrated by green crosses, with the orange surfaces indicating its approximant: (a)* $c_1(x, y, t = 0.58)$ *and (b)* $c_2(x, y, t = 0.78)$*. Different viewing angles are used for better visibility.*



(a) (b)

FIG. 7. *The variation of (a) the cost* G *and (b) the error* $E(T)$ *with* η *for the two-dimensional example.*

target function.

Visualizing the control $\mathbf{c} = (c_1, c_2)$ as a function of (x, y, t) requires higher-dimensions. Instead, in Fig. 6, we show computed values of c_1 and c_2 at different instances in time. The green crosses are the computed values based on our algorithm for single-trajectory optimal control, while the orange surface indicates the result of applying `gridfit` [20] to obtain approximating functions. While visualizing the full control is difficult, the computations did not present any significant difficulty.

Finally, we validate $\eta \to 0$ behavior in Fig. 7. The fitted regression lines (green) give the facts that $G \sim \eta^{0.990}$ and $E(T) \sim \eta^{0.975}$. Thus, both the cost G and the final error $E(T)$ demonstrate the behavior as intimated in Theorem 2.3 with $\alpha \approx 1$.

3.3. ABC flow. Having validated our theorems in several elementary flows, we next investigate the flow associated with an exact solution to the three-dimensional steady Euler equations of fluid motion: Arnold-Beltrami-Childress (ABC) flow, whose velocity field is given by [3, 21]

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} v_1(x,y,z) \\ v_2(x,y,z) \\ v_2(x,y,z) \end{pmatrix} = \begin{pmatrix} A\sin z + C\cos y \\ B\sin x + A\cos z \\ C\sin y + B\cos x \end{pmatrix}, \tag{3.4}$$

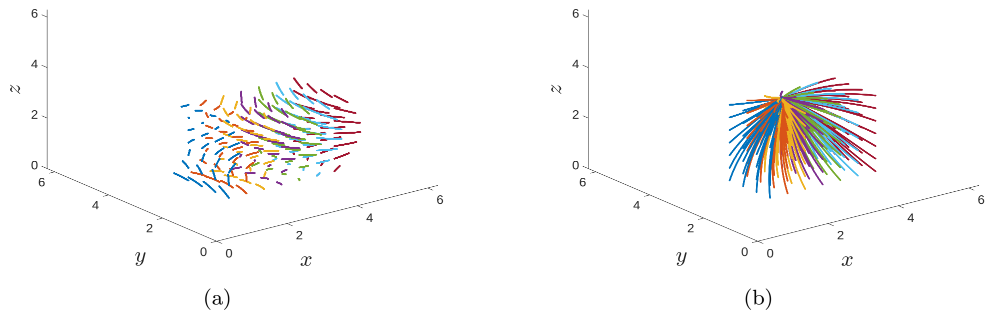


FIG. 8. *ABC-flow trajectories from time* 0 *to* 0.4*: (a) uncontrolled, and (b) controlled using* $\eta = 0.00001$*, all with a target destination* (π, π, π) *for all trajectories.*

where $\mathbf{x} = (x, y, z)$. When considered on the cell $[0, 2\pi) \times [0, 2\pi) \times [0, 2\pi)$ with triply-periodic boundary conditions, the resulting trajectories are well-known to be chaotic [21]; Arnold's criterion for generic integrability of trajectories arising from steady Euler flow [3] fails in this instance because the velocity and vorticity fields are collinear. The ABC velocity field (3.4) is also an exact solution to the Navier-Stokes equation under a particular choice of body force [21]. We use the parameter values $A = 1$, $B = 2/3$ and $C = 1/3$ (also considered in [21]) for our simulations. In the spirit of chaos control [42, 22, 51, 55], we seek here to make trajectories all approach the same final destination (π, π, π).

In using our algorithm, since the problem is spatially three-dimensional, we need to use a seven-point stencil at each point $\mathbf{p}$ (the central point, plus points adjacent to this in all three coordinate directions) in conjugate momentum space to estimate the gradient of the flow map with respect to $\mathbf{p}$, and then have to invert the 3×3 matrix in the Newton-Raphson step. Additionally, we require the usage of the higher-dimensional `regularizeNd` [47] rather than `gridfit` [20] in determining the control velocity globally. We demonstrate in Fig. 8(a) the trajectories in (x, y, z)-space for a grid of initial conditions, evolved from time 0 to $T = 0.4$ using the ABC velocity (3.4), using an Euler method with $\Delta t = 0.001$. Our control algorithm is then applied with the identical time-spacing, and with $\eta = 0.0001$, thereby desiring the achievement of the target at a high level of accuracy. The controlled trajectories are displayed in Fig. 8(b) with each trajectory shown in a different color. All trajectories are seen to approach (π, π, π) as required.

The control velocity $\mathbf{c} = (c_1, c_2, c_3)$ has three components, with each component being a function of (x, y, z, t). Illustratubg the computed control in a complete way is therefore difficult. We show several time-slices, in several $z =$ constant planes, for one of the components in Fig. 9. These are shown as contour fields. We note that since these are computed based on where trajectories are at each time-instance (i.e., from the trajectory data in Fig. 8(b)), we can only obtain reliable nformation in sets which are within a convex domain of the existing data points. That is, extrapolation in the (xy)-plane beyond the available data points is unreasonable. Hence, the information at each time is confined to the current locations of the controlled trajectories. At each time-frame, for the demonstration of the control $c_2(x, y, z, t)$ in Fig. 9, we choose a z-plane which is exactly in the middle of the z-range of all the current trajectory

