Iterative Methods for Navier–Stokes Inverse Problems

Liam O'Connor[1,2], Daniel Lecoanet[1,2], Evan H. Anders[3], Kyle C. Augustson[1,2],
Keaton J. Burns[4], Geoffrey M. Vasil[5], Jeffrey S. Oishi[6], Benjamin P. Brown[7]

[1]*Department of Engineering Sciences and Applied Mathematics,*
Northwestern University, Evanston, IL, 60208 USA
[2]*Center for Interdisciplinary Exploration and Research in Astrophysics,*
Northwestern University, Evanston, IL, 60201 USA
[3]*Kavli Institute for Theoretical Physics, University of California Santa Barbara, Santa Barbara, CA, 93106 USA*
[4]*Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 02142 USA*
[5]*School of Mathematics, Edinburgh University, EH9 3FD, UK*
[6]*Department of Physics and Astronomy, Bates College, Lewiston, ME, 04240 USA and*
[7]*Department of Astrophysical and Planetary Sciences,*
University of Colorado Boulder, Boulder, CO, 80309 USA

Even when the partial differential equation underlying a physical process can be evolved forward in time, the retrospective (backward in time) inverse problem often has its own challenges and applications. Direct Adjoint Looping (DAL) is the defacto approach for solving retrospective inverse problems, but it has not been applied to deterministic retrospective Navier–Stokes inverse problems in 2D or 3D. In this paper, we demonstrate that DAL is ill-suited for solving retrospective 2D Navier–Stokes inverse problems. Alongside DAL, we study two other iterative methods: Simple Backward Integration (SBI) and the Quasi-Reversible Method (QRM). Our iterative SBI approach is novel while iterative QRM has previously been used. Using these three iterative methods, we solve two retrospective inverse problems: 1D Korteweg–de Vries–Burgers (decaying nonlinear wave) and 2D Navier–Stokes (unstratified Kelvin–Helmholtz vortex). In both cases, SBI and QRM reproduce the target final states more accurately and in fewer iterations than DAL. We attribute this performance gap to additional terms present in SBI and QRM's respective backward integrations which are absent in DAL.

I. INTRODUCTION

Consider a retrospective inverse problem where the variable of interest $\overline{X}$ is constrained by a partial differential equation (PDE) involving space x and time t:

$$\mathcal{F}[\overline{X}(x,t)] = 0. \tag{1}$$

We aim to determine the unique initial condition $\overline{X}(x,0)$ corresponding to a given final state $\overline{X}(x,t_f)$ where the final time t_f is known. These "retrospective inverse problems" are often ill-conditioned or ill-posed, even when the forward problem is readily solvable.

Despite potential difficulties, retrospective inverse problems arise in a wide variety of contexts. Scientists and engineers have documented numerous cases where retrospective inversion is feasible and practical. [1] inverted chemical reaction fronts governed by a 1D reaction-diffusion equation. [2] inverts tumor growth using biophysically-motivated constraints on the initial condition. [3] inverted supersonic supernova expansions governed by the 1D compressible Navier–Stokes equation. [4, 5] examined the history of buoyancy-driven flows in Earth's mantle by approximating palaeo temperature distributions as well as variable parameters such as the thermal diffusivity.

Although there are problem-specific details in their methodologies, [1–5] use the same optimal control technique known as Direct Adjoint Looping (DAL). Using DAL, they minimize the cost functional

$$\mathcal{J}_f^X \equiv \frac{1}{2}\left\langle \left| X'(x,t_f) \right|^2 \right\rangle = \frac{1}{2}\left\langle \left| X(x,t_f) - \overline{X}(x,t_f) \right|^2 \right\rangle, \tag{2}$$

where the angled brackets $\langle \cdot \rangle$ denote a spatial integral over the problem domain and the trial solution X has some deviation $X' \equiv X - \overline{X}$. They construct sequences of trial solutions, such as $\{X_n(x,t)\}$, where each element X_n is computed from a corresponding trial initial condition $X_n(x,0)$. Convergence to the target initial condition $\overline{X}(x,0)$ is seldom guaranteed because the user provides an initial guess $X_0(x,0)$.

Related 2D Navier–Stokes inverse problems have been studied analytically from several perspectives. [6] uses boundary conditions as control inputs (with a given initial condition) and verifies controllability of certain 2D fluid systems. Later, the existence of solutions to a more demanding retrospective Navier–Stokes inverse problem with final overdetermination was established by [7]. In [7], the authors do not attempt to solve their inverse problem. Instead they provide an analysis of the problem where a separable source term (as well as the initial condition) are left to be determined. Following this, the data-completion problem of determining an unknown viscosity was studied analytically by [8].

More recently, researchers extended these analytical studies of Navier–Stokes inverse problems by implementing data assimilation algorithms. [9] studies an inverse problem governed by the parabolized Navier–Stokes (PNS) equation. Following this, [10] studies data assimilation algorithms for non-deterministic Navier–Stokes

inverse problems. They demonstrate several advantages of using filters for numerical weather prediction. Yet more recently, [11–14] used neural networks to solve analogous Navier–Stokes data-completion problems. In particular, [14] compares DAL with a physics-informed neural network approach while solving inverse problems governed by the viscous Burgers equation as well as 2D incompressible Navier–Stokes.

In this paper, we will also compare the performance of several iterative methods (such as DAL) by applying them to a retrospective Navier–Stokes inverse problem. Various Navier–Stokes inverse problems have been studied analytically and numerically using optimal control. However, we will demonstrate that conventional optimal control methods are ill-suited for this class of inverse problems, even when a numerically-consistent dataset is available.

With DAL, we can minimize an arbitrary cost functional such as $\mathcal{J}_f^X$. First, we must solve the forward problem on $t : 0 \to t_f$. This forward problem is constrained by some PDE (such as Navier–Stokes) which we express generally as eqn 1. At this point, we can evaluate $\mathcal{J}_f^X$ and compare the trial $(X(x, t_f))$ and target $(\overline{X}(x, t_f))$ final states. Information from time $t = t_f$ is then propagated backward, to time $t = 0$, by solving the linear adjoint problem $\mathcal{F}^\dagger[\chi, X] = 0$ on $t : t_f \to 0$, where $\chi(x, t)$ denotes the adjoint variable. Backward integration of the adjoint equations allows us to compute the gradient or functional derivative of $\mathcal{J}_f^X$ with respect to $X(x, 0)$. Using this gradient, we adjust our trial initial condition by some amplitude γ: $X_{n+1}(x, 0) = X_n(x, 0) + \gamma\chi_n(x, 0)$.

In practice, researchers often customize the DAL algorithm to improve performance in specific cases. For example, [15] used the conjugate-gradient method with a discrete adjoint formulation to optimize initial conditions subject to norm constraints. For the PNS inverse problem, [9] combined Principle Orthogonal Decomposition (POD) with four-dimensional variational assimilation (4DVAR), an optimal control technique analogous to DAL. In both examples, researchers reduced the computational cost of each DAL iteration by reformulating the adjoint while simultaneously reducing the required number of DAL iterations by implementing quasi-Newton minimization routines. For this investigation, we take a different approach. Instead of reformulating the adjoint or changing the way in which gradient information is used, we will demonstrate that DAL's shortcomings in the context of Navier–Stokes inversion can be mitigated by appending the adjoint system with additional advective terms.

The additional advective terms not present in DAL are present in two existing nonlinear backward integration methods: Simple Backward Integration (SBI) and the Quasi-Reversible Method (QRM). SBI and QRM are designed to invert the target final state by approximating the (ill-posed) integration of the original equations back-

ward in time. SBI accomplishes this by reversing the sign of the problematic diffusion term, i.e. $a\partial_x^2 X \to -a\partial_x^2 X$. In contrast, QRM is a regularization framework which preserves the ill-posed diffusion term [16]. With QRM, we append a small hyperdiffusion term to the otherwise ill-posed constraint equation, i.e. $a\partial_x^2 X \to a(\partial_x^2 + \varepsilon\partial_x^4)X$ where $0 < \varepsilon \ll 1$.

We introduce these concepts as backward integration methods rather than iterative methods because their previous implementations were rarely iterated. SBI (which has never been iterated) was introduced by [17] to infer the history of Earth's convective mantle in the viscinity of the southern African superswell. Following this, [4] and [5] used SBI to construct initial guesses $X_0(x, 0)$, which they refined iteratively using DAL. SBI was also examined by [18] to dismiss a time-reversibility hypothesis on Subgrid-Scale (SGS) Large-Eddy Simulations (LES).

[19] applied iterative QRM to ill-posed linear inverse problems and derives rigorous statements of convergence. More recently, [20] used iterative QRM to compute the unknown source in a nonlinear hyperbolic equation. Using Carleman estimates, they demonstrate QRM's convergence to the correct source at an exponential rate. While an analysis involving Carleman estimates is outside the scope of this paper, we refer to [21].

We compare DAL, SBI, and QRM by solving two retrospective inverse problems, respectively constrained by Korteweg–de Vries–Burgers (KdVB) and incompressible Navier–Stokes. The novelty of our work lies in the *iterative* application of SBI and QRM. At the start of iteration n, we evolve some initial trial state $X_n(x, 0)$ to its final state $X_n(x, t_f)$. Then, we initialize SBI and QRM using the final deviation $X_n'(x, t_f)$ in order to approximate the initial deviation $X_n'(x, 0)$. Finally, assuming we know $X_n'(x, 0)$, we refine the trial state as $X_{n+1}(x, 0) := X_n(x, 0) - X_n'(x, 0)$ to conclude the iteration. Equivalently, iterative SBI and QRM minimize the inaccessible objective functional

$$\mathcal{J}_0^X \equiv \frac{1}{2}\left\langle \left|X'(x, 0)\right|^2 \right\rangle = \frac{1}{2}\left\langle \left|X(x, 0) - \overline{X}(x, 0)\right|^2 \right\rangle \quad (3)$$

by repeatedly approximating its gradient $X'(x, 0)$. $\mathcal{J}_0^X$ measures the trial state's error at the initial time, whereas $\mathcal{J}_f^X$ does so at the final time.

The remainder of this paper is organized as follows: In Section II we provide a conceptual description of our iterative methods (DAL, SBI, and QRM). In Section III we illustrate these methods by applying them to a retrospective KdVB inverse problem. In Section IV we generalize these methods for multiple spatial dimensions, and compare their implementations using a 2D incompressible Navier–Stokes inverse problem. In Section V, we demonstrate that the adjoint equation which yields the gradient of $\mathcal{J}_0^X$ is ill-posed, nonlinear, and equivalent to the perturbed constraint eqn 1. In Section VI we summarize our results and discuss future applications for iterative SBI and QRM.

