

— See overleaf comment

Iterative Looping Methods for Navier–Stokes Inverse Problems

Liam O'Connor^{1,2}, Daniel Lecoanet^{1,2}, Evan Anders², Kyle Augustson^{1,2}, Keaton Burns³, Geoff Vasil⁴, Jeffrey S. Oishi⁵, Ben Brown⁶

¹Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, IL 60208 USA

²Center for Interdisciplinary Exploration and Research in Astrophysics, Northwestern University, Evanston, IL, 60201 USA

³Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02142 USA

⁴School of Mathematics, Edinburgh University, EH9 3FD, UK

⁵Department of Physics and Astronomy, Bates College, Lewiston, Maine 04240 USA

⁶Department of Astrophysical and Planetary Sciences, University of Colorado Boulder, Boulder, CO 80309 USA

Abstract. 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 Navier–Stokes inverse problems. 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 looping methods: Simple Backward Integration (SBI) and the Quasi-Reversible Method (QRM). Our iterative SBI approach is novel while iterative QRM has been used previously in more limited capacities. Using these three iterative looping methods, we solve two retrospective inverse problems which have seen minimal attention: 1D Korteweg–de Vries–Burgers (decaying nonlinear wave) and 2D Navier–Stokes (unstratified Kelvin–Helmholtz vortex). In both cases, SBI and QRM outperform DAL by an astonishing margin. We attribute this performance gap to SBI and QRM having additional nonlinear terms in their respective backward integrations which are not present in DAL.

*SBI
& QRM's*

absent

(present)

*vague ; colloquial.
outperform how? By what
measure ; how much?*

1. Introduction

Do you need to define \mathcal{F} ?

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

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

We aim to determine the unique initial condition $\bar{X}(x, 0)$ corresponding to a given final state $\bar{X}(x, t_f)$ where the final time t_f is known. These so-called “retrospective inverse

problems are often ill-conditioned or ill-posed, even when the forward problem is readily solvable. *solvability* *appear (?)*

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] examined the prospects of inverting tumor growth using Gaussian basis functions to localize their origins. [3] inverted supersonic supernova expansions governed by the 1D compressible Navier–Stokes equation. Studies of inverse problems for convection in Earth’s mantle are abundant [4, 5]. Buoyancy-driven flows in Earth’s mantle operate in the Stokes regime where $Pr \gg 1$ implies that velocity can be expressed purely as a function of temperature. Computational geophysicists in many cases have successfully approximated both the initial temperature distribution as well as variable parameters such as the thermal diffusivity [4, 5].

Although there are problem-specific details in their methodologies, [1, 2, 3, 4, 5] use the same optimal control technique known as Direct Adjoint Looping (DAL). Using DAL, they minimize the cost functional [1]

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

where the angled brackets $\langle \cdot \rangle$ denote a spatial integral over the problem domain and the trial solution X has some deviation $X' \equiv X - \bar{X}$. To traverse the space of initial conditions, they construct sequences of trial solutions $\{X^n(x, t)\}$, where each element X^n is computed from a corresponding trial initial condition $X^n(x, 0)$. Because we are generally required to provide this initial guess, convergence to the target state $\bar{X}(x, 0)$ is seldom guaranteed. *is provided by the user*

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 on, 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 (constant) viscosity was studied analytically [8].

More recently, researchers have elaborated on these analytical studies of Navier–Stokes inverse problems by implementing data assimilation algorithms. Two noteworthy efforts involve dimensionality reduction of the solution manifold via Principle Orthogonal Decomposition (POD): [9] studies an inverse problem governed by the parabolized Navier–Stokes (PNS) equation. They identify a trust region of solutions where the POD basis is accurate. They then navigate this trust region using four-dimensional variational (4DVAR), an optimal control technique analogous to DAL. Following this, [10] provides a comprehensive overview of data assimilation algorithms used for non-deterministic Navier–Stokes inverse problems. Their findings confirm that a Bayesian

perspective is appropriate when dealing with Navier–Stokes inverse problems arising in numerical weather prediction due to the spurious effects of noisy data. Even more recently, [11, 12] used artificial neural networks and deep learning to solve analogous Navier–Stokes data-completion problems.

Taken as a whole, these efforts confirm that Navier–Stokes inverse problems can be prohibitively difficult, especially in the presence of data uncertainty. 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. In this paper, we compare the performance of various iterative methods (such as DAL) by applying them to a retrospective Navier–Stokes inverse problem.

With DAL, we ~~can~~ minimize an arbitrary cost functional ~~such as~~ \mathcal{J}_f^X . First, we must solve the forward problem, which consists of the PDE constraint 1 on $t : 0 \rightarrow t_f$. At this point, we ~~can~~ evaluate \mathcal{J}_f^X and compare the ~~trial/target final states~~ ($X(x, t_f)$ and $\bar{X}(x, t_f)$ respectively). Information from time $t = t_f$ is then propagated backward, to time $t = 0$, by solving the linear adjoint problem $\mathcal{F}^\dagger = 0$ on $t : t_f \rightarrow 0$. Adjoint backward integration gives ~~us~~ a gradient which approximates the functional derivative $\frac{\delta \mathcal{L}}{\delta X}(x, 0)$. Using this gradient, we ~~can~~ adjust our trial initial condition by some amplitude γ : $X(x, 0) \rightarrow X(x, 0) - \gamma \frac{\delta \mathcal{L}}{\delta X}(x, 0)$. Quasi-newton methods have also been implemented to accelerate the algorithm. Recent efforts involving the discrete adjoint formulation achieved a significant performance gain for optimal control problems involving norm constraints on the initial condition [13]. [5] employs a preconditioner matrix motivated by a singular limit ~~arizing~~ in their discretization. Although these efforts broaden the range of problems we can address, we will later demonstrate that DAL’s shortcomings in the context of Navier–Stokes inversion are due to missing terms in the backward integration, as opposed to how the adjoint is formulated (discrete or POD) or how gradient information is used (first-order or Quasi-Newton).

The additional 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 an ill-posed backward calculation. SBI accomplishes this by ~~simply~~ reversing the sign of the problematic diffusion term, i.e. $a\partial_x^2 X \rightarrow -a\partial_x^2 X$. In contrast, QRM is a regularization framework which preserves the ill-posed term. This is enabled by appending a small hyperdiffusive term to the otherwise ill-posed constraint equation, i.e. $a\partial_x^2 X \rightarrow a(\partial_x^2 + \varepsilon\partial_x^4)X$ where $0 < \varepsilon \ll 1$.

We introduce these concepts as backward integration methods rather than iterative looping methods because their previous implementations were rarely iterated. SBI has never been iterated whereas [14] examined the iterative application of QRM when applied to an ill-posed linear inverse problem. Thanks to linearity, [14] derives rigorous statements of convergence and monotonicity over several QRM iterations. We elaborate on [14] by providing a straightforward comparison with DAL for the nonlinear Navier–Stokes inverse problem.

[15] introduced SBI to infer the history of Earth’s convective mantle along with its role
 used

trial ($X(t)$) and
target ($\bar{X}(t)$)
final states

in the southern African superswell. Following this, [4] and [5] repurposed this technique to construct initial guesses, which they then refined iteratively using the conventional DAL algorithm. SBI was also examined by [16] to dismiss a time-reversibility hypothesis on Subgrid-Scale (SGS) Large-Eddy Simulations (LES). As was the case with viscous mantle convection, LES were shown to be non-reversible, especially over long durations ($t_f \gg 1$). A DNS with viscosity cannot be reversed in the trivial sense due to some losses in precision at the floating point error.

We compare DAL, SBI, and QRM by solving two retrospective inverse problems, respectively constrained by KdV-Burgers and incompressible Navier–Stokes. The novelty of our work lies in the *iterative* application of QRM and SBI. Rather than solving for an adjoint variable, we use backward integration to approximate the trial state's deviation $X'(x, t)$. When minimizing \mathcal{J}_f^X via DAL, we initialize the adjoint variable with $-X'(x, t_f)$. A related (but inaccessible) objective functional

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

has its own gradient $X'(x, 0)$. With SBI/QRM, we repeatedly approximate $X'(x, 0)$ to refine the trial state $X(x, 0)$ and, in doing so, we achieve a significant performance increase.

The remainder of this paper is organized as follows: In Section II we provide a conceptual description of our iterative looping methods (DAL, SBI, and QRM). In Section III we illustrate these methods by applying them to a retrospective Korteweg–de Vries–Burgers (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 conclude by contextualizing each method into a general conceptual framework involving gradients and control theory principles.