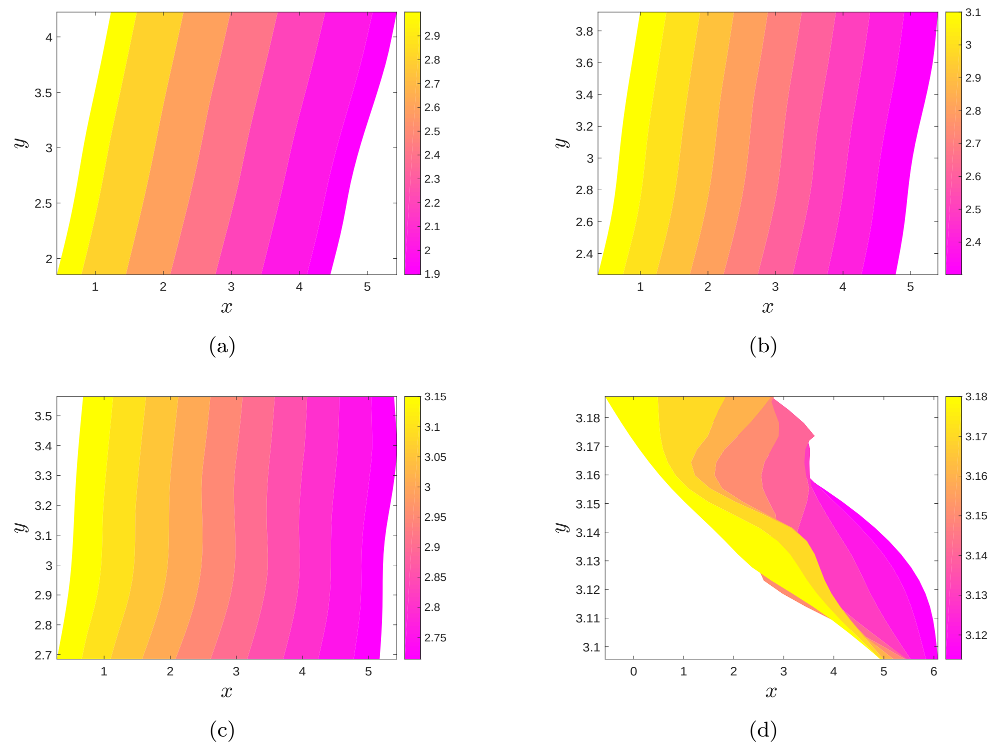


Fig. 9. *Displaying the control velocity component $c_2(x, y, z, t)$ for the ABC-flow control at several time- and z-slices: (a) $t = 0.1$ and $z = 2.3106$, (b) $t = 0.2$ and $z = 2.599$, (c) $t = 0.3$ and $z = 2.869$, and (d) $t = 0.4$ and $z = 3.139$.*

locations. At $t = 0.4$, since the trajectories have all closely approached (π, π, π), information is only available in a small neighborhood near this.

3.4. A Navier-Stokes data example. Finally, we demonstrate applicability when velocities are genuinely given by data, by generating them from a computational fluid dynamics simulation of the Navier-Stokes equations. The spatial domain $[0, 1] \times [0, 1]$ is used, with periodic boundary conditions in both directions. The Reynolds number is moderate at 5000, and 100 equally spaced intervals are used in each direction to define a spatial grid. The Navier-Stokes equations are solved in this case using the vorticity formulation, with a specified forcing function and a randomly generated initial vorticity distribution. A pseudo-spectral code is used: discrete Fourier transforms in space, and a Crank-Nicholson algorithm in time, with $\Delta t = 0.01$. The equations are numerically solved from an initial time $t = 0$ to a final time $T = 2$. Thus, the two components of the velocity field $\mathbf{v} = (v_1, v_2)$ are generated on a spatiotemporal grid. To get a sense of the computed velocity, we show in Fig. 10 the components v_1 and v_2 at a couple of instances in time.

The velocity data from the Navier-Stokes simulation is then stored, and used as input into a spatiotemporal control strategy. We take an equally-spaced grid of 25 initial points (x_0, y_0), and first plot their evolution under the uncontrolled unsteady velocity data in Fig. 11(a). Our control aim in this instance is to have these approach

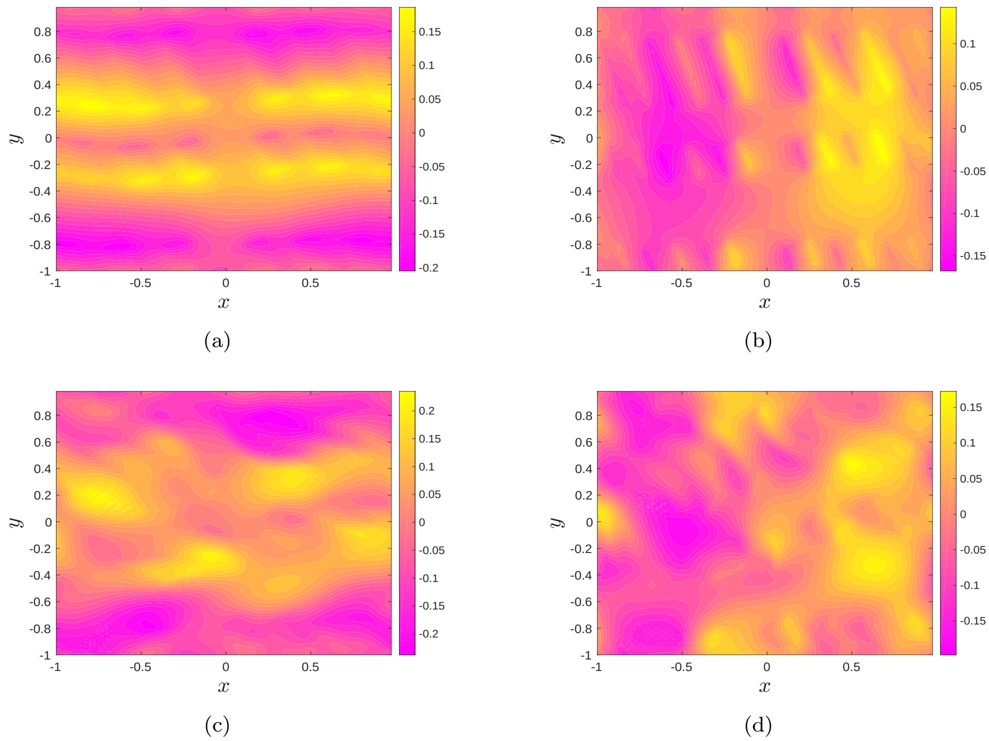


FIG. 10. *The velocity components v_1 (left) and v_2 (right), computed at times* 0.5 *(top row) and* 2.0 *(bottom row), as generated from the Navier-Stokes computational fluid dynamics solver.*

