

Sequential data assimilation for PDEs using shape-morphing solutions

Zachary T. Hilliard and Mohammad Farazmand*

Department of Mathematics, North Carolina State University, 2311 Stinson Drive, Raleigh, NC 27695, USA

Abstract

Shape-morphing solutions (also known as evolutionary deep neural networks, reduced-order nonlinear solutions, and neural Galerkin schemes) are a new class of methods for approximating the solution of time-dependent partial differential equations (PDEs). Here, we introduce a sequential data assimilation method for incorporating observational data in a shape-morphing solution (SMS). Our method takes the form of a predictor-corrector scheme, where the observations are used to correct the SMS parameters using Newton-like iterations. Between observation points, the SMS equations—a set of ordinary differential equations—are used to evolve the solution forward in time. We prove that, under certain conditions, the data assimilated SMS (DA-SMS) converges uniformly towards the true state of the system. We demonstrate the efficacy of DA-SMS on three examples: the nonlinear Schrödinger equation, the Kuramoto–Sivashinsky equation, and a two-dimensional advection-diffusion equation. Our numerical results suggest that DA-SMS converges with relatively sparse observations and a single iteration of the Newton-like method.

1 Introduction

Shape-morphing solutions (SMS) are a new class of approximate solutions to spatiotemporal partial differential equations (PDEs). A distinguishing feature of SMS is that they depend nonlinearly on a set of time-dependent parameters. In their simplest form, SMS can be expressed as

$$\hat{u}(\mathbf{x}, \boldsymbol{\theta}(t)) = \sum_{i=1}^N a_i(t) \phi_i(\mathbf{x}, \boldsymbol{\beta}_i(t)), \quad (1)$$

where the parameters $\boldsymbol{\theta}(t) = \{a_i(t), \boldsymbol{\beta}_i(t)\}_{i=1}^N$ comprise the amplitudes a_i and the shape parameters $\boldsymbol{\beta}_i$. Equation (1) can be viewed as a spectral method with the basis functions ϕ_i [1, 3]. Unlike classical spectral methods where the basis functions are static, shape-morphing solutions allow the basis to change over time via their dependence on the shape parameters $\boldsymbol{\beta}_i(t)$. As such, shape-morphing solutions can more easily adapt to the true solution of the PDE.

*Corresponding author: farazmand@ncsu.edu

An alternative interpretation is to view equation (1) as a shallow neural network with time-dependent parameters. In this case, ϕ_i is the activation function of the i -th node, β_i are the weights and biases of the node, and a_i is the corresponding output weight. This interpretation can be easily extended to deep neural networks with time-dependent parameters [15].

Given a shape-morphing solution $\hat{u}(\mathbf{x}, \boldsymbol{\theta}(t))$, the evolution of its parameters $\boldsymbol{\theta}(t)$ are determined by a set of ordinary differential equations (ODEs), ensuring that the SMS closely approximates the true solution of the PDE. As we review in Section 1.1, the theory of shape-morphing solutions has been developing rapidly over the past few years. In the literature, the approximate solution $\hat{u}(\mathbf{x}, \boldsymbol{\theta}(t))$ is referred to by various names, such as evolutionary deep neural networks [15], reduced-order nonlinear solutions [1], and neural Galerkin schemes [9]. Here, with hopes of establishing a unifying terminology, we refer to them as shape-morphing solutions or SMS, for short. We refer to the ODEs that govern the evolution of the parameters $\boldsymbol{\theta}(t)$ as the shape-morphing equation or the SMS equation.

The purpose of the present work is to introduce a data assimilation method which uses observational data from the system to steer the shape-morphing solution closer to the true dynamics of the system. As with any other numerical method, data assimilation is needed to correct for modeling error, truncation error, and uncertainties in the initial condition [31]. Our method takes the form of a predictor-corrector scheme, where the shape-morphing equations are used to evolve the solution between observation points. At time instances where observational data are available, they are utilized to correct for errors using Newton-like iterations. We prove that, if enough observations are available, the data assimilated SMS converges uniformly towards the true solution of the system. We demonstrate the efficacy of our method on three numerical examples.

1.1 Related Work

Shape-morphing solutions were first developed by Du and Zaki [15] in the context of deep neural networks and simultaneously by Anderson and Farazmand [1] in the context of model order reduction. Du and Zaki [15] approximate the solution of a PDE using a deep neural network whose weights and biases are time-dependent, hence referring to the resulting method as the *evolutional deep neural network* (EDNN). They derive a set of ordinary differential equations for evolving the parameters of the network, eliminating the need for a training process. Anderson and Farazmand [1] independently arrived at the same set of equations where they considered reduced-order models whose modes depend nonlinearly on a set of time-dependent parameters, thus allowing for the modes to adapt to the solution of the PDE. The resulting method is referred to as *reduced-order nonlinear solutions* (RONS).

Ever since these original papers, there has been a rapidly growing body of work addressing various theoretical and computational aspects of shape-morphing solutions. For instance, Bruna et al. [9] introduced an adaptive sampling strategy to reduce the computational cost of the Monte Carlo sampling required by EDNN. Coining yet another terminology, they refer to their method as *neural Galerkin schemes*.

Anderson and Farazmand [3] develop a number of methods to address the computational cost of SMS: (i) They developed a symbolic computing approach to reduce the computational cost of forming the SMS equations. (ii) When symbolic computing is not feasible, they introduced a collocation point method as an alternative and uncovered its relation to the Monte Carlo sampling method of [15]. (iii) They also introduced a regularization to the SMS equations which speeds up their time integration. We also refer to Ref. [19] which addresses similar computational issues.

Anderson and Farazmand [4] applied these computational advances to solve the Fokker–Planck equation in higher dimensions and discovered that the metric tensor in SMS coincides with the Fisher information metric.

Shape-morphing solutions may over-parameterize the solution of the PDE, leading to redundant parameters. Berman and Peherstorfer [8] introduced a randomized projection method for eliminating such redundant parameters.

Enforcing the PDE’s boundary conditions in shape-morphing solutions, and in particular evolutionary neural networks, can be challenging [15, 44]. Kast and Hesthaven [26] use embedding methods to ensure that the boundary conditions of the PDE are enforced directly into the neural network architecture.

Anderson and Farazmand [1] had already derived a set of SMS equations which enforce nonlinear constraints such as conservation of mass, energy, and other first integrals. As mentioned by Hilliard and Farazmand [22], these equations conserve the first integrals in continuous-time, necessitating a special time stepping scheme to ensure their conservation after discretization. To address this issue, Schwerdtner et al. [41] propose a discrete-in-time version of conservative SMS.

Shape-morphing solutions can be viewed as the flow of a vector field on a manifold embedded in the function space of the PDE [1]. Often this vector field is evaluated pointwise during the time integration process. Gaby et al. [20] propose to learn the vector field on the SMS manifold as a neural network. They subsequently evolve the SMS equations by replacing the vector field with its neural net representation.

Following Ref. [15], Kim and Zaki [28] developed a multi-network approach to shape-morphing solutions so that each state variable is represented by a separate EDNN. In addition, they partition the spatial domain into subsets so that the PDE is solved on each subset with its own corresponding EDNN.

In spite of the fast growing body of work on this topic, assimilating observational data in shape-morphing solutions has remained unaddressed. The purpose of the present paper is to introduce a data assimilation (DA) method for SMS approximation of PDEs. To this end, standard data assimilation methods, such as extended Kalman filters [23, 24, 33, 40], variational DA [5, 12, 35] and Bayesian DA [6, 37], can in principle be used. However, compared to our proposed method, Kalman filtering and variational methods are computationally more expensive as they require multiple solves of the underlying PDE and/or the corresponding adjoint equation [45]. Furthermore, the nonlinear dependence of SMS $\hat{u}(\mathbf{x}, \boldsymbol{\theta}(t))$ on its parameters $\boldsymbol{\theta}$ inhibits straightforward implementation of classical DA methods. In contrast, our proposed predictor-corrector DA scheme is easy to implement and avoids repeated solves of the PDE.

1.2 Main Contributions

The main contributions of this paper can be summarized as follows.

1. Discrete-time data assimilated SMS (DA-SMS): We introduce a sequential data assimilation algorithm that propagates the solution forwards in time and incorporates observational data into the solution as they become available at discrete times.
2. Convergence result: We prove that, if enough observational data is available, the DA-SMS method converges towards the true solution of the system.
3. Continuous-time data assimilation: We also present a DA method for SMS in case the observational data is measured with high temporal frequency.

4. As a byproduct of our work, we introduce a new method for directly encoding mixed Dirichlet–Neumann boundary conditions in the architecture of a neural network.
5. We demonstrate the application of our method on three examples with one and two spatial dimensions.

1.3 Outline

The remainder of the paper is organized in the following manner. In Section 2, we review the method of shape-morphing solutions and how we evolve the solutions forwards in time. In Section 3, we introduce our main results for sequential data-assimilation. We discuss both discrete DA-SMS algorithm and its continuous-time version. In Section 4, we present three numerical examples: the nonlinear Schrödinger (NLS) equation in Section 4.1, the Kuramoto–Sivashinsky (KS) equation in Section 4.2, and the two-dimensional advection-diffusion (AD) equation in Section 4.3. We present our concluding remarks in Section 5.

2 Mathematical Preliminaries

In this section, we summarize the theory of shape-morphing modes to obtain numerical solutions to PDEs [1, 3, 4, 2]. Consider a PDE of the form,

$$\frac{\partial u}{\partial t} = F(u), \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad (2)$$

where F is a differential operator and $u : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}$ denotes the solution, where $\Omega \subset \mathbb{R}^d$ is the spatial domain of the PDE. For any time t , the solution $u(\cdot, t)$ belongs to an appropriate Hilbert space \mathcal{H} with associated inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and induced norm $\| \cdot \|_{\mathcal{H}}$. For the purposes of this exposition, we assume that the solution $u(\mathbf{x}, t)$ is scalar-valued. However, this framework can easily be extended to vector-valued systems of PDEs as discussed in Ref. [22]. Additionally, here we only consider real-valued PDEs, but the extension to complex-valued PDEs is straightforward and can be found in Refs. [2, 22].

Next, we introduce the approximate shape-morphing solution,

$$\hat{u}(\mathbf{x}, \boldsymbol{\theta}(t)) = \sum_{i=1}^N a_i(t) \phi_i(\mathbf{x}, \boldsymbol{\beta}_i(t)), \quad (3)$$

with the *shape-morphing modes* ϕ_i , the corresponding amplitudes a_i , and the vector of *shape parameters* $\boldsymbol{\beta}_i \in \mathbb{R}^{p-1}$. We denote the full set of parameters by $\boldsymbol{\theta}(t) = \{a_i(t), \boldsymbol{\beta}_i(t)\}_{i=1}^N \in \mathbb{R}^{pN}$, where each term in (3) contains p parameters. The choice of the shape-morphing modes depends on the PDE. For instance, one can take a Gaussian mixture model where the shape-morphing modes are Gaussians and the shape parameters are the mean and the variance [2, 4]. A more general choice is an artificial neural network (ANN), where ϕ_i is the activation function, and the shape parameters are the corresponding weights and biases of the network [3, 15]. For example, a shallow neural network with a hyperbolic tangent activation function coincides with the shape-morphing solution,

$$\hat{u}(x, \boldsymbol{\theta}(t)) = \sum_{i=1}^N a_i(t) \tanh(w_i(t)x + b_i(t)), \quad (4)$$

whose shape-parameters are the N amplitudes a_i , weights w_i , and biases b_i , so that $\boldsymbol{\theta} = \{a_i, w_i, b_i\}_{i=1}^N \in \mathbb{R}^{3N}$.

In the conventional approach to ANNs for PDEs, the parameters are determined by minimizing a loss function [25, 29, 32]. The crucial difference in our approach is that SMS eliminates the need for this training step by evolving the time-dependent parameters $\boldsymbol{\theta}(t)$ through an explicit set of ordinary differential equations (ODEs).

