

Accelerated Multi-Block ADMM for Nonlinear 4D-Variational Data Assimilation

Chris Junchi Li[◊]

Department of Electrical Engineering and Computer Sciences[◊]
University of California, Berkeley

October 7, 2024

Abstract

This paper addresses the nonlinear 4D variational data assimilation (4D-Var) problem by leveraging a novel multi-block Alternating Direction Method of Multipliers (ADMM) framework. We introduce an efficient linearized multi-block ADMM algorithm with regularization, which significantly reduces the computational complexity associated with solving large-scale nonlinear 4D-Var problems. The proposed approach allows for parallelization, making it suitable for long-term data assimilation in applications like meteorology and oceanography. Numerical results demonstrate the superiority of the method in terms of convergence rates and accuracy, compared to traditional first-order methods. Key theoretical contributions are presented regarding the convergence guarantees of the ADMM-based algorithm in nonlinear settings, and its performance is validated through applications to the viscous Burgers' equation.

Keywords: 4D-Var, multi-block ADMM, regularization, nonlinear data assimilation, viscous Burgers' equation.

1 Introduction

Data assimilation plays a pivotal role in integrating model predictions with observational data to provide more accurate estimates of the state of a dynamical system. Among the many methods for data assimilation, the four-dimensional variational assimilation (4D-Var) method has gained significant attention, especially in fields like meteorology, oceanography, and climatology. 4D-Var aims to optimize the initial conditions of a model by minimizing a cost function that measures the misfit between model outputs and observations over a time window.

Solving 4D-Var in the context of nonlinear dynamics presents a number of challenges, particularly in terms of computational complexity and convergence. Classical approaches based on gradient descent methods often fail to efficiently handle large-scale, nonlinear 4D-Var problems. To address these limitations, we propose an advanced multi-block ADMM framework, which extends the classical ADMM approach by introducing regularization techniques and exploiting the inherent structure of 4D-Var problems. The resulting algorithm is not only computationally efficient but also highly parallelizable, making it practical for large-scale applications.

Backgrounds. By leveraging both sources, it produces more accurate state estimates than relying solely on either observations or model predictions [ABN16]. Unlike approaches such as machine learning, image analysis, or statistical methods, data assimilation is fundamentally anchored in the dynamics of the system under study. As a result, it has emerged as a prominent research area in mathematics [LSZ15, SAST23]. A key application of data assimilation lies in determining initial conditions, where distributed observations collected over time are combined with the dynamic model, which significantly enhances forecast accuracy [Kal02]. This methodology has found

widespread applications across various scientific fields, including meteorology [BC02], oceanography [MW82, Wun96], and climatology [SGGC13, WH13, SBH⁺16].

The application of the calculus of variations to data assimilation was first pioneered by [Sas58] for meteorological analysis, with subsequent extensions that incorporated the time dimension [Sas69, Sas70, Tho69]. These advancements laid the groundwork for what is now known as *4D Variational Data Assimilation*, commonly referred to as **4D-Var**. The introduction of the adjoint method by [Lio71] and [Mar75a, Mar75b] provided an efficient framework for obtaining gradient information, enabling the use of classical first-order optimization algorithms to solve these types of problems. The adoption of **4D-Var** in meteorological data assimilation gained significant momentum during the 1980s, with its theoretical foundations established by [LDT86] and [TC87].

Consider a finite-dimensional dynamical system described by:

$$\begin{cases} \mathbf{u}_t = G(\mathbf{u}) \\ \mathbf{u}|_{t=0} = \mathbf{u}_0, \end{cases} \quad (1)$$

which $\mathbf{u} = (u_1, u_2, \dots, u_m)^\top \in \mathbb{R}^m$ is an m -dimensional vector, and its Euclidean norm is defined as:

$$\|\mathbf{u}\| = \left(\sum_{i=1}^m |u_i|^2 \right)^{\frac{1}{2}}. \quad (2)$$

Let $\hat{\mathbf{u}}$ represent the (partial) observational data collected over time, with indices chosen from a subset $\mathcal{S} \subseteq \{1, 2, \dots, m\}$. Additionally, let $\hat{\mathbf{u}}_0^b$ denote the data obtained from a prior prediction. The **4D-Var** problem can be rigorously formulated in the following variational form:

$$\begin{cases} \min F(\mathbf{u}) := \frac{1}{2} \int_0^T \|\mathbf{u} - \hat{\mathbf{u}}\|^2 dt + \frac{\alpha}{2} \|\mathbf{u}_0 - \hat{\mathbf{u}}_0^b\|^2 \\ \text{s.t. } \mathbf{u} \text{ satisfies (1),} \end{cases} \quad (3)$$

where $\alpha > 0$ is a regularization coefficient. In practice, observational data are typically collected at discrete time points. For simplicity, assume that observation times are uniformly spaced with $T_o = T/n$, meaning that the data is recorded at times $t = kT_o$ for $k = 0, 1, \dots, n$. Let $\mathbf{u}_k = \mathbf{u}(kT_o)$ represent the solution at time $t = kT_o$. The dynamical system (1) can then be expressed at these observation times as:

$$\mathbf{u}_{k+1} = H_{T_o}(\mathbf{u}_k), \quad k = 0, 1, \dots, n-1, \quad (4)$$

where H_{T_o} is an operator acting over the observational time interval T_o , which may be linear or nonlinear, and \mathbf{u}_0 is the given initial condition. Consequently, the continuous **4D-Var** problem (3) is transformed into the following discrete form:

$$\begin{cases} \min F(\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_n) := \frac{T_o}{2} \sum_{k=0}^n \|\mathbf{u}_k - \hat{\mathbf{u}}_k\|^2 + \frac{\alpha}{2} \|\mathbf{u}_0 - \hat{\mathbf{u}}_0^b\|^2 \\ \text{s.t. } \mathbf{u}_{k+1} = H_{T_o}(\mathbf{u}_k), \quad \text{for } k = 0, 1, \dots, n-1. \end{cases} \quad (5)$$

1.1 Linear dynamical system

When the dynamical system (1) is linear, the update in the dynamical iteration (4) can be written as $H_{T_o}(\mathbf{u}_k) = L\mathbf{u}_k$, where L is a linear operator or matrix. This simplifies the discrete dynamical system (4) to:

$$\mathbf{u}_k = L^k \mathbf{u}_0, \quad \text{for } k = 0, 1, \dots, n, \quad (6)$$

where \mathbf{u}_0 is the given initial condition. In this case, the 4D-Var Problem (5) reduces to a function that depends solely on the initial condition \mathbf{u}_0 , rather than on all the intermediate states $\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_n$. The linear 4D-Var Problem can thus be reformulated as:

$$\min_{\mathbf{u}_0 \in \mathbb{R}^m} F(\mathbf{u}_0) = \frac{T_o}{2} \sum_{k=0}^n \|L^k \mathbf{u}_0 - \hat{\mathbf{u}}_k\|^2 + \frac{\alpha}{2} \|\mathbf{u}_0 - \hat{\mathbf{u}}_0^b\|^2. \quad (7)$$

This formulation shows that the linear 4D-Var Problem (5) is essentially a quadratic function of the initial condition \mathbf{u}_0 . To solve the linear 4D-Var problem (7), classical first-order optimization algorithms, such as the conjugate gradient method [HS52] and the limited-memory BFGS-B method [BLNZ95, ZBLN97], can be applied. The gradient required for these algorithms is computed as follows:

$$\nabla F(\mathbf{u}_0) = T_o \sum_{k=0}^n (L^k)^\top (L^k \mathbf{u}_0 - \hat{\mathbf{u}}_k) + \alpha (\mathbf{u}_0 - \hat{\mathbf{u}}_0^b), \quad (8)$$

where it can be observed that a key aspect of this gradient computation is the adjoint operator, $(L^k)^\top$. This procedure, commonly known as the adjoint method, was first introduced in [LDT86] and [TC87]. When the system's dimension is too large to be stored L^k explicitly, iterative methods are typically employed to compute the adjoint operator $(L^k)^\top$, which poses significant challenges, particularly in extending the approach to nonlinear systems.

1.2 The Lorenz system — a nonlinear dynamical system

For the nonlinear dynamical system (1), where the operator H_{T_o} in the dynamical iteration (4) is nonlinear, the 4D-Var Problem (5) can still be reformulated in terms of the initial condition \mathbf{u}_0 as follows:

$$\min_{\mathbf{u}_0 \in \mathbb{R}^m} F(\mathbf{u}_0) = \frac{T_o}{2} \sum_{k=0}^n \|H_{T_o}^k(\mathbf{u}_0) - \hat{\mathbf{u}}_k\|^2 + \frac{\alpha}{2} \|\mathbf{u}_0 - \hat{\mathbf{u}}_0^b\|^2. \quad (9)$$

We can generalize the procedure from the linear case to solve the nonlinear problem. To apply first-order optimization algorithms to solve the 4D-Var problem (9), the gradient must be computed as follows:

$$\nabla F(\mathbf{u}_0) = T_o \sum_{k=0}^n (\nabla H_{T_o}^k(\mathbf{u}_0))^\top (H_{T_o}^k(\mathbf{u}_0) - \hat{\mathbf{u}}_k) + \alpha (\mathbf{u}_0 - \hat{\mathbf{u}}_0^b). \quad (10)$$

The key difference between the linear case (8) and the nonlinear case (10) is that in the nonlinear case, the gradient involves the Jacobian of the nonlinear operator $H_{T_o}^k$, denoted as $\nabla H_{T_o}^k(\mathbf{u}_0)$. To compute this Jacobian efficiently, the adjoint method [LDT86, TC87] is extended to nonlinear systems using the tangent linear system, which for the dynamical system (6) is given by:

$$\delta \mathbf{u}_t = \nabla G(\mathbf{u}) \cdot \delta \mathbf{u}. \quad (11)$$

From this tangent linear system (11), the Jacobian can be derived as:

$$\nabla H_{T_o}^k(\mathbf{u}_0) = \exp \left(\int_0^{kT_o} \nabla G(\mathbf{u}(\mathbf{u}_0, t)) dt \right). \quad (12)$$

Consider the classical Lorenz system, a well-known example of a nonlinear, aperiodic, and three-dimensional deterministic system known for its chaotic behavior, first studied by [Lor63]. It

is described by the following system of differential equations:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(\rho - z) - y \\ \dot{z} = xy - \beta z, \end{cases} \quad (13)$$

where the parameters are set to the classical values: $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. To numerically solve the Lorenz system (13), we employ the 4th-order Runge–Kutta method with a time step size of $\delta t = 0.01$ over a total time horizon of $T = 3$. The observational interval is set to $T_o = 0.3$. Following the precise observation strategy proposed in [TC87], the numerical solution serves as the observational data. The observational data, denoted as $\hat{\mathbf{u}}_k = \mathbf{u}(kT_o)$ for $k = 0, 1, \dots, n = T/T_o = 10$, is generated using the 4th-order Runge-Kutta method for the Lorenz system (13) with the initial condition $\mathbf{u}_0 = (-0.5, 0.5, 20.5)$. In the context of the **4D-Var** problem (5), we set the regularization parameter $\alpha = 0.1$. To illustrate the nonconvex nature of the objective function (9) associated with the Lorenz system (13), we visualize slices of the objective function along the X -, Y -, and Z -directions, as shown in Figure 1. These visualizations highlight the nonconvex landscape of the objective function, which arises from the chaotic dynamics inherent to the Lorenz system (13). As the system’s nonlinearity and sensitivity to initial conditions increase, the optimization problem becomes significantly more challenging.