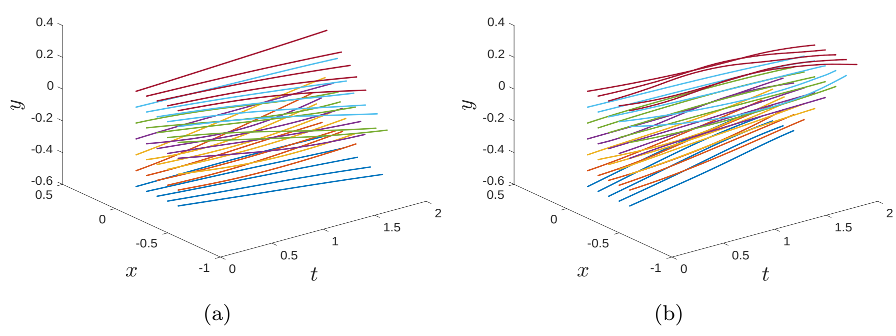


FIG. 11. *The (a) uncontrolled, and (b) controlled, trajectories from the Navier-Stokes example.*

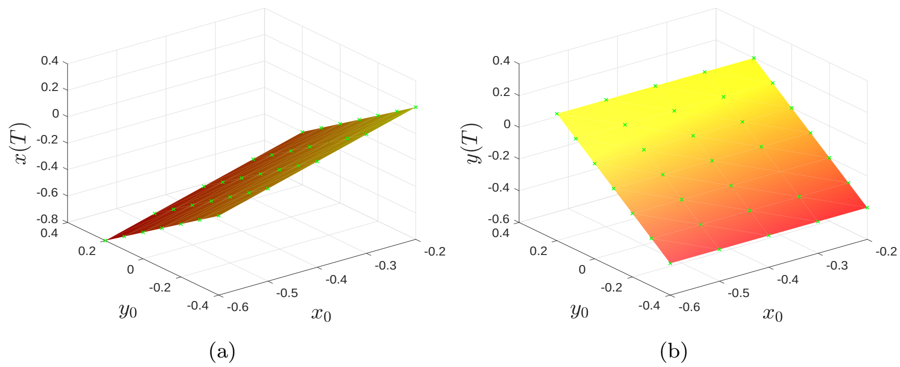


FIG. 12. *The (a) x-coordinate, and (b) the y-coordinate, of the controlled trajectories at the final time $T = 2$ (green crosses), along with the relevant target surface (3.5) (orange planes).*

the target destination function

$$\Theta(\mathbf{x}_0) = \begin{pmatrix} x_0 - y_0 \\ y_0 \end{pmatrix} . \tag{3.5}$$

by the final time ($T = 2$), and we choose $\eta = 0.0001$. Applying the methodology that we have described is now straightforward. Even though the velocity is given purely in terms of data on a discrete grid, it is possible to approximate quantities such as $\nabla \mathbf{v}$ (as needed for implementation of (2.6) by standard finite-differencing, and interpolating as needed when trajectories are off the grid. The controlled trajectories derived from this process are shown in Fig. 11(b). To verify that the desired targets have been achieved, in Fig. 12 we illustrate with green crosses the x- and y-coordinates of the final locations as functions of the initial location (x_0, y_0). The planes displayed are the exact target functions given in (3.5). Clearly, the targets have been achieved to excellent accuracy. It turns out that the global error $E(T)$ in (2.10) is 0.00184, and the total cost (2.3) is 4.26×10^{-6}.

Finally, we demonstrate the computed control velocity $\mathbf{c}(x, y) = (c_1, c_2)$ at several intermediate time-instances in Fig. 13. As before, the green crosses indicate the computed value of the control velocity from the integration along trajectories, while the orange surface (the global control velocity) is obtained by applying the `gridfit` technique. In all cases, the (x, y) domain is automatically limited here to the spatial regions the relevant trajectories traverse, and not the full domain of the Navier-Stokes simulation (which would entail spurious extrapolation). We have thus demonstrated the applicability of our optimal control technique to computational fluid dynamics data as well.

4. Discussion and conclusions. By combining and adapting different techniques (Hamiltonian formulation of optimal control, Newton-Raphson method, approximating surfaces using `gridfit` [20] and/or `regularizeNd` [47] and the applied analysis of differential equations), we have developed a methodology for determining a spatiotemporal optimal control function for a finite-horizon globally-specified target achievement. With this specific focus in mind, the cost function we chose takes the most amenable convex form (2.3); however, the algorithm we propose can be extended to more general forms of G. For this cost function, we were able to provide theoretical estimates which show the rate of decay of the error, and the comparative