To derive the ODEs governing the evolution of parameters, we introduce the residual function,

$$R(\mathbf{x}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) := \hat{u}_t - F(\hat{u}) = \sum_{j=1}^{pN} \frac{\partial \hat{u}}{\partial \theta_j} \dot{\theta}_j - F(\hat{u}). \quad (5)$$

which measures the discrepancy between the rate of change of the approximate shape-morphing solution and the dynamics governed by the PDE. To derive the corresponding system of ODEs, we seek $\dot{\boldsymbol{\theta}}$ which minimizes the instantaneous error,

$$\min_{\dot{\boldsymbol{\theta}} \in \mathbb{R}^{pN}} \frac{1}{2} \|R(\cdot, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|_{\mathcal{H}}^2. \quad (6)$$

The solution to this minimization problem is given explicitly in terms of the *shape-morphing equation* [1, 15],

$$M(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} = \mathbf{f}(\boldsymbol{\theta}), \quad (7)$$

where $M \in \mathbb{R}^{pN \times pN}$ is the *metric tensor*,

$$M_{ij}(\boldsymbol{\theta}) = \left\langle \frac{\partial \hat{u}}{\partial \theta_i}, \frac{\partial \hat{u}}{\partial \theta_j} \right\rangle_{\mathcal{H}}, \quad (8)$$

and the components of the vector field $\mathbf{f} : \mathbb{R}^{pN} \rightarrow \mathbb{R}^{pN}$ are given by

$$f_i(\boldsymbol{\theta}) = \left\langle \frac{\partial \hat{u}}{\partial \theta_i}, F(\hat{u}) \right\rangle_{\mathcal{H}}. \quad (9)$$

This set up is especially useful in the event that one can symbolically compute the integrals in Hilbert inner products for M and \mathbf{f} . In this case, one can exploit the structure of the metric tensor and the vector \mathbf{f} to make the implementation of symbolic computations more efficient. We refer to Ref. [3] for further detail regarding these computational aspects.

When symbolic computations of M and \mathbf{f} are not feasible, we instead opt for the collocation point method which was originally derived in Ref. [3]. In this case, we specify the N_c collocation points $\{\mathbf{x}_1, \dots, \mathbf{x}_{N_c}\} \subset \Omega$, and require that the residual function vanishes at these points, i.e.

$$R(\mathbf{x}_i, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \sum_{j=1}^{pN} \frac{\partial \hat{u}}{\partial \theta_j}(\mathbf{x}_i, \boldsymbol{\theta}) \dot{\theta}_j - F(\hat{u}) \Big|_{\mathbf{x}=\mathbf{x}_i} = 0, \quad i = 1, \dots, N_c. \quad (10)$$

We can more conveniently express (10) in one equation by defining the *collocation metric tensor* $\tilde{M}(\boldsymbol{\theta}) \in \mathbb{R}^{N_c \times pN}$ whose elements are give by

$$\tilde{M}_{ij}(\boldsymbol{\theta}) = \frac{\partial \hat{u}}{\partial \theta_j}(\mathbf{x}_i, \boldsymbol{\theta}), \quad (11)$$

and the vector field $\tilde{\mathbf{f}} \in \mathbb{R}^{N_c}$ such that

$$\tilde{f}_i(\boldsymbol{\theta}) = F(\hat{u}(\cdot, \boldsymbol{\theta})) \Big|_{\mathbf{x}=\mathbf{x}_i}. \quad (12)$$

With these definitions, we can express (10) as

$$\tilde{M}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \tilde{\mathbf{f}}(\boldsymbol{\theta}). \quad (13)$$

Note that no integration is required in the construction of the collocation metric tensor \tilde{M} or the vector field $\tilde{\mathbf{f}}$.

The collocation metric tensor \tilde{M} is rectangular which means that the system (13) can be either overdetermined or underdetermined. In such case, we obtain $\dot{\boldsymbol{\theta}}$ by solving the linear least squares problem,

$$\min_{\dot{\boldsymbol{\theta}} \in \mathbb{R}^{pN}} \frac{1}{2} \|\tilde{M}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - \tilde{\mathbf{f}}(\boldsymbol{\theta})\|_2^2, \quad (14)$$

which is equivalent to minimizing $\sum_i |R(\mathbf{x}_i, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})|^2$. Here, $\|\cdot\|_2$ denotes the Euclidean norm. The resulting *collocation shape-morphing equation* is

$$\dot{\boldsymbol{\theta}} = \tilde{M}^+(\boldsymbol{\theta})\tilde{\mathbf{f}}(\boldsymbol{\theta}), \quad (15)$$

in terms of the Moore–Penrose pseudo-inverse \tilde{M}^+ .

Lastly, in either the shape-morphing equation (7) or the collocation shape-morphing equation (13), the metric tensors can be ill-conditioned which leads to stiffness in the corresponding ODEs. In order to alleviate this issue, Anderson and Farazmand [3] regularize the optimization problems (6) and (14) using Tikhonov regularization. Introducing the regularization parameter γ , the regularized optimization problem (6) reads

$$\min_{\dot{\boldsymbol{\theta}} \in \mathbb{R}^{pN}} \frac{1}{2} \|R(\mathbf{x}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|_{\mathcal{H}}^2 + \frac{\gamma}{2} \|\dot{\boldsymbol{\theta}}\|_2^2, \quad (16)$$

whose solution is given by

$$(M(\boldsymbol{\theta}) + \gamma I)\dot{\boldsymbol{\theta}} = \mathbf{f}(\boldsymbol{\theta}), \quad (17)$$

where I denotes the $pN \times pN$ identity matrix. We similarly regularize the least squares problem (14) and solve

$$\min_{\dot{\boldsymbol{\theta}} \in \mathbb{R}^{pN}} \frac{1}{2} \|\tilde{M}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - \tilde{\mathbf{f}}(\boldsymbol{\theta})\|_2^2 + \frac{\gamma}{2} \|\dot{\boldsymbol{\theta}}\|_2^2. \quad (18)$$

The resulting regularized collocation shape-morphing equation reads

$$\left(\tilde{M}^\top(\boldsymbol{\theta})\tilde{M}(\boldsymbol{\theta}) + \gamma I\right)\dot{\boldsymbol{\theta}} = \tilde{M}^\top(\boldsymbol{\theta})\tilde{\mathbf{f}}(\boldsymbol{\theta}). \quad (19)$$

To evolve the parameters $\boldsymbol{\theta}$ in time, we solve the ODEs (17) or (19) using standard numerical time integrators such as explicit Runge–Kutta schemes.

Lastly, we emphasize that the metric tensors M and \tilde{M} as well as the vectors \mathbf{f} and $\tilde{\mathbf{f}}$ are functions of the parameters $\boldsymbol{\theta}$; but for the remainder of this manuscript we omit their dependence on $\boldsymbol{\theta}$ for notational simplicity.

3 Data Assimilation

In this section, we introduce our sequential data assimilated shape-morphing solutions algorithms. More specifically, in Section 3.1 we present the main result of our paper: a discrete sequential data assimilation algorithm used in conjunction with the shape-morphing solutions. We refer to this methodology as data-assimilated SMS or DA-SMS for short. In Section 3.2, we derive a convergence guarantees for DA-SMS. Lastly, in Section 3.3, we introduce a continuous time version of DA-SMS that instead uses the time derivative of the observational data.

3.1 Discrete Sequential Data Assimilation: DA-SMS

Consider a system described by PDE (2) with the initial condition u_0 . Any method for approximating the system’s evolution needs to contend with various sources of error: modeling error resulting from simplifying assumptions in deriving the PDE, uncertainties in the initial condition, truncation error from representing the solution with a finite-dimensional approximation, and time integration error. Data assimilation seeks to remedy these errors by incorporating direct observational data from the system into the numerical solution. Here, we develop data assimilation methods that are applicable when the approximate solution is obtained as a shape-morphing solution $\hat{u}(\cdot, \boldsymbol{\theta}(t))$ with nonlinear dependence on the parameters $\boldsymbol{\theta}$.

To begin, we assume that we have access to r sensors which take measurements of observable quantities associated with the system. We denote these observables by $y_j(t)$ for $j \in \{1, \dots, r\}$. These measurements are collected at m discrete points in time: $t_1 < t_2 < \dots < t_m$. We refer to the time interval between the initial time t_0 and the final observation time t_m as the *data-assimilation (DA) time window*, $\mathcal{T}_{DA} = [t_0, t_m]$. At each t_i , we collect the r measurements into a vector $\mathbf{y}(t_i) \in \mathbb{R}^r$ whose components we denote by,

$$y_j(t_i) = C_j(u(\cdot, t_i)) + \eta, \quad j = 1, \dots, r, \quad (20)$$

where $C_j : \mathcal{H} \rightarrow \mathbb{R}$ is the associated observation operator and η is the observational error or noise. The observables can be pointwise observations of the state of the system, i.e. $y_j(t_i) = u(\mathbf{x}_j, t_i)$, or they could be a nonlinear operator applied to the state of the system such as the modulus of the solution to the nonlinear Schrödinger equation as discussed later in Section 4.1.

Figure 1 illustrates the overall DA-SMS algorithm. Our goal is to use the data to correct for the errors introduced by the approximations. Our algorithm achieves this objective in two main steps. The first is the prediction step where we use the SMS equations derived in Section 2 to evolve the approximate solution from time t_{i-1} to t_i . In Figure 1, this step is represented by the black curves. The second piece of the DA-SMS algorithm, is the correction step, depicted by the red dashed lines in Figure 1. At time t_i , we freeze time and make a correction to the parameters $\boldsymbol{\theta}(t_i)$, using the true observational data $\mathbf{y}(t_i)$.

More specifically, during the correction step, we compute the SMS estimated observations defined by,

$$\hat{y}_j(t_i) = C_j[\hat{u}(\cdot, \boldsymbol{\theta}(t_i))], \quad j = 1, \dots, r. \quad (21)$$

For notational simplicity, we often write $C_j(\boldsymbol{\theta}(t_i))$ instead of $C_j[\hat{u}(\cdot, \boldsymbol{\theta}(t_i))]$. In the DA step of our algorithm, we seek a correction to the parameters $\boldsymbol{\theta}(t_i)$ that minimizes the discrepancy between the true data $\mathbf{y}(t_i)$ of the system and the predicted data $\hat{\mathbf{y}}(t_i)$. To this end, we solve the nonlinear equation,

$$\mathbf{C}(\boldsymbol{\theta}(t_i) + \delta\boldsymbol{\theta}) = \mathbf{y}(t_i), \quad (22)$$

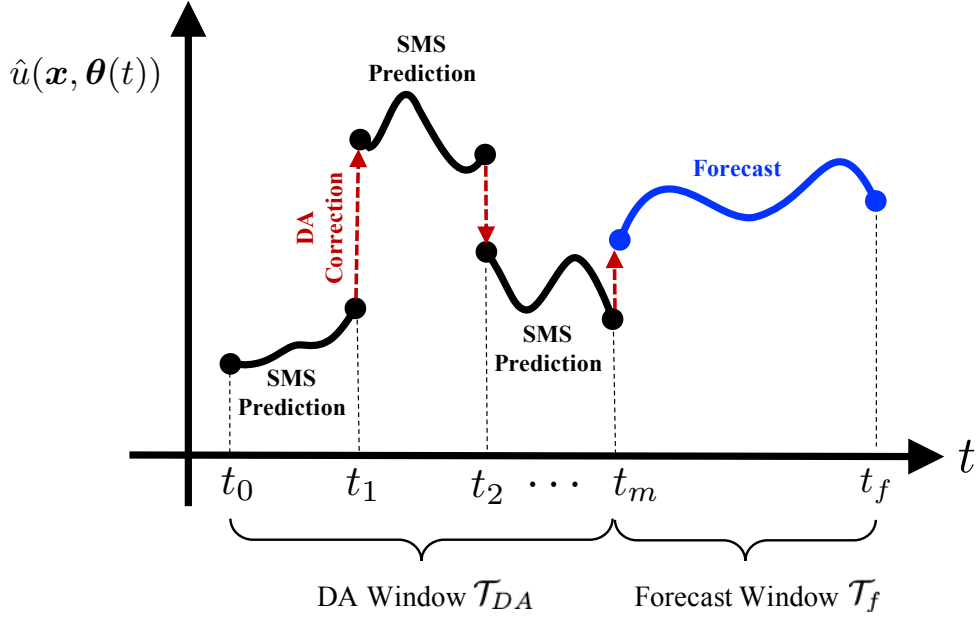


Figure 1: Schematic illustration of the DA-SMS algorithm.