We then apply two classical first-order optimization algorithms, the nonlinear conjugate gradient methods [HS52, FR64, PR69, DY99] and the limited-memory BFGS-B method [BLNZ95, ZBLN97], to tackle the **4D-Var** problem (5). For the Lorenz system (13), its tangent linear system is derived as:

$$\begin{pmatrix} \delta\dot{x} \\ \delta\dot{y} \\ \delta\dot{z} \end{pmatrix} = K[x, y, z] \begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} = \begin{pmatrix} -\sigma & \sigma & 0 \\ \rho - z & -1 & -x \\ y & x & -\beta \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix}. \quad (14)$$

Utilizing the expression (38) along with the 4th-order Runge-Kutta iteration, we compute the Jacobian of the nonlinear operator $\nabla H_{T_o}^k(\mathbf{u}_0)$. Taking its transpose, $(\nabla H_{T_o}^k(\mathbf{u}_0))^\top$, allows us to obtain the gradient required for optimization. However, when these algorithms are applied to the objective function (9), both fail to converge the true solution, as demonstrated in Figure 2. This failure is attributed to the highly nonconvex nature of the objective function, which is riddled with multiple local minima. Figure 1 illustrates the landscape of the objective function, characterized by numerous valleys and peaks. While the global minimum corresponds to the true solution, the nonconvex landscape traps both algorithms in local minima, preventing them from reaching the optimal solution. This is particularly evident in Figure 2, where the final solutions produced by both methods remain far from the global minimum. The inherent nonconvexity of the problem poses significant challenges. These classical first-order optimization methods, such as the nonlinear conjugate gradient methods and the Limited memory BFGS-B method, which are typically effective for convex problems, struggle to escape local minima. Furthermore, the nonconvex nature of the **4D-Var** problem (9) makes the solutions obtained by these classical first-order optimization methods highly sensitive to initial conditions. This sensitivity, as noted in [Tal14], is a well-known open problem that requires further in-depth investigation.

1.3 4D-Var for PDEs

In this section, we provide a brief outline of the **4D-Var** problem for partial differential equations (PDEs) and its numerical implementation. Let $\Omega \subseteq \mathbb{R}^d$ be an open set with a smooth boundary

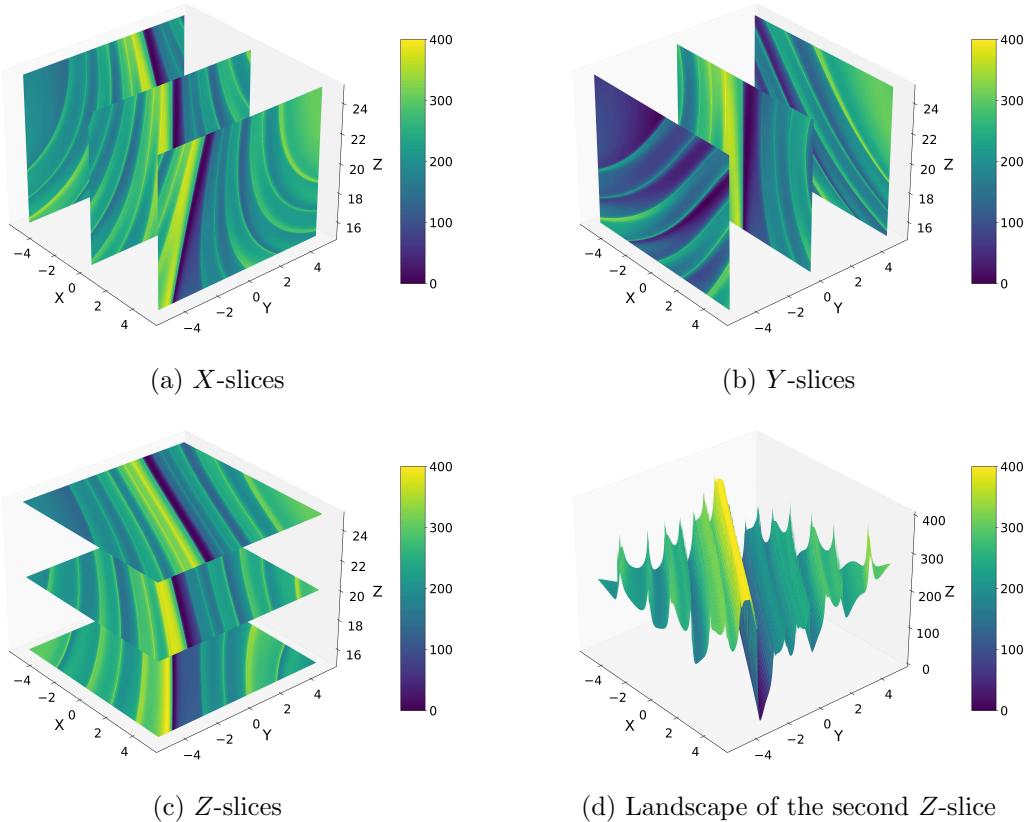


Figure 1. The objective function in (9) is evaluated at $\mathbf{u}_0 = (x_0, y_0, z_0) \in [-6, 6] \times [-6, 6] \times [14, 26]$ for the Lorenz system described in (13). In subfigures (a), (b), and (c), the colors correspond to the function values, with darker shades representing lower values and lighter shades indicating higher values.

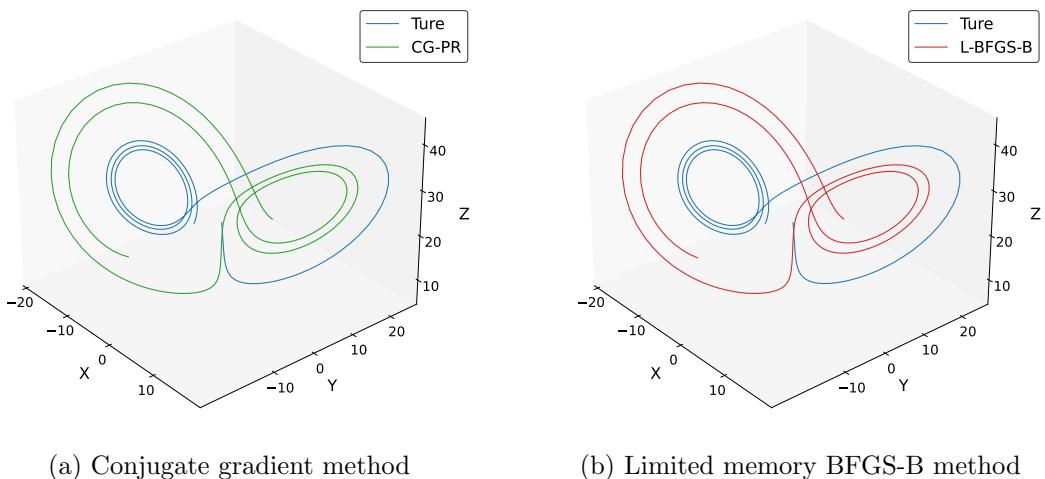


Figure 2. The optimization algorithms are applied to the 4D-Var problem (9) for the Lorenz system (13), starting with the initial condition $\mathbf{u}_0 = (x_0, y_0, z_0) = (-3, -3, 10)$. The numerical implementation is carried out using the Python subpackage — SciPy 1.14.1.

$\partial\Omega$. The general form of a PDE governing the dynamics of a system can be written as:

$$\begin{cases} u_t = G(u) \\ u|_{t=0} = u_0 \\ u|_{\partial\Omega} = g, \end{cases} \quad (15)$$

where $u = u(t, x)$ is a function of time $t \in (0, +\infty)$ and the spatial variable $x \in \mathbb{R}^d$, and its Energy norm, or L^2 -norm, is defined as:

$$\|u(t, \cdot)\|^2 = \int_{\Omega} |u(t, x)|^2 dx. \quad (16)$$

The numerical implementation of PDEs given by (15) often involves discretization techniques that transform the continuous system into a more tractable, finite-dimensional one. The most widely used methods include the finite difference method, finite element method, and spectral methods. While these methods differ in terms of theoretical error bounds and convergence rates, they share the goal of approximating the continuous PDE by converting it into a system of ordinary differential equations (ODEs). The solution to this discretized system, at the observational time points, can then be iteratively expressed through a discrete scheme in alignment with the observational data as outlined in (4).

For the **4D-Var** problem, which involves data assimilation over a specific time window, the numerical implementation focuses on minimizing a cost function that quantifies the misfit between model predictions and observations, as outlined in (5). This minimization problem can often be reformulated as a quadratic function of the initial condition u_0 , as shown in (7), allowing for efficient solution of the linearized **4D-Var** problem. Classical first-order optimization algorithms, such as the conjugate gradient method [HS52] and the limited-memory BFGS-B method [BLNZ95, ZBLN97], are well-suited for this task. These optimization techniques have been successfully applied to a wide range of linear PDEs, including the heat equation [BO18, LHGD24] and the wave equation [BFO20].

2 Solving the **4D-Var** problem via ADMM

In this section, we introduce a practical variant of the alternating direction method of multipliers (ADMM), known as multi-block ADMM, to solve the **4D-Var** problem (5). The use of multi-block ADMM arises naturally from reformulating the **4D-Var** problem (5) as a constraint optimization problem, which facilitates the derivation of its augmented Lagrangian. By efficiently implementing the linearized version of multi-block ADMM with regularization, we demonstrate strong numerical performance. Finally, we highlight several advantages of this approach over classical first-order unconstrained optimization algorithms, as discussed in Section 1.

2.1 Motivation and basic idea for implementing ADMM

To facilitate the integration of observational data with the system's dynamics, it is necessary to expand the number of variables in the objective function without alternating its values. This expansion enables a more accurate representation of the system's state over time. Given a total number of iterations $N = T/\delta t$ and $n = T_o/\delta t$ iterations per observational time interval, we increase the number of variables from n to N . This expansion allows for closer tracking of the state of the

system over each discrete time step. For any $k = 0, 1, \dots, N$, the sub-objective function $f_k(\mathbf{u}_k)$ is defined as follows:

$$f_k(\mathbf{u}_k) = \begin{cases} \frac{T_o}{2} \|\mathbf{u}_0 - \hat{\mathbf{u}}_0\|^2 + \frac{\alpha}{2} \|\mathbf{u}_0 - \hat{\mathbf{u}}_0^b\|^2, & \text{for } k = 0, \\ \frac{T_o}{2} \|\mathbf{u}_{k/n} - \hat{\mathbf{u}}_{k/n}\|^2, & \text{for } k/n \in \mathbb{N}^+, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

The overall 4D-Var problem (5) can then be reformulated as:

$$\begin{cases} \min F(\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_N) := \sum_{k=0}^N f_k(\mathbf{u}_k), \\ \text{s.t. } \mathbf{u}_{k+1} = H(\mathbf{u}_k), \quad \text{for } k = 0, 1, \dots, N-1, \end{cases} \quad (18)$$

where H represents the nonlinear operator that governs the evolution of the system within some numerical scheme at each time step δt .