II. OVERVIEW OF METHODS

The Korteweg–de Vries–Burgers (KdVB) equation

$$\mathcal{F}[\overline{u}(x,t)] \equiv \partial_t \overline{u} + \overline{u}\partial_x \overline{u} - a\partial_x^2 \overline{u} + b\partial_x^3 \overline{u} = 0 \qquad (4)$$

provides a concise context for demonstrating each method (DAL, SBI, QRM). Let $\overline{u}(x,t)$ denote to the target solution where $0 \leq t \leq t_f$ in a 1D periodic domain $0 \leq x < L_x$. The retrospective inverse problem is solved by recovering the target initial state $\overline{u}(x,0)$ where the target final state $\overline{u}(x,t_f)$ is given. The general strategy has us construct trial solutions $u(x,t)$ satisfying eqn 4, which we compute from trial initial states $u(x,0)$. We measure a trial state's proximity from the target using the cost functionals $\mathcal{J}_0^u$ and $\mathcal{J}_f^u$, which are defined for an arbitrary variable X in eqns 2 and 3. Both $\mathcal{J}_0^u$ and $\mathcal{J}_f^u$ vary with respect to the trial initial state $u(x,0)$. However, given that our knowledge of the target solution is limited to the final time t_f, it follows that we can only evaluate $\mathcal{J}_0^u$ if the target solution is known apriori.

A. Direct Adjoint-Looping (DAL)

In general, we can cast an inverse problem into the form of an optimization problem by defining and minimizing an appropriate cost functional [22]. DAL provides an efficient avenue for computing a cost functional's gradient. For KdVB, we minimize $\mathcal{J}_f^u$ by computing its gradient with respect to $u(x,0)$. Once this gradient is known, we refine the trial initial state $u(x,0)$ by following the direction of steepest descent.

Using DAL, the gradient is computed as follows. We define the Lagrangian $\mathcal{L}$

$$\mathcal{L} \equiv \int_0^{t_f} \Big\langle \mu(x,t) \cdot \mathcal{F}[u(x,t)] \Big\rangle dt + \mathcal{J}_f^u \qquad (5)$$

where μ is the adjoint variable corresponding to u. $\mathcal{L}$'s variations with respect to u and μ must disappear at the optimum $\overline{u}$, yielding

$$0 < t < t_f: \quad 0 = \partial_t u + u\partial_x u - a\partial_x^2 u + b\partial_x^3 u \qquad (6)$$

$$t = t_f: \quad \mu = -u' \qquad (7)$$

$$0 < t < t_f: \quad 0 = \partial_t \mu + u\partial_x \mu + a\partial_x^2 \mu + b\partial_x^3 \mu. \qquad (8)$$

The gradient of $\mathcal{J}_f^u$ resides in the manifold of solutions satisfying eqns 6, 7, and 8. Figure 1 illustrates how these equations are solved sequentially. Given a trial initial condition, we start by solving the forward problem eqn 6 from $t : 0 \rightarrow t_f$ (shown in blue). Next, we initialize the adjoint variable using the compatibility condition eqn 7. Finally, we solve the adjoint eqn 8 backward in time $t : t_f \rightarrow 0$ (shown in yellow), yielding the desired gradient $-\mu(x,0)$. The adjoint involves the trial solution u, so we must store data from the forward solve in memory. In practice this leads to redundant computations (due to memory constraints), offering another area for tangible improvement.
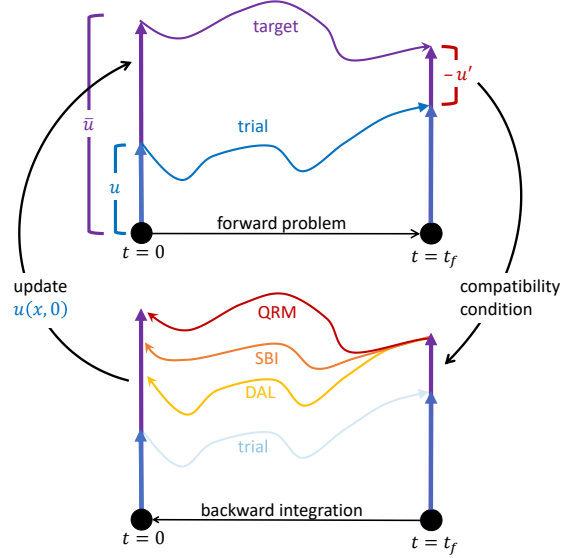

FIG. 1. Conceptual illustration of iterative methods. Purple: the target solution $\overline{u}(x,t)$ is known only at the final time t_f. Blue: we evolve a trial state $u(x,0)$ to $u(x,t_f)$ where we then initialize one of three backward integration systems using a single compatibility condition. Each backward integration gives a solution at $t = 0$ which we use to update our trial initial condition. We seek the trial solution's deviation $u' = u - \overline{u}$ at the initial time $t = 0$. Yellow: Direct Adjoint Looping (DAL) is a linear backward integration. For advective systems such as KdVB and Navier–Stokes, DAL advects the final deviation $u'(x,t_f)$ by the trial solution u. Orange: Simple Backward Integration (SBI) is a hybrid method, which supplements the linear DAL system with two additional advective terms. Red: the Quasi-Reversible Method (QRM) aims to compute the deviation by introducing a small hyperdiffusion term with coefficient ε. The QRM backward integration is ill-posed when $\varepsilon = 0$ and numerically unstable as $\varepsilon \rightarrow 0$.

Once $\mathcal{J}_f^u$'s gradient is known, we use gradient descent with the popular Barzilai–Borwein method [23] to select step sizes. In addition to gradient descent, we also minimize $\mathcal{J}_f^u$ using the second-order sparse L-BFGS algorithm [24] which utilizes the gradient's history over several iterations. We select these two optimization routines for comparison after observing that they converge faster than alternatives. [25]

B. SBI and QRM Setup

The KdVB retrospective inverse problem can be solved by finding the initial deviation $u'(x, 0)$ of some trial state $u(x, 0)$. $u' = u - \overline{u}$ obeys a perturbation equation

$$0 = \partial_t u' - a\partial_x^2 u' + b\partial_x^3 u' + u\partial_x u' + u'\partial_x u - u'\partial_x u' \quad (9)$$

which is ill-posed over $t : t_f \to 0$. We cannot compute $u'(x, t < t_f)$ numerically due to the intrinsic loss of information in diffusive systems. DAL circumvents this obstacle by computing a gradient, whereas SBI and QRM approximate the deviation u'. Let $\tilde{\mu}$ denote our SBI and QRM approximations for $-u'$.

C. Simple Backward Integration (SBI)

SBI is one method for approximating the ill-posed backward integration, in which we reverse the sign of the problematic diffusion term, i.e. $a\partial_x^2 \tilde{\mu} \to -a\partial_x^2 \tilde{\mu}$ such that

$$0 = \partial_t \tilde{\mu} + a\partial_x^2 \tilde{\mu} + b\partial_x^3 \tilde{\mu} + u\partial_x \tilde{\mu} + \tilde{\mu}\partial_x u + \tilde{\mu}\partial_x \tilde{\mu}. \quad (10)$$

This backward integration is well-posed. Eqn 10 contains every term in eqn 8 (DAL) with two additional advective terms: $\tilde{\mu}\partial_x u$ and $\tilde{\mu}\partial_x \tilde{\mu}$.

For the KdVB inverse problem, we first initialize eqn 10 with $\tilde{\mu}(x, t_f) = -u'(x, t_f)$, just as with with DAL. We then solve eqn 10 on $t : t_f \to 0$. Finally, SBI yields an approximation $\tilde{\mu}(x, 0) \approx -u'(x, 0)$. The SBI backward integration loop is illustrated in orange in Figure 1, where the additional SBI terms cause $\tilde{\mu}$ to deviate from the linear DAL calculation of μ. After completing an SBI loop, we update the trial state $u_{n+1}(x, 0) = u_n(x, 0) + \tilde{\mu}_n(x, 0)$. This is analogous to performing gradient descent with a fixed step size of unity.[26]

In previous investigations, SBI was applied once to the target solution. Our iterative implementation differs, because we apply SBI to the deviation equation. Despite the inaccuracies introduced by reversing the sign of the diffusive term, we will demonstrate that iterative SBI is an effective algorithm. SBI has already been implemented to develop initial guesses for retrospective inverse problems. We extend this work by comparing SBI with DAL rather than using them separately for different purposes.

D. Quasi-Reversible Method (QRM)

QRM deals with the ill-posed diffusion term by introducing a small, well-posed hyperdiffusion term: $a\partial_x^2 \tilde{\mu} \to a(\partial_x^2 \tilde{\mu} + \varepsilon\partial_x^4 \tilde{\mu})$. This method replaces eqn 9 by

$$0 = \partial_t \tilde{\mu} - a(\partial_x^2 \tilde{\mu} + \varepsilon\partial_x^4 \tilde{\mu}) + b\partial_x^3 \tilde{\mu} + u\partial_x \tilde{\mu} + \tilde{\mu}\partial_x u + \tilde{\mu}\partial_x \tilde{\mu}. \quad (11)$$

$0 < \varepsilon \ll 1$ is a free parameter chosen according to the numerical resolution as well as the solutions' spatial scales. Notice how the ill-posed term remains, but the equation becomes well-posed because small-scale modes do not grow arbitrarily fast.

The QRM loop is carried out using the same procedure as SBI. We initialize eqn 11 with $\tilde{\mu}(x, t_f) = -u'(x, t_f)$ and update the trial state $u_{n+1}(x, 0) = u_n(x, 0) + \tilde{\mu}_n(x, 0)$. In Figure 1, we illustrate how QRM (red) recovers the unknown deviation precisely as $\varepsilon \to 0$. However this limit is numerically unstable.

III. KDVB NUMERICAL EXPERIMENTS

We implement each iterative method (DAL, SBI, QRM) to solve an inverse problem constrained by KdVB (eqn 4). The target solution $\overline{u}(x, t)$ (shown left in Figure 2) consists of a dissipating nonlinear wave where $a = 0.02$ and $b = 0.04$. The spatial domain is 1D periodic $x \in [0, 2\pi]$ and we solve eqn 4 over $t : 0 \to 3\pi$. The target initial condition

$$\overline{u}(x, 0) = 3\cosh^{-2}\left(\frac{x - \pi}{2\sqrt{b}}\right), \quad (12)$$

would propagate rightwards as a stable soliton with a constant speed of one if diffusion were not present (i.e. if $a = 0$ in eqn 4). This retrospective problem is nontrivial, as the final state cannot be evolved backwards in time via conventional timestepping algorithms.