Symbol	Description
$\theta_i^{(k)}$	Parameters at time t_i and the k -th Newton iteration: $\theta_i^{(k)} = \theta^{(k)}(t_i)$
$\hat{\mathbf{y}}_i^{(k)}$	SMS estimated observations at time t_i and k -th iteration: $\hat{\mathbf{y}}_i^{(k)} = \mathbf{C}[\hat{u}(\cdot, \theta_i^{(k)})]$
\mathbf{y}_i	True observational data at time t_i : $\mathbf{y}_i = \mathbf{y}(t_i)$

Table 1: The quantities used in the DA-SMS algorithm for notational simplicity.

where $\delta\theta$ is the correction step and $\mathbf{C} = [C_1(\cdot), C_2(\cdot), \dots, C_r(\cdot)]^\top$. Note that even if the observation operator is linear, $\mathbf{C}(\theta(t_i))$ is a nonlinear function of $\theta(t_i)$ due to the nonlinear dependence of the shape-morphing solution $\hat{u}(\cdot, \theta)$ on its parameters.

To solve the nonlinear equation (22), we introduce the Newton-like iterations,

$$\theta^{(k+1)}(t_i) = \theta^{(k)}(t_i) + \left[\nabla_{\theta} \mathbf{C} \left(\theta^{(k)}(t_i) \right) \right]^+ \left[\mathbf{y}(t_i) - \mathbf{C} \left(\theta^{(k)}(t_i) \right) \right], \quad (23)$$

where the superscript on θ denotes the k -th iteration.

To further simplify the notation, we introduce the following. We denote the parameters at time t_i and the k -th Newton iteration by $\theta_i^{(k)} = \theta^{(k)}(t_i)$, the true observational data at time t_i by $\mathbf{y}_i = \mathbf{y}(t_i)$, and the SMS estimated data at time t_i and the k -th Newton iteration by $\hat{\mathbf{y}}_i^{(k)} = \mathbf{C}[\hat{u}(\cdot, \theta^{(k)}(t_i))]$. In each of these, the subscript denotes the i -th time instance, and the superscript denotes the k -th Newton iteration. We summarize this notation in Table 1. Lastly, we introduce the Jacobian matrix

$J_r(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) \in \mathbb{R}^{r \times pN}$, so that

$$J_r(\boldsymbol{\theta}) = \begin{bmatrix} - [\nabla_{\boldsymbol{\theta}} C_1(\boldsymbol{\theta})]^\top & - \\ - [\nabla_{\boldsymbol{\theta}} C_2(\boldsymbol{\theta})]^\top & - \\ \vdots & \\ - [\nabla_{\boldsymbol{\theta}} C_r(\boldsymbol{\theta})]^\top & - \end{bmatrix}. \quad (24)$$

Using this notation, the iterates in (23) read

$$\boldsymbol{\theta}_i^{(k+1)} = \boldsymbol{\theta}_i^{(k)} + \left[J_r(\boldsymbol{\theta}_i^{(k)}) \right]^+ \left(\mathbf{y}_i - \hat{\mathbf{y}}_i^{(k)} \right). \quad (25)$$

We stop the iterations once the relative error between the true observations, \mathbf{y}_i , and the SMS estimated observations $\hat{\mathbf{y}}_i^{(k)}$ falls below a user-defined tolerance δ ,

$$\frac{\|\hat{\mathbf{y}}_i^{(k)} - \mathbf{y}_i\|_2}{\|\mathbf{y}_i\|_2} < \delta. \quad (26)$$

The tolerance δ should be no less than the amount of observational noise in the data to avoid overfitting to noisy data. In addition, we also specify a maximum number of iterations at each data assimilation step to avoid infinite iterations. However, as we will see in Section 4, often only one Newton iteration is sufficient in practice.

In practice, the number of sensors or data points r is much smaller than the total number of parameters pN . As such, the nonlinear problem in (22) is underdetermined, and therefore its solution is not necessarily unique. Nonetheless, under certain conditions, underdetermined Newton iterations are guaranteed to converge to a solution [11, 27]. In particular, underdetermined Newton-like iterations (25) converge to a solution $\boldsymbol{\theta}_i^*$ under the following assumptions:

1. The observation operator $\mathbf{C}(\boldsymbol{\theta})$ is Lipschitz continuous in a neighborhood around $\boldsymbol{\theta}_i^*$, and
2. The Jacobian matrix $J_r(\boldsymbol{\theta}_i^*)$ has full row rank, i.e., $\text{rank}(J_r(\boldsymbol{\theta}_i^*)) = r$,

as long as the initial guess $\boldsymbol{\theta}_i^{(0)}$ is close enough to a solution $\boldsymbol{\theta}_i^*$; see Theorem 2.4.2 of Ref. [27].

If the Jacobian matrix J_r is ill-conditioned, the Newton-like iterations (25) will amplify the observational noise. To address this issue, we use a regularized version of the Newton-like iterations [34, 38]. Letting $\tilde{\gamma}$ be the regularization parameter, the iterates in (25) become,

$$\begin{aligned} \boldsymbol{\theta}_i^{(k+1)} &= \boldsymbol{\theta}_i^{(k)} + J_r(\boldsymbol{\theta}_i^{(k)})^\top \left(J_r(\boldsymbol{\theta}_i^{(k)}) J_r(\boldsymbol{\theta}_i^{(k)})^\top + \tilde{\gamma} I \right)^{-1} \left(\mathbf{y}_i - \hat{\mathbf{y}}_i^{(k)} \right) \\ &= \boldsymbol{\theta}_i^{(k)} + \tilde{J}_r(\boldsymbol{\theta}_i^{(k)})^+ \left(\mathbf{y}_i - \hat{\mathbf{y}}_i^{(k)} \right), \end{aligned} \quad (27)$$

where $\tilde{J}_r(\boldsymbol{\theta}_i^{(k)})^+ = J_r(\boldsymbol{\theta}_i^{(k)})^\top \left(J_r(\boldsymbol{\theta}_i^{(k)}) J_r(\boldsymbol{\theta}_i^{(k)})^\top + \tilde{\gamma} I \right)^{-1}$ is the regularized pseudo-inverse of $J_r(\boldsymbol{\theta}_i^{(k)})$.

At time t_m , the last time where observational data is available, we obtain a final set of corrected parameters $\boldsymbol{\theta}_m$. Since no more data is available, we use the shape-morphing equations to evolve from t_m to the final time t_f . We refer to this time interval as the *forecast window* $\mathcal{T}_f = [t_m, t_f]$. At time t_f , we use the set of parameters $\boldsymbol{\theta}_f$ to reconstruct the final forecast of the system $\hat{u}(\cdot, \boldsymbol{\theta}_f)$. We summarize the DA-SMS method in Algorithm 1.

Algorithm 1: Data assimilated shape-morphing solutions (DA-SMS) algorithm.

Inputs: Initial parameter values $\boldsymbol{\theta}_0$, Observation times $t_1 < t_2 < \dots < t_m$, Observations $\{\mathbf{y}_1, \dots, \mathbf{y}_m\}$, Maximum Newton iterations (maxits), Newton iteration tolerance δ , Final forecast time t_f

for $i = 1, 2, \dots, m$ **do**

$\boldsymbol{\theta}_i^{(0)} = \text{evolve}(\boldsymbol{\theta}_{i-1}, [t_{i-1}, t_i])$; ▷ Use SMS to predict from time t_{i-1} to t_i

$\hat{\mathbf{y}}_i^{(0)} = \mathbf{C}(\boldsymbol{\theta}_i^{(0)})$;

$error = \|\mathbf{y}_i - \hat{\mathbf{y}}_i^{(0)}\|_2 / \|\mathbf{y}_i\|_2$;

$k = 0$;

while ($error \geq \delta$ and $k \leq \text{maxits}$) **do**

$J_r = \nabla_{\boldsymbol{\theta}} \mathbf{C}(\boldsymbol{\theta}_{i+1}^{(k)})$;

$\boldsymbol{\theta}_i^{(k+1)} = \boldsymbol{\theta}_i^{(k)} + \tilde{J}_r \left(\boldsymbol{\theta}_i^{(k)} \right)^+ (\mathbf{y}_i - \hat{\mathbf{y}}_i^{(k)})$; ▷ Regularized Newton iteration

$\hat{\mathbf{y}}_i^{(k+1)} = \mathbf{C}(\boldsymbol{\theta}_i^{(k+1)})$;

$error = \|\mathbf{y}_i - \hat{\mathbf{y}}_i^{(k+1)}\|_2 / \|\mathbf{y}_i\|_2$;

$k = k + 1$;

end

$\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^{(k)}$;

end

$\boldsymbol{\theta}_f = \text{evolve}(\boldsymbol{\theta}_m, [t_m, t_f])$; ▷ Use SMS to forecast to final time t_f

Output: DA-SMS forecasted parameters $\boldsymbol{\theta}_f$

3.2 Convergence Analysis

Consider pointwise observations where $C_j[u(\cdot, t_i)] = u(\mathbf{x}_j, t_i)$ with \mathbf{x}_j being a sensor location. Convergence of the Newton iterations in Algorithm 1 ensures that the shape-morphing solution $\hat{u}(\cdot, \boldsymbol{\theta}_i)$ converges to the true solution $u(\cdot, t_i)$ at the sensor locations. However, this does not immediately guarantee convergence to the true solution at other points $\mathbf{x} \in \Omega$ where no sensors are available. In this section, we prove that, under certain conditions, the shape-morphing solution in fact converges uniformly to the true solution throughout the spatial domain Ω .

Since time is frozen during the Newton iterations, we omit the dependence of the true solution $u(\mathbf{x}, t_i)$ on time and simply write $u(\mathbf{x})$ throughout this section. In addition, we assume that $\Omega \subset \mathbb{R}^n$ is bounded. Consider r sensors which are located at $\{\mathbf{x}_1, \dots, \mathbf{x}_r\} \subset \Omega$, so that $u(\mathbf{x}_j)$ is known for all $j \in \{1, 2, \dots, r\}$. We define

$$\Delta = \sup_{\mathbf{x} \in \Omega} \inf_{1 \leq j \leq r} \|\mathbf{x} - \mathbf{x}_j\|, \quad (28)$$

which quantifies the maximum distance between any point in Ω and its closest sensor. More specifically, for any $\mathbf{x} \in \Omega$, the ball $B_\Delta(\mathbf{x})$, with radius Δ centered at \mathbf{x} , contains at least one sensor.

Assumption 1. We assume that the true solution u and the shape-morphing approximation \hat{u} have the following properties.

1. Both u and \hat{u} are Lipschitz continuous with respect to \mathbf{x} . We denote their smallest Lipschitz constants by L_u and $L_{\hat{u}}$ respectively.

2. There exists a sequence of parameters $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots\}$ such that

$$\lim_{k \rightarrow \infty} |\hat{u}(\mathbf{x}_j, \boldsymbol{\theta}^{(k)}) - u(\mathbf{x}_j)| = 0, \quad \forall j \in \{1, \dots, r\}. \quad (29)$$

These assumptions are not necessarily strong constraints on our approximate solution \hat{u} . Many choices of shape morphing solutions, such as Gaussian mixture models or neural networks, are Lipschitz continuous [21, 46]. Additionally, the second assumption simply ensures that our approximate solution \hat{u} is capable of approximating the unknown function u at the sensor locations. This assumption can be easily satisfied by choosing shape-morphing modes which satisfy the universal approximation theorem. In equation (29), we used an arbitrary convergent sequence of parameters $\boldsymbol{\theta}^{(k)}$. Although in this paper we obtain this sequence through Newton-like iterations (25), alternative methods can also be used.

The following lemma provides an upper bound for the pointwise error between the true solution u and the shape-morphing approximation \hat{u} .