2. Korteweg–de Vries–Burgers (KdVB): Overview of Methods

The KdVB equation

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

provides a concise context for demonstrating each method (DAL, SBI, QRM). Let $\bar{u}(x, t)$ denote the target solution where $0 < t < t_f$ in a 1D periodic domain $0 < x < L_x$. The retrospective inverse problem is solved by recovering the target initial state $\bar{u}(x, 0)$ where the target final state $\bar{u}(x, t_f)$ is given. The general strategy has us construct trial solutions $u(x, t)$ satisfying (4), which are computing from trial initial states $u(x, 0)$. We measure a trial state's proximity from the target using cost functionals \mathcal{J}_0^u and \mathcal{J}_f^u , which are defined for an arbitrary variable X in equations 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.

when

*There are lots of measures of performance.
What kind of performance improves?*

2.1. Direct Adjoint-Looping (DAL)

In general we can ~~caste~~ an inverse problem into the form of an optimization problem by defining and minimizing an appropriate cost functional [17]. DAL provides a highly 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 can refine the trial initial state $u(x, 0)$ by following the direction of steepest-descent. ~~Though our choice of cost functional is somewhat arbitrary, it is analogous to a huge variety of studies involving similar, but more intricate problems with direct applications [1, 2, 3, 4, 5].~~

With DAL, we can ~~(in theory)~~ compute the local extremum of any arbitrary objective functional without varying the individual constituents of $u(x, 0)$. Gradient-based minimization of any cost functional (via DAL or otherwise) *fundamentally differs* from our repeated applications of SBI and QRM. With DAL we have access to a huge suite of direct optimization routines which have already been implemented to increase performance in a number of problems. In this case the gradient can be obtained as follows:

First let us define the Lagrangian \mathcal{L} and associated Lagrangian multiplier

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

\mathcal{L} 's variations with respect to u and μ must disappear at the optimum \bar{u} , yielding

$$0 < t < t_f : \quad 0 = \mathcal{F}[u] = \partial_t u + u \partial_x u - a \partial_x^2 u + b \partial_x^3 u \quad (6)$$

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

$$0 < t < t_f : \quad 0 = \mathcal{F}^\dagger[u, \mu] = \partial_t \mu + u \partial_x \mu + a \partial_x^2 \mu + b \partial_x^3 \mu. \quad (8)$$

The gradient of \mathcal{J}_f^u resides in the manifold of solutions satisfying 6, 7, and 8. ~~Figure 1~~ move Fig 1 closer to this text. illustrates how these equations are solved sequentially. Given a trial initial condition, we start by solving the forward problem 6 from $t : 0 \rightarrow t_f$ (shown in blue). Next, we initialize the adjoint variable using the compatibility condition 7. Finally, we solve the adjoint equation 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.

Once \mathcal{J}_f^u 's gradient ($\mu(x, 0)$) is known, we use gradient-descent (GD) with the Barzilai-Borwein method [18] to select step sizes. In addition to GD, we also implement the second-order sparse L-BFGS-B algorithm [19] to minimize \mathcal{J}_f^u .

2.2. SBI/QRM Setup

SBI/QRM aim to approximate the deviation u' whereas DAL computes a gradient. From a numerical perspective, we cannot determine $u'(x, t < t_f)$ due to the intrinsic loss of information in diffusive systems. From an analytical perspective, we cannot solve the

ill-posed backward calculation because of the problem term $-a\partial_x^2 u$ in 6. Let $\tilde{\mu}$ denote our SBI/QRM approximations for $-u'$.

Should this be 2.2.1?

2.3. Simple Backward Integration (SBI)

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

$$\partial_t \tilde{\mu} + \tilde{\mu} \partial_x u + u \partial_x \tilde{\mu} + \tilde{\mu} \partial_x \tilde{\mu} + a\partial_x^2 \tilde{\mu} + b\partial_x^3 \tilde{\mu} = 0. \quad (9)$$

This backward integration is well-posed. In fact eqn 9 is identical to 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 9 with $\tilde{\mu}(x, t_f) = -u'(x, t_f)$, just as with eqn 7 with DAL. We then solve eqn 9 on $t : t_f \rightarrow 0$, yielding an approximation $\tilde{\mu}(x, 0) \approx -u'(x, 0)$.

After completing an SBI loop, we update the trial state $u(x, 0) \rightarrow u(x, 0) - \tilde{\mu}(x, 0)$. This is analogous to performing GD with a fixed step-size of unity§.

In previous investigations, SBI is implemented or scrutinized within a framework where the entire solution is treated as time-reversible in the sense that $a\partial_x^2 u \approx -a\partial_x^2 u$. Our iterative implementation of SBI differs, because we implement this hypothesis on the deviation rather than the full solution. [16] demonstrated that Navier–Stokes is fundamentally irreversible in the sense that $a\partial_x^2 u \not\approx -a\partial_x^2 u$. Our iterative approach for SBI is not necessarily incompatible with this finding. 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.]

2.4. Quasi-Reversible Method (QRM)

These last few sentences feel out of place

QRM deals with the problem term by introducing an additional, higher-order term: $a\partial_x^2 \tilde{\mu} \rightarrow a(\partial_x^2 \tilde{\mu} + \varepsilon \partial_x^4 \tilde{\mu})$.

$$\partial_t \tilde{\mu} + \tilde{\mu} \partial_x u + u \partial_x \tilde{\mu} + \tilde{\mu} \partial_x \tilde{\mu} - a(\partial_x^2 \tilde{\mu} + \varepsilon \partial_x^4 \tilde{\mu}) + b\partial_x^3 \tilde{\mu} = 0. \quad (10)$$

$0 < \varepsilon \ll 1$ is a free parameter chosen in accordance with the numerical method as well as the target/trial states' dynamics. Notice how the ill-posed term remains, but the backward integration becomes well-posed because modes of arbitrarily high order do not grow at arbitrarily fast rates.

The QRM loop is carried out using the same procedure as SBI. We initialize 10 with $\tilde{\mu}(x, t_f) = -u'(x, t_f)$ and update the trial state $u(x, 0) \rightarrow u(x, 0) - \tilde{\mu}(x, 0)$.

§ SBI/QRM do not generally return any sensible gradient, so the specified updates are not formally GD. 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

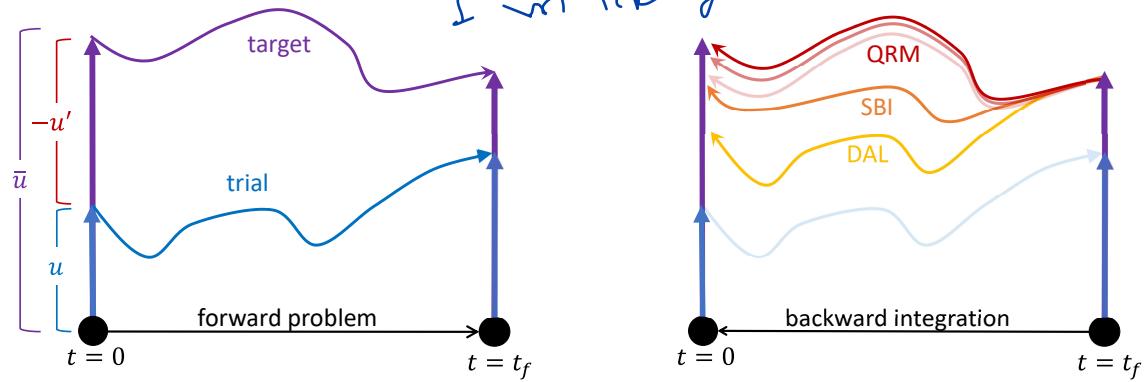


Figure 1. Conceptual illustration of iterative looping methods. Purple: the target solution $\bar{u}(x, t)$ is known only at the final time t_f . Blue: we evolve a trial state $u(x, 0)$ to $u(x, t_f)$ where we then initialize one of three backward integration systems using a single compatibility condition. Each backward integration gives a solution at $t = 0$ which we then use to update our trial state. The trial solution's deviation $u' = u - \bar{u}$ is known only at the final time t_f and is sought at the initial time $t = 0$. Red: the Quasi-Reversible Method (QRM) aims to compute this deviation, despite this calculation being ill-posed when $\varepsilon = 0$. Yellow: Direct Adjoint Looping (DAL) is a linear backward integration. For advective systems such as DAL advects the final deviation $u'(x, t_f)$ by the trial solution u . Orange: Simple Backward Integration (SBI) is a hybrid method, which combines the nonlinear terms of QRM with the linear adjoint.