The target and trial solutions along with each backward integration are carried out using the Dedalus open-source pseudospectral python framework [27]. We represent u, μ, and $\tilde{\mu}$ as vectors of 128 real Fourier modes. While computing $u(x, t)$, we store the solution vector at each timestep. Linear (diffusive and dispersive) terms are treated implicitly whereas the nonlinear advective term is evaluated explicitly by multiplying u's grid data on a 3/2 dealiased grid. The same holds when we perform backward integrations, where terms such as $\mu\partial_x u$ must be treated explicitly to avoid dense matrix operations at every timestep. Compared to DAL, the SBI and QRM backward integration equations contain more of these terms. Consequently, each SBI and QRM iteration requires approximately 30% more computation time. For every solve we use a 2nd-order Runge–Kutta scheme with a fixed timestep $\Delta t = 0.01$. We verify the adjoint solver's accuracy by comparing its output to a gradient obtained via finite-difference. Our code for this paper is available at https://github.com/liamoconnor9/adjop

We initialize each iterative method using an initial guess $u^0(x, 0) = 0$. (See Appendix A for the same comparison beginning with an SBI initial guess.) The first SBI iteration is shown in the center panel of Figure 2. Diffusion of the nonlinear wave causes its amplitude and
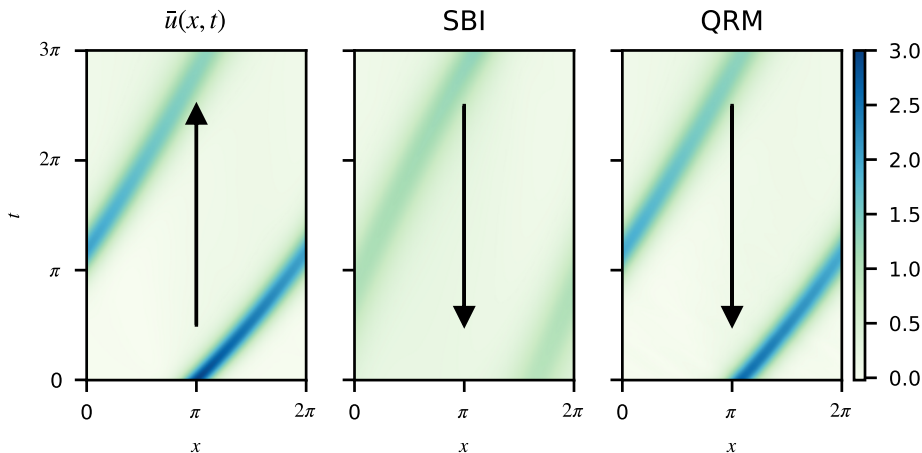
FIG. 2. Target KdVB solution $\bar{u}(x,t)$ (left) plotted as a function of space x and time t. We aim to recover the initial condition of a nonlinear wave which propagates right at an initial speed of one. The problem's spatial domain is 1D periodic. We include diffusion ($a = 0.04$) such that its speed decreases as the wave propagates. Beginning at $t = 3\pi$, SBI (center) evolves $\bar{u}(x,t_f)$ backward to $t = 0$ by reversing the sign of the diffusive term. The wave diffuses during the forward and backward integrations, such that its amplitude and speed of propagation decrease in both directions. We also perform the QRM backward integration (right) on $t: 3\pi \to 0$. Unlike SBI, QRM repopulates small-scale modes which are lost during forward integration. As the wave's peak regains its former amplitude, its speed of propagation increases mimicking $\bar{u}(x,t)$.

speed of propagation to decrease such that the wave's peak does not return to its original position. The first QRM iteration (shown right in Figure 2) reconstructs the wave by inverting the advection alongside diffusion. In this case, the wave's peak returns to its approximate initial position. For QRM we use $\varepsilon = 0.01$ whereas SBI involves no free parameter.

The top row of Figure 3 illustrates the evolution of DAL trial initial states $u_n(x,0)$. We minimize $\mathcal{J}_f^u$ using gradient descent (top left) and L-BFGS (top right). For advective systems such as KdVB, the amplitude of each feature is proportional to its speed of propagation. Consequently, gradient descent gives a sequence of initial states which increase in amplitude while shifting leftwards. $u_{10}(x,0)$ contains a shallow peak centered near $x = \pi/2$. $u_{50}(x,0)$ is centered near $x = 0$ with approximately twice the amplitude of $u_{10}(x,0)$. Similarly, $u_{200}(x,0)$ is centered near $x = 3\pi/2$ with approximately triple the amplitude of $u_{10}(x,0)$. Their respective final states ($u_{10}(x,3\pi)$, $u_{50}(x,3\pi)$, and $u_{200}(x,3\pi)$) all contain peaks which approximately align with the target's ($\bar{u}(x,3\pi)$).

Rather than following the direction of steepest descent, L-BFGS anticipates changes in the gradient by constructing a sparse Hessian representation. Although this quasi-Newton method recovers the target initial state better than gradient descent at $n = 200$, L-BFGS still does not yield a quality approximation of $\bar{u}(x,0)$. This case introduces additional wave features in $u_{10}(x,0)$, $u_{50}(x,0)$, and $u_{200}(x,0)$ which do not appear in the target.

To locate the target state, we introduce additional advective terms in the backward integration. The bottom row of Figure 3 illustrates the evolution of trial initial states $u_n(x,0)$ refined via SBI (bottom left) and QRM

(bottom right) over 200 iterations. SBI does not effectively approximate the initial deviation $u_n'(x,0)$ (as shown for $n = 0$ in Figure 2), nor does it follow the direction of steepest descent of $\mathcal{J}_f^u$. This algorithm smoothly guides the initial states' peaks directly toward the target. In this case, $u_{50}(x,0)$ is indistinguishable from $\bar{u}(x,0)$. QRM loosely captures the target's structure in a single iteration by approximating the initial deviation $u_n'(x,0)$. In this case, $u_{10}(x,0)$ is nearly indistinguishable from $\bar{u}(x,0)$.

After one iteration, the SBI and QRM initial states' peaks are centered rightwards of the target feature (see $u_1(x,0)$ in Figure 3, bottom left and right panels). These nonlinear backward integrations couple the deviation's amplitude with the resulting direction of our update for $u_n(x,0)$. SBI refines each trial initial state by advecting the final deviation backward in time as it continues to diffuse. This decreases the speed of backward advection such that each update does not return to the position of the initial deviation. This is illustrated by $u_{10}(x,0)$ and $u_{50}(x,0)$ in the bottom left panel of Figure 3. Their peaks shift leftward, approaching the target with each iteration. When the trial final state intercepts the target, the deviation shrinks and SBI's backward diffusion has less influence on the direction of each subsequent update. QRM refines each trial state by advecting the final deviation while simultaneously inverting diffusion. The hyperdiffusion term slows backward advection to a lesser extent than SBI, yielding smaller rightward misalignments. These updates approximate the initial deviation as it shrinks with each iteration, such that iterative QRM behaves like a one-shot method (e.g. [28]).
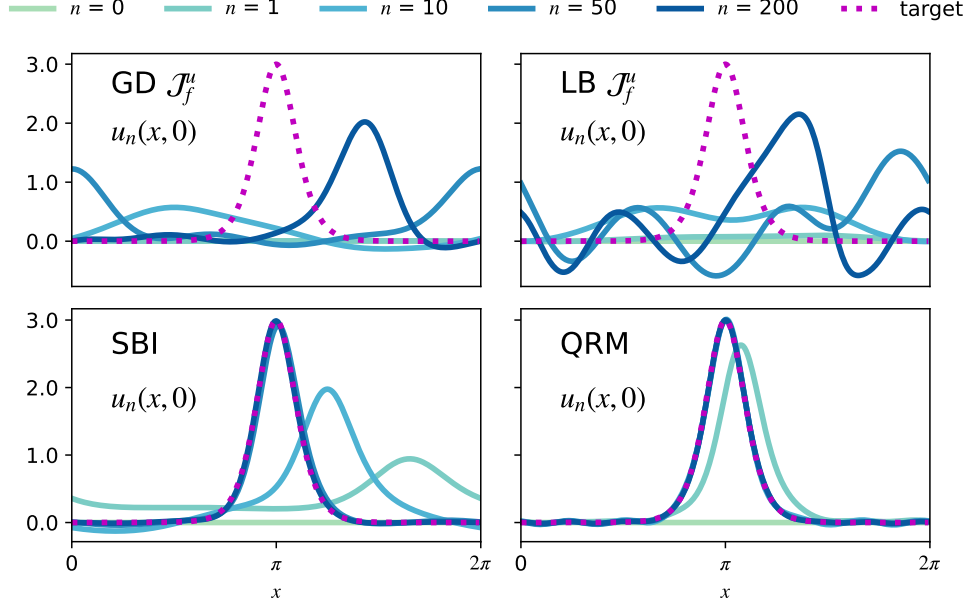
FIG. 3. Evolution of the KdVB trial initial conditions $u_n(x,0)$ using an initial guess $u_0(x,0) = 0$ (shown in green). We compare two DAL runs (top row) with SBI (bottom left) and QRM (bottom right). (Top left) gradient descent approaches the target initial state gradually. (Top right) DAL minimization of $\mathcal{J}_f^u$ using L-BFGS approaches the target state more rapidly, but this method introduces undesired medium-scale features. SBI captures the target state faster and more directly than DAL. QRM resembles the target state immediately ($n = 1$). However, this method introduces low-amplitude small-scale errors which do not subside at $n = 200$.

For QRM and SBI, we plot the deviation $u'(x,0) = u_{200}(x,0) - \overline{u}(x,0)$ at iteration $n = 200$ in Figure 4. Both curves have relatively large features coinciding with the target initial condition's peak at $x = \pi$. The SBI deviation has a larger maximum magnitude while the QRM deviation oscillates at a particular wavenumber. When we increase ε, these oscillations subside, but it requires more iterations to obtain a comparable result. We select $\varepsilon = 0.01$ to illustrate the presence of these oscillations, while still achieving the desired outcome in a small number of iterations.