Lemma 1. Let Assumption 1 hold. For any $\mathbf{x} \in \Omega$, let \hat{j} denote the integer such that $\mathbf{x}_{\hat{j}}$ is the closest sensor to point \mathbf{x} . Then for any parameter $\boldsymbol{\theta}$, we have

$$|\hat{u}(\mathbf{x}, \boldsymbol{\theta}) - u(\mathbf{x})| \leq (L_u + L_{\hat{u}}) \Delta + |\hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta}) - u(\mathbf{x}_{\hat{j}})|. \quad (30)$$

Proof. Using the triangle inequality, we have

$$\begin{aligned} |\hat{u}(\mathbf{x}, \boldsymbol{\theta}) - u(\mathbf{x})| &= |\hat{u}(\mathbf{x}, \boldsymbol{\theta}) - \hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta}) + \hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta}) - u(\mathbf{x}_{\hat{j}}) + u(\mathbf{x}_{\hat{j}}) - u(\mathbf{x})|, \\ &\leq |\hat{u}(\mathbf{x}, \boldsymbol{\theta}) - \hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta})| + |\hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta}) - u(\mathbf{x}_{\hat{j}})| + |u(\mathbf{x}_{\hat{j}}) - u(\mathbf{x})|, \\ &\leq (L_u + L_{\hat{u}}) \|\mathbf{x} - \mathbf{x}_{\hat{j}}\| + |\hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta}) - u(\mathbf{x}_{\hat{j}})|, \end{aligned}$$

where we used the Lipschitz continuity of the functions for the last inequality. Since $\mathbf{x}_{\hat{j}}$ is the closest sensor to \mathbf{x} , we have that $\|\mathbf{x} - \mathbf{x}_{\hat{j}}\| \leq \Delta$, which concludes the proof. \square

We now state our main result which shows that, if enough sensors are provided, the shape-morphing approximation \hat{u} becomes arbitrarily close to the true solution u in the uniform norm.

Theorem 1. Let Assumption 1 hold. Furthermore, for any $\epsilon > 0$, assume that there are enough sensors distributed such that $\Delta < \epsilon/2(L_u + L_{\hat{u}})$. Then there exists $K \in \mathbb{N}$ such that for any $k > K$,

$$\sup_{\mathbf{x} \in \Omega} |\hat{u}(\mathbf{x}, \boldsymbol{\theta}^{(k)}) - u(\mathbf{x})| < \epsilon. \quad (31)$$

Proof. Consider the upper bound in Lemma 1. Equation (29) in Assumption 1 implies that $\lim_{k \rightarrow \infty} |\hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta}^{(k)}) - u(\mathbf{x}_{\hat{j}})| = 0$. Therefore, there exists $K \in \mathbb{N}$, such that for any $k > K$ we have $|\hat{u}(\mathbf{x}_{\hat{j}}, \boldsymbol{\theta}^{(k)}) - u(\mathbf{x}_{\hat{j}})| < \epsilon/2$. Moreover, since Lemma 1 holds for any $\boldsymbol{\theta}$, it also holds for the sequence $\{\boldsymbol{\theta}^{(k)}\}_{k \geq 1}$. Using this fact, Lemma 1, and the fact that $\Delta < \epsilon/2(L_u + L_{\hat{u}})$, we have

$$|\hat{u}(\mathbf{x}, \boldsymbol{\theta}^{(k)}) - u(\mathbf{x})| < \epsilon, \quad (32)$$

for any $\mathbf{x} \in \Omega$ and $k > K$. \square

This result yields both an indication of the number of required sensors as well as their distribution throughout the domain Ω . However, it is important to note that these bounds are rather pessimistic. As we show in Section 4, a relatively smaller number of sensors, uniformly distributed over the spatial domain, is often sufficient in practice.

3.3 Continuous-time Data Assimilation

The algorithm developed in Section 3.1 considers data obtained at discrete time instances, which is the situation one encounters in practice. Here, for completeness, we also introduce a data assimilation method for shape-morphing modes in case observational data are available in continuous time. To begin, we consider the instantaneous optimization problem (6) from which we derived the shape-morphing equation (7). The extension of our framework to the collocation shape-morphing equation (15) is straightforward, as shown in Appendix A.

Recall from Section 2 that we seek to minimize the residual function $R(\cdot, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ which quantifies the discrepancy between the rate of change \hat{u}_t of the approximate solution and the right-hand side of the PDE $F(\hat{u})$. In this case we assume that we have access to a continuum of data $\mathbf{y}(t)$, and that we can compute its time derivative $\dot{\mathbf{y}}$. To assimilate this data, we require that the time derivative of the SMS estimated observations $\hat{\mathbf{y}}$ are equal to $\dot{\mathbf{y}}$, i.e. $\dot{\hat{\mathbf{y}}} = \dot{\mathbf{y}}$. In terms of the observation operators C_j , we require,

$$\dot{\hat{y}}_j = \frac{d}{dt}C_j(\boldsymbol{\theta}) = \langle \nabla_{\boldsymbol{\theta}}[C_j(\boldsymbol{\theta})], \dot{\boldsymbol{\theta}} \rangle = \dot{y}_j, \quad \forall j \in \{1, \dots, r\}, \quad (33)$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and we used the fact that the observation operator C_j has no explicit time dependence. Using the definition of J_r from (24), we write equation (33) more compactly as

$$J_r(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\mathbf{y}}. \quad (34)$$

To arrive at the *continuous-time DA-SMS equation*, we constrain the optimization problem (6) so that (34) is satisfied. Namely, we seek the solution to the constrained optimization problem,

$$\begin{aligned} \min_{\dot{\boldsymbol{\theta}}} \quad & \frac{1}{2} \|R(\cdot, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|_{\mathcal{H}}^2, \\ \text{s.t.} \quad & J_r(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\mathbf{y}}. \end{aligned} \quad (35)$$

As we show in Appendix A, if the metric tensor M is non-singular and J_r is full-rank, the solution to the constrained optimization problem (35) is given by

$$M\dot{\boldsymbol{\theta}} = \left(I - J_r^\top (J_r M^{-1} J_r^\top)^{-1} J_r M^{-1} \right) \mathbf{f} + J_r^\top (J_r M^{-1} J_r^\top)^{-1} \dot{\mathbf{y}}, \quad (36)$$

where we have suppressed the dependence of M , \mathbf{f} , and J_r on $\boldsymbol{\theta}$. The solution to the system of ODEs in (36) ensures that $\dot{\hat{\mathbf{y}}}(t) = \dot{\mathbf{y}}(t)$ for all times t .

We emphasize that, although this methodology can be useful theoretically, it is not necessarily suited for practical applications. First, all real-world observational data are polluted by noise; numerical computations of the time derivative will amplify this noise. In addition, if there is an initial difference in the SMS approximate observations, $\hat{\mathbf{y}}(0) = \mathbf{y}(0) + c$, then this difference will persist in the solution of (36). Due to these issues, we do not recommend this methodology for most practical applications, and instead suggest using the discrete DA-SMS algorithm presented in Section 3.1.

4 Numerical Results

We present three numerical examples: the nonlinear Schrödinger (NLS) equation, the Kuramoto-Sivashinsky (KS) equation, and the advection-diffusion (AD) equation. For each example, we compare direct numerical simulations (DNS) to approximate solutions obtained from SMS (without data

assimilation) and data assimilated SMS. For NLS, we take a reduced-order modeling approach, using one Gaussian mode to approximate the dynamics of the PDE. We incorporate measurements of the envelope of the solution of NLS into the DA-SMS solution. For KS and AD equations, we use shallow neural networks as our approximate solution and take pointwise measurements of the state of the system for DA-SMS. In each case, we add 5% noise to the data to mimic real-world application where observational data is noisy.

4.1 Nonlinear Schrödinger Equation

For our first example, we consider the nonlinear Schrödinger equation. NLS models unidirectional, slowly modulating surface waves in deep fluids [2, 13, 18, 47]. Denoting the surface elevation by $\tilde{\eta}(\tilde{x}, \tilde{t})$, we consider perturbations of a sinusoidal carrier wave,

$$\tilde{\eta}(\tilde{x}, \tilde{t}) = \text{Re} \left\{ \tilde{u}(\tilde{x}, \tilde{t}) \exp \left[\hat{i} k_0 \tilde{x} - \omega_0 \tilde{t} \right] \right\}, \quad (37)$$

where $\hat{i} = \sqrt{-1}$, $\tilde{u}(\tilde{x}, \tilde{t}) \in \mathbb{C}$ is the wave envelope, k_0 is the wave number of the carrier wave, and ω_0 is its angular frequency. In dimensional variables, NLS reads [47]

$$\frac{\partial \tilde{u}}{\partial \tilde{t}} = -\hat{i} \frac{w_0}{8k_0^2} \frac{\partial^2 \tilde{u}}{\partial \tilde{x}^2} - \hat{i} \frac{\omega_0 k_0^2}{2} |\tilde{u}|^2 \tilde{u}, \quad \tilde{u}(x, 0) = \tilde{u}_0(x). \quad (38)$$

We introduce the non-dimensional variables $x = 2\sqrt{2}k_0\tilde{x}$, $t = -\omega_0\tilde{t}$, and $u = (k_0/\sqrt{2})\tilde{u}$ as in [1], so that Eq. (38) becomes

$$\frac{\partial u}{\partial t} = \hat{i} \frac{\partial^2 u}{\partial x^2} + \hat{i} |u|^2 u, \quad u(x, 0) = u_0(x). \quad (39)$$

The associated boundary conditions for NLS are that the solution vanishes as $|x|$ approaches infinity.

We solve (39) using a Fourier pseudo-spectral truncation in space. In addition, we use an exponential time-differencing scheme as described in Ref. [14] with a time step of $\Delta t = 0.025$ as in Refs. [2, 13]. We discretize the spatial domain $\Omega = [-L/2, L/2]$ with length $L = 256\sqrt{2}\pi$, using 2^{11} Fourier modes with periodic boundary conditions. Our solutions are localized around $x = 0$ and the domain is large enough to mimic the infinite domain size in NLS.

For SMS, our approximate solution takes the form,

$$\hat{u}(x, \boldsymbol{\theta}(t)) = A(t) \exp \left[-\frac{x^2}{L^2(t)} + \hat{i} \frac{V(t)}{L(t)} x^2 + \hat{i} \varphi(t) \right], \quad (40)$$

in accordance with Refs. [1, 2, 39]. The corresponding shape parameters are given by $\boldsymbol{\theta}(t) = \{A(t), L(t), V(t), \varphi(t)\}$, which control the amplitude A , length scale L , speed V and phase φ of the SMS solution. We evolve these parameters according to the SMS equation (7) which reads

$$\dot{A} = -\frac{2AV}{L}, \quad \dot{L} = 4V, \quad \dot{V} = \frac{4}{L^3} - \frac{A^2}{\sqrt{2}L}, \quad \dot{\varphi} = \frac{5A^2}{4\sqrt{2}} - \frac{2}{L^2}. \quad (41)$$

These equations are integrated in time using Matlab's fourth-order Runge–Kutta scheme `ode45`. At the initial time, the parameters are $A_0 = 0.2$, $L_0 = 20$, $V_0 = 0$, and $\varphi_0 = 0$. For DNS we pick $u_0(x, 0) = \hat{u}(x, \boldsymbol{\theta}(0))$, so that SMS and DNS solutions coincide initially.