I don't think you ever describe the various shades of red?

which are evolve from

3. KdVB: Numerical Experiments

We implement each iterative looping method (DAL, SBI, QRM) to solve an inverse problem constrained by KdVB (Equation 4). The target solution $\bar{u}(x, t)$ (shown left in Fig. 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 over $t \in 0 \rightarrow 3\pi$. We specify a target initial condition

$$\bar{u}(x, 0) = 3 \cosh^{-2} \left(\frac{x - \pi}{2\sqrt{b}} \right), \quad (11)$$

which would propagate rightwards as a stable soliton with constant speed $\Delta x / \Delta t = 1$ if diffusion were not present (i.e. if $a = 0$ in Equation 4). The 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 [20]. 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 as grid points in memory. 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 performing backwards integration, where additional terms such as $\mu \partial_x u$ must be treated explicitly

Say what version you use

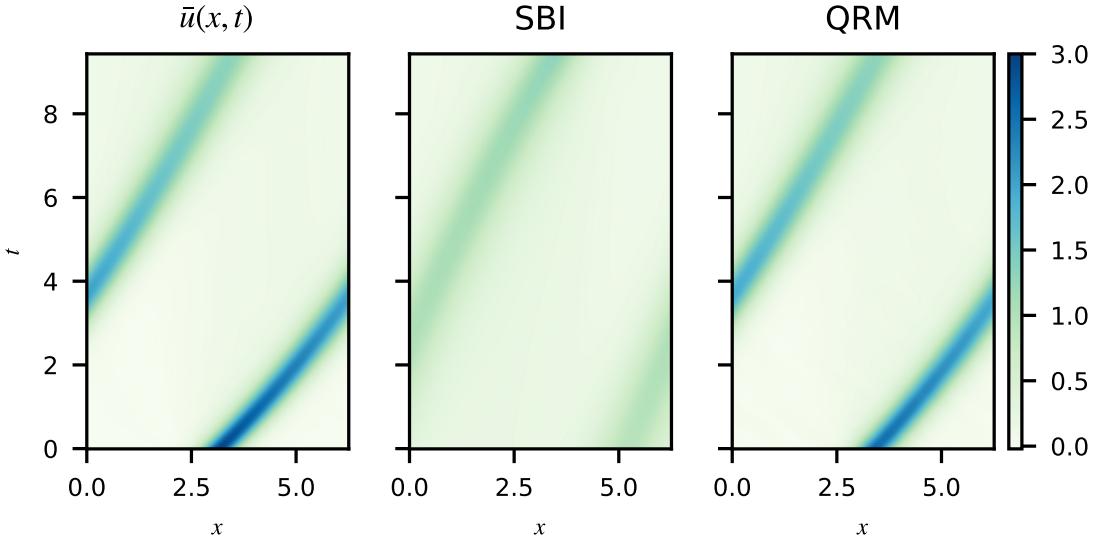


Figure 2. Target KdVB solution $\bar{u}(x, t)$ (left) plotted as a function of space x and time t . We aim to recover the initial condition of a nonlinear wave which propagates right at an initial speed of $\Delta x/\Delta t = 1$. The problem’s spatial domain is 1D periodic. We include diffusion ($a = 0.04$) such that its speed decreases as the wave propagates. SBI backward integration (center) is used to generate an initial guess for $\bar{u}(x, 0)$ in Appendix 6.1. Beginning at $t = 3\pi$, SBI evolves $\bar{u}(x, t_f)$ backward to $t = 0$ by reversing the sign of the diffusive term. The wave diffuses during forward and backward integration, such that its amplitude and speed of propagation decrease. The QRM backward integration (right) is performed on $t : 3\pi \rightarrow 0$. Unlike SBI, QRM repopulates small-scale modes which are lost during forward integration. As the waves peak regains its former amplitude, its speed of propagation increases mimicing $\bar{u}(x, t)$.

to avoid dense matrix solves at every timestep. For every solve we use a 2nd-order Runge-Kutta scheme with a fixed timestep $dt = 0.01$. (cite timestep paper)

We initialize all looping algorithms using an initial guess $u^0(x, 0) = 0$. (See Appendix 6.1 for the same comparison beginning with an SBI initial guess.) The first SBI loop iteration is shown in the center plot of Figure 2. Excessive diffusion of the nonlinear wave causes its amplitude and speed of propagation to decrease such that the wave’s peak does not return to its original position. The first QRM loop iteration (shown right in Figure 2) reconstructs the wave by inverting the advection ^{both} alongside ^{and} diffusion. In this case, the wave’s peak returns to its approximate initial position. For QRM we use an experimentally-determined free parameter $\varepsilon = 0.01$ whereas SBI involves no free parameter.

Figure 3 illustrates the evolution of trial initial states $u(x, 0)$ as they are refined via each method. Over 200 loop iterations, DAL gradually constructs a new wave peak which increases in magnitude while also moving left toward the target feature. This behavior is due to the local minimization of \mathcal{J}_f^u , as the final trial states’ peaks do coincide with the target. DAL with L-BFGS-B approaches the target state more rapidly than DAL with gradient-descent. However, L-BFGS-B introduces additional wave features which do not appear in the target. QRM approximates the target wave form in a single

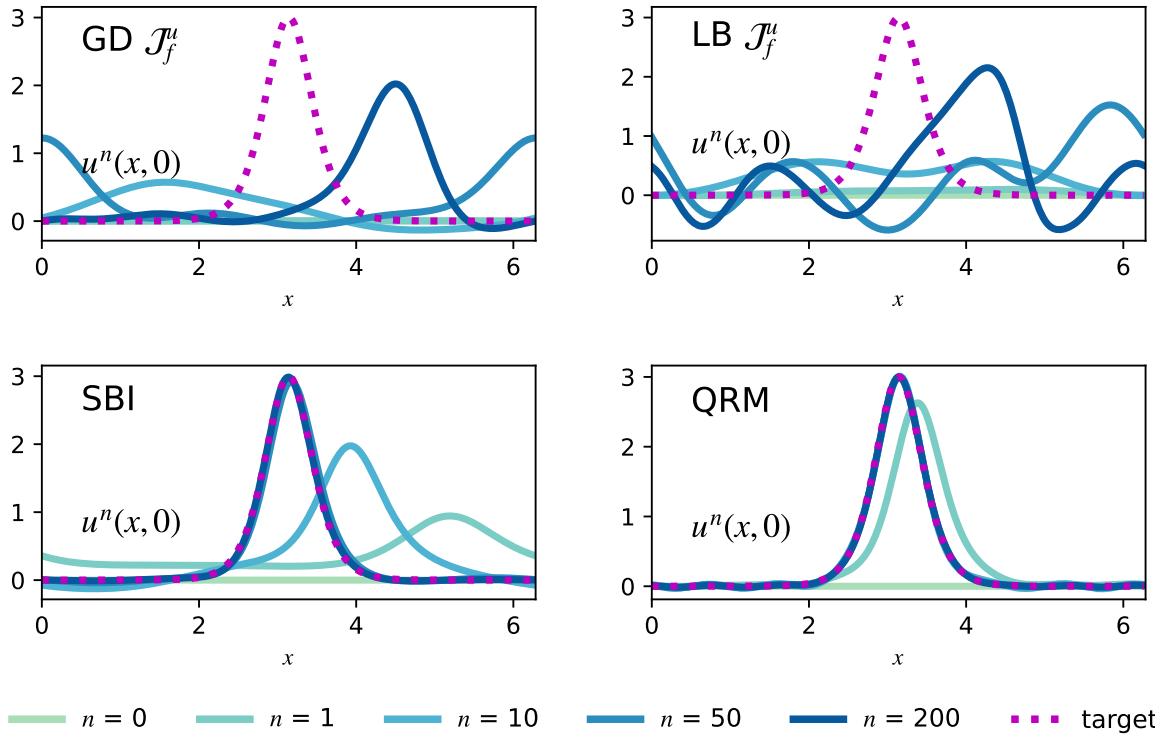


Figure 3. Evolution of a KdVB trial initial condition $u^n(x, 0)$ using various iterative looping methods. We initialize with $u(x, 0) = 0$, as shown in green. (See Appendix 6.1, Figure 10 for the same study using the SBI initial guess) (Top left) DAL minimization of \mathcal{J}_f^u using gradient-descent approaches the target state gradually. (Top right) DAL minimization of \mathcal{J}_f^u using L-BFGS-B approaches the target state more rapidly, but this method introduces undesired small-scale features. (Bottom left) SBI captures the target state much faster than DAL without introducing small-scale errors. (Bottom right) QRM captures the target state immediately ($n = 1$). However, this method introduces low-amplitude small-scale errors which do not subside at $n = 200$.

iteration by approximating the initial deviation $u'(x, 0)$. At $n = 10$, its trial state is nearly indistinguishable from the target. However, at $n = 200$ we observe small-scale oscillations in the QRM trial state which do not exist in the target. SBI does not effectively approximate $u'(x, 0)$ (as shown in Figure 2) nor does it follow the direction of steepest-descent of \mathcal{J}_f^u . It smoothly guides the trial states' peaks directly toward the target, mimicing aspects of both DAL and QRM.

Results from each optimization study are illustrated in Figure 11. DAL does not effectively minimize \mathcal{J}_0^u or \mathcal{J}_f^u . In similar investigations, L-BFGS-B has received significant attention thanks to its DAL-compatible features. It frequently converges faster than GD which is also reflected in our experiment. In this case, DAL's inefficiency supersedes our choice of gradient minimization routine. The L-BFGS-B algorithm embeds curvature estimations in its sparse Hessian reconstruction, allowing us to follow the gradient at a slightly accelerated rate. Our findings suggest that even in this highly-controlled 1D case, the L-BFGS-B algorithm still does not yield a quality approximation.