Augmented Lagrangian Formulation Based on the constraint optimization form of the 4D-Var problem (18), we can then derive its augmented Lagrangian as follows:

$$\begin{aligned} L &= L(\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_N; \boldsymbol{\lambda}_0, \dots, \boldsymbol{\lambda}_{N-1}; s) \\ &= \sum_{k=0}^N f_k(\mathbf{u}_k) - \sum_{k=0}^{N-1} \langle \boldsymbol{\lambda}_k, \mathbf{u}_{k+1} - H(\mathbf{u}_k) \rangle + \frac{1}{2s} \sum_{k=0}^{N-1} \|\mathbf{u}_{k+1} - H(\mathbf{u}_k)\|^2, \end{aligned} \quad (19)$$

where $s > 0$ is a given parameter. By completing the square on the penalty terms, this formulation (19) can be simplified further:

$$L = \sum_{k=0}^N f_k(\mathbf{u}_k) + \frac{1}{2s} \sum_{k=0}^{N-1} \|\mathbf{u}_{k+1} - H(\mathbf{u}_k) - s\boldsymbol{\lambda}_k\|^2 - \frac{s}{2} \sum_{k=0}^{N-1} \|\boldsymbol{\lambda}_k\|^2. \quad (20)$$

From the simplified expression (20), we observe that \mathbf{u}_k is only related to its neighboring time steps, \mathbf{u}_{k-1} and \mathbf{u}_{k+1} , since the sub-objective function f_k depends only on \mathbf{u}_k . This local dependence simplifies the optimization process by focusing on interactions with adjacent time steps in the iterative solution.

Multi-block ADMM To demonstrate the iterative process of multi-block ADMM, we apply the arg min operation to the augmented Lagrangian during the following iteration:

$$\begin{cases} \mathbf{u}_k^{\ell+1} = \arg \min_{\mathbf{u}_k} L(\mathbf{u}_0^\ell, \dots, \mathbf{u}_{k-1}^\ell, \mathbf{u}_k, \mathbf{u}_{k+1}^\ell, \dots, \mathbf{u}_N^\ell; \boldsymbol{\lambda}_0^\ell, \dots, \boldsymbol{\lambda}_{N-1}^\ell; s), & \text{for } k = 0, \dots, N, \\ s\boldsymbol{\lambda}_k^{\ell+1} = s\boldsymbol{\lambda}_k^\ell - (\mathbf{u}_{k+1}^{\ell+1} - H(\mathbf{u}_k^{\ell+1})), & \text{for } k = 0, \dots, N-1, \end{cases} \quad (21)$$

where the second equation represents the update for the dual variables (Lagrange multipliers). This procedure alternates between updating the primal variables \mathbf{u}_k and the dual variables $\boldsymbol{\lambda}_k$.

Substituting the simplified expression (20) into the arg min iteration (21), we can derive that the multi-block ADMM iterates as follows:

$$\begin{cases} \mathbf{u}_0^{\ell+1} = \arg \min_{\mathbf{u}_0} \left\{ f_0(\mathbf{u}_0) + \frac{1}{2s} \left\| \mathbf{u}_1^\ell - H(\mathbf{u}_0) - s\boldsymbol{\lambda}_0^\ell \right\|^2 \right\}, \\ \mathbf{u}_k^{\ell+1} = \arg \min_{\mathbf{u}_k} \left\{ f_k(\mathbf{u}_k) + \frac{1}{2s} \left\| \mathbf{u}_k - H(\mathbf{u}_{k-1}^\ell) - s\boldsymbol{\lambda}_{k-1}^\ell \right\|^2 + \frac{1}{2s} \left\| \mathbf{u}_{k+1}^\ell - H(\mathbf{u}_k) - s\boldsymbol{\lambda}_k^\ell \right\|^2 \right\}, \\ \quad \text{for } k = 1, 2, \dots, N-1, \\ \mathbf{u}_N^{\ell+1} = \arg \min_{\mathbf{u}_N} \left\{ f_N(\mathbf{u}_N) + \frac{1}{2s} \left\| \mathbf{u}_N - H(\mathbf{u}_{N-1}^\ell) - s\boldsymbol{\lambda}_{N-1}^\ell \right\|^2 \right\}, \\ s\boldsymbol{\lambda}_k^{\ell+1} = s\boldsymbol{\lambda}_k^\ell - (\mathbf{u}_{k+1}^{\ell+1} - H(\mathbf{u}_k^{\ell+1})), \quad \text{for } k = 0, 1, \dots, N-1. \end{cases} \quad (22)$$

In this formulation, it is observed that each block \mathbf{u}_k is updated independently, which is the key characteristic of the multi-block ADMM algorithm. While the classical two-block ADMM was originally proposed in [GM75, GM76], the multi-block ADMM was first studied by [HHY15] in the context of linear constraints. It is important to note that, even in the two-block case, the algorithm described in (22) may exhibit divergence, as discussed in [HHY15]. The potential for divergence is a known issue in multi-block ADMM, particularly when the assumptions regarding the problem structure or linear constraints are not satisfied.

2.2 Efficient ADMM implementation

As demonstrated in [HHY15], the introduction of a regularization term is a common approach to guarantee the convergence of the multi-block ADMM (22). Originally proposed by [ZBBO10] for the two-block case, this regularization technique is crucial for stabilizing the iterative updates and mitigating potential divergence, particularly when the underlying problem structure does not inherently guarantee convergence. Additionally, each subproblem for $k = 0, 1, \dots, N-1$ involves solving a nonlinear least square problem, $\|\mathbf{u}_{k+1}^\ell - H(\mathbf{u}_k) - s\boldsymbol{\lambda}_k^\ell\|^2$, which incurs significant computational overhead.

Linearized multi-block ADMM with regularization To reduce computational costs and enhance efficiency, a linearization technique introduced by [DY16] is employed. This technique, combined with regularization, allows for a modified version of the multi-block ADMM, expressed as

follows:

$$\left\{ \begin{array}{l} \mathbf{u}_0^{\ell+1} = \arg \min_{\mathbf{u}_0} \left\{ f_0(\mathbf{u}_0) - \frac{1}{s} \left\langle \nabla H(\mathbf{u}_0^\ell)^\top (\mathbf{u}_1^\ell - H(\mathbf{u}_0^\ell) - s\boldsymbol{\lambda}_0^\ell), \mathbf{u}_0 \right\rangle + \frac{\|\mathbf{u}_0 - \mathbf{u}_0^\ell\|^2}{2\eta} \right\}, \\ \mathbf{u}_k^{\ell+1} = \arg \min_{\mathbf{u}_k} \left\{ f_k(\mathbf{u}_k) + \frac{1}{2s} \left\| \mathbf{u}_k - H(\mathbf{u}_{k-1}^\ell) - s\boldsymbol{\lambda}_{k-1}^\ell \right\|^2 \right. \\ \quad \left. - \frac{1}{s} \left\langle \nabla H(\mathbf{u}_k^\ell)^\top (\mathbf{u}_{k+1}^\ell - H(\mathbf{u}_k^\ell) - s\boldsymbol{\lambda}_k^\ell), \mathbf{u}_k \right\rangle + \frac{\|\mathbf{u}_k - \mathbf{u}_k^\ell\|^2}{2\eta} \right\}, \\ \quad \text{for } k = 1, 2, \dots, N-1, \\ \mathbf{u}_N^{\ell+1} = \arg \min_{\mathbf{u}_N} \left\{ f_N(\mathbf{u}_N) + \frac{1}{2s} \left\| \mathbf{u}_N - H(\mathbf{u}_{N-1}^\ell) - s\boldsymbol{\lambda}_{N-1}^\ell \right\|^2 + \frac{\|\mathbf{u}_N - \mathbf{u}_N^\ell\|^2}{2\eta} \right\}, \\ s\boldsymbol{\lambda}_k^{\ell+1} = s\boldsymbol{\lambda}_k^\ell - (\mathbf{u}_{k+1}^{\ell+1} - H(\mathbf{u}_k^{\ell+1})), \quad \text{for } k = 0, \dots, n-1, \end{array} \right. \quad (23)$$

where $\eta > 0$ is a regularization parameter that helps control the deviation between successive iterations. In this formulation (23), the linearization involves computing the gradient, which can still be efficiently obtained using the adjoint method [LDT86, TC87]. The regularization term ensures that successive iterations do not deviate too far from one another, stabilizing the updates and preventing large, destabilizing steps. Thus, the linearization serves as an effective approximation for the nonlinear least square problem. Recently, several studies, including [XW21, EBNP23], and [HP24], have explored the convergence properties of the linearized multi-block ADMM with regularization (23). The convergence analysis of this method has emerged as a compelling area of research.

2.3 Numerical Performance

To demonstrate the high numerical efficiency of the linearized multi-block ADMM with regularization (23), we use the Lorenz system (13) as a benchmark for comparison with the classical first-order optimization algorithms, as outlined in Section 1. The numerical solution is obtained using the 4th-order Runge-Kutta method with a time step size of $\delta t = 0.01$ over a total time horizon of $T = 3$. The observational interval is set to $T_o = 0.3$, leading to a total number of $N = T/\delta t = 300$ iterations, with $n = T/T_o + 1 = 11$ observational time points. We present the numerical performance of the linearized multi-block ADMM with regularization (23) applied to the 4D-Var problem (5) under both precise and noisy observation conditions, highlighting its effectiveness across varying scenarios.

Precise observation The precise observational data, $(\hat{x}_1(k), \hat{y}_1(k), \hat{z}_1(k))$ for $k = 0, 1, \dots, n = 10$, is generated using the numerical solution from the 4th-order Runge-Kutta method, recorded at kT_o . To solve the 4D-Var problem (5), we apply the linearized multi-block ADMM with regularization (23), with the parameters set as $\mu = 100$, $\eta = 0.1$, and $s = 2/3$. As outlined in Section 1, the classical first-order optimization algorithms begin with the initial condition $\mathbf{u}_0 = (x_0, y_0, z_0) = (-3, -3, 10)$. To ensure consistency in comparison, the linearized multi-block ADMM with regularization (23) also starts with the numerical solution recorded at kT_o for $k = 0, 1, \dots, n = 10$, which is obtained using the 4th-order Runge-Kutta method under the same initial condition. The numerical performance is illustrated in Figure 3. Unlike the classical first-

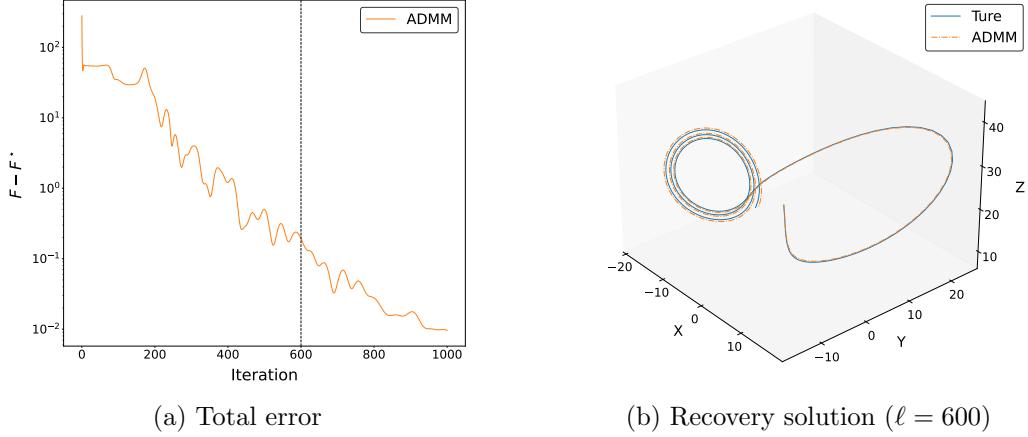


Figure 3. Numerical performance of the linearized multi-block ADMM with regularization (23), which is applied to the 4D-Var problem (9) for the Lorenz system (13) under the same settings as depicted in Figure 2.