Errors from each optimization study are plotted in Figure 5. DAL does not effectively minimize $\mathcal{J}_0^u$ or $\mathcal{J}_f^u$. In similar investigations, L-BFGS has received significant attention thanks to its DAL-compatible features [5]. It converges faster than gradient descent which is also reflected in our experiment. However, the SBI and QRM iterative methods both outperform DAL by several orders of magnitude. QRM minimizes both $\mathcal{J}_0^u$ and $\mathcal{J}_f^u$ more rapidly than SBI, as shown in Figure 5. At $n \approx 100$, SBI stalls and QRM approaches floating-point precision in terms of $\mathcal{J}_f^u$. We attribute this gap to an energy discrepancy between the SBI and QRM updates. For QRM applied to a highly diffusive system, the energy $\langle |\tilde{\mu}(x,t)|^2 \rangle$ increases during backward integration (provided ε is sufficiently small) such that $\langle |\tilde{\mu}(x,0)|^2 \rangle \sim \langle |u'(x,0)|^2 \rangle$. This is illustrated in the first QRM iteration (Figure 2, right). In contrast, SBI diffuses during backward integration (Figure 2, center) such that $\langle |\tilde{\mu}(x,0)|^2 \rangle < \langle |u'(x,0)|^2 \rangle$.
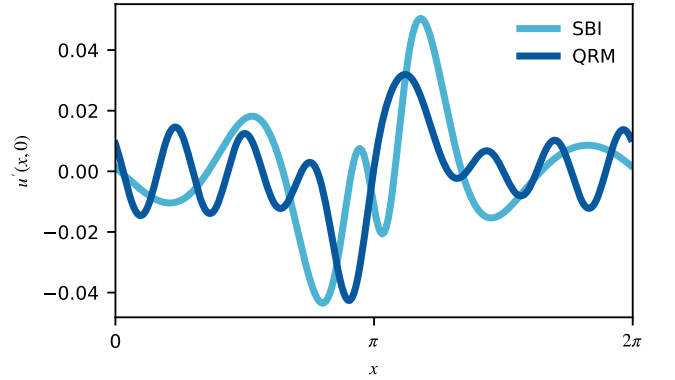

FIG. 4. [NEW FIGURE] Deviation of SBI and QRM trial initial conditions, plotted as a function of x at iteration $n = 200$. Both curves have prominent peaks which overlap near $x = \pi$. The SBI deviation has a slightly larger magnitude while the QRM deviation oscillates at a particular wavenumber.

We can eliminate this energy discrepancy and often accelerate SBI by increasing the step sizes of the updates (e.g. $u_{n+1}(x,0) = u_n(x,0) + 1.1\mu_n(x,0)$).
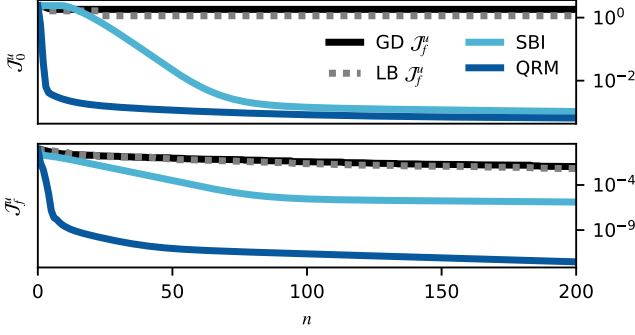
FIG. 5. Initial errors ($\mathcal{J}_0^u$, top) and final errors ($\mathcal{J}_f^u$, bottom) for the 1D KdVB inverse problem plotted as a function of iteration n. We compare DAL, SBI, and QRM. DAL performs slightly better when paired with the second-order sparse minimization routine L-BFGS (LB $\mathcal{J}_f^u$), whereas gradient descent (GD $\mathcal{J}_f^u$) performs the worst. Our iterative applications of SBI and QRM minimize $\mathcal{J}_0^u$ and $\mathcal{J}_f^u$ more effectively than DAL. QRM minimizes both $\mathcal{J}_0^u$ and $\mathcal{J}_f^u$ more rapidly than SBI.

IV. 2D NAVIER–STOKES NUMERICAL EXPERIMENTS

Consider 2D flow with spatial coordinate $\boldsymbol{x} = x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}}$. The velocity $\overline{\boldsymbol{u}}(\boldsymbol{x}, t)$ obeys the incompressible Navier–Stokes equation

$$\partial_t \overline{\boldsymbol{u}} + \overline{\boldsymbol{u}} \cdot \nabla \overline{\boldsymbol{u}} + \nabla \overline{p} = \nu \nabla^2 \overline{\boldsymbol{u}} \quad \text{and} \quad \nabla \cdot \overline{\boldsymbol{u}} = 0 \quad (13)$$

where $\overline{p}$ is the pressure and $\mathrm{Re} \equiv \nu^{-1} = 50{,}000$. The domain is doubly-periodic with $x \in [0, 1]$ and $y \in [-1, 1]$, such that its boundary conditions are given by $\overline{\boldsymbol{u}}|_{x=0} = \overline{\boldsymbol{u}}|_{x=1}$ and $\overline{\boldsymbol{u}}|_{y=-1} = \overline{\boldsymbol{u}}|_{y=1}$.

For our inverse problem, we use divergence-cleaning to construct the target initial condition $\overline{\boldsymbol{u}}(\boldsymbol{x}, 0)$. First, consider the following velocity field which has a nonzero divergence

$$\overline{\boldsymbol{v}}(\boldsymbol{x}) \cdot \hat{\boldsymbol{x}} = \frac{1}{2}\left[\tanh\left(10\left(y - \frac{1}{2}\right)\right) - \tanh\left(10\left(y + \frac{1}{2}\right)\right) \right];$$

$$\overline{\boldsymbol{v}}(\boldsymbol{x}) \cdot \hat{\boldsymbol{y}} = \frac{1}{10} \sin\left(2\pi x\right)\left[\exp\left(-100\left(y - \frac{1}{2}\right)^2\right) \right. \quad (14)$$

$$\left. - \exp\left(-100\left(y + \frac{1}{2}\right)^2\right) \right].$$

Next, we perform divergence cleaning on $\overline{\boldsymbol{v}}$ by solving the boundary value problem $\nabla^2 p_v + \nabla \cdot \overline{\boldsymbol{v}} = 0$. The divergence-free target initial condition is then given by $\overline{\boldsymbol{u}}(\boldsymbol{x}, 0) = \overline{\boldsymbol{v}}(\boldsymbol{x}) + \nabla p_v(\boldsymbol{x})$.

For our inverse problem, we run a target simulation by evolving $\overline{\boldsymbol{u}}(\boldsymbol{x}, 0)$ according to eqn 13 until $t_f = 20$. The target initial condition has two thin layers of positive and negative vorticity concentrated near $y = 1/2$ and $y = -1/2$. Advection causes these vortex sheets to wind

counter-clockwise and clockwise while viscosity smooths its small-scales.

Given the final velocity field $\overline{\boldsymbol{u}}(\boldsymbol{x}, t_f)$, we develop trial solutions $\boldsymbol{u}(\boldsymbol{x}, t)$ (also obeying eqn 13) which are meant to approach the target solution $\overline{\boldsymbol{u}}(\boldsymbol{x}, t)$. All PDE solves (except the GD $\mathcal{J}_f^{\boldsymbol{u}}$ double-resolution study) are performed on a 128 by 256 grid of real Fourier modes using the Dedalus pseudospectral framework [27]. We timestep with $\Delta t = 0.002$ using a second-order Runge–Kutta scheme. Nonlinear operations are evaluated on a 3/2 dealiased grid. We plot the upper half the domain (unit square) because the solutions have an approximate symmetry about $y = 0$. The target solution's vorticity evolution is illustrated in the top row of Figure 6.

A. DAL

Let $\boldsymbol{\mu}$ and Π be the adjoint variables respectively corresponding to $\boldsymbol{u}$ and p. The Navier–Stokes adjoint equations are then given by

$$0 = \nabla \cdot \boldsymbol{\mu} \quad (15)$$

$$\boldsymbol{0} = \partial_t \boldsymbol{\mu} + \boldsymbol{u} \cdot \nabla \boldsymbol{\mu} \;\boxed{-\boldsymbol{\mu} \cdot (\nabla \boldsymbol{u})^T}\; + \nabla \Pi + \nu \nabla^2 \boldsymbol{\mu}.$$

[29] provides a detailed derivation of eqn 15. Using DAL, we minimize two objective functionals with associated compatibility conditions

$$\mathcal{J}_f^{\boldsymbol{u}} \equiv \frac{1}{2} \left\langle \|\boldsymbol{u}'(\boldsymbol{x}, t_f)\|^2 \right\rangle \rightarrow \boldsymbol{\mu}(\boldsymbol{x}, t_f) = -\boldsymbol{u}'(\boldsymbol{x}, t_f); \quad (16)$$

$$\mathcal{J}_f^{\omega} \equiv \frac{1}{2} \left\langle (\omega'(\boldsymbol{x}, t_f))^2 \right\rangle \rightarrow \boldsymbol{\mu}(\boldsymbol{x}, t_f) = \nabla^{\perp} \omega'(\boldsymbol{x}, t_f). \quad (17)$$

The skew-gradient $\nabla^{\perp} \equiv (-\partial_y, \ \partial_x)$ and the vorticity $\omega \equiv \nabla^{\perp} \cdot \boldsymbol{u}$. As with KdVB, a prime denotes the deviation of a trial variable from its target, e.g. $\omega' \equiv \omega - \overline{\omega}$ and so on. $\mathcal{J}_f^{\boldsymbol{u}}$ quantifies the trial states' proximity to the target at the final time t_f in terms of velocity whereas $\mathcal{J}_f^{\omega}$ does so using vorticity. We implement the gradient descent (GD) and L-BFGS optimization routines to minimize these two cost functionals. We verify the accuracy of the adjoint solver by using the finite-difference approximation to reproduce the gradient at low-resolution.