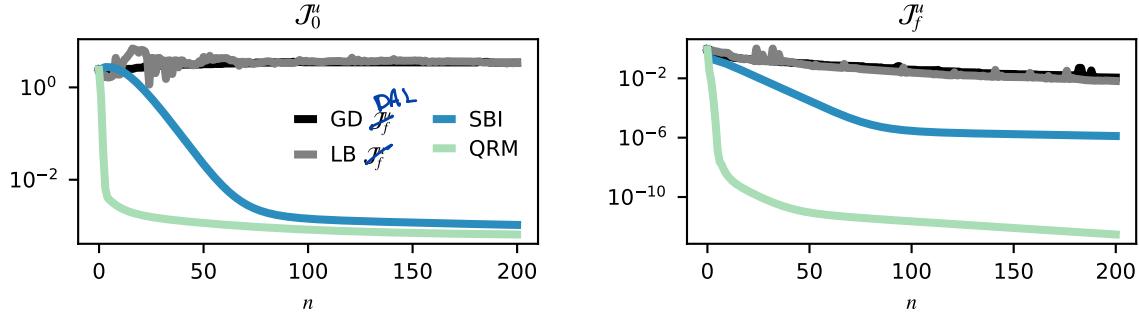


Figure 4. Initial errors (J_0^u , left) and final errors (J_f^u , right) for the 1D KdVB inverse problem plotted as a function of loop iteration n . We compare DAL, SBI, and QRM. DAL performs slightly better when paired with the second-order sparse minimization routine L-BFGS-B (LB J_f^u), whereas Gradient-Descent (GD J_f^u) performs the worst. Our iterative applications of SBI and QRM minimize J_0^u and J_f^u much more effectively than DAL. QRM minimizes both J_0^u and J_f^u more rapidly than SBI.

of $\bar{u}(x, 0)$ in a reasonable number of iterations.

SBI and QRM looping algorithms both outperform DAL by a tremendous margin. QRM minimizes both J_0^u and J_f^u more rapidly than SBI. QRM has a simple advantage compared to SBI: with QRM applied to a highly diffusive system, the energy $\|\tilde{\mu}(x, t)\|^2$ ought to increase during backward integration (provided ε is sufficiently small) such that $\|\tilde{\mu}(x, 0)\|^2 \sim \|u'(x, 0)\|^2$. This is illustrated in the first QRM loop iteration (Figure 2, right). In contrast, SBI diffuses during backward integration (Figure 2, center). Therefore, even if there was no interaction between advection and diffusion, SBI is always curtailed by the loop's round-trip diffusion.

4. 2D Navier–Stokes

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

$$\partial_t \bar{\mathbf{u}} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \nabla \bar{p} = \nu \nabla^2 \bar{\mathbf{u}} \quad \text{and} \quad \nabla \cdot \bar{\mathbf{u}} = 0 \quad (12)$$

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

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

$$\bar{\mathbf{v}}(\mathbf{x}) \cdot \hat{\mathbf{x}} = \frac{1}{2} \left[\tanh \left(10 \left(y - \frac{1}{2} \right) \right) - \tanh \left(10 \left(y + \frac{1}{2} \right) \right) \right]; \quad (13)$$

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

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

For our inverse problem, we run a target simulation by evolving $\bar{\mathbf{u}}(\mathbf{x}, 0)$ according to 12 until $t_f = 20$. The target solution's vorticity evolution is illustrated in the top row of Figure 5. The initial state has a thin layer of positive vorticity concentrated near $y = 1/2$. Advection causes this feature to wind counter-clockwise while viscosity smooths its small-scales.

Given the final velocity field $\bar{\mathbf{u}}(\mathbf{x}, t_f)$, we develop trial solutions $\mathbf{u}(\mathbf{x}, t)$ (also obeying 12) which are meant to approach the target solution $\bar{\mathbf{u}}(\mathbf{x}, t)$. All PDE solves (except the GD \mathcal{J}_f^u double-resolution study) were performed on a 128 by 256 grid of real Fourier modes using the **Dedalus** pseudospectral framework [20]. We timestep with $\Delta t = 0.002$ using a second-order Runge-Kutta IMEX 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 $y = 0$.

4.1. DAL

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

$$\partial_t \boldsymbol{\mu} + \mathbf{u} \cdot \nabla \boldsymbol{\mu} + \nabla \Pi + \nu \nabla^2 \boldsymbol{\mu} = \mathbf{0}. \quad (15)$$

[21] provides a detailed derivation of 15. For our inverse problem, we use DAL to minimize two objective functionals with associated compatibility conditions

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

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

Is this grammatically correct? ✓ The skew-gradient $\nabla^\perp \equiv (-\partial_y, \partial_x)$ and the vorticity $\omega \equiv \nabla^\perp \cdot \mathbf{u}$. As with KdVB, apostrophes denote the deviation of a trial variable from its target, i.e. $\omega' \equiv \omega - \bar{\omega}$ and so on. \mathcal{J}_f^u quantifies the trial states' proximity to the target at the final time t_f in terms of velocity whereas \mathcal{J}_f^ω does so using vorticity. We implement the gradient-descent (GD) and L-BFGS-B optimization routines to minimize these two cost functionals. We also measure the corresponding functionals \mathcal{J}_0^u and \mathcal{J}_0^ω evaluated at the initial state $t = 0$.

4.2. SBI/QRM

Our iterative procedures for SBI/QRM are analogous to those described in the previous section. The ill-posed backward integration of \mathbf{u}' is just the nonlinear perturbation equation

$$\mathbf{0} = \partial_t \mathbf{u}' + \mathbf{u} \cdot \nabla \mathbf{u}' + \mathbf{u}' \cdot \nabla \mathbf{u} - \mathbf{u}' \cdot \nabla \mathbf{u}' + \nabla p' - \nu \nabla^2 \mathbf{u}', \quad (18)$$

where the negative nonlinear term follows from the definition of \mathbf{u}' . Next we substitute the approximations $-\tilde{\boldsymbol{\mu}}$ for \mathbf{u}' and $-\tilde{\Pi}$ for p' . For SBI, we reverse the sign of the viscous

term $\nu \nabla^2 \tilde{\mu}$. Its corresponding approximation for 18 is given by

$$0 = \partial_t \tilde{\mu} + \mathbf{u} \cdot \nabla \tilde{\mu} + \tilde{\mu} \cdot \nabla \mathbf{u} + \tilde{\mu} \cdot \nabla \tilde{\mu} + \nabla \tilde{\Pi} + \nu \nabla^2 \tilde{\mu}. \quad (19)$$

For QRM, we introduce a small hyperdiffusive term, approximating 18 as

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

We illustrate QRM using $\varepsilon = 0.001$ after confirming that this choice performs better than $\varepsilon = 0.01$.

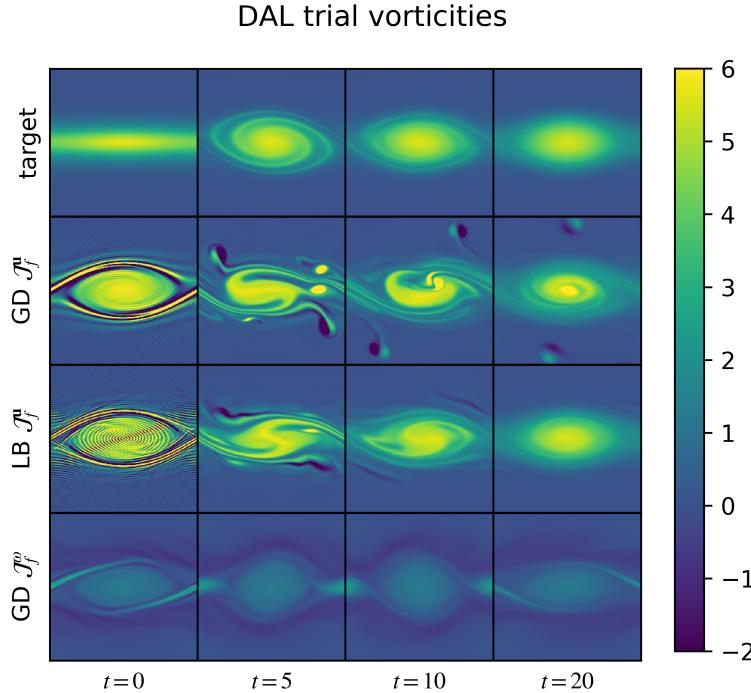


Figure 5. Target (top row) and trial vorticity snapshots at simulation times $t = 0, 5, 10, 20$. The second, third, and bottom row trial solutions are obtained by applying DAL over 200 loop iterations. GD \mathcal{J}_f^u (second row) uses gradient-descent to minimize \mathcal{J}_f^u via DAL. For this case, we use double resolution in space and time (256 by 512 modes, $\Delta t = 0.001$) to ensure that each solve is resolved. Despite this extra resolution, we observe symmetry breaking when $t \gg 0$ due to numerical errors. LB \mathcal{J}_f^u (third row) uses L-BFGS-B to minimize \mathcal{J}_f^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. This confirms that our Navier–Stokes inverse problem is ill-conditioned. GD \mathcal{J}_f^w (bottom row) uses gradient-descent to minimize \mathcal{J}_f^w via DAL. The DAL trial initial conditions (bottom three rows) have small-scale banded features with high concentrations of positive vorticity whose final states have varying degrees of resemblance to the target at $t = 20$.