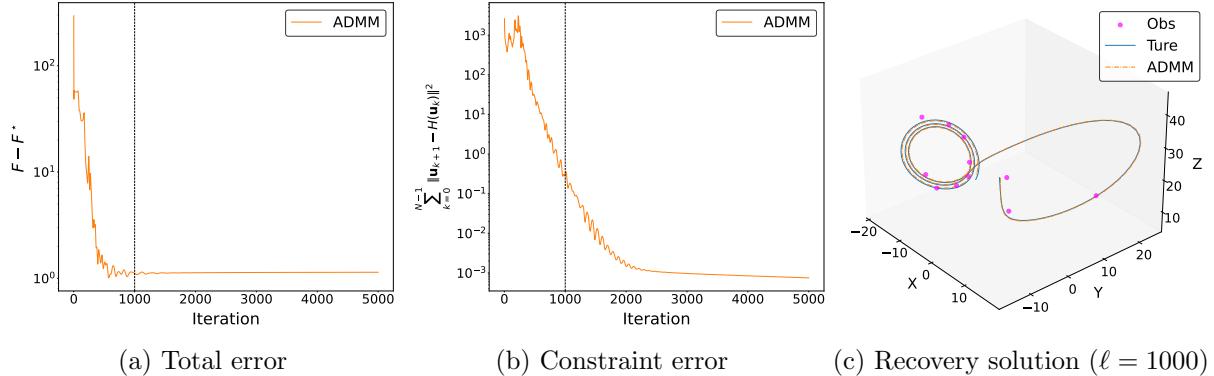


Figure 4. Numerical performance of the linearized multi-block ADMM with regularization (23), which is applied to the 4D-Var problem (18) for the Lorenz system (13) under noisy observation conditions, as specified in (24).

order optimization algorithms, such as the nonlinear conjugate gradient method and the Limited memory BFGS-B method, which are prone to get trapped in local minima (see Figure 2), the linearized multi-block ADMM with regularization (23) exhibits robust convergence. It consistently reaches the true solution, as evidenced by the numerical performance shown in Figure 3.

Noisy observation The noisy observational data is generated by adding Gaussian noise to the precise observational data. Specifically, the noisy data is given by:

$$(\hat{x}_2(k), \hat{y}_2(k), \hat{z}_2(k)) = (\hat{x}_1(k) + \varepsilon_1, \hat{y}_1(k) + \varepsilon_2, \hat{z}_1(k) + \varepsilon_3), \quad (24)$$

where $\varepsilon_i \sim \mathcal{N}(0, 1)$ for any $i = 1, 2, 3$. Under the same settings as in the precise observation case, we apply the linearized multi-block ADMM with regularization (23) to solve the 4D-Var problem (5). The resulting numerical performance is depicted in Figure 4. By comparing Figure 3a (precise observation) and Figure 4a (noisy observation), we observe a key difference: in the precise observation scenario, the total error converges consistently to zero, while in the noisy observation scenario, the total error converges to a non-zero value due to the influence of the noise.

However, despite this noise-induced offset, the convergence remains stable, indicating robustness in the optimization process. Moreover, as illustrated in Figure 4b, we continue to monitor the constraint, $\sum_{k=0}^{N-1} \|\mathbf{u}_{k+1} - H(\mathbf{u}_k)\|^2$, which ensures that the solution recovered by the linearized multi-block ADMM with regularization (23) remains accurate. Finally, Figure 4c shows that, despite the presence of noise, the recovered solution remains close to the true numerical solution, demonstrating the method's resilience in handling noisy data while maintaining a high degree of accuracy.

2.4 Advantages of the linearized multi-block ADMM with regularization

The algorithmic structure of the linearized multi-block ADMM with regularization (23), along with its numerical performance as demonstrated in Figure 3 and Figure 4, highlights several significant advantages for solving the 4D-Var problem (5). These advantages are outlined as follows:

- (1) **Effective Utilization of Observational Data** In solving the 4D-Var problem with classical first-order optimization algorithms, the solution is assumed to strictly follow the governing dynamics, making it highly sensitive to the selection of initial conditions. This sensitivity, combined with noise in observational data, can significantly degrade performance. However, when using the linearized multi-block ADMM with regularization (23), the recovered solution captures the whole dynamics, not just the initial condition. At the outset, the iterative points do not need to strictly satisfy the constraints, or is a specific dynamical solution; they only need to converge toward the constraints. Particularly in the early stages, if the initial condition is far from the observational data, the arg min operation effectively pulls the solution closer to the data without being constrained to a specific dynamical solution. Additionally, a scaling parameter $\mu > 0$ can be introduced in the objective function to balance the solution between the observational data and constraints, improving overall performance.
- (2) **Quadratic Subproblems** In the linearized multi-block ADMM with regularization (23), each subproblem is framed as a quadratic function, simplifying the optimization process. By adjusting the regularization coefficient $\eta > 0$, the difference between consecutive iterations $\mathbf{u}^{\ell+1}$ and \mathbf{u}^ℓ becomes smaller, making the linearization a reasonable approximation. The flexibility in tuning the parameters $\mu > 0$ and $\eta > 0$ allows us to improve the optimization performance. However, as the time step size δt increases, the nonlinearity in the updates becomes more pronounced, which can degrade its performance.
- (3) **Parallelizable Implementation** The linearized multi-block ADMM with regularization (23) is well-suited for parallel implementation. This feature is especially advantageous for problems involving long-term nonlinear evolutions, where computational demands can be substantial. With sufficient computing resources, the method can be implemented effectively, making it scalable and practical for real-world applications.

Overall, these advantages highlight the efficiency and practical utility of the linearized multi-block ADMM with regularization (23) in addressing the complex 4D-Var problem (5).

3 Viscous Burgers' equation

In this section, we apply the 4D-Var problem (5) to a classical nonlinear PDE — the viscous Burgers' equation. The solution is assumed to satisfy $u \in L^2([0, T); H_0^1([0, \pi]))$ and $u_t \in L^2([0, T); H^{-1}([0, \pi]))$.

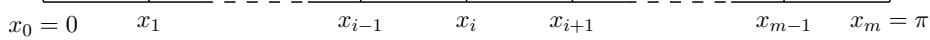


Figure 5: Uniform spatial discretization of the interval $[0, \pi]$.

Under the Dirichlet boundary condition, the viscous Burgers' equation is given by:

$$\begin{cases} \partial_t u + u \cdot \partial_x u = \gamma \partial_{xx} u \\ u(0, x) = \sin x \\ u(t, 0) = u(t, \pi) = 0, \end{cases} \quad (25)$$

where the viscous coefficient is $\gamma = 0.05$. We demonstrate the numerical performance of the linearized multi-block ADMM with regularization (23), based on the 4D-Var reformulation (18), across various numerical discretization methods, including finite difference, (Galerkin) finite element, and spectral methods. For time discretization, we employ the forward Euler scheme.

3.1 Finite difference method

Let the time step size be $\delta t = 0.02$ and the total evolution time $T = 2$. The spatial grid size is set to $m = 100$, resulting in a spatial discretization of $\delta x = \pi/m$. The spatial grid points x_i , for $i = 0, \dots, m$, are illustrated in Figure 5. For spatial discretization, we apply the central difference scheme. Given the Dirichlet boundary condition, $u(t, 0) = u(t, \pi) = 0$, we focus on the interior spatial grid points x_i for $i = 1, 2, \dots, m - 1$.

For $k = 0, 1, \dots, N = T/\delta t = 100$, let $u_{k,i} = u_{k,i}^{(m)}$ represent the finite difference approximation of the analytic solution $u(k\delta t, i\delta x)$. For the viscous Burgers' equation (25), the central difference scheme is given by:

$$\frac{u_{k+1,i} - u_{k,i}}{\delta t} + \frac{(u_{k,i+1})^2 - (u_{k,i-1})^2}{4\delta x} = \gamma \cdot \frac{u_{k,i+1} + u_{k,i-1} - 2u_{k,i}}{(\delta x)^2}. \quad (26)$$

This scheme (26) can be reformulated into the iterative update form $\mathbf{u}_{k+1} = H(\mathbf{u}_k)$, as follows:

$$\begin{aligned} u_{k+1,i} &= \left[\frac{\gamma\delta t}{(\delta x)^2} \cdot u_{k,i-1} + \frac{\delta t}{4\delta x} \cdot (u_{k,i-1})^2 \right] \\ &\quad + \left(1 - \frac{2\gamma\delta t}{(\delta x)^2} \right) u_{k,i} + \left[\frac{\gamma\delta t}{(\delta x)^2} \cdot u_{k,i+1} - \frac{\delta t}{4\delta x} \cdot (u_{k,i+1})^2 \right], \end{aligned} \quad (27)$$

for $i = 1, 2, \dots, m - 1$. Let the observational time be $T_o = 0.2$, resulting in $n + 1$ observational time points, where $n = T/T_o = 10$. The numerical solution is generated using the central difference scheme (26), with the initial condition $u_{0,i} = \sin(i\pi/m)$ for $i = 1, 2, \dots, m - 1$. The precise observational data is recorded at intervals of T_o , corresponding to the numerical solution \mathbf{u}_{10k} for $i = 0, 1, \dots, n$, where each component is $u_{10k,i}$ for $i = 1, 2, \dots, m - 1$. Gaussian noise is then added to the precise observations to produce noisy observational data $\hat{\mathbf{u}}_k$, given by:

$$\hat{u}_{k,i} = u_{10k,i} + 0.1\varepsilon_i \quad (28)$$

where $\varepsilon_i \sim \mathcal{N}(0, 1)$ for $i = 1, 2, \dots, m - 1$.

The tangent linear iteration corresponding to the iterative update (27) is given by:

$$\delta u_{k+1,i} = \left[\frac{\gamma\delta t}{(\delta x)^2} + \frac{\delta t}{2\delta x} \cdot u_{k,i-1} \right] \delta u_{k,i-1}$$

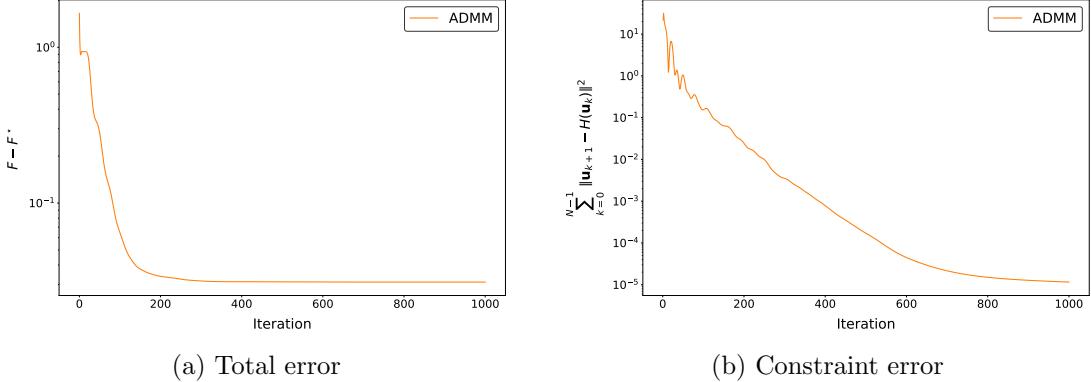


Figure 6. The linearized multi-block ADMM with regularization (23) is applied to the 4D-Var problem (18) for the viscous Burgers' equation (25), using the finite difference method (26).