For DA-SMS, at every $\Delta t = 0.5$ time units, we take pointwise measurements of the modulus of the envelope of the DNS solution at three sensor locations, $x_1 = 0$, $x_2 = 5$, and $x_3 = -10$. In

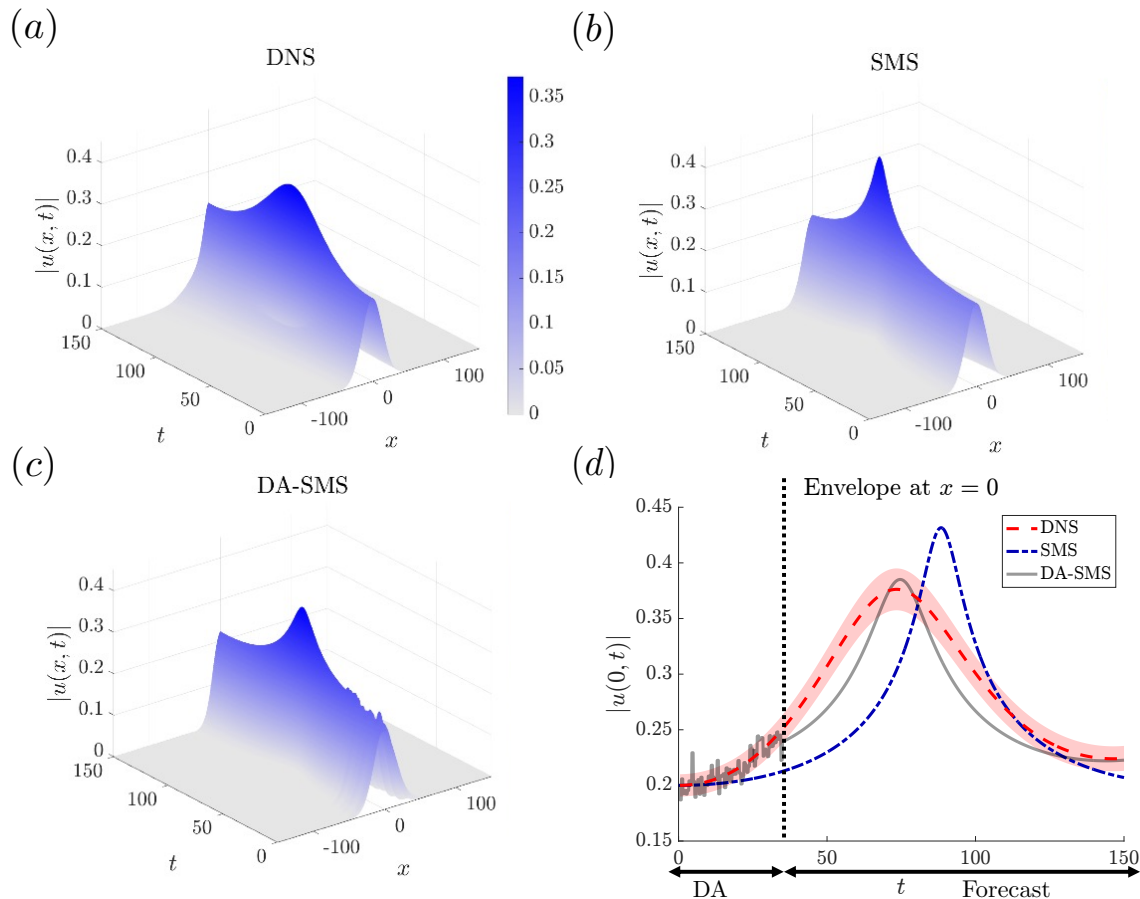


Figure 2: Here we compare the modulus of the envelope $u(x, t)$ over the time interval $[0, 150]$ using the initial parameters $A_0 = 0.2$, $L_0 = 20$, $V_0 = 0$, $\phi_0 = 0$. We show (a.) DNS, (b.) SMS, (c.) DA-SMS. In (d.) we compare the envelopes of the three solutions at $x = 0$, where DNS corresponds to the dashed red line, DA-SMS is the solid grey line, and SMS is the dot dashed blue line. The shaded red region denotes one standard deviation (5%) away from the DNS solution.

other words, we have three pointwise observations $y_j(t_i) = |u(x_j, t_i)|$ at times $t_i \in \{0.5, 1, 1.5, \dots\}$. Since the solution is symmetric around $x = 0$, the second and third sensor locations, $\{x_2, x_3\}$, are chosen asymmetrically to avoid redundant data. The data-assimilation window is $\mathcal{T}_{DA} = [0, 35]$, and the forecast window is $\mathcal{T}_f = [35, 150]$. We incorporate this data into the DA-SMS solution using Algorithm 1.

Figure 2 compares the solutions for DNS, DA-SMS, and SMS, as well as the amplitudes of the solutions at $x = 0$. For the chosen combination of initial amplitude A_0 and length scale L_0 , we know that this wave exhibits a focusing behavior, meaning it increases in amplitude and decreases in width [13]. This focusing behavior is often used to model rogue waves on the ocean surface [10].

As shown in Figure 2(b,d), SMS correctly predicts that the wave will focus, but it overestimates the maximum amplitude. In addition, the maximum occurs with a time delay compared to the

true DNS solution. In contrast, DA-SMS not only predicts the focusing of the wave, it also more accurately predicts the maximum amplitude. Moreover, DA-SMS correctly predicts the time at which the maximum wave height is attained; see Figure 2(c,d). We note that the DA-SMS solution over the DA time window, $\mathcal{T}_{DA} = [0, 35]$, appears jagged because of the 5% noise added to the observational data.

4.2 Kuramoto–Sivashinsky Equation

For our second example, we consider the Kuramoto–Sivashinsky (KS) equation which models thermal instabilities in laminar flame fronts [30, 43]. Here we approximate the solution to KS using a shallow neural network as the shape-morphing solution and compare the results to the DNS solution obtained from a Fourier pseudo-spectral method. KS equation is a fourth-order PDE,

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}, \quad u(x, 0) = u_0(x), \quad (42)$$

with periodic boundary conditions on the domain $\Omega = [-L/2, L/2]$. Under certain conditions KS is known to exhibit chaotic behavior [30, 43]. For our study, we choose $L = 22$ and define the function,

$$u_0^*(x) = \sin\left(\frac{2\pi}{L}x\right) + \sum_{i=2}^4 \sin\left(\frac{2i\pi}{L}x\right) + \cos\left(\frac{2i\pi}{L}x\right). \quad (43)$$

To ensure a reasonable, initial maximum amplitude, we rescale u_0^* and define the initial condition,

$$u_0(x) = \frac{u_0^*(x)}{\max_{x \in \Omega} |u_0^*(x)|}. \quad (44)$$

For DNS, we use a Fourier pseudo-spectral truncation using 2^7 Fourier modes. We integrate in time using Matlab’s fourth-order Runge–Kutta scheme `ode45`.

On the other hand, for SMS we use a shallow neural network with a hyperbolic-tangent activation function in keeping with Anderson and Farazmand [3]. To enforce periodic boundary conditions, we use the methodology outlined in Ref. [15], where the shape-morphing solution is given by

$$\begin{aligned} \hat{u}(x, \boldsymbol{\theta}(t)) &= \sum_{i=1}^N a_i(t) \tanh(w_i(t) s_i(x, c_i(t)) + b_i(t)), \\ s_i(x, c_i(t)) &= \sin\left(\frac{2\pi}{L}x + c_i(t)\right). \end{aligned} \quad (45)$$

In this case, the shape parameters, $\boldsymbol{\theta}(t)$, are the N amplitudes $a_i(t)$, weights $w_i(t)$, and biases $\{b_i(t), c_i(t)\}$, so that $\boldsymbol{\theta} = \{a_i(t), w_i(t), b_i(t), c_i(t)\}_{i=1}^N$. Since symbolic computation is not possible in this case, we use the collocation method outlined in Section 2 to evolve the parameters in time. We use 2^7 equidistant collocation points and a Tikhonov regularization parameter $\gamma = 10^{-3}$ to counter the stiffness of the metric tensor. For the time stepping of the SMS equation, we again use Matlab’s `ode45`. To find the initial parameters $\boldsymbol{\theta}_0$, we solve the nonlinear least squares problem,

$$\min_{\boldsymbol{\theta}_0 \in \mathbb{R}^{4N}} \|\hat{u}(\cdot, \boldsymbol{\theta}_0) - u_0\|_{\mathcal{L}^2}^2, \quad (46)$$

using midpoint quadrature to discretize the integral in the \mathcal{L}^2 norm. For our study, we chose $N = 10$ modes, and found initial parameters θ_0 which correspond to an initial relative error less than 0.1%.

For DA-SMS, we take pointwise measurements of the state of the system at $r = 10$ locations as our observational data: $y_j(t) = u(x_j, t)$ for $j = 1, 2, \dots, r$. The sensors are equispaced throughout the domain, so that $x_j = -\frac{L}{2} + (j-1)\frac{L}{r}$. We incorporate the data into the shape-morphing solution at every $\Delta t = 2.0$ time units, using the DA-SMS algorithm presented in Section 3.1. We assume we have data for the first 30 time units, so that $\mathcal{T}_{DA} = [0, 30]$. We then forecast over the interval $\mathcal{T}_f = [30, 100]$. For KS, we set the Tikhonov regularization parameter $\tilde{\gamma} = 10^{-3}$ for DA, which is the same as the regularization parameter used in the collocation shape-morphing equation. We noticed that excellent agreement with observations are achieved with even a single Newton-like iteration. Therefore, to achieve computational speed up, we only take one Newton-like iteration in each step of Algorithm 1.

Figure 3 compares DA-SMS results to both DNS and SMS without data assimilation. For DA-SMS, we consider two types of observational data: measurements without noise and noisy measurements where 5% Gaussian noise is added to the observations y_j . As shown in Figure 3 (d,g), during the first 20 time units, the SMS solution closely matches the overall dynamics of the DNS solution. Soon after, however, the errors incurred by SMS move the trajectory of the approximate solution away from the true solution. More specifically, without data assimilation, the errors for SMS steadily grow from their minimum around 10^{-3} , reach approximately one around $t = 20$, and steadily grow for the remainder of the simulation. This growth of errors over time is expected in a chaotic system.

On the other hand, DA-SMS which incorporates data over the DA time window $t \in [0, 30]$ keeps the approximate solution closer to the truth for a longer period of time. When there is no noise in the data, as shown in figure 3(b,e), the errors over both DA time window and the forecast window remain near zero. Even when 5% error is added to observations (panels c and f), the error remains low for up to $t = 75$. Eventually, as expected for a chaotic system, those errors also grow over the forecast time window where no data is available. However, compared to SMS without data assimilation, DA-SMS significantly extends the predictability horizon.

4.3 Advection-Diffusion Equation

For our final example, we consider the passive advection-diffusion of temperature in a time-dependent fluid flow. Consider the temperature $\tilde{T}(\tilde{x}, \tilde{z}, \tilde{t})$ of the fluid in a rectangular domain $\Omega = [0, L] \times [0, H]$; see figure 4. At the horizontal boundaries, $z = 0$ and $z = H$, the temperature is kept constant, so that $\tilde{T}(\tilde{x}, 0, \tilde{t}) = \tilde{T}_{bottom}$ and $\tilde{T}(\tilde{x}, H, \tilde{t}) = \tilde{T}_{top}$, where $\tilde{T}_{bottom} > \tilde{T}_{top}$. For the left and right sides of the domain, the box is perfectly insulated so that no heat can escape in the x direction, leading to the boundary conditions $\partial_{\tilde{x}}\tilde{T}(0, \tilde{z}, \tilde{t}) = 0 = \partial_{\tilde{x}}\tilde{T}(L, \tilde{z}, \tilde{t})$. The fluid is driven by a prescribed velocity field $\tilde{\mathbf{v}}(\tilde{x}, \tilde{z}, \tilde{t})$. We denote the horizontal and vertical components of the velocity by $\tilde{v}_1(\tilde{x}, \tilde{z}, \tilde{t})$ and $\tilde{v}_2(\tilde{x}, \tilde{z}, \tilde{t})$, respectively.