4.3. Results

a lot of this is repeated in the caption & main text. Describe the figure physically here, but put interpretation in the main text. Aim for short captions.

We initialize each looping algorithm using an initial guess $\mathbf{u}^0(\mathbf{x}, 0) = \mathbf{0}$. For DAL, we minimize the velocity objective using gradient-descent (GD \mathcal{J}_f^u) and L-BFGS-B (LB

\mathcal{J}_f^u). We run GD \mathcal{J}_f^u at double resolution to ensure that the trial state's evolution is well-resolved. We also minimize the vorticity objective via gradient-descent (GD \mathcal{J}_f^ω). L-BFGS-B minimization of \mathcal{J}_f^ω frequently gives unstable trial initial conditions which cannot be evolved without additional resolution. Figure 5 consists of vorticity snapshots belonging to the target solution and trial solutions at $t = 0, 5, 10, 20$. The trial solutions in these cases were all obtained using DAL until loop iteration $n = 200$. GD \mathcal{J}_f^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^u . LB \mathcal{J}_f^u (third row) is clearly under-resolved at $t = 0$, with the same highly-concentrated features as GD \mathcal{J}_f^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 $t \rightarrow 20$. GD \mathcal{J}_f^ω (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.

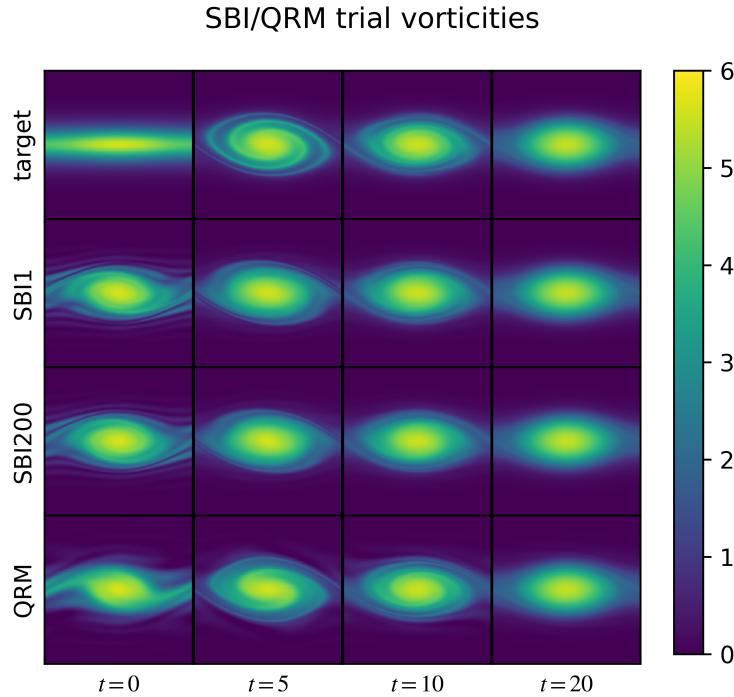


Figure 6. Same as Figure 5 using the SBI and QRM iterative looping methods. Despite the ill-conditioning which hinders DAL in Figure 5, SBI and QRM both recover the target initial condition (top left) to a greater extent. SBI1 and SBI200 (middle two rows) denote SBI trial states after 1 and 200 looping iterations respectively. We do not observe significant qualitative evolution of the trial state with SBI after loop iteration $n = 1$. QRM (bottom row) almost recovers the target initial condition at $n = 200$. Qualitatively, its structure resembles an unwinded spiral.

Figure 6 illustrates the analogous trial vorticity snapshots using SBI and QRM. In the

top row, we ~~reproduce~~^{show} the target. The second row (SBI1) denotes the trial state after a single SBI loop iteration. Its initial state resembles the target's better than any of the DAL runs in Figure 5. Its final state is nearly indistinguishable from the target. The third row (SBI200) denotes the SBI trial solution after 200 loop iterations. Qualitatively, we observe no significant refinement toward the target as compared to SBI1. The bottom row (QRM) denotes the trial solution after 200 QRM loop 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 is nearly indistinguishable from the target. In general, SBI and QRM capture the target state better than DAL without introducing the undesired small-scale features shown in Figure 5.

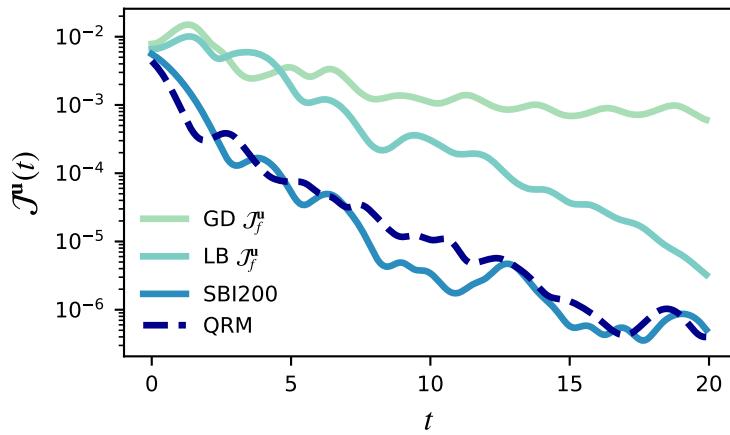


Figure 7. Velocity error $\mathcal{J}^u(t) \equiv \frac{1}{2}\langle|\mathbf{u}(\mathbf{x}, t) - \bar{\mathbf{u}}(\mathbf{x}, t)|^2\rangle$ vs. time of Navier–Stokes trial solutions at loop iteration $n = 200$. GD \mathcal{J}_f^u and LB \mathcal{J}_f^u denote DAL minimization of \mathcal{J}_f^u via gradient-descent and L-BFGS-B respectively. SBI200 and QRM denote the trial solutions refined by SBI and QRM respectively. Every trial solution's error decreases nonmonotonically as the simulation evolves toward the final state. QRM has the smallest error at the initial and final states, though SBI200 has less error for large intervals inside time domain.

Figure 7 gives the time-dependent velocity error $\mathcal{J}^u(t) \equiv \frac{1}{2}\langle|\mathbf{u}(\mathbf{x}, t) - \bar{\mathbf{u}}(\mathbf{x}, t)|^2\rangle$ for the trial solutions GD \mathcal{J}_f^u , LB \mathcal{J}_f^u , SBI, and QRM at loop iteration $n = 200$. Due to the inverse-problem's ill-conditioning, all trial states begin with substantially larger errors than their respective final states. These errors decrease without monotonicity on $t : 0 \rightarrow 20$. The DAL trial solutions' (GD \mathcal{J}_f^u and LB \mathcal{J}_f^u) errors increase on $0 < t < 2$ due to as their small-scale vortex bands (illustrated in the left column of Figure 5) rapidly disappear. LB \mathcal{J}_f^u performs significantly better than GD \mathcal{J}_f^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 by huge margins.

Figure 8 tracks the error functionals \mathcal{J}_0^u , \mathcal{J}_0^ω , \mathcal{J}_f^u , and \mathcal{J}_f^ω vs. loop iteration n for each iterative looping method until $n = 20$. Every method's performance is comparable

Show off details

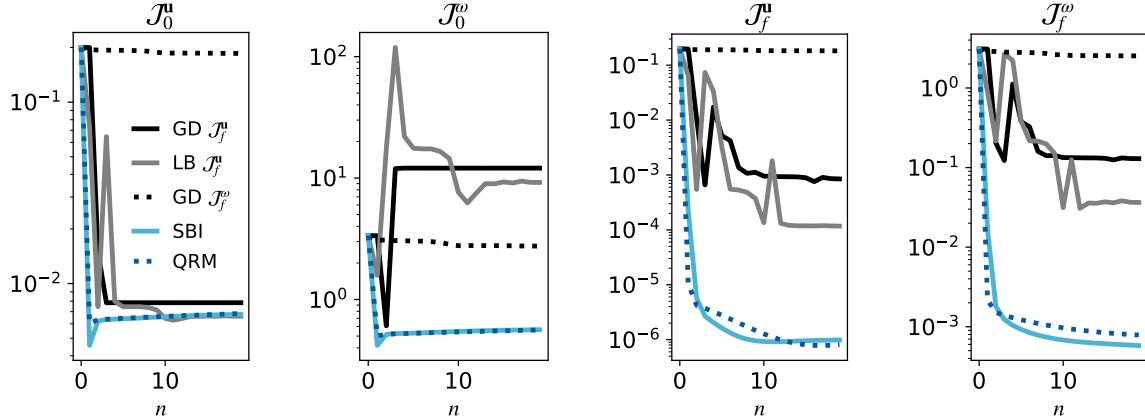


Figure 8. Cost functionals for the 2D Navier–Stokes inverse problem over 20 iterations starting with an initial guess $\mathbf{u}^0(\mathbf{x}, 0) = \mathbf{0}$. (See Appendix 6.2 for the same study beginning with an SBI initial guess.) We compare how Direct Adjoint Looping (DAL), Simple Backward Integration (SBI), and the Quasi-Reversible Method (QRM) minimize these cost-functionals. For DAL we compute the gradients of \mathcal{J}_f^u and \mathcal{J}_f^w (shown black and grey). For \mathcal{J}_0^u , we implement gradient-descent (GD) and L-BFGS-B (solid black and solid grey respectively). These two algorithms cause \mathcal{J}_0^u to increase whereas GD \mathcal{J}_f^w slightly decreases \mathcal{J}_f^w . Using L-BFGS-B to minimize \mathcal{J}_f^w , we frequently encounter trial initial conditions which cannot be evolved without additional resolution. All errors decrease smoothly and rapidly when SBI/QRM are used, and their respective performances are comparable in the first 20 loop iterations.

in terms of \mathcal{J}_0^u (left) except GD \mathcal{J}_f^w which does not appreciably decrease this error. The other two DAL cases (LB \mathcal{J}_f^u and GD \mathcal{J}_f^u) cause the initial vorticity error \mathcal{J}_0^w (center left) to increase rather than decrease. GD \mathcal{J}_f^w performs slightly better by slowly decreasing this error metric. In terms of the final velocity (center right) and final vorticity (right) errors, LB \mathcal{J}_f^u performs best among DAL runs, followed by GD \mathcal{J}_f^u , followed by GD \mathcal{J}_f^w . SBI (light blue solid) and QRM (dark blue dashed) perform similarly well in all cases, immediately outperforming each DAL minimization by substantial margins.