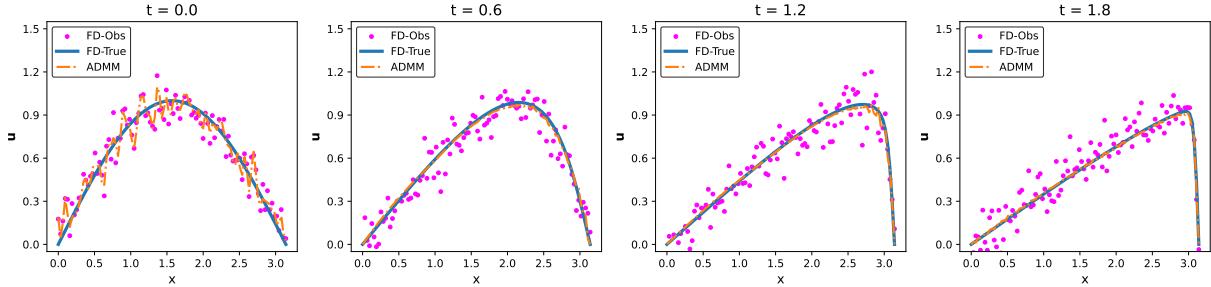


Figure 7. Comparison of noisy observational data with the time evolution dynamics recovered via the linearized multi-block ADMM with regularization (23), with the true numerical solution as the reference.

$$+ \left(1 - \frac{2\gamma\delta t}{(\delta x)^2}\right) \delta u_{k,i} + \left[\frac{\gamma\delta t}{(\delta x)^2} - \frac{\delta t}{2\delta x} \cdot u_{k,i+1} \right] \delta u_{k,i+1}, \quad (29)$$

for $i = 1, 2, \dots, m-1$, which allows us to compute the Jacobian matrix, $\nabla H(\mathbf{u}_k)$. Since $m = 99$ is not large, its adjoint operator, $\nabla H(\mathbf{u}_k)^\top$, is easily obtained. The linearized multi-block ADMM with regularization (23) is then applied to solve the 4D-Var problem (18). The initial condition is set as $\mathbf{u}_k^0 = (0, 0, \dots, 0)^\top$ for $k = 0, 1, \dots, N$, with parameters $\mu = 20$, $\eta = 0.1$, and $s = 2/3$.

Figure 6 illustrates the numerical performance, highlighting the convergence behavior of both the total error and the constraint error for the linearized multi-block ADMM with regularization (23), applied to the 4D-Var problem (18) for the viscous Burgers' equation using the finite difference method (26). The total error stabilizes while the constraint error consistently converges, exhibiting a similar pattern to that observed in the Lorenz system (Figure 4). Additionally, Figure 7 compares the dynamical evolution recovered via the linearized multi-block ADMM with regularization (23) with the noisy observational data, using the true numerical solution as a reference. Despite the presence of noise, the recovered dynamics closely match the true numerical solution, with accuracy improving over time.

3.2 (Galerkin) Finite element method

In the finite element method, the time step size is set to $\delta t = 0.01$ with a total evolution time of $T = 2$. The spatial grid points are identical to those used in the finite difference method, as shown in Figure 5. Since the solution $u = u(t, x)$ satisfies the Dirichlet boundary condition

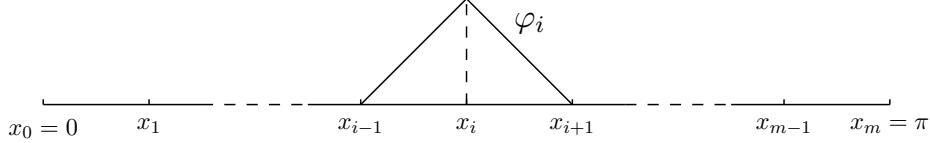


Figure 8. Uniform spatial discretization of the interval $[0, \pi]$ and Lagrange nodal basis functions φ_i for $i = 1, 2, \dots, m - 1$.

$u(t, 0) = u(t, \pi) = 0$, the Lagrange nodal basis functions are defined as:

$$\varphi_i = \begin{cases} \frac{x - x_{i-1}}{\delta x}, & x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{\delta x}, & x \in [x_i, x_{i+1}] \\ 0, & \text{otherwise} \end{cases} \quad (30)$$

for $i = 1, 2, \dots, m - 1$, as depicted in Figure 8.

For $k = 0, 1, \dots, N = T/\delta t$ and $i = 1, 2, \dots, m - 1$, the numerical solution can be expressed as the following linear combination:

$$u_k(x) = u_k^{(m)}(x) = \sum_{i=1}^{m-1} u_{k,i} \varphi_i(x),$$

which represents the Galerkin approximation for the analytic solution $u(k\delta t, x)$. The vector of coefficients, $\mathbf{u}_k = (u_{k,1}, u_{k,2}, \dots, u_{k,m-1})^\top$, satisfies the following iterative scheme:

$$\frac{\mathbf{u}_{k+1} - \mathbf{u}_k}{\delta t} + R^{-1} S_1[\mathbf{u}_k] \mathbf{u}_k = -\gamma R^{-1} T \mathbf{u}_k, \quad (31)$$

where the matrices R , $S_1[\mathbf{u}_k]$, and T are given as follows:

$$R = \delta x \begin{pmatrix} \frac{2}{3} & \frac{1}{6} & & \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{1}{6} \\ & & & \frac{1}{6} & \frac{2}{3} \end{pmatrix}, \quad T = \frac{1}{\delta x} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix},$$

and

$$S_1[\mathbf{u}_k] = \begin{pmatrix} \frac{u_k^2 - u_k^0}{3} & \frac{u_k^2 - u_k^1}{6} & & & \\ \frac{u_k^2 - u_k^1}{6} & \frac{u_k^3 - u_k^1}{3} & \frac{u_k^3 - u_k^2}{6} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{u_k^{m-1} - u_k^{m-2}}{6} \\ & & & \frac{u_k^m - u_k^{m-1}}{6} & \frac{u_k^m - u_k^{m-2}}{3} \end{pmatrix}.$$

The observational setup follows the same configuration described in Section 3.1, with the observational time $T_o = 0.2$ and $n + 1$ observational time points, where $n = T/T_o = 10$. The numerical solution is generated using the finite element scheme (31) with the initial condition $\mathbf{u}_0 = R^{-1} T \sin[x]$,

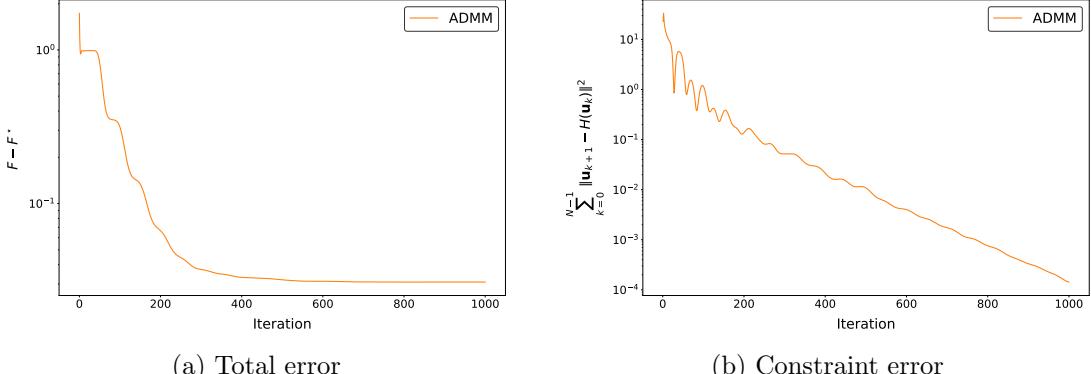


Figure 9. The linearized multi-block ADMM with regularization (23) is applied to the 4D-Var problem (18) for the viscous Burgers' equation (25)y, using the finite element method (31)

where $\sin[x] = (\sin x_1, \sin x_2, \dots, \sin x_{m-1})^\top$. At each observational time point, the numerical solution \mathbf{u}_{10k} is recorded for $i = 0, 1, \dots, n$ with each component given by $u_{10k,i}$ for $i = 1, 2, \dots, m-1$. Noisy observational data $\hat{\mathbf{u}}_k$ is generated by adding Gaussian noise to the observations:

$$\hat{u}_{k,i} = u_{10k,i} + 0.1\epsilon_i \quad (32)$$

where $\epsilon_i \sim \mathcal{N}(0, 1)$ for $i = 1, 2, \dots, m-1$. The tangent linear iteration corresponding to this iterative update (31) is:

$$\delta\mathbf{u}_{k+1} = [\mathbb{I} - \delta t (R^{-1}S_1[\mathbf{u}_k] + R^{-1}S_2[\mathbf{u}_k] + \gamma R^{-1}T)] \delta\mathbf{u}_k, \quad (33)$$

where the matrix $S_2[\mathbf{u}_k]$ is given by:

$$S_2[\mathbf{u}_k] = \begin{pmatrix} -\frac{u_k^2}{6} & \frac{2u_k^1+u_k^2}{6} & & \\ -\frac{u_k^1+2u_k^2}{6} & \frac{u_k^1-u_k^3}{6} & \frac{2u_k^2+u_k^3}{6} & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \frac{2u_k^{m-1}+u_k^m}{6} \\ & & & -\frac{u_k^{m-1}+2u_k^m}{6} & \frac{u_k^{m-2}-u_k^m}{3} \end{pmatrix},$$

which leads to the Jacobian matrix $\nabla H(\mathbf{u}_k)$ and its adjoint $\nabla H(\mathbf{u}_k)^\top$.

We then apply the linearized multi-block ADMM with regularization (23) to solve the 4D-Var problem (18). The initial condition is set as $\mathbf{u}_k^0 = (0, 0, \dots, 0)^\top$ for $k = 0, 1, \dots, N$, with parameters $\mu = 20$, $\eta = 0.1$, and $s = 2/3$. The numerical performance, displayed in Figure 9, show the convergence behaviors similar to that observed in Figure 6. As the iterations increase, the total error stabilizes, while the constraint error consistently decreases. Figure 10 demonstrates the numerical performance of the dynamics recovered via the linearized multi-block ADMM with regularization (23). The recovered dynamics closely track the true numerical solution, with only minor derivations due to noise in the observational data, and the accuracy improves over time. A key observation is that the finite difference method exhibits numerical dissipation without high-frequency oscillations, while the finite element method lacks numerical dissipation but contains high-frequency oscillations. The solution recovered via the multi-block ADMM with regularization(23) reflects the characteristics of the different numerical discretizations used to generate the observational data, highlighting this method's effectiveness in solving the 4D-Var problem (5) and its dependence on the observational data.