B. SBI and QRM

Our iterative procedures for SBI and QRM are analogous to those described in the previous section. The ill-posed backward integration of $\boldsymbol{u}'$ obeys the nonlinear perturbation equation

$$0 = \nabla \cdot \boldsymbol{u}' \quad (18)$$

$$\boldsymbol{0} = \partial_t \boldsymbol{u}' + \boldsymbol{u} \cdot \nabla \boldsymbol{u}' + \boldsymbol{u}' \cdot \nabla \boldsymbol{u} - \boldsymbol{u}' \cdot \nabla \boldsymbol{u}' + \nabla p' - \nu \nabla^2 \boldsymbol{u}'.$$

Next we substitute the approximations $-\tilde{\boldsymbol{\mu}}$ for $\boldsymbol{u}'$ and $-\tilde{\Pi}$ for p'. For SBI, we reverse the sign of the viscous term $\nu\nabla^2\tilde{\boldsymbol{\mu}}$. Its corresponding approximation for eqn 18 is given by

$$0 = \nabla \cdot \tilde{\boldsymbol{\mu}} \tag{19}$$
$$\boldsymbol{0} = \partial_t\tilde{\boldsymbol{\mu}} + \boldsymbol{u}\cdot\nabla\tilde{\boldsymbol{\mu}} + \tilde{\boldsymbol{\mu}}\cdot\nabla\boldsymbol{u} + \tilde{\boldsymbol{\mu}}\cdot\nabla\tilde{\boldsymbol{\mu}} + \nabla\tilde{\Pi} + \nu\nabla^2\tilde{\boldsymbol{\mu}}$$

For QRM, we introduce a small hyperdiffusion term, approximating eqn 18 as

$$0 = \nabla \cdot \tilde{\boldsymbol{\mu}} \tag{20}$$
$$\boldsymbol{0} = \partial_t\tilde{\boldsymbol{\mu}} + \boldsymbol{u}\cdot\nabla\tilde{\boldsymbol{\mu}} + \tilde{\boldsymbol{\mu}}\cdot\nabla\boldsymbol{u} + \tilde{\boldsymbol{\mu}}\cdot\nabla\tilde{\boldsymbol{\mu}} + \nabla\tilde{\Pi}$$
$$- \nu(\nabla^2 + \varepsilon\nabla^4)\tilde{\boldsymbol{\mu}}.$$

For this problem, we set $\varepsilon = 0.001$, because the oscillatory errors described in Section III trigger a numerical instability when $\varepsilon \lesssim 0.001$.
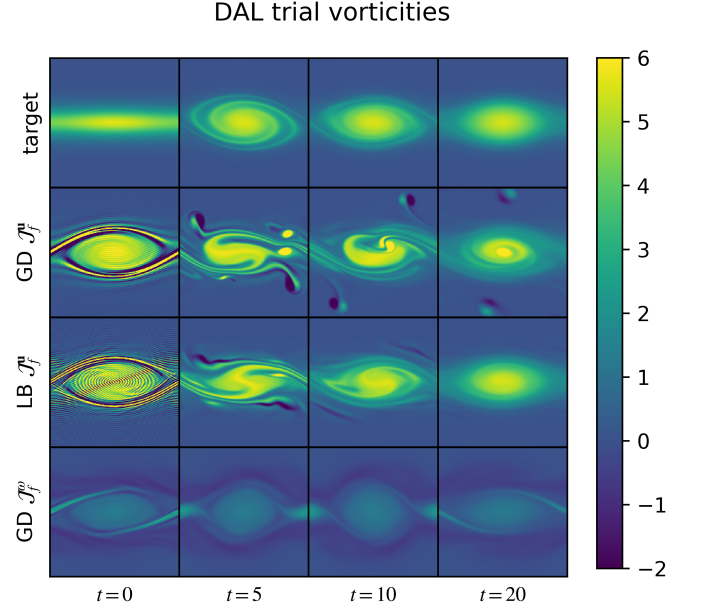

DAL trial vorticities

FIG. 6. Target (top row) and trial vorticity snapshots in the upper half of the domain at simulation times $t = 0, 5, 10, 20$. The second, third, and bottom row trial solutions are obtained by applying DAL over 200 iterations. GD $\mathcal{J}_f^{\boldsymbol{u}}$ (second row) uses gradient descent to minimize $\mathcal{J}_f^{\boldsymbol{u}}$. For this case, we use double resolution in space and time (256 by 512 modes, $\Delta t = 0.001$) to ensure that each simulation is resolved. Despite this extra resolution, we observe symmetry breaking when $t \geq 5$ due to numerical errors. LB $\mathcal{J}_f^{\boldsymbol{u}}$ (third row) uses L-BFGS to minimize $\mathcal{J}_f^{\boldsymbol{u}}$ with the default resolution. Although the initial condition is under-resolved in this case, we still recover a final state which resembles the target. GD $\mathcal{J}_f^{\omega}$ (bottom row) uses gradient descent to minimize $\mathcal{J}_f^{\omega}$. The DAL trial initial conditions (bottom three rows) have small-scale banded features with high concentrations of positive vorticity. Their final states have varying degrees of resemblance to the target at $t = 20$.

C. Results

We initialize each iterative method using an initial guess $\boldsymbol{u}^0(\boldsymbol{x}, 0) = \boldsymbol{0}$. For DAL, we minimize the velocity error using gradient descent (GD $\mathcal{J}_f^{\boldsymbol{u}}$) and L-BFGS (LB $\mathcal{J}_f^{\boldsymbol{u}}$). We run GD $\mathcal{J}_f^{\boldsymbol{u}}$ at double resolution to ensure that the trial state's evolution is well-resolved. We also minimize the vorticity error via gradient descent (GD $\mathcal{J}_f^{\omega}$). L-BFGS minimization of $\mathcal{J}_f^{\omega}$ frequently gives unstable trial initial conditions which cannot be evolved without additional resolution. Figure 6 consists of vorticity snapshots belonging to the target solution and trial solutions at $t = 0, 5, 10, 20$. The trial solutions in these cases are obtained using DAL until iteration $n = 200$. GD $\mathcal{J}_f^{\boldsymbol{u}}$ (second row) contains highly concentrated bands of positive vorticity encompassing a larger, lower-amplitude vortex. Although this case is resolved, we observe symmetry breaking when $t \gg 0$. This trial solution adequately captures the target at $t = 20$, but not as well as LB $\mathcal{J}_f^{\boldsymbol{u}}$. LB $\mathcal{J}_f^{\boldsymbol{u}}$ (third row) is clearly under-resolved at $t = 0$, with the same highly-concentrated features as GD $\mathcal{J}_f^{\boldsymbol{u}}$ along with small-scale ringing throughout the domain. These ringing features subside after a brief interval ($t \geq 5$), after which, the trial solution approaches the target as $t \to 20$. GD $\mathcal{J}_f^{\omega}$ (bottom row) is well-resolved, but its vortex's amplitude is appreciably lower than that of the target. In this case, we still observe band-like regions of concentrated vorticity encompassing a larger low-amplitude vortex. The DAL trial initial conditions (illustrated in the left column, lower three rows) do not share similar structures with the target (top left). Despite this, advection sweeps the concentrated vorticity bands into a single coherent vortex after 20 advective times. This process illustrates ill-conditioning of the Navier–Stokes inverse problem.

Figure 7 illustrates the analogous trial vorticity snapshots using SBI and QRM. In the top row, we reproduce

the target. The second row (SBI1) denotes the trial state after a single SBI iteration. Its initial state resembles the target's better than any of the DAL runs in Figure 6. Its final state is nearly indistinguishable from the target. The third row (SBI200) denotes the SBI trial solution after 200 iterations. The bottom row (QRM) denotes the trial solution after 200 QRM iterations. This case shares the most resemblance with the target throughout the time domain. At $t = 0$, QRM has curved arm-like features which mimic the target's initially confined shear layer. At $t \geq 5$, this trial solution resembles the target. In general, SBI and QRM capture the target state better than DAL without introducing the undesired features shown in Figure 6.

Figure 8 shows the time-dependent velocity error $\mathcal{J}^{\boldsymbol{u}}(t) \equiv \frac{1}{2}\langle|\boldsymbol{u}(\boldsymbol{x}, t) - \overline{\boldsymbol{u}}(\boldsymbol{x}, t)|^2\rangle$ for the trial solutions GD $\mathcal{J}_f^{\boldsymbol{u}}$, LB $\mathcal{J}_f^{\boldsymbol{u}}$, SBI, and QRM at iteration $n = 200$. Due to the inverse-problem's ill-conditioning, all trial states begin with substantially larger errors than their respec-
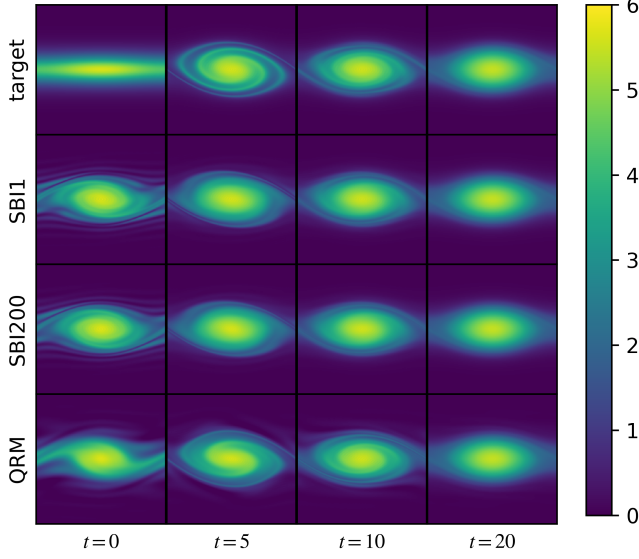
SBI/QRM trial vorticities

FIG. 7. Same as Figure 6 using the SBI and QRM iterative methods. Despite the ill-conditioning which hinders DAL in Figure 6, SBI and QRM both recover the target initial condition (top left) to a greater extent. SBI1 and SBI200 (middle two rows) denote SBI trial states after 1 and 200 iterations respectively. The QRM initial state (bottom left) loosely resembles the target initial condition at $n = 200$.

tive final states. These errors decrease nonmonotonically on $t : 0 \to 20$. The DAL trial solutions' (GD $\mathcal{J}_f^{\boldsymbol{u}}$ and LB $\mathcal{J}_f^{\boldsymbol{u}}$) errors increase on $0 < t < 2$ as their small-scale vortex bands (illustrated in the left column of Figure 6) rapidly disappear. LB $\mathcal{J}_f^{\boldsymbol{u}}$ performs significantly better than GD $\mathcal{J}_f^{\boldsymbol{u}}$ throughout most of the time domain, especially near the final state. In contrast, the velocity errors of SBI and QRM decrease immediately after initialization. Throughout the time domain, SBI and QRM perform better than either DAL case.

Figure 9 plots the initial ($\mathcal{J}_0^{\boldsymbol{u}}$, top) and final ($\mathcal{J}_f^{\boldsymbol{u}}$, bottom) velocity errors vs. iteration n for each iterative method until $n = 20$. Every method's performance is comparable in terms of $\mathcal{J}_0^{\boldsymbol{u}}$ except GD $\mathcal{J}_f^{\omega}$ which does not appreciably decrease this error. In terms of $\mathcal{J}_f^{\boldsymbol{u}}$, SBI (light blue solid) and QRM (dark blue dotted) both outperform every DAL case by several orders of magnitude.

Figure 10 extends Figure 9 to iteration $n = 200$. Both DAL gradient descent algorithms (GD $\mathcal{J}_f^{\boldsymbol{u}}$ and GD $\mathcal{J}_f^{\omega}$) stall after $n = 20$, whereas LB $\mathcal{J}_f^{\boldsymbol{u}}$ continues to minimize the final error $\mathcal{J}_f^{\boldsymbol{u}}$. The relative success of LB $\mathcal{J}_f^{\boldsymbol{u}}$ indicates that the gradient of $\mathcal{J}_f^{\boldsymbol{u}}$ has large curvature. SBI and QRM continue to outperform all DAL runs, with QRM decreasing $\mathcal{J}_0^{\boldsymbol{u}}$ slightly more. With QRM, the errors decrease nonmonotonically whereas SBI exhibits a smooth decline.