Figure 9 extends the metrics from Figure 8 to loop iteration $n = 200$. Both DAL gradient-descent algorithms (GD \mathcal{J}_f^u and GD \mathcal{J}_f^w) stall after $n = 20$, whereas LB \mathcal{J}_f^u continues to minimize the final errors \mathcal{J}_0^u (center right) and \mathcal{J}_f^w (right) at steady rates. We observe sharp spikes with LB \mathcal{J}_f^u as the L-BFGS-B algorithm surveys the local curvature as the trial state evolves. The relative success of LB \mathcal{J}_f^u indicates that the gradient of \mathcal{J}_f^u has large curvature. SBI and QRM continue to outperform all DAL runs, with QRM performing slightly better in \mathcal{J}_0^u and \mathcal{J}_0^w . QRM decreases the errors nonmonotonically whereas SBI exhibits a smooth decline which generally surpasses QRM in terms of \mathcal{J}_f^u and \mathcal{J}_f^w .

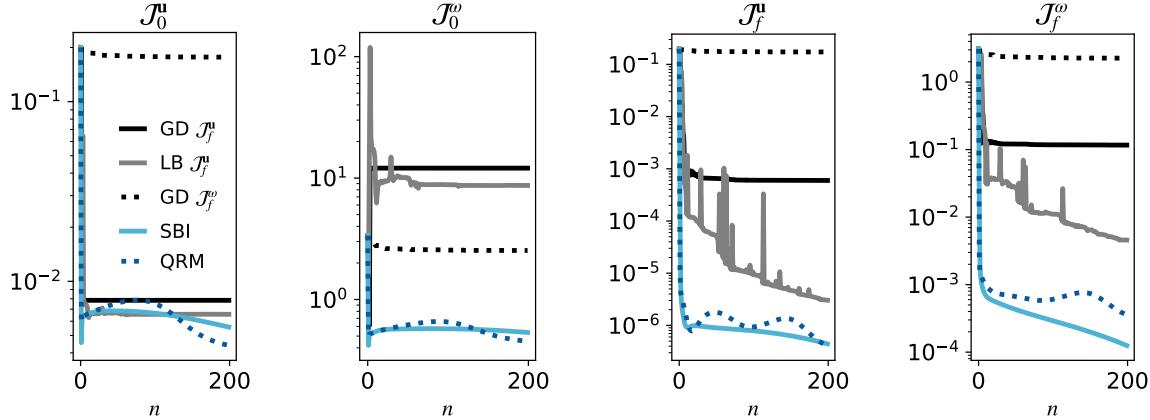


Figure 9. Same as 8 extended over 200 loops. [Notice how DAL minimization of \mathcal{J}_f^u using L-BFGS-B (grey, corresponding to LB \mathcal{J}_f^u) performs adequately after many iterations, whereas SBI/QRM perform well in only a few iterations. Figure 5 illustrates the corresponding trial vorticities, where it is clear that LB \mathcal{J}_f^u fails to approximate the target initial state. DAL minimization of \mathcal{J}_f^w performs the worst by every metric except \mathcal{J}_0w where continues to do better than both other DAL runs. QRM does not decrease monotonically in any of the four metrics whereas SBI exhibits smooth convergence.] → This should all be in the main text.

5. Conclusion

In this investigation, we elaborate on 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 of simulation inputs, as it allows us to compute the gradient of an arbitrary objective/cost functional. This method is a linearization about the trial state, making its behavior highly dependent on the trial state's locality within the manifold of solutions. DAL delivers a gradient which, in theory, can be used to refine the trial initial condition toward a local extremum.

The SBI/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/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. From the perspective of optimal control, SBI is identical to DAL with the inclusion of two additional terms, one of which is nonlinear.

Iterative QRM has been used previously in more limited capacities. With this backward integration, we preserve the ill-posed diffusion term by introducing a small hyperdiffusive term to relax the numerical instability. With QRM we must tune this hyperdiffusive 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 viscosity. 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-B optimization routine. When following the direction of steepest-descent in the space of trial initial states, we accumulate undesired small-scale features. This is due to high curvature in the manifolds of steepest-descent. QRM captures the target final state almost immediately in this case 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 $\text{Re} \equiv \nu^{-1} = 50,000$. We evolve the target state over 20 advective time units, then use the final state to recover its associated initial condition. We apply each iterative looping method until loop 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^ω . DAL minimization of \mathcal{J}_f^ω performs very poorly as illustrated in Figures 5, 8 and 9. Using L-BFGS-B to minimize \mathcal{J}_f^u , we produce an initial condition whose final state resembles the target after 200 loop iterations (Figure 5). This L-BFGS-B case successfully minimizes the cost functionals \mathcal{J}_f^u and \mathcal{J}_f^ω , as illustrated in Figure 9. However, the trial initial condition is under-resolved because the path of steepest-descent accumulates small-scale features as in the KdVB problem. This highlights severe ill-conditioning in the Navier–Stokes inverse problem, as we were able to minimize final errors 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-B, even though the simulations are well-resolved.

Just like with KdVB, SBI/QRM outperform DAL by every metric in the Navier–Stokes inverse problem. Both methods approximate the target initial state in only a few loop iterations ($n \sim 10$, Figure 8). The methods’ respective behaviors differ in close proximity to their target, as illustrated in Figure 9. Specifically, QRM exhibits non-monotonicity in every cost functional (\mathcal{J}_0^u , \mathcal{J}_0^ω , \mathcal{J}_f^u , and \mathcal{J}_f^ω) whereas SBI decreases these quantities with each loop iteration.

SBI/QRM are meant to minimize \mathcal{J}_0^u , whose paths of steepest-descent are radial (pointing directly at \bar{u}). The gradient of \mathcal{J}_0^u is simply the unknown deviation $u'(x, 0) = u(x, 0) - \bar{u}(x, 0)$. This gradient captures global information related to the trial solutions u and their deviations u' . In Appendix 6.3, we demonstrate that the adjoint system corresponding to \mathcal{J}_0^u is equivalent to the ill-posed nonlinear perturbation equation. The QRM backward integration approaches this ill-posed nonlinear adjoint as $\varepsilon \rightarrow 0$. SBI is a hybrid method which combines the linear adjoint of \mathcal{J}_f^u with the additional adjoint

terms from \mathcal{J}_0^u .

Although the SBI/QRM iterative looping methods do not provide us with conventional gradients, we can treat these algorithms as discrete maps. From a dynamical systems perspective, the target state is a fixed point. Our results demonstrate that these fixed points are generally attracting for advective-diffusive systems. Potential difficulties might arise in future investigations due to limit-cycles or even strange attractors. However, we have also highlighted an important limitation of using gradients to optimize initial conditions: gradients of functionals do not generally point toward the extremum, even when the trial state is in close proximity to the extremum (see Appendices 6.1 and 6.2). By including nonlinear terms in our backward integrations, we relax the gradient's sensitive dependence on the trial state 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.

Optimal control via DAL is the defacto approach for dealing with retrospective inverse problems. In future work, we will apply the SBI and QRM iterative looping methods to other optimal control problems. One relavent study is [22], who computes time-periodic solutions for the Benjamin-Ono equation using optimal control. Using DAL, they minimize a cost functional which resembles \mathcal{J}_f^u . Although they are not afforded a target final state, we can still interpret their adjoint loop as a dynamic inverse problem where the target state changes with each iteration. Conceptually, any properly-constrained optimal control problem has a corresponding inverse problem. The compatibility condition may differ, but in general the adjoint system is the same. For example, [22] initializes their adjoint with the discrepancy between their trial initial and final states. 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 our iterative looping methods might do the same when seeking periodic solutions. Efficient algorithms for computing time-periodic solutions would have direct applications in studying cyclic dynamos and forseeing nonlinear resonances in engineering systems such as turbines.