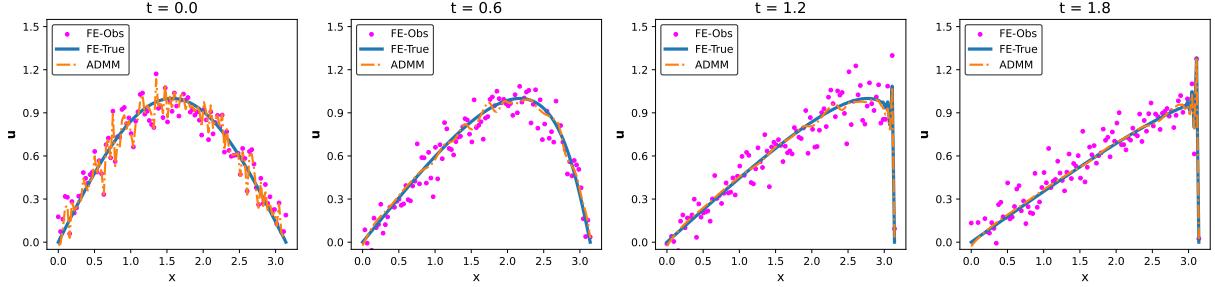


Figure 10. Comparison of noisy observational data with the time evolution dynamics recovered via the linearized multi-block ADMM with regularization (23), with the true numerical solution as the reference.

3.3 Spectral method

For the spectral method, we also set the time step size $\delta t = 0.01$ and the total evolution time $T = 2$. Let m be the total degrees of freedom. Given the Dirichlet boundary condition $u(t, 0) = u(t, \pi) = 0$, we select sine functions $\sin ix$ for $i = 1, 2, \dots, m$ as the orthogonal basis. For each time step $k = 0, 1, \dots, N = T/\delta t$, the numerical solution is expressed as the following linear combination:

$$u_k(x) = u_k^{(m)}(x) = \sum_{i=1}^m u_{k,i} \sin ix,$$

which serves as the spectral approximation for the analytic solution $u(k\delta t, x)$. The vector of coefficients $\mathbf{u}_k = (u_{k,1}, u_{k,2}, \dots, u_{k,m})^\top$, satisfies the following iterative scheme:

$$\begin{cases} \frac{u_{k+1,1} - u_{k,1}}{\delta t} - \frac{1}{2} \sum_{l=1}^{m-1} u_{k,l} u_{k,l+1} = -\gamma u_{k,1}, \\ \frac{u_{k+1,i} - u_{k,i}}{\delta t} + \frac{i}{4} \left(\sum_{l=1}^i u_{k,l} u_{k,i-l} - 2 \sum_{l=1}^{m-i} u_{k,l} u_{k,i+l} \right) = -\gamma i^2 u_{k,i}, \quad \text{for } i = 2, 3, \dots, m. \end{cases} \quad (34)$$

The observational setup remains the same, with observational time $T_o = 0.2$ and $n + 1$ observational time points, where $n = T/T_o = 10$. The numerical solution is generated using the spectral method (34), starting from the initial condition $\mathbf{u}_0 = (1, 0, \dots, 0)^\top$. At each time observational point, the numerical solution \mathbf{u}_{10k} is recorded for $i = 0, 1, \dots, n$ with each component denoted as $u_{10k,i}$ for $i = 1, 2, \dots, m - 1$. Noisy observational data $\hat{\mathbf{u}}_k$ is generated by adding Gaussian noise to the observations:

$$\hat{u}_{k,i} = u_{10k,i} + 0.1\sqrt{2} \cdot 0.1\varepsilon_i \quad (35)$$

where $\varepsilon_i \sim \mathcal{N}(0, 1)$ for $i = 1, 2, \dots, m - 1$ and the scaling factor $0.1\sqrt{2}$ ensures consistency with previous cases.

Before applying the linearized multi-block ADMM with regularization (23) to solve the 4D-Var problem (18), we first derive the tangent linear iteration corresponding to the iterative scheme (34)

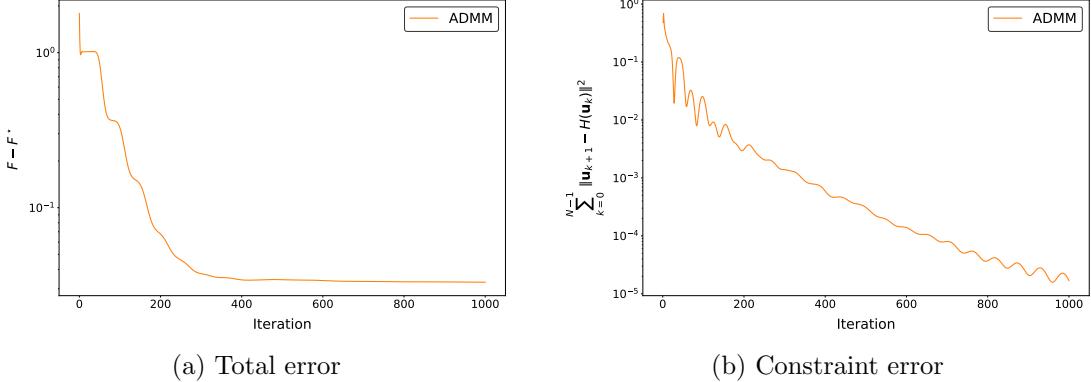


Figure 11. The linearized multi-block ADMM with regularization (23) is applied to the 4D-Var problem (9) for the viscous Burgers' equation (25), using the spectral method (34).

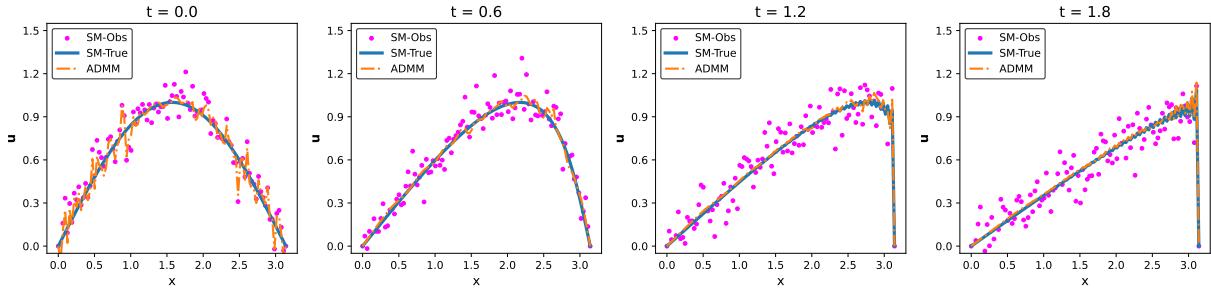


Figure 12. Comparison of noisy observational data with the time evolution dynamics recovered via the linearized multi-block ADMM with regularization (23), with the true numerical solution as the reference.

as follows:

$$\begin{cases} \delta u_{k+1,1} = \delta u_{k,1} + \delta t \left(\frac{1}{2} \sum_{l=1}^{m-1} \delta u_{k,l} \cdot u_{k,l+1} + \frac{1}{2} \sum_{l=1}^{m-1} u_{k,l} \cdot \delta u_{k,l+1} - \gamma \delta u_{k,1} \right), \\ \delta u_{k+1,i} = \delta u_{k,i} - \delta t \gamma i^2 \cdot \delta u_{k,i} \\ \quad - \delta t \cdot \frac{i}{4} \left(\sum_{l=1}^i \delta u_{k,l} \cdot u_{k,i-l} + \sum_{l=1}^i u_{k,l} \cdot \delta u_{k,i-l} - 2 \sum_{l=1}^{m-i} \delta u_{k,l} \cdot u_{k,i+l} - 2 \sum_{l=1}^{m-i} u_{k,l} \cdot \delta u_{k,i+l} \right), \end{cases} \quad (36)$$

for $i = 2, 3, \dots, m$,

which helps us derive the Jacobian matrix $\nabla H(\mathbf{u}_k)$ and its adjoint $\nabla H(\mathbf{u}_k)^\top$. The iterative process begins with the initial condition as $\mathbf{u}_k^0 = (0, 0, \dots, 0)^\top$ for $k = 0, 1, \dots, N$, using the parameters $\mu = 20$, $\eta = 0.1$, and $s = 2/3$. The numerical performance is illustrated in Figure 11, showing consistency with the convergence behaviors observed in Figure 6 and Figure 9. Furthermore, Figure 12 highlights the recovered dynamics, indicating that the solution recovered via the multi-block ADMM with regularization(23) effectively captures the characteristics of the different numerical discretizations used to generate the observational data.

Finally, a comparison of the spectral method (34) with the finite difference method (26) and the finite element method (31) reveals key distinctions in their computational complexity. The spectral method includes $m^2 - 1$ quadratic terms in the spectral method, while the finite difference method has $2m - 2$, and the finite element method involves $6m - 10$ quadratic terms. This difference leads

Numerical Methods	Computational Time
Finite Difference	65.5021s
Finite Element	232.0715s
Spectral Method	5113.0345s

Table 1. Comparison of computational times for the linearized multi-block ADMM with regularization (23) across the finite difference (26), finite element (31), and spectral (34) discretizations. Python 3.11.5 and Numpy 1.25.2 were executed on an Intel® Core™ i5-12500, 3.00 GHz Processor (12th Generation).

the tangent linear iteration (36) to introduce nonlinear terms that scale as $O(m^2)$ for the spectral method. As a result, the computational cost, particularly for calculating the adjoint operator, increases significantly. This higher complexity makes the spectral method substantially slower compared to the other methods, as evidenced by the computational times presented in Table 1.

4 Vorticity concentration in large-scale 2D turbulence

In this section, we apply the linearized multi-block ADMM with regularization (23) to solve the 4D-Var problem (18) in the context of the large-scale 2D turbulence, with a particular focus on the phenomenon of vorticity concentration. Let Ω be an open domain, and the vorticity field is assumed to satisfy $\omega \in L^2([0, T); H_0^1(\Omega))$ and $\omega_t \in L^2([0, T); H^{-1}(\Omega))$. The governing 2D vorticity equation is given by:

$$\begin{cases} \partial_t \omega + J(\psi, \omega) = -\kappa \Delta^2 \omega, \\ \omega = \Delta \psi, \end{cases} \quad (37)$$

where the Jacobian J , for any bivariate functions u and v , is defined as:

$$J(u, v) = \partial_x u \partial_y v - \partial_y u \partial_x v, \quad (38)$$

and the biharmonic operator Δ^2 serves as a scale-selective dissipation, filtering out high-frequency waves, as described in [McW84]. We impose a slip boundary condition:

$$\omega|_{\partial\Omega} = 0, \quad (39)$$

which indicates the absence of tangential stress along the boundary for large-scale flow. Small-scale processes occurring near the boundary act as a buffer, allowing the large-scale flow to slip smoothly along the boundary. This condition is commonly used in large-scale ocean circulation models [Ped96, Section 2.4].