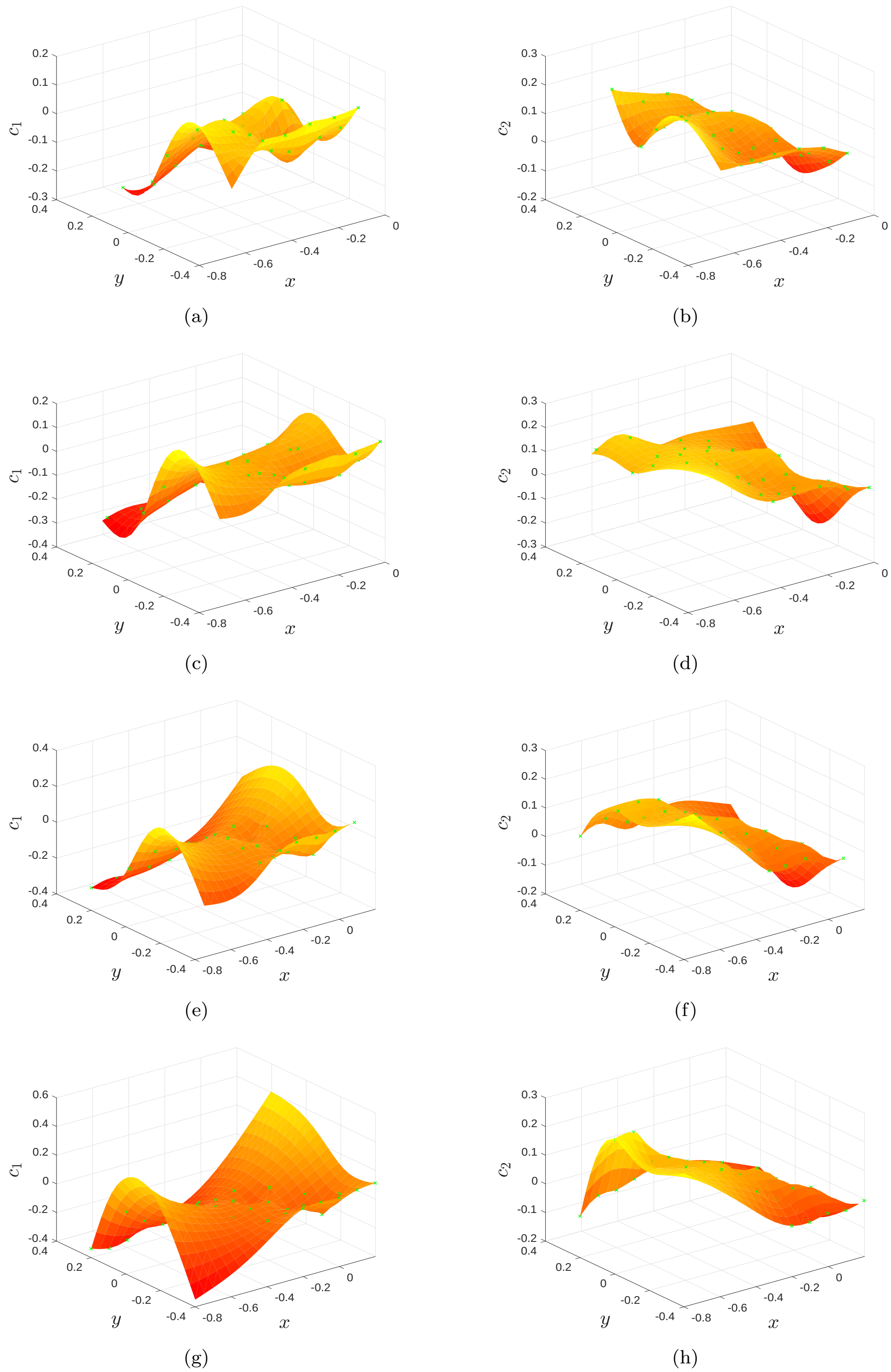


FIG. 13. *The control velocity components* c_1 *(left) and* c_2 *(right), computed at times* 0.5, *1.0* 1.5 *and* 2.0 *(in order of rows). The green crosses are computed from the controlled trajectories, while the orange surface extends these globally using* `gridfit`.

η-dependence on the error and cost, in elementary ways, avoiding elaborate functional analytic arguments.

We have highlighted that our formulation is particularly relevant to fluid mechanical systems in which $\mathbf{v}$ is a observed/computationally-determined velocity field, and then the spatiotemporal control $\mathbf{c}$ will be something physically realizable by imposing additional flow conditions such as boundary vibration [19, 58], or sinks/sources positioned at strategic locations [11, 53, 46, 60, 29]. Often, fluid mixing is to be controlled or enhanced by pushing fluid trajectories in some specified way over a time duration; our assumptions in this paper (i.e., that Θ is specified) is particularly suited to this situation. Significant future applications of this method in fluid mechanical systems is therefore anticipated.

Controlling the Navier-Stokes equations of fluid mechanics is mature research field (see the reviews [11, 31]), in which principal difficulties arise in the infinite-dimensionality of the control problem, finite-dimensional projections also being of sufficiently large dimension to make the control procedure computationally expensive, highly turbulent situations requiring highly resolved information, and unpredictability over longer time-horizons. Generally, the task is to control the Eulerian velocity by limiting its 'turbulence level' as measured in terms of its gradients, vorticity, enstropy, etc. Controllability is usually via the boundary, thereby restricting the nature of the control. Our approach is different, instead targeting the eventual Lagrangian locations of trajectories, while seeking a spatiotemporally distributed control velocity. Consequently, our control problem has a dimensionaliity equal to that of the physical space in which the fluid resides (i.e., no more than three), allowing the effective usage of a Hamiltonian formulation of optimal control. We recover the spatiotemporal nature of the control velocity by using an approximant based on the control algorithm applied to an ensemble of trajectories. Of course, we would expect the method to face greater difficulties when the turbulence or the time-horizon is large; these require higher resolutions spatially and temporally.

Our framework can also be thought of as an interesting approach for controlling chaotic systems which may be autonomous or nonautonomous [10, 59, 63]. We have the ability to steer trajectories *globally* over some finite time using our method. We have demonstrated the application of this to an example from fluid mechanics—ABC flow. Thus, this provides a contribution to chaos control theory which is different from standard ones such as chaotic synchronization [50, 12, 36] and local control near chaotic saddles [23, 26]. The smoothness our theorems require in $\mathbf{v}$ is consonant with chaotic systems; the unpredictability of corresponding Lagrangian trajectories because of sensitivity to initial conditions is apparently not an impediment to our theory and algorithm. As in the turbulent flow situation, the difficulty will be that the control velocity would need to be specified on finer and finer scales, and the control will be achievable for times which are not too large. We also note that the criteria we have developed apply even for nonsmooth Θ allowing, for example, the separation of trajectories into specified clusters. Thus, we expect this methodology to be a promising new approach for chaos control.

The numerical simulations we presented in Section 3 demonstrated the power of the method. We have illustrated the usage in both analytically-defined velocities, and velocities on a spatio-temporal grid obtained from a computational fluid dynamics simulation of the Navier-Stokes equation. While we showed one- two- and three-dimensional examples, the method works in any spatial dimension. However, the computational complexity does increase with the dimension, rendering the method impractical in large dimensions. Moreover, in instances in which the initial velocity field is highly turbulent, the presence of large velocity gradients will mean that the control velocities may become difficult to compute. Put another way, highly turbulent situations will have large $\|\boldsymbol{\cdot}\|_b$ norms in the velocity fields, and thus our theorems which provide decay rates and rubustness of the optimal control methodology have less value because the size of this norm is relevant. There is also a subtle issue which requires further exploration: the implicit assumption that an optimal control $\mathbf{c}$ *exists as a function of* $(\mathbf{x}, t)$. Should different trajectories give different predictions for $\mathbf{c}$ at a point of intersection of spacetime curves, determining $\mathbf{c}$ as a genuine spatiotemporal function becomes problematic. We plan to explore this issue, and seek an alternative formulation which is both analytically and physically reasonable, in future work. Additionally, we are seeking improvements in computational efficiency of the algorithm, for improved performance on densely defined and/or higher-dimensional data.

Appendix A. Proofs of theorems.

Here, we provide the proofs of the theorems of Section 2.