In dimensional variables, the temperature evolves according to the advection-diffusion equation,

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} + \tilde{\mathbf{v}} \cdot \nabla \tilde{T} = \tilde{\kappa} \Delta \tilde{T}, \quad (47)$$

where $\tilde{\kappa}$ is the thermal diffusivity of the fluid. If the fluid velocity is zero, $\tilde{\mathbf{v}} \equiv 0$, heat transfer occurs only by conduction, and the temperature profile of the system varies linearly with respect

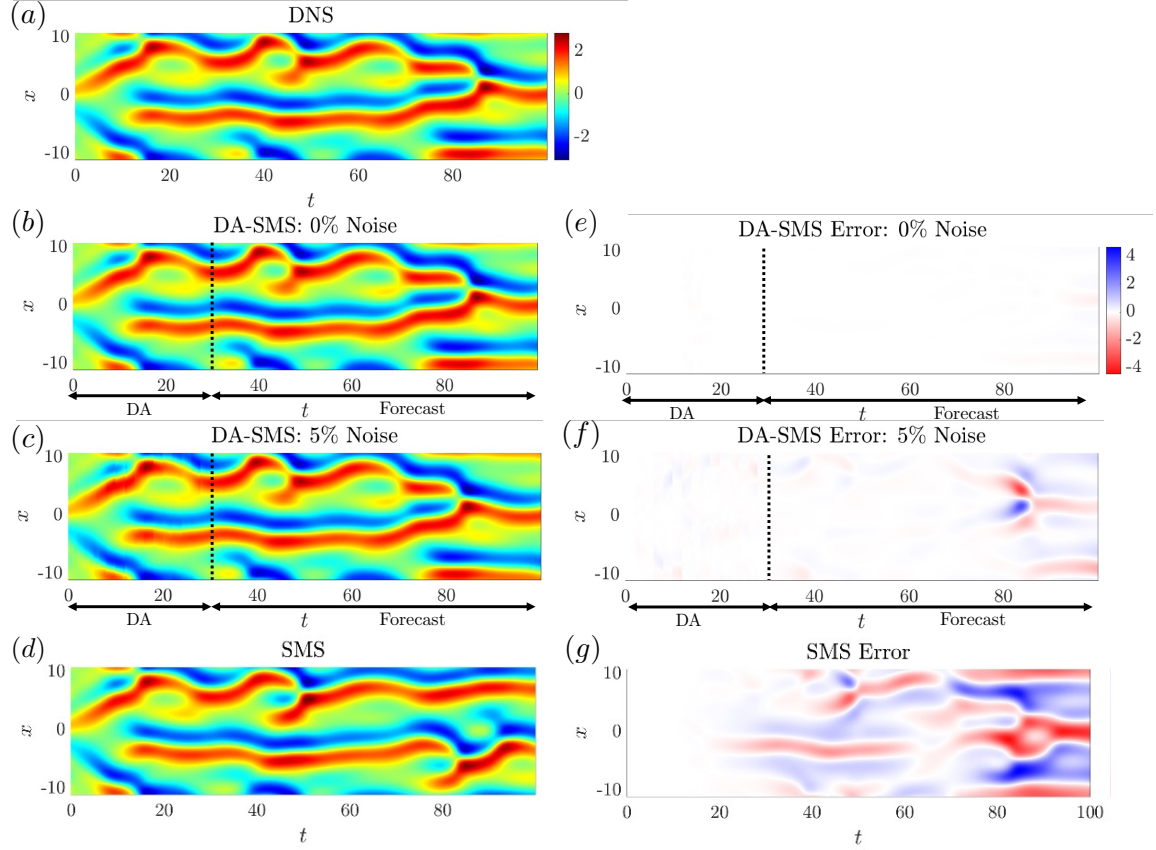


Figure 3: Comparisons of the state of the system over the entire spatial domain $\Omega = [-11, 11]$ and the entire duration of the simulation $t \in [0, 100]$. The column on the left shows the results for: (a) DNS, (b) DA-SMS with clean (0% noise) data, (c) DA-SMS with 5% noise in the data, and (d) SMS with no data assimilation. In the column on the right we present the corresponding error defined by: $u_{DNS}(x, t) - \hat{u}(x, t)$ for (e) DA-SMS with clean data, (f) DA-SMS with noisy data, (g) SMS without data assimilation.

to \tilde{z} . More specifically, the conductive temperature profile \tilde{T}_c is given by,

$$\tilde{T}_c(\tilde{z}) = \tilde{T}_{bottom} - \frac{\Delta\tilde{T}}{H}\tilde{z}, \quad (48)$$

where $\Delta\tilde{T} = \tilde{T}_{bottom} - \tilde{T}_{top}$. If the fluid velocity is nonzero, the temperature will fluctuate away from the linear profile \tilde{T}_c . Thus it is convenient to instead study the evolution of the fluctuations $\tilde{u}(\tilde{x}, \tilde{z}, \tilde{t}) = \tilde{T}(\tilde{x}, \tilde{z}, \tilde{t}) - \tilde{T}_c(\tilde{z})$. Substituting this expression into (47), we obtain the following PDE for the fluctuations $\tilde{u}(\tilde{x}, \tilde{z}, \tilde{t})$,

$$\frac{\partial\tilde{u}}{\partial\tilde{t}} + \tilde{\mathbf{v}} \cdot \nabla\tilde{u} = \frac{\Delta\tilde{T}}{H}\tilde{v}_2 + \tilde{\kappa}\Delta\tilde{u}. \quad (49)$$

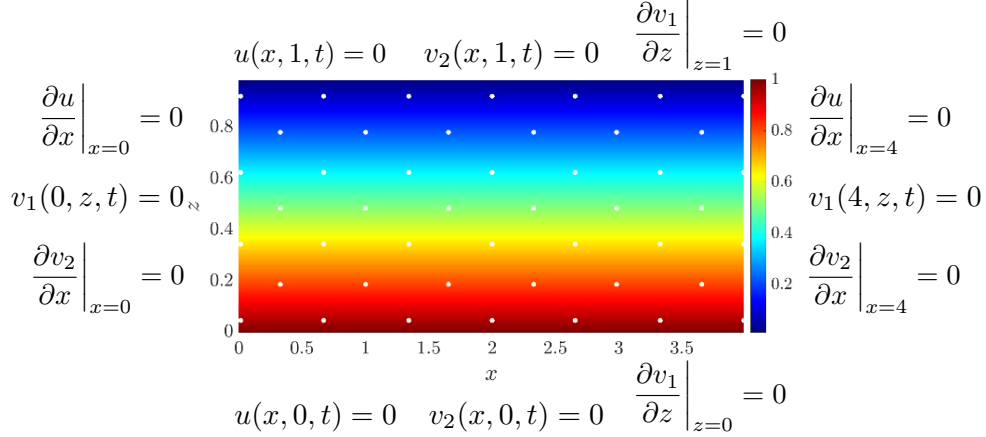


Figure 4: Schematic illustration of the boundary conditions for the advection-diffusion equation in dimensionless variables. The domain size is $[0, 4] \times [0, 1]$. The white dots indicate the sensor locations for DA-SMS.

We introduce the dimensionless variables,

$$\mathbf{x} = \frac{\tilde{\mathbf{x}}}{H}, \quad T = \frac{\tilde{T}}{\Delta\tilde{T}}, \quad t = \frac{\tilde{t}v_f}{H}, \quad \text{and} \quad \tilde{\mathbf{v}} = \frac{\mathbf{v}}{v_f}, \quad (50)$$

where $\mathbf{x} = (x, z)$, and v_f is the free-fall velocity given by $v_f = \sqrt{\beta g \Delta\tilde{T} H}$ where β and g are the thermal expansion of the fluid and the gravitational acceleration, respectively. In these dimensionless variables, the governing equation for the temperature fluctuation is given by

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = v_2 + \kappa \Delta u, \quad (51)$$

where $\kappa = \tilde{\kappa} / (Hv_f)$. The corresponding boundary conditions for (51) are homogeneous Neumann in x and homogeneous Dirichlet in z , as shown in figure 4.

Next, we specify the fluid velocity field. In order to be physically consistent, we must ensure that the velocity satisfies the appropriate boundary conditions. Namely, we require that there are no slip conditions on the horizontal component, v_1 , at the left and right boundaries and no-flux conditions in the z direction on the top and bottom. Additionally, the vertical component of the velocity v_2 should have no slip conditions on the top and bottom of the domain, and no-flux conditions in the x direction on the left and right of the domain. In summary, the fluid velocity field must satisfy the following boundary conditions,

$$\begin{aligned} v_1(x, z, t) \Big|_{x=0,L} &= 0, & \frac{\partial v_1}{\partial z} \Big|_{z=0,H} &= 0, \\ v_2(x, z, t) \Big|_{z=0,H} &= 0, & \frac{\partial v_2}{\partial x} \Big|_{x=0,L} &= 0. \end{aligned} \quad (52)$$

We summarize the boundary conditions for the temperature fluctuations u and the fluid velocity \mathbf{v} in figure 4. Furthermore, we require $\nabla \cdot \mathbf{v} = 0$, ensuring that the fluid flow is incompressible.

To attain these requirements on \mathbf{v} , we define the velocity field,

$$\mathbf{v}(x, z, t) = \pi A \begin{pmatrix} -\sin(\pi f(x, t)) \cos(\pi z) \\ \cos(\pi f(x, t)) \sin(\pi z) \frac{\partial f}{\partial x} \end{pmatrix} \quad (53)$$

where

$$f(x, t) = m \frac{x}{L} + \epsilon L^4 \sin(\omega t) \left[\frac{x}{L} - 2 \left(\frac{x}{L} \right)^3 + \left(\frac{x}{L} \right)^4 \right]. \quad (54)$$

This velocity field, which satisfies the boundary conditions (52), is a modified version of double gyre flow that is often used in oceanography [17, 42]. The parameter m roughly determines the number of gyres (or vortices) in the x direction. The parameter ω is the frequency of gyre oscillations and the small parameter ϵ determines the amplitude of these oscillations. For our simulations we set $m = 2$, $A = 0.1$, $\epsilon = 0.025$, and $\omega = \pi$. For the spatial domain, we set the dimensionless width of the domain $L = 4$ and the height $H = 1$. We set the temperatures at the bottom and top of the domain as $T_{bottom} = 1$ and $T_{top} = 0$, respectively, and we set $\kappa = 10^{-3}$ for the dimensionless thermal diffusivity. Lastly, we choose the initial condition,

$$u_0(x, z) = 10^{-1} \sin\left(\frac{\pi}{L}x\right) \cos\left(\frac{\pi}{H}z\right).$$

For DNS, we use a Fourier pseudo-spectral method to solve equation (51), using 2^8 modes in the x direction, and 2^6 modes in the z direction. The DNS solution is shown in figure 5(a). In order to visualize the space-time evolution of the solution, we only show the horizontal slice at $z = 0.9$.

For the shape-morphing solution \hat{u} , we use a shallow neural network of width $N = 100$ as described in Appendix B. In solving PDEs with neural networks, the boundary conditions are often approximately enforced by adding them to the loss function as a penalty term [36]. Here, in contrast, we construct our neural network such that the boundary conditions are explicitly built into the network and therefore are exactly satisfied for any set of network parameters θ . The details of our construction can be found in Appendix B.

The evolution of network parameters $\theta(t)$ is determined by solving the SMS equation (19) with Tikhonov regularization parameter $\gamma = 5 \times 10^{-2}$ and using $2^6 \times 2^4$ collocation points evenly spaced throughout the domain $[0, 4] \times [0, 1]$. To initialize the SMS equation, we solve the nonlinear optimization problem, $\min_{\theta_0 \in \mathbb{R}^{6N}} \|\hat{u}(\cdot, \theta_0) - u_0\|_{\mathcal{L}^2}^2$, to obtain the initial condition θ_0 . The resulting approximate initial condition $\hat{u}(\cdot, \theta_0)$ matches the true initial condition u_0 within 0.1% error. Lastly, the temporal integration of the SMS equation is carried out using `ode45`. The resulting SMS solution without data assimilation is shown in figure 5 (d,h).

Now we turn our attention to DA-SMS, where the observables are pointwise measurements $y_i(t) = u(x_i, z_i, t)$. We place $r = 46$ sensors on a scattered grid as shown in figure 4. The observations are gathered every $\Delta t = 0.5$ time units over the data assimilation window $\mathcal{T}_{DA} = [0, 25]$. The forecast window is $\mathcal{T}_f = [25, 45]$. For the Newton iterations, we set the Tikhonov regularization parameter to $\tilde{\gamma} = 5 \times 10^{-2}$, which is equal to the Tikhonov regularization parameter for SMS. Similar to the KS equation, we observed that even one Newton iteration suffices to obtain satisfactory agreement with observations.