4.1 Computational Implementation

For the numerical implementation, we utilize a finite difference scheme, applying the Arakawa Jacobian, which is specifically designed to conserve energy and enstrophy in 2D fluid dynamics. In particular, we adopt the second-order Arakawa Jacobian as outlined in [CRB11, Section 16.7],

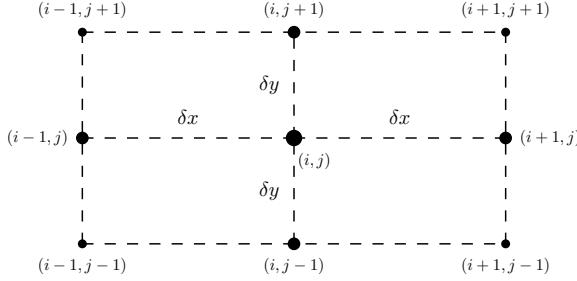


Figure 13. Spatial grids for numerical Arakawa Jacobian $J(u, v)$ around the central point labeled (i, j) .

which consists of three components:

$$\left\{ \begin{array}{l} J_1^{i,j} = \frac{(u^{i+1,j} - u^{i-1,j})(v^{i,j+1} - v^{i,j-1}) - (u^{i,j+1} - u^{i,j-1})(v^{i+1,j} - v^{i-1,j})}{4\delta x \delta y}, \\ J_2^{i,j} = \frac{u^{i+1,j}(v^{i+1,j+1} - v^{i+1,j-1}) - u^{i-1,j}(v^{i-1,j+1} - v^{i-1,j-1})}{4\delta x \delta y} \\ \quad - \frac{u^{i,j+1}(v^{i+1,j+1} - v^{i-1,j+1}) - u^{i,j-1}(v^{i+1,j-1} - v^{i-1,j-1})}{4\delta x \delta y}, \\ J_3^{i,j} = \frac{(u^{i+1,j+1} - u^{i-1,j+1})v^{i,j+1} - (u^{i+1,j-1} - u^{i-1,j-1})v^{i,j-1}}{4\delta x \delta y} \\ \quad - \frac{(u^{i+1,j+1} - u^{i+1,j-1})v^{i+1,j} - (u^{i-1,j+1} - u^{i-1,j-1})v^{i-1,j}}{4\delta x \delta y}. \end{array} \right.$$

The total Arakawa Jacobian is then computed as the average of these three components:

$$J^{i,j} = \frac{J_1^{i,j} + J_2^{i,j} + J_3^{i,j}}{3}. \quad (40)$$

Figure 13 illustrates the spatial grid around the central grid point (i, j) , highlighting the neighboring points used in the computation of the Jacobian. Numerically, both \mathbf{u} and \mathbf{v} are treated as vectors, when substituting a variable into the Arakawa Jacobian (40), it acts as a linear operator on the other variable, expressed as

$$J(\mathbf{u}, \mathbf{v}) = (J^{i,j})_{m^2 \times m^2} = \mathbb{J}[\mathbf{u}]\mathbf{v} = -\mathbb{J}[\mathbf{v}]\mathbf{u}. \quad (41)$$

To solve the inverse Laplacian, we utilize the successive over-relaxation (**SOR**) method [GVL13], though the conjugate gradient method [HS52] is also a widely used approach. For time discretization, we employ a prediction-correction scheme:

$$\left\{ \begin{array}{l} \frac{\omega_k^p - \omega_k}{\delta t} + J(\psi_k, \omega_k) = -\kappa \Delta^2 \omega_k, \end{array} \right. \quad (42a)$$

$$\left. \begin{array}{l} \frac{\omega_{k+1} - \omega_k}{\delta t} + J(\psi_k, \omega_k^p) = -\kappa \Delta^2 \omega_k^p. \end{array} \right. \quad (42b)$$

which ensures both accuracy and stability. The tangent linear iterations for the prediction-correction scheme (42) are given by:

$$\left\{ \begin{array}{l} \frac{\delta \omega_k^p - \delta \omega_k}{\delta t} - \mathbb{J}[\omega_k] \Delta^{-1} \delta \omega_k + \mathbb{J}[\Delta^{-1} \omega_k] \delta \omega_k = -\kappa \Delta^2 \delta \omega_k, \end{array} \right. \quad (43a)$$

$$\left. \begin{array}{l} \frac{\delta \omega_{k+1} - \delta \omega_k}{\delta t} - \mathbb{J}[\omega_k^p] \Delta^{-1} \delta \omega_k + \mathbb{J}[\Delta^{-1} \omega_k] \delta \omega_k^p = -\kappa \Delta^2 \delta \omega_k^p. \end{array} \right. \quad (43b)$$

Reformulating (42a) gives:

$$\omega_k^p = \mathbb{P}\omega_k = [\mathbb{I} - \delta t (\mathbb{J}[\Delta^{-1}\omega_k] - \kappa\Delta^2)]\omega_k, \quad (44)$$

and similarly, reformulating (43a) gives:

$$\delta\omega_k^p = \delta\mathbb{P}\delta\omega_k = [\mathbb{I} + \delta t (\mathbb{J}[\omega_k]\Delta^{-1} - \mathbb{J}[\Delta^{-1}\omega_k] - \kappa\Delta^2)]\delta\omega_k. \quad (45)$$

Substituting (44) and (45) into (43b) allows us to derive the tangent linear iteration:

$$\delta\omega_{k+1} = [\mathbb{I} + \delta t (\mathbb{J}[\mathbb{P}\omega_k]\Delta^{-1} - (\mathbb{J}[\Delta^{-1}\omega_k] + \kappa\Delta^2)\delta\mathbb{P})]\delta\omega_k \quad (46)$$

and the adjoint operator of $\nabla H(\omega_k)$ is given by:

$$\nabla H(\omega_k)^\top = [\mathbb{I} + \delta t (\Delta^{-1}\mathbb{J}[\mathbb{P}\omega_k]^\top - \delta\mathbb{P}^\top(\mathbb{J}[\Delta^{-1}\omega_k]^\top + \kappa\Delta^2))], \quad (47)$$

which highlights the complexity of computing the adjoint operator in this context. Before implementing the linearized multi-block **ADMM** with regularization (23) to solve the **4D-Var** problem (18), we must establish the energy norm for the objective function as:

$$\begin{aligned} F(\omega_0, \omega_1, \dots, \omega_n) &= \frac{T_o}{2} \sum_{k=0}^n \|\nabla^\top(\psi_k - \hat{\psi}_k)\|^2 + \frac{\alpha}{2} \|\nabla^\top(\psi_0 - \hat{\psi}_0)\|^2 \\ &= \frac{T_o}{2} \sum_{k=0}^n \|\Delta^{-\frac{1}{2}}(\omega_k - \hat{\omega}_k)\|^2 + \frac{\alpha}{2} \|\Delta^{-\frac{1}{2}}(\omega_0 - \hat{\omega}_0)\|^2, \end{aligned} \quad (48)$$

where the second equation results from integration by parts.

4.2 Numerical performance

We demonstrate the numerical performance within a box domain defined as $\Omega = [-2L, 2L] \times [-2L, 2L]$ where $L = 1$. The spatial grids are configured with $\delta x = \delta y = 0.2$, resulting in $4L/\delta x = 4L/\delta y = m = 20$ grid points along each axis. The time step size is chosen as $\delta t = 3\delta x\delta y$, and the biharmonic coefficient is specified as $\kappa = 0.001\delta x\delta y$. Notably, the equations under consideration have been nondimensionalized. With an iteration count of $N = 300$, we achieve a total simulation time $T = N\delta t = 36$. The initial condition is randomly generated as $\omega_{0;ij} \sim 5\mathcal{N}(0, 1)$ for $i, j = 1, 2, \dots, m-1$. The numerical solution, considered as the precise observation, is obtained following the procedures outlined in Section 4.1. To generate noisy observational data, we add Gaussian noise to the observations: $\hat{\omega}_{k,ij} = \omega_{30,k,ij} + 0.5\varepsilon_{ij}$, where $\varepsilon_{ij} \sim \mathcal{N}(0, 1)$ for $i, j = 1, 2, \dots, m-1$. The linearized multi-block **ADMM** with regularization (23) begins with the initial condition as $\omega_k^0 = 0_{(m-1) \times (m-1)}^\top$ for $k = 0, 1, \dots, N$, using the parameters $\mu = 20$, $\eta = 0.1$, and $s = 2/3$. The numerical performance is illustrated in Figure 14, which shows consistency with the convergence behaviors observed for the Lorenz system in Section 2 and various numerical schemes of the viscous Burgers' equation in Section 3. Furthermore, Figure 15 highlights the recovered solution closely matches the true dynamics, even in the presence of noisy observational data. The numerical errors, presented in Figure 16, further verify that the accuracy improves over time as the simulation progresses.

By considering the scales of large-scale ocean circulation, with $L \sim 2.5\text{km}$ and vorticity $\omega_0 \sim 10^{-5}\text{s}^{-1}$, we derive a characteristic time scale of $\delta t \sim 7.2\text{h}$. With an iteration count of $N = 300$, the total simulation time becomes $T = N\delta t \approx 6$ years, allowing for a comprehensive characterization of

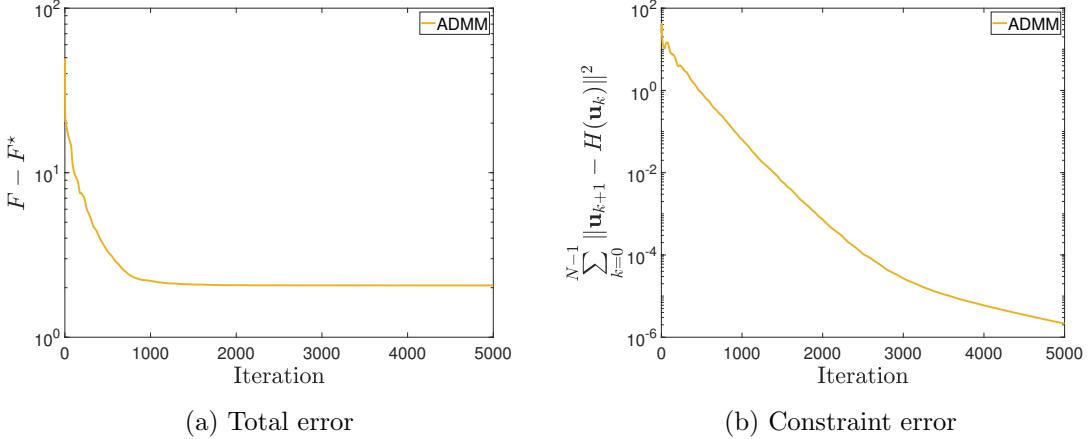


Figure 14. The linearized multi-block ADMM with regularization (23) is applied to the 4D-Var problem (18) for the Burgers' equation with the 2D large-scale vorticity equation (37), using the finite difference scheme described in Section 4.1.