A.1. Proof of Theorem 2.1. This is a result emerging from classical optimal control theory [39], which works even when the evolution law is nonautonomous. We first use the following standard result (e.g., see Sections 2.5–2.6 in [39]), and written here in our notation. The notation $\nabla_{\mathbf{y}}$ represents the $n \times n$ matrix derivative with respect to the variable $\mathbf{y} \in \mathbb{R}^n$.

THEOREM A.1 (Nonautonomous optimal control). *Consider for* $\mathbf{x} \in \mathbb{R}^n$ *a system*

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{c}, t\right) \quad ; \quad t \in [0, T]$$

in which $\mathbf{c}$ *is the control, and the optimization of a quantity*

$$g = h_1\left(\mathbf{x}(T)\right) + \int_0^T h_2\left(\mathbf{x}(t), \mathbf{c}, t\right) \, \mathrm{d}t$$

is sought. Upon definition of the Hamiltonian

$$H(\mathbf{x}, \mathbf{c}, \mathbf{p}, t) := h_2(\mathbf{x}, \mathbf{c}, t) + \mathbf{f}\left(\mathbf{x}, \mathbf{c}, t\right)^\top \mathbf{p}\,,$$

a necessary condition for $\mathbf{c}$ *to be a local optimizer of* g *is* $\nabla_{\mathbf{c}} H = 0$*, in which* $\mathbf{x} = \mathbf{x}(t)$ *and* $\mathbf{p} = \mathbf{p}(t)$ *are solutions to the system*

$$\left.\begin{array}{rcl} \dot{\mathbf{x}} & = & \nabla_{\mathbf{p}} H \\ \dot{\mathbf{p}} & = & -\nabla_{\mathbf{x}} H \end{array}\right\}, \text{ where } \left.\begin{array}{rcl} \mathbf{x}(0) & = & \mathbf{x}_0 \\ \mathbf{p}(T) & = & \nabla_{\mathbf{x}} h_1(\mathbf{x}(T)) \end{array}\right\}.$$

Moreover, the solution corresponds to a minimizer if $\frac{\partial^2}{\partial c^2} H$ *is positive definite.*

To prove Theorem 2.1, we apply Theorem A.1 with the choice $\mathbf{f}\left(\mathbf{x}, \mathbf{c}, t\right) = \mathbf{v}\left(\mathbf{x}, t\right) + \mathbf{c}$, $h_1(\mathbf{x}) = \left\|\mathbf{x} - \Theta(\mathbf{x}_0)\right\|^2$ and $h_2(\mathbf{x}, \mathbf{c}, t) = \eta \left\|\mathbf{c}\right\|^2$. Then, the Hamiltonian is

$$H(\mathbf{x}, \mathbf{c}, \mathbf{p}, t) = \eta \left\|\mathbf{c}\right\|^2 + \left(\mathbf{v}(\mathbf{x}, t) + \mathbf{c}\right)^\top \mathbf{p}\,.$$

The condition $\nabla_{\mathbf{c}} H = 0$ yields $\eta 2\mathbf{c} + \mathbf{p} = 0$, and thus $\mathbf{c}\left(\mathbf{x}(\mathbf{x}_0, t), t\right) = -1/(2\eta)\mathbf{p}(t)$. Now, since

$$\nabla_{\mathbf{x}} H = \left[\nabla_{\mathbf{x}} \mathbf{v}\right]^\top \mathbf{p} \quad \text{and} \quad \nabla_{\mathbf{p}} H = \mathbf{v}(\mathbf{x}, t) + \mathbf{c}\,,$$

the differential equations (2.6) emerge immediately. Moreover, $\nabla_{\mathbf{x}} h_1\left(\mathbf{x}(T)\right) = 2\left(\mathbf{x}(T) - \Theta(\mathbf{x}_0)\right)$, ∎ which gives the end condition for $\mathbf{p}$ in (2.7). To establish that this critical $\mathbf{c}$ corresponds to a minimizer of g, we observe that $\frac{\partial^2}{\partial c^2} H = 2\eta \mathbb{I}$, where $\mathbb{I}$ is the $n \times n$ identity matrix. Since $\eta > 0$, this is positive definite. (More simply, the convexity of H in c in fact ensures that this is a global minimizer.)

A.2. Proof of Theorem 2.2. By taking the t-derivative of $E(t)^2$, we get

$$\frac{d}{dt}\left[E(t)^2\right] = 2\int_{\Omega_0} \frac{d}{dt}\left[\mathbf{x}(\mathbf{x}_0,t)\right]^\top \left[\mathbf{x}(\mathbf{x}_0,t) - \Theta(\mathbf{x}_0)\right] \mathrm{d}\mathbf{x}_0$$

Now, using the fact that $(d/dt)\mathbf{x} = \mathbf{v} + \mathbf{c}$, and subsequently applying the Cauchy-Schwarz inequality on the right-hand side, we get

$$\begin{aligned}
\left|\frac{d}{dt}\left[E(t)^2\right]\right| &\leq 2\left(\int_{\Omega_0} \|\mathbf{v}+\mathbf{c}\|^2 \mathrm{d}\mathbf{x}_0\right)^{1/2} E(t) \\
&\leq 2\left(2\int_{\Omega_0} \|\mathbf{v}\|^2 \mathrm{d}\mathbf{x}_0 + 2\int_{\Omega_0} \|\mathbf{c}\|^2 \mathrm{d}\mathbf{x}_0\right)^{1/2} E(t) \\
&\leq 2\sqrt{2}\left[\left(\int_{\Omega_0} \|\mathbf{v}\|^2 \mathrm{d}\mathbf{x}_0\right)^{1/2} + \left(\int_{\Omega_0} \|\mathbf{c}\|^2 \mathrm{d}\mathbf{x}_0\right)^{1/2}\right] E(t) \\
&\leq 2\sqrt{2}\left[A\sqrt{\mu(\Omega_0)} + \left(\int_{\Omega_0} \|\mathbf{c}\|^2 \mathrm{d}\mathbf{x}_0\right)^{1/2}\right] E(t)\,. \qquad \text{(A.1)}
\end{aligned}$$