Figure 5 compares the results of DA-SMS and SMS without data assimilation. Panel (e) shows the relative \mathcal{L}^2 error,

$$\mathcal{E}(t) = \frac{\|u(\cdot, t) - \hat{u}(\cdot, \theta(t))\|_{\mathcal{L}^2}}{\|u(\cdot, t)\|_{\mathcal{L}^2}}, \quad (55)$$

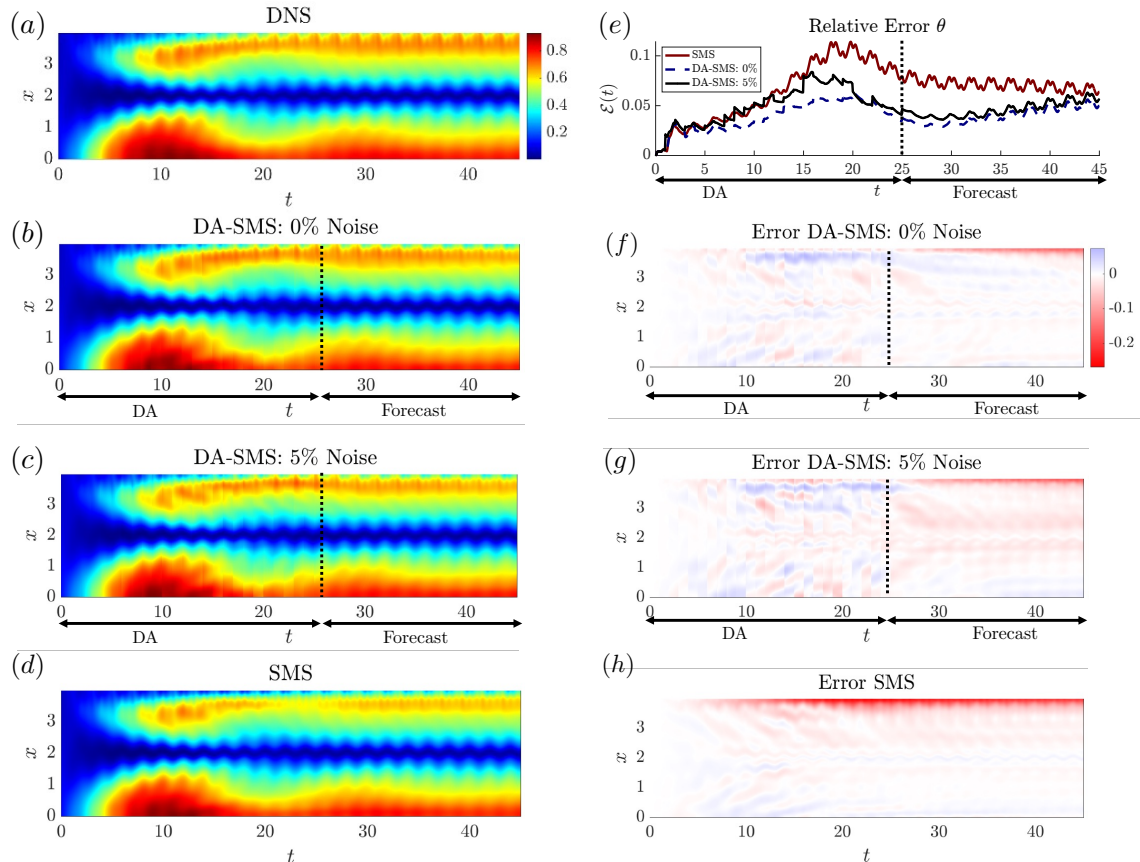


Figure 5: Space-time plots of the solutions to ADE (51) where we have taken one slice in space located at $z = 0.9$ at every time instance over the interval $[0, 45]$. The left column shows the results obtained from (a) DNS, (b) DA-SMS with 0% noise in the data, (c) DA-SMS with 5% noise in the data, and (d) SMS without data assimilation. Panels (f,g,h) show the corresponding difference $u_{DNS}(x, 0.9, t) - \hat{u}(x, 0.9, \theta(t))$. Panel (e) shows the total relative errors.

for each simulation. Although the SMS solution captures the overall dynamics, its relative error increases to about 12% over time. Recall that the boundary conditions are exactly enforced in the construction of our neural network, so that the error is exactly zero at the boundary. However, immediately near the boundary we observe a large accumulation of error for the SMS solution. This is particularly pronounced near $x = 4$. On the other hand, the DA-SMS is able to suppress these errors significantly. For DA-SMS with clean data (no noise added to the observations), the total relative error is at most 5%. When noise is added to the observational data, the DA-SMS error increases slightly to at most 7%. However, this error is still smaller than the SMS simulations without data assimilation.

5 Conclusions

Shape-morphing solutions are a computational method for obtaining numerical solutions to PDEs. In spite of the rapidly growing literature on this subject, data assimilation with SMS had remained unaddressed. Here, we developed two data assimilation methods for shape-morphing solutions: (i) Discrete-time data assimilated SMS and (ii) Continuous-time data assimilated SMS. Both methods seek to use observations to correct for modeling error, approximation error, and uncertainties in the initial condition.

Our discrete method takes the form of a predictor-corrector scheme. During the prediction phase, the SMS equations are used to evolve the solution between consecutive observation points. At time instances where observations are available, a Newton-like iteration is used to correct for the discrepancy between the SMS and the observational data. We proved that, under certain conditions, the DA-SMS method converges uniformly to the true solution.

Our continuous-time method takes a different approach. It treats the observational data as constraints to the optimization problem underlying SMS equations. The resulting constrained SMS equation ensures that the solution agrees with the observational data up to a constant which is independent of time.

Several open questions remain to be addressed. These include:

(i) Better convergence analysis: Our theoretical results concerning the convergence of the discrete-time DA-SMS are rather pessimistic. They indicate that a dense set of observational data is required for the method to converge. In contrast, our numerical results indicate that the method works with relatively few observations and a single Newton-like iteration. Thus, our numerical results suggest that tighter error bounds for the DA-SMS error are feasible.

(ii) Sensor placement: It is well-known that the quality of data assimilation methods depend significantly on the location of the sensors which gather the data [7, 16]. Here, we placed our sensors in an ad hoc manner. Specialized sensor placement methods, tailored for use with shape-morphing solutions, are desirable.

(iii) Other data assimilation techniques: Classical data assimilation methods, such as extended Kalman filtering and variational data assimilation, can in principle be used with SMS. However, the nonlinear dependence of SMS on its parameters $\theta(t)$ complicates a straightforward implementation of these methods. Future work should explore the necessary modifications to existing DA methods for use with SMS. Furthermore, the nonlinearities necessitate a separate convergence analysis for existing DA methods used in conjunction with SMS.

Acknowledgments

This work was supported by the National Science Foundation under the grants DMS-1745654 (Research Training Group), DMS-2208541 (Computational Mathematics Program), and DMS-2220548 (Algorithms for Threat Detection).

A Derivation of Continuous-time Data Assimilation Equations

Here we present a detailed derivation of the results presented in Section 3.3 for both types of shape-morphing equation. To begin, we consider the optimization problem

$$\begin{aligned} \min_{\dot{\boldsymbol{\theta}}} \quad & \frac{1}{2} \|R(\mathbf{x}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|_{\mathcal{H}}^2 + \frac{\gamma}{2} \|\dot{\boldsymbol{\theta}}\|_2^2 \\ \text{s.t.} \quad & J_r \dot{\boldsymbol{\theta}} = \dot{\mathbf{y}}. \end{aligned}$$

To solve the constrained optimization problem we introduce the Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^r$. To this end, we construct the constrained cost functional

$$\mathcal{J}_c(\boldsymbol{\theta}; \dot{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \frac{1}{2} \|R(\mathbf{x}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|_{\mathcal{H}}^2 + \boldsymbol{\lambda}^\top (J_r \dot{\boldsymbol{\theta}} - \dot{\mathbf{y}}) + \frac{\gamma}{2} \|\dot{\boldsymbol{\theta}}\|_2^2 \quad (56)$$

Next, we expand the Hilbert norm in terms of the Hilbert inner product, which yields

$$\begin{aligned} \mathcal{J}_c(\boldsymbol{\theta}; \dot{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = & \frac{1}{2} \left\langle \sum_{i=1}^N \frac{\partial \hat{u}}{\partial \theta_i} \dot{\theta}_i, \sum_{j=1}^N \frac{\partial \hat{u}}{\partial \theta_j} \dot{\theta}_j \right\rangle_{\mathcal{H}} - \left\langle \sum_{i=1}^N \frac{\partial \hat{u}}{\partial \theta_i} \dot{\theta}_i, F(\hat{u}) \right\rangle_{\mathcal{H}} \\ & - \frac{1}{2} \|F(\hat{u})\|_{\mathcal{H}}^2 + \boldsymbol{\lambda}^\top (J_r \dot{\boldsymbol{\theta}} - \dot{\mathbf{y}}) + \frac{\gamma}{2} \|\dot{\boldsymbol{\theta}}\|_2^2. \end{aligned} \quad (57)$$

Next, following Ref. [1], we introduce the metric tensor M and the vector \mathbf{f} which we describe in Section 2, so we can re-express (57) as

$$\mathcal{J}_c(\boldsymbol{\theta}; \dot{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \frac{1}{2} \dot{\boldsymbol{\theta}}^\top M \dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}^\top \mathbf{f} + \|F(\hat{u})\|_{\mathcal{H}}^2 + \boldsymbol{\lambda}^\top (J_r \dot{\boldsymbol{\theta}} - \dot{\mathbf{y}}). \quad (58)$$

To follow, we compute the gradient with respect to $\boldsymbol{\theta}$ and set equal it to zero to solve for $\dot{\boldsymbol{\theta}}$.

$$\nabla_{\boldsymbol{\theta}} \mathcal{J}_c = (M + \gamma I) \dot{\boldsymbol{\theta}} - \mathbf{f} + J_r^\top \boldsymbol{\lambda} \quad (59)$$

By construction M is symmetric positive semidefinite; because $\gamma > 0$, $(M + \gamma I)$ is invertible, thus

$$\dot{\boldsymbol{\theta}} = (M + \gamma I)^{-1} (\mathbf{f} - J_r^\top \boldsymbol{\lambda}). \quad (60)$$

Next, we take the gradient with respect to $\boldsymbol{\lambda}$ and substitute the expression found in (60) to solve for $\boldsymbol{\lambda}$. Letting $M_\gamma = M + \gamma I$ we have,

$$J_r (M_\gamma^{-1} \mathbf{f} - M_\gamma^{-1} J_r^\top \boldsymbol{\lambda}) = \dot{\mathbf{y}} \quad (61)$$

Isolating $\boldsymbol{\lambda}$ we find

$$J_r M_\gamma^{-1} J_r^\top \boldsymbol{\lambda} = J_r M_\gamma^{-1} \mathbf{f} - \dot{\mathbf{y}} \quad (62)$$

In most cases $r \ll kN$, so J_r is a short-fat matrix. Thus if J_r is full rank, then $J_r M_\gamma^{-1} J_r^\top$ is symmetric positive definite. Hence

$$\boldsymbol{\lambda} = (J_r M_\gamma^{-1} J_r^\top)^{-1} J_r M_\gamma^{-1} \mathbf{f} - (J_r M_\gamma^{-1} J_r^\top)^{-1} \dot{\mathbf{y}} \quad (63)$$

Lastly, we use this expression for $\boldsymbol{\lambda}$ in Eq. (60), and we arrive at

$$\dot{\boldsymbol{\theta}} = \left[I - M_\gamma^{-1} J_r^\top (J_r M_\gamma^{-1} J_r^\top)^{-1} J_r \right] M_\gamma^{-1} \boldsymbol{f} + M_\gamma^{-1} J_r^\top (J_r M_\gamma^{-1} J_r^\top)^{-1} \dot{\boldsymbol{y}}. \quad (64)$$

Next, we turn our attention to the collocation shape-morphing equation. To begin we amend the collocation optimization problem in Section 2 by adding the same constraint as above. In this case we have

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \mathbb{R}^{N_k}} \quad & \frac{1}{2} \|\tilde{M}\dot{\boldsymbol{\theta}} - \tilde{\boldsymbol{f}}\|_2^2 + \frac{\gamma}{2} \|\dot{\boldsymbol{\theta}}\|_2^2, \\ \text{s.t.} \quad & J_r \dot{\boldsymbol{\theta}} = \dot{\boldsymbol{y}} \end{aligned} \quad (65)$$

To find the solution we follow the same procedure as above, first introducing the Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^r$ and constructing the constrained cost function,

$$\tilde{\mathcal{J}}_c(\boldsymbol{\theta}; \dot{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \frac{1}{2} \|\tilde{M}\dot{\boldsymbol{\theta}} - \tilde{\boldsymbol{f}}\|_2^2 + \frac{\gamma}{2} \|\dot{\boldsymbol{\theta}}\|_2^2 + \boldsymbol{\lambda}^\top (J_r \dot{\boldsymbol{\theta}} - \dot{\boldsymbol{y}}). \quad (66)$$

Next we expand the cost function, compute the derivative with respect to $\dot{\boldsymbol{\theta}}$,

$$\nabla_{\dot{\boldsymbol{\theta}}} \tilde{\mathcal{J}}_c = (\tilde{M}^\top \tilde{M} + \gamma I) \dot{\boldsymbol{\theta}} - \tilde{M}^\top \tilde{\boldsymbol{f}} + J_r^\top \boldsymbol{\lambda} \quad (67)$$