Whereas previous work used SBI and QRM to perform a single loop, we demonstrate in Figures 9 and 10 that
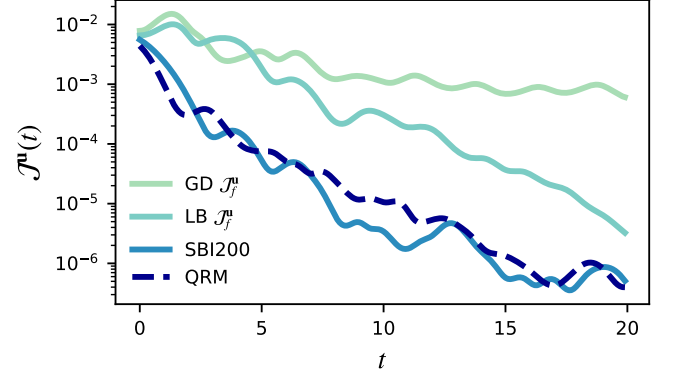

FIG. 8. Velocity error $\mathcal{J}^{\boldsymbol{u}}(t) \equiv \frac{1}{2}\langle|\boldsymbol{u}(\boldsymbol{x}, t) - \overline{\boldsymbol{u}}(\boldsymbol{x}, t)|^2\rangle$ vs. time of Navier–Stokes trial solutions at iteration $n = 200$. GD $\mathcal{J}_f^{\boldsymbol{u}}$ and LB $\mathcal{J}_f^{\boldsymbol{u}}$ denote DAL minimization of $\mathcal{J}_f^{\boldsymbol{u}}$ via gradient descent and L-BFGS respectively. SBI200 and QRM denote the trial solutions refined by SBI and QRM respectively. Every trial solution's error descreases nonmonotonically with respect to time. QRM has the smallest error at the initial and final times, though SBI200 has less error over large intervals in the time domain.
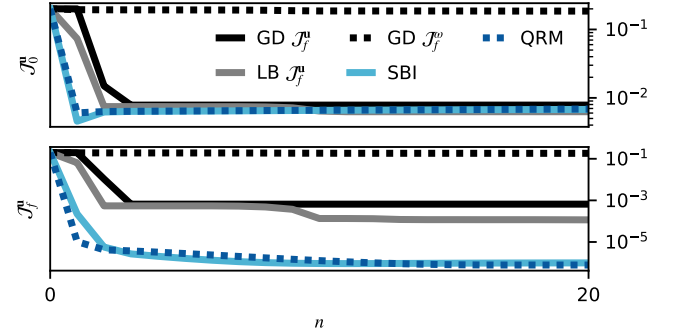

FIG. 9. Initial (top, $\mathcal{J}_0^{\boldsymbol{u}}$) and final (bottom, $\mathcal{J}_f^{\boldsymbol{u}}$) velocity errors for the 2D Navier–Stokes inverse problem over 20 iterations starting with an initial guess $\boldsymbol{u}^0(\boldsymbol{x}, 0) = \boldsymbol{0}$. We compare how Direct Adjoint Looping (DAL), Simple Backward Integration (SBI), and the Quasi-Reversible Method (QRM) minimize these errors. For DAL we compute the gradients of $\mathcal{J}_f^{\boldsymbol{u}}$ and $\mathcal{J}_f^{\omega}$ (shown black and grey). Using the gradient of $\mathcal{J}_f^{\boldsymbol{u}}$, we implement gradient descent and L-BFGS (solid black and solid grey respectively). $\mathcal{J}_f^{\boldsymbol{u}}$ decreases rapidly when SBI and QRM are used. Their respective performances are comparable in the first 20 iterations.

these methods can be iterated. SBI and QRM both decrease the errors by more than an order of magnitude in a single iteration. For $n > 150$ in Figure 10, SBI and QRM continue to gradually minimize $\mathcal{J}_f^{\boldsymbol{u}}$. This prolonged minimization of $\mathcal{J}_f^{\boldsymbol{u}}$ using iterative SBI is a surprising result of our study. In terms of $\mathcal{J}_0^{\boldsymbol{u}}$, every method fails to recover the target state due to ill-conditioning of this inverse problem. QRM surpasses the other methods in this respect, as illustrated in Figure 10.
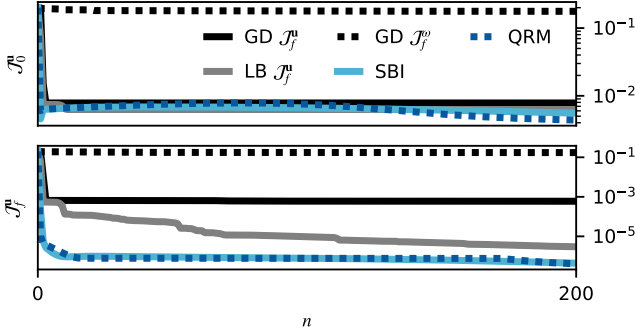
FIG. 10. Same as Figure 9 extended over 200 loops. DAL minimization of $\mathcal{J}_f^\omega$ (dotted black) is ineffective in terms of both $\mathcal{J}_0^{\boldsymbol{u}}$ (top) and $\mathcal{J}_f^{\boldsymbol{u}}$ (bottom). After the initial 20 iterations, every other method decreases $\mathcal{J}_0^{\boldsymbol{u}}$ by less than an order of magnitude. DAL minimization of $\mathcal{J}_f^{\boldsymbol{u}}$ using L-BFGS (grey, LB $\mathcal{J}_f^{\boldsymbol{u}}$) is the most effective DAL case in terms of both errors. At $n = 200$, LB $\mathcal{J}_f^{\boldsymbol{u}}$ does not minimize $\mathcal{J}_f^{\boldsymbol{u}}$ to the same extent as SBI and QRM do in very few iterations. Figures 6 and 7 illustrate their corresponding trial vorticities, where it is clear that LB $\mathcal{J}_f^{\boldsymbol{u}}$ is under-resolved and does not approximate the target initial state.

V. DERIVATION OF ILL-POSED NONLINEAR ADJOINT EQUATION

Consider the KdVB inverse problem from Sections II and III. Throughout this investigation, we assert that SBI and QRM perform better than DAL because they approximate the initial deviation rather than computing a gradient. In fact, the gradient of $\mathcal{J}_0^u$ taken with respect to $u(x,0)$ is the unknown deviation $u'(x,0) = u(x,0) - \overline{u}(x,0)$. $\mathcal{J}_0^u$ measures the squared distance between the trial and target initial states. Its levelsets form hyperspheres centered around $\overline{u}(x,0)$ and its paths of steepest descent are radial (pointing directly at $\overline{u}(x,0)$). The gradient of $\mathcal{J}_0^u$ embeds global information, whereas the gradient of $\mathcal{J}_f^u$ gives local information in the space of trial initial conditions.

Using optimal control, we will demonstrate that the ill-posed backward problem (eqn 9) is equivalent to the adjoint which minimizes $\mathcal{J}_0^u$. First we multiply eqn 9 by u' and integrate over the space-time domain, yielding $\mathcal{J}_0^u$ in terms of u', u, and $\mathcal{J}_f^u$.

$$\mathcal{J}_0^u = \mathcal{J}_f^u - \int_0^{t_f} \langle au' \partial_x^2 u' - u' \partial_x[uu'] \rangle dt. \tag{21}$$

Eqn 21 amounts to an energy equation for the deviation between our target and trial states. Notice how terms which are conservative of the L_2 norm cancel ($u' \partial_x u'$ and $b\partial_x^3 u'$) whereas terms with non-zero energy flux ($\partial_x[uu']$ and $a\partial_x^2 u'$) persist. Using eqn 21 we construct a new

Lagrangian

$$\hat{\mathcal{L}} = \int_0^{t_f} \langle \hat{\mu}(x,t) \cdot \mathcal{F}[u(x,t)] \rangle dt + \mathcal{J}_0^u \tag{22}$$

$$= \int_0^{t_f} \Big\langle \hat{\mu}[\partial_t u + u\partial_x u - a\partial_x^2 u + b\partial_x^3 u]$$
$$- au' \partial_x^2 u' + u' \partial_x[uu'] \Big\rangle dt + \mathcal{J}_f^u.$$

with a corresponding adjoint variable $\hat{\mu}$. Taking the variation of $\hat{\mathcal{L}}$ with respect to $\hat{\mu}(x,t)$ still returns the original constraint eqn 4 (on $0 < t < t_f$) and compatibility condition eqn 7 (at $t = t_f$). However, varying $\hat{\mathcal{L}}$ with respect to $u(x,t)$ along $0 < t < t_f$ gives a new adjoint equation

$$0 = -\partial_t \hat{\mu} - u\partial_x \hat{\mu} - a\partial_x^2 \hat{\mu} - b\partial_x^3 \hat{\mu}$$
$$- 2a\partial_x^2 u' + u' \partial_x u - u' \partial_x u'. \tag{23}$$

Finally, by summing eqns 9 and 23, it is evident that $\hat{\mu}(x,t) + u'(x,t) = 0$ not just at $t = t_f$, but throughout the time domain. We substitute $\hat{\mu} \to -u'$ in $\hat{\mathcal{L}}$'s adjoint, yielding

$$0 = \partial_t \hat{\mu} - a\partial_x^2 \hat{\mu} + b\partial_x^3 + u\partial_x \hat{\mu} + \hat{\mu}\partial_x u - \hat{\mu}\partial_x \hat{\mu}. \tag{24}$$

This new adjoint (equivalent to eqn 9) is ill-posed and effectively nonlinear. The same demonstration can be applied for any retrospective inverse problem with an advective constraint, including the Navier–Stokes inverse problem outlined in Section IV. If eqn 24 could be solved, we would obtain the exact deviation in our trial initial condition by performing a single adjoint loop. For example, the non-diffusive Korteweg–de Vries equation and the inviscid Euler equation admit well-posed nonlinear adjoints as these equations are time-reversible. However when $a > 0$ (diffusive forward problem), eqn 24 cannot be solved backwards using conventional numerical solvers. The ill-posed backward integration problem motivates the SBI and QRM modifications. As $\varepsilon \to 0$, QRM backward integration approaches the ill-posed nonlinear adjoint which minimizes $\mathcal{J}_0^u$. From this perspective, QRM is a regularized form of DAL. SBI backward integration combines the linear adjoint which minimizes $\mathcal{J}_f^u$ with the additional terms appearing in the ill-posed nonlinear adjoint.