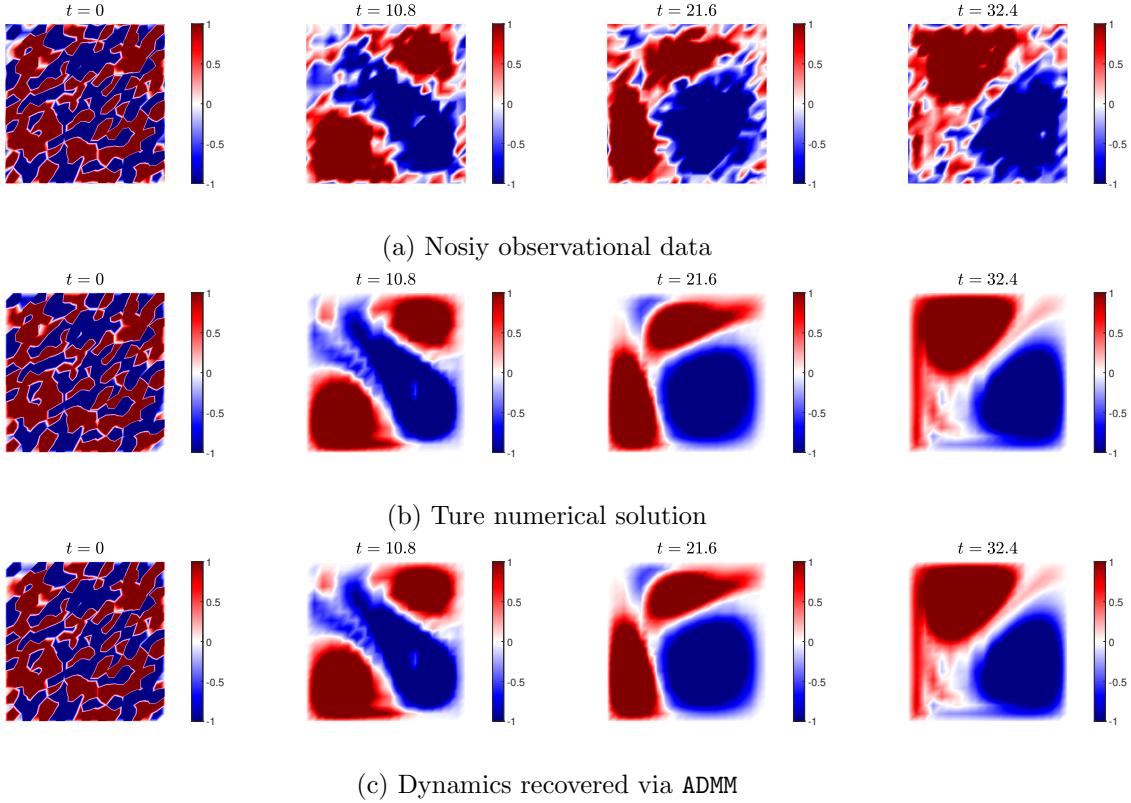


Figure 15. Comparison of noisy observational data with the time evolution dynamics recovered via the linearized multi-block ADMM with regularization (23), with the true numerical solution as the reference.

the large-scale ocean circulation. Additionally, we note that a commonly used boundary condition is the periodic boundary condition [McW84], which necessitates the use of spectral methods rather than finite difference methods. The reason for this preference is that convergence using SOR and the conjugate gradient methods is typically poor due to the presence of a zero eigenvalue in the

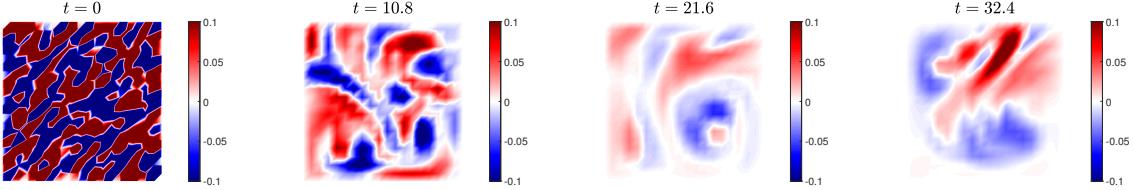


Figure 16. The error between the solution recovered via the linearized multi-block ADMM with regularization (23) and the true numerical solution.

Laplacian operator.

5 Conclusions

In this study, we proposed a linearized multi-block ADMM with regularization to solve the 4D-Var problem, leveraging its separable structure to improve computational efficiency and robustness. Unlike classical gradient-based algorithms, which are sensitive to initial conditions, our approach derives the Euler-Lagrange equation for the entire dynamical system, allowing for more effective utilization of observational data. When the initial condition is poorly chosen, the arg min operation steers the iteration toward the observational data, reducing sensitivity to the initial guess. The quadratic subproblems simplify the solution process, while the parallel structure enhances computational efficiency, especially with modern hardware.

We demonstrated the superior performance of our method using the Lorenz system and the viscous Burgers' equation, across various numerical schemes, including finite difference, finite element, and spectral methods. Additionally, we showed its robustness in handling noisy observational data and complex scenarios like 2D turbulence with vorticity concentration. Despite these advances, several challenges remain, including the efficient computation of the adjoint operator, especially in higher-dimensional problems such as 2D turbulence.

Future research may explore convergence proofs for the linearized multi-block ADMM with regularization, particularly over long time horizons, and investigate its extension to infinite-dimensional spaces. The application of stochastic gradient descent (SGD) and other advanced optimization techniques in 4D-Var may offer further improvements, especially for complex physical systems like the nonlinear Schrödinger or Ginzburg-Landau equations. Additionally, connections with classical iterative methods, such as Picard and Newton iterations, and advanced frameworks like KAM and Nash-Moser iterations, could deepen the understanding and applications of ADMM in infinite-dimensional optimization.

References

- [ABN16] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.
- [BC02] François Bouttier and Philippe Courtier. Data assimilation concepts and methods march 1999. *Meteorological training course lecture series. ECMWF*, 718:59, 2002.
- [BFO20] Erik Burman, Ali Feizmohammadi, and Lauri Oksanen. A finite element data assimilation method for the wave equation. *Mathematics of Computation*, 89(324):1681–1709, 2020.
- [BLNZ95] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

- [BO18] Erik Burman and Lauri Oksanen. Data assimilation for the heat equation using stabilized finite element methods. *Numerische Mathematik*, 139:505–528, 2018.
- [CRB11] Benoit Cushman-Roisin and Jean-Marie Beckers. *Introduction to geophysical fluid dynamics: physical and numerical aspects*, volume 101 of *International Geophysics*. Academic press, 2nd edition, 2011.
- [DY99] Yu-Hong Dai and Yaxiang Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on optimization*, 10(1):177–182, 1999.
- [DY16] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66:889–916, 2016.
- [EBNP23] Lahcen El Bourkhissi, Ion Necoara, and Panagiotis Patrinos. Linearized ADMM for nonsmooth nonconvex optimization with nonlinear equality constraints. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 7312–7317. IEEE, 2023.
- [FR64] Reeves Fletcher and Colin M Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964.
- [GM75] Roland Glowinski and Americo Marroco. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975.
- [GM76] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & mathematics with applications*, 2(1):17–40, 1976.
- [GVL13] Gene H Golub and Charles F Van Loan. *Matrix computations*. Johns Hopkins University Press, 4th edition, 2013.
- [HHY15] Bingsheng He, Liusheng Hou, and Xiaoming Yuan. On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming. *SIAM Journal on Optimization*, 25(4):2274–2312, 2015.
- [HP24] Le Thi Khanh Hien and Dimitri Papadimitriou. An inertial ADMM for a class of nonconvex composite optimization with nonlinear coupling constraints. *Journal of Global Optimization*, pages 1–22, 2024.
- [HS52] Magnus Rudolph Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.
- [Kal02] E Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2002.
- [LDT86] François-Xavier Le Dimet and Olivier Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, 38(2):97–110, 1986.
- [LHGD24] Xuejian Li, Xiaoming He, Wei Gong, and Craig C Douglas. Variational data assimilation with finite-element discretization for second-order parabolic interface equation. *IMA Journal of Numerical Analysis*, page drae010, 2024.
- [Lio71] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, 1971.
- [Lor63] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141, 1963.
- [LSZ15] Kody Law, Andrew Stuart, and Kostas Zygalakis. Data assimilation: A mathematical introduction. *Cham, Switzerland: Springer*, 214:52, 2015.

- [Mar75a] G. I. Marchuk. Formulation of the theory of perturbations for complicated models. *Applied Mathematics and Optimization*, 2(1):1–33, 1975.
- [Mar75b] G. I. Marchuk. Formulation of the theory of perturbations for complicated models, Part II: Weather prediction. *Geofisica International*, 15(2):169–182, 1975.
- [McW84] James C McWilliams. The emergence of isolated coherent vortices in turbulent flow. *Journal of Fluid Mechanics*, 146:21–43, 1984.
- [MW82] Walter Heinrich Munk and Carl Isaac Wunsch. Observing the ocean in the 1990s. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 307(1499):439–464, 1982.
- [Ped96] Joseph Pedlosky. *Ocean circulation theory*. Springer Science & Business Media, 1996.
- [PR69] E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 3(R1):35–43, 1969.
- [Sas58] Yoshikazu Sasaki. An objective analysis based on the variational method. *Journal of the Meteorological Society of Japan. Ser. II*, 36(3):77–88, 1958.
- [Sas69] Yoshikazu Sasaki. Proposed inclusion of time. variation terms, observational and theoretical, in numerical variational objective analysis. *Journal of the Meteorological Society of Japan. Ser. II*, 47(2):115–124, 1969.
- [Sas70] Yoshikazu Sasaki. Some basic formalisms in numerical variational analysis. *Monthly Weather Review*, 98(12):875–883, 1970.
- [SAST23] Daniel Sanz-Alonso, Andrew Stuart, and Armeen Taeb. *Inverse Problems and Data Assimilation*. London Mathematical Society Student Texts. Cambridge University Press, 2023.
- [SBH⁺16] Detlef Stammer, M Balmaseda, P Heimbach, A Köhl, and A Weaver. Ocean data assimilation in support of climate applications: Status and perspectives. *Annual review of marine science*, 8(1):491–518, 2016.
- [SGGC13] Gerold Siedler, Stephen M Griffies, John Gould, and John A Church. *Ocean circulation and climate: a 21st century perspective*. Academic Press, 2013.
- [Tal14] O. Talagrand. 4D-VAR: four-dimensional variational assimilation. In *Advanced Data Assimilation for Geosciences: Lecture Notes of the Les Houches School of Physics: Special Issue, June 2012*. Oxford University Press, 2014.
- [TC87] Olivier Talagrand and Philippe Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. Part I: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113(478):1311–1328, 1987.
- [Tho69] Philip D Thompson. Reduction of analysis error through constraints of dynamical consistency. *Journal of Applied Meteorology (1962-1982)*, pages 738–742, 1969.
- [WH13] Carl Wunsch and Patrick Heimbach. Dynamically and kinematically consistent global ocean circulation and ice state estimates. In *International Geophysics*, volume 103, pages 553–579. Elsevier, 2013.
- [Wun96] Carl Wunsch. *The Ocean Circulation Inverse Problem*. Cambridge University Press, 1996.
- [XW21] Yue Xie and Stephen J Wright. Complexity of proximal augmented lagrangian for nonconvex optimization with nonlinear equality constraints. *Journal of Scientific Computing*, 86:1–30, 2021.
- [ZBBO10] Xiaoqun Zhang, Martin Burger, Xavier Bresson, and Stanley Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.
- [ZBLN97] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.