To find the critical point, we set this expression to zero and solve for $\dot{\boldsymbol{\theta}}$. Since $\gamma > 0$ we have that $\tilde{M}_\gamma = \tilde{M}^\top \tilde{M} + \gamma I$ is invertible, so

$$\dot{\boldsymbol{\theta}} = \tilde{M}_\gamma^{-1} \tilde{M}^\top \tilde{\boldsymbol{f}} - \tilde{M}_\gamma^{-1} J_r^\top \boldsymbol{\lambda} \quad (68)$$

Next, we compute the gradient with respect to $\boldsymbol{\lambda}$, which recovers the constraint, and then substitute our expression for $\dot{\boldsymbol{\theta}}$. If J_r is full rank, then $J_r \tilde{M}_\gamma^{-1} J_r$ is invertible, and we find that

$$\boldsymbol{\lambda} = (J_r \tilde{M}_\gamma^{-1} J_r)^{-1} (J_r \tilde{M}_\gamma^{-1} \tilde{M}^\top \tilde{\boldsymbol{f}} - \dot{\boldsymbol{y}}). \quad (69)$$

Combining these expressions and simplifying, we find that the continuous data assimilation algorithm for the collocation method is given by

$$\begin{aligned} \dot{\boldsymbol{\theta}} = & \left[I - \tilde{M}_\gamma^{-1} J_r^\top (J_r \tilde{M}_\gamma^{-1} J_r^\top)^{-1} \right] \tilde{M}_\gamma^{-1} \tilde{M}^\top \tilde{\boldsymbol{f}} \\ & + \tilde{M}_\gamma^{-1} J_r^\top (J_r \tilde{M}_\gamma^{-1} J_r^\top)^{-1} \dot{\boldsymbol{y}} \end{aligned} \quad (70)$$

B Neural Network Construction for Mixed Dirichlet-Neumann Boundary Conditions

In this section we discuss the construction of the neural network used for the advection diffusion equation in Section 4.3. This neural network is designed such that the mixed Dirichlet-Neumann boundary conditions of the AD equation are automatically satisfied by the network. This objective is achieved in three steps:

1. First, we define an extended computational domain Ω_e using copies of the physical rectangular domain Ω .
2. We then enforce periodic boundary conditions on the extended domain Ω_e .
3. Finally, we enforce odd and even symmetries into the network to ensure the mixed Dirichlet-Neumann boundary conditions are satisfied on the physical domain Ω .

Consider a general neural network $N(\mathbf{x}, \boldsymbol{\theta})$, which may be deep or shallow. The neural network is defined on the physical spatial domain $\mathbf{x} \in \Omega = [0, L] \times [0, H]$. Recall that the AD equation has homogeneous Dirichlet boundary conditions in z and homogeneous Neumann boundary conditions in x . First, we define the extended computational domain $\Omega_e = [-L, L] \times [-H, H]$. We impose periodic boundary conditions over the entire computational domain Ω_e using the methodology presented in Ref. [15]. Namely, in each node of the network $N(\mathbf{x}, \boldsymbol{\theta})$, we apply a change of variables through the function,

$$\mathbf{s}_i(\mathbf{x}) = \left[\sin\left(\frac{\pi}{L}x + c_i^x\right), \sin\left(\frac{\pi}{H}z + c_i^z\right) \right]^\top. \quad (71)$$

For example, in AD equation where we consider a shallow neural network with a hyperbolic tangent activation function, we have

$$N_p(\mathbf{x}, \boldsymbol{\theta}) = \sum_{i=1}^N a_i \tanh(\mathbf{w}_i \cdot \mathbf{s}_i(\mathbf{x}) + b_i). \quad (72)$$

This neural network has periodic boundary conditions on the extended domain Ω_e .

To enforce the mixed Dirichlet-Neumann boundary conditions on the physical domain Ω , we exploit the inherent symmetries of even and odd functions. For a continuous odd function, $f(-x) = -f(x)$, we have $f(0) = 0$. Additionally, for a continuously differentiable even function, $f(-x) = f(x)$, its derivative must be odd, and therefore, $f'(0) = 0$. As a result, if the construction of the neural network is odd with respect to z and even with respect to x , then it satisfies the correct boundary conditions of the AD equation.

Finally, we introduce these symmetries by using the neural network $N_p(\mathbf{x}, \boldsymbol{\theta})$ to define the SMS solution,

$$\hat{u}(\mathbf{x}, \boldsymbol{\theta}) = N_p(x, z, \boldsymbol{\theta}) - N_p(x, -z, \boldsymbol{\theta}) + N_p(-x, z, \boldsymbol{\theta}) - N_p(-x, -z, \boldsymbol{\theta}). \quad (73)$$

It is straightforward to verify that \hat{u} is odd with respect to z and even with respect to x . This, together with periodicity \hat{u} on the extended computational domain Ω_e , ensures that the SMS solution \hat{u} satisfies the mixed homogeneous Dirichlet-Neumann boundary conditions on the physical domain Ω .

References

- [1] W. Anderson and M. Farazmand. Evolution of nonlinear reduced-order solutions for PDEs with conserved quantities. *SIAM Journal on Scientific Computing*, 44(1):A176–A197, 2022.
- [2] W. Anderson and M. Farazmand. Shape-morphing reduced-order models for nonlinear Schrödinger equations. *Nonlinear Dynamics*, 108(4):2889–2902, 2022.

- [3] W. Anderson and M. Farazmand. Fast and scalable computation of shape-morphing nonlinear solutions with application to evolutionary neural networks. *Journal of Computational Physics*, 498:112649, 2024.
- [4] W. Anderson and M. Farazmand. Fisher information and shape-morphing modes for solving the Fokker–Planck equation in higher dimensions. *Applied Mathematics and Computation*, 467:128489, 2024.
- [5] E. Andersson, J. Haseler, P. Undén, P. Courtier, G. Kelly, D. Vasiljevic, C. Brankovic, C. Gaffard, A. Hollingsworth, C. Jakob, P. Janssen, E. Klinker, A. Lanzinger, M. Miller, F. Rabier, A. Simmons, B. Strauss, P. Viterbo, C. Cardinali, and J.-N.l Thépaut. The ECMWF implementation of three-dimensional variational assimilation (3D-VAR). III: Experimental results. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1831–1860, 1998.
- [6] A. Apte, M. Hairer, A. M. Stuart, and J. Voss. Sampling the posterior: An approach to non-Gaussian data assimilation. *Physica D: Nonlinear Phenomena*, 230(1):50–64, 2007.
- [7] A. Attia, A. Alexanderian, and A. K Saibaba. Goal-oriented optimal design of experiments for large-scale Bayesian linear inverse problems. *Inverse Problems*, 34(9):095009, 2018.
- [8] J. Berman and B. Peherstorfer. Randomized sparse neural Galerkin schemes for solving evolution equations with deep networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, volume 36, pages 4097–4114. Curran Associates, Inc., 2023.
- [9] J. Bruna, B. Peherstorfer, and Vanden-Eijnden E. Neural Galerkin schemes with active learning for high-dimensional evolution equations. *Journal of Computational Physics*, 496:112588, 2024.
- [10] A. Chabchoub, N. P. Hoffmann, and N. Akhmediev. Rogue wave observation in a water wave tank. *Phys. Rev. Lett.*, 106(20):204502, 2011.
- [11] X. Chen and T. Yamamoto. Newton-like methods for solving underdetermined nonlinear equations with nondifferentiable terms. *Journal of Computational and Applied Mathematics*, 55(3):311–324, 1994.
- [12] P. Courtier, E. Andersson, W. Heckley, D. Vasiljevic, M. Hamrud, A. Hollingsworth, F. Rabier, M. Fisher, and J. Pailleux. The ECMWF implementation of three-dimensional variational assimilation (3d-var). I: Formulation. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1783–1807, 1998.
- [13] W. Cousins and T. P. Sapsis. Reduced-order precursors of rare events in unidirectional nonlinear water waves. *J. Fluid Mech.*, 790:368–388, 3 2016.
- [14] S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002.
- [15] Y. Du and T. A. Zaki. Evolutional deep neural network. *Phys. Rev. E*, 104:045303, 2021.
- [16] M. Farazmand. Sparse discrete empirical interpolation method: State estimation from few sensors. *SIAM Journal on Scientific Computing (in press)*, *arXiv:2401.16411*, 2024.

- [17] M. Farazmand and G. Haller. Computing Lagrangian coherent structures from their variational theory. *Chaos*, 22:013128, 2012.
- [18] M. Farazmand and T. P. Sapsis. Reduced-order prediction of rogue waves in two-dimensional deep-water waves. *J. Comput. Phys.*, 340:418 – 434, 2017.
- [19] M. Finzi, A. Potapczynski, M. Choptuik, and A. G. Wilson. A stable and scalable method for solving initial value PDEs with neural networks, 2023.
- [20] N. Gaby, X. Ye, and H. Zhou. Neural control of parametric solutions for high-dimensional evolution PDEs. *SIAM Journal on Scientific Computing*, 46(2):C155–C185, 2024.
- [21] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.
- [22] Z. T. Hilliard and M. Farazmand. Enforcing conserved quantities in Galerkin truncation and finite volume discretization. *Nonlinear Dynamics*, 112(16):14051–14069, 2024.
- [23] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.
- [24] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95–108, 03 1961.
- [25] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nat Rev Phys*, 3:422–440, 2021.
- [26] M. Kast and J. S. Hesthaven. Positional embeddings for solving PDEs with evolutionary deep neural networks. *Journal of Computational Physics*, 508:112986, 2024.
- [27] C. T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics, 1999.
- [28] H. Kim and T. A. Zaki. Multi evolutionary deep neural networks (Multi-EDNN). *arXiv (arXiv:2407.12293)*, 2024.
- [29] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [30] Y. Kuramoto. Diffusion-induced chaos in reaction systems. *Progress of Theoretical Physics Supplement*, 64:346–367, 02 1978.
- [31] K. Law, A. Stuart, and K. Zygalakis. *Data Assimilation: A Mathematical Introduction*. Texts in Applied Mathematics. Springer, 2015.
- [32] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat Mach Intell*, 3:218–229, 2021.
- [33] B. A. McElhoe. An assessment of the navigation and course corrections for a manned flyby of Mars or Venus. *IEEE Transactions on Aerospace and Electronic Systems*, AES-2(4):613–623, 1966.

- [34] F. Pes and G. Rodriguez. The minimal-norm Gauss-Newton method and some of its regularized variants. *ETNA - Electronic Transactions on Numerical Analysis*, 53:459–480, 01 2020.
- [35] F. Rabier, A. McNally, E. Andersson, P. Courtier, P. Undén, J. Eyre, A. Hollingsworth, and F. Bouttier. The ECMWF implementation of three-dimensional variational assimilation (3D-Var). II: Structure functions. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1809–1829, 1998.
- [36] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686 – 707, 2019.
- [37] S. Reich and C. Cotter. *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, 2015.
- [38] A. Rieder. On the regularization of nonlinear ill-posed problems via inexact Newton iterations. *Inverse Problems*, 15(1):309, feb 1999.
- [39] V. P. Ruban. Gaussian variational ansatz in the problem of anomalous sea waves: Comparison with direct numerical simulation. *Journal of Experimental and Theoretical Physics*, 120(5):925–932, 2015.
- [40] S. F. Schmidt. Application of state-space methods to navigation problems. volume 3 of *Advances in Control Systems*, pages 293–340. Elsevier, 1966.
- [41] P. Schwerdtner, P. Schulze, J. Berman, and B. Peherstorfer. Nonlinear embeddings for conserving Hamiltonians and other quantities with neural Galerkin schemes. *SIAM Journal on Scientific Computing*, 46(5):C583–C607, 2024.
- [42] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212:271–304, 2005.
- [43] G. I. Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames—I. Derivation of basic equations. *Acta Astronautica*, 4(11):1177–1206, 1977.
- [44] N. Sukumar and A. Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 389:114333, 2022.
- [45] Y. Trémolet. Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504, 2006.
- [46] A. Virmaux and K. Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.
- [47] V. E. Zakharov. Stability of periodic waves of finite amplitude on the surface of a deep fluid. *Journal of Applied Mechanics and Technical Physics*, 9(2):190–194, 1968.