Given that the SBI and QRM backward integrations return approximations for the gradient of $\mathcal{J}_0^u$, it follows that we can repeatedly apply the approximate gradients to refine our trial state. Taking the dynamical systems perspective, we can treat these iterative methods as discrete maps where the target state is a fixed point. Our results demonstrate that these fixed points are generally attracting for advection–diffusion systems. Potential difficulties might arize in future investigations due to limit-cycles or even strange attractors. We have simultaneously highlighted an important limitation of using gradients to optimize initial conditions: gradients can have large curvatures, even when the trial state is in close

proximity to the extremum (see Appendices A and B). By including additional terms in our backward integrations, we relax the gradient's sensitive dependence on the trial initial condition by replacing the gradient with a smoother discrete map. Higher-order algorithms for finding fixed points could also be paired with SBI and QRM to further accelerate convergence.

VI. DISCUSSION AND CONCLUSIONS

A. Summary of Results

In this investigation, we extend several previous studies involving Direct Adjoint Looping (DAL), Simple Backward Integration (SBI), and the Quasi-Reversible Method (QRM) by applying them to a pair of retrospective inverse problems. DAL has been used previously to solve inverse problems and optimize simulation inputs, as it allows us to compute the gradient of an arbitrary functional. This gradient can be used to refine the trial initial condition toward a local extremum. However, the number of iterations required to approximate said extremum depends on the user's initial guess.

The SBI and QRM methods are designed to approximate the trial solution's deviation rather than taking a gradient. Using the approximate deviation at the initial state, we update the trial initial condition using a "step size" of unity. In practice, we cannot compute the initial deviation because the necessary backward integration is ill-posed. SBI and QRM provide well-posed approximations for the ill-posed backward integration.

SBI has been used to make coarse approximations and generate initial guesses for further refinement via DAL. We extend this use by developing an iterative algorithm which acts on the trial state's deviation. With this method, we evolve the trial state's final deviation $u'(x, t_f)$ backwards in time using a modified constraint equation. This modified constraint equation is well-posed because we reverse the sign of the diffusive term.

Iterative QRM has previously been implented. With this backward integration, we preserve the ill-posed diffusion term by introducing a small hyperdiffusion term to relax the numerical instability. We must also tune this hyperdiffusion term's coefficient ε according to the problem's dynamics and numerical resolution.

Our first inverse problem is constrained by the Korteweg–de Vries–Burgers (KdVB) equation. The target solution consists of a nonlinear wave whose advective velocity decays due to diffusion. We demonstrate that the gradient of $\mathcal{J}_f^u$, obtained via DAL or otherwise, is not an efficient means of recovering the target initial state. DAL performs poorly when coupled with gradient descent as well as the L-BFGS optimization routine. When following the direction of steepest descent in the space of trial initial conditions, we accumulate undesired wave features. This is due to high curvature in the path of steepest descent. QRM captures the target final state

almost immediately while SBI does so more gradually.

The second inverse problem is constrained by the 2D incompressible Navier–Stokes equation. The target solution is a symmetric Kelvin–Helmholtz vortex with $\mathrm{Re} \equiv \nu^{-1} = 50,000$. We evolve the target state over 20 advective time units, then use the final state to recover its associated initial condition. The problem domain is doubly-periodic which likely reduces the ill-conditioning of our retrospective inverse problem. We apply each iterative method until iteration $n = 200$. Here we utilize the flexibility of DAL by minimizing the velocity error $\mathcal{J}_f^u$ as well the vorticity error $\mathcal{J}_f^\omega$. DAL minimization of $\mathcal{J}_f^\omega$ performs poorly as illustrated in Figures 6, 9 and 10. Using L-BFGS to minimize $\mathcal{J}_f^u$, we produce an initial condition whose final state resembles the target after 200 iterations (Figure 6). This L-BFGS case successfully minimizes $\mathcal{J}_f^u$, as illustrated in Figure 10. However, the trial initial condition is under-resolved because L-BFGS accumulates small-scale features. This highlights severe ill-conditioning in the Navier–Stokes inverse problem, as we are able to minimize the final error without approximating the target initial condition. We implement gradient descent with double resolution to demonstrate this more clearly. This gradient descent case does not approximate the target final state as well as L-BFGS, even though the simulations are well-resolved.

Just as with KdVB, SBI and QRM outperform DAL by every metric in the Navier–Stokes inverse problem. Although every method decreases $\mathcal{J}_0^u$ with a similar trend (Figures 9 and 10), SBI and QRM (Figure 7) approximate the target initial state better than DAL (Figure 6) at $n = 200$. The methods' respective behaviors differ in close proximity to their target, as illustrated in Figure 10. Specifically, QRM exhibits nonmonotonicity in both $\mathcal{J}_0^u$ and $\mathcal{J}_f^u$ whereas SBI decreases these quantities with each iteration.

B. Future Work

Optimal control via DAL is the most common approach for retrospective inverse problems. However, SBI and QRM are more effective than DAL when applied to the 2D Navier–Stokes inverse problem. In future work, we will implement SBI and QRM for 3D Navier–Stokes inverse problems. Chaos and the direct energy cascade (as opposed to the inverse energy cascade in 2D) will likely present additional challenges. We will also use SBI and QRM to solve other optimal control problems, such as the mantle convection inverse problems studied by [4] and [5]. Numerical weather prediction (NWP) could also be enhanced by implementing these nonlinear methods. SBI and QRM are well-suited for this application because NWP models are increasingly nonlinear. Predictions are limited by our ability to incorporate real-time observations in a reasonable number of iterations [30].

Another relevant study is [31], which computes time-

periodic solutions of vortex sheets with surface tension. Using DAL, they minimize a cost functional which resembles $\mathcal{J}_f^u$. Although they are not afforded a target final state, we can interpret their adjoint loop as a dynamic inversion where the target state changes with each iteration. Conceptually, any properly-constrained optimal control problem has a corresponding retrospective inverse problem. The compatibility condition may differ (e.g. [31] initializes their adjoint with the deviation between their trial initial and final states), but in general the adjoint evolution equations remain the same. Given that our SBI and QRM loops tend toward the mutual extremum of $\mathcal{J}_0^u$ and $\mathcal{J}_f^u$, it is reasonable to suggest that these methods might also have attracting extrema when seeking periodic solutions. Efficient algorithms for computing time-periodic solutions would have direct applications in studying cyclic dynamos and anticipating nonlinear resonances in engineering systems such as turbines.

Appendix A: KdVB Inverse Problem Objectives using SBI Initial Guess

For the KdVB inverse problem described in Section III, we initialize $u^0(x, 0)$ using an initial guess constructed via SBI. This SBI initial guess is computed by evolving $\overline{u}(x, t_f)$ backwards using a modified constraint equation (eqn 4 with the diffusive term's sign reversed). For DAL, this implementation mimics [4, 5] who demonstrated that their SBI initial guess converges more rapidly compared to a neutral guess of zero.

Using the SBI initial guess does improve our DAL approximations of $\overline{u}(x, 0)$, as shown in the top row of Figure 11. L-BFGS (top right) continues to outperform gradient descent (top left), though both cases have significant deviations from $\overline{u}(x, 0)$. SBI and QRM both outperform the DAL cases, converging in approximately the same number of iterations as in Section III.

Figure 12 plots the initial (top) and final (bottom) errors for the KdVB inverse problem using the SBI initial guess. LB $\mathcal{J}_f^u$ (shown in grey) performs appreciably better than in Figure 5. However SBI and QRM both surpass DAL, with QRM performing best by several orders of magnitude in terms of $\mathcal{J}_f^u$.

Appendix B: Navier–Stokes Inverse Problem Objectives using SBI Initial Guess

For the Navier–Stokes inverse problem described in Section IV, we compare each method (DAL, SBI, and QRM) starting with an initial guess generated via SBI. This case mimics [4, 5], who began by rewinding their target state from $t : t_f \to 0$ using SBI. They subsequently refined their SBI initial guess by using DAL to minimize an error functional (analogous to DAL $\mathcal{J}_f^{\boldsymbol{u}}$, shown in black). Figures 13 and 14 plot the initial ($\mathcal{J}_0^{\boldsymbol{u}}$, top) and final ($\mathcal{J}_f^{\boldsymbol{u}}$, bottom) velocity errors as functions of iteration n. Figure 13 shows the first 20 iterations while Figure 14 terminates at $n = 200$. In the black curve (GD $\mathcal{J}_f^{\boldsymbol{u}}$), we use DAL with gradient descent to minimize $\mathcal{J}_f^{\boldsymbol{u}}$.

First, we confirm that using an SBI initial guess with DAL does approach the target final state faster than in Section IV. Next we implement SBI and QRM as before. Just as with the previous initial guess $\boldsymbol{u}^0(\boldsymbol{x}, 0) = \boldsymbol{0}$, we find that SBI and QRM outperform DAL, albeit by a narrower margin (ranging from less than one to two orders of magnitude in terms of $\mathcal{J}_f^{\boldsymbol{u}}$). Crucially, SBI and QRM both outperform DAL in very few iterations, suggesting that the efficacy of these methods is somewhat robust to the initial guess. Every iterative method increases the initial error $\mathcal{J}_0^{\boldsymbol{u}}$ (top) compared to the SBI initial guess, except for QRM which eventually decreases $\mathcal{J}_0^{\boldsymbol{u}}$ below its original value near $n \approx 190$ (shown in dotted dark blue). Iterative SBI appears to follow a similar trend at a slower pace, whereas GD $\mathcal{J}_f^{\boldsymbol{u}}$ shows no such trend. This behavior with GD $\mathcal{J}_f^{\boldsymbol{u}}$ indicates that our Navier–Stokes inverse problem is highly non-convex, even in close proximity to the extremum. In this case, L-BFGS minimization of $\mathcal{J}_f^{\boldsymbol{u}}$ gives unstable trial initial conditions. We obtain similar results by initializing each iterative method using the QRM trial initial state at $n = 200$.