In the above, we have suppressed the arguments $(\mathbf{x}(\mathbf{x}_0,t),t)$ in both $\mathbf{v}$ and $\mathbf{c}$ for brevity, and at the last step used the bound on $\|\mathbf{v}\|_a$. Now from Theorem 2.1, for any fixed $\mathbf{x}_0$, we know that $\mathbf{c} = -\mathbf{p}/(2\eta)$ with $\mathbf{p}$ obeying (2.6) with condition for $\mathbf{p}(T)$ given in (2.7). Using the abuse of notation $\mathbf{c}(t) := \mathbf{c}\left(\mathbf{x}(\mathbf{x}_0,t),t\right)$, this means that

$$\dot{\mathbf{c}} = -\left[\nabla\mathbf{v}\left(\mathbf{x}(\mathbf{x}_0,t),t\right)\right]^\top \mathbf{c}$$

subject to the condition $\mathbf{c}(T) = -\left[\mathbf{x}\left(\mathbf{x}_0,T\right)\right) - \Theta(\mathbf{x}_0)\right]/\eta$. We rewrite this in a new independent variable $\tau = T - t$, and let $\hat{\mathbf{c}}(\tau) = \mathbf{c}(t)$. Setting $L(\tau) := \left[\nabla\mathbf{v}\left(\mathbf{x}(\mathbf{x}_0,t),t\right)\right]^\top$, we have

$$\frac{\partial}{\partial\tau}\hat{\mathbf{c}} = L(\tau)\hat{\mathbf{c}} \ ; \ \hat{\mathbf{c}}(0) = -\frac{\mathbf{x}\left(\mathbf{x}_0,T\right) - \Theta(\mathbf{x}_0)}{\eta}\,.$$

Premultiplying the differential equation above by $\hat{\mathbf{c}}^\top$, we obtain

$$\frac{1}{2}\frac{\partial}{\partial\tau}\|\hat{\mathbf{c}}\|^2 = \hat{\mathbf{c}}^\top L(\tau)\hat{\mathbf{c}}\,,$$

and consequently

$$\begin{aligned}
\frac{\partial}{\partial\tau}\|\hat{\mathbf{c}}\|^2 &\leq 2\left\|\hat{\mathbf{c}}^\top\right\| \|L(\tau)\hat{\mathbf{c}}\| \\
&\leq 2\left\|\hat{\mathbf{c}}^\top\right\| B \|\hat{\mathbf{c}}\| = 2B\|\hat{\mathbf{c}}\|^2
\end{aligned}$$

using the bound on $\|\mathbf{v}\|_b$. Separating variables and integrating from $\tau = 0$ to a general τ value in $[0,T]$, we have

$$\ln\frac{\|\hat{\mathbf{c}}(\tau)\|^2}{\|\hat{\mathbf{c}}(0)\|^2} \leq 2B\tau\,,$$

and applying the value of $\hat{\mathbf{c}}(0)$ we acquire the bound

$$\|\hat{\mathbf{c}}(\tau)\|^2 \leq \frac{\|\mathbf{x}(\mathbf{x}_0,T) - \Theta(\mathbf{x}_0)\|^2}{\eta^2} e^{2B\tau}\,.$$

Reverting to $t \in [0,T]$ as the independent variable, this means that

$$\|\mathbf{c}(t)\|^2 \leq \frac{\|\mathbf{x}(\mathbf{x}_0,T) - \Theta(\mathbf{x}_0)\|^2}{\eta^2} e^{2B(T-t)}\,. \qquad \text{(A.2)}$$

Inserting this bound into (A.1) yields

$$\left|\frac{d}{dt}\left[E(t)^2\right]\right| \leq 2\sqrt{2}\left[A\sqrt{\mu(\Omega_0)} + \frac{E(T)e^{B(T-t)}}{\eta}\right]E(t)\,.$$

This means that

$$\left|\frac{d}{dt}\left[E(t)\right]\right| \leq \sqrt{2}\left[A\sqrt{\mu(\Omega_0)} + \frac{E(T)e^{B(T-t)}}{\eta}\right], \tag{A.3}$$

and integrating from a general time t to T results in

$$E(T)-E(t) \leq \sqrt{2}\left[A\sqrt{\mu(\Omega_0)}(T-t) - \frac{E(T)\left(1-e^{B(T-t)}\right)}{B\eta}\right].$$

Similarly working with the fact that $(d/dt)E(t)^2$ is greater than negative the term on the right of (A.3) enables

$$E(T)-E(t) \geq -\sqrt{2}\left[A\sqrt{\mu(\Omega_0)}(T-t) - \frac{E(T)\left(1-e^{B(T-t)}\right)}{B\eta}\right].$$

Combining these two results gives us the required equation (2.13).

A.3. Proof of Theorem 2.3. We can write (2.3) as

$$G = E(T)^2 + \eta\int_{\Omega_0}\int_0^T \left\|\mathbf{c}\left(\mathbf{x}(\mathbf{x}_0,t),t\right)\right\|^2\,\mathrm{d}t\,\mathrm{d}\mathbf{x}_0\,. \tag{A.4}$$

Using (A.2) we have

$$\int_{\Omega_0}\int_0^T \left\|\mathbf{c}\left(\mathbf{x}(\mathbf{x}_0,t),t\right)\right\|^2\,\mathrm{d}t\,\mathrm{d}\mathbf{x}_0 \leq \frac{E(T)^2(e^{2BT}-1)}{2B\eta^2},$$

and so

$$G \leq E(T)^2\left[1 + \frac{e^{2BT}-1}{2B\eta}\right].$$

Now if $E(T) = \mathcal{O}(\eta^\alpha)$ for some $\alpha > 1/2$, then

$$G \leq \mathcal{O}(\eta^{2\alpha})\left[1 + \frac{e^{2BT}-1}{2B\eta}\right] = \mathcal{O}(\eta^{2\alpha-1})$$

as required.