6. Appendix

6.1. KdVB Inverse Problem Objectives using SBI Initial Guess

In Figure 10, we initialize $u^0(x, 0)$ using an initial guess constructed via SBI. This SBI initial guess is computed by evolving $\bar{u}(x, t_f)$ backwards using a modified constraint equation (Equation 4 with the diffusive term's sign reversed). For DAL, this implementation mimics [5, 4] who demonstrated that their SBI initial guess converges more rapidly compared to a neutral guess of zero.

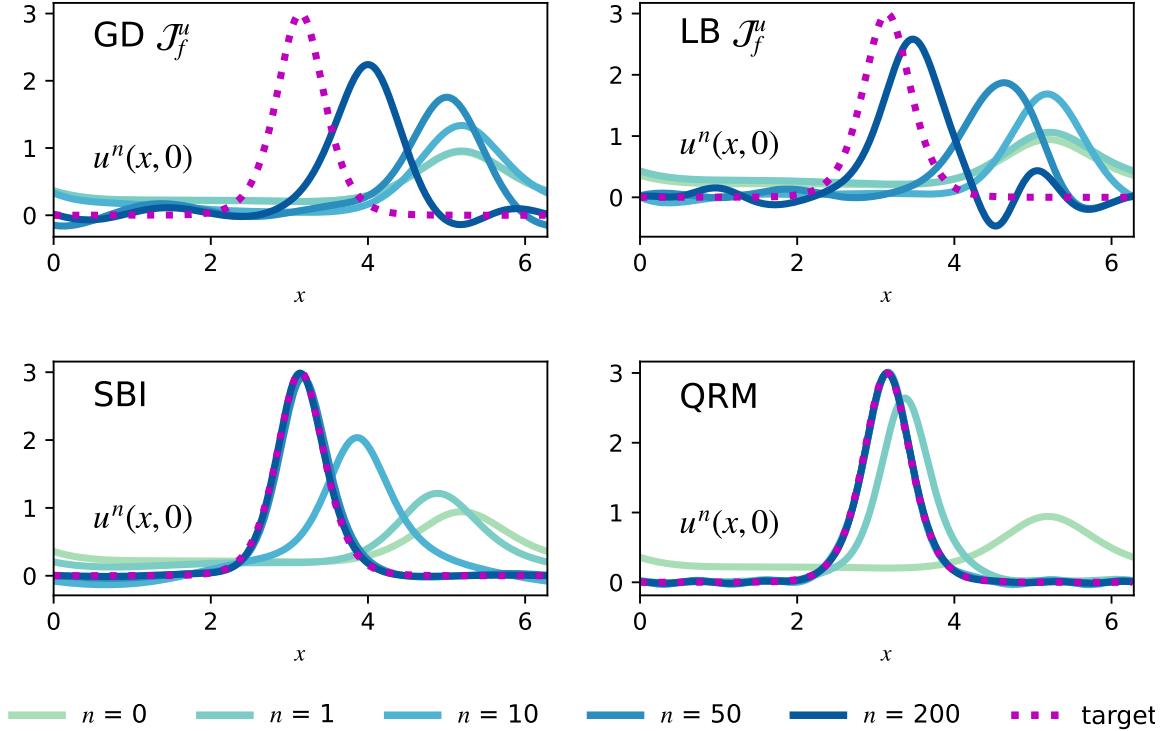


Figure 10. Evolution of a KdVB trial initial condition $u^n(x, 0)$ using various iterative looping methods. We initialize with the SBI-generated guess (shown in green, see Figure 2). (Top left) DAL minimization of \mathcal{J}_f^u using gradient-descent approaches the target state gradually. (Top right) DAL minimization of \mathcal{J}_f^u using L-BFGS-B approaches the target state more rapidly, but this method introduces undesired small-scale features. (Bottom left) SBI captures the target state much faster than DAL without introducing small-scale errors. (Bottom right) QRM captures the target state immediately ($n = 1$). However, this method introduces low-amplitude small-scale errors which do not subside at $n = 200$.

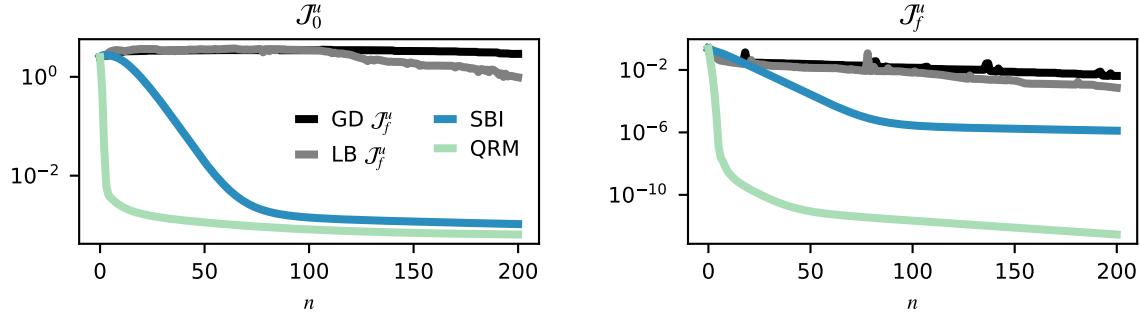


Figure 11. Initial errors (J_0^u , left) and final errors (J_f^u , right) for the 1D KdVB inverse problem plotted as a function of loop iteration n . We compare DAL, SBI, and QRM. DAL performs adequately when paired with the second-order sparse minimization routine L-BFGS-B (LB J_f^u), whereas Gradient-Descent (GD J_f^u) performs the worst. Our iterative applications of SBI and QRM minimize J_0^u and J_f^u much more effectively than DAL. QRM minimizes J_f^u most rapidly whereas SBI surpasses QRM in J_0^u near iteration 25. This shortcoming of QRM is due to the small-scale errors observed in Figure 10