REFERENCES

[1] D. Lukyanenko, T. Yeleskina, I. Prigorniy, T. Isaev, A. Borzunov, and M. Shishlenin, Inverse problem of re-
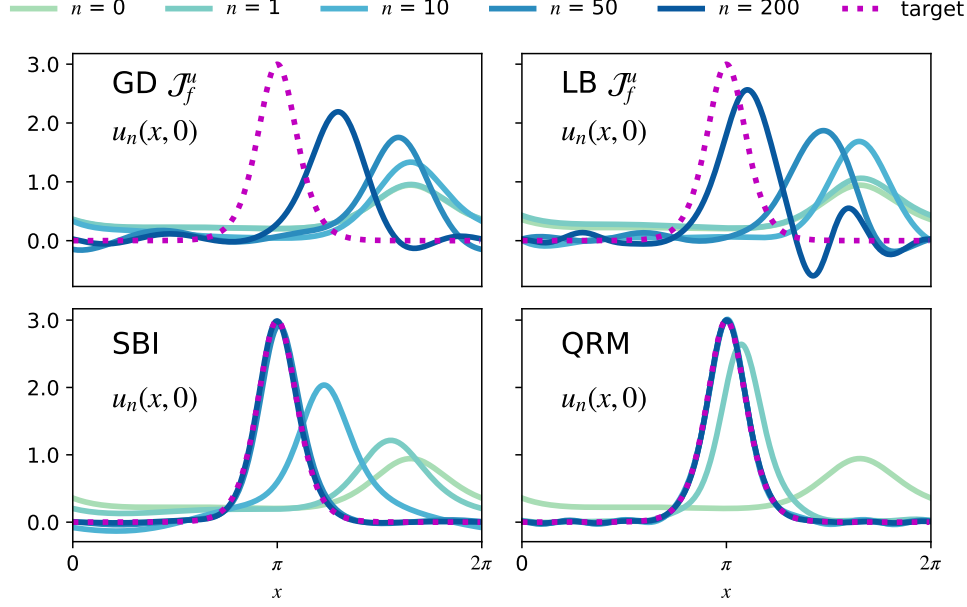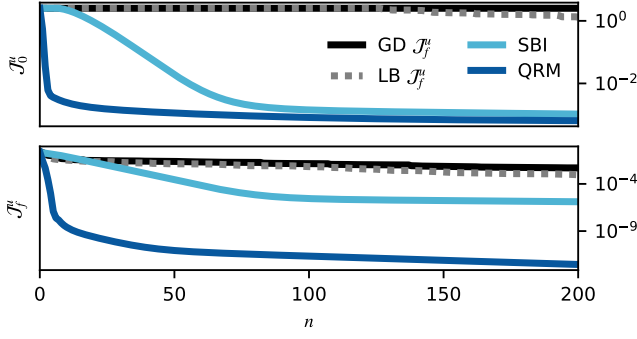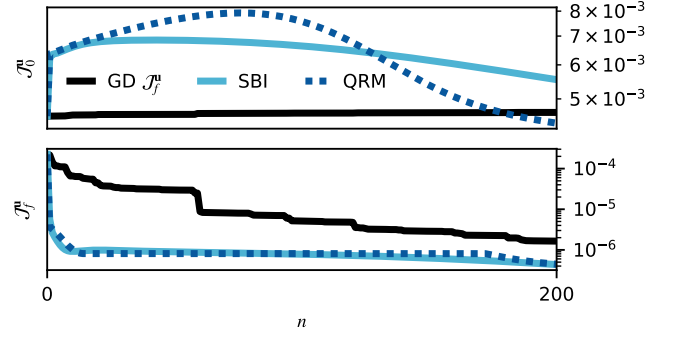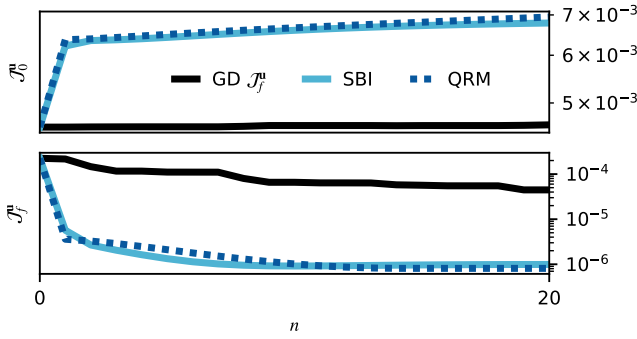covering the initial condition for a nonlinear equation of

FIG. 11. Same is Figure 3 using the SBI initial guess (shown in green).

FIG. 12. Same is Figure 5 using initial guess $u^0(x,0)$ generated via SBI.

FIG. 14. Same as Figure 10 using initial guess $\boldsymbol{u}^0(\boldsymbol{x},0)$ generated via SBI.

FIG. 13. Same as Figure 9 using initial guess $\boldsymbol{u}^0(\boldsymbol{x},0)$ generated via SBI.

the reaction–diffusion–advection type by data given on the position of a reaction front with a time delay, Mathematics **9**, 1 (2021).

[2] S. Subramanian, K. Scheufele, M. Mehl, and G. Biros, Where did the tumor start? an inverse solver with sparse localization for tumor growth models, Inverse Problems **36**, 10.1088/1361-6420/ab649c (2020).

[3] S. I. Kabanikhin, I. M. Kulikov, and M. A. Shishlenin, An algorithm for recovering the characteristics of the initial state of supernova, Computational Mathematics and Mathematical Physics **60**, 10.1134/S0965542520060135 (2020).

[4] L. Liu and M. Gurnis, Simultaneous inversion of mantle properties and initial conditions using an adjoint of mantle convection, Journal of Geophysical Research: Solid Earth **113**, 10.1029/2008jb005594 (2008).

[5] D. Li, M. Gurnis, and G. Stadler, Towards adjoint-based inversion of time-dependent mantle convection with non-

linear viscosity, Geophysical Journal International **209**, 10.1093/gji/ggw493 (2017).

[6] O. Y. Imanuvilov, Local exact controllability for the 2-d navier-stokes equations with the navier slip boundary conditions, Turbulence Modeling and Vortex Dynamics , 148 (1997).

[7] J. Fan and G. Nakamura, Well-posedness of an inverse problem of navier-stokes equations with the final overdetermination, Journal of Inverse and Ill-Posed Problems **17**, 565 (2009).

[8] J. Fan, M. D. Cristo, Y. Jiang, and G. Nakamura, Inverse viscosity problem for the navier-stokes equation, Journal of Mathematical Analysis and Applications **365**, 750 (2010).

[9] J. Du, I. M. Navon, J. Zhu, F. Fang, and A. K. Alekseev, Reduced order modeling based on pod of a parabolized navier–stokes equations model ii: Trust region pod 4d var data assimilation, Computers & Mathematics with Applications **65**, 380 (2013).

[10] K. J. Law and A. M. Stuart, Evaluating data assimilation algorithms, Monthly Weather Review **140**, 3757 (2012).

[11] T. Fan, K. Xu, J. Pathak, and E. Darve, Solving inverse problems in steady-state navier-stokes equations using deep neural networks (2020), arXiv:2008.13074 [math.NA].

[12] T. Frerix, D. Kochkov, J. A. Smith, D. Cremers, M. P. Brenner, and S. Hoyer, Variational data assimilation with a learned inverse observation operator (2021), arXiv:2102.11192 [cs.LG].

[13] Q. Zhao, D. B. Lindell, and G. Wetzstein, Learning to solve pde-constrained inverse problems with graph networks (2022), arXiv:2206.00711 [cs.LG].

[14] S. Mowlavi and S. Nabi, Optimal control of PDEs using physics-informed neural networks, Journal of Computational Physics **473**, 111731 (2023).

[15] P. M. Mannix, C. S. Skene, D. Auroux, and F. Marcotte, Discrete adjoint-based control: A robust gradient descent procedure for optimisation with pde and norm constraints (2022), arXiv:2210.17194 [physics.flu-dyn].

[16] R. Lattès, J. Lions, and R. Bellman, *The Method of Quasi-reversibility: Applications to Partial Differential Equations* (American Elsevier Publishing Company).

[17] C. P. Conrad and M. Gurnis, Seismic tomography, surface uplift, and the breakup of gondwanaland: Integrating mantle convection backwards in time, Geochemistry, Geophysics, Geosystems **4**, 10.1029/2001GC000299 (2003).

[18] L. Fang, W. J. T. Bos, S. Liang, J.-P. Bertoglio, and L. Shao, Time reversibility of navier-stokes turbulence and its implication for subgrid scale models, Journal of Turbulence **13**, 1 (2012).

[19] J. Dardé, Iterated quasi-reversibility method applied to elliptic and parabolic data completion problems, Inverse Problems and Imaging **10**, 379 (2016).

[20] L. H. Nguyen and M. V. Klibanov, Carleman estimates and the contraction principle for an inverse source problem for nonlinear hyperbolic equations, Inverse Problems **38**, 035009 (2022).

[21] O. Y. Imanuvilov, L. Lorenzi, and M. Yamamoto, Carleman estimate for the Navier–Stokes equations and applications, Inverse Problems **38**, 085006 (2022), publisher: IOP Publishing.

[22] W. Steiner and S. Reichl, The optimal control approach to dynamical inverse problems, Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME **134**, 10.1115/1.4005365/455882 (2012).

[23] J. Barzilai and J. M. Borweigh, Two-Point Step Size Gradient Methods, IMA Journal of Numerical Analysis **8**, 141 (1988), https://academic.oup.com/imajna/article-pdf/8/1/141/2402762/8-1-141.pdf.

[24] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM Journal on Scientific Computing **16**, 1190 (1995), https://doi.org/10.1137/0916069.

[25] In addition to the results shown in section III, we perform numerical experiments on the KdVB inverse problem using conjugate-gradient and line search routines (provided by scipy version 1.12.0). In both cases we use the default parameters. For the line search, we use the negative gradient for the descent direction.

[26] SBI and QRM do not generally return any sensible gradient, so the specified updates are not technically gradient descent. Attempting a constant step size of $\gamma \sim \mathcal{O}(1)$ with DAL is unstable for the direct minimization of $\mathcal{J}_f^u$. SBI can often be accelerated using a larger step size.

[27] K. J. Burns, G. M. Vasil, J. S. Oishi, D. Lecoanet, and B. P. Brown, Dedalus: A flexible framework for numerical simulations with spectral methods, Physical Review Research **2**, 023068 (2020).

[28] S. Nabi, P. Grover, and C. P. Caulfield, Robust preconditioned one-shot methods and direct-adjoint-looping for optimizing reynolds-averaged turbulent flows, Computers and Fluids **238**, 10.1016/j.compfluid.2022.105390 (2022).

[29] R. R. Kerswell, C. C. T. Pringle, and A. P. Willis, An optimization approach for analysing nonlinear stability with transition to turbulence in fluids as an exemplar, Reports on Progress in Physics **77**, 085901 (2014).

[30] M. Bonavita, P. Lean, and E. Holm, Nonlinear effects in 4d-var, Nonlinear Processes in Geophysics **25**, 713 (2018).

[31] D. Ambrose, M. Kondrla, and M. Valle, Computing time-periodic solutions of a model for the vortex sheet with surface tension, Quarterly of Applied Mathematics **73**, 10.1090/s0033-569x-2015-01364-8 (2015).