A.4. Proof of Theorem 2.4. If we consider the initial system (2.1) with $\tilde{\mathbf{v}}$ instead of $\mathbf{v}$, the procedure outlined will then generate a control $\tilde{\mathbf{c}}$, a global error $\tilde{E}(t)$ at a general time t, a global error $\tilde{E}(T)$ at the final time T, and the minimizing cost $\tilde{G}$. Now, since $\tilde{\mathbf{v}}$ is $\mathcal{O}(\epsilon)$-close to $\mathbf{v}$ in both the norms $\|\bullet\|_a$ and $\|\bullet\|_b$, this Theorem is a simple consequence that *every* step in the previous theorems inherits this closeness because Ω_0 is bounded, and the time-interval $[0,T]$ over which integration is performed is finite. Specifically, $\tilde{\mathbf{x}}(\mathbf{x}_0,t)$ and $\tilde{\mathbf{p}}$ must be $\mathcal{O}(\epsilon)$-close to the variables $\mathbf{x}(\mathbf{x}_0,t)$ and $\mathbf{p}$ generated via Theorem 2.1. Since $\tilde{\mathbf{c}} = -\tilde{\mathbf{p}}/(2\eta)$, this property transfers to $\tilde{\mathbf{c}}$. Then (2.10) ensures that $\tilde{E}(t)$ is $\mathcal{O}(\epsilon)$-close to $E(t)$, and furthermore, (A.4) ensures that $\tilde{G} = G + \mathcal{O}(\epsilon)$ as well. The decay expression in Theorem 2.2 also remains valid, with of course the replacements $A \to \tilde{A}$ and $B \to \tilde{B}$, thereby only perturbing the results by $\mathcal{O}(\epsilon)$.

REFERENCES

[1] M. Aghababa and H. Aghababa, *Finite-time stabilization of uncertain non-autonomous chaotic gyroscopes with nonlinear inputs*, Appl. Math. Mech., 33 (2012), pp. 155–164.

[2] M. P. Aghababa and H. P. Aghababa, *Adaptive finite-time synchronization of non-autonomous chaotic systems with uncertainty*, J. Comput. Nonlin. Dyn., 8 (2013), p. 031006.

[3] V. Arnol'd, *Sur la topologie des écoulements stationnaires des fluides parfaits*, C. R. Acad. Sci. Paris, 261 (1965), pp. 17–20.

[4] S. Balasuriya, *Optimal frequency for microfluidic mixing across a fluid interface*, Phys. Rev. Lett., 105 (2010), p. 064501.

[5] S. Balasuriya, *Unsteadily manipulating internal flow barriers*, J. Fluid Mech., 818 (2017), pp. 382–406.

[6] S. Balasuriya, *Stochastic sensitivity: a computable Lagrangian measure of uncertainty for unsteady flows*, SIAM Review, in press (2020).

[7] S. Balasuriya and M. Finn, *Energy constrained transport maximization across a fluid interface*, Phys. Rev. Lett., 108 (2012), p. 244503.

[8] S. Balasuriya, N. T. Ouellette, and I. I. Rypina, *Generalized Lagrangian coherent structures*, Physica D., 372 (2018), pp. 31–51.

[9] S. Balasuriya and K. Padberg-Gehle, *Accurate control of hyperbolic trajectories in any dimension*, Phys. Rev. E, 90 (2014), p. 032903.

[10] V. Y. Belozyorov, *Universal approach to the problem of emergence of chaos in autonomous dynamical systems*, Nonlinear Dynamics, 95 (2019), pp. 579–595.

[11] T. Bewley, P. Moin, and R. Temam, *DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms*, J. Fluid Mech., 447 (2001), pp. 179–225.

[12] S. Boccaletti, J. Kurths, G. Osipov, D. Valladares, and C. Zhou, *The synchronization of chaotic systems*, Physics Reports, 366 (2002), pp. 1–101.

[13] O. Bokanowski, A. Briani, and H. Zidani, *Minimum time control problems for non-autonomous differential equations*, Syst. Control Lett., 58 (2009), pp. 742–746.

[14] A. E. Botha, I. Rahmonov, and Y. Shukrinov, *Spontaneous and controlled chaos synchronization in intrinsic Josephson junctions*, IEEE Trans. Appl. Superconductivity, 28 (2018), p. 1800806.

[15] T. Botmart, P. Niamsup, and X. Liu, *Synchronization of non-autonomous chaotic systems with time-varying delay via delayed feedback control*, Commun. Nonlinear Sci., 17 (2012), pp. 1894–1907.

[16] A. E. Bryson, *Applied optimal control: optimization, estimation and control*, Routledge, 2018.

[17] Y. Chen, X. Wu, and Z. Gui, *Global synchronization criteria for a class of third-order non-autonomous chaotic systems via linear state error feedback control*, Appl. Math. Model., 34 (2010), pp. 4161–4170.

[18] I. Couchman, E. Kerrigan, and J. Vassilicos, *Optimization-based feedback control of mixing in a stokes fluid flow*, in 2009 European Control Conference (ECC), IEEE, 2009, pp. 1227–1232.

[19] J. de Jong, R. Lammertink, and M. Wessling, *Membranes and microfluidics: a review*, Lab Chip, 6 (2006), pp. 1125–1139.

[20] J. D'Errico, *Surface fitting using gridfit, Matlab Central File Exchange*, 2016, https://au.mathworks.com/matlabcentral/fileexchange/8998-surface-fitting-using-gridfit (accessed 11 January 2019).

[21] T. Dombre, U. Frisch, J. Greene, M. Hénon, A. Mehr, and A. Soward, *Chaotic streamlines in the ABC flows*, J. Fluid Mech., 167 (1986), pp. 353–391.

[22] O. E, C. Grebogi, and J. Yorke, *Controlling chaos*, Phys. Rev. Lett., 64 (1990), pp. 1196–1199.

[23] M. El-Dessoky, M. Yassen, and E. Aly, *Bifurcation analysis and chaos control in Shimizu–Morioka chaotic system with delayed feedback*, Applied Math. Comp., 243 (2014), pp. 283–297.

[24] G. Froyland and N. Santitissadeekorn, *Optimal mixing enhancement*, SIAM J. Appl. Math., 77 (2017), pp. 1444–1470.

[25] T. Glad and L. Ljung, *Control theory*, CRC press, 2014.

[26] E. F. D. Goufo, M. Mbehou, and M. M. K. Pene, *A peculiar application of Atangana–Baleanu fractional derivative in neuroscience: Chaotic burst dynamics*, Chaos, Solitons & Fractals, 115 (2018), pp. 170–176.

[27] R. Hajiloo, H. Salarieh, and A. Alasty, *Chaos control in delayed phase space constructed by the takens embedding theory*, Commun, Nonlin. Sci. Numer. Simul., 54 (2018), pp. 453–465.