6.2. Navier–Stokes Inverse Problem Objectives using SBI Initial Guess

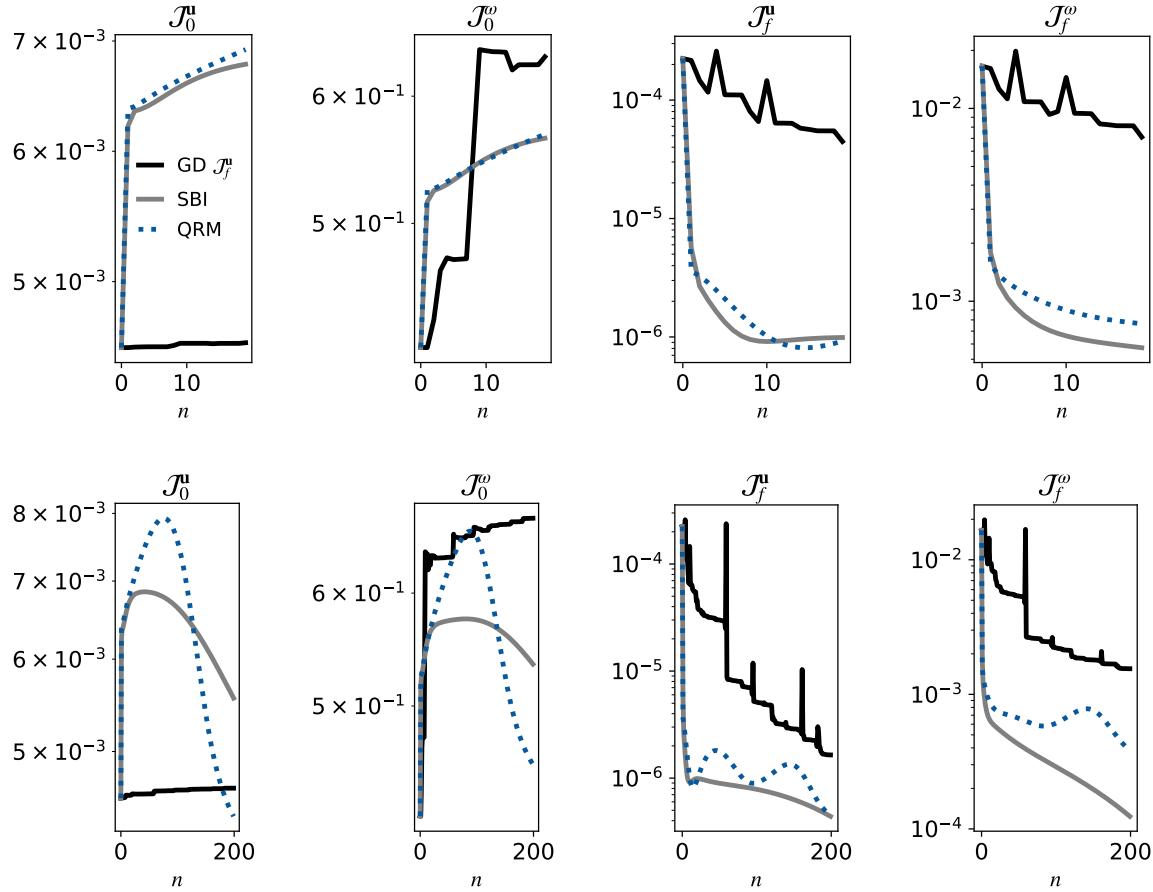


Figure 12. Same as Figure 8 using initial guess $\mathbf{u}^0(\mathbf{x}, 0)$ generated via SBI. Our use of SBI mimics [5, 4], who used this method to rewind their target state from $t : 0 \rightarrow t_f$. This is congruent to taking a single SBI iteration with step size $\gamma = 1$ after starting with the previous initial guess $\mathbf{u}(\mathbf{x}, 0) = 0$. They subsequently refined their SBI initial guess using DAL to minimize a cost functional (analogous to DAL \mathcal{J}_f^u , shown in black). Using our Navier–Stokes inverse problem, we compare each method (DAL, SBI, and QRM) in a highly-controlled setting. First, we confirm that using SBI followed by DAL does indeed improve performance dramatically (in this limited case). Next we compare direct minimization (DAL) of our two cost functionals \mathcal{J}^u and \mathcal{J}^w . Their own respective descents behave as expected. Finally we implement SBI and QRM as before. Just as with the previous initial guess $\mathbf{u}^0(\mathbf{x}, 0) = 0$, we find that SBI and QRM outperform both DAL runs, albeit by a narrower margin. Crucially, SBI and both QRM runs outperform DAL with very few iterations, suggesting that the efficacy of these methods is somewhat robust to the initial guess.

6.3. Derivation of Ill-Posed Nonlinear Adjoint for KdVB

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

$$0 = \partial_t u' - a\partial_x^2 u' + b\partial_x^3 u' + \bar{u}\partial_x u' + u'\partial_x \bar{u} + u'\partial_x u' \quad (21)$$

$$= \partial_t u' - a\partial_x^2 u' + b\partial_x^3 u' + u\partial_x u' + u'\partial_x u - u'\partial_x u' \quad (22)$$

which is ill-posed over $t : t_f \rightarrow 0$. Using the optimal control perspective, we demonstrate that Equation 22 is equivalent to the adjoint which minimizes \mathcal{J}_0^u .

First we multiply 22 by u'

$$0 = u'\partial_t u' - au'\partial_x^2 u' + bu'\partial_x^3 u' + u'^2\partial_x \bar{u} + uu'\partial_x u' - u'^2\partial_x u', \quad (23)$$

and integrate over the space-time domain, yielding \mathcal{J}_0^u in terms of u' , u , and \mathcal{J}_f^u .

$$\mathcal{J}_0^u = \mathcal{J}_f^u - \int_0^{t_f} \langle au'\partial_x^2 u' - u'\partial_x[uu'] \rangle dt. \quad (24)$$

24 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 24 we construct a new Lagrangian

$$\bar{\mathcal{L}} = \int_0^{t_f} \langle \bar{\mu}(x, t) \cdot \mathcal{F}[u(x, t)] \rangle dt + \mathcal{J}_0^u \quad (25)$$

$$= \int_0^{t_f} \langle \bar{\mu}[\partial_t u + u\partial_x u - a\partial_x^2 u + b\partial_x^3 u] - au'\partial_x^2 u' + u'\partial_x[uu'] \rangle dt + \mathcal{J}_f^u. \quad (26)$$

Taking the variation wrt $\bar{\mu}(x, t)$ still returns the constraint equation 4 (on $0 < t < t_f$) and 7 (at $t = t_f$). However, varying wrt $u(x, t)$ along $0 < t < t_f$ gives a new adjoint equation

$$\frac{\delta \mathcal{L}^w}{\delta u} = -\partial_t \bar{\mu} - u\partial_x \bar{\mu} - a\partial_x^2 \bar{\mu} - b\partial_x^3 \bar{\mu} - 2a\partial_x^2 u' + \partial_x[uu'] - (u + u')\partial_x u' \quad (27)$$

$$= -\partial_t \bar{\mu} - u\partial_x \bar{\mu} - a\partial_x^2 \bar{\mu} - b\partial_x^3 \bar{\mu} - 2a\partial_x^2 u' + u'\partial_x u - u'\partial_x u'. \quad (28)$$

Due to the presence of \mathcal{J}_f^u in our derivation of \mathcal{J}_0^u , the adjoint is initialized as before, i.e. $\mu(x, t_f) = \bar{\mu}(x, t_f) = -u'(x, t_f)$. Finally, by summing equations 22 and 28, it is evident that $\bar{\mu}(x, t) + u'(x, t) = 0$ not just at $t = t_f$, but throughout the time domain. We substitute $\bar{\mu} \rightarrow -u'$ in $\bar{\mathcal{L}}$'s adjoint, yielding a highly straightforward result

$$0 = \partial_t \bar{\mu} + u\partial_x \bar{\mu} + \bar{\mu}\partial_x u + \bar{\mu}\partial_x \bar{\mu} - a\partial_x^2 \bar{\mu} + b\partial_x^3 \bar{\mu}. \quad (29)$$

This new adjoint (equivalent to Equation 22) is ill-posed and effectively nonlinear. If Equation 29 could be solved exactly, we would obtain the exact deviation in our trial initial condition by performing a single adjoint loop. This is indeed the case for the non-diffusive Korteweg–de Vries equation. However when $a > 0$ (diffusive forward problem), Equation 29 cannot be solved backwards using conventional numerical solvers. This problem of ill-posed backward integration motivates the SBI/QRM modifications.

References

- [1] Lukyanenko D, Yeleskina T, Prigorniy I, Isaev T, Borzunov A and Shishlenin M 2021 *Mathematics* **9**(4) 1–12 ISSN 22277390
- [2] Subramanian S, Scheufele K, Mehl M and Biros G 2020 *Inverse Problems* **36**(4) ISSN 13616420
- [3] Kabanikhin S I, Kulikov I M and Shishlenin M A 2020 *Computational Mathematics and Mathematical Physics* **60**(6) ISSN 15556662
- [4] Liu L and Gurnis M 2008 *Journal of Geophysical Research: Solid Earth* **113**(B8)
- [5] Li D, Gurnis M and Stadler G 2017 *Geophysical Journal International* **209**(1) ISSN 1365246X
- [6] Imanuvilov O Y 1997 *Turbulence Modeling and Vortex Dynamics* 148–168 URL <https://link.springer.com/chapter/10.1007/BFb0105035>
- [7] Fan J and Nakamura G 2009 *Journal of Inverse and Ill-Posed Problems* **17**(6) 565–584 ISSN 09280219
- [8] Fan J, Cristo M D, Jiang Y and Nakamura G 2010 *Journal of Mathematical Analysis and Applications* **365**(2) 750–757 ISSN 0022-247X
- [9] Du J, Navon I M, Zhu J, Fang F and Alekseev A K 2013 *Computers & Mathematics with Applications* **65**(3) 380–394 ISSN 0898-1221
- [10] Law K J and Stuart A M 2012 *Monthly Weather Review* **140**(11) 3757–3782 ISSN 1520-0493 URL <https://journals.ametsoc.org/view/journals/mwre/140/11/mwr-d-11-00257.1.xml>
- [11] Fan T, Xu K, Pathak J and Darve E 2020 URL <https://arxiv.org/abs/2008.13074v2>
- [12] Zhao Q, Lindell D B and Wetzstein G 2022 *arxiv* URL <https://arxiv.org/abs/2206.00711v1>
- [13] Mannix P M, Skene C S, Auroux D and Marcotte F 2022 *Inverse Problems* URL <https://arxiv.org/abs/2210.17194>
- [14] Dardé J and Dardé J 2016 *Inverse Problems and Imaging* **10**(2) 379–407 ISSN 1930-8337
- [15] Conrad C P and Gurnis M 2003 *Geochemistry, Geophysics, Geosystems* **4**(3) ISSN 15252027
- [16] Fang L, Bos W J T, Liang S, Bertoglio J P and Shao L 2012 *Journal of Turbulence* **13**(3) 1–14 URL <https://www.tandfonline.com/action/journalInformation?journalCode=tjot20>
- [17] Steiner W and Reichl S 2012 *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME* **134**(2) ISSN 00220434
- [18] BARZILAI J and BORWEIN J M 1988 *IMA Journal of Numerical Analysis* **8** 141–148 ISSN 0272-4979 (*Preprint* <https://academic.oup.com/imajna/article-pdf/8/1/141/2402762/8-1-141.pdf>) URL <https://doi.org/10.1093/imajna/8.1.141>
- [19] LIU D C and NOCEDAL J 1989 *Mathematical programming* **45** 503–528 ISSN 0025-5610
- [20] Burns K J, Vasil G M, Oishi J S, Lecoanet D and Brown B P 2020 *Physical Review Research* **2**(2) 023068 ISSN 26431564
- [21] Kerswell R R, Pringle C C and Willis A P 2014 An optimization approach for analysing nonlinear stability with transition to turbulence in fluids as an exemplar
- [22] Ambrose D M and Wilkening J 2010 *Journal of Nonlinear Science* **20**(3) ISSN 09388974