[28] G. Haller, *Lagrangian coherent structures*, Annu. Rev. Fluid Mech., 47 (2015), pp. 137–162.
[29] C. Hernandez, Y. Bernard, and A. Razek, *A global assessment of piezoelectric actuated micro-pumps*, Eur. Phys. J. Appl. Phys., 51 (2010), p. 20101.
[30] K. Hicke, X. Porte, and I. Fischer, *Characterizing the deterministic nature of individual power dropouts in semiconductor lasers subject to delayed feedback*, Phys. Rev. E, 88 (2013), p. 052904.
[31] M. Hinze and K. Kunisch, *Second-order methods for optimal control of time-dependent fluid flow*, SIAM J. Control Optim., 40 (2001), pp. 925–946.
[32] C.-M. Ho and Y.-C. Tai, *Micro-electro-mechanical systems (mems) and fluid flows*, Annu. Rev. Fluid Mech., 30 (1998), pp. 579–612.
[33] A.-C. Huang and Y.-C. Chen, *Adaptive multiple-surface sliding control for non-autonomous systems with mismatched uncertainties*, Automatica, 40 (2004), pp. 1939–1945.
[34] K. Iwamoto, Y. Suzuki, and N. Kasagi, *Reynolds number effect on wall turbulence: towards effective feedback control*, Int. J. Heat Fluid Flow, 23 (2002), pp. 678–689.
[35] J.-D. Jansen, O. H. Bosgra, and P. M. Van den Hof, *Model-based control of multiphase flow in subsurface oil reservoirs*, J. Process Contr., 18 (2008), pp. 846–855.
[36] J. Kabziński, *Synchronization of an uncertain Duffing oscillator with higher order chaotic systems*, Int. J. Applied Math. Comp. Sci., 28 (2018), pp. 625–634.
[37] H. B. Keller, *Numerical methods for two-point boundary-value problems*, Courier Dover Publications, 2018.
[38] J. Kim, *Control of turbulent boundary layers*, Phys. Fluids, 15 (2003), pp. 1093–1105.
[39] D. Lawden, *Analytical methods of optimization*, Scottish Academic Press, Edinburgh, 1975.
[40] D. Lee, W. Yoo, and S. Won, *An integral control for synchronization of a class of unknown non-autonomous chaotic systems*, Phys. Lett. A, 374 (2010), pp. 4231–4237.
[41] Z. Lin, J.-L. Thiffeaut, and C. Doering, *Optimal stirring strategies for passive scalar mixing*, J. Fluid Mech., 675 (2011), pp. 465–476.
[42] M. C. Mackey and L. Glass, *Oscillation and chaos in physiological control systems*, Science, 197 (1977), pp. 287–289.
[43] T. Medjo, R. Temam, and M. Ziane, *Optimal and robust control of fluid flows: some theoretical and computational aspects*, Appl. Mech. Rev., 61 (2008), p. 010802.
[44] S. Miah, M. M. Fallah, and D. Spinello, *Non-autonomous coverage control with diffusive evolving density*, IEEE T. Automat. Contr., 62 (2017), pp. 5262–5268.
[45] C. Miles and C. Doering, *A shell model for optimal mixing*, J. Nonlin. Sci., 28 (2018), pp. 2153–2186.
[46] N. Mishchuk, T. Heldal, T. Volden, J. Auerswald, and H. Knapp, *Microfluidic pump based on the phenomenon of electroosmosis of the second kind*, Microfluid. Nanofluid., 11 (2011), pp. 675–684.
[47] J. Nicholson, *regularizeNd, (matlab file exchange) (https://www.mathworks.com/matlabcentral/fileexchange/61436-regularizend)*, 2020.
[48] S. Ober-Blöbaum and K. Padberg-Gehle, *Multiobjective optimal control of fluid mixing*, Proc. Appl. Math. Mech., 15 (2015), pp. 639–640.
[49] Y. Orlov, *Finite time stability and robust control synthesis of uncertain switched systems*, SIAM J. Control Optim., 43 (2004), pp. 1253–1271.
[50] L. Pecora and I. Carrol, *Synchronization in chaotic systems*, Phys. Rev. Lett., 64 (1990), p. 8212.
[51] K. Pyragas, *Continuous control of chaos by self-controlling feedback*, Phys. Lett. A, 170 (1992), pp. 421–428.
[52] M. Rafikov and J. Balthazar, *On an optimal control design for Rössler system*, Phys. Lett. A, 333 (2004), pp. 241–245.
[53] S. S. Ravindran, *A reduced-order approach for optimal control of fluids using proper orthogonal decomposition*, Int. J. Numer. meth. Fl., 34 (2000), pp. 425–448.
[54] K. L. Schlueter-Kuck and J. O. Dabiri, *Coherent structure colouring: Identification of coherent structures from sparse data using graph theory*, J. Fluid Mech., 811 (2017), pp. 468–486.
[55] T. Shinbrot, C. Grebogi, J. A. Yorke, and E. Ott, *Using small perturbations to control chaos*, Nature, 363 (1993), p. 411.
[56] S. Strogatz, *Nonlinear dynamics and chaos*, Perseus Books Publishing, New York, 1994.
[57] S. Varghese, M. Speetjens, and R. Trieling, *Lagrangian transport and chaotic advection in two-dimensional anisotropic systems*, Transp. Porous Media, 119 (2017), pp. 225–246.
[58] C.-H. Wang and G.-B. Lee, *Automatic bio-sampling chips integrated with micro-pumps and micro-valves for disease detection*, Biosensors Bioelectronics, 21 (2005), pp. 419–425.
[59] J. Wang, Y. Li, S. Zhong, and X. Hou, *Analysis of bifurcation, chaos and pattern formation*

in a discrete time and space Gierer–Meinhardt system, Chaos, Solitons & Fractals, 118 (2019), pp. 1–17.

[60] P. Woias, *Micropumps – summarizing the first two decades*, Proc. SPIE, 4560 (2001), pp. 39–52.

[61] Y. Yang and X. Wu, *Global finite-time synchronization of a class of the non-autonomous chaotic systems*, Nonlinear Dynam., 70 (2012), pp. 197–208.

[62] Q. You, Q. Wen, J. Fang, M. Guo, Q. Zhang, and C. Dai, *Experimental study on lateral flooding for enhanced oil recovery in bottom-water reservoir with high water cut*, J. Petrol. Sci. Eng., 174 (2019), pp. 747–756.

[63] J. Zhou, W. Zhou, T. Chu, Y.-x. Chang, and M.-j. Huang, *Bifurcation, intermittent chaos and multi-stability in a two-stage Cournot game with R&D spillover and product differentiation*, Applied Math. Comp., 341 (2019), pp. 